

MELSEC FX-Familie

Speicherprogrammierbare Steuerungen

Programmieranleitung

**FX1S, FX1N,
FX2N, FX2NC,
FX3G, FX3GC, FX3GE,
FX3S,
FX3U, FX3UC**



Zu diesem Handbuch

Die in diesem Handbuch vorliegenden Texte, Abbildungen, Diagramme und Beispiele dienen ausschließlich zur Erläuterung der Installation, Bedienung, Programmierung und Anwendung der speicherprogrammierbaren Steuerungen der MELSEC FX1S-, FX1N-, FX2N-,FX2NC-, FX3G-, FX3GC-, FX3GE-, FX3S-, FX3U- und FX3UC-Serie.

Sollten sich Fragen zur Programmierung und zum Betrieb der in diesem Handbuch beschriebenen Geräte ergeben, zögern Sie nicht, Ihr zuständiges Verkaufsbüro oder einen Ihrer Vertriebspartner (siehe Umschlagrückseite) zu kontaktieren.
Aktuelle Informationen sowie Antworten auf häufig gestellte Fragen erhalten Sie über das Internet (<https://de3a.mitsubishielectric.com>).

Ohne vorherige ausdrückliche schriftliche Genehmigung der MITSUBISHI ELECTRIC EUROPE B.V. dürfen keine Auszüge dieses Handbuchs vervielfältigt, in einem Informationssystem gespeichert oder weiter übertragen werden.

Die MITSUBISHI ELECTRIC EUROPE B.V. behält sich vor, jederzeit technische Änderungen dieses Handbuchs ohne besondere Hinweise vorzunehmen.

Programmieranleitung
Speicherprogrammierbare Steuerungen der MELSEC-FX-Familie
FX1S, FX1N, FX2N, FX2NC, FX3G, FX3GC, FX3GE, FX3S, FX3U und FX3UC
Artikel-Nr.: 136748

Version			Änderungen / Ergänzungen / Korrekturen
A	03/2001	pdp-dk	—
B	03/2002	pdp-dk	Erläuterungen zu den Positionieranweisungen eingefügt (Kap. 7.6.1 bis 7.6.5) Erweiterung der Funktionalität für die RD3A-Anweisung (FNC176, Kap. 7.9) Neue Anweisung: EXTR (FNC180) in Kap. 7.10 Korrekturen bei den folgenden Anweisungen: PWM (FNC58), PLSY (FNC57), PLSR (FNC59), INCD (FNC63), DSW (FNC72), SEGL (FROM74), FROM (FNC78), TO (FNC79), RS (FNC80), CCD (FNC84), VRRD (FNC85), VRSC (FNC86), PID (FNC88) Ergänzungen im Kap. 9 (Sondermerker und -register) Ergänzungen im Kap. 10 (Programmfehler)
C	03/2003	pdp-dk	Aufnahme der Steuerungen der FX2NC-Serie
D	04/2004	pdp-dk	Korrektur der Anzahl der bei der FX1N im EEPROM gespeicherten Datenregister (Seite A-7) Korrektur der Anzahl der Index-Register bei der FX1N von 2 auf 16 (Seite A-7)
E	03/2006	pdp-dk	Aufnahme der Steuerungen der FX3U-Serie
F	08/2007	pdp-dk	Korrektur des Wertes für D8008 in Abschnitt 8.10 von „-4“ in „-1“
G	04/2008	pdp-dk	Korrektur der Fußnote zu Tab. 3-12 auf Seite 3-16
H	11/2009	pdp-dk	Berücksichtigung der Steuerungen der FX3G- und der FX3UC-Serie Ergänzungen im Kap. 9 (Sondermerker und -register)
I	12/2013	pdp-dk	Berücksichtigung der Steuerungen der FX3GC-, FX3GE- und der FXS-Serie Neue Anweisungen: MEP, MEF, IVMC (FNC275), ADPRW (FNC276), FLCRT (FNC300), FLDEL (FNC301), FLWR (FNC302), FLRD (FNC303), FLCMD (FNC304), FLSTRD (FNC305) Beschreibung der PID-Anweisung (PID88) erweitert Ergänzungen im Abschnitt 8.8 (RUN-/STOP-Umschaltung)
J	01/2015	pdp-dk	Korrektur der Tabelleneinträge für die Sonderregister D8136/D8137 und D8140/D8141 auf Seite 9-38

Sicherheitshinweise

Zielgruppe

Dieses Handbuch richtet sich ausschließlich an anerkannt ausgebildete Elektrofachkräfte, die mit den Sicherheitsstandards der Automatisierungstechnik vertraut sind. Projektierung, Installation, Inbetriebnahme, Wartung und Prüfung der Geräte dürfen nur von einer anerkannt ausgebildeten Elektrofachkraft, die mit den Sicherheitsstandards der Automatisierungstechnik vertraut ist, ausgeführt werden. Eingriffe in die Hard- und Software unserer Produkte, soweit sie nicht in diesem Handbuch beschrieben sind, dürfen nur durch unser Fachpersonal vorgenommen werden.

Bestimmungsgemäßer Gebrauch

Die speicherprogrammierbaren Steuerungen der FX1S-, FX1N-, FX2N-, FX2NC-, FX3G-, FX3GC-, FX3GE-, FX3S-, FX3U- und FX3UC-Serie sind nur für die Einsatzbereiche vorgesehen, die in diesem Handbuch beschrieben sind. Achten Sie auf die Einhaltung aller im Handbuch angegebenen Kenndaten. Die Produkte wurden unter Beachtung der Sicherheitsnormen entwickelt, gefertigt, geprüft und dokumentiert. Unqualifizierte Eingriffe in die Hard- oder Software bzw. Nichtbeachtung der in diesem Handbuch angegebenen oder am Produkt angebrachten Warnhinweise können zu schweren Personen- oder Sachschäden führen. Es dürfen nur von MITSUBISHI ELECTRIC empfohlene Zusatz- bzw. Erweiterungsgeräte in Verbindung mit den speicherprogrammierbaren Steuerungen der FX1S-, FX1N-, FX2N-, FX2NC-, FX3G-, FX3GC-, FX3GE-, FX3S-, FX3U- und FX3UC-Serie verwendet werden.

Jede andere darüber hinausgehende Verwendung oder Benutzung gilt als nicht bestimmungsgemäß.

Sicherheitsrelevante Vorschriften

Bei der Projektierung, Installation, Inbetriebnahme, Wartung und Prüfung der Geräte müssen die für den spezifischen Einsatzfall gültigen Sicherheits- und Unfallverhütungsvorschriften beachtet werden.

Es müssen besonders folgende Vorschriften (ohne Anspruch auf Vollständigkeit) beachtet werden:

- VDE-Vorschriften
 - VDE 0100
Bestimmungen für das Errichten von Starkstromanlagen mit einer Nennspannung bis 1000V
 - VDE 0105
Betrieb von Starkstromanlagen
 - VDE 0113
Elektrische Anlagen mit elektronischen Betriebsmitteln
 - VDE 0160
Ausrüstung von Starkstromanlagen und elektrischen Betriebsmitteln
 - VDE 0550/0551
Bestimmungen für Transformatoren
 - VDE 0700
Sicherheit elektrischer Geräte für den Hausgebrauch und ähnliche Zwecke
 - VDE 0860
Sicherheitsbestimmungen für netzbetriebene elektronische Geräte und deren Zubehör für den Hausgebrauch und ähnliche Zwecke.
- Brandverhütungsvorschriften

- Unfallverhütungsvorschrift
 - VBG Nr.4
Elektrische Anlagen und Betriebsmittel

Gefahrenhinweise

Die einzelnen Hinweise haben folgende Bedeutung:



GEFAHR:

Bedeutet, dass eine Gefahr für das Leben und die Gesundheit des Anwenders besteht, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.



ACHTUNG:

Bedeutet eine Warnung vor möglichen Beschädigungen des Gerätes oder anderen Sachwerten, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Allgemeine Gefahrenhinweise und Sicherheitsvorkehrungen

Die folgenden Gefahrenhinweise sind als generelle Richtlinie für den Umgang mit der SPS in Verbindung mit anderen Geräten zu verstehen. Diese Hinweise müssen Sie bei der Projektierung, Installation und Betrieb einer Steuerungsanlage unbedingt beachten.



GEFAHR

- **Die im spezifischen Einsatzfall geltenden Sicherheits- und Unfallverhütungsvorschriften sind zu beachten. Der Einbau, die Verdrahtung und das Öffnen der Baugruppen, Bauteile und Geräte muss im spannungslosen Zustand erfolgen.**
- **Baugruppen, Bauteile und Geräte müssen in einem berührungssicheren Gehäuse mit einer bestimmungsgemäßen Abdeckung und Schutzeinrichtung installiert werden.**
- **Bei Geräten mit einem ortsfesten Netzanschluss muss ein allpoliger Netztrennschalter oder eine Sicherung in die Gebäudeinstallation eingebaut werden.**
- **Überprüfen Sie spannungsführende Kabel und Leitungen, mit denen die Geräte verbunden sind, regelmäßig auf Isolationsfehler oder Bruchstellen. Bei Feststellung eines Fehlers in der Verkabelung müssen Sie die Geräte und die Verkabelung sofort spannungslos schalten und die defekte Verkabelung ersetzen.**
- **Überprüfen Sie vor der Inbetriebnahme, ob der zulässige Netzspannungsbereich mit der örtlichen Netzspannung übereinstimmt.**
- **Treffen Sie die erforderlichen Vorkehrungen, um nach Spannungseinbrüchen und -ausfällen ein unterbrochenes Programm ordnungsgemäß wieder aufnehmen zu können. Dabei dürfen auch kurzzeitig keine gefährlichen Betriebszustände auftreten. Gegebenenfalls ist ein „NOT-AUS“ zu erzwingen.**
- **NOT-AUS-Einrichtungen gemäß EN 60204/IEC 204 VDE 0113 müssen in allen Betriebsarten der Steuerung wirksam bleiben. Ein Entriegeln der NOT-AUS-Einrichtung darf keinen unkontrollierten oder undefinierten Wiederanlauf bewirken.**
- **Damit ein Leitungs- oder Aderbruch auf der Signalseite nicht zu undefinierten Zuständen in der Steuerung führen kann, sind hard- und softwareseitig entsprechende Sicherheitsvorkehrungen zu treffen.**

Inhaltsverzeichnis

1	Einleitung	
1.1	Allgemeines	1-1
1.2	Verwendbare Programmiergeräte/Bediengeräte	1-2
2	Grundlagen der Programmierung	
2.1	Programmverarbeitung in der SPS	2-1
2.1.1	Prozessabbildverfahren	2-2
2.1.2	Signalverarbeitung in der SPS im Unterschied zur verbindungs- programmierten Steuerung	2-3
2.2	Steuerungsanweisungen	2-4
2.2.1	Aufbau einer Steuerungsanweisung	2-4
2.2.2	Operanden	2-5
2.2.3	Darstellungsarten von Steuerungsanweisungen	2-6
2.2.4	Zuordnungsliste und Beschaltung der SPS	2-8
3	Operanden	
3.1	Übersicht der Operanden	3-1
3.2	Ein- und Ausgänge	3-2
3.2.1	Ein- und Ausgänge adressieren	3-2
3.2.2	Ein- und Ausgänge programmieren	3-4
3.3	Merker	3-5
3.3.1	Merker adressieren	3-5
3.3.2	Merker programmieren	3-6
3.4	Timer	3-7
3.4.1	Adressierung der Timer	3-7
3.4.2	Programmierung der Timer	3-9
3.4.3	Vorgabe eines Zeitwertes mit den integrierten Potentiometern der Steue- rungen der FX1S-, FX1N-, FX3G-, FX3GE- und FX3S-Serie	3-10
3.4.4	Genauigkeit der Timer	3-11
3.4.5	Remanente Timer	3-11

3.5	Counter	3-12
3.5.1	16-Bit-Counter	3-13
3.5.2	32-Bit-Counter	3-15
3.5.3	32-Bit-High-Speed-Counter	3-17
3.6	Schrittstatus	3-23
3.6.1	Schrittstatusoperanden adressieren	3-23
3.7	Konstanten	3-25
3.7.1	Dezimal- und Hexadezimalkonstanten	3-25
3.7.2	Konstanten mit Gleitkommazahlen	3-25
3.7.3	Konstante Zeichenfolgen	3-25
3.8	Register	3-26
3.8.1	Einteilung der Register	3-26
3.8.2	Aufbau der Register	3-27
3.8.3	Adressierung der Register	3-28
3.8.4	Verwendung der Sonderregister	3-30
3.8.5	Extern veränderbare Register	3-30
3.8.6	Indexvergabe	3-31
3.8.7	Einsatz der File-Register	3-33
3.8.8	Erweiterte Register und erweiterte File-Register	3-34
3.8.9	Zahlenwertbereiche von Registern	3-35
3.8.10	Zahlendarstellungen	3-36
3.9	Pointer	3-42
3.9.1	Pointer adressieren	3-42
3.9.2	Nesting-Ebenen	3-42
3.10	Interrupt-Pointer	3-42
3.10.1	Interrupt-Pointer adressieren	3-42
3.11	Nesting	3-46
3.11.1	Nesting-Operanden adressieren	3-46
3.12	Pufferspeicher eines Sondermoduls	3-47

4 Grundbefehlssatz

4.1	Allgemeine Hinweise	4-1
4.1.1	Erläuterung der Grundbefehlssatztabellen	4-1
4.1.2	Anzahl der Programmschritte bei einer FX3G, FX3GC, FX3GE, FX3S, FX3U oder FX3UC	4-5
4.2	Beginn von Verknüpfungen (LD, LDI)	4-6

4.3	Ausgabe eines Verknüpfungsergebnisses (OUT)	4-8
4.4	UND-Verknüpfungen (AND, ANI)	4-10
4.5	ODER-Verknüpfungen (OR, ORI)	4-12
4.6	Gepulster Beginn von Verknüpfungen (LDP, LDF)	4-14
4.7	Gepulste UND-Verknüpfungen (ANDP, ANDF)	4-16
4.8	Gepulste ODER-Verknüpfungen (ORP, ORF)	4-18
4.9	UND-Block-Verknüpfung (ANB)	4-20
4.10	ODER-Block-Verknüpfung (ORB)	4-21
4.11	Verknüpfungsergebnis verarbeiten (MPS, MRD, MPP)	4-22
4.12	Kontrollbedingung setzen/zurücksetzen (MC, MCR)	4-25
4.13	Setzen und Rücksetzen von Operanden (SET, RST)	4-28
4.14	Erzeugen eines einmaligen Impulses (PLS, PLF)	4-30
4.15	Inversion von Verarbeitungsergebnissen (INV)	4-32
4.16	Impuls aus Verknüpfungsergebnis (MEP, MEF)	4-33
4.17	Leerzeile im Programm (NOP)	4-35
4.18	SPS-Programmende (END)	4-36
4.19	Programmbeispiele	4-37
4.19.1	Abfrage eines Eingangs	4-38
5	STL-Anweisung	
5.1	Allgemeine Hinweise	5-1
5.1.1	Anwendungsbeispiel zum Einsatz der STL-Anweisung	5-2
5.1.2	Schematischer Ablauf einer Schrittsteuerung	5-3
5.1.3	Darstellung einer Ablaufsteuerung in einem Flussdiagramm	5-4
5.2	STL-Anweisung programmieren	5-5
5.3	Schrittstatus initialisieren	5-10
5.4	STL-Verzweigungen	5-11
5.4.1	Einfachverlauf	5-11
5.4.2	Selektive Verzweigung	5-12
5.4.3	Parallele Verzweigung	5-14
5.4.4	Kombination aus selektiver und paralleler Verzweigung	5-16
5.4.5	Leerstatus programmieren	5-17
5.4.6	Sprungverzweigung	5-18

5.5 Beispiel für eine Be- und Entladekontrolle5-20

5.6 Beispiel für einen Transportier- und Sortiervorgang5-22

6 Applikationsanweisungen

6.1 Allgemeine Hinweise6-1

6.1.1 Erläuterung zur Beschreibung der Applikationsanweisungen 6-1

6.1.2 Beschreibung der Operanden6-2

6.1.3 Zusammenfassen von Bit-Operanden.6-3

6.1.4 Datenstruktur.6-5

6.1.5 Ausführung von Applikationsanweisungen 6-6

6.1.6 Einsatz der Index-Register V, Z.6-7

6.1.7 Bedeutung der Flags.6-8

6.1.8 Programmablauffehler bei der Ausführung von Applikationsanweisungen6-8

6.1.9 32-Bit-Anweisungen6-8

6.1.10 Übersicht der Applikationsanweisungen6-9

6.2 Programmablaufanweisungen6-11

6.2.1 Sprung innerhalb eines Programms (CJ) 6-12

6.2.2 Aufruf eines Unterprogramms (CALL) 6-16

6.2.3 Ende eines Unterprogramms (SRET) 6-17

6.2.4 Einsatz eines Interrupt-Programms (IRET, EI, DI) 6-18

6.2.5 Ende eines Programmbereichs (FEND) 6-22

6.2.6 Watch-Dog-Timer auffrischen (WDT) 6-23

6.2.7 Programmteile wiederholen (FOR, NEXT) 6-25

6.3 Vergleichs- und Transferanweisungen.6-27

6.3.1 Numerische Daten vergleichen (CMP, DCMP) 6-28

6.3.2 Numerische Datenbereiche vergleichen (ZCP, DZCP) 6-30

6.3.3 Datentransfer (MOV, DMOV)6-32

6.3.4 Shift-Transfer (SMOV)6-33

6.3.5 Kopieren und invertieren (CML)6-36

6.3.6 Block-Transfer (BMOV)6-37

6.3.7 Transfer von gleichen Daten (FMOV)6-38

6.3.8 Austausch von Daten (XCH)6-39

6.3.9 BCD-Konvertierung (BCD, DBCD)6-41

6.3.10 Binär-Konvertierung (BIN, DBIN)6-43

6.4	Arithmetische Anweisungen	6-45
6.4.1	Addition numerischer Daten (ADD, DADD)	6-45
6.4.2	Subtraktion numerischer Daten (SUB, DSUB)	6-47
6.4.3	Multiplikation numerischer Daten (MUL, DMUL)	6-49
6.4.4	Division numerischer Daten (DIV, DDIV)	6-51
6.4.5	Inkrementieren (INC, DINC)	6-53
6.4.6	Dekrementieren (DEC)	6-54
6.4.7	Logische UND-Verknüpfung binärer Daten (WAND, DAND)	6-55
6.4.8	Logische ODER-Verknüpfung binärer Daten (WOR, DOR)	6-56
6.4.9	Logische Exklusiv-ODER-Verknüpfung binärer Daten (WXOR, DXOR)	6-57
6.4.10	Negation von Daten (NEG)	6-58
6.5	Verschiebeanweisungen	6-59
6.5.1	Rotation nach rechts (ROR)	6-60
6.5.2	Rotation nach links (ROL)	6-61
6.5.3	Rotieren von Bits nach rechts (RCR)	6-62
6.5.4	Rotieren von Bits nach links (RCL)	6-63
6.5.5	Binäre Daten bitweise verschieben (SFTR, SFTL)	6-64
6.5.6	Daten wortweise nach rechts verschieben (WSFR)	6-66
6.5.7	Daten wortweise nach links verschieben (WSFL)	6-67
6.5.8	Schreiben in einen FIFO-Speicher (SFWR)	6-68
6.5.9	Lesen aus einem FIFO-Speicher (SFRD)	6-69
6.6	Datenoperationen	6-71
6.6.1	Operandenbereiche zurücksetzen (ZRST)	6-72
6.6.2	Daten decodieren (DECO)	6-73
6.6.3	Daten codieren (ENCO)	6-75
6.6.4	Ermittlung gesetzter Bits (SUM)	6-77
6.6.5	Überprüfen eines Bits (BON)	6-78
6.6.6	Ermittlung von Durchschnittswerten (MEAN)	6-79
6.6.7	Starten eines Zeitintervalls (ANS)	6-80
6.6.8	Rücksetzen von Anzeige-Bits (ANR)	6-81
6.6.9	Ermittlung der Quadratwurzel (SQR)	6-82
6.6.10	Umwandlung des Zahlenformats (FLT)	6-83

6.7	High-Speed-Anweisungen	6-84
6.7.1	Ein- und Ausgänge auffrischen (REF)	6-85
6.7.2	Einstellen der Eingangfilter (REFF).	6-87
6.7.3	Einlesen einer Matrix (MTR)	6-89
6.7.4	Setzen und Rücksetzen durch High-Speed-Counter (DHSCS, DHSCR).	6-92
6.7.5	Bereichsvergleich (DHSZ).	6-94
6.7.6	Geschwindigkeitserkennung (SPD, DSPD).	6-99
6.7.7	Ausgabe einer definierten Anzahl von Impulsen (PLSY, DPLSY)	6-104
6.7.8	Impulsausgabe mit Modulation der Impulsweite (PWM).	6-106
6.7.9	Ausgabe einer bestimmten Anzahl von Impulsen (PLSR)	6-108
6.8	Anwendungsbezogene Anweisungen	6-111
6.8.1	Schrittstatus initialisieren (IST)	6-112
6.8.2	Suchanweisung (SER)	6-118
6.8.3	Absoluter Counter-Vergleich (ABSD)	6-120
6.8.4	Inkrementaler Counter-Vergleich (INCD)	6-122
6.8.5	Teaching-Timer (TTMR)	6-124
6.8.6	Sonder-Timer (STMR).	6-125
6.8.7	Flip-Flop-Funktion (ALT)	6-127
6.8.8	Rampenfunktion (RAMP)	6-129
6.8.9	Rundtisch-Positionierung (ROTC).	6-131
6.8.10	Sortieranweisung (SORT)	6-134

7 Spezielle Anweisungen

7.1	Allgemeine Hinweise	7-1
7.1.1	Gesamtübersicht der speziellen Anweisungen	7-1
7.2	Ein-/Ausgabeanweisungen	7-5
7.2.1	Zehnertastatur (TKY)	7-6
7.2.2	Hexadezimale Tastatur (HKY).	7-8
7.2.3	Digitaler Schalter (DSW)	7-11
7.2.4	7-Segment-Anzeige (SEGD).	7-13
7.2.5	7-Segment-Anzeige mit Latch (SEGL)	7-14
7.2.6	7-Segment-Anzeige mit zusätzlichen Tasten (ARWS)	7-17
7.2.7	ASCII-Konvertierung (ASC).	7-19
7.2.8	Datenausgabe über die Ausgänge (PR)	7-21

7.3	Datenaustausch mit Sondermodulen	7-23
7.3.1	Auslesen von Daten aus einem Sondermodul (FROM)	7-25
7.3.2	Schreiben von Daten in ein Sondermodul (TO)	7-27
7.4	Serielle Kommunikation	7-29
7.4.1	Serielle Datenübertragung (RS)	7-30
7.4.2	Umlegen von Eingängen oder Merckern (PRUN)	7-36
7.4.3	ASCII-Umwandlung (ASCI)	7-37
7.4.4	Hexadezimal-Umwandlung (HEX)	7-39
7.4.5	Summen- und Paritätsprüfung (CCD)	7-41
7.4.6	Einlesen von Sollwerten vom FX□-8AV-BD (VRRD)	7-43
7.4.7	Einlesen von Schalterstellungen vom FX□-8AV-BD (VRSC)	7-44
7.4.8	Serielle Datenübertragung (RS2)	7-45
7.5	PID-Regelung (PID)	7-47
7.6	Datentransfer mit Index-Register	7-54
7.6.1	Inhalt der Index-Register sichern (ZPUSH)	7-54
7.6.2	Inhalt der Index-Register wiederherstellen (ZPOP)	7-57
7.7	Anweisungen mit Gleitkommazahlen	7-58
7.7.1	Vergleich von Gleitkommazahlen (DECMP)	7-59
7.7.2	Vergleich von Gleitkommazahlen mit einem Bereich (DEZCP)	7-60
7.7.3	Transfer von Gleitkommazahlen (DEMOV)	7-62
7.7.4	Gleitkommazahl in Zeichenfolge wandeln (DESTR)	7-63
7.7.5	Zeichenfolge in Gleitkommazahl wandeln (DEVAL)	7-71
7.7.6	Umwandlung des Gleitkommaformats ins wissenschaftliche Zahlenformat (DEBCD)	7-76
7.7.7	Umwandlung wissenschaftliches Zahlenformat ins Gleitkommaformat (DEBIN)	7-77
7.7.8	Addition von Gleitkommazahlen (DEADD)	7-78
7.7.9	Subtraktion von Gleitkommazahlen (DESUB)	7-79
7.7.10	Multiplikation von Gleitkommazahlen (DEMUL)	7-80
7.7.11	Division von Gleitkommazahlen (DEDIV)	7-81
7.7.12	Gleitkommazahl als Exponent zur Basis e (DEXP)	7-82
7.7.13	Logarithmus-naturalis-Berechnung (DLOGE)	7-84
7.7.14	Berechnung des dekadischen Logarithmus (DLOG10)	7-86
7.7.15	Quadratwurzel aus Gleitkommazahlen (DESQR)	7-88
7.7.16	Vorzeichenumkehr von Gleitkommazahlen (DNEG)	7-89
7.7.17	Umwandlung des Gleitkommaformats ins Dezimal-Format (INT)	7-90
7.7.18	Sinusberechnung mit Gleitkommazahlen (DSIN)	7-91
7.7.19	Cosinusberechnung mit Gleitkommazahlen (DCOS)	7-92

7.7.20	Tangensberechnung mit Gleitkommazahlen (DTAN)	7-93
7.7.21	Arcussinusberechnung mit Gleitkommazahlen (DASIN)	7-94
7.7.22	Arcuscosinusberechnung mit Gleitkommazahlen (DACOS)	7-96
7.7.23	Arcustangensberechnung mit Gleitkommazahlen (DATAN)	7-98
7.7.24	Umrechnung von Grad in Radiant (DRAD)	7-100
7.7.25	Umrechnung von Radiant in Grad (DDEG)	7-102
7.8	Datenverarbeitungsanweisungen	7-105
7.8.1	Summe der Inhalte von Wortoperanden bilden (WSUM)	7-106
7.8.2	Daten in Wortoperanden in Bytes aufteilen (WTOB)	7-108
7.8.3	Wortoperanden aus einzelnen Bytes bilden (BTOW)	7-110
7.8.4	Gruppen von 4 Bits zu Wortoperanden zusammenfassen (UNI)	7-112
7.8.5	Wortoperanden in Gruppen von 4 Bit aufteilen (DIS)	7-114
7.8.6	High-Low-Byte-Tausch (SWAP)	7-116
7.8.7	Daten in Tabelle sortieren (SORT2)	7-117
7.9	Positionieranweisungen	7-120
7.9.1	Hinweise zum Einsatz der Positionieranweisungen bei FX1S und FX1N	7-120
7.9.2	Impulsausgabe an den Servoverstärker	7-121
7.9.3	Operanden für die Positionierung	7-122
7.9.4	Anschluss an einem Servoverstärker	7-123
7.9.5	Beispielprogramm	7-124
7.9.6	Referenzpunktfahrt mit Annäherungsschalter (DSZR)	7-131
7.9.7	Positionierung durch Interrupt (DVIT)	7-132
7.9.8	Positionierung nach Datentabelle (TBL)	7-133
7.9.9	Lesen der absoluten Ist-Position (DABS)	7-134
7.9.10	Referenzpunkt anfahren (ZRN)	7-135
7.9.11	Ausgabe von Impulsen mit variabler Frequenz (PLSV)	7-137
7.9.12	Positionieren auf einen Inkrementalwert (DRVI)	7-139
7.9.13	Positionieren auf einen Absolutwert (DRVA)	7-141
7.10	Anweisungen für die integrierte Uhr der SPS	7-143
7.10.1	Vergleich von Uhr-Daten (TCMP)	7-144
7.10.2	Vergleich von Uhr-Daten mit einem Bereich (TZCP)	7-146
7.10.3	Addition von Uhr-Daten (TADD)	7-148
7.10.4	Subtraktion von Uhr-Daten (TSUB)	7-150
7.10.5	Wandlung der Einheit „Stunden“ in „Sekunden“ (HTOS)	7-152
7.10.6	Wandlung der Einheit „Sekunden“ in „Stunden“ (STOH)	7-154
7.10.7	Lesen der Uhrzeit und des Datums (TRD)	7-156
7.10.8	Interne Uhr der SPS stellen (TWR)	7-158

7.10.9	Betriebsstundenzähler (HOUR)	7-159
7.11	Gray-Code-Anweisungen	7-161
7.11.1	Umwandlung Integer in Gray-Code (GRY)	7-161
7.11.2	Umwandlung Gray-Code in Integer (GBIN)	7-162
7.12	Datenaustausch mit Analogmodulen.	7-163
7.12.1	Lesen der analogen Eingangswerte (RD3A)	7-163
7.12.2	Schreiben eines analogen Ausgangswertes (WR3A).	7-164
7.13	Anweisungen aus externem Speicher.	7-165
7.13.1	Anweisung aus externem ROM ausführen (EXTR)	7-165
7.14	Verschiedene Anweisungen	7-186
7.14.1	Operandenkommentar lesen (COMRD)	7-186
7.14.2	Zufallszahl generieren (RND)	7-189
7.14.3	Taktgenerator mit einstellbarem Tastverhältnis (DUTY)	7-190
7.14.4	Daten prüfen (CRC)	7-192
7.14.5	Istwert eines High-Speed-Counters transferieren (DHCMOV)	7-195
7.15	Anweisungen für Datenblöcke	7-199
7.15.1	Daten in zwei Datenblöcken addieren (BK+)	7-200
7.15.2	Daten in zwei Datenblöcke subtrahieren (BK-)	7-203
7.15.3	Daten in Datenblöcke vergleichen (BKCMP□)	7-206
7.16	Verarbeitungsanweisungen für Zeichenfolgen	7-211
7.16.1	Binärdaten in Zeichenfolgen wandeln (STR)	7-212
7.16.2	Zeichenfolgen in Binärdaten wandeln (VAL)	7-217
7.16.3	Zeichenfolgen zusammenfügen (\$+)	7-222
7.16.4	Länge von Zeichenfolgen ermitteln (LEN).	7-224
7.16.5	Auszug der Zeichenfolgedaten von rechts (RIGHT)	7-226
7.16.6	Auszug der Zeichenfolgedaten von links (LEFT)	7-229
7.16.7	Zeichen aus Zeichenfolge kopieren (MIDR)	7-232
7.16.8	Zeichenfolge ersetzen (MIDW)	7-235
7.16.9	Zeichenfolge suchen (INSTR)	7-238
7.16.10	Zeichenfolge transferieren (\$MOV)	7-240
7.17	Verarbeitungsanweisungen für Datenlisten.	7-242
7.17.1	Daten aus Datenliste löschen (FDEL).	7-242
7.17.2	Daten in Datenliste einfügen (FINS)	7-244
7.17.3	Daten lesen, die zuletzt in eine Datenliste eingetragen wurden (POP)	7-246
7.17.4	16-Bit-Datenwort nach rechts verschieben (SFR)	7-249
7.17.5	16-Bit-Datenwort nach links verschieben (SFL)	7-251

7.18	Vergleichsanweisungen (2)	7-253
7.18.1	Lade Vergleiche (LD□)	7-253
7.18.2	UND-verknüpfte Vergleiche (AND□).	7-255
7.18.3	ODER-verknüpfte Vergleiche (OR□)	7-257
7.19	Datenkontrollanweisungen	7-259
7.19.1	Ausgabebereich von Werten begrenzen (LIMIT)	7-260
7.19.2	Eingangs-Offset festlegen (BAND)	7-263
7.19.3	Ausgangs-Offset festlegen (ZONE).	7-266
7.19.4	Werte skalieren (SCL).	7-269
7.19.5	Zahl im ASCII-Code in Binärwert wandeln (DABIN).	7-273
7.19.6	Binärwert in ASCII-Code wandeln (BINDA)	7-276
7.19.7	Werte skalieren (SCL2).	7-279
7.20	Kommunikation mit Frequenzumrichtern.	7-283
7.20.1	Status eines Frequenzumrichters prüfen (IVCK)	7-285
7.20.2	Frequenzumrichter steuern (IVDR).	7-287
7.20.3	Parameter des Frequenzumrichters lesen (IVRD)	7-289
7.20.4	Parameter in Frequenzumrichter schreiben (IVWR).	7-290
7.20.5	Parameter blockweise in Frequenzumrichter schreiben (IVBWR)	7-292
7.20.6	Mehrfachanweisung für Frequenzumrichter (IVMC).	7-293
7.21	Kommunikation über MODBUS	7-295
7.21.1	MODBUS Lesen/Schreiben (ADPRW)	7-295
7.22	Datenaustausch mit Sondermodulen	7-299
7.22.1	Aus Pufferspeicher von Sondermodulen lesen (RBFM)	7-299
7.22.2	In Pufferspeicher von Sondermodulen schreiben (WBFM)	7-301
7.22.3	Hinweise zur Anwendung der RBFM- und WBFM-Anweisung.	7-302
7.22.4	Programmbeispiel zur RBFM- und WBFM-Anweisung.	7-304
7.23	Anweisung für High-Speed-Counter	7-305
7.24	Anweisungen für erweiterte File-Register	7-309
7.24.1	Daten aus erweiterten File-Registern lesen (LOADR)	7-309
7.24.2	Daten in erweiterte File-Register schreiben (SAVER)	7-311
7.24.3	Erweiterte Register und erweiterte File-Register initialisieren (INITR).	7-316
7.24.4	Operandenwerte in erweiterte Register/File-Register speichern (LOGR).	7-319
7.24.5	Daten aus erweiterte Register in erweiterte File-Register übertragen (RWER)	7-323
7.24.6	Erweiterte File-Register initialisieren (INITER)	7-326

7.25	Anweisungen für Adaptermodul FX3U-CF-ADP	7-329
7.25.1	Datei erzeugen/prüfen (FLCRT)	7-330
7.25.2	Datei löschen/CF-Speicherkarte formatieren (FLDEL)	7-333
7.25.3	Daten in CF-Speicherkarte schreiben (FLWR)	7-334
7.25.4	Daten aus CF-Speicherkarte lesen (FLRD)	7-337
7.25.5	Anweisung für FX3U-CF-ADP (FLCMD)	7-339
7.25.6	Status des FX3U-CF-ADP lesen (FLSTRD)	7-340

8 Sonderfunktionen

8.1	Datenerhalt im STOP-Modus	8-2
8.2	Betrieb mit konstanter Programmzykluszeit	8-3
8.3	Pulse-Catch-Funktion	8-4
8.4	Impulsweiten- und Periodendauermessung	8-7
8.5	Eingangsfiler einstellen	8-12
8.5.1	FX1S-, FX1N-, FX3G-, FX3GC-, FX3GE-, FX3S-, FX3U- und FX3UC-Serie	8-12
8.5.2	FX2N -, FX2NC-, FX3U- und FX3UC-Serie	8-12
8.6	Integrierte Potentiometer	8-13
8.7	Integrierte Uhr der SPS	8-14
8.8	RUN-/STOP-Umschaltung	8-16
8.9	FX2N-Grundgeräte mit 24 V DC-Versorgung	8-17
8.10	Anzeigemodul FX1N-5DM für FX1S und FX1N	8-18
8.10.1	Funktionen	8-18
8.10.2	Operanden zur Steuerung des Anzeigemoduls	8-19
8.10.3	Festlegung der angezeigten Operanden	8-20
8.10.4	Schutz vor unbefugten „Zugriffen“	8-21
8.10.5	Änderung des Zustandes oder der Werte von Operanden	8-21
8.10.6	Automatisches Ausschalten der Beleuchtung der Anzeige	8-22
8.10.7	Freigeben und Sperren der Fehleranzeige	8-22

9	Sondermerker, Sonderregister	
9.1	Sondermerker (M8000 bis M8511)	9-1
9.1.1	SPS-Status (M8000–M8009)	9-2
9.1.2	Zeittakte und Uhr (M8011–M8019)	9-3
9.1.3	Flags (M8020–M8029)	9-4
9.1.4	SPS-Modus (M8030–M8039)	9-5
9.1.5	STL-Status (M8040–M8049)	9-6
9.1.6	Interrupt-Programm (M8050–M8059)	9-7
9.1.7	Sondermerker für Fehlermeldungen (M8060–M8069)	9-8
9.1.8	Pulse-Catch-Funktion (M8170–M8177)	9-9
9.1.9	Link- und Sonderfunktionen (M8070–M8199)	9-10
9.1.10	Auf-/Abwärts-Counter (M8200–M8254)	9-15
9.1.11	Analoge Adaptermodule und Erweiterungsadapter (M8260–M8299)	9-18
9.1.12	Flags (M8300–M8311)	9-19
9.1.13	Sondermerker zur Fehlerdiagnose (M8312–M8329)	9-20
9.1.14	Sondermerker für Impulsausgabe und Positionierung (M8330–M8379)	9-21
9.1.15	Sondermerker für Kommunikation (M8370–M8379)	9-23
9.1.16	Sondermerker für High-Speed-Counter (M8380–M8392)	9-23
9.1.17	Sondermerker für Interrupt-Programme und High-Speed-Counter (M8393–M8397)	9-24
9.1.18	Sondermerker für Ringzähler (M8398)	9-24
9.1.19	Sondermerker für Kommunikation (M8400–M8459)	9-25
9.1.20	Sondermerker für Positionierung (M8460–M8467)	9-29
9.1.21	Sondermerker zur Fehlerdiagnose (2) (M8468–M8511)	9-30
9.2	Sonderregister (D8000 bis D8511)	9-31
9.2.1	SPS-Status (D8000–D8009)	9-31
9.2.2	Zykluszeitmessung und Uhrzeit/Datum (D8010–D8019)	9-32
9.2.3	EingangsfILTER und Indexregister (D8020–D8029)	9-33
9.2.4	Analoge Potentiometer und konstante Zykluszeit (D8030–D8039)	9-33
9.2.5	STL-Status (D8040–D8059)	9-34
9.2.6	Fehlerdiagnose (D8060–D8069)	9-34
9.2.7	Register für Link- und Sonderfunktionen (D8070–D8100)	9-35
9.2.8	Sonstige Register (D8101–D8119)	9-36
9.2.9	Register für Kommunikation (D8120–D8129)	9-37
9.2.10	Ausführungsregister für HSZ-, PLSY- und Positionieranweisungen (D8130–D8149)	9-38
9.2.11	Sonderregister für Kommunikation mit Frequenzumrichtern	9-39

9.2.12	Verschiedene Funktionen (D8158–D8169)	9-40
9.2.13	Sonderregister für ein n:n-Netzwerk (D8173–D8180, D8201–D8259) . .	9-41
9.2.14	Index-Register (D8182–D8199)	9-42
9.2.15	Sonderregister für analoge Adaptermodule und Erweiterungsadapter (D8260–D8299)	9-43
9.2.16	Sonderregister für Anzeigemodule (D8300–D8309)	9-44
9.2.17	Sonderregister für die RND-Anweisung (D8310 und D8311)	9-44
9.2.18	Sonderregister zur Fehlerdiagnose (D8312–D8328)	9-45
9.2.19	Sonderregister für Impulsausgabe und Positionierung	9-45
9.2.20	Sonderregister für Kommunikation (D8370–D8392)	9-47
9.2.21	Sonderregister für Interrupt-Programme (D8393–D8397)	9-48
9.2.22	Ringzähler (D8398 und D8399)	9-48
9.2.23	Sonderregister für Kommunikation (D8400–D8439)	9-49
9.2.24	Fehlererkennung bei Sondermodulen (D8440–D8459)	9-57
9.2.25	Sonderregister für Positionierung (D8460–D8467)	9-57
9.2.26	Sonderregister zur Fehlerdiagnose (2) (D8468–D8511)	9-58

10 Programmfehler

10.1	Fehlererkennung	10-1
10.1.1	Sondermerker (M8060–M8069)	10-1
10.1.2	Sonderregister (D8060–D8069, D8449)	10-2
10.2	Fehlercodes	10-3
10.2.1	Fehlercodes (3801–3840)	10-3
10.2.2	Fehlercodes (6101–6409)	10-4
10.2.3	Fehlercodes (6501–6511)	10-6
10.2.4	Fehlercodes (6601–6609)	10-7
10.2.5	Fehlercodes (6610–6632)	10-8
10.2.6	Fehlercodes (6701–6710)	10-9
10.2.7	Fehlercodes (6730–6773)	10-10
10.2.8	Fehlercodes (8702–8730)	10-12
10.2.9	Fehlercodes bei Fehlern in Sondermodulen	10-12
10.2.10	Fehlercodes bei Fehlern in Sondermodulparameter	10-13
10.2.11	Anzeige von Fehlern in Sondermodule (FX3U, FX3UC ab V 3.00)	10-13

A	Technische Daten	
A.1	Übersicht der Grundbefehle	A-1
A.2	Allgemeine Systemdaten MELSEC FX1S	A-4
A.3	Operanden MELSEC FX1S	A-4
A.4	Allgemeine Systemdaten MELSEC FX1N	A-6
A.5	Operanden MELSEC FX1N	A-6
A.6	Applikationsanweisungen MELSEC FX1S/FX1N	A-8
A.7	Allgemeine Systemdaten MELSEC FX2N/FX2NC	A-10
A.8	Operanden MELSEC FX2N/FX2NC	A-10
A.9	Applikationsanweisungen MELSEC FX2N/FX2NC	A-12
A.10	Allgemeine Systemdaten MELSEC FX3G/3GC/3GE	A-14
A.11	Operanden MELSEC FX3G/FX3GC/FX3GE	A-14
A.12	Applikationsanweisungen MELSEC FX3G/3GC/3GE	A-16
A.13	Allgemeine Systemdaten MELSEC FX3S	A-18
A.14	Operanden MELSEC FX3S	A-18
A.15	Applikationsanweisungen MELSEC FX3S	A-20
A.16	Allgemeine Systemdaten MELSEC FX3U/FX3UC	A-22
A.17	Operanden MELSEC FX3U/FX3UC	A-22
A.18	Applikationsanweisungen MELSEC FX3U/FX3UC	A-24
B	Ausführungszeiten der Anweisungen	
B.1	Ausführungszeiten der FX1S-/FX1N-Serie	B-1
B.1.1	Grundbefehle und Schrittstatus-Anweisungen	B-1
B.1.2	Programmverzweigungsanweisungen	B-3
B.1.3	Vergleichs- und Transferanweisungen	B-3
B.1.4	Arithmetische Anweisungen	B-4
B.1.5	Rotations- und Shift-Anweisungen	B-4
B.1.6	Datenoperationen	B-5
B.1.7	High-Speed-Anweisungen	B-5

B.1.8	Anwendungsbezogene Anweisungen	B-6
B.1.9	Spezielle Anweisungen	B-6
B.1.10	Positionier-Anweisungen	B-7
B.1.11	Anweisungen für die integrierte Uhr	B-7
B.1.12	Datenaustausch mit Analogeingangsmodule	B-7
B.1.13	Vergleichsanweisungen	B-8
B.2	Ausführungszeiten der FX2N-/FX2NC-Serie	B-9
B.2.1	Grundbefehle und Schrittstatus-Anweisungen	B-9
B.2.2	Programmverzweigungsanweisungen	B-11
B.2.3	Vergleichs- und Transferanweisungen	B-11
B.2.4	Arithmetische Anweisungen	B-12
B.2.5	Rotations- und Shift-Anweisungen	B-13
B.2.6	Datenoperationen	B-14
B.2.7	High-Speed-Anweisungen	B-14
B.2.8	Anwendungsbezogene Anweisungen	B-15
B.2.9	Spezielle Anweisungen	B-16
B.3	Ausführungszeiten der FX3G-/FX3GC-/FX3GE-Serie	B-19
B.3.1	Grundbefehle und Schrittstatus-Anweisungen	B-19
B.3.2	Programmverzweigungsanweisungen	B-21
B.3.3	Vergleichs- und Transferanweisungen	B-22
B.3.4	Arithmetische Anweisungen	B-23
B.3.5	Rotations- und Shift-Anweisungen	B-24
B.3.6	Datenoperationen	B-24
B.3.7	High-Speed-Anweisungen	B-25
B.3.8	Anwendungsbezogene Anweisungen	B-25
B.3.9	Spezielle Anweisungen	B-26
B.4	Ausführungszeiten der FX3S-Serie	B-31
B.4.1	Grundbefehle und Schrittstatus-Anweisungen	B-31
B.4.2	Programmverzweigungsanweisungen	B-33
B.4.3	Vergleichs- und Transferanweisungen	B-34
B.4.4	Arithmetische Anweisungen	B-34
B.4.5	Rotations- und Shift-Anweisungen	B-35
B.4.6	Datenoperationen	B-35
B.4.7	High-Speed-Anweisungen	B-36
B.4.8	Anwendungsbezogene Anweisungen	B-36
B.4.9	Spezielle Anweisungen	B-37
B.5	Ausführungszeiten der FX3U-/FX3UC-Serie	B-40

B.5.1	Grundbefehle und Schrittstatus-Anweisungen	B-40
B.5.2	Programmverzweigungsanweisungen	B-43
B.5.3	Vergleichs- und Transferanweisungen	B-43
B.5.4	Arithmetische Anweisungen	B-44
B.5.5	Rotations- und Shift-Anweisungen	B-45
B.5.6	Datenoperationen	B-45
B.5.7	High-Speed-Anweisungen	B-46
B.5.8	Anwendungsbezogene Anweisungen.	B-46
B.5.9	Spezielle Anweisungen.	B-47
B.6	Verarbeitungszeiten bei Puls-Ausführung	B-53
B.6.1	Verarbeitungszeiten für Grundgeräte der FX3G/FX3GC/FX3GE-Serie . . .	B-53
B.6.2	Verarbeitungszeiten für Grundgeräte der FX3S-Serie	B-54
B.6.3	Verarbeitungszeiten für Grundgeräte der FX3U/FX3UC-Serie.	B-54
B.7	Abhängigkeit von Operanden und Index-Vergabe	B-55
B.7.1	Ausführungszeiten der Grundbefehle	B-55
B.7.2	Ausführungszeiten der Applikationsanweisungen	B-55
B.8	Ausführungszeiten für Pointer (P, I).	B-61
B.8.1	Grundgeräte der FX3G-, FX3GC- und FX3GE-Serie.	B-61
B.8.2	Grundgeräte der FX3S-Serie.	B-61
B.8.3	Grundgeräte der FX3U- und FX3UC-Serie.	B-61

C ASCII-Code

1 Einleitung

1.1 Allgemeines

Anwendungsbereich

Das vorliegende Handbuch beschreibt die elementaren Grundlagen zur Programmierung der speicherprogrammierbaren Steuerungen der MELSEC FX1S-, FX1N-, FX2N-, FX2NC-, FX3G-, FX3GC-, FX3GE-, FX3S-, FX3U- und FX3UC-Serie.

Informationen zur Installation, Inbetriebnahme, Wartung und Fehlerbehebung entnehmen Sie bitte den entsprechenden Hardware-Handbüchern der Module.

HINWEISE

Wenn nicht besonders gekennzeichnet, betreffen allgemeingültige Aussagen immer alle Mitglieder der FX-Familie. Die im Verlauf des Handbuches verwendete Bezeichnung „FX-Familie“ bezieht sich auf die folgenden Steuerungstypen:

- FX1S
- FX1N
- FX2N
- FX2NC
- FX3G
- FX3GC
- FX3GE
- FX3S
- FX3U
- FX3UC

Angaben oder Besonderheiten, die immer jeweils nur einen bestimmten Steuerungstyp betreffen, sind entsprechend gekennzeichnet.

1.2 Verwendbare Programmiergeräte/Bediengeräte

In der folgenden Tabelle sind die Programmiergeräte und Bediengeräte aufgeführt, die direkt mit den Anweisungen der FX-Familie arbeiten können:

Beschreibung	Modell	Neue Version, völlig kompatibel
Handprogrammiergeräte	FX-10P-E	FX2N ab V 3.00 FX1N/FX1S ab V 4.00
	FX-20P-MFXA-E	FX2N ab V 3.00 FX1N/FX1S ab V 5.00
	FX-30P	
Bediengeräte	FX-10DU-E	V 4.00
	FX-20DU-E	Unterstützt nur die Operanden der FX-Serie
	FX-30DU-E	V 3.00
	FX-40DU-E(S)	Unterstützt nur die Operanden der FX- Serie
	FX-40DU-TK-ES	V 3.00
	FX-50DU-TK(S)-E	V 2.10

Außerdem können die Bediengeräte der F-GOT-, A-GOT- und MAC E-Serie verwendet werden.

Mit einem Personal Computer, auf dem die Programmier-Software GX Developer FX, GX Developer, GX IEC Developer oder GX Works2 installiert ist, kann die MELSEC FX-Familie besonders komfortabel programmiert werden.

2 Grundlagen der Programmierung

2.1 Programmverarbeitung in der SPS

Funktionsprinzip

Über die Eingänge einer speicherprogrammierbaren Steuerung (SPS) werden analoge oder binäre Signale erfasst und im SPS-Programm verarbeitet. Die Verknüpfungsergebnisse des Programms werden entweder intern gespeichert oder schalten die Ausgänge der SPS.

Arbeitsweise

Eine SPS arbeitet nach einem vorgegebenen Programm, das in der Regel außerhalb der Steuerung erstellt, in die Steuerung übertragen und im Programmspeicher abgelegt wird.

Das Programm besteht aus einer Folge einzelner Anweisungen, die die Funktion der Steuerung festlegen. Die SPS arbeitet die Steuerungsanweisungen in der programmierten Reihenfolge nacheinander (sequentiell) ab.

Der gesamte Programmdurchlauf wird ständig wiederholt, es findet also ein zyklischer Programmdurchlauf statt.

Zur Ausführung der einzelnen Anweisungen benötigt eine SPS Zeit (siehe Anhang B). Die Summe der Ausführungszeiten der Anweisungen ergibt zusammen mit der Zeit, die für systeminterne Vorgänge gebraucht wird, die Zykluszeit eines SPS-Programms. Das ist die Zeit, die für einen Programmdurchlauf benötigt wird. Die Zykluszeit ist abhängig vom Typ der SPS sowie von der Anzahl und der Art der Steuerungsanweisungen.

2.1.1 Prozessabbildverfahren

Bei der Programmbearbeitung in der SPS wird nicht direkt auf die Ein- und Ausgänge, sondern auf ihr Prozessabbild zugegriffen:

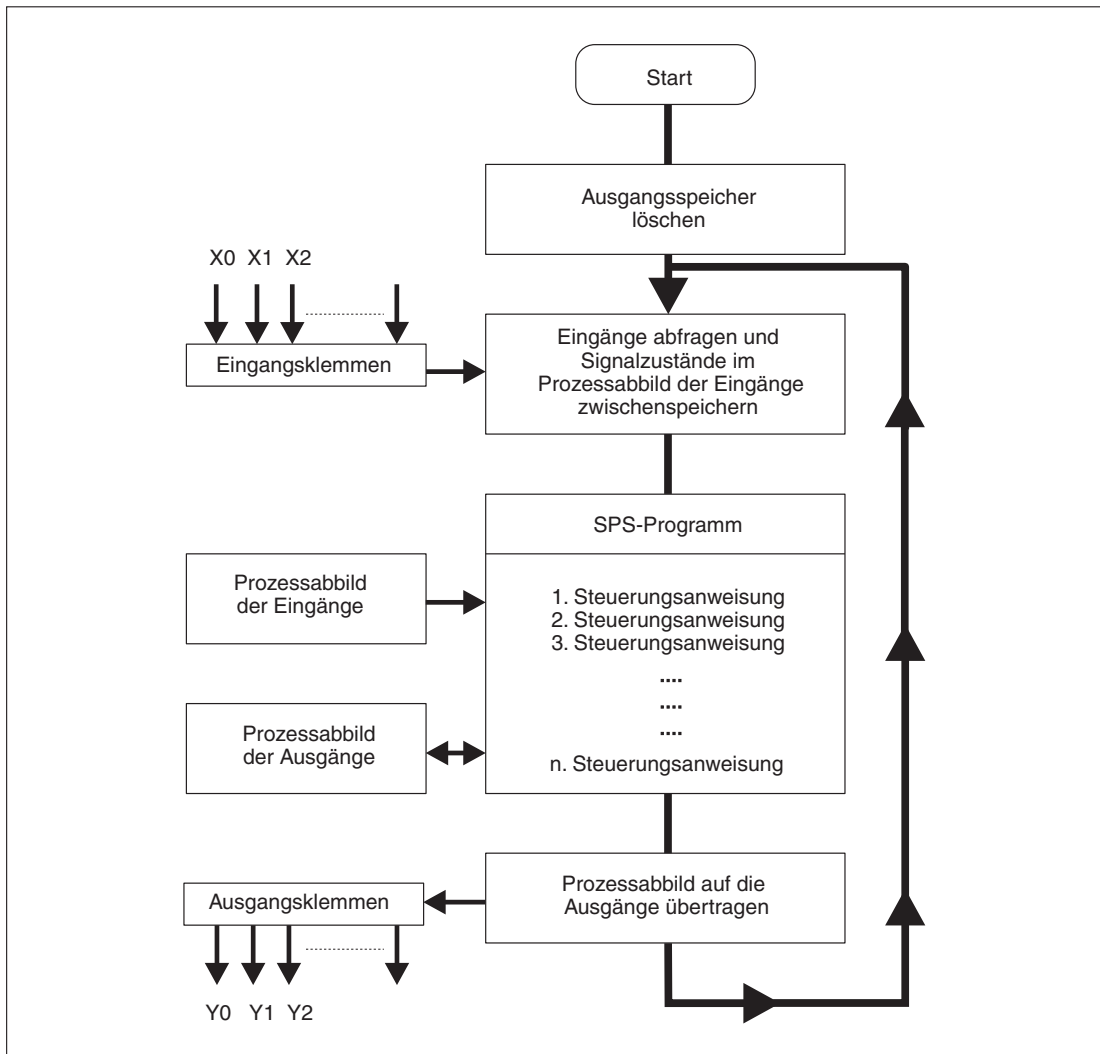


Abb. 2-1: Abarbeitung eines Programms nach dem Prozessabbildverfahren

Prozessabbild der Eingänge

Am Anfang eines Programmzyklus werden die Signalzustände der Eingänge abgefragt und zwischengespeichert: Es wird ein sogenanntes Prozessabbild der Eingänge erstellt.

Programmdurchlauf

Während des anschließenden Programmdurchlaufs greift die SPS auf die gespeicherten Eingangszustände im Prozessabbild zu. Signaländerungen an den Eingängen werden daher erst im nächsten Programmzyklus erkannt.

Prozessabbild der Ausgänge

Verknüpfungsergebnisse, die die Ausgänge betreffen, werden in einem Ausgangszwischenspeicher hinterlegt (Prozessabbild der Ausgänge). Erst am Ende des Programmdurchlaufs werden die Zwischenergebnisse an die Ausgänge übertragen. Im Ausgangszwischenspeicher bleibt das Prozessabbild der Ausgänge bis zum nächsten Überschreiben erhalten. Nach der Wertzuweisung an die Ausgänge wird der Programmzyklus wiederholt.

2.1.2 **Signalverarbeitung in der SPS im Unterschied zur verbindungsprogrammierten Steuerung**

Bei einer verbindungsprogrammierten Steuerung ist das Programm durch die Art der Funktionsglieder und deren Verbindung (Verdrahtung) vorgegeben. Alle Steuerungsvorgänge werden gleichzeitig (parallel) ausgeführt. Jede Änderung der Eingangssignalzustände bewirkt sofort eine Änderung der Ausgangssignalzustände.

HINWEIS

Bei einer SPS kann eine Änderung der Eingangssignalzustände während des Programmdurchlaufs erst wieder beim nächsten Programmzyklus berücksichtigt werden. Dieser Nachteil wird durch entsprechend kurze Zykluszeiten weitgehend wieder ausgeglichen.

2.2 Steuerungsanweisungen

Das SPS-Programm besteht aus einer Folge von Verknüpfungen, die die Funktion der Steuerung festlegen. Zur Programmierung ist es daher notwendig, die Steuerungsaufgabe in einzelne Steuerungsanweisungen zu zerlegen. Eine Steuerungsanweisung ist die kleinste Einheit eines Programms.

2.2.1 Aufbau einer Steuerungsanweisung

Eine Steuerungsanweisung besteht aus einer Schrittnummer, einer Anweisung (Befehl) und einem Operanden.

Steuerungsanweisung			
Schritt- nummer	Anweisung (Befehl)	Operand	
		Operanden- kennzeichen	Operanden- adresse
„015“	„AND“	„Y“	„003“

Tab. 2-1:
Aufbau einer Steuerungsanweisung

- Die Anweisungen werden in einer bestimmten Reihenfolge abgearbeitet, die durch die Angabe der Schrittnummer festgelegt wird.
- Die Anweisung (Befehl) beschreibt die auszuführende Funktion, also die Art der Verknüpfung.
- Der Operand gibt an, womit eine Verknüpfung (Anweisung) ausgeführt werden soll. Ein Operand kann zum Beispiel ein Eingangs-, ein Ausgangs- oder ein interner Zähler sein.

HINWEIS

Bei bestimmten Steuerungsanweisungen (Befehlen) kann die Angabe des Operanden und/oder der Operandenadresse entfallen.

2.2.2 Operanden

Der Operand besteht aus einem

- Operandenkennzeichen und
- einer Operandenadresse.

Das Operandenkennzeichen definiert die Art des Operanden z. B. Eingang oder Ausgang.

Die Angabe der Operandenadresse ermöglicht

- eine Unterscheidung bei einer mehrfachen Benutzung des gleichen Operandenkennzeichens oder
- die Festlegung von Zahlenwerten z. B. für Konstanten.

Die folgende Tabelle enthält eine Übersicht aller programmierbaren Operanden und deren entsprechenden Operandenkennzeichen:

Operand	Operandenkennzeichen	Bedeutung
Eingang	X	Eingangsklemme der SPS
Ausgang	Y	Ausgangsklemme der SPS
Merker	M	Hilfsrelais; Speicher für binäre Zwischenergebnisse
Timer	T	Zeitglied; Speicher zur Realisierung von Zeiten
Counter	C	Zähler; Speicher zur Realisierung von Zählern
Schrittstatus	S	Festgelegter Schritt; Programmierung von Ablaufsteuerungen
Dezimalkonstante	K	Festgelegter dezimaler Zahlenwert
Hexadezimalkonstante	H	Festgelegter hexadezimaler Zahlenwert
Datenregister	D	Datenspeicher (16-Bit- oder 32-Bit-Format)
Index-Register	V, Z	Datenspeicher für Zwischenergebnisse, Indizierung (16-Bit-Format)
Pointer	P	Sprungzieladresse (Markierung für einen Programmsprung)
Interrupt-Pointer	I	Programmunterbrechung (Sprung zum Interrupt-Programm)
Nesting	N	Programmverzweigung

Tab. 2-2: Operanden und Operandenkennzeichen

2.2.3 Darstellungsarten von Steuerungsanweisungen

Eine SPS-Programmierung kann in drei verschiedenen Darstellungsarten erfolgen:

- Anweisungsliste (AWL)
- Funktionsplan (FUP)
- Kontaktplan (KOP)

HINWEISE

Abhängig vom genutzten Programmiersystem können nicht alle drei Darstellungsarten genutzt werden.

Eine Programmierung mit den in den IEC 1131.3 definierten Darstellungsarten ist ebenfalls möglich.

Anweisungsliste

Die Anweisungsliste stellt das Programm als eine Abfolge von Steuerungsanweisungen in einer Liste dar.

Schritt- nummer	Anweisung	Operanden- kennzeichen	Operanden- adresse
000	LD	X	000
001	ORI	X	001
002	OUT	Y	000
003	END	—	—

Tab. 2-3: Beispiel einer Anweisungsliste

Funktionsplan

Der Funktionsplan stellt das Programm als eine Abfolge von Netzwerken dar, wobei die Steueranweisungen innerhalb der Netzwerke als Funktionsblöcke erscheinen.

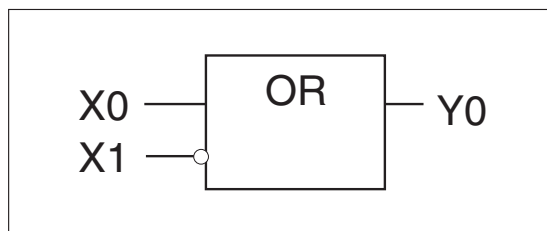




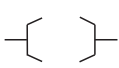

Abb. 2-2:

Beispiel eines Funktionsplans

Kontaktplan

Der Kontaktplan ist eine Anlehnung an den Stromlaufplan in aufgelöster Darstellung. Entgegen der dort üblichen senkrechten Anordnung der Strompfade werden im Kontaktplan die Strompfade waagrecht dargestellt und untereinander angeordnet.

Im Wesentlichen werden die folgenden vier Grundsymbole verwendet.

Symbol	Bedeutung
	Symbol für einen Signaleingang mit Abfrage auf Signalzustand „1“
	Symbol für einen Signaleingang mit Abfrage auf Signalzustand „0“
	Symbol für einen Signalausgang. Bei Ansteuern mit einem „1“-Signal wird dem angegebenen Operanden ein „1“-Signal zugewiesen
	Symbol für Sonderfunktionen

Tab. 2-4:
Kontaktplansymbolik

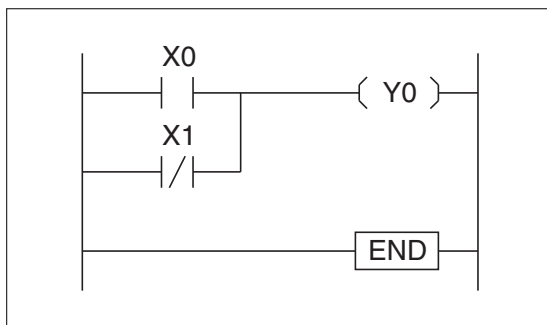


Abb. 2-3:
Beispiel eines Kontaktplans

2.2.4 Zuordnungsliste und Beschaltung der SPS

Neben der Beschreibung des SPS-Programms ist für die Praxis eine Zuordnungsliste und die Beschaltung der SPS wichtig.

Zuordnungsliste

Aus der Zuordnungsliste wird ersichtlich,

- mit welchen Geräten die Ein- und Ausgänge beschaltet sind.
- welche in der SPS vorhandenen Funktionen (Zähler, Merker, usw.) für den Steuerungsprozess eingesetzt werden.

Benennung	Stromlaufplan- kennzeichen	Operanden- kennzeichen	Operanden- adresse
Schließer „Ein“	S1	X	000
Öffner „Aus“	S2	X	001
Melder	H1	Y	000
Melder	H2	Y	001
Timer (100 ms)	—	T	003

Tab. 2-5: Beispiel einer Zuordnungsliste

Beschaltung der SPS

Die Beschaltung der SPS stellt die Verbindungen zwischen der SPS und den angeschlossenen Ein- und Ausgabegeräten dar.

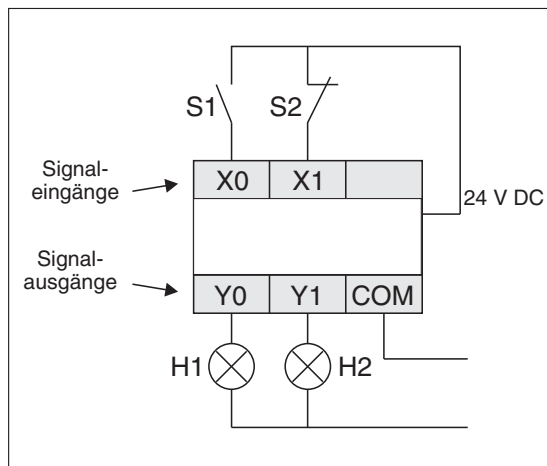


Abb. 2-4:
Beispiel einer SPS-Beschaltung

3 Operanden

3.1 Übersicht der Operanden

Dieses Kapitel beschreibt alle verfügbaren Operanden und deren Einsatzmöglichkeiten innerhalb eines SPS-Programms. Mit der Angabe eines Operanden wird festgelegt, womit eine Operation (Anweisung) ausgeführt wird.

Ein Operand besteht aus einem

- Operandenkennzeichen und
- einer Operandenadresse.

MELSEC-Operanden		Max. Anzahl der Operanden								
		FX1S	FX1N	FX2N	FX2NC	FX3G	FX3GC	FX3GE	FX3S	FX3U
Eingänge	X	Max. 16*	Summe bis 128	Summe bis 256	Summe bis 256	Summe bis 256	Summe bis 256	Max. 16*	Summe bis 256	Summe bis 256
Ausgänge	Y	Max. 14*						Max. 14*		
Merker	M	384	384	3072		8192		2048		8192
Timer	T	64	256	256		320		138		512
Counter	C	53	256	256		256		256		235
High-Speed-Counter (gleichzeitig nutzbar)	C	21	21	21		21		21		21
Schrittstatus	S	128	1000	1000		4096		256		4096
Dezimale Konstante	K	16/32 Bit	16/32 Bit	16/32 Bit		16/32 Bit		16/32 Bit		16/32 Bit
Hexadezimale Konstante	H	16/32 Bit	16/32 Bit	16/32 Bit		16/32 Bit		16/32 Bit		16/32 Bit
Gleitkommakonstante	E	—	—	—		32 Bit		32 Bit		32 Bit
Zeichenketten	" "	—	—	—		—		—		max. 32 Zeichen
Datenregister	D	256	8000	8000		8000		3000		8000
File-Register	D	—	7000 (anteilig)	7000 (anteilig)		7000 (anteilig)		2000 (anteilig)		7000 (anteilig)
Erweiterte Register	R	—	—	—		24000		—		32768
Erweiterte File-Register	ER	—	—	—		24000		—		32768
Index-Register	V, Z	16	16	16		16		16		16
Pointer	P	64	128	128		2048		256		4096
Interrupt-Pointer	I	6	6	6 Eingänge 3 Timer		6 Eingänge 3 Timer		6 Eingänge 3 Timer		6 Eingänge 3 Timer 3 Counter
Nesting	N	8	8	8		8		8		8

Tab. 3-1: Operanden und Operandenkennzeichen

* abhängig von der Art der Steuerung

Eine detaillierte Übersicht der Operanden und Operandenadressen für jeden Steuerungstyp befindet sich im Anhang dieses Handbuchs.

3.2 Ein- und Ausgänge

Die Ein- und Ausgänge werden im SPS-Programm durch Operanden dargestellt. Durch die Angabe einer zusätzlichen Operandenadresse können Sie gezielt die einzelnen Ein- und Ausgänge beim Programmieren ansprechen.

3.2.1 Ein- und Ausgänge adressieren

Die Ein- und Ausgänge werden oktal adressiert, d. h. nach 8 Ziffern findet ein Stellensprung statt (0,1, 2, 3, 4, 5, 6, 7,10,11,..., 16, 17). Die Ziffern 8 und 9 existieren im Oktalsystem nicht.

Steuerung	Art der Operanden	Operandenkennzeichen, Operandenadressen	Anzahl der Adressen
FX1S	Eingänge	X0 bis X17	6–16
	Ausgänge	Y0 bis Y15	4–14
FX1N	Eingänge	X0 bis X157	max. 128 ^①
	Ausgänge	Y0 bis Y157	max. 128 ^①
FX2N FX2NC	Eingänge	X0 bis X317	max. 256 ^②
	Ausgänge	Y0 bis Y317	max. 256 ^②
FX3G FX3GC FX3GE	Eingänge	X0 bis X177	max. 128 ^①
	Ausgänge	Y0 bis Y177	max. 128 ^①
FX3S	Eingänge	X0 bis X17	6–16
	Ausgänge	Y0 bis Y15	4–14
FX3U FX3UC	Eingänge	X0 bis X367	max. 256 ^②
	Ausgänge	Y0 bis Y367	max. 256 ^②

Tab. 3-2: Anzahl der maximal adressierbaren Ein- und Ausgänge

- ① Die Summe der Ein-/Ausgänge (Hardware) ist 128. Durch die Software können 128 Eingänge und 128 Ausgänge adressiert werden.
- ② Die Summe der Ein-/Ausgänge (Hardware) ist 256. Durch die Software können 256 Eingänge und 256 Ausgänge adressiert werden.

Verarbeitung von Eingangssignalen mit kurzen Impulszeiten

Um Eingangssignale korrekt erfassen zu können, müssen sie länger als die Programmzykluszeit sein.

Bei einer Programmzykluszeit von z. B. 10 ms und einer Schaltverzögerung von 10 ms darf sich der Zustand der Eingänge nur alle 20 ms ändern.

Bei diesem Beispiel können Eingangssignale, deren Schaltfrequenz größer als 25 Hz ist (20 ms EIN und 20 ms AUS), nicht direkt verarbeitet werden. Eine Programmverarbeitung dieser Signale ist jedoch mit Hilfe von Applikationsanweisungen möglich.

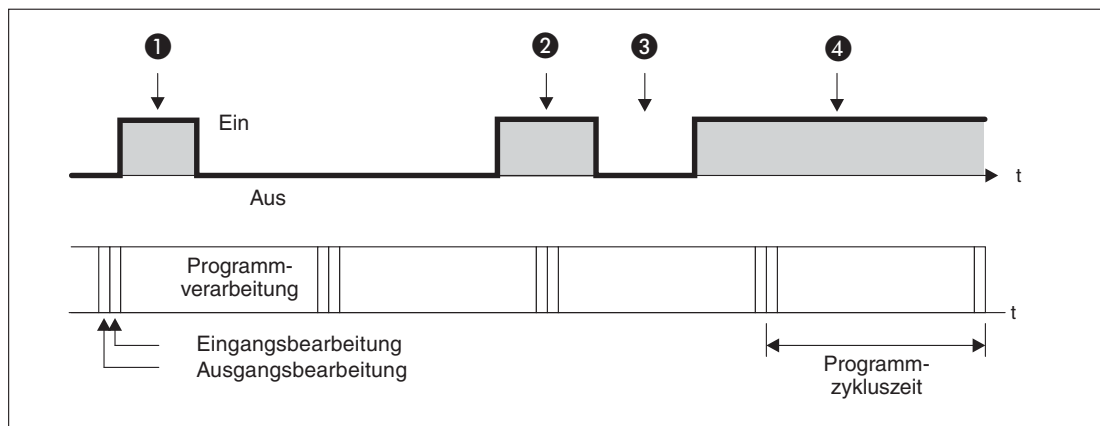


Abb. 3-1: Verarbeitung von Eingangssignalen mit kurzen Impulszeiten

- ① und ③: Dieser Eingangsstatus wird nicht erkannt;
- ②: Dieser Eingangsstatus wird zufällig erkannt;
- ④: Dieser Eingangsstatus wird immer korrekt erkannt.

3.2.2 Ein- und Ausgänge programmieren

Die Signalzustände der Ein- und Ausgänge können mit verschiedenen Anweisungen im Programm abgefragt werden.

Über die Ausgänge können Verknüpfungsergebnisse ausgegeben werden. Zusätzlich lassen sich die Signalzustände der Ausgänge im Programm direkt festlegen (Setzen oder Zurücksetzen).

Beispiel ▾ Einsatz der Ein- und Ausgänge

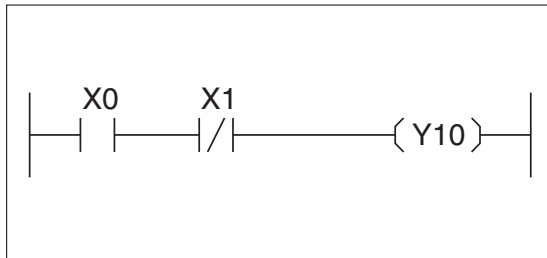


Abb. 3-2:
Programmierbeispiel zum Einsatz von Ein- und Ausgängen

Der Ausgang Y10 weist den Signalzustand „1“ auf, wenn folgende Bedingungen erfüllt sind:

- Der Eingang X0 weist den Signalzustand „1“ auf,
und
- der Eingang X1 den Signalzustand „0“ aufweist.

△

HINWEIS

Das Relais oder der Transistor des Ausgangs Y10 wird nach der Abarbeitung des SPS-Zyklus eingeschaltet.

3.3 Merker

Zum Speichern von binären Verknüpfungsergebnissen (Signalzustand „0“ oder „1“) innerhalb eines Programms werden Zwischenspeicher (Merker) eingesetzt. Diese Merker entsprechen in der Verwendung den Hilfsrelais in den Relaissteuerungen.

Die FX-Familie stellt neben den „normalen“ Merkern auch sogenannte Latch- und Sondermerker zur Verfügung.

- Latch-Merker behalten auch bei einem Spannungsausfall ihre Informationen. Die Informationen werden in einem spannungsausfallsicheren Speicher zwischengespeichert.
- Sondermerker stellen spezielle Sonderfunktionen zur Verfügung (siehe Abschnitt 10.1).

3.3.1 Merker adressieren

Merker und Latch-Merker werden dezimal adressiert.

Steuerung	Operand	Speicherinhalt im EEPROM gesichert	Operandenkennzeichen Operandenadressen	Anzahl der Adressen
FX1S	Merker	—	M0–M383	384
	Latch-Merker	●	M384–M511	128
	Sondermerker	●	M8000–M8255	256
FX1N	Merker	—	M0–M383	384
	Latch-Merker	●	M384–M1535	1152
	Sondermerker	●	M8000–M8255	256
FX2N FX2NC	Merker ^①	—	M0–M3071	3072
	Latch-Merker	—	M500–M3071	2572 (anteilig)
	Sondermerker	—	M8000–M8255	256
FX3G FX3GC FX3GE	Merker	—	M0–M383 M1536–M7679 ^②	384 6144
	Latch-Merker	●	M384–M1535	1152
	Sondermerker	●	M8000–M8511	512
FX3S	Merker	—	M0–M383, M512–M1535	1408
	Latch-Merker	●	M384–M511	128
	Sondermerker	●	M8000–M8511	512
FX3U FX3UC	Merker	—	M0–M499 ^③	500
	Latch-Merker	●	M500–M1023 ^④ M1024–M7679	524 6656
	Sondermerker	●	M8000–M8511	512

Tab. 3-3: Merker und die zugehörigen Operandenadressen

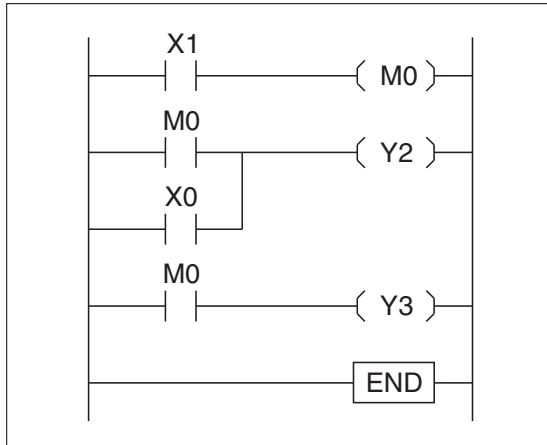
- ① Die Merker M2800 bis M3071 können in Verbindung mit gepulsten Anweisungen (LDP, LDF etc.) als flankengesteuerte Merker verwendet werden.
- ② Wenn die optionale Batterie installiert ist, kann diesen Merkern in den SPS-Parametern die Funktion von Latch-Merkern zugewiesen werden. Sie werden dann durch die Batterie gepuffert.
- ③ Diesen Merkern kann in den SPS-Parametern auch die Funktion von Latch-Merkern zugewiesen werden.
- ④ Diesen Merkern kann in den SPS-Parametern auch die Funktion von ungepufferten Merkern zugewiesen werden

3.3.2 Merker programmieren

Merker werden wie Ausgänge programmiert. Es besteht aber keine Möglichkeit, außerhalb der SPS an diese Merker Geräte anzuschließen, weil Merker nur Speicherstellen im Arbeitsspeicher der SPS darstellen.

Beispiel ▾

Einsatz der Merker

**Abb. 3-3:***Beispiel zum Einsatz von Merkern*

Wenn der Eingang X1 den Signalzustand „1“ aufweist, hat auch der Merker M0 den Signalzustand „1“. Der Merker M0 schaltet dann die Ausgänge Y2 und Y3 auf den Signalzustand „1“.

Wenn der Eingang X0 den Signalzustand „1“ aufweist, wird der Ausgang Y2 unabhängig von M0 auf den Signalzustand „1“ geschaltet. △

3.4 Timer

Für einige Steuerungsprozesse, wie z. B. einem zeitabhängigen Schalten eines Lüftungsmotors, werden Zeitglieder benötigt. In der Relais-technik werden hierfür Zeitrelais mit Einschalt- oder Ausschaltverzögerung eingesetzt. Die SPS-Technik verwendet interne Zeitglieder, deren Verhalten durch das Programm bestimmt werden kann.

Zum Starten und Rücksetzen werden Timer wie Ausgänge programmiert. Der Schaltkontakt eines Timers kann innerhalb des Programms beliebig oft abgefragt werden.

3.4.1 Adressierung der Timer

Die Timer werden dezimal adressiert.

Steuerung	Operandenkennzeichen	Anzahl der Adressen	Zeitschritte	Zeitbereich	Sondermerker M8028
FX1S	T0–T31	32	100 ms	0,1–3276,7 s	—
	T32–T62	31	100 ms		Aus
			10 ms	0,01–327,67 s	Ein
	T63	1	1 ms	0,001–32,767 s	—
FX1N	T0–T199	200	100 ms	0,1–3276,7 s	—
	T200–T245	46	10 ms	0,01–327,67 s	—
	T246–T249	4	1 ms	0,001–32,767 s	—
	T250–T255 (remanent)	6	100 ms	0,1–3276,7 s	—
FX2N FX2NC	T0–T199	200	100 ms	0,1–3276,7 s	—
	T200–T245	46	10 ms	0,01–327,67 s	—
	T246–T249	4	1 ms	0,001–32,767 s	—
	T250–255 (remanent)	6	100 ms	0,1–3276,7 s	—
FX3G FX3GC FX3GE	T0–T199	200	100 ms	0,1–3276,7 s	—
	T200–T245	46	10 ms	0,01–327,67 s	—
	T256–T319	64	1 ms	0,001–32,767 s	—
	T246–249 (remanent)	4	1 ms	0,001–32,767 s	—
	T250–255 (remanent)	6	100 ms	0,1–3276,7 s	—
FX3S	T0–T31	32	100 ms	0,1–3276,7 s	—
	T32–T62	31	100 ms	0,1–3276,7 s	Aus
			10 ms	0,01–327,67 s	Ein
	T63–T127	65	1 ms	0,001–32,767 s	—
	T128–T131 (remanent)	4	1 ms	0,001–32,767 s	—
	T132–T137 (remanent)	6	100 ms	0,1–3276,7 s	—
FX3U FX3UC	T0–T199	200	100 ms	0,1–3276,7 s	—
	T200–T245	46	10 ms	0,01–327,67 s	—
	T246–T249 (remanent)	3	1 ms	0,001–32,767 s	—
	T250–T255 (remanent)	6	100 ms	0,1–3276,7 s	—
	T256–T511	256	1 ms	0,001–32,767 s	—

Tab. 3-4: Einstellbarer Zeitbereich der Timer und deren Operandenadressen

HINWEIS

Wenn bei einer Steuerung der FX1S-Serie mit den Versionsnummern V1.00 bis V1.30 der Timer T63 (Zeitschritt 1 ms) im Programm verwendet wird, muss das Programm um die folgende Anweisung erweitert werden. Die Anweisung braucht nur einmal vor dem Zugriff auf T63 eingefügt werden. Bei Steuerungen ab Version 1.40 ist diese Anweisung nicht notwendig.

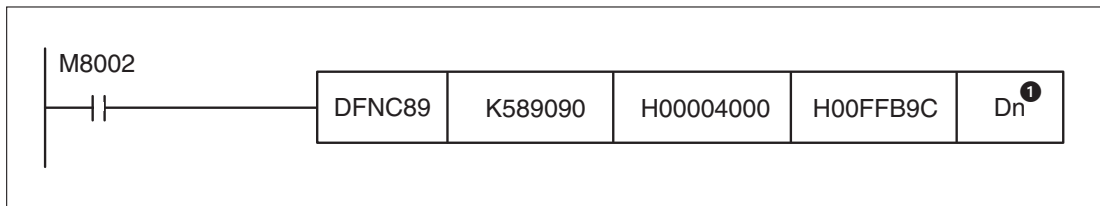


Abb. 3-4: Programmerweiterung bei Verwendung von T63 bei einer FX1S der Versionen V1.00 bis V1.30

- ① Die Nummer des Registers wird vom Anwender festgelegt. Durch die Anweisung werden zwei aufeinanderfolgende Register belegt.

3.4.2 Programmierung der Timer

Der gewählte Zeitsollwert wird durch eine zusätzliche Dezimalkonstante K festgelegt, die die Anzahl der Zeitschritte angibt.

Beispiel ▾ Bei einem 100-ms-Timer, bei dem die Dezimalkonstante mit $K = 5$ definiert ist, entspricht dies einem Zeitwert von $5 \times 100 \text{ ms} = 500 \text{ ms}$.

△

Die Timer arbeiten als Einschaltverzögerung. Ein Timer wird durch Ansteuern mit einem „1“-Signal aktiviert. Nach Ablauf des eingestellten Zeitsollwertes schaltet der Timer auf den Signalzustand „1“. Ein Timer fällt in den Ruhezustand zurück, sobald kein „1“-Signal an seinem Eingang mehr ansteht.

HINWEIS

Die Angabe des Zeitsollwertes kann auch indirekt über den in einem Datenregister gespeicherten dezimalen Zahlenwert vorgenommen werden.

Beispiel

Das folgende Kontaktplanbeispiel zeigt den Einsatz der Timer und die indirekte Festlegung des Zeitsollwertes.

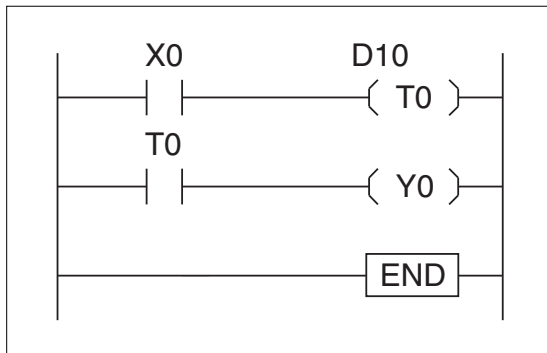


Abb. 3-6:

Programmierbeispiel zum Einsatz der Timer

Der Zeitsollwert ist im Datenregister D10 abgespeichert.

△

Beispiel

Das folgende Kontaktplanbeispiel zeigt den Einsatz der Timer und die direkte Festlegung des Zeitsollwertes.

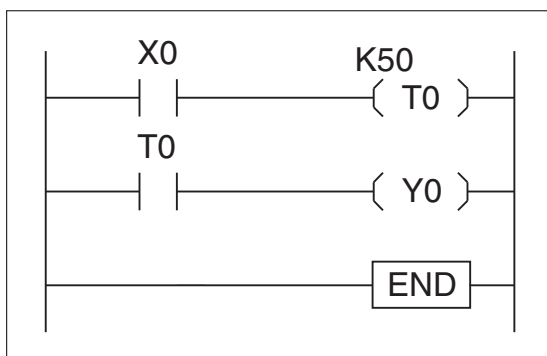


Abb. 3-5:

Vorgabe des Zeitsollwertes durch eine Konstante

Wenn der Eingang X0 den Signalzustand „1“ aufweist, beginnt die eingestellte Zeit abzulaufen. Nach Ablauf der programmierten Zeit $t = 5 \text{ s}$ wird der Ausgang Y0 auf den Signalzustand „1“ geschaltet. Der Timer T0 fällt in den Ruhezustand zurück, sobald der Eingang X0 den Signalzustand „0“ aufweist.

△

3.4.3 Vorgabe eines Zeitwertes mit den integrierten Potentiometern der Steuerungen der FX1S-, FX1N-, FX3G-, FX3GE- und FX3S-Serie

Über zwei Drehpotentiometer kann der Datenwert in zwei Sonderregistern manuell im Bereich von 0 bis 255 variiert werden. Der Wert des oberen Potentiometers VR1 kann aus dem Sonderregister D8030 gelesen werden, das untere Potentiometer VR2 beeinflusst den Inhalt des Sonderregisters D8031. Der Inhalt des Sonderregisters kann dann als Sollwertvorgabe für Timer, aber auch für Counter (Zähler) im Programm verwendet werden.

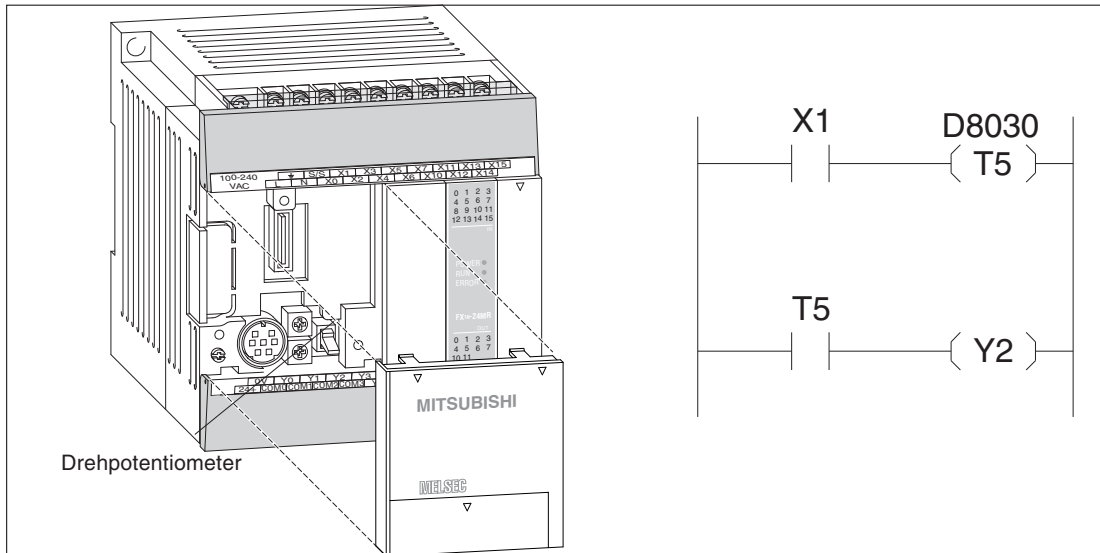


Abb. 3-7: Beispiel zur Einstellung eines Zeitsollwertes bei einem FX1N-Grundgerät im Sonderregister D8030 mit einem Potentiometer

Bei den Steuerungen der FX2N-, FX3G-, FX3GE- und FX3S-Serie ist diese Art der Sollwertvorgabe optional möglich, indem ein Sollwertvorgabe-Adapter FX2N-8AV-BD bzw. FX3G-8AV-BD mit acht Potentiometern installiert wird.

Steuerung	Anzahl der Potentiometer	Zugehöriges Datenregister
FX1S FX1N	2 (integriert)	D8030 (VR1) D8031 (VR2)
	optional 8	siehe Applikationsanweisung VRRD
FX2N FX2NC	optional 8	siehe Applikationsanweisung VRRD
FX3G FX3GE FX3S	2 (integriert)	D8030 (VR1) D8031 (VR2)
	optional 8	siehe Applikationsanweisung VRRD

Tab. 3-5: Zuordnung der Potentiometer

3.4.4 Genauigkeit der Timer

Der Ablaufvorgang eines Timers beginnt, sobald die Eingangsbedingung gesetzt ist.

Die Genauigkeit der Timer beträgt: $(T - \alpha) \leq T \leq (T + T_0)$

T: Zeitsollwert

T₀: Programmzykluszeit

α: Timer-Zeitschritt (100 ms, 10 ms, 1 ms)

Befindet sich die Setzanweisung des Timer-Arbeitskontakts im Programm vor der Festlegung des Timers, kann die Verzögerung maximal (+2 T₀) betragen.

Beträgt der Zeitsollwert T = 0, wird der Timer-Arbeitskontakt eingeschaltet, sobald im nächsten Programmzyklus die entsprechende Setzanweisung abgearbeitet wird.

3.4.5 Remanente Timer

Die Steuerungen der der FX1N-, der FX2N-, der FX2NC-, der FX3G-, der FX3U- und der FX3UC-Serie verfügen neben den bereits beschriebenen Zeitgliedern auch über remanente Zeitglieder, die auch nach dem Abschalten der ansteuernden Verknüpfung den bereits erreichten Zeitwert behalten.

Die Zeitistwerte werden in einem Speicher abgelegt, dessen Inhalt auch bei einem Spannungsausfall erhalten bleibt.

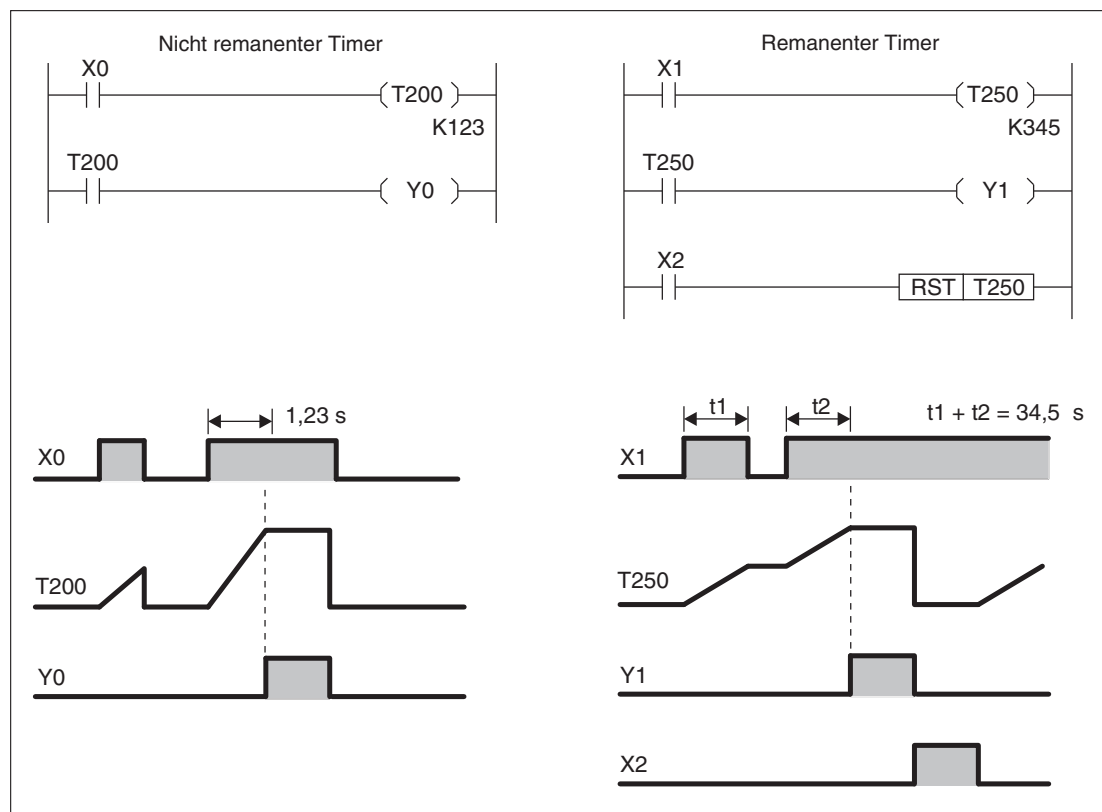


Abb. 3-8: Arbeitsweise der remanenten Timer

3.5 Counter

Damit Sie Zählvorgänge programmieren können, stellt Ihnen die FX-Familie mehrere interne Zähler (Counter) zur Verfügung.

Die Counter lassen sich in folgende Gruppen einteilen:

- 16-Bit-Counter, aufwärtszählend
Es werden programminterne Zählsignale verarbeitet. Der Zähleristwert bleibt bei einem Spannungsausfall der SPS nicht gespeichert.
- Remanente 16-Bit-Counter, aufwärtszählend
Es werden programminterne Zählsignale verarbeitet. Der Zähleristwert wird in einem spannungsausfallsicheren Speicher abgespeichert und bleibt bei einem Spannungsausfall der SPS erhalten.
- 32-Bit-Counter, auf-/ abwärtszählend
Es werden programminterne Zählsignale verarbeitet. Die Zählrichtung kann durch den Zustand eines Sondermerkers beeinflusst werden. Die Zähleristwerte bleiben bei einem Spannungsausfall der SPS nicht gespeichert.
- Remanente 32-Bit-Counter, auf-/ abwärtszählend
Es werden programminterne Zählsignale verarbeitet. Die Zählrichtung kann durch den Zustand eines Sondermerkers beeinflusst werden. Der Zähleristwert wird bei einem Spannungsausfall der SPS in einem spannungsausfallsicheren Speicher abgelegt.
- 32-Bit-High-Speed-Counter (schnelle Zähler), auf-/ abwärtszählend
High-Speed-Counter verarbeiten sehr schnell aufeinanderfolgende **externe** Zählsignale unabhängig von der benötigten Programmzykluszeit.

3.5.1 16-Bit-Counter

16-Bit-Counter adressieren

Die 16-Bit-Counter werden dezimal adressiert.

Steuerung	Operandenadresse	Anzahl	Spannungsausfallsicher
FX1S	C0–C15	16	Nein
	C16–C31	16	Ja
FX1N	C0–C15	16	Nein
	C16–C199	184	Ja
FX2N	C0–C99	100	Über Parameter wählbar*
FX2NC	C100–C199	100	
FX3G	C0–C15	16	Nein
FX3GC	C16–C199	184	Ja
FX3GE			
FX3S	C0–C15	16	Nein
	C16–C31	16	Ja
FX3U	C0–C99	100	Über Parameter wählbar*
FX3UC	C100–C199	100	

Tab. 3-6: 16-Bit-Counter und ihre Operandenadressen

* In den SPS-Parametern kann eingestellt werden, ob die Istwerte dieser Counter beim Ausschalten der Versorgungsspannung erhalten bleiben sollen.

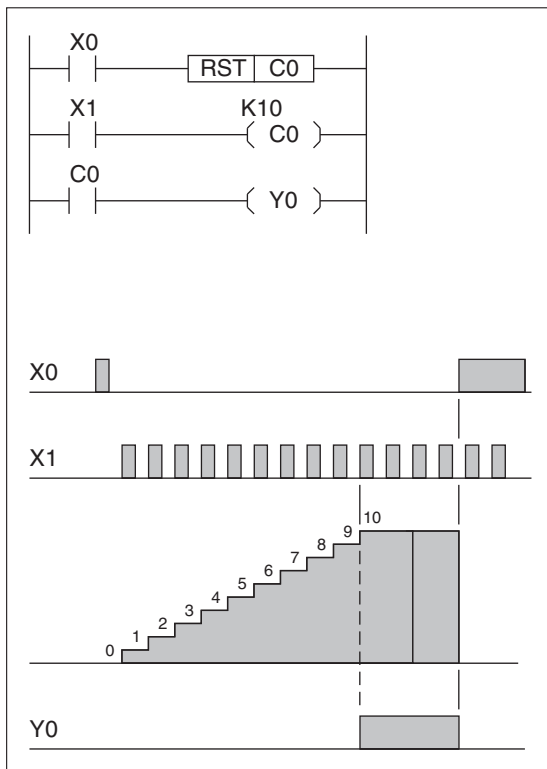
16-Bit-Counter programmieren

Der gewählte Zählersollwert wird durch eine zusätzliche Dezimalkonstante K festgelegt. Für die Dezimalkonstante K kann ein Zahlenwert zwischen +1 und +32767 verwendet werden.

Der Zählvorgang wird durch jedes Ansteuern mit einem „1“-Signal aktiviert. Der Zähleristwert wird dabei jeweils um den Wert 1 erhöht (Aufwärtszähler). Nach Erreichen des vorher festgelegten Zählersollwertes schaltet der Zähler auf den Signalzustand „1“.

HINWEIS

Die Angabe des Zählersollwertes kann auch indirekt über den in einem Datenregister gespeicherten dezimalen Zahlenwert vorgenommen werden.

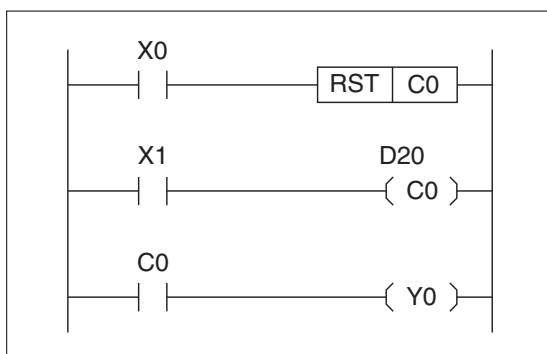
Beispiel ▾ Einsatz der 16-Bit-Counter mit direkter Vorgabe des Zählersollwertes**Abb. 3-9:**

Programmierbeispiel zum Einsatz der 16-Bit-Counter mit direkter Zählersollwertvorgabe

Bei jedem Ansteuern mit einem „1“-Signal am Eingang X1 zählt der Counter C0 um den Zahlenwert 1 aufwärts. Der Ausgang Y0 wird nach 10 Zählsignalen am Eingang X1 gesetzt (Zählersollwert K10).

Nach Erreichen des Zählersollwertes K10 bleibt der Counter von den dann folgenden Setzimpulsen am Eingang X1 unbeeinflusst.

Über den Eingang X0 wird der Zähler mit Hilfe einer RST-Anweisung zurückgesetzt. Der Istwert des Counters wird auf 0 gesetzt. Der Ausgang Y0 wird ausgeschaltet. △

Beispiel ▾ Einsatz der 16-Bit-Counter mit indirekter Zählersollwertvorgabe**Abb. 3-10:**

Programmierbeispiel zum Einsatz der 16-Bit-Counter mit indirekter Zählersollwertvorgabe

Der Zählersollwert wird indirekt über das Datenregister D20 festgelegt. △

3.5.2 32-Bit-Counter

32-Bit-Counter adressieren

Steuerung	Operandenadresse	Anzahl	Spannungsausfallsicher
FX1N	C200–C219	20	Nein
	C220–C234	15	Ja
FX2N FX2NC	C200–C219	20	Über Parameter wählbar*
	C220–C234	15	
FX3G FX3GC FX3GE	C200–C219	20	Nein
	C220–C234	15	Ja
FX3S	C200–C234	35	Nein
FX3U FX3UC	C200–C219	20	Über Parameter wählbar*
	C220–C234	15	

Tab. 3-7: 32-Bit-Counter und ihre Operandenadressen

* In den SPS-Parametern kann eingestellt werden, ob die Istwerte dieser Counter beim Ausschalten der Versorgungsspannung erhalten bleiben sollen.

Die Counter C200 bis C234 sind Auf-/Abwärtszähler; die Zählrichtung wird durch den Zustand eines zugeordneten Sondermerkers vorgegeben. Die Zählrichtung kann auch während des Zählvorgangs geändert werden.

Zugeordnete Sondermerker

Operandenadresse	C200	C201	C202	C203	C204	C205	C206
Sondermerker	M8200	M8201	M8202	M8203	M8204	M8205	M8206
Operandenadresse	C207	C208	C209	C210	C211	C212	C213
Sondermerker	M8207	M8208	M8209	M8210	M8211	M8212	M8213
Operandenadresse	C214	C215	C216	C217	C218	C219	C220
Sondermerker	M8214	M8215	M8216	M8217	M8218	M8219	M8220
Operandenadresse	C221	C222	C223	C224	C225	C226	C227
Sondermerker	M8221	M8222	M8223	M8224	M8225	M8226	M8227
Operandenadresse	C228	C229	C230	C231	C232	C233	C234
Sondermerker	M8228	M8229	M8230	M8231	M8232	M8233	M8234

Tab. 3-8: Zuordnung der Sondermerker zu den Zählern

Die Zählrichtung der Zähler wird durch den logischen Zustand des zugeordneten Sondermerkers festgelegt.

- Sondermerker eingeschaltet (1): Abwärtszähler
- Sondermerker ausgeschaltet (0): Aufwärtszähler

Die Arbeitsweise entspricht der eines 16-Bit-Zählers.

HINWEIS

Bei indirekter Adressierung werden 2 Datenregister benötigt. Zur Zuweisung von Sollwerten müssen 32-Bit-Anweisungen verwendet werden.

Beispiel ▾ Einsatz der 32-Bit-Counter mit direkter Zählersollwertvorgabe

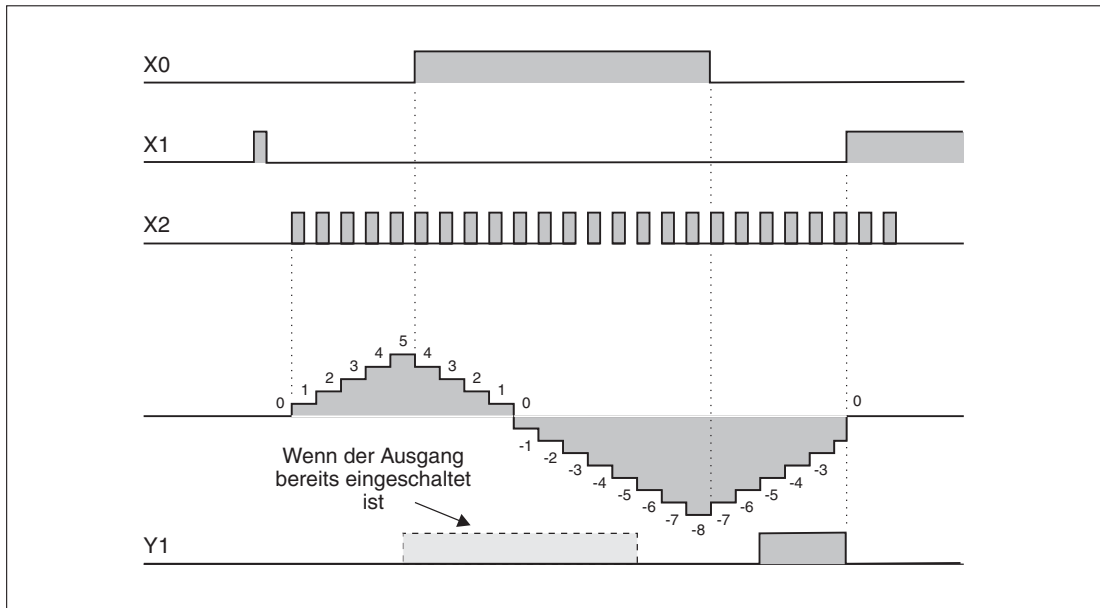


Abb. 3-11: Programmierbeispiel zum Einsatz der 32-Bit-Counter bei direkter Vorgabe des Zählersollwertes

Sobald der Eingang X2 einschaltet, beginnt der Zählvorgang. Der Zähler C200 zählt die Einschaltimpulse von X2.

Der Ausgang Y1 wird eingeschaltet, wenn der Istwert von -6 auf -5 springt. Y1 wird zurückgesetzt, wenn der Istwert von -5 auf -6 springt.

Der Zählvorgang (aufwärts und abwärts) findet unabhängig vom aktuellen Status des Ausgangs statt. Wenn der Counter jedoch oberhalb von +2147483647 arbeitet, wird automatisch der Wert -2147483648 gültig. Wird unterhalb von -2147483648 gezählt, wird der Wert +2147483647 gültig.

Diese Counter werden „Ring-Counter“ genannt.

Über den Eingang X1 wird die RST-Anweisung ausgeführt. Der Istwert des Counters wird auf 0 gesetzt. Der Ausgang Y1 wird ausgeschaltet. △

Beispiel ▾ Einsatz der 32-Bit-Counter mit indirekter Zählersollwertvorgabe

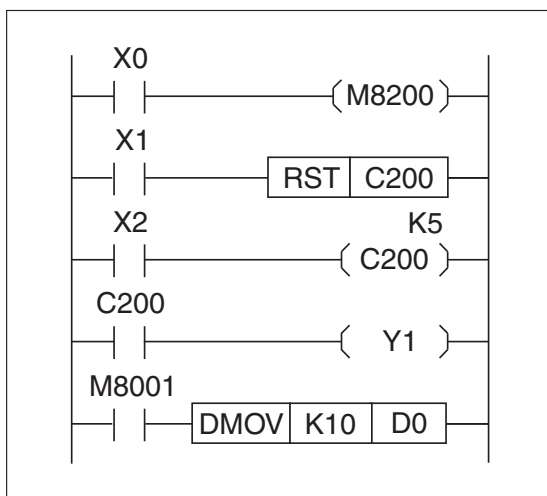


Abb. 3-12: Programmierbeispiel zum Einsatz der 32-Bit-Counter bei indirekter Zählersollwertvorgabe

△

3.5.3 32-Bit-High-Speed-Counter

Die High-Speed-Counter sind 32-Bit-Counter, die schnelle externe Zählsignale verarbeiten. Als Zähl Eingänge stehen die Eingänge X0 bis X7 zur Verfügung. Die Eingänge X6 und X7 arbeiten – außer bei der FX3G, FX3GC, FX3GE, FX3S, FX3U und FX3UC – nur als Startsignale. Sie können nicht für den High-Speed-Zählvorgang verwendet werden. Da eine Doppelbelegung der Eingänge nicht erlaubt ist, stehen pro SPS-Programm maximal sechs High-Speed-Counter zur Verfügung (8 Counter bei FX3G, FX3GC, FX3GE, FX3S, FX3U und FX3UC).

Den High-Speed-Countern sowie den verschiedenen Eingängen sind festgelegte Funktionen zugeordnet.

High-Speed-Counter arbeiten nach dem Prinzip des Interrupts. Der Vorteil hierbei ist, dass das Zählsignal unabhängig von der Programmzykluszeit verarbeitet wird.

Zuordnung der Eingänge

X	1-Phasen-Counter ohne Start und Reset						1-Phasen-Counter mit Start und Reset					2-Phasen-Counter Bidirektional					A/B-Phasen-Counter						
	C 235	C 236	C 237	C 238	C 239	C 240	C 241	C 242	C 243	C244	C245	C 246	C 247	C248	C 249	C 250	C 251	C 252	C253	C 254	C 255		
X0	U/D						U/D			U/D		U	U			U		A	A			A	
X1		U/D					R			R		D	D			D		B	B			B	
X2			U/D					U/D			U/D		R			R			R			R	
X3				U/D				R			R			U	U*		U			A	A*		A
X4					U/D				U/D					D	D*		D			B	B*		B
X5						U/D			R					R		R			R			R	
X6										S	U/D*					S						S	
X7											S	U/D*				S						S	

Tab. 3-9: High-Speed-Counter und die zugehörigen Zähl Eingänge

* Bei einer FX3G, FX3U oder FX3UC können die Eingänge der Counter C244, C245, C248 und C253 durch die Sondermerker M8390 bis M8392 umgeschaltet werden. (M8392 beeinflusst C248 und C253.)
Bei einer FX3S können die Eingänge der Counter C248 und C253 durch den Sondermerker M8392 umgeschaltet werden.

U: Aufwärtszählender Eingang
D: Abwärtszählender Eingang
A: A-Phasen-Eingang
B: B-Phasen-Eingang
R: Reset-Eingang
S: Start-Eingang

Es können mehrere High-Speed-Counter gleichzeitig im SPS-Programm eingesetzt werden.

Beim Einsatz verschiedener High-Speed-Counter muss darauf geachtet werden, dass kein Counter verwendet wird, dessen Eingänge bereits durch einen anderen Counter belegt sind. Eine Doppelbelegung von Eingängen ist nicht erlaubt.

Maximale Zählfrequenz und Zählgeschwindigkeit

Die maximale Frequenz bzw. Geschwindigkeit der Zählsignale, die noch von der SPS verarbeitet werden, beträgt beim Einsatz von nur einem High-Speed-Counter in einem SPS-Programm:

- 1- und 2-Phasen-Counter: max. 100 kHz bei FX3U und FX3UC, max. 60 kHz bei FX1N, FX1S, FX3G, FX3GC, FX3GE und FX3S, max. 10 kHz bei FX2N und FX2NC
- A/B-Phasen-Counter: max. 50 kHz bei FX3U und FX3UC, max. 30 kHz bei FX1N, FX1S, FX3G, FX3GC, FX3GE und FX3S, max. 5 kHz bei FX2N und FX2NC

Die Summe aller Zählfrequenzen darf bei einer Steuerung der FX1S-, FX1N-, FX3G-, FX3GC-, FX3GE- oder FX3S-Serie 60 kHz, bei einer Steuerung aus der FX2N- oder FX2NC-Serie 30 kHz und einer FX3U- oder FX3UC-SPS 80 kHz nicht überschreiten. Dabei ist zu beachten, dass die Frequenz von A/B-Phasen-Countern in diese Berechnung mit dem doppelten Wert eingeht.

Die Eingänge X0 und X1 einer Steuerung der FX2N- oder FX2NC-Serie und die Eingänge X0 bis X5 einer Steuerung der FX3G-, FX3GC-, FX3GE-, FX3S-, FX3U- oder FX3UC-Serie sind durch ihren Aufbau in der Lage, sehr hohe Frequenzen zu zählen.

- FX2N/FX2NC
 - 1-Phasen-Counter C235, C236 oder C246: 60 kHz
 - 2-Phasen-Counter C251: 30 kHz
- FX3G/FX3GC/FX3GE
 - 1-Phasen-Counter C235, C236, C238, C239 oder C241: 60 kHz
 - 2-Phasen-Counter C246: 60 kHz
- FX3S
 - 1-Phasen-Counter C235, C236 oder C241: 60 kHz
 - 2-Phasen-Counter C246: 60 kHz
- FX3U/FX3UC
 - 1-Phasen-Counter C235 bis C240 oder C246: 100 kHz
 - 2-Phasen-Counter C251 und C253: 50 kHz

Wenn Anweisungen zum Setzen- oder Rücksetzen von High-Speed-Countern (Anweisungen DHSCS und DHSCR) oder ein Bereichsvergleich (DHSZ) im Programm verwendet werden, wird die zulässige Summe der Eingangsfrequenzen aller Zähler reduziert. In diesem Fall gilt die folgende Tabelle:

Steuerung	Anweisung	Summe aller Zählfrequenzen
FX1S FX1N	DHSCS oder DHSCR	30 kHz
FX2N FX2NC	DHSCS oder DHSCR	11 kHz
	DHSZ	5,5 kHz
FX3G, FX3GC, FX3GE, FX3S FX3U FX3UC	DHSCS, DHSCR oder DHSZ	60 kHz

Tab. 3-10:

Einschränkungen bei Verwendung der Anweisungen DHSCS, DHSCR und DHSZ

Die Istwerte aller High-Speed-Counter werden in einem spannungsausfallsicheren Speicher abgelegt.

HINWEISE

Die Zählengänge X0 bis X5 dürfen nicht als Einschaltbedingungen für High-Speed-Counter programmiert werden.

High-Speed-Counter können nicht im Zusammenhang mit 16-Bit-Anweisungen eingesetzt werden.

Die maximale Zählfrequenz an den SPS-Eingängen muss beachtet werden. (AB-Phasen-Counter sind doppelt zu zählen).

Die SPD-Anweisung hat die Counter- und Interrupt-Charakteristik eines High-Speed-Counters. Aus diesem Grund sollten für die SPD-Anweisung die Eingänge X0 bis X5 verwendet werden. Auch für diese Eingänge gilt, dass sie nicht gleichzeitig von anderen High-Speed-Countern verwendet werden können.

1-Phasen-Counter mit einem Zählengang

1-Phasen-Counter sind High-Speed-Counter mit nur einem Zählengang. Sie lassen sich in drei Gruppen einteilen:

- Ohne Start- und Reset-Eingang (C235 bis C240)
- Mit Reset-Eingang (C241 bis C243)
- Mit Start- und Reset-Eingang (C244 bis C245)

Die Zählrichtung (auf- oder abwärtszählend) wird durch Einschalten eines Sondermerkers festgelegt.

Sondermerker eingeschaltet: Abwärtszählend

Sondermerker ausgeschaltet: Aufwärtszählend

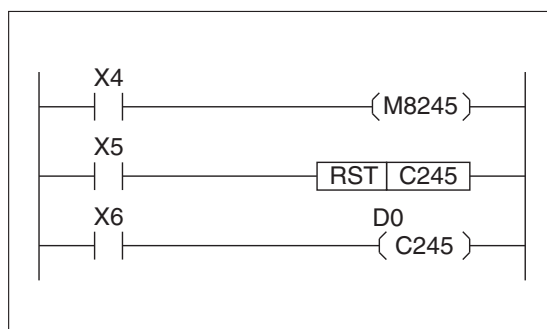
1-Phasen-Counter	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C245
Sondermerker	M8235	M8236	M8237	M8238	M8239	M8240	M8241	M8242	M8243	M8244	M8245

Tab. 3-11: 1-Phasen-Counter und die zugehörigen Sondermerker

Der Zählbereich geht von -2 147 483 648 bis 2 147 483 647.

Beispiel ▾

Einsatz eines 1-Phasen-Counters mit Start- und Reset-Eingang (C244).

**Abb. 3-13:**

Programmierbeispiel zum Einsatz eines 1-Phasen-Counters mit Start- und Reset-Eingang (C244)

Wenn der Sondermerker M8245 eingeschaltet ist, zählt der Zähler M8245 abwärts. Ist der Merker M 8245 nicht gesetzt, zählt der Zähler aufwärts. Mit dem Schalter X5 wird der Istwert des Zählers auf den Wert 0 zurückgesetzt. Dies ist ebenfalls über den automatisch zugeordneten Rücksetzeingang X3 möglich.

Durch das Einschalten von X6 und dem automatisch zugeordneten Starteingang X7 wird der Zähler aktiviert und zählt die Impulse aus seinem Zählengang X2. Da es sich um einen 32-Bit-Zähler handelt, werden die Register D0 und D1 für die Sollwertvorgabe genutzt.

Beispiel ▾

Im Vergleich zu den im Programm verwendeten Eingängen X5 und X6 bietet die Benutzung der Eingänge X7 und X3 den Vorteil, dass die Verarbeitung der externen Start- und Reset-Signale nicht von der Programmzykluszeit abhängt.

△

2-Phasen-Counter mit zwei Zähleringängen

2-Phasen-Counter besitzen je einen Zähleringang zum Auf- und Abwärtszählen.

Die 2-Phasen-Counter lassen sich in drei Gruppen einteilen:

- Ohne Start- und Reset-Eingang (C246)
- Mit Reset-Eingang (C247, C248)
- Mit Start- und Reset-Eingang (C249, C250)

Die SPS setzt automatisch einen Sondermerker, der die aktuelle Zählrichtung des 2-Phasen-Counters anzeigt:

Sondermerker eingeschaltet: Abwärtszählend

Sondermerker ausgeschaltet: Aufwärtszählend

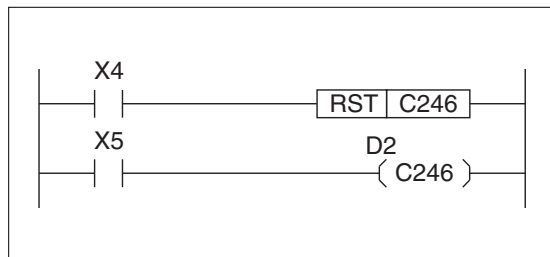
2-Phasen-Counter	C246	C247	C248	C249	C250
Sondermerker	M8246	M8247	M8248	M8249	M8250

Tab. 3-12:
2-Phasen-Counter und zugehörige Sondermerker

Der Zählbereich geht von -2 147 483 648 bis 2 147 483 647.

Beispiel ▾

| Einsatz eines 2-Phasen-Counters ohne Start- und Reset-Eingang (C246).

**Abb. 3-14:**

Programmierbeispiel zum Einsatz eines 2-Phasen-Counters ohne Start- und Reset-Eingang (C246)

Wird der Eingang X4 eingeschaltet, wird der Counter C246 zurückgesetzt. Der Zählvorgang wird gestartet, wenn der Eingang X5 eingeschaltet ist. Für den Counter C246 sind die Zähleringänge X0 und X1 zum Aufwärts- und Abwärtszählen reserviert.

Bei Signalimpulsen am Eingang X0 zählt der Counter aufwärts, und bei Signalimpulsen am Eingang X1 zählt der Counter abwärts.

△

A/B-Phasen-Counter mit zwei Zählwegen

Die A/B-Phasen-Counter verfügen über je einen A- und einen B-Phasen-Zählweg. Über die Signale an den A- und B-Phasen-Eingängen wird festgelegt, ob der Counter aufwärts oder abwärts zählen soll.

- Aufwärtszählend

A-Phasen-Eingang: „1“-Signal

B-Phasen-Eingang: ansteigende Signalfanke (Signalwechsel von „0“ auf „1“)

- Abwärtszählend

A-Phasen-Eingang: „1“-Signal

B-Phasen-Eingang: abfallende Signalfanke (Signalwechsel von „1“ auf „0“)

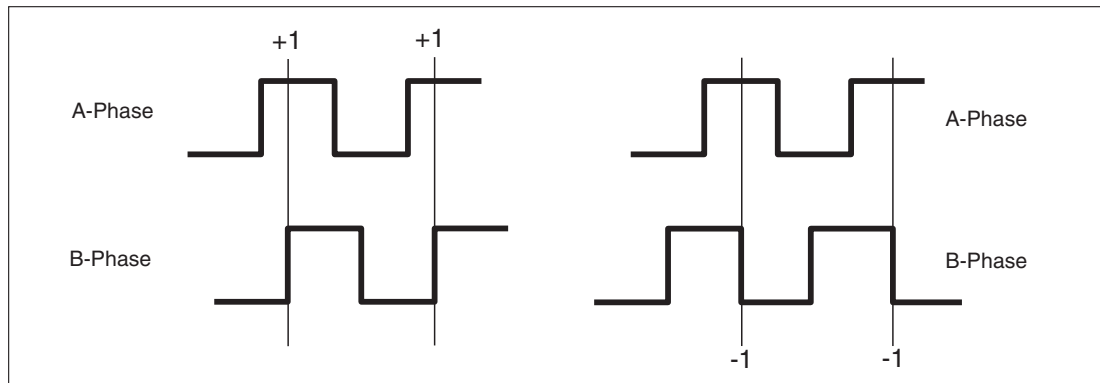


Abb. 3-15: AB-Phasen-Counter

Jeder Eingangssignalwechsel mit ansteigender Flanke am B-Phasen-Eingang lässt den Zähler um 1 aufwärtszählen und jeder Signalwechsel mit abfallender Flanke um 1 abwärtszählen. Während der Signalwechsel muss am A-Phasen-Eingang ein „1“-Signal anliegen.

Die A/B-Phasen-Counter lassen sich in drei Gruppen einteilen:

- Ohne Start- und Reset-Eingang (C251)
- Mit Reset-Eingang (C252, C253)
- Mit Start- und Reset-Eingang (C254, C255)

Die SPS setzt automatisch einen Sondermerker, der die aktuelle Zählrichtung des 2-Phasen-Counters anzeigt:

Sondermerker eingeschaltet: Abwärtszählend

Sondermerker ausgeschaltet: Aufwärtszählend

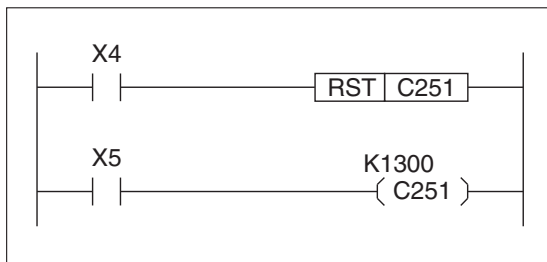
AB-Phasen-Counter	C251	C252	C253	C254	C255
Sondermerker	M8251	M8252	M8253	M8254	M8255

Tab. 3-13:
A/B-Phasen-Counter und zugehörige Sondermerker

Der Zählbereich geht von -2 147 483 648 bis 2 147 483 647.

Beispiel ▾

Einsatz eines A/B-Phasen-Counters ohne Start- und Reset-Eingang (C251)

**Abb. 3-16:**

Programmierbeispiel zum Einsatz eines AB-Phasen-Counters ohne Start- und Reset-Eingang (C251)

Bei eingeschaltetem Eingang X5 zählt der Counter C251 die Signale an den Zählengängen X0 (A-Phasen-Eingang) und X1 (B-Phasen-Eingang). △

3.6 Schrittstatus

Schrittstatusoperanden werden in Zusammenhang mit Ablaufsteuerungen (STL-Anweisung) eingesetzt. Mit den Schrittstatusoperanden werden die einzelnen Schritte einer Ablaufsteuerung festgelegt.

3.6.1 Schrittstatusoperanden adressieren

Es stehen bis zu 4096 Schrittstatusoperanden im Bereich von S0 bis S4095 zur Verfügung.

Die Schrittstatusoperanden S lassen sich in zwei Gruppen einteilen:

- S0 bis S9 (10 Adressen): Schrittstatus initialisieren
- S10 bis max. S4095: frei wählbarer Einsatzbereich

Steuerung	Operanden	Davon gepuffert
FX1S	S0 – S127 (128)	S0 – S127 (128)
FX1N	S0 – S999 (1000)	S0 – S999 (1000)
FX2N	S0 – S999 (1000)	S500 – S999 (500)
FX2NC		
FX3G	S0 – S4095 (4096)	S0 – S999 (1000)
FX3GC		
FX3GE		
FX3S	S0 – S255 (256)	S0 – S127 (128)
FX3U	S0 – S4095 (4096)	S500 – S4095 (3596)
FX3UC		

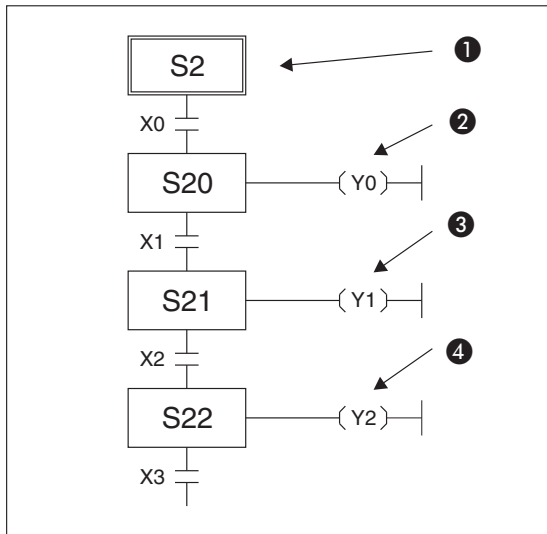
Tab. 3-14:
Übersicht der Schrittstatusoperanden

Detaillierte Informationen zum Einsatz der STL-Anweisung und der Schrittstatusoperanden S enthält Kapitel 5

Werden in einem Programm keine Schrittsteuerungen verwendet, können die Schrittstatusoperanden S wie Merker eingesetzt werden.

Beispiel ▾

Einsatz der Schrittstatusoperanden

**Abb. 3-17:**

Programmierbeispiel zum Einsatz der Schrittstatusoperanden

- ① Initialisierungsstatus
- ② Absenken
- ③ Greifen
- ④ Anheben

Der Schrittstatusoperand S20 wird aktiviert, wenn der Eingang X0 eingeschaltet wird. Anschließend wird der Ausgang Y0 (② Absenken) eingeschaltet.

Nach Erreichen des unteren Endschalters X1 wird der Schrittstatusoperand S21 aktiviert, indem der Ausgang Y1 (③ Greifen) eingeschaltet wird.

Wird der Endschalter X2 erreicht, wird S22 aktiviert. Der Ausgang Y2 (④ Anheben) wird eingeschaltet. △

3.7 Konstanten

3.7.1 Dezimal- und Hexadezimalkonstanten

Mit den Dezimal- und Hexadezimalkonstanten (K, H) lassen sich numerische Zahlenwerte innerhalb eines SPS-Programms festlegen (z. B. Zeit- oder Zählerollwert). Der Zahlenwert wird intern von der SPS in einen binären Zahlenwert codiert.

Im Abschnitt 3.8.7 sind alle wichtigen Zahlensysteme und deren Codierungen ausführlich beschrieben.

Konstanten	16 Bit	32 Bit
Dezimal K	-32 768 bis +32 767	-2 147 483 648 bis +2 147 483 647
Hexadezimal H	0 bis FFFF	0 bis FFFFFFFF

Tab. 3-15: Zahlenwertebereiche der Dezimal- und Hexadezimalkonstanten

Im Programm wird der Konstanten der Buchstabe „K“ oder „H“ vorangestellt.

Beispiele: K100 (Dezimaler Wert „100“), H64 (Hexadezimaler Wert „64“)

3.7.2 Konstanten mit Gleitkommazahlen

Dezimale Konstanten sind ganze Zahlen ohne Nachkommastellen. Gleitkommazahlen dagegen können Stellen vor und nach einem Dezimalkomma aufweisen und bieten dadurch Vorteile bei arithmetischen Operationen.

Im Programm werden Konstante, die aus Gleitkommazahlen bestehen, durch ein vorangestelltes „E“ gekennzeichnet (Zum Beispiel E1.234 oder E1.234 + 3). Gleitkommazahlen können auf zwei verschiedene Arten definiert werden:

- Angabe einer Konstanten ohne Exponenten
Der Wert wird wie üblich angegeben. Das Komma muss allerdings durch einen Punkt ersetzt werden. Zum Beispiel kann der Wert „10,2345“ im Programm als „E10.2345“ übergeben werden.
- Angabe einer Konstanten mit Exponenten
Der Wert wird mit Basis und einem Exponenten angegeben. Der Exponent hat die Basis 10 (10^n). Der Wert „1234“ zum Beispiel kann auch als „1,234 x 1000“ oder – in exponentieller Schreibweise – als „1,234 x 10^3 “ dargestellt werden. Im Programm einer FX3U-Steuerung wird diese Zahl als „E1.234 + 3“ angegeben („+3“ entspricht „ 10^3 “).

Gleitkommazahlen können Werte aus den folgenden Bereichen annehmen:

$-1,0 \times 2^{128}$ bis $-1,0 \times 2^{-126}$,
0
und $1,0 \times 2^{-126}$ bis $1,0 \times 2^{+128}$

3.7.3 Konstante Zeichenfolgen

Werden Zeichen innerhalb eines Programms in Anführungsstrichen angegeben, werden sie als ASCII-Zeichen interpretiert (z. B. "MOTOR12"). Eine Zeichen belegt 1 Byte. Eine Zeichenfolge kann bis zu 32 Zeichen enthalten.

3.8 Register

Register stellen einen Datenspeicher innerhalb der SPS dar. In einem Register können Sie Zahlenwerte und aufeinanderfolgende binäre Informationen zusammenfassen und abspeichern. Dadurch lassen sich z. B. die Signalzustände mehrerer Eingänge auf einmal abspeichern und im Programm verarbeiten.

Die Daten werden in einem 16-Bit-Format gespeichert. Durch Zusammenschalten von zwei 16-Bit-Registern haben Sie die Möglichkeit, „Doppelregister“ zu bilden. In einem Doppelregister können Daten in einem 32-Bit-Format gespeichert werden.

3.8.1 Einteilung der Register

Folgende Registertypen werden unterschieden:

- **Datenregister** (ungepuffert)

Register ohne Datensicherung bei einem Spannungsausfall der SPS.

- **Datenregister** (gepuffert)

Register mit Datensicherung bei einem Spannungsausfall der SPS. Die Daten werden in einem spannungsausfallsicheren Speicher abgespeichert.

- **Index-Register**

Diese Register dienen zum Speichern von Zwischenergebnissen und zur Indizierung von Operanden (siehe Abschnitt 3.8.5).

- **Sonderregister**

Für bestimmte Kontroll- oder Überwachungsfunktionen stehen eine Reihe von speziellen Registern zur Verfügung. Nähere Angaben siehe Abschnitt 3.8.4.

- **File-Register**

Zum Speichern von beispielsweise Parametern oder Rezepturen werden File-Register benötigt. Die Speicherbereiche dieser Register werden vom Anwender festgelegt. Diese File-Register sind ein Teil der Datenregister.

- **Erweiterte Register**

Die erweiterten Register (Operandenkennzeichen: R) sind eine erweiterte Form der Datenregister. Erweiterte Register stehen nur bei den Grundgeräten der FX3G-, FX3GC-, FX3GE-, FX3U- und FX3UC-Serie zur Verfügung.

- **Erweiterte File-Register**

In erweiterten File-Registern (Operandenkennzeichen: ER) kann der Inhalt von erweiterten Registern (R) gespeichert werden. Erweiterte File-Register stehen nur bei den Grundgeräten der FX3G-, FX3GC-, FX3GE-, FX3U- und FX3UC-Serie zur Verfügung. Bei einem FX3U- oder FX3UC-Grundgerät muss eine Speicherkassette installiert sein, um die erweiterten File-Register nutzen zu können.

3.8.2 Aufbau der Register

Jedes Register besteht aus einem Vorzeichenbit und mehreren Datenbits.

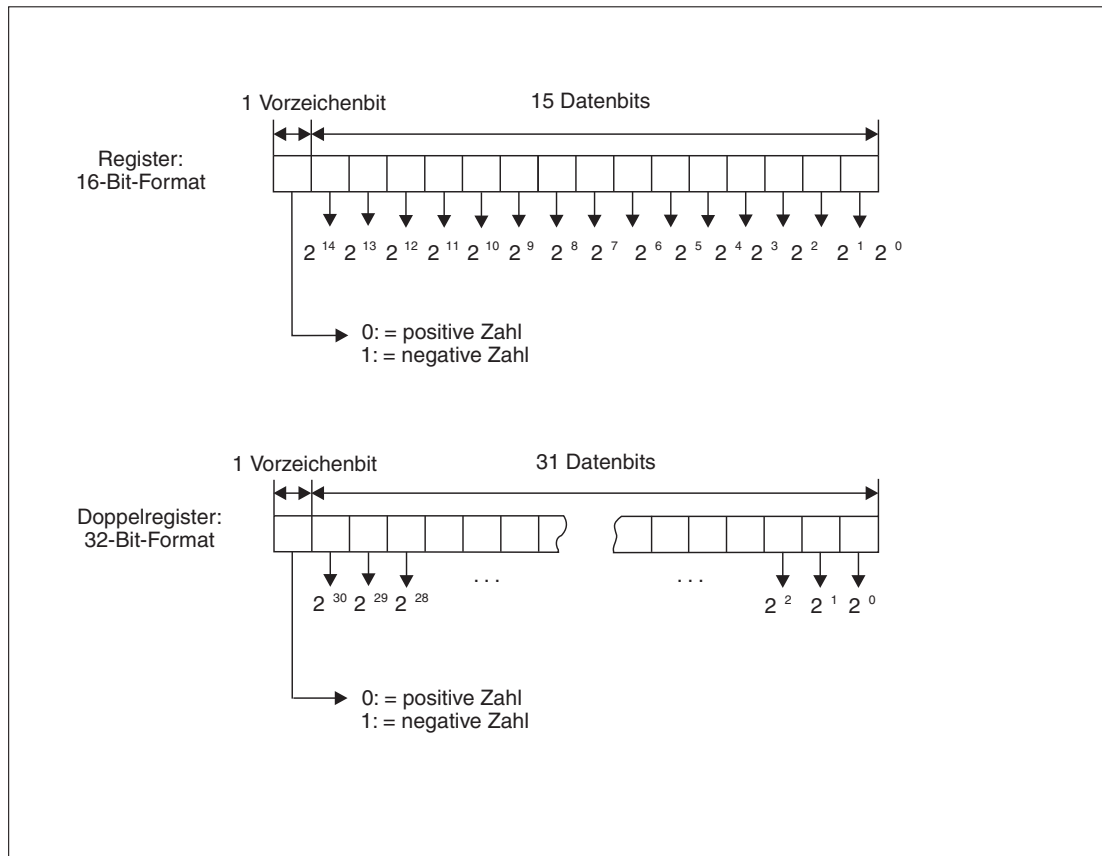


Abb. 3-18: Aufbau der Register (16 Bit) und Doppelregister (32 Bit)

3.8.3 Adressierung der Register

Die Adressierung der Datenregister erfolgt dezimal. Bei einem Doppelregister beginnt die Adressierung mit dem unteren 16-Bit-Register.

Steuerung	Register	Adressen	Anzahl	Davon gepuffert	Anzahl
FX1S	Datenregister	D0–D255	256	D128–D255	128
	Index-Register	V, Z	16	—	—
	Sonderregister	D8000–D8255	256	D8000–D8255	256
	Extern veränderbare Register ^①	D8030 und D8031	2	—	—
	File-Register	D1000–D2499	1500	D1000–D2499	1500
	Erweiterte Register	—	—	—	—
	Erweiterte File-Register	—	—	—	—
FX1N	Datenregister	D0–D7999	8000	D128–D7999	7872
	Index-Register	V, Z	16	—	—
	Sonderregister	D8000–D8255	256	D8000–D8255	256
	Extern veränderbare Register ^①	D8030 und D8031	2	—	—
	File-Register	D1000–D7999	7000 (anteilig)	D1000–D7999	7000 (anteilig)
	Erweiterte Register	—	—	—	—
	Erweiterte File-Register	—	—	—	—
FX2N FX2NC	Datenregister	D0–D7999	8000	D200–D7999 ^③	7800
	Index-Register	V0–V7, Z0–Z7	16	—	—
	Sonderregister	D8000–D8255	256	D8000–D8255	256
	File-Register	D1000–D7999	7000 (anteilig)	D1000–D7999	7000 (anteilig)
	Erweiterte Register	—	—	—	—
	Erweiterte File-Register	—	—	—	—
FX3G FX3GC FX3GE	Datenregister	D0–D7999	8000	D128–D1099 ^④	972
	Index-Register	V0–V7, Z0–Z7	16	—	—
	Sonderregister	D8000–D8511	512	D8000–D8511	512
	Extern veränderbare Register ^{①②}	D8030 und D8031	2	—	—
	File-Register	D1000–D7999	7000 (anteilig)	D1000–D7999	7000 (anteilig)
	Erweiterte Register	R0–R23999 ^⑤	24000	—	—
	Erweiterte File-Register	ER0–ER23999 ^⑥	24000	—	—

Tab. 3-16: Register und ihre Operandenadressen

Steuerung	Register	Adressen	Anzahl	Davon gepuffert	Anzahl
FX3S	Datenregister	D0–D255	256	D128–D255	128
	Index-Register	V0–V7, Z0–Z7	16	—	—
	Sonderregister	D8000–D8511	512	D8000–D8511	512
	Extern veränderbare Register ^①	D8030 und D8031	2	—	—
	File-Register	D1000–D2999	2000 (anteilig)	D1000–D2999	2000 (anteilig)
	Erweiterte Register	—	—	—	—
	Erweiterte File-Register	—	—	—	—
FX3U FX3UC	Datenregister	D0–D7999	7000 (anteilig)	D200–D7999 ^③	7800
	Index-Register	V0–V7, Z0–Z7	16	—	—
	Sonderregister	D8000–D8511	512	D8000–D8511	512
	Extern veränderbare Register	—	—	—	—
	File-Register	D1000–D7999	7000 (anteilig)	D1000–D7999	7000 (anteilig)
	Erweiterte Register	R0–R32767	32768	R0–R32767	32768
	Erweiterte File-Register	ER0–ER32767	32768	ER0–ER32767 ^⑦	32768

Tab. 3-17: Register und ihre Operandenadressen

- ① Diese beiden Register sind auch bei der Anzahl der Sonderregister enthalten. Werte von 0 bis 255 sind durch integrierte Potentiometer einstellbar.
- ② Nicht bei einer FX3GC.
- ③ Von dem angegebenen Datenregisterbereich der FX2N-, der FX2NC- und der FX3U/FX3UC-Serie kann durch den Anwender nur bei den Registern D200 bis D511 bestimmt werden, ob diese gepuffert sind oder nicht. Die Datenregister ab D512 sind bei diesen Steuerungen immer gepuffert. Den Registern D0 bis D199 kann in den SPS-Parametern auch die Funktion von gepufferten Registern zugewiesen werden.
- ④ Wenn die optionale Batterie installiert ist, kann den Registern D1100 bis D7999 in den SPS-Parametern die Funktion von gepufferten Registern zugewiesen werden.
- ⑤ Wenn die optionale Batterie installiert ist, kann den erweiterten Registern R0 bis R23999 in den SPS-Parametern die Funktion von gepufferten Registern zugewiesen werden. Der gepufferte Bereich ist jedoch nicht einstellbar.
- ⑥ Die erweiterten File-Register werden im integrierten EEPROM des SPS-Grundgeräts oder, wenn eine EEPROM-Speicherkassette installiert ist, in der Speicherkassette gespeichert.
- ⑦ Bei einem FX3U- oder FX3UC-Grundgerät stehen die erweiterten File-Register nur zur Verfügung, wenn eine Speicherkassette installiert sein. (Die erweiterten File-Register werden in der Speicherkassette gespeichert.)

Bei einer FX3U oder FX3UC können einzelne Bits eines Datenregisters als Bit-Operanden angegeben werden. Die Adressierung erfolgt in der Form „D□.b“. Zum Beispiel wird mit D10.7 das Bit 7 im Datenregister 10 angesprochen.

HINWEIS

Ein als „D□.b“ angegebener Operand kann nicht über Index-Register (V, Z) verändert werden.

3.8.4 Verwendung der Sonderregister

Den Sonderregistern D8000 bis D8511 sind intern feste Kontroll- oder Überwachungsfunktionen (Monitor-Funktionen) zugeordnet.

Beim Einschalten der Steuerung werden von der Systemsoftware automatisch die Standardwerte in die Sonderregister geschrieben. So wird z. B. automatisch der Datenwert des Watch-Dog-Timers in das Sonderregister D8000 eingetragen. Möchten Sie diesen Datenwert verändern, dann müssen Sie den alten Datenwert mit Hilfe einer MOV-Anweisung überschreiben (Weitere Hinweise siehe Abschnitt 6.3.3).

HINWEIS

Die Daten gehen beim Schalten der Steuerung in den STOP-Modus nicht verloren. Ein Spannungsausfall führt jedoch zu einem Datenverlust.

In Kapitel 9 sind alle vorhandenen Sonderregister und deren Funktionen aufgeführt.

3.8.5 Extern veränderbare Register

Bei den Steuerungen der FX1S-, FX1N-, FX3G-, FX3GE- und der FX3S-Serie sind zwei Potentiometer integriert, mit denen der Inhalt zweier Sonderregister im Bereich von 0 bis 255 verändert werden kann. Für Steuerungen der FX2N-Serie ist der Sollwertvorgabe-Adapter FX2N-8AV-BD erhältlich, mit dem die Eingabe von 8 analogen Sollwerten möglich ist. Ein Grundgerät der FX1S-, FX1N-, FX3G-, FX3GE- oder der FX3S-Serie kann zusätzlich mit einem Sollwertvorgabe-Adapter FX□□-8AV-BD um weitere 8 Potentiometer erweitert werden.

Zum Einlesen der Sollwerte sowie der Bereichseinteilung der Potentiometer werden die Applikationsanweisungen VRRD (FNC85) und VRSC (FNC86) verwendet.

Mit Hilfe der Potentiometer können z. B. Sollwerte für Timer und Counter verändert werden, ohne ein Programmiergerät anschließen zu müssen (siehe Abschnitt 8.6).

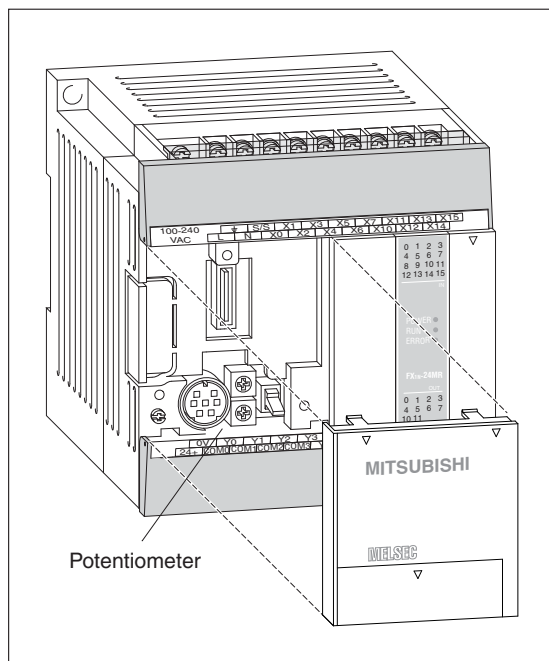


Abb. 3-19:
Potentiometer an einer Steuerung der FX1N-Serie

Merkmal	FX1S	FX1N	FX2N	FX3G, FX3GE, FX3S
Anzahl der Potentiometer	2 integrierte Potentiometer (VR1 und VR2), bei Verwendung des Sollwertvorgabe-Adapters FX1N-8AV-BD können 8 zusätzliche Potentiometer genutzt werden		Bei Verwendung des Sollwertvorgabe-Adapters FX2N-8AV-BD können 8 Potentiometer genutzt werden.	2 integrierte Potentiometer (VR1 und VR2), zusätzliche 8 Potentiometer können bei Verwendung des Sollwertvorgabe-Adapters FX3G-8AV-BD genutzt werden
Extern veränderbare Register	D8030 (wird von VR1) beeinflusst D8031 (wird von VR2) beeinflusst Die zusätzlichen 8 Register bei Verwendung des Sollwertvorgabe-Adapters FX1N-8AV-BD werden durch den Anwender mit den Anweisungen VRRD und VRSC bestimmt.		Die 8 Register werden durch den Anwender mit den Anweisungen VRRD und VRSC festgelegt.	D8030 (VR1) D8031 (VR2) Die zusätzlichen 8 Register bei Verwendung des Sollwertvorgabe-Adapters FX3G-8AV-BD werden mit den Anweisungen VRRD und VRSC festgelegt.

Tab. 3-18: Extern veränderbare Register

3.8.6 Indexvergabe

Durch Index-Register können Operanden indirekt adressiert werden. Bei Verwendung der Index-Vergabe in einem Programm erhält der Operand die direkt eingegebene Operandenadresse plus den Inhalt des Index-Registers als Adresse (siehe auch Abschnitt 6.1.6).

Es stehen 16 Index-Register (V0 bis V7 und Z0 bis Z7) zur Verfügung. Wird im Programm nur „V“ oder „Z“ angegeben, wird dies wie „V0“ bzw. „Z0“ behandelt. Index-Register selbst können nicht indiziert werden.

Angabe von Operanden mit Hilfe von Index-Registern

- Dezimal adressierbare Operanden und dezimale numerische Werte

Bei dezimal adressierbaren Operanden (M, S, T, C, D, R, KnM, KnS, P) und dezimalen Konstanten (K) wird der Inhalt eines Index-Registers als dezimaler Wert behandelt.

Beispiele ▾

Enthält V0 den Wert „5“, wird bei der Angabe des Operanden als „D20V0“ eine Anweisung mit dem Register D25 ($D20 + 5$) ausgeführt.

△

Mit der Angabe „K30V0“ als Operanden wird eine Anweisung mit dem Wert K35 ausgeführt, wenn V0 den Wert „5“ enthält ($D30 + 5$).

△

- Oktal adressierte Operanden

Bei oktal adressierte Operanden (X, Y, KnX, KnY) wird der Inhalt eines Index-Registers in einen oktalen Wert gewandelt und entsprechend den Rechenregeln zu der Operandenadresse addiert (siehe Abschnitt 3.8.10).

Beispiele ▾

Wird in Z1 der Wert „K8“ übertragen, wird bei der Angabe des Operanden als „X0Z1“ eine Anweisung mit dem Eingang X10 ($X0 + 10_{\text{Oktal}} (8_{\text{Dezimal}} = 10_{\text{Oktal}})$) ausgeführt.

△

Wird in Z1 der dezimale Wert „K10“ gespeichert, wird bei der Angabe des Operanden als „X0Z1“ die Anweisung nicht mit dem Eingang X10, sondern dem Eingang X12 ausgeführt ($10_{\text{Dezimal}} = 12_{\text{Oktal}}$).

△

- Hexadezimale numerische Werte

Bei der indizierten Angabe von hexadezimalen Konstanten (H) wird der Inhalt eines Index-Registers als hexadezimaler Wert interpretiert.

Beispiele ▾

Wird in das Index-Register V5 der dezimale Wert „K30“ übertragen, wird bei der Angabe des Operanden als „H30V5“ eine Anweisung mit dem Wert H4E ausgeführt. (K30 in V5 entspricht H1E, $H30 + H1E = H4E$).

△

Enthält V5 aber den hexadezimalen Wert „H30“, wird bei der Angabe des Operanden als „H30V5“ eine Anweisung mit dem Wert H60 ausgeführt. ($H30 + H30 = H60$).

△

Speichern von 32-Bit-Daten in Index-Registern

Index-Register sind 16-Bit-Register. In 32-Bit-Anweisungen können die Index-Register V (V0 – V7) und Z (Z0 – Z7) kombiniert eingesetzt werden (V0 und Z0, V1 und Z1, V2 und Z2 usw.).

Zn enthält die niederwertigen 16 Bit, Vn enthält die höherwertigen 16 Bit. In einer 32-Bit-Anweisung darf nur der Operand Z angegeben werden. Wird der Operand V angegeben, kann das Programm nicht verarbeitet werden.

Werden Index-Register als 32-Bit-Operanden angegeben, werden beide Register V und Z gleichzeitig angesprochen. Wird dagegen z. B. mit einer 16-Bit-Anweisung nur in Z ein Wert geschrieben und enthält V noch einen numerischer Wert aus einer anderen Operation, können extrem hohe numerische Werte entstehen, die zu Programmfehlern führen. Auch wenn in einer 32-Bit-Anweisung Werte verwendet werden, die nicht den Bereich von 16-Bit-Werten überschreiten, sollten zum Transfer dieser Werte in die Index-Register 32-Bit-Anweisungen (DMOV) verwendet werden.

Beispiel ▾

Schreiben des Wert 300 in die Index-Register V2 und Z2

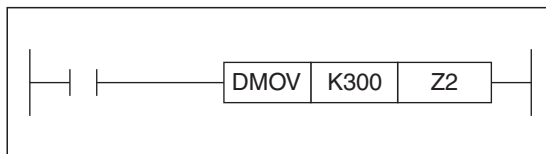


Abb. 3-20:

Mit einer DMOV-Anweisung (32-Bit-Anweisung) wird gleichzeitig in die Index-Register V2 und Z2 geschrieben.

△

Anwendung der Index-Register

Beispiel ▾

Datentransfer mit indizierter Angabe von Datenregistern

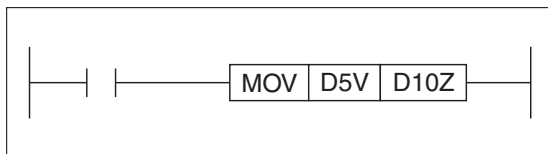


Abb. 3-21:

Programmierbeispiel zum Datentransfer vom Datenregister D5V zum Datenregister D10Z

Berechnung der Ausgangsadresse D5V:

V0 enthält den Wert 8
 $5 + 8 = 13 \rightarrow D13$

Berechnung der Zieladresse D10Z:

Z0 enthält den Wert 14
 $10 + 14 = 24 \rightarrow D24$

Es findet ein Datentransfer vom Datenregister D13 zum Datenregister D24 statt.

△

3.8.7 Einsatz der File-Register

File-Register werden in Blöcken von 500 Adressen im Programmspeicherbereich (EPROM oder EEPROM) der Steuerung abgelegt. Die Anzahl der Blöcke wird über die Parameter festgelegt. Der Zugang zu den File-Registern ist über Programmiergeräte und Bedienterminals möglich.

HINWEISE

Wenn File-Register eingesetzt werden, verringert sich der für das SPS-Programm nutzbare Speicherbereich. Für jeden Block von 500 File-Registern verringert sich die Anzahl der nutzbaren Programmschritte um 500 Schritte. Die Anzahl der File-Register variiert in Abhängigkeit vom Steuerungstyp.

Außer bei der FX1S-Serie ist bei Verwendung der File-Register zu beachten, dass der Speicherbereich mit dem gepufferten Speicherbereich überlappt.

Lesen aus File-Registern

Während des Betriebes der SPS können die Daten der File-Register mittels der BMOV-Anweisung ausgelesen werden.

Schreiben in File-Registern

Bei der FX1S-Serie können die File-Register nur mit Programmiergeräten oder mit PCs und entsprechender Software beschrieben werden.

Bei Steuerungen der FX1N-, der FX2N-, der FX2NC-, der FX3G-, der FX3GC-, der FX3GE-, der FX3S-, der FX3U- und der FX3UC-Serie können die Inhalte der File-Register auch vom SPS-Programm unter Verwendung der BMOV-Anweisung verändert werden.

Für nähere Informationen verwenden Sie bitte die Anleitungen der entsprechenden Programmiersysteme.

Bei der FX1S ist eine Veränderung der Daten im RUN-Modus nicht möglich.

HINWEISE

Eine Veränderung der File-Registerdaten ist im RUN-Modus nur bei RAM-Registern oder File-Registern im internen Speicher möglich.

File-Register, die sich im RAM, im internen Speicher oder auf EEPROM-Speicherkassetten befinden, können im STOP-Modus geändert werden.

File-Register, die sich auf einer EPROM-Speicherkassette befinden, können nicht verändert werden.

3.8.8 Erweiterte Register und erweiterte File-Register

Erweiterte Register (Operandenkennzeichen: R) und erweiterte File-Registern (Operandenkennzeichen: ER) stehen nur bei den Grundgeräten der FX3G-, FX3GC-, FX3GE-, FX3U- und FX3UC-Serie zur Verfügung.

Der Inhalt von erweiterten Registern und erweiterten File-Registern kann mit Hilfe von Anweisungen ausgetauscht werden. Bei einem FX3U- oder FX3UC-Grundgerät muss eine Speicherkassette installiert sein, um die erweiterten File-Register nutzen zu können.

Operand	SPS	Operanden gespeichert im
Erweiterte Register	FX3G/FX3GC/FX3GE	integrierten RAM
	FX3U/FX3UC	integrierten RAM (batteriegepufferter Bereich)
Erweiterte File-Register	FX3G/FX3GE	integrierten EEPROM oder in einer EEPROM-Speicherkassette
	FX3GC/	integrierten EEPROM
	FX3U/FX3UC	Speicherkassette (Flash-Speicher)

Tab. 3-19: Die erweiterten Register und erweiterten File-Register werden in den verschiedenen Steuerungen in verschiedenen Speichern abgelegt

Die folgende Tabelle zeigt, wie auf erweiterte Register und erweiterte File-Register zugegriffen werden kann.

Methode beim Zugriff		Erweiterte Register	Erweiterte File-Register
Lesen im Programm		●	Nur durch Applikationsanweisungen
Schreiben im Programm		●	
Anzeigemodul		●	●
Änderung des Inhalts	Operandentest durch Programmier-Software	●	○
	Schreiben in einen Operandenbereich durch die Programmier-Software	●	●
	per Computer-Link	●	○

Tab. 3-20: Unterschiede beim Zugriff auf erweiterte Register und erweiterte File-Register

●: Funktion ist möglich; ○: Funktion ist nicht möglich

3.8.9 Zahlenwertbereiche von Registern

Werden binär codierte Zahlen in einem Register abgespeichert, ist der Zahlenwertebereich aufgrund der begrenzten Größe eines Registers eingeschränkt.

- Dezimalzahlen
16 Bit: -32 768 bis +32 767 32 Bit: -2 147 483 648 bis +2 147 483 647
- Hexadezimalzahlen
16 Bit: 0 bis FFFF 32 Bit: 0 bis FFFFFFFF

Darstellung negativer Zahlen

Negative Zahlen werden als 2er-Komplement dargestellt.

Bei der Bildung eines 2er-Komplements wird die duale Zahl invertiert (1er-Komplementbildung) und anschließend der binäre Zahlenwert 1 addiert.

Beispiel ▾

0101101 (dual) → +45 (dezimal)
 1010010 (dual) → 1er-Komplement
 1010011 (dual) → 2er-Komplement
 1010011 (dual) → -45 (dezimal)

△

Der im Datenregister abgelegte Wert ist negativ, wenn das höchstwertige Bit (Vorzeichenbit) gesetzt ist (1).

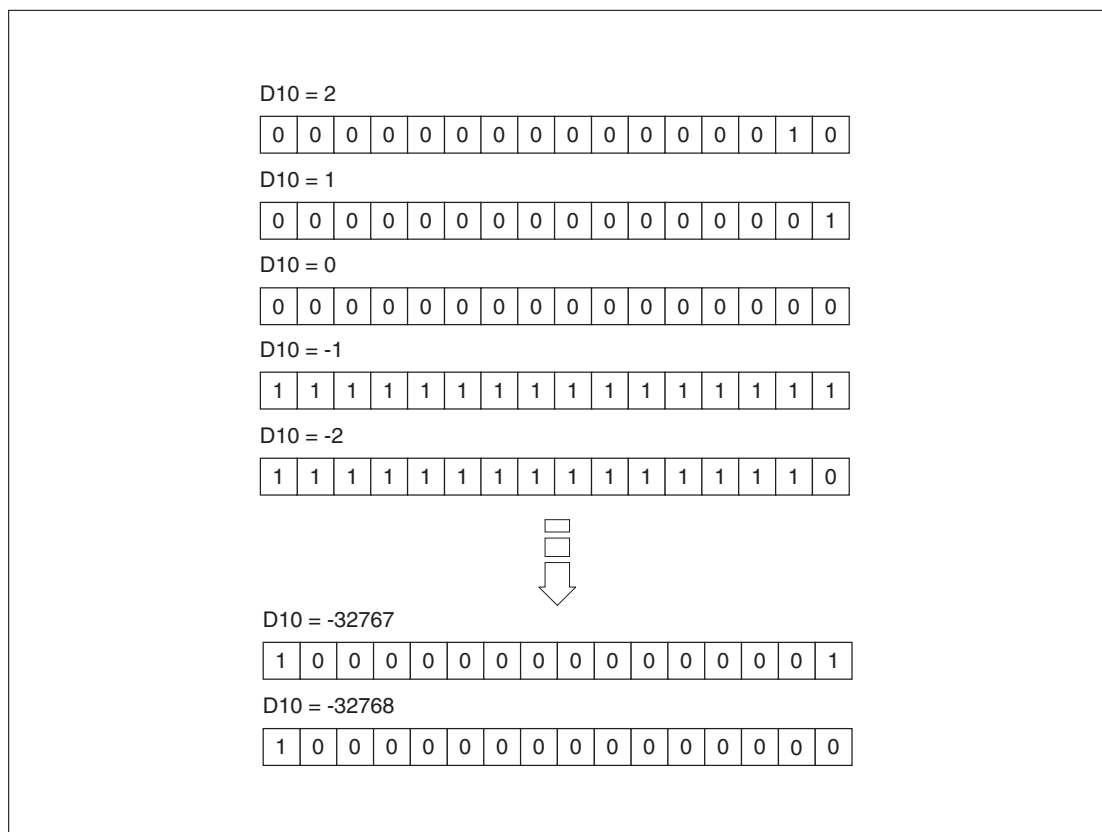


Abb. 3-22: Beispiel für die Darstellung negativer Zahlen

△

3.8.10 Zahlendarstellungen

Die Steuerungen der FX-Familie geben Ihnen die Möglichkeit, mit Zahlenwerten in folgenden Darstellungen zu arbeiten:

- Dezimalzahlen
- Zahlen im wissenschaftlichen Format
- Gleitkommazahlen
- Dualzahlen (Binärzahlen)
- Hexadezimalzahlen
- BCD-Format
- Bit-Muster

Interne Darstellungen der Zahlen in der SPS

Der Mikroprozessor der SPS verarbeitet grundsätzlich nur binäre Informationen. Die kleinste Einheit einer binären Information heißt Bit. Mit einem Bit lassen sich die zwei Signalzustände „0“ und „1“ darstellen.

Alle Zahlenwerte, die nicht in einem binären Format vorliegen, werden deshalb von der SPS in ein binäres Format umgewandelt (codiert).

HINWEIS

Die SPS stellt intern alle Zahlen als 16- oder 32-stellige Dualzahlen oder als Bit-Muster (16-Bit-, 32-Bit-Format) dar.

In den nächsten Abschnitten folgt eine Einführung in die verschiedenen Zahlensysteme und die Umwandlung von Zahlenwerten (Codierungen) zwischen diesen Zahlensystemen.

Dezimalzahlensystem

Basis: 10

Ziffern: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Wertebereich:

- 16-Bit-Format: -32768 bis +32767
- 32-Bit-Format: -2147483648 bis +2147483647

Beispiel ▾

$$351 \text{ (dezimal)} = 3 \times 10^2 + 5 \times 10^1 + 1 \times 10^0$$

△

Zahlen im wissenschaftlichen Format

Dieses Format ist an die wissenschaftliche Darstellung von besonders großen und besonders kleinen Zahlen angelehnt. Die Darstellung erfolgt im 32-Bit-Format mit Fließkomma.

Format: Mantisse $\times 10^{\text{Exponent}}$

Wertebereich:

- Mantisse: ± 1000 bis 9999, oder 0
- Exponent: -41 bis +35

Beispiel ▾

Geschwindigkeit des Lichtes:

- als Dezimalzahl: 299792458 m/s
- im wissenschaftlichen Format: 2998×10^5 m/s

Hierbei ist 2998 die Mantisse und 5 der Exponent. In Datenregistern gespeichert hätte die Zahl zum Beispiel die Form $D120 \times 10^{D121}$.

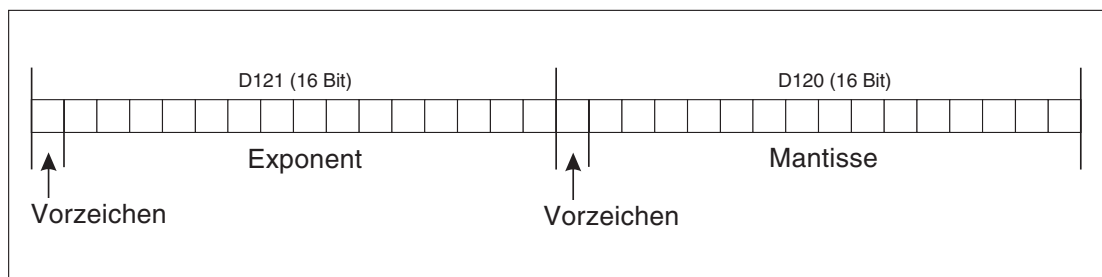


Abb. 3-23: Belegung im Datenregister



Gleitkoma-Zahlensystem

Da Operationen mit Zahlen sehr schnell den zulässigen Wertebereich überschreiten würden, bieten die FX-Serien zusätzlich die Darstellung sehr großer und sehr kleiner Zahlen im Gleitkoma-Format, wie es in Personal- und Mikro-Computern eingesetzt wird.

Das Format des Gleitkoma-Zahlensystems speichert Mantisse und Exponent als Binärzahlen in einem 32-Bit-Doppelwort, wobei die Mantisse 23 Bit und der Exponent 8 Bit belegt.

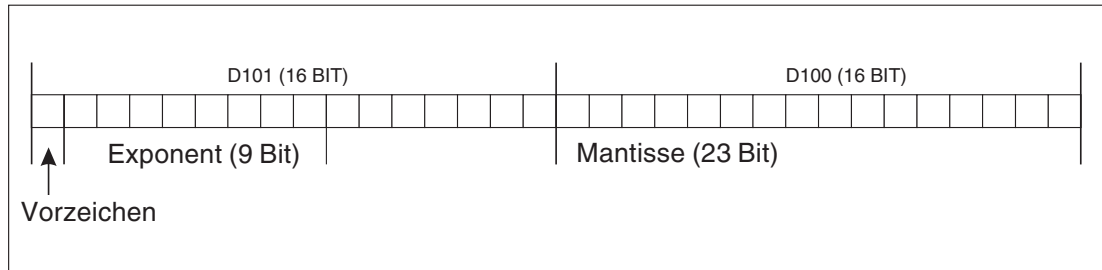


Abb. 3-24: Belegung im Datenregister

Format: $\pm \text{Mantisse} \times 2^{\text{Exponent}}$

Wertebereich:

$$\text{Mantisse: } 1 \times 2^0 + A_{22} \times 2^{-1} + A_{21} \times 2^{-2} + \dots + A_0 \times 2^{-23}$$

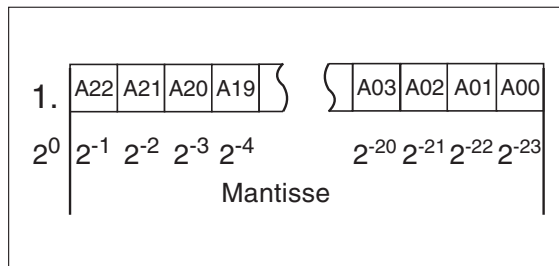


Abb. 3-25:
Mantisse

$$\text{Exponent: } (E_7 \times 2^7 + E_6 \times 2^6 + \dots + E_0 \times 2^0) - 127, \text{ das ergibt } -126 \text{ bis } +127$$

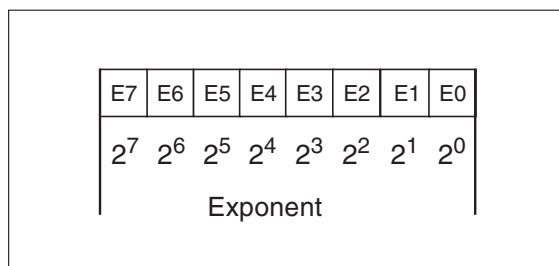


Abb. 3-26:
Exponent

Beispiel ▾ D101 = 16592 = 40D0_{HEX}
 D100 = 0 = 0000_{HEX}

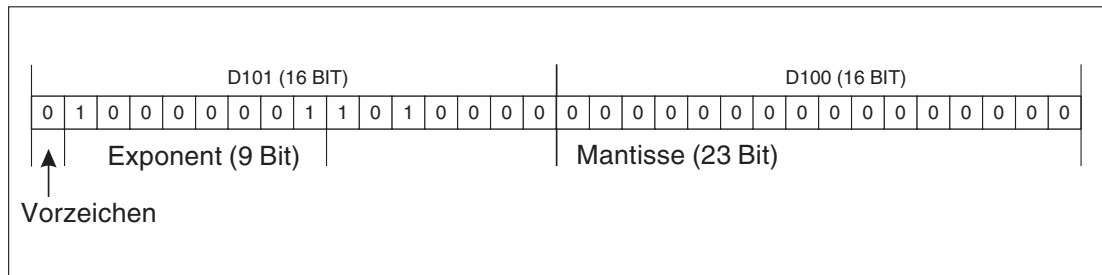


Abb. 3-27: Belegung im Datenregister

Das Vorzeichenbit ist auf 0 gesetzt; positiv.

Der Exponent ist auf 1000001 gesetzt, das entspricht

$$\begin{aligned} & (1 \times 2^7 + 0 \times 2^6 + \dots + 1 \times 2^0) - 127 \\ & = (128 + 0 + \dots + 1) - 127 \\ & = 2 \end{aligned}$$

Die Mantisse ist auf 10100000000000000000000 gesetzt, das entspricht $1,101_{\text{BIN}}$ oder

$$\begin{aligned} & 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + \dots + 0 \times 2^{-23} \\ & = 1,625 \end{aligned}$$

Daraus ergibt sich die Zahl $+ 1,625 \times 2^2 = 6,5$.

△

Dualzahlensystem

Basis: 2

Ziffern: 0, 1

Beispiel ▾ 11001 (dual)
 11001 (dual) = $1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
 11001 (dual) = $16 + 8 + 1$
 1001 (dual) = 25 (dezimal)

△

BIN-Codierung: Dezimalzahl → Dualzahl

Beispiel ▾ 30 (dezimal)
 30 : 2 = 15 Rest 0
 15 : 2 = 7 Rest 1
 7 : 2 = 3 Rest 1
 3 : 2 = 1 Rest 1
 1 : 2 = 0 Rest 1
 30 (dezimal) = 11110 (dual)

△

Codierung: Dualzahl → Dezimalzahl**Beispiel** ▾

111000 (dual)

$$111000 \text{ (dual)} = 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$$

$$111000 \text{ (dual)} = 32 + 6 + 8$$

$$111000 \text{ (dual)} = 56 \text{ (dezimal)}$$

△

Oktalzahlensystem

Basis: 8

Ziffern: 0, 1, 2, 3, 4, 5, 6, 7

Beispiel ▾

245 (oktal)

$$245 \text{ (oktal)} = 2 \times 8^2 + 4 \times 8^1 + 5 \times 8^0$$

$$245 \text{ (oktal)} = 128 + 32 + 5$$

$$245 \text{ (oktal)} = 165 \text{ (dezimal)}$$

△

Codierung: Dezimalzahl → Oktalzahl**Beispiel** ▾

30 (dezimal)

$$30 : 8 = 3 \text{ Rest } 6$$

$$3 : 8 = 0 \text{ Rest } 3$$

$$30 \text{ (dezimal)} = 36 \text{ (oktal)}$$

△

Codierung: Oktalzahl → Dezimalzahl**Beispiel** ▾

374 (oktal)

$$374 \text{ (oktal)} = 3 \times 8^2 + 7 \times 8^1 + 4 \times 8^0$$

$$374 \text{ (oktal)} = 192 + 56 + 4$$

$$374 \text{ (oktal)} = 252 \text{ (dezimal)}$$

△

Hexadezimalzahlensystem

Basis: 16

Ziffern: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

(A=10, B=11, C=12, D=13, E=14, F=15)

Beispiel ▾ 1E (hexadezimal)

$$1E \text{ (hexadezimal)} = 1 \times 16^1 + 14 \times 16^0$$

$$1E \text{ (hexadezimal)} = 16 + 14$$

$$1E \text{ (hexadezimal)} = 30 \text{ (dezimal)}$$

△

Codierung: Dezimalzahl → Hexadezimalzahl

Beispiel ▾ 63 (dezimal)

$$63 : 16 = 3 \text{ Rest } 15 \rightarrow F \text{ (hexadezimal)}$$

$$3 : 16 = 0 \text{ Rest } 3 \rightarrow 3 \text{ (hexadezimal)}$$

$$63 \text{ (dezimal)} = 3F \text{ (hexadezimal)}$$

△

Codierung: Hexadezimalzahl → Dezimalzahl

Beispiel ▾ 7A (hexadezimal)

$$7A \text{ (hexadezimal)} = 7 \times 16^1 + 10 \times 16^0$$

$$7A \text{ (hexadezimal)} = 112 + 10$$

$$7A \text{ (hexadezimal)} = 122 \text{ (dezimal)}$$

△

BCD-Format

Im BCD-Format (**B**inär-**C**odierte-**D**ezimalziffer) wird jede Ziffer einer Dezimalzahl durch eine 4-Bit-Binärzahl dargestellt. Bei einer 4-Bit-Darstellung ist es möglich, die dezimalen Ziffern 0 bis 15 binär zu codieren. Im BCD-Format ist jedoch nur die Codierung der dezimalen Ziffern von 0 bis 9 zulässig.

Codierung: Dezimalzahl → BCD-Format

Beispiel ▾ 67 (dezimal) \leftrightarrow Ziffern: 6, 7

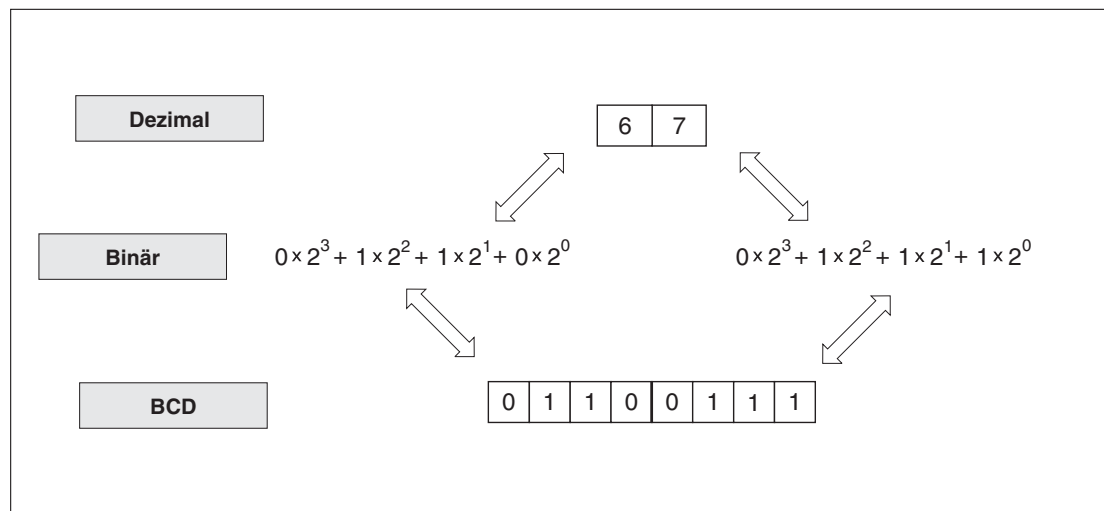


Abb. 3-28: Codierung einer Dezimalzahl in ein BCD-Format und umgekehrt

3.9 Pointer

Pointer werden in Zusammenhang mit der CJ-Sprunganweisung oder der CALL-Anweisung programmiert.

Pointer sind Sprungzieladressen, mit denen das Sprungziel oder das Unterprogramm im Programm markiert werden (Pointer-Markierung).

3.9.1 Pointer adressieren

Bei einer Steuerung der FX1S-Serie stehen die Pointer-Markierungen P0 bis P63 (64 Adressen) zur Verfügung. Bei den Steuerungen der FX1N-, FX2N- und FX2NC-Serie umfasst der Pointerbereich 128 Adressen (P0 bis P127), bei einer FX3G/FC3GC/FX3GE 2048 Adressen (P0 bis P2047) bei einer FX3S 256 Adressen (P0 bis P255) und bei einer FX3U- oder FX3UC-Steuerung 4096 Adressen (P0 bis P4095). In Verbindung mit der Anweisung CJ kann bei allen Steuerungen das Programmende angesprungen werden, wenn die Pointer-Adresse P63 verwendet wird. Die END-Anweisung braucht in diesem Fall keine Pointer-Markierung.

HINWEIS

Die gleiche Pointer-Markierung darf nicht mehrfach in einem SPS-Programm benutzt werden.

3.9.2 Nesting-Ebenen

Während der Ausführung eines Interrupts sind alle anderen Interrupts inaktiv. Um geschachtelte Interrupts zu erhalten, müssen die EI-DI-Anweisungen innerhalb einer Interrupt-Routine programmiert werden. Die Interrupts können in zwei Nesting-Ebenen geschachtelt werden.

3.10 Interrupt-Pointer

Mit Hilfe der Interrupt-Pointer kann ein Sprung innerhalb eines SPS-Programms zu einem Interrupt-Programm ausgeführt werden (siehe auch Abschnitt 6.2.4).

3.10.1 Interrupt-Pointer adressieren

FX1S/FX1N

MELSEC FX1S und FX1N:

Es stehen 6 Interrupt-Pointer zur Verfügung. Die Adressierung eines Interrupt-Pointers müssen Sie wie folgt vornehmen:

Interrupt-Pointer: I ① 0 ②

- ① Adresse 0 bis 3; entspricht Eingang X0 bis X3
- ② 0: = Interrupt bei abfallender Eingangssignalfanke
1: = Interrupt bei ansteigender Eingangssignalfanke

Beispiel ▾

Interrupt-Pointer: I201

Das über den Interrupt-Pointer aufgerufene Interrupt-Programm wird bei ansteigender Signalfanke am Eingang X2 ausgeführt.

Die Rückkehr ins Hauptprogramm erfolgt, nachdem die IRET-Anweisung ausgeführt wurde.



FX1s

Bei einer Steuerung der FX1s-Serie mit den Versionsnummern 1.00 bis 1.30 muss das Programm erweitert werden, wenn Interrupt-Pointer verwendet werden.

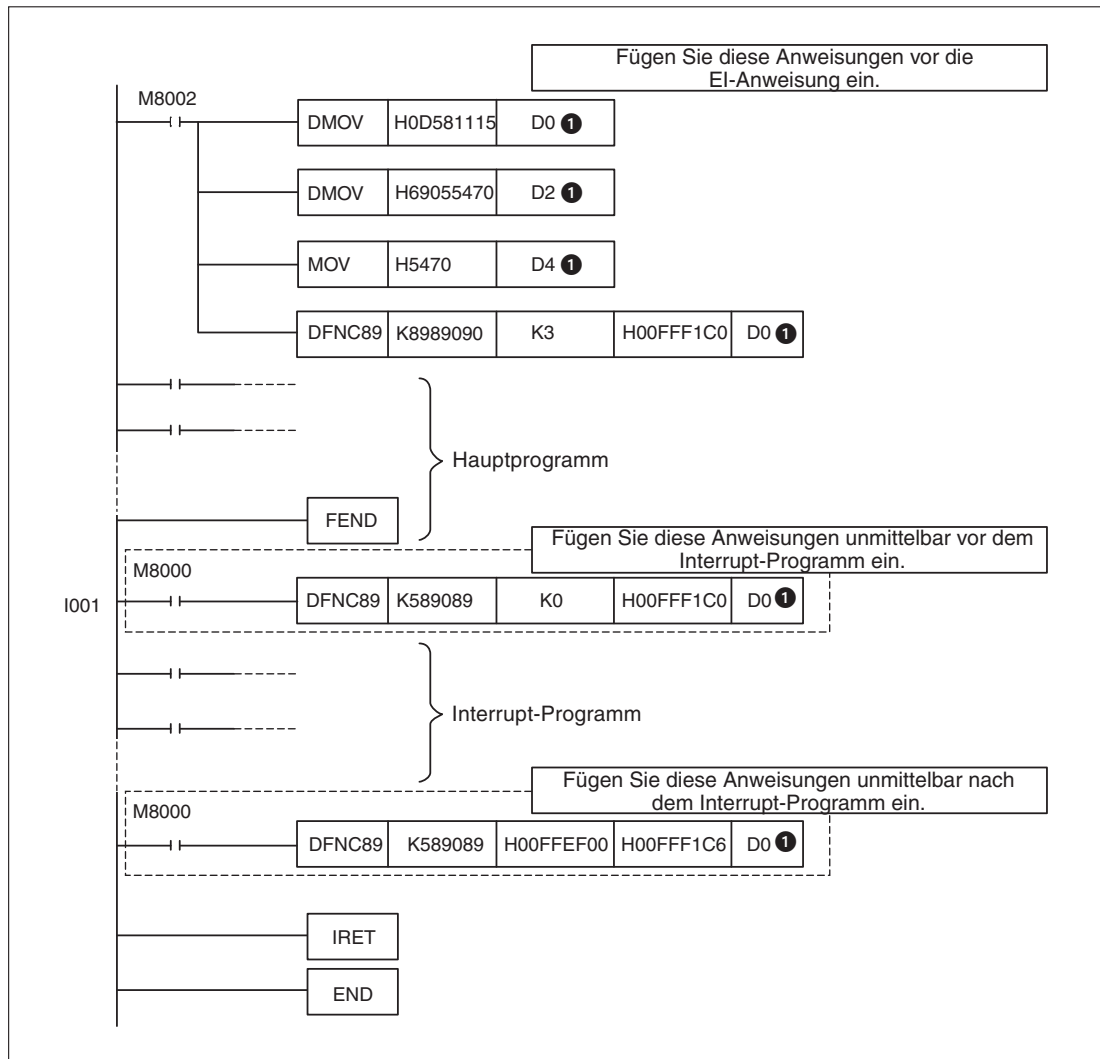


Abb. 3-29: Notwendige Programmiererweiterungen für FX1s (V1.00 bis V1.30) bei Verwendung von Interrupt-Pointern

- ❶ Die Register D0 bis D4 sind durch diese Programmteile belegt und können nicht im SPS-Programm verwendet werden.

HINWEIS

Bei einer FX1s ab der Versionsnummer 1.40 sind diese Programmiererweiterungen nicht notwendig.

FX1N

Bei einer FX1N der Version 1.00 muss bei Verwendung von Interrupt-Pointern das Programm mit den nachfolgend dargestellten Anweisungen erweitert werden.

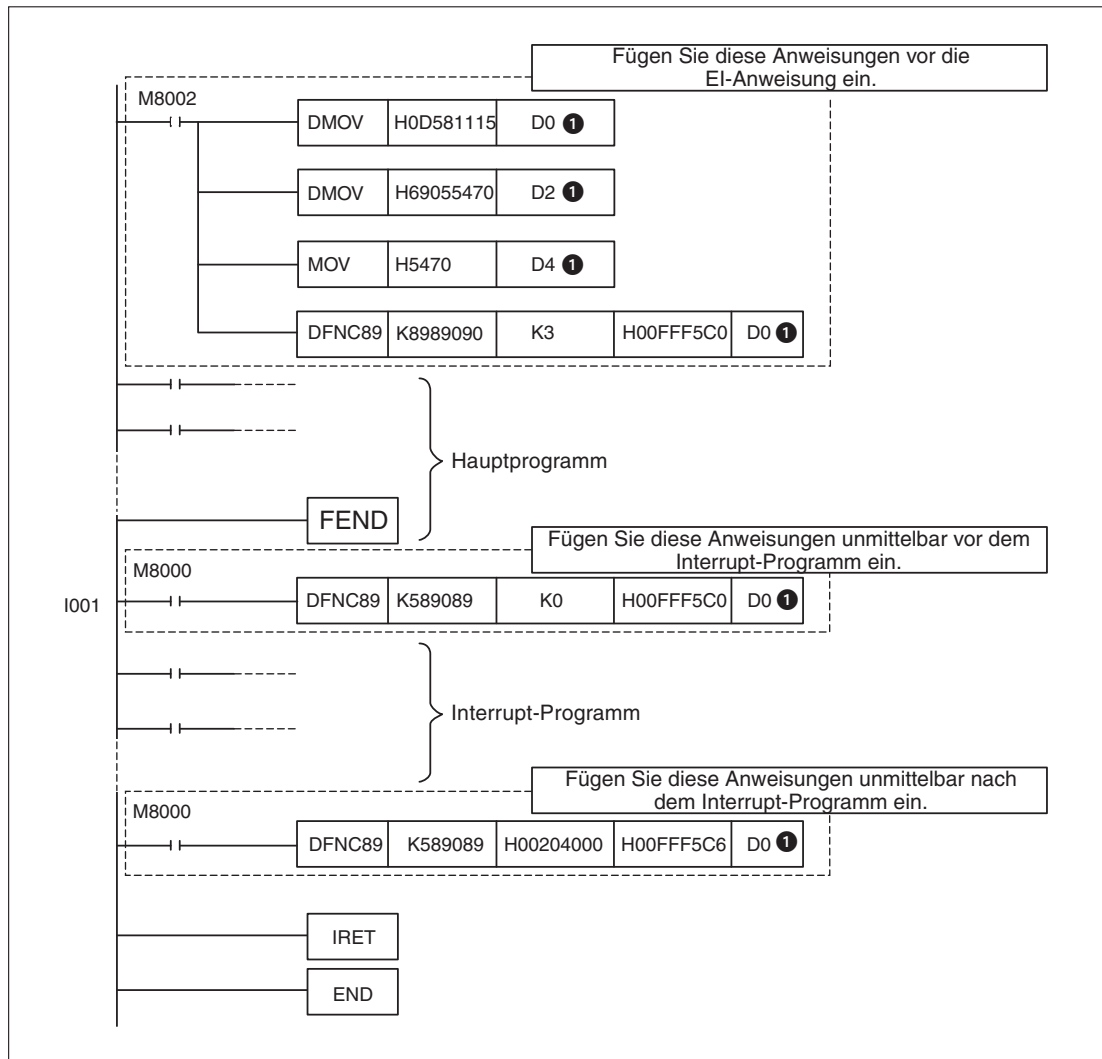


Abb. 3-30: Notwendige Programmerweiterungen für FX1N (V1.00 bis V1.30) bei Verwendung von Interrupt-Pointern

- ① Die Register D0 bis D4 sind durch diese Programmteile belegt und können nicht im SPS-Programm verwendet werden.

HINWEIS

Bei einer FX1N ab der Versionsnummer 1.10 sind diese Programmerweiterungen nicht notwendig.

FX2□/FX3□**MELSEC FX2N, FX2NC, FX3G, FX3GC, FX3GE, FX3S, FX3U und FX3UC**

Bei einer Steuerung der FX2□- oder FX3□-Serie stehen bis zu 15 Interrupt-Pointer zur Verfügung. Bei der Adressierung eines Interrupt-Pointers werden drei Gruppen unterschieden:

- Eingangs-Interrupts (6 Interrupt-Pointer)

Diese Interrupts werden durch den Signalwechsel an einem Eingang der Steuerung ausgelöst.

Angabeformat eines Eingangs-Interrupt-Pointers: I ① 0 ②

- ① Adresse 0 bis 5
Jede Adresse darf nur einmal verwendet werden.
- ② 0: Interrupt bei abfallender Flanke
1: Interrupt bei ansteigender Flanke

Beispiel ▾

Interrupt-Pointer: I001

Das über den Interrupt-Pointer aufgerufene Interrupt-Programm wird bei ansteigender Flanke von X0 ausgeführt.

Die Rückkehr ins Hauptprogramm erfolgt, nachdem die IRET-Anweisung ausgeführt wurde.



- Timer-Interrupts (3 Interrupt-Pointer)

Ein Interrupt wird in festen Intervallen (1 bis 99 ms) ausgelöst.

Angabe eines Pointers für einen Timer-Interrupt: I ① ②

- ① Adresse 6 bis 8
Jede Adresse darf nur einmal verwendet werden.
- ② 10 bis 99 ms

Beispiel ▾

Interrupt-Pointer: I610

Das über den Interrupt-Pointer I610 aufgerufene Interrupt-Programm wird in Intervallen von 10 ms ausgeführt.

Die Rückkehr ins Hauptprogramm erfolgt, nachdem die IRET-Anweisung ausgeführt wurde.

**HINWEIS**

Interrupt-Pointer werden nach einer FEND-Anweisung programmiert. Dabei dürfen nicht mehr als 9 Interrupt-Pointer eingesetzt werden. Mehr als zwei Verzweigungsebenen sind nicht zulässig.

- Counter-Interrupts (6 Interrupt-Pointer), nicht bei FX3G, FX3GC, FX3GE und FX3S

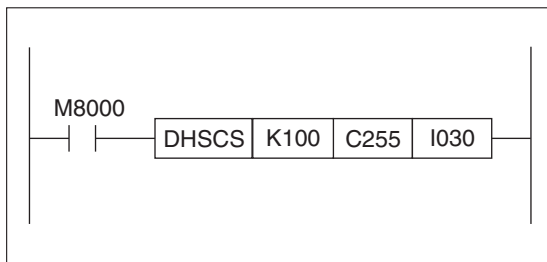
Bei Übereinstimmung eines Counter-Zählwerts mit dem in einer Vergleichsanweisung angegebenen Vergleichswert wird ein Interrupt ausgelöst.

Angabeformat für einen Counter-Interrupt-Pointer: I 0 ① 0

- ① Adresse 1 bis 6
Counter-Interrupts können als Operanden zum Setzen (HSCS, FNC 53) oder Rücksetzen (HSCR, FNC 54) durch High-Speed-Counter verwendet werden. Zum Ausschalten des Counter-Interrupts schalten Sie den Sondermerker M8059 ein.

Beispiel ▾

Interrupt-Pointer: I030

**Abb. 3-31:**

Programmierbeispiel zum Einsatz eines Counter-Interrupts

Das über den Interrupt-Pointer I030 aufgerufene Interrupt-Programm wird ausgeführt, sobald der Wert des High-Speed-Counters C255 den in K100 angegebenen Wert erreicht. △

HINWEIS

Bitte beachten Sie Abschnitt 6.7.4 mit den näheren Informationen zum Einsatz der Befehle zum Setzen oder Zurücksetzen durch High-Speed-Counter.

Ausschalten beliebiger Interrupts

Sie können beliebige Interrupts durch Einschalten der zugehörigen Sondermerker zeitweilig oder permanent Ausschalten. Die entsprechenden Sondermerker sind in Kapitel 6 angegeben. Bei allen Steuerungen schaltet der erste Sondermerker M8050 den Interrupt I0①② aus.

HINWEISE

Setzen Sie nie einen Sondermerker, ohne sich seiner Funktion sicher zu sein. Nicht alle Steuerungen arbeiten immer mit den gleichen Sondermerkern.

High-Speed-Counter-Interrupts können immer nur als einzelne Gruppe mit dem Sondermerker M8059 ausgeschaltet werden.

3.11 Nesting

Mit Hilfe von Nesting-Operanden können Verzweigungsebenen innerhalb eines Programms realisiert werden. Nesting-Operanden werden in Zusammenhang mit den MC- und MCR-Anweisungen eingesetzt.

Der genaue Einsatz der Nesting-Operanden wird in Abschnitt 4.9 in der Beschreibung zu den Kontrollbedingungen (MC, MCR) beschrieben.

3.11.1 Nesting-Operanden adressieren

Es stehen acht Nesting-Operanden N0 bis N7 zur Verfügung.

3.12 Pufferspeicher eines Sondermoduls

Bei den SPS-Grundgeräten der FX3U- und FX3UC-Serie kann auf den Pufferspeicher eines Sondermoduls statt mit TO- und FROM-Anweisungen auch direkt, z. B. mit einer MOV-Anweisung, zugegriffen werden.

Angabe der Operandenadresse:



Die Sondermoduladresse (U) kann im Bereich von 0 bis 7 liegen. Als Pufferspeicheradresse (\G) können Werte von 0 bis 32766 angegeben werden. Eine ausführliche Beschreibung der Adressierung von Sondermodulen und deren Pufferspeicher finden Sie im Abschnitt 7.3.

Bei der Operandenadresse U1\G11 zum Beispiel wird im Sondermodul mit der Adresse 1 die Pufferspeicheradresse 11 angesprochen.

Beispiel ▾

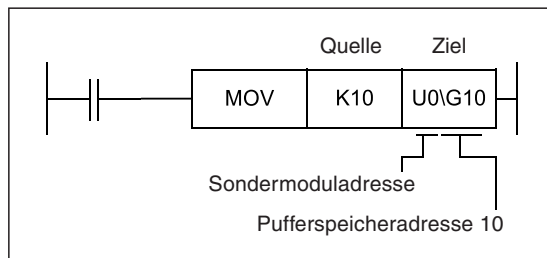


Abb. 3-32:

In diesem Beispiel wird der Wert „10“ in die Pufferspeicheradresse 10 des Sondermoduls mit der Adresse 0 übertragen.

△

Die Pufferspeicheradresse kann auch über Index-Register angegeben werden.

Beispiel ▾

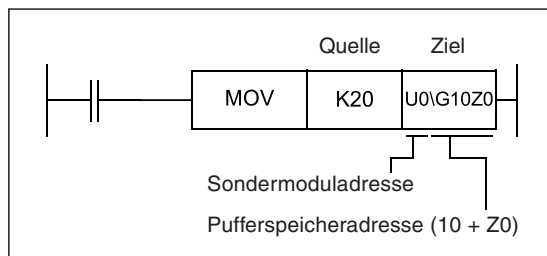


Abb. 3-33:

Im links abgebildeten Beispiel wird der Wert „20“ in die Pufferspeicheradresse des Sondermoduls mit der Adresse 0 übertragen, die sich aus der Addition des Inhalts des Index-Registers Z0 mit dem festen Wert „10“ ergibt.

△

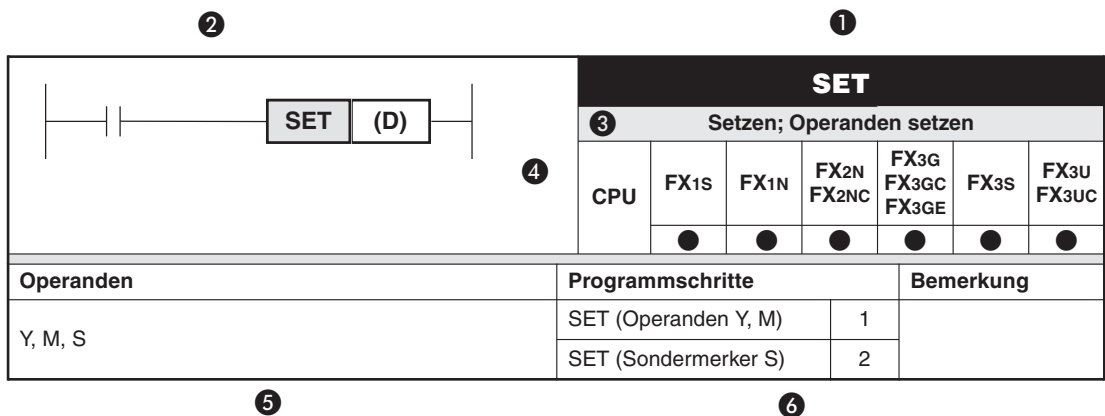
4 Grundbefehlssatz

4.1 Allgemeine Hinweise

Dieses Kapitel beschreibt den Grundbefehlssatz der FX-Familie. Mit diesen Anweisungen aus lassen sich alle logischen Grundverknüpfungen programmieren. Die Anweisungen aus dem Grundbefehlssatz können jeweils nur eine Operandenadresse ansprechen.

4.1.1 Erläuterung der Grundbefehlssatztabellen

Alle Grundbefehle sind in einer tabellarischen Form auf den nächsten Seiten aufgelistet. Dieser Abschnitt erläutert kurz die Struktur der Übersichtstabellen.



1 Anweisung

An dieser Stelle wird der Name der Anweisung angegeben, der bei der Programmierung in Anweisungsliste eingesetzt wird.

2 Kontaktplansymbol

Das Kontaktplansymbol wird bei der Kontaktplanprogrammierung verwendet. Das Kontaktplansymbol besteht aus der Anweisung und den einsetzbaren Operanden.

3 Bedeutung

Hier finden Sie eine kurze Beschreibung zur Bedeutung der Anweisung.

4 CPU

Die MELSEC FX-Serie, bei der diese Anweisung angewendet werden kann, ist durch einen Punkt (●) gekennzeichnet.

5 Operanden

In diesem Feld werden die in Zusammenhang mit der Anweisung einsetzbaren Operanden angegeben.

6 Programmschritte

Es wird die Anzahl von Programmschritten angegeben, die bis zur vollständigen Ausführung der Anweisung erforderlich sind. Bei einer FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC ist die Anzahl der Programmschritte teilweise von den verwendeten Operanden abhängig. Eine Übersicht der Programmschritte für diese Steuerungen enthält Abschnitt 4.1.2.

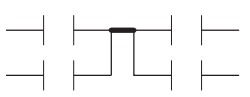
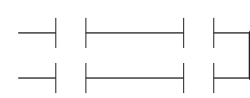
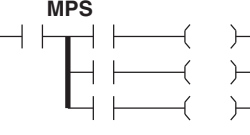
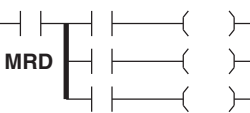
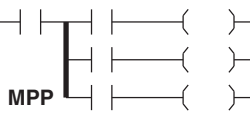
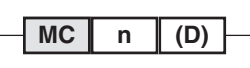



Übersicht der Grundbefehle

Anweisung	Kontaktplansymbol	Bedeutung	Operanden	Programmschritte ^①	Referenz
LD		LADE; Beginn einer Verknüpfung mit Abfrage auf Signalzustand „1“	X, Y, M, S, T, C, D□.b ^②	1	Abschnitt 4.2
LDI		LADE NICHT; Beginn einer Verknüpfung mit Abfrage auf Signalzustand „0“			
OUT		AUSGABE; Ausgabe, Zuweisung eines Verknüpfungsergebnisses	Y, M, S, T, C, D□.b ^②	Y, M: 1 S, Sondermerker: 2 T: 3, C: 3 C (32 Bit): 5	Abschnitt 4.3
AND		UND; UND-Verknüpfung mit Abfrage auf Signalzustand „1“	X, Y, M, S, T, C, D□.b ^②	1	Abschnitt 4.4
ANI		UND Nicht; UND-Verknüpfung mit Abfrage auf Signalzustand „0“			
OR		ODER; ODER-Verknüpfung mit Abfrage auf Signalzustand „1“	X, Y, M, S, T, C, D□.b ^②	1	Abschnitt 4.5
ORI		ODER Nicht; ODER-Verknüpfung mit Abfrage auf Signalzustand „0“			
LDP		LADE; (gepulst) Beginn einer Verknüpfung mit Abfrage der ansteigenden Flanke;	X, Y, M, S, T, C, D□.b ^②	2	Abschnitt 4.6
LDF		LADE; (gepulst) Beginn einer Verknüpfung mit Abfrage der abfallenden Flanke			
ANDP		UND; (gepulst) UND-Verknüpfung mit Abfrage der ansteigenden Flanke	X, Y, M, S, T, C, D□.b ^②	2	Abschnitt 4.7
ANDF		UND; (gepulst) UND-Verknüpfung mit Abfrage der abfallenden Flanke			
ORP		ODER; ODER-Verknüpfung mit Abfrage der ansteigenden Flanke	X, Y, M, S, T, C, D□.b ^②	2	Abschnitt 4.8
ORF		ODER; ODER-Verknüpfungen mit Abfrage der abfallenden Flanke			

Tab. 4-1: Grundbefehlsübersicht (Teil 1)







① Die Anzahl der Programmschritte bei einer FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC ist im Abschnitt 4.1.2 angegeben.

② Nur bei FX3U und FX3UC

Anweisung	Kontaktplansymbol	Bedeutung	Operanden	Programmschritte ^①	Referenz
ANB		UND-Block; Koppelbefehl: Reihenschaltung von Parallelverknüpfungen	—	1	Abschnitt 4.9
ORB		ODER-Block; Koppelbefehl: Parallelschaltung von Reihenverknüpfungen	—	1	Abschnitt 4.10
MPS		Push Down Stack; Abspeichern eines Verknüpfungsergebnisses	—	1	Abschnitt 4.11
MRD		Read Down Stack; Lesen eines Verknüpfungsergebnisses	—		
MPP		Pop Up Stack; Lesen und Löschen des Verknüpfungsspeichers	—		
MC		Master Control; Setzen einer Kontrollbedingung	Y, M, keine Sondermerker	3	Abschnitt 4.12
MCR		Master Control Reset; Rücksetzen einer Kontrollbedingung	N	2	
SET		Setzen; Operanden setzen	Y, M, S, D□.b ^②	Y, M: 1 S, Sondermerker: 2	Abschnitt 4.13
RST		Rücksetzen; Operanden zurücksetzen	Y, M, S, D, V, Z, T, C, D□.b ^②	Y, M: 1 D, V, Z, Sondermerker: 3 T, C: 2	

Tab. 4-2: Grundbefehlsübersicht (Teil 2)

- ① Die Anzahl der Programmschritte bei einer FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC ist im Abschnitt 4.1.2 angegeben.
- ② Nur bei FX3U und FX3UC

Anweisung	Kontaktplansymbol	Bedeutung	Operanden	Programmschritte ^①	Referenz
PLS		Impulserzeugung; Erzeugen eines einmaligen Impulses bei ansteigender Flanke	Y, M	2	Abschnitt 4.14
PLF		Impulserzeugung; Erzeugen eines einmaligen Impulses bei abfallender Flanke			
INV		Inversion; Umkehrung von Verarbeitungsergebnissen	—	1	Abschnitt 4.15
MEP		Impulserzeugung; Erzeugen eines einmaligen Impulses, wenn das Verknüpfungsergebnis „1“ wird	—	1	Abschnitt 4.16
MEF		Impulserzeugung; Erzeugen eines einmaligen Impulses, wenn das Verknüpfungsergebnis „0“ wird	—	1	
NOP	—	Leerzeile; Leerzeile ohne Funktion	—	1	Abschnitt 4.17
END		Ende; SPS-Programmende	—	1	Abschnitt 4.18

Tab. 4-3: Grundbefehlsübersicht (Teil 3)

① Die Anzahl der Programmschritte bei einer FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC ist im Abschnitt 4.1.2 angegeben.

4.1.2 Anzahl der Programmschritte bei einer FX3G, FX3GC, FX3GE, FX3S, FX3U oder FX3UC

Bei den Steuerungen der FX3G-, FX3GC-, FX3GE-, FX3S-, FX3U- und FX3UC-Serie hängt die Anzahl der Programmschritte, die für die Ausführung von bestimmten Grundanweisungen benötigt werden, von den in der Anweisung verwendeten Operanden ab.

Bei den Grundanweisungen, die in der folgenden Tabelle nicht aufgeführt sind (ORB, ANB, MPS, MRD, INV, MEP, MEF, NOP und END) gelten die bei den jeweiligen Anweisungen angegebenen Programmschritte.

Operand		Anweisung						
		LD, LDI, AND, ANI, OR, ORI	OUT	SET	RST	PLS, PLF	LDP, LDF, ANDP, ANDF, ORP, ORF	MC
Bit-Operanden	X000–X357	1	—	—	—	—	2	—
	Y000–Y357	1	1	1	1	2	2	3
	M0–M1535	1	1	1	1	2	2	3
	M1536–M3583	2	2	2	2	2	2	3
	M3584–M7679	3	3	3	3	3	3	4
	S0–S1023	1	2	2	2	—	2	—
	S1024–S4095	2	2	2	2	—	2	—
	T0–T191, T200–T245	1	3	—	2	—	2	—
	T192–T199, T246–T511	1	3	—	2	—	2	—
	C0–C199	1	3	—	2	—	2	—
	C200–C255	1	5	—	2	—	2	—
	Sondermerker M8000–M8255	1	2	2	2	—	2	—
	Sondermerker M8256–M8511	2	2	2	2	—	2	—
Durch Index-Ver-gabe adres-sierte Bit-Operanden	X000–X357	3	—	—	—	—	—	—
	Y000–Y357	3	3	3	3	3	—	—
	M0–M7679	3	3	3	3	3	—	—
	S0–S4095	—	—	—	—	—	—	—
	T0–T511	3	4	—	—	—	—	—
	C0–C199	3	4	—	3	—	—	—
	C200–C255	—	—	—	—	—	—	—
Sondermerker M8000–M8511	3	3	3	3	—	—	—	
Wort-Operanden	D0–D7999, Sonderregister D8000–D851	—	—	—	3	—	—	—
	R0–R32767	—	—	—	—	—	—	—
Durch Index-Ver-gabe adres-sierte Wort-Operanden	D0–D7999, Sonderregister D8000–D851	—	—	—	—	—	—	—
	R0–R32767	—	—	—	—	—	—	—
Bits in Wort-Operanden	D□.b	3	3	3	3	—	3	—
	Sonderregister SD□.b							

Tab. 4-4: Programmschritte bei Steuerungen der FX3G□-, FX3S-, FX3U- und FX3UC-Serie

4.2 Beginn von Verknüpfungen (LD, LDI)

	LD						
	LADE; Beginn einer Verknüpfung mit Abfrage auf Signalzustand „1“						
	LDI						
	LADE NICHT; Beginn einer Verknüpfung mit Abfrage auf Signalzustand „0“						
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
	●	●	●	●	●	●	
Operanden	Programmschritte			Bemerkung			
X, Y, M, S, T, C, nur bei FX3U/FX3UC: D□.b	LD			1*			
	LDI			1*			

* Die Anzahl der Programmschritte bei einer FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC ist im Abschnitt 4.1.2 angegeben.

Funktion

Programmieren eines Verknüpfungsbeginns

Beschreibung

- Der Beginn einer Verknüpfung wird mit einer LD- oder LDI-Anweisung programmiert.
- Die Programmierung eines Strompfades beginnt immer mit einer LD- oder LDI-Anweisung.
- Die LD- und LDI-Anweisung wird auch im Zusammenhang mit der ANB- und ORB-Anweisung zum Starten einer Verzweigung verwendet (siehe auch Abschnitte 4.9 und 4.10).

Beispiel ▾ Einsatz der Anweisungen LD, LDI

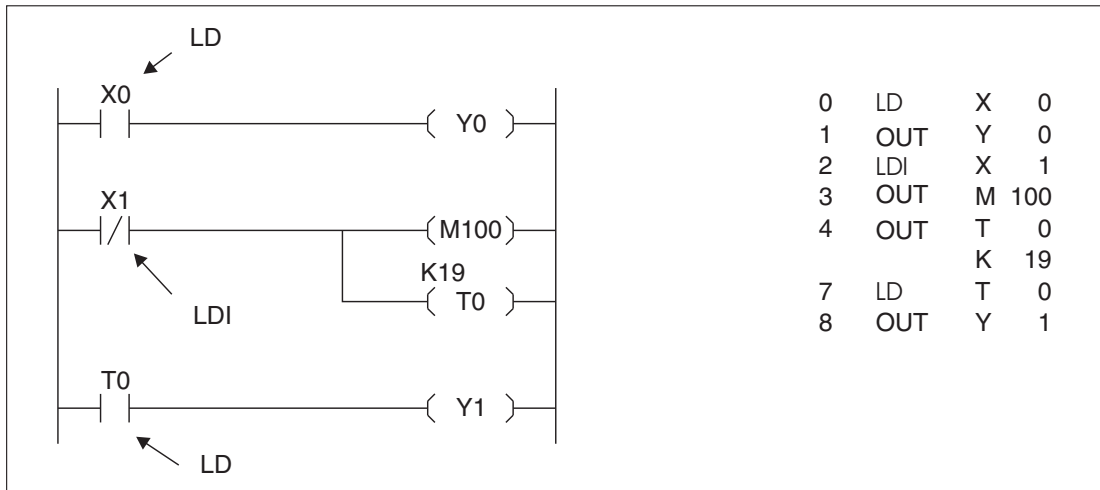


Abb. 4-1: Programmierbeispiel zum Einsatz der Anweisungen LD, LDI

Der Eingang X0 wird auf den Signalzustand „1“ abgefragt. Der Ausgang Y0 wird auf den Signalzustand „1“ geschaltet, sobald der Eingang X0 ein „1“-Signal erhält.

Der Eingang X1 wird auf den Signalzustand „0“ abgefragt. Der Merker M100 erhält den Signalzustand „1“, und die eingestellte Zeit des Timers T0 beginnt abzulaufen, sobald der Eingang X1 ein „0“-Signal erhält.

Nach Ablauf des eingestellten Zeitsollwertes (19 x 100 ms = 1,9 s) schaltet der Timer T0 den Ausgang Y1 auf den Signalzustand „1“.

△

4.3 Ausgabe eines Verknüpfungsergebnisses (OUT)

		OUT					
		AUSGABE; Ausgabe, Zuweisung eines Verknüpfungsergebnisses					
CPU		FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
		●	●	●	●	●	●
Operanden		Programmschritte				Bemerkung	
Y, M, S, T, C, nur bei FX3U/FX3UC: D□.b		Y, M	1	T, C (16 Bit)	3*		
		S,	2	C (32 Bit)	5*		

* Die Anzahl der Programmschritte bei einer FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC ist im Abschnitt 4.1.2 angegeben.

Funktion

Zuweisung eines Signalzustandes in Abhängigkeit vom Verknüpfungsergebnis

Beschreibung

- Mit der OUT-Anweisung kann die Programmierung eines Strompfades abgeschlossen werden.
- Die Programmierung mehrerer OUT-Anweisungen als Ergebnis einer Verknüpfung ist möglich.
- Das durch die OUT-Anweisung dargestellte Verknüpfungsergebnis kann in den nachfolgenden Programmschritten als Eingangssignalzustand eingesetzt werden.
- Das Verknüpfungsergebnis, das durch die OUT-Anweisung dargestellt wird, ist nur solange aktiv, wie die Einschaltbedingung erfüllt ist.

Beispiel ▾

Einsatz der Anweisung OUT

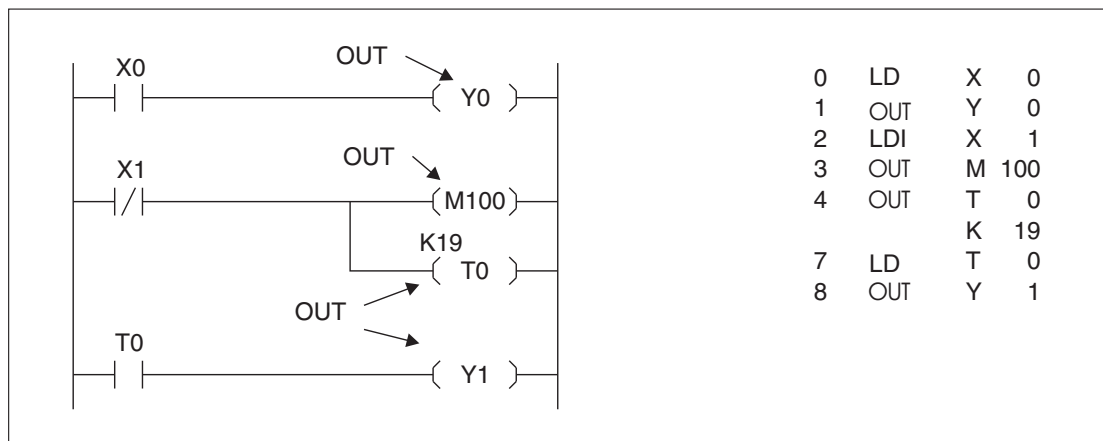


Abb. 4-2: Programmierbeispiel zum Einsatz der Anweisung OUT

Der Eingang X0 wird auf den Signalzustand „1“ abgefragt. Der Ausgang Y0 wird auf den Signalzustand „1“ geschaltet, sobald der Eingang X0 ein „1“-Signal erhält.

Der Eingang X1 wird auf den Signalzustand „0“ abgefragt. Der Merker M100 und der Timer T0 werden auf den Signalzustand „1“ geschaltet, sobald der Eingang X1 ein „0“-Signal erhält.

Nach Ablauf des eingestellten Zeitsollwertes (19 x 100 ms = 1,9 s) schaltet der Timer T0 den Ausgang Y1 auf den Signalzustand „1“.

Anzahl der Programmschritte beim Einsatz von Timern und Countern

OUT-Anweisungen, die sich auf Timer oder Counter beziehen, werden in mehreren Schritten ausgeführt. Im zweiten Programmschritt erfolgt die Einstellung des Zeit- bzw. des Zählerwertes. Dies geschieht durch die Eingabe einer dezimalen Konstante K.

Die Programmierung von Timern und Countern ist in den Abschnitten 3.4 und 3.5 ausführlich beschrieben.

Doppelbelegung von Ausgängen

Bei der Programmierung von doppelten Ausgangsbelegungen kann es zu Problemen beim Programmablauf kommen. Das folgende Beispiel verdeutlicht diese Problematik.

Beispiel ▾

Doppelbelegung eines Ausgangs

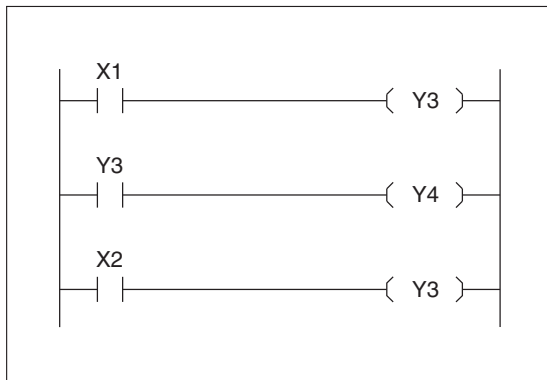


Abb. 4-3:
Programmierbeispiel zur Doppelbelegung eines Ausgangs

Es wird davon ausgegangen, dass der Eingang X1 eingeschaltet (Signal „1“) und der Eingang X2 ausgeschaltet (Signal „0“) ist.

Der erste Ausgang Y3 wird durch den eingeschalteten Eingang X1 aktiviert. Im Prozessabbild der Ausgänge ist Y3 eingeschaltet. Entsprechend wird auch der Ausgang Y4 aktiviert.

Im nächsten Schritt wird der Ausgang Y3 wieder deaktiviert, da der Eingang X2 ausgeschaltet ist. Im Prozessabbild der Ausgänge ist Y3 ausgeschaltet.

Diese Programmsequenz hat zur Folge, dass Y3 ausgeschaltet und Y4 eingeschaltet ist.

In Abschnitt 2.1 wird die Abarbeitung eines SPS-Programms detailliert beschrieben. △

HINWEIS

Vermeiden Sie eine Doppelbelegung von Ausgängen, da dies zu Programmablaufstörungen führen kann.

Beispiel ▾

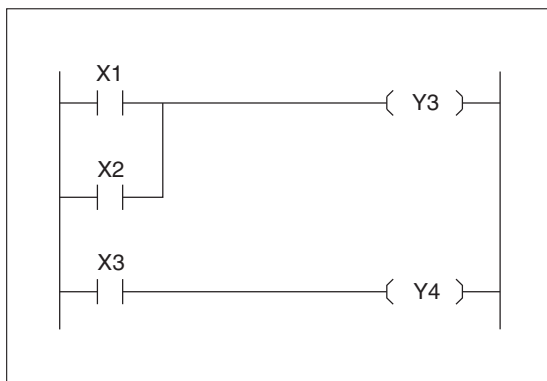




Abb. 4-4:
Programmierbeispiel

△

4.4 UND-Verknüpfungen (AND, ANI)

	AND						
	UND; UND-Verknüpfung mit Abfrage auf Signalzustand „1“						
	ANI						
	UND NICHT; UND-Verknüpfung mit Abfrage auf Signalzustand „0“						
	CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
		●	●	●	●	●	●
	CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
		●	●	●	●	●	●
Operanden	Programmschritte				Bemerkung		
X, Y, M, S, T, C, nur bei FX3U/FX3UC: D□.b	AND-Anweisung				1*		
	ANI-Anweisung				1*		

* Die Anzahl der Programmschritte bei einer FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC ist im Abschnitt 4.1.2 angegeben.

Funktion

Programmieren von logischen UND-Verknüpfungen

Beschreibung

- Die Anweisungen AND und ANI werden für die Reihenschaltung von Kontakten (logische UND-Verknüpfung) eingesetzt.
- Beide Anweisungen stellen logische Verknüpfungen dar und dürfen daher nicht am Anfang eines Strompfades programmiert werden. Der Beginn einer Verknüpfung wird mit einer LD- oder LDI-Anweisung programmiert (siehe Abschnitt 4.2).
- Wenn Sie mehrere aufeinanderfolgende Blockschaltungen in Reihe schalten möchten, können Sie auch die ANB-Anweisung verwenden (siehe Abschnitt 4.9).

HINWEIS

Es dürfen nur 10 Kontakte pro Strompfad und maximal 24 Strompfade pro Spule programmiert werden.

Beispiel ▾ Einsatz der Anweisungen AND, ANI

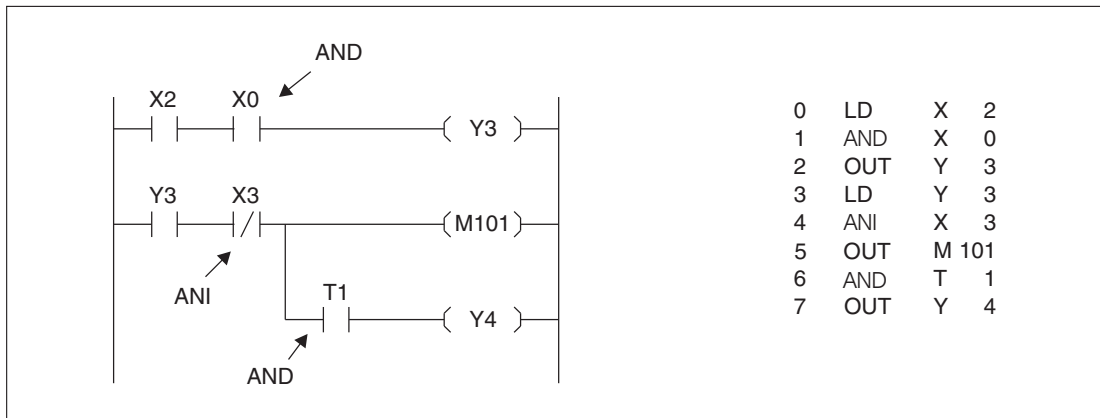


Abb. 4-5: Programmierbeispiel zum Einsatz der Anweisungen AND und ANI

Der Ausgang Y3 weist den Signalzustand „1“ auf, wenn folgende Bedingungen erfüllt sind:

- der Eingang X2 weist den Signalzustand „1“ auf
UND
- der Eingang X0 weist den Signalzustand „1“ auf.

Der Merker M101 weist den Signalzustand „1“ auf, wenn folgende Bedingungen erfüllt sind:



- der Ausgang Y3 weist den Signalzustand „1“ auf,
UND
- der Eingang X3 weist den Signalzustand „0“ auf.

Der Ausgang Y4 weist den Signalzustand „1“ auf, wenn folgende Bedingungen erfüllt sind:

- der Ausgang Y3 weist den Signalzustand „1“ auf,
UND
- der Eingang X3 weist den Signalzustand „0“ auf,
UND
- der Timer-Kontakt T1 weist den Signalzustand „1“ auf.

△

4.5 ODER-Verknüpfungen (OR, ORI)

		OR					
		ODER; ODER-Verknüpfung mit Abfrage auf Signalzustand „1“					
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
	●	●	●	●	●	●	
		ORI					
		ODER NICHT; ODER-Verknüpfung mit Abfrage auf Signalzustand „0“					
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
	●	●	●	●	●	●	
Operanden		Programmschritte			Bemerkung		
X, Y, M, S, T, C, nur bei FX3U/FX3UC: D□.b		OR-Anweisung			1*		
		ORI-Anweisung			1*		

* Die Anzahl der Programmschritte bei einer FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC ist im Abschnitt 4.1.2 angegeben.

Funktion

Programmieren von logischen ODER-Verknüpfungen

Beschreibung

- Die Anweisungen OR und ORI werden für die Parallelschaltung von Kontakten (logische ODER-Verknüpfungen) eingesetzt.
- Beide Anweisungen stellen logische Verknüpfungen dar und dürfen daher nicht am Anfang eines Strompfades programmiert werden. Der Beginn einer Verknüpfung wird mit einer LD- oder LDI-Anweisung programmiert (siehe Abschnitt 4.2)
- Wenn Sie mehrere aufeinanderfolgende Blockschaltungen parallel schalten möchten, können Sie auch die ORB-Anweisung verwenden (siehe Abschnitt 4.10)

HINWEIS

| Sie sollten nicht mehr als 24 Strompfade als Parallelschaltung programmieren.

Beispiel ▾

Einsatz der Anweisungen OR, ORI

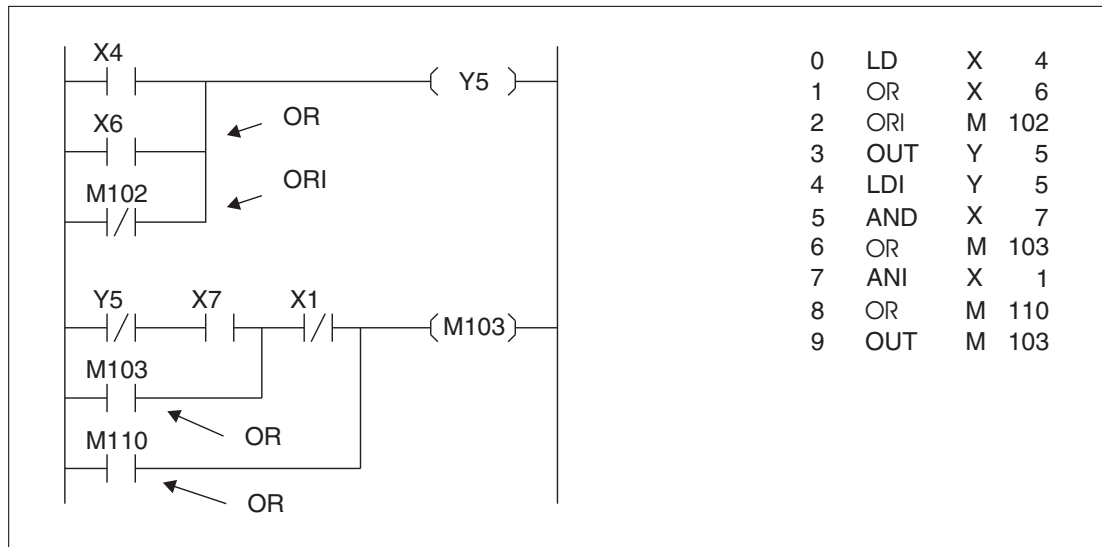


Abb. 4-6: Programmierbeispiel zum Einsatz der Anweisungen OR und ORI

Der Ausgang Y5 weist den Signalzustand „1“ auf, wenn eine der folgenden Bedingungen erfüllt ist:



- Der Eingang X4 weist den Signalzustand „1“ auf,
ODER
- der Eingang X6 weist den Signalzustand „1“ auf,
ODER
- der Merker M102 weist den Signalzustand „0“ auf.

Der Merker M103 weist den Signalzustand „1“ auf, wenn eine der folgenden Bedingungen erfüllt ist:

- Der Ausgang Y5 weist den Signalzustand „0“ auf,
UND
- der Eingang X7 weist den Signalzustand „1“ auf,
UND
- der Eingang X1 weist den Signalzustand „0“ auf,
ODER
- der Eingang X1 weist den Signalzustand „0“ auf,
UND
- der Merkerkontakt M103 weist den Signalzustand „1“ auf (Merker wird über eine Selbsthaltung auf „1“-Signal gesetzt),
ODER
- der Merkerkontakt M110 weist den Signalzustand „1“ auf.

△

4.6 Gepulster Beginn von Verknüpfungen (LDP, LDF)

	LDP					
	LADE (gepulst); Beginn einer Verknüpfung mit Abfrage der ansteigenden Flanke					
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
●	●	●	●	●	●	●
	LDF					
	LADE (gepulst); Beginn einer Verknüpfung mit Abfrage der abfallenden Flanke					
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
●	●	●	●	●	●	●
Operanden	Programmschritte			Bemerkung		
X, Y, M, S, T, C, nur bei FX3U/FX3UC: D□.b	LDP		2*			
	LDF		2*			

* Die Anzahl der Programmschritte bei einer FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC ist im Abschnitt 4.1.2 angegeben.

Funktion

Programmieren eines gepulsten Verknüpfungsbeginns

Beschreibung

- Der Beginn einer gepulsten Verknüpfung wird mit einer LDP- (ansteigende Flanke) oder einer LDF-Anweisung (abfallende Flanke) programmiert.
- Die LDP- und LDF-Anweisungen müssen am Beginn eines Strompfades programmiert werden.
- Die LDP- und LDF-Anweisungen werden auch im Zusammenhang mit der ANB- und ORB-Anweisung zum Starten einer Verzweigung benutzt (siehe Abschnitte 4.9 und 4.10).
- Die LDP-Anweisung bleibt nach der positiven Flanke für einen Programmzyklus gesetzt.
- Die LDF-Anweisung bleibt nach der negativen Flanke für einen Programmzyklus gesetzt.

HINWEISE

Werden im Zusammenhang mit der LDP- oder LDF-Anweisung die gepulsten Merker M2800 bis M3071 verwendet, wird bei mehrmaliger Programmierung des selben gepulsten Merkers in einem Programm nur der erste Merker verarbeitet. Diese Eigenschaft wird im Zusammenhang mit der STL-Programmierung eingesetzt (siehe Kapitel 5).

Die Funktionen der LD-, AND-, OR-Anweisungen etc. bleiben unverändert.

Beispiel ▾ Einsatz der Anweisungen LDP, LDF

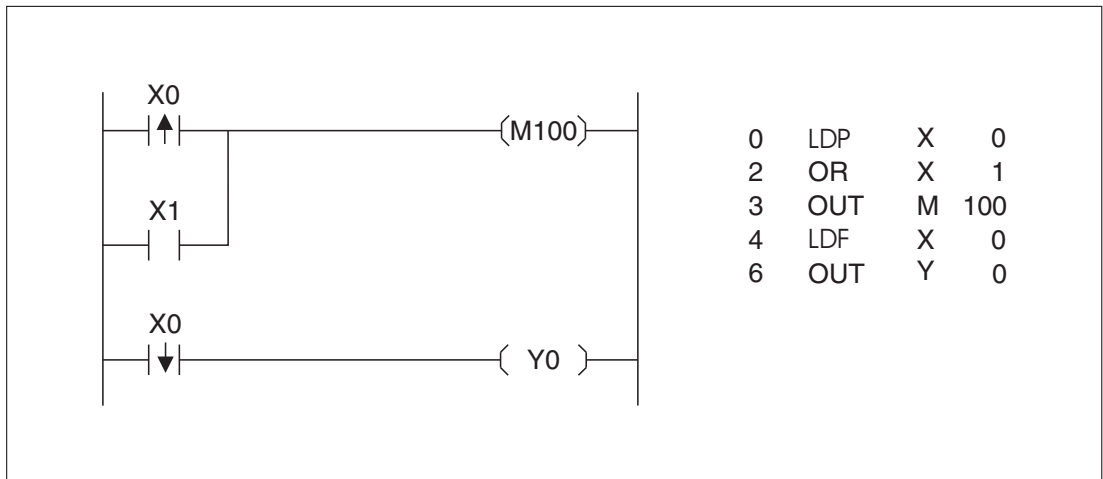



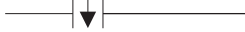
Abb. 4-7: Programmierbeispiel zum Einsatz der Anweisung LDP, LDF

Der Merker M100 hat für die Einschaltdauer von X1 oder bei positiver Flanke von X0 den Zustand „1“.

Der Ausgang Y0 hat bei negativer Flanke von X0 den Zustand „1“.

△

4.7 Gepulste UND-Verknüpfungen (ANDP, ANDF)

	ANDP						
	UND-Verknüpfung (gepulst), UND-Verknüpfung mit Abfrage der ansteigenden Flanke						
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
●	●	●	●	●	●	●	
	ANDF						
	UND-Verknüpfung (gepulst), UND-Verknüpfung mit Abfrage der abfallenden Flanke						
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
●	●	●	●	●	●	●	
Operanden		Programmschritte			Bemerkung		
X, Y, M, S, T, C, nur bei FX3U/FX3UC: D□.b		ANDP			2*		
		ANDF			2*		

* Die Anzahl der Programmschritte bei einer FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC ist im Abschnitt 4.1.2 angegeben.

Funktion

Programmieren einer gepulsten UND-Verknüpfung

Beschreibung

- Eine gepulste UND-Verknüpfung wird mit einer ANDP- (ansteigende Flanke) oder einer ANDF-Anweisung (abfallende Flanke) programmiert.
- Die ANDP- und ANDF-Anweisungen können wie die AND- und ANI-Anweisungen verwendet werden.
- Die ANDP-Anweisung mit der positiven Flanke verarbeitet.
- Die ANDF-Anweisung wird mit der negativen Flanke verarbeitet.

HINWEISE

Werden im Zusammenhang mit der ANDP- oder ANDF-Anweisung die gepulsten Merker M2800 bis M3071 verwendet, wird bei mehrmaliger Programmierung des selben gepulsten Merkers in einem Programm nur der erste Merker verarbeitet. Diese Eigenschaft wird im Zusammenhang mit der STL-Programmierung eingesetzt (siehe Kapitel 5).

Die Funktionen der LD-, AND-, OR-Anweisungen etc. bleiben unverändert.

Beispiel ▾

Einsatz der Anweisungen ANP, ANF

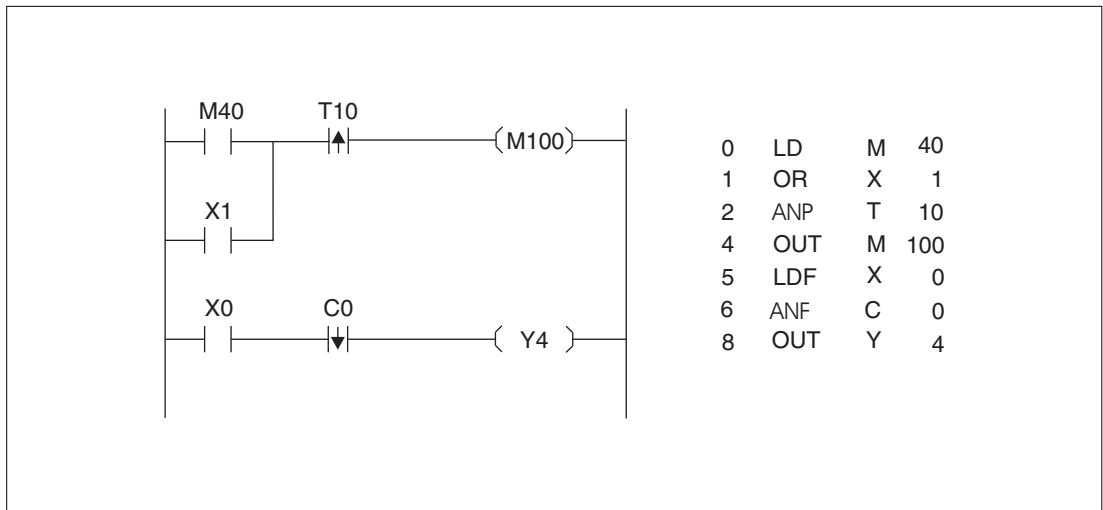


Abb. 4-8: Programmierbeispiel zum Einsatz der Anweisungen ANP, ANF

Der Merker M100 hat bei gesetztem Merker M40 oder gesetztem Eingang X1 und der ansteigenden Flanke vom Timer-Kontakt T10 den Zustand „1“.

Der Ausgang Y4 hat bei eingeschaltetem Eingang X0 und negativer Flanke des Counter-Kontakts C0 den Zustand „1“.

△

4.8 Gepulste ODER-Verknüpfungen (ORP, ORF)

	ORP					
	ODER-Verknüpfung (gepulst); ODER-Verknüpfung mit Abfrage der ansteigenden Flanke					
	CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S
	●	●	●	●	●	●
	ORF					
	ODER-Verknüpfung (gepulst); ODER-Verknüpfung mit Abfrage der abfallenden Flanke					
	CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S
	●	●	●	●	●	●
Operanden	Programmschritte			Bemerkung		
X, Y, M, S, T, C, nur bei FX3U/FX3UC: D□.b	ORP			2*		
	ORF			2*		

* Die Anzahl der Programmschritte bei einer FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC ist im Abschnitt 4.1.2 angegeben.

Funktion

Programmieren einer gepulsten ODER-Verknüpfung

Beschreibung

- Eine gepulste ODER-Verknüpfung wird mit einer ORP- (ansteigende Flanke) oder einer ORF-Anweisung (abfallende Flanke) programmiert.
- Die ORP- und ORF-Anweisungen können wie die OR- und ORI-Anweisungen verwendet werden.
- Die ORP-Anweisung wird mit der positiven Flanke verarbeitet.
- Die ORF-Anweisung wird mit der negativen Flanke verarbeitet.

HINWEISE

Werden im Zusammenhang mit der ORP- oder ORF-Anweisung die gepulsten Merker M2800 bis M3071 verwendet, wird bei mehrmaliger Programmierung des selben gepulsten Merkers in einem Programm nur der erste Merker verarbeitet. Diese Eigenschaft wird im Zusammenhang mit der STL-Programmierung eingesetzt (siehe Kapitel 5).

Die Funktionen der LD-, AND-, OR-Anweisungen etc. bleiben unverändert.

Beispiel ▾

Einsatz der Anweisungen ORP, ORF

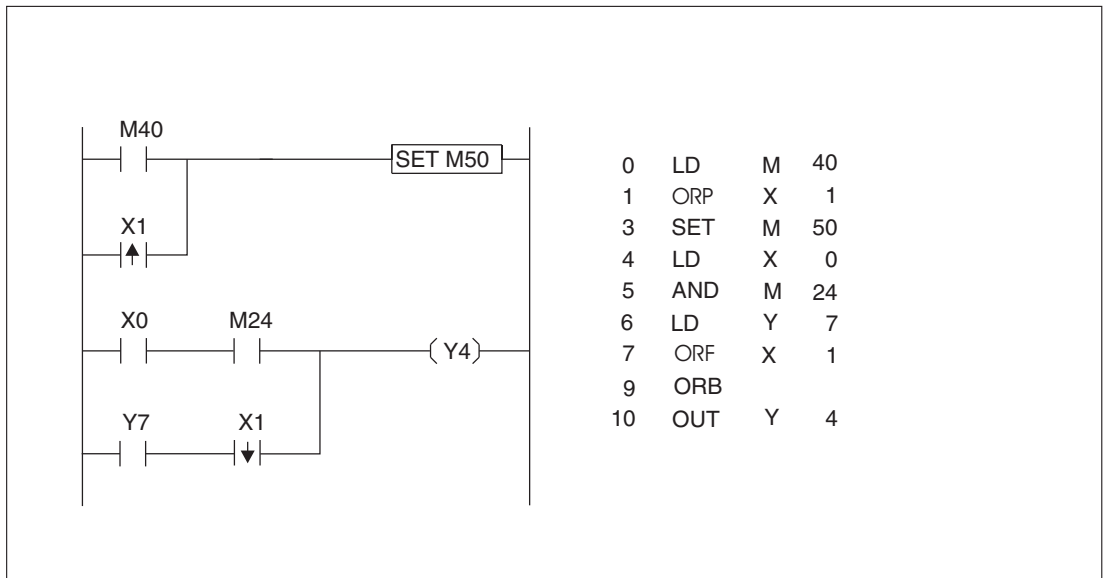


Abb. 4-9: Programmierbeispiel zum Einsatz der Anweisungen ORP, ORF

Der Merker M50 wird über die SET-Anweisung bei gesetztem Merker M40 oder der ansteigenden Flanke des Eingangs X1 gesetzt.

Der Ausgang Y4 hat bei eingeschaltetem Eingang X0 und gesetztem Merker M24 oder bei eingeschaltetem Ausgang Y7 und abfallender Flanke von X1 den Zustand „1“.



4.9 UND-Block-Verknüpfung (ANB)

		ANB					
		UND-BLOCK; Koppelbefehl: Reihenschaltung von Parallelverknüpfungen					
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
	●	●	●	●	●	●	
Operanden		Programmschritte			Bemerkung		
—		ANB-Anweisung			1		

Funktion

Reihenschaltung von parallelen Block-Verknüpfungen

Beschreibung

- Einzelne parallelgeschaltete Blöcke werden separat eingegeben. Um diese Blöcke anschließend in Reihe zu schalten, ist nach jedem Block die ANB-Anweisung zu programmieren.
- Der Beginn einer Verzweigung wird mit der Anweisung LD bzw. LDI programmiert (siehe Abschnitt 4.2).
- Die ANB-Anweisung ist eine unabhängige Anweisung und erfordert keine Angabe eines Operanden.
- Die ANB-Anweisung kann innerhalb des gesamten Programms beliebig oft programmiert werden.
- Im Kontaktplan wird die ANB-Anweisung als eine serielle Verbindung dargestellt. Die ANB-Anweisung erscheint automatisch in der Anweisungsliste, nachdem das Programm im Kontaktplan konvertiert wurde.

HINWEIS

Wenn Sie mehrere einzelne Blöcke direkt hintereinander programmieren, müssen Sie die Anzahl der LD- und LDI-Anweisungen und somit auch die Anzahl der ANB-Anweisungen auf 8 beschränken.

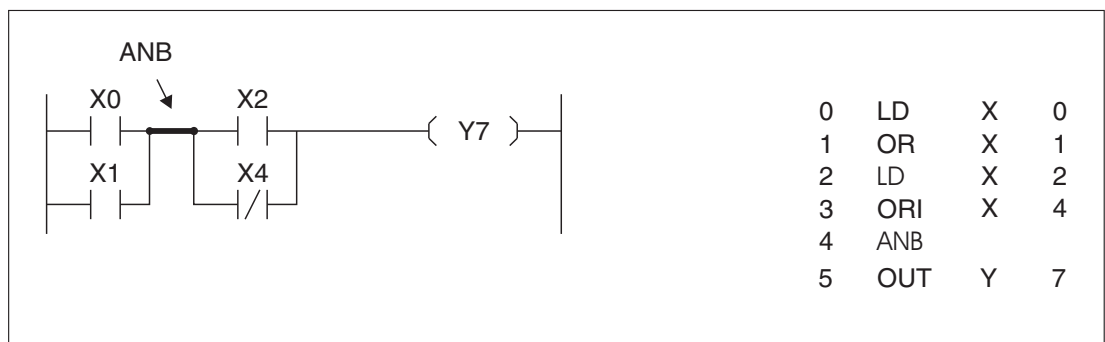


Abb. 4-10: Programmierbeispiel zum Einsatz der Anweisung ANB

4.10 ODER-Block-Verknüpfung (ORB)

		ORB					
		ODER-BLOCK; Koppelbefehl: Parallelschaltung von Reihenverknüpfungen					
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
	●	●	●	●	●	●	
Operanden		Programmschritte			Bemerkung		
—		ORB-Anweisung			1		

Funktion

Parallelschaltung von in Reihe geschalteten Block-Verknüpfungen

Beschreibung

- Werden mehrere in Reihe geschaltete Blöcke parallel geschaltet, muss nach der Programmierung jedes einzelnen Blocks eine ORB-Anweisung eingegeben werden.
- Der Beginn einer Verzweigung wird mit der Anweisung LD bzw. LDI programmiert (siehe Abschnitt 4.2).
- Die ORB-Anweisung ist eine unabhängige Anweisung und erfordert keine Angabe eines Operanden.
- Die ORB-Anweisung kann innerhalb des gesamten Programms beliebig oft programmiert werden.
- Im Kontaktplan wird die ORB-Anweisung als eine parallele Verbindung dargestellt. Die ORB-Anweisung erscheint automatisch in der Anweisungsliste, nachdem das Programm im Kontaktplan konvertiert wurde.

HINWEIS

Wenn Sie mehrere einzelne Blöcke direkt hintereinander programmieren, müssen Sie die Anzahl der LD- und LDI-Anweisungen und somit auch die Anzahl der ORB-Anweisungen auf 8 beschränken.

	Empfohlene Programmierung	Ungünstige Programmierung
	0 LD X 0	0 LD X 0
	1 AND X 1	1 AND X 1
	2 LD X 2	2 LD X 2
	3 AND X 3	3 AND X 3
	4 ORB	4 LDI X 4
	5 LDI X 4	5 AND X 5
	6 AND X 5	6 ORB
	7 ORB	7 ORB
8 OUT Y 6	8 OUT Y 6	

Abb. 4-11: Programmierbeispiel zum Einsatz der Anweisung ORB

4.11 Verknüpfungsergebnis verarbeiten (MPS, MRD, MPP)

	MPS					
	Push Down Stack; Abspeichern eines Verknüpfungsergebnisses					
	CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S
	●	●	●	●	●	●
	MRD					
	Read Down Stack; Lesen eines Verknüpfungsergebnisses					
	CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S
	●	●	●	●	●	●
	MPP					
	Pop Up Stack; Lesen und Löschen eines Verknüpfungsspeichers					
	CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S
	●	●	●	●	●	●
Operanden	Programmschritte			Bemerkung		
—	MPS-Anweisung	1				
	MRD-Anweisung	1				
	MPP-Anweisung	1				

Funktion

Die Anweisungen MPS, MRD und MPP dienen dazu, Verknüpfungsebenen aufzubauen. Mit Hilfe dieser Anweisungen wird der Programmieraufwand erheblich reduziert.

Beschreibung

- Mit der Anweisung MPS wird das vorangegangene Verknüpfungsergebnis gespeichert.
- Mit Hilfe der MRD-Anweisung werden mehrere Teilverzweigungen zwischen dem Beginn (MPS) und dem Ende (MPP) der Verzweigung möglich.
- Die letzte Teilverzweigung wird mit der MPP-Anweisung gestartet.
- Die mit einer MPS-Anweisung eröffnete Verzweigung muss immer mit einer MPP-Anweisung abgeschlossen werden.
- Alle drei Anweisungen erfordern keine Angabe eines Operanden.
- Im Kontaktplan werden diese Anweisungen nicht dargestellt. Erfolgt die Programmierung im Kontaktplan, werden die Verzweigungen wie bisher gesetzt. Die MPS-, MRD- und MPP-Anweisungen erscheinen automatisch in der Anweisungsliste, nachdem das Programm im Kontaktplan konvertiert wurde.

HINWEIS

Es sind maximal 11 Verknüpfungsebenen zulässig.

Eine detaillierte Beschreibung der drei Anweisungen soll anhand der folgenden Programmierbeispiele gegeben werden.

Beispiel ▾ Einsatz der Anweisungen MPS, MRD, MPP

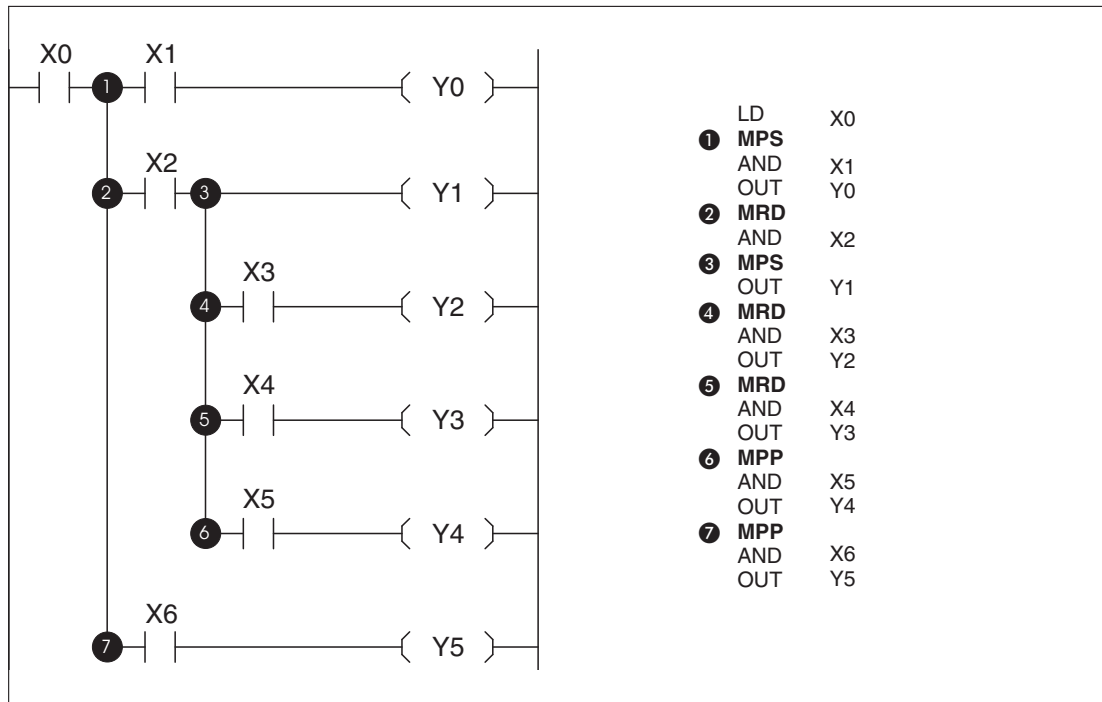


Abb. 4-12: Programmierbeispiel zum Einsatz der Anweisungen MPS, MRD und MPP

- ① **MPS**
Das Zwischenergebnis (hier X0) in der 1. Verknüpfungsebene wird an der 1. Stelle im Verknüpfungsspeicher abgelegt.
- ② **MRD**
Vor Ausführung der nächsten Anweisung wird das Zwischenergebnis an der 1. Stelle des Verknüpfungsspeichers abgefragt.
- ③ **MPS**
Das Zwischenergebnis in der 2. Verknüpfungsebene wird an der 1. Stelle im Verknüpfungsspeicher abgelegt. Der bereits vorhandene Wert an der 1. Stelle rutscht an die 2. Stelle.
- ④ **MRD**
Vor Ausführung der nächsten Anweisung wird das Zwischenergebnis an der 1. Stelle des Verknüpfungsspeichers abgefragt.
- ⑤ **MRD**
Vor Ausführung der nächsten Anweisung wird das Zwischenergebnis an der 1. Stelle des Verknüpfungsspeichers abgefragt.
- ⑥ **MPP**
Vor Ausführung der nächsten Anweisung wird das Zwischenergebnis an der 1. Stelle im Verknüpfungsspeicher abgefragt. Die Operationen in der 2. Verknüpfungsebene werden abgeschlossen. Der Wert an der 1. Stelle im Verknüpfungsspeicher wird gelöscht. Der Wert der 2. Stelle rutscht zurück an die 1. Stelle.
- ⑦ **MPP**
Vor Ausführung der nächsten Anweisung wird das Zwischenergebnis an der 1. Stelle im Verknüpfungsspeicher abgefragt. Die Operationen in der 1. Verknüpfungsebene werden abgeschlossen, und der Verknüpfungsspeicher wird gelöscht.



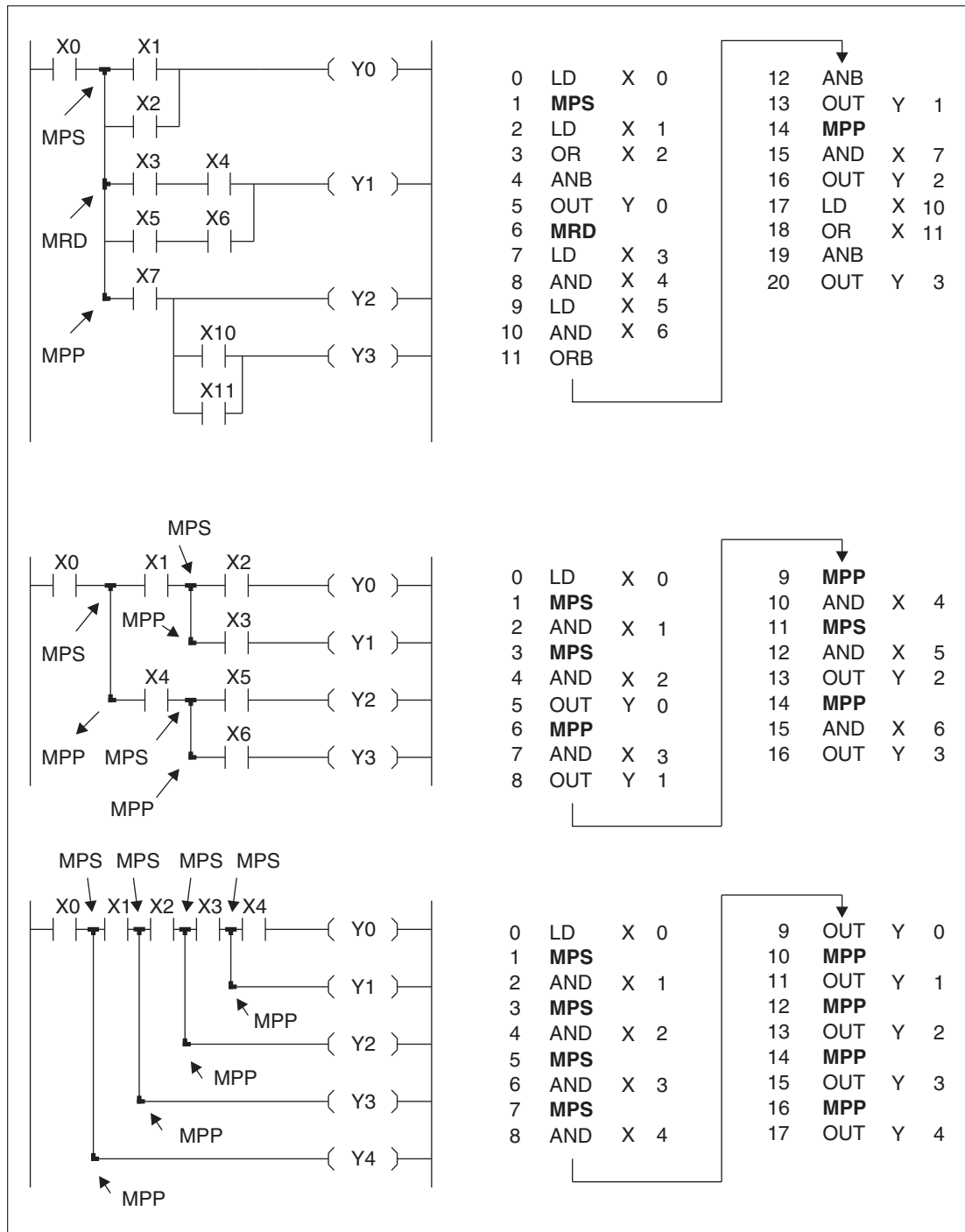


Abb. 4-13: Programmierbeispiel zum Einsatz der Anweisungen MPS, MRD und MPP

4.12 Kontrollbedingung setzen/zurücksetzen (MC, MCR)

		MC						
		Master Control; Setzen einer Kontrollbedingung						
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC		
	●	●	●	●	●	●		
		MCR						
		Master Control Reset; Rücksetzen einer Kontrollbedingung						
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC		
	●	●	●	●	●	●		
Operanden		Programmschritte			Bemerkung			
MC: Y, M, keine Sondermerker		MC-Anweisung			3*			
MCR: N		MCR-Anweisung			2			

* Die Anzahl der Programmschritte bei einer FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC ist im Abschnitt 4.1.2 angegeben.

Funktion

Durch Setzen (MC) oder Rücksetzen (MCR) einer Kontrollbedingung können einzelne Programmbereiche aktiviert bzw. deaktiviert werden. Die Funktion arbeitet also wie ein Hauptkontakt auf der linken Sammelschiene (Kontaktplanprogrammierung).

Beschreibung

- Mit der MC-Anweisung wird eine Kontrollbedingung zur Aktivierung eines bestimmten Programmbereiches gesetzt.
 - Welcher Programmbereich aktiviert werden soll, wird durch die Angabe der Programmverzweigungsadresse n: N0 bis N7 (Nesting-Adresse) festgelegt.
 - Die Angabe des Operanden Y oder M definiert einen Einschaltkontakt. Dieser Kontakt aktiviert den Programmbereich n, sobald die Eingangsbedingung für die MC-Anweisung erfüllt ist.
- Nach der Programmierung der MC-Anweisung müssen Sie immer eine LD- oder LDI-Anweisung programmieren (siehe Abschnitt 4.2)
- Die MCR-Anweisung setzt den MC-Kontakt zurück und stellt somit das Ende des Programmbereichs n dar.
- Ist die Eingangsbedingung nicht erfüllt, verändern sich die Zustände der Operanden zwischen MC und MCR wie folgt:
 - Bei remanenten Countern und Operanden, die in Zusammenhang mit den SET-, RST-Anweisungen programmiert werden, bleibt der Zustand erhalten (siehe Abschnitt 4.13).
 - Alle Timer und Operanden, die in Zusammenhang mit der OUT-Anweisung programmiert wurden, werden zurückgesetzt.
- Innerhalb eines Programms können bis zu 8 Verzweigungsebenen aufgebaut werden. Die Verzweigungsebenen werden durch den Parameter „n“ gekennzeichnet.
- Was Sie beim Einsatz von mehreren MC- und MCR-Anweisungen innerhalb eines Programms beachten müssen, wird in einem der folgenden Beispiele erläutert.

HINWEIS | Durch den Einsatz der MC- bzw. MCR-Anweisung wird die Programmzykluszeit nicht verkürzt.

Beispiel ▾ Einsatz der Anweisungen MC, MCR

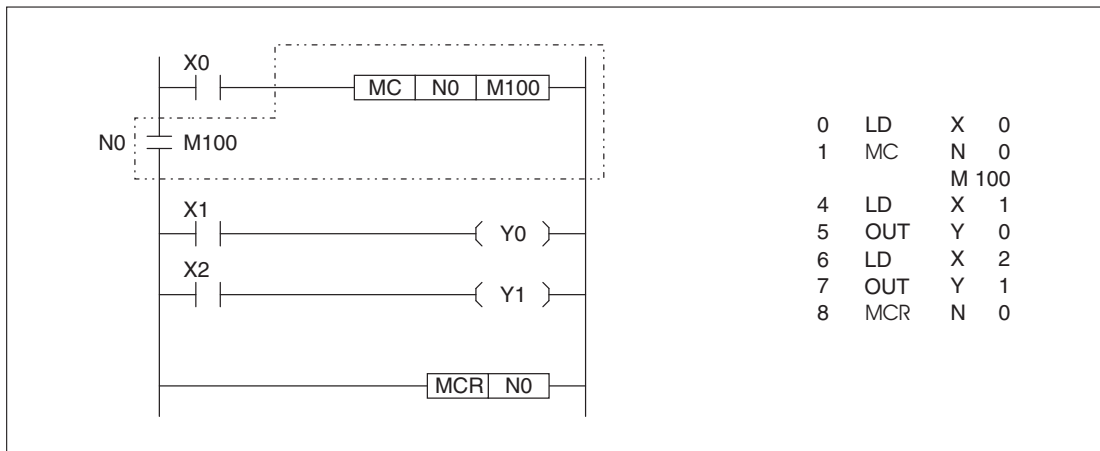


Abb. 4-14: Programmierbeispiel zum Einsatz der Anweisungen MC und MCR

Doppelbelegungen von Operanden innerhalb und außerhalb eines Master-Control-Bereichs führen auch bei inaktivem Master-Control-Bereich zu den in Abschnitt 4.3 (Doppelbelegung von Ausgängen) beschriebenen Problemen.

Sobald die Eingangsbedingung für die MC-Anweisung erfüllt ist, wird der Merkerkontakt M100 (Programmverzweigungsadresse N0) durchgeschaltet. Alle Strompfade zwischen der MC- und der MCR-Anweisung sind jetzt aktiviert. Der Signalzustand des Ausgangs Y0 bzw. Y1 ist dann nur noch vom Signalzustand des Eingangs X1 bzw. X2 abhängig.

△

Einsatz von mehreren MC- und MCR-Anweisungen innerhalb eines Programms

Bei der Programmierung von mehreren MC- und MCR-Anweisungen innerhalb eines Programms müssen Sie darauf achten,

- dass die **erste MC**-Anweisung mit der **niedrigsten** Programmverzweigungsadresse N beginnt und
- dass die **erste MCR**-Anweisung mit der **höchsten** Programmverzweigungsadresse N beginnt.

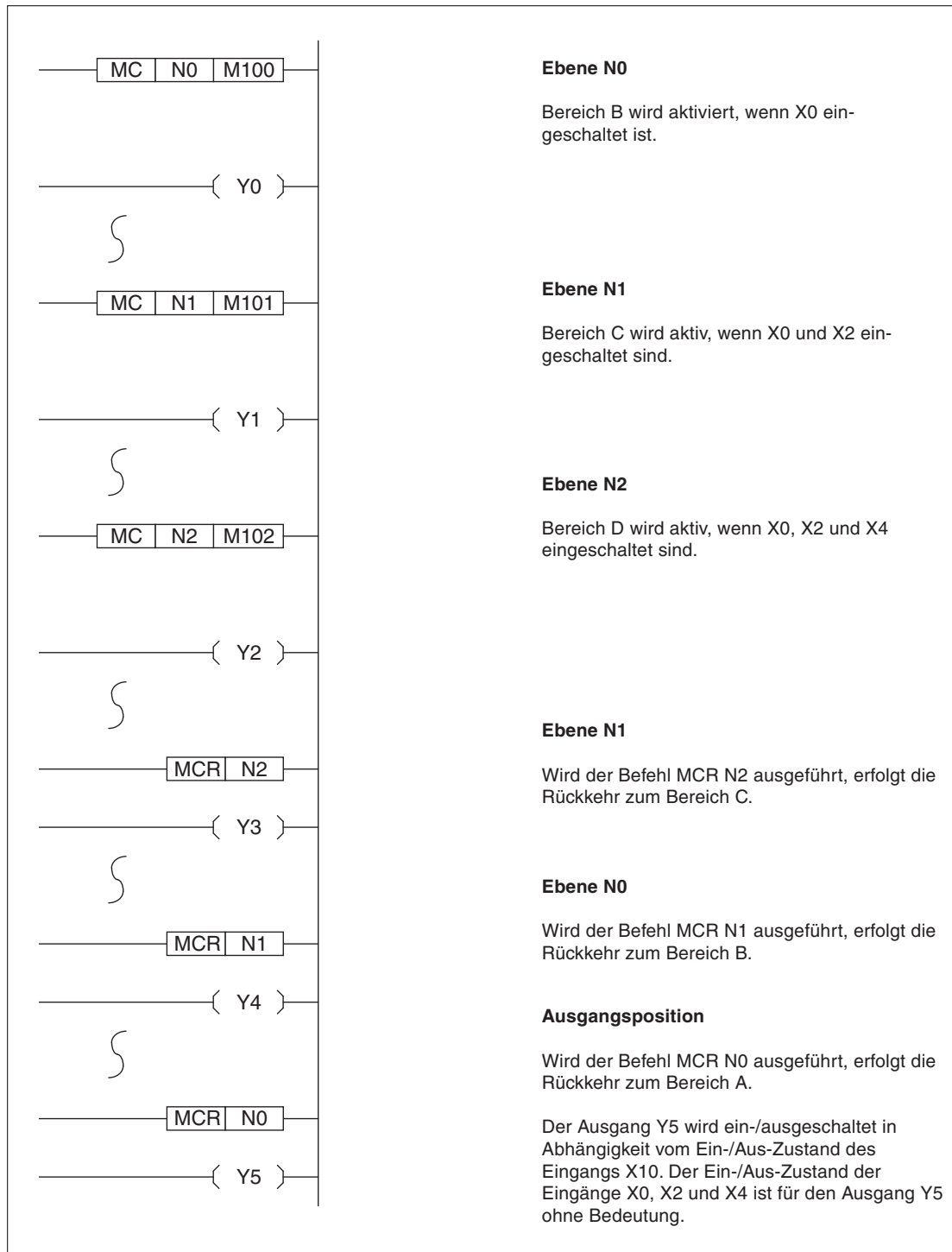


Abb. 4-15: Programmierbeispiel zum Einsatz von mehreren MC- und MCR-Anweisungen innerhalb eines Programms

4.13 Setzen und Rücksetzen von Operanden (SET, RST)

SET

	SET						
	Setzen; Operanden setzen						
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
	●	●	●	●	●	●	
Operanden	Programmschritte			Bemerkung			
Y, M, S, nur bei FX3U/FX3UC: D□.b	Y, M		1*				
	S, Sondermerker		2*				

* Die Anzahl der Programmschritte bei einer FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC ist im Abschnitt 4.1.2 angegeben.

Funktion

Die Signalzustände von Operanden lassen sich mit der Anweisung SET (Setzen) direkt festlegen.

Beschreibung

- Mit der SET-Anweisung können Sie einen Operanden Y, M, S oder – bei einer FX3U oder FX3UC – ein Bit eines Datenregisters (D□.b) auf den Signalzustand „1“ setzen.
 - Sobald die Eingangsbedingung (Signal „1“) für die SET-Anweisung ansteht, wird der angegebene Operand gesetzt.
 - Auch wenn die Eingangsbedingung für die SET-Anweisung nicht mehr ansteht, bleibt der angegebene Operand gesetzt.

RST

	RST						
	Rücksetzen; Operanden rücksetzen						
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
	●	●	●	●	●	●	
Operanden	Programmschritte			Bemerkung			
Y, M, S, D, V, Z, T, C, nur bei FX3U/FX3UC: D□.b	Y, M		1	S, T, C		2*	
	D, V, Z, Sonderregister					3*	

* Die Anzahl der Programmschritte bei einer FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC ist im Abschnitt 4.1.2 angegeben.

Funktion

Die Signalzustände von Operanden lassen sich mit der Anweisung RST (Rücksetzen) direkt festlegen.

- Mit einer RST-Anweisung können Sie den angegebenen Operanden zurücksetzen. Dies bedeutet:
 - Ausgänge Y, Merker M, Schrittstatusoperanden S und Bits eines Datenregisters (D□.b) werden ausgeschaltet (Signalzustand „0“).

- Istwerte von Timern und Countern sowie Inhalte von Registern D, V und Z werden auf 0 zurückgesetzt.
- Sobald die Eingangsbedingung (Signal „1“) für die RST-Anweisung ansteht, wird der angegebene Operand zurückgesetzt.
- Auch wenn die Eingangsbedingung für die RST-Anweisung nicht mehr ansteht, bleibt der angegebene Operand zurückgesetzt.

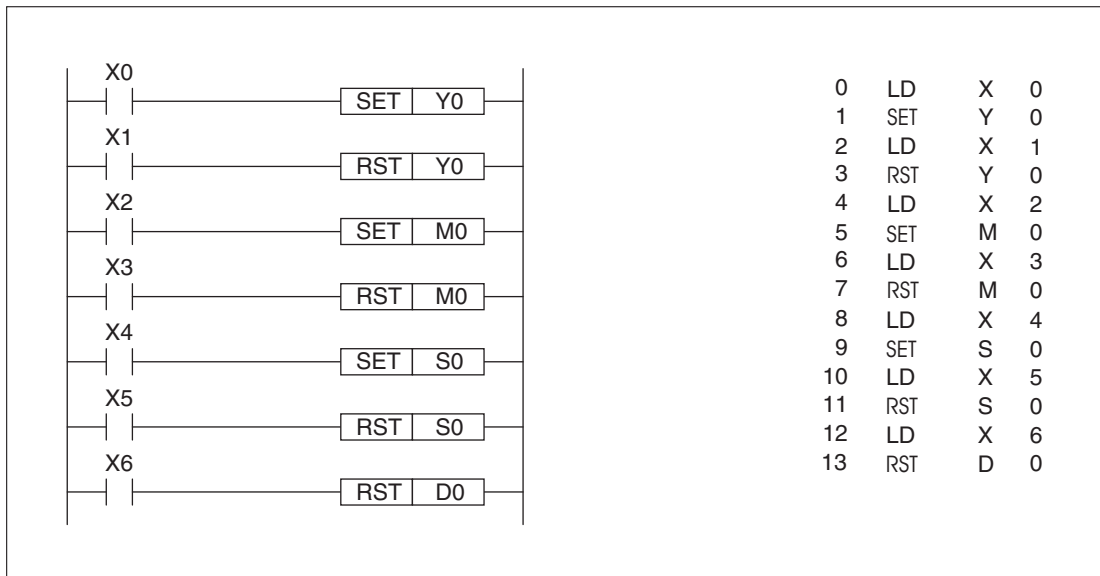


Abb. 4-16: Programmierbeispiel für den Einsatz der Anweisungen SET und RST

Beispiel ▾

Rücksetzen eines 16-Bit-Counters mit Hilfe der RST-Anweisung

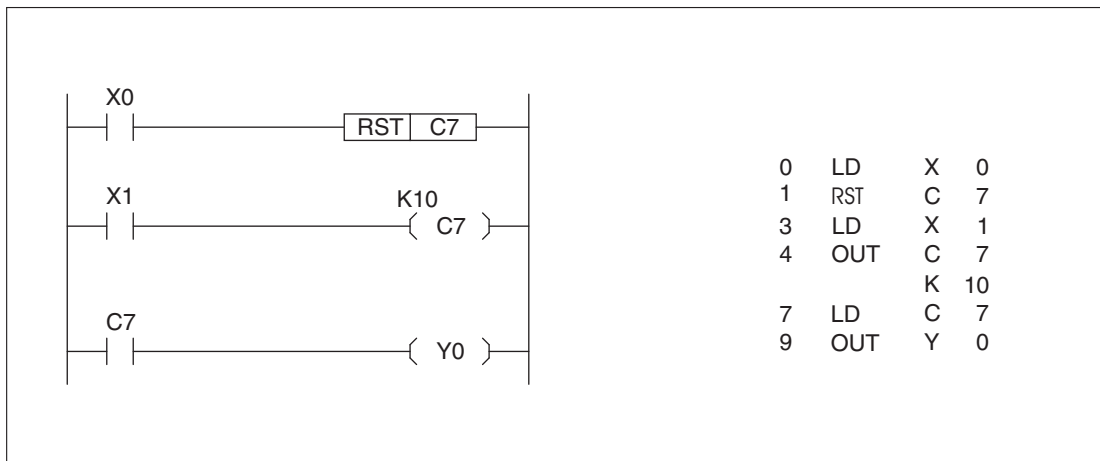
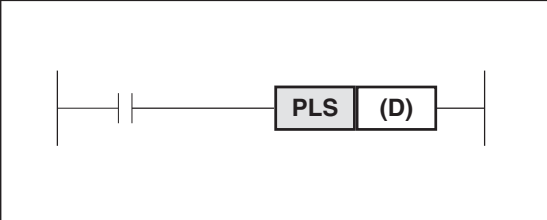
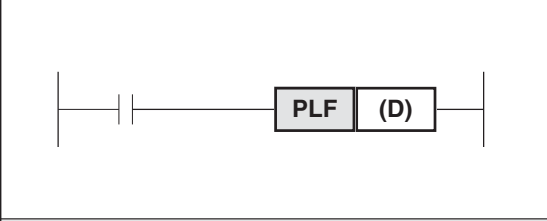


Abb. 4-17: Programmierbeispiel zum Rücksetzen eines 16-Bit-Counters mit Hilfe der RST-Anweisung

Der Ausgang Y0 wird aktiviert, wenn der Sollwert K10 erreicht ist. Sobald der Eingang X0 eingeschaltet ist, wird der Ausgang Y0 zurückgesetzt und der Istwert des Zählers C7 auf 0 gesetzt.

△

4.14 Erzeugen eines einmaligen Impulses (PLS, PLF)

		PLS					
		Impulserzeugung; Erzeugen eines einmaligen Impulses bei ansteigender Flanke					
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
	●	●	●	●	●	●	
		PLF					
		Impulserzeugung; Erzeugung eines einmaligen Impulses bei abfallender Flanke					
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
	●	●	●	●	●	●	
Operanden		Programmschritte			Bemerkung		
Y, M		PLS-Anweisung			2*		
		PLF-Anweisung			2*		

* Die Anzahl der Programmschritte bei einer FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC ist im Abschnitt 4.1.2 angegeben.

Funktion

Erzeugen eines einmaligen Impulses – Flankenerkennung –, unabhängig von der Dauer des anstehenden Eingangssignals, zum Einschalten des zugehörigen Operanden. Der Operand bleibt für die Dauer eines Programmzyklus eingeschaltet.

Beschreibung

- Die PLS- und PLF-Anweisungen können im Zusammenhang mit Merkern M und digitalen Ausgängen Y benutzt werden. Die Anweisungen erzeugen einen gleichbleibenden Impuls, unabhängig von der Dauer des anstehenden Eingangssignals.
- Nach Ausführung einer PLS- oder PLF-Anweisung steht das Signal des zugehörigen Operanden (Y oder M) während der Dauer eines Programmzyklus an.
- Die PLS-Anweisung erzeugt einen einmaligen Impuls bei ansteigender Flanke des Eingangssignals.
- Die PLF-Anweisung erzeugt einen einmaligen Impuls bei abfallender Flanke des Eingangssignals.

HINWEIS

| Sondermerker dürfen nicht mit einer PLS- oder PLF-Anweisung aktiviert werden.

Beispiel ▾ Einsatz der Anweisungen PLS, PLF

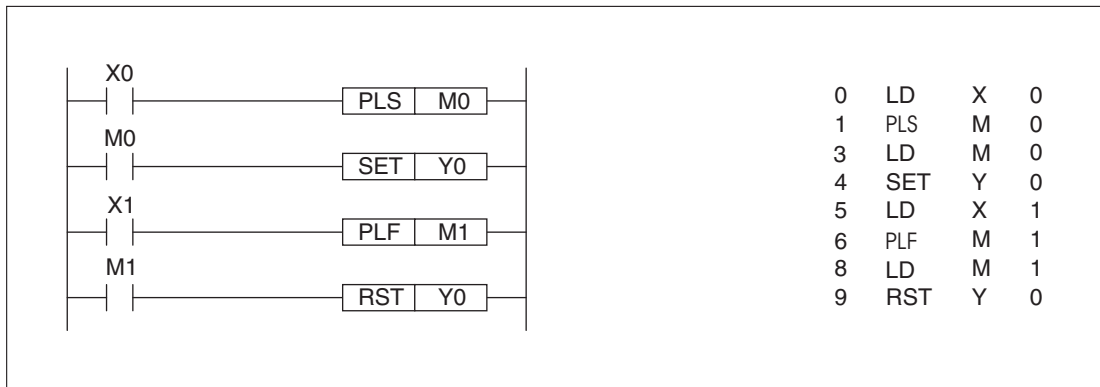


Abb. 4-18: Programmierbeispiel zum Einsatz der Anweisungen PLS und PLF

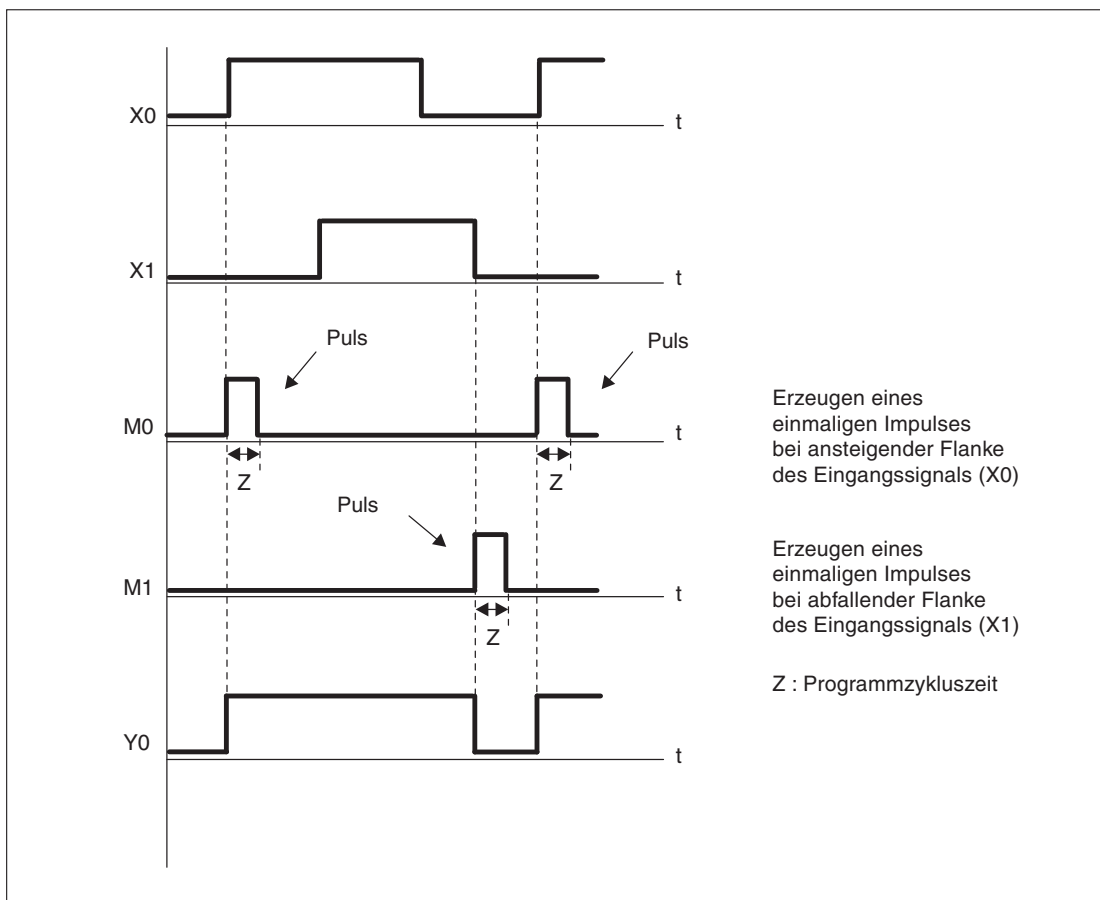


Abb. 4-19: Programmierbeispiel, Darstellung der Eingangssignalverarbeitung und der Impulserzeugung

Beim Wechseln des Eingangssignals am Eingang X0 von „0“ auf „1“ (ansteigende Flanke) erhält der Merker M0 durch die PLS-Anweisung einen Impuls. Mit diesem Impuls wird über den Merkerkontakt M0 der Ausgang Y0 gesetzt. Erst wenn am Eingang X1 ein Eingangssignalwechsel von „1“ auf „0“ (abfallende Flanke) auftritt, wird der Ausgang Y0 wieder zurückgesetzt. ▴

4.15 Inversion von Verarbeitungsergebnissen (INV)

	INV					
	Inversion; Umkehrung eines Verknüpfungsergebnisses					
	CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S
	●	●	●	●	●	●
Operanden	Programmschritte			Bemerkung		
—	INV-Anweisung			1		

Funktion

Umkehren des Signalzustandes von Verarbeitungsergebnissen

Beschreibung

Eine INV-Anweisung invertiert den Signalzustand des Ergebnisses vorstehender Anweisung.

- Lautet das Verarbeitungsergebnis 1, wird es nach der Inversion 0.
- Lautet das Verarbeitungsergebnis 0, wird es nach der Inversion 1.
- Eine INV-Anweisung kann wie eine AND- und ANI-Anweisungen verwendet werden.

HINWEISE

Eine INV-Anweisung kann zur Signalumkehrung des Ergebnisses einer komplexen Schaltung verwendet werden.

Eine INV-Anweisung kann zur Signalumkehr der Ergebnisse der gepulsten Anweisungen LDP, LDF, ANP etc. verwendet werden.

Beispiel ▾

Einsatz der INV-Anweisung

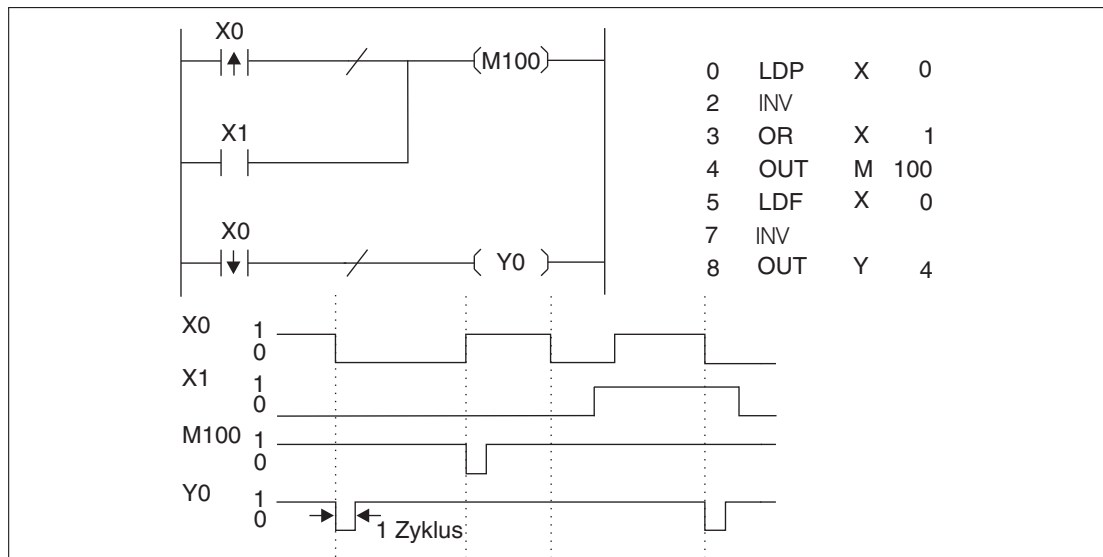


Abb. 4-20: Programmierbeispiel zum Einsatz der INV-Anweisung

Der Merker M100 hat bei positiver Flanke von X0 den Zustand „0“.

Der Ausgang Y0 hat bei abfallender Flanke von X0 den Zustand „0“.



4.16 Impuls aus Verknüpfungsergebnis (MEP, MEF)

↑	MEP						
	Impulserzeugung; Erzeugen eines einmaligen Impulses bei ansteigender Flanke des Verknüpfungsergebnisses						
↓	CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
					●	●	●
↓	MEF						
	Impulserzeugung; Erzeugen eines einmaligen Impulses bei abfallender Flanke des Verknüpfungsergebnisses						
↓	CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
					●	●	●
Operanden		Programmschritte				Bemerkung	
—		MEP		1			
—		MEF		1			

Funktion

Wandlung des Verknüpfungsergebnisses in einen Impuls

Beschreibung

- Die MEP- und die MEF-Anweisung erzeugen bei der ansteigenden bzw. abfallenden Flanke des Verknüpfungsergebnisses der vor diesen Anweisungen liegenden Verknüpfungen einen Impuls. Dadurch wird die Wandlung der Zustände von mehreren in Reihe geschalteten Kontakten in einen Impuls vereinfacht.
- Die MEP-Anweisung erzeugt einen Impuls, wenn das Verknüpfungsergebnis der vor der MEP-Anweisung liegenden Kontakte wahr wird („1“).
- Die MEF-Anweisung erzeugt einen Impuls, wenn das Verknüpfungsergebnis der vor der MEF-Anweisung liegenden Kontakte unwahr wird („0“).

HINWEISE

Die MEP- und die MEF-Anweisung werden eventuell nicht normal ausgeführt, wenn ein durch Indexvergabe adressierter Operand modifiziert und in einen Impuls gewandelt wird (beispielsweise durch Unterprogramme, FOR/NEXT-Anweisungen etc.).

Da die MEP- und die MEF-Anweisung das Verknüpfungsergebnis der unmittelbar vor ihnen ausgeführten Anweisungen verwenden, können sie nur in Form einer AND-Anweisung eingesetzt werden. Die MEP- und die MEF-Anweisung können nicht in Form einer LD- oder OR-Anweisung verwendet werden.

Fehlerquellen

Bei einer MEP- oder MEF-Anweisung tritt kein Ausführungsfehler auf.

Verhalten beim Ändern des Programms im Zustand RUN der SPS

● MEP-Anweisung

Wird ein Programm mit einer MEP-Anweisung in die SPS geschrieben, während sich das SPS-Grundgerät in der Betriebsart RUN befindet, erzeugt die MEP-Anweisung danach einen Impuls, wenn das Verknüpfungsergebnis der vor der MEP-Anweisung liegenden Kontakte wahr ist.

● MEF-Anweisung

Wird ein Programm mit einer MEF-Anweisung in die SPS geschrieben, während sich das SPS-Grundgerät in der Betriebsart RUN befindet, bleibt unabhängig vom Verknüpfungsergebnis der vor der MEF-Anweisung liegenden Kontakte der Ausgang der MEF-Anweisung ausgeschaltet. (Es wird kein Impuls erzeugt.)

Die MEF-Anweisung erzeugt einen Impuls, wenn das Verknüpfungsergebnis der vor der MEF-Anweisung liegenden Kontakte unwahr wird.

Verknüpfungsergebnis vor der MEP/MEF-Anweisung (bei der Übertragung des Programms in RUN)	MEP-Anweisung	MEF-Anweisung
Unwahr („0“)	Kein Impuls	Kein Impuls
Wahr („1“)	Impuls	Kein Impuls

Tab. 4-5: Verhalten der MEP- und der MEF-Anweisung beim Schreiben in die SPS im Zustand RUN

Beispiel ▾

MEP-Anweisung

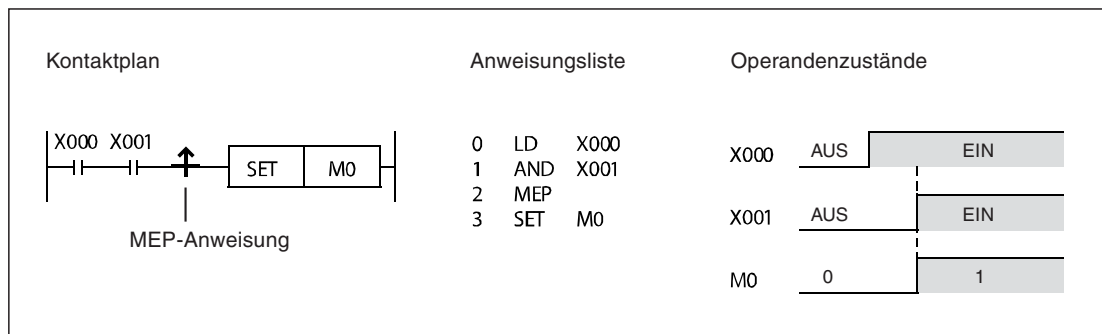


Abb. 4-21: Beispiel zur MEP-Anweisung



Beispiel ▾

MEF-Anweisung

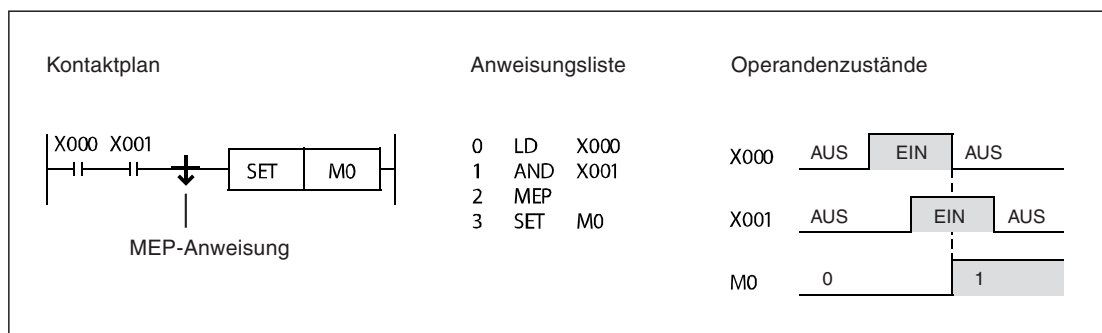


Abb. 4-22: Beispiel zur MEF-Anweisung



4.17 Leerzeile im Programm (NOP)

		NOP					
		Leerzeile; Leerzeile im Programm ohne logische Funktion					
CPU		FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
		●	●	●	●	●	●
Operanden	Programmschritte			Bemerkung			
—	NOP-Anweisung			1			

Funktion

Es wird eine Leerzeile ohne eine logische Funktion erzeugt, die später durch weitere Anweisungen in einem noch nicht fertiggestellten Programm aufgefüllt werden kann.

Beschreibung

- Nach Abschluss der Programmierfolge sollten NOP-Befehle gelöscht werden, da ansonsten die Programmzykluszeit unnötig verlängert wird.
- Die Anzahl der NOP-Befehle ist nicht begrenzt.
- Beim Löschen des gesamten Programms werden sämtliche Anweisungen durch NOP-Anweisungen überschrieben.
- Nachträgliches Einfügen von NOP-Anweisungen mit einem Handprogrammiergerät sollten Sie mit der INSERT-Funktion durchführen.

HINWEIS

Werden die Anweisungen LD, LDI, ANB oder ORB durch eine NOP-Anweisung ersetzt, kann sich der logische Schaltungsaufbau wesentlich verändern. Hierdurch kann es zu Programmablauf Fehlern kommen.

Beispiel ▾

Einsatz der Anweisung NOP

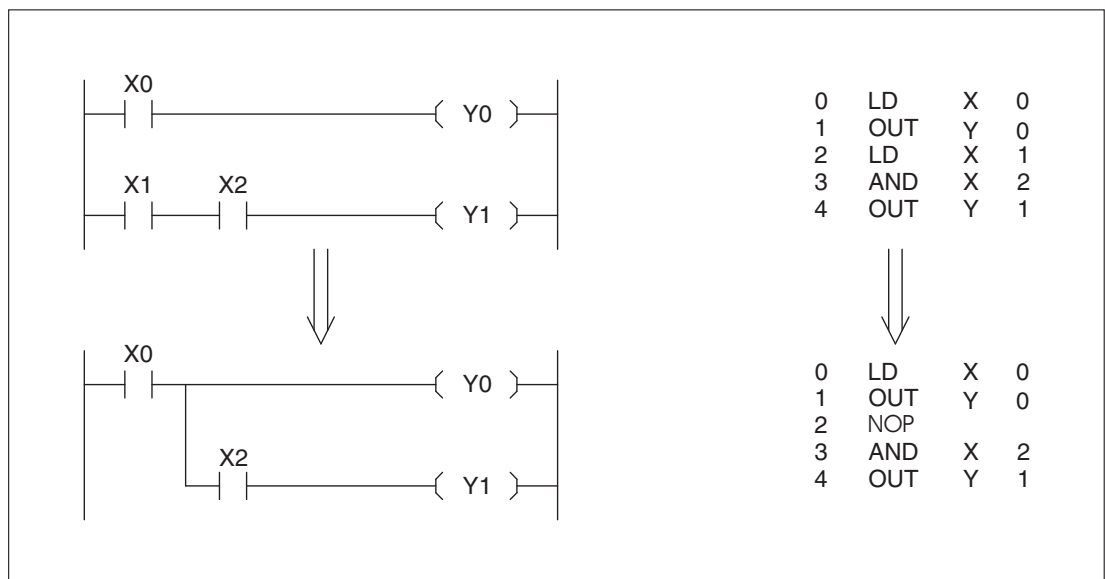


Abb. 4-23: Programmierbeispiel zum Einsatz der Anweisung NOP

Der Eingang X1 wird durch eine NOP-Anweisung ersetzt. Der logische Schaltungsaufbau hat sich hierdurch geändert.

4.18 SPS-Programmende (END)

		END					
		Ende; SPS-Programmende Sprung zum Programmanfang					
CPU		FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
		●	●	●	●	●	●
Operanden		Programmschritte			Bemerkung		
—		END-Anweisung			1		

Funktion

Abschließen eines SPS-Programms und Sprung zum Programmanfang (Schritt 0)

HINWEIS

Der GX Developer, der GX IEC Developer und GX Works2 erzeugen die END-Anweisung automatisch.

Beschreibung

- Jedes SPS-Programm wird mit einer END-Anweisung abgeschlossen.
- Wurde eine END-Anweisung programmiert, wird an dieser Stelle die Programmabarbeitung beendet. Nachfolgende Programmbereiche werden nicht mehr berücksichtigt. Nach Abarbeitung der END-Anweisung erfolgt die Ausgangsbearbeitung. Die Programmabarbeitung springt dann zum Programmanfang (Schritt 0) zurück.
- Um zur schrittweisen Programmüberprüfung einzelne Programmabschnitte zu erzeugen, kann die END-Anweisung auch innerhalb des Programms eingesetzt werden. Die Anweisungen nach der END-Anweisung werden bei der Überprüfung nicht berücksichtigt. Diese „eingeschobenen“ END-Anweisungen müssen Sie anschließend wieder löschen.

HINWEIS

Nach Ausführung der END-Anweisung findet eine Auffrischung des Watch-Dog-Timers und des Prozessabbildes statt.

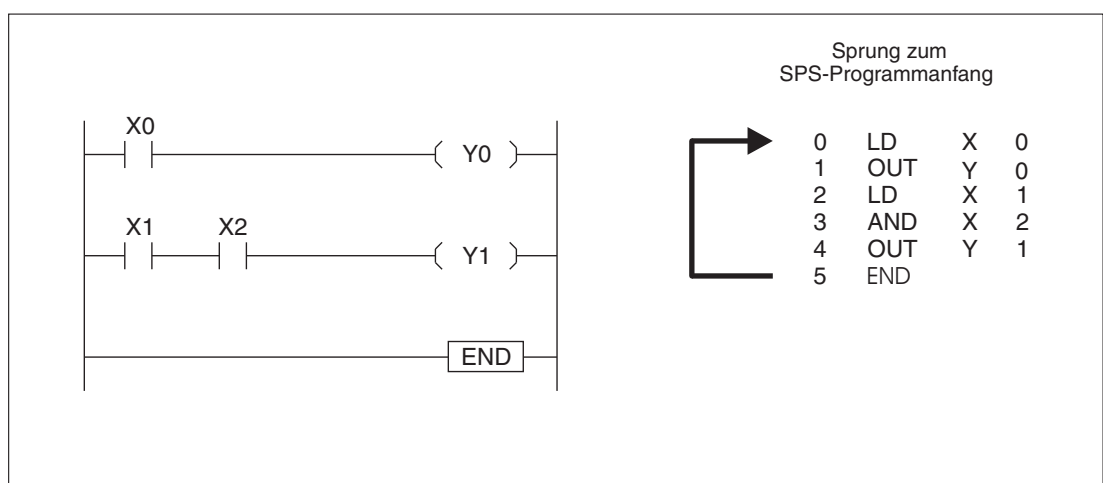


Abb. 4-24: Programmierbeispiel zum Einsatz der END-Anweisung

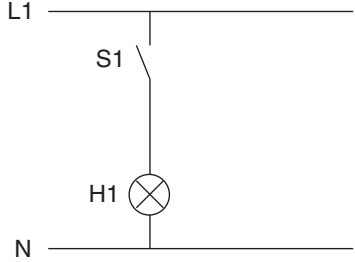
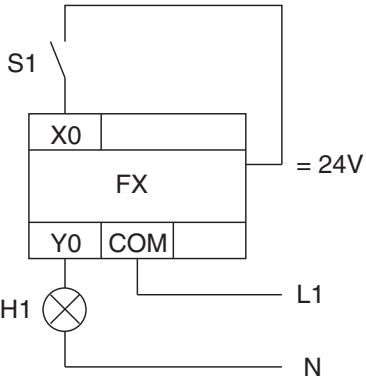
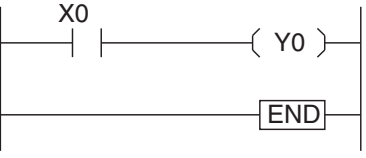
4.19 Programmbeispiele

Der folgende Abschnitt zeigt einige einfache Beispiele zur Anwendung des Grundbefehlssatzes. Die Beispiele können direkt programmiert und ausgeführt werden.

- Abfrage eines Eingangs (Öffner und Schließer)
- Reihenschaltung von Eingängen
- Parallelschaltung von Eingängen
- Selbsthaltung eines Ausgangs
- Einschaltverzögerung
- Ausschaltverzögerung
- Aufwärtzzähler

4.19.1 Abfrage eines Eingangs

Betätigter Schließer

Beispiel	Stromlaufplan
<p>Bei Betätigen des Schließers S1 soll der Melder H1 leuchten.</p>	
Zuordnungsliste	Beschaltung der SPS
<p>Schließer: S1 X0 Melder: H1 Y0</p>	
Anweisungsliste	Kontaktplan
<p>0 LD X0 1 OUT Y0 2 END</p>	
Anmerkung	
<p>Der Ausgang Y0 führt das Signal „1“, wenn an Eingang X0 das Signal „1“ ansteht.</p>	

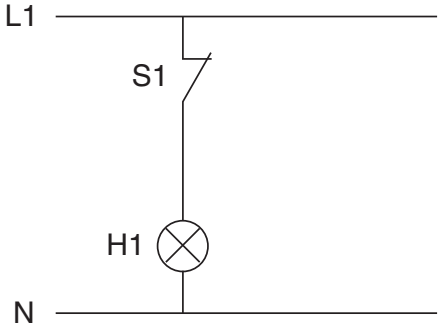
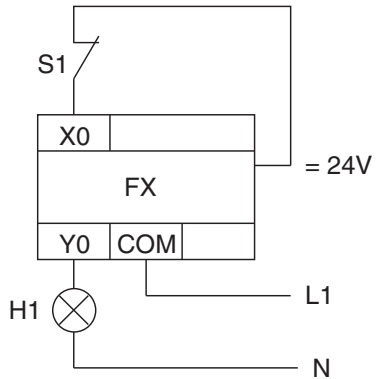
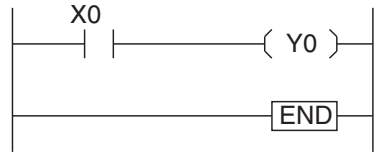
Tab. 4-6: Beispiel für einen betätigten Schließer



ACHTUNG:

Verwenden Sie als Befehlsgeber zum Einschalten von Betriebszuständen immer Schließer, damit ein Drahtbruch nicht zum unbeabsichtigtem Einschalten führt.

Nichtbetätigter Öffner

Beispiel	Stromlaufplan
<p>Bei Nichtbetätigen des Öffners S1 soll der Melder H1 leuchten.</p>	
Zuordnungsliste	Beschaltung der SPS
<p>Öffner: S1 X0 Melder: H1 Y0</p>	
Anweisungsliste	Kontaktplan
<pre> 0 LD X0 1 OUT Y0 2 END </pre>	
Anmerkung	
<p>Der Ausgang Y0 führt das Signal „1“, wenn an Eingang X0 das „1“-Signal ansteht. Der Öffner S1 wird im Programm auf den Signalzustand „1“ abgefragt, damit der Ausgang Y0 bei einer Betätigung des Öffners S1 ein „0“-Signal aufweist.</p>	

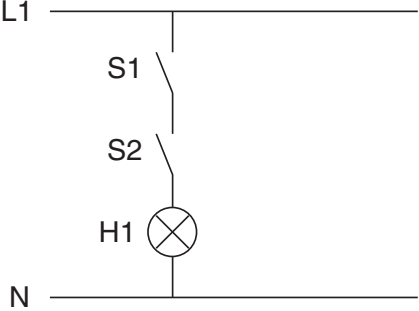
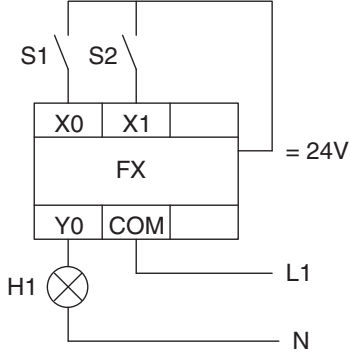
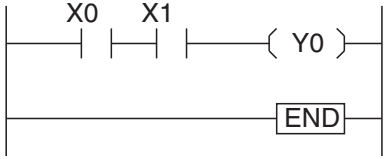
Tab. 4-7: Beispiel für einen nichtbetätigten Öffner



ACHTUNG:

Verwenden Sie zum Ausschalten von Betriebszuständen als Befehlsgeber immer Öffner, damit bei einem Drahtbruch ein Ausschalten gewährleistet ist.

Reihenschaltung

Beispiel	Stromlaufplan
<p>Bei Betätigen des Schließers S1 UND des Schließers S2 soll der Melder H1 leuchten.</p>	
Zuordnungsliste	Beschaltung der SPS
<p>Schließer: S1 X0 Schließer: S2 X1 Melder: H1 Y0</p>	
Anweisungsliste	Kontaktplan
<p>0 LD X0 1 AND X1 2 OUT Y0 3 END</p>	
Anmerkung	
<p>Der Ausgang Y0 führt das Signal „1“, wenn der Eingang X0 und der Eingang X1 ein „1“-Signal aufweisen.</p>	

Tab. 4-8: Beispiel für eine UND-Verknüpfung

Parallelschaltung

Beispiel	Stromlaufplan
<p>Bei Betätigen des Schließers S1 ODER des Schließers S2 soll der Melder H1 leuchten.</p>	
Zuordnungsliste	Beschaltung der SPS
<p>Schließer: S1 X0 Schließer: S2 X1 Melder: H1 Y0</p>	
Anweisungsliste	Kontaktplan
<p>0 LD X0 1 OR X1 2 OUT Y0 3 END</p>	
Anmerkung	
<p>Der Ausgang Y0 führt das Signal „1“, wenn mindestens ein Eingang X0 oder X1 den Signalzustand „1“ aufweist.</p>	

Tab. 4-9: Beispiel für eine ODER-Verknüpfung

Selbsthaltung (I)

Setzen eines Ausgangs mit Selbsthaltung

<p>Beispiel</p> <p>Bei Betätigen des Schließers S1 soll der Melder H1 leuchten, auch wenn der Schließer S1 nicht mehr betätigt wird. Nach kurzzeitigen Betätigen des Öffners S2 soll der Melder H1 nicht mehr leuchten.</p>	<p>Stromlaufplan</p>
<p>Zuordnungsliste</p> <p>Schließer: S1 X0 Öffner: S2 X1 Melder: H1 Y0</p>	<p>Beschaltung der SPS</p>
<p>Anweisungsliste</p> <pre> 0 LD X0 1 OR Y0 2 AND X1 3 OUT Y0 4 END </pre>	<p>Kontaktplan</p>
<p>Anmerkung</p> <p>Der Ausgang Y0 wird eingeschaltet (Signalzustand „1“), wenn der Eingang X0 kurzzeitig durchgeschaltet wird (Schließer S1 betätigt). Der Ausgang Y0 wird ausgeschaltet (Signalzustand „0“), wenn der Eingang X1 kurzzeitig betätigt wird (Öffner S2 betätigt).</p>	

Tab. 4-10: Beispiel zum Setzen und Rücksetzen eines Ausgangs mit Selbsthaltung

Selbsthaltung (II)

Setzen und Rücksetzen eines Ausgangs mit SET-/RST-Anweisung

Beispiel	Stromlaufplan
<p>Bei Betätigen des Schließers S1 soll der Melder H1 leuchten, auch wenn der Schließer S1 nicht mehr betätigt wird. Nach kurzzeitigen Betätigen des Öffners S2 soll der Melder H1 nicht mehr leuchten.</p>	
Zuordnungsliste	Beschaltung der SPS
<p>Schließer: S1 X0 Öffner: S2 X1 Melder: H1 Y0</p>	
Anweisungsliste	Kontaktplan
<pre> 0 LD X0 1 SET Y0 2 LDI X1 3 RST Y0 4 END </pre>	
Anmerkung	
<p>Der Ausgang Y0 wird eingeschaltet (Signalzustand „1“), wenn der Eingang X0 kurzzeitig durchgeschaltet wird (Schließer S1 betätigt). Der Ausgang Y0 wird ausgeschaltet (Signalzustand „0“), wenn der Eingang X1 kurzzeitig betätigt wird (Öffner S2 betätigt).</p>	

Tab. 4-11: Beispiel zum Setzen (SET) und Rücksetzen (RST) eines Ausgangs

Einsatz eines Timers zur Einschaltverzögerung

Beispiel	Zuordnungsliste
Bei Betätigen des Schließers S1 soll der Melder H1 nach $t = 5\text{ s}$ leuchten.	Schließer: S1 X0 Melder: H1 Y0 Timer: T0 100 ms
Zeitdiagramm	Beschaltung der SPS
Anweisungsliste	Kontaktplan
<pre> 0 LD X0 1 OUT T0 K50 4 LD T0 5 OUT Y0 6 END </pre>	
Anmerkung	
<p>Wenn der Eingang X0 den Signalzustand „1“ aufweist, beginnt die Zeit abzulaufen. Nach Ablauf der programmierten Zeit $t = 5\text{ s}$ wird der Ausgang Y0 auf den Signalzustand „1“ geschaltet. Der Timer T0 fällt in den Ruhezustand „0“ zurück, sobald der Eingang X0 den Signalzustand „0“ aufweist.</p>	

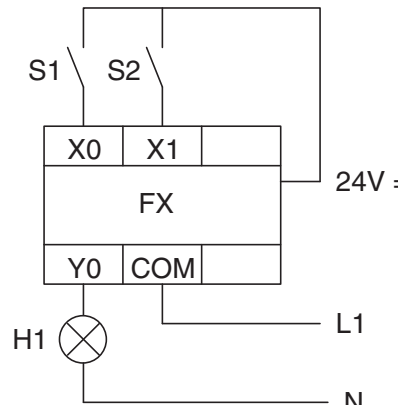
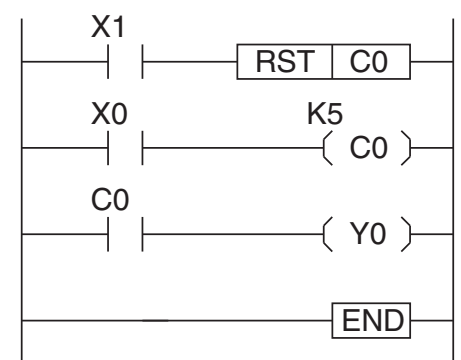
Tab. 4-12: Beispiel zum Einsatz eines Timers für eine Einschaltverzögerung

Einsatz eines Timers zur Ausschaltverzögerung

Beispiel	Zuordnungsliste
Bei Betätigen des Schließers S1 soll der Melder H1 sofort leuchten. Der Melder H1 soll jedoch $t = 5\text{ s}$ länger leuchten, als der Schließer S1 betätigt wird.	Schließer: S1 X0 Melder: H1 Y0 Timer: T0 100 ms
Zeitdiagramm	Beschaltung der SPS
Anweisungsliste	Kontaktplan
<pre> 0 LD X0 1 OR Y0 2 ANI T0 3 OUT Y0 4 LD Y0 5 ANI X0 5 OUT T0 K50 9 END </pre>	
Anmerkung	
Wenn der Eingang X0 den Signalzustand „1“ aufweist, schaltet der Ausgang Y0 in „Selbsthaltung“. Sobald der Eingang X0 auf das „0“-Signal zurückfällt, läuft die programmierte Zeit. Nach Ablauf der eingestellten Zeit $t = 5\text{ s}$ wird der Ausgang Y0 auf den Signalzustand „0“ zurückgesetzt.	

Tab. 4-13: Beispiel für den Einsatz eines Timers zur Ausschaltverzögerung

Einsatz eines Aufwärtszählers

<p>Beispiel</p> <p>Nach 5 Betätigungen des Schließers S1 soll der Melder H1 aufleuchten. Mit dem Schließer S2 soll der Zähler wieder auf den Anfangszustand zurückgesetzt werden und der Melder H1 verlöschen.</p>	<p>—</p> <p>—</p>
<p>Zuordnungsliste</p> <p>Schließer: S1 X0 Schließer: S2 X1 Melder: H1 Y0 Zähler: C0</p>	<p>Beschaltung der SPS</p> 
<p>Anweisungsliste</p> <pre> 0 LD X1 1 RST C0 3 LD X0 4 OUT C0 K5 7 LD C0 8 OUT Y0 9 END </pre>	<p>Kontaktplan</p> 
<p>Anmerkung</p> <p>Bei jedem Ansteuern des Zählers C0 mit einem „1“-Signal wird der Zähleristwert um den Wert 1 erhöht. Nachdem der Zähleristwert den Zählersollwert 5 erreicht hat, schaltet der Zähler den Ausgang Y0 auf den Signalzustand „1“. Durch ein „1“-Signal am Eingang X1 wird der Zähler wieder auf den Signalzustand „0“ zurückgesetzt.</p>	

Tab. 4-14: Programmierbeispiel zum Einsatz eines Aufwärtszählers

5 STL-Anweisung

5.1 Allgemeine Hinweise

Die STL-Anweisung ist eine elementare SPS-Anweisung zur einheitlichen Programmierung von Steuerungsabläufen. Die STL-Anweisung wird in Verbindung mit einem Schrittstatus eingesetzt und ermöglicht die komfortable Programmierung von Schrittsteuerungen.

Aufwendige Programme für einfache Start-/Stopp-Sequenzen entfallen bei der Programmierung, so dass auch der Programmieranfänger die Steuerung effektiv nutzen kann. Sie können damit den Programmieraufwand für derartige Sequenzen erheblich einschränken.

Die STL-Anweisung wird in Zusammenhang mit dem Schrittstatusoperanden S programmiert. Je nach verwendetem CPU-Typ stehen bis zu 1000 Schrittstatusoperanden im Bereich von S0 bis S999 zur Verfügung, wobei den Operanden S0 bis S9 feste Funktionen zugeordnet sind.

Bedeutung der Schrittstatusoperanden S0 bis S9

Die Schrittstatusoperanden S0 bis S9 sind Initialisierungsoperanden, mit denen verschiedene Schrittabläufe innerhalb eines STL-Programms erstellt werden können, um z. B. unterschiedliche Betriebsabläufe (Hand-/Automatikbetrieb, Nullpunktfahrt usw.) zu realisieren. Dies gilt insbesondere, wenn die Applikationsanweisung IST genutzt wird.

Wenn keine gesonderten Abläufe für Hand-, Automatikbetrieb und Nullpunktfahrt vorgesehen sind, können die Operanden S0 bis S9 als „normale“ Operanden ohne Sonderfunktionen genutzt werden.

5.1.1 Anwendungsbeispiel zum Einsatz der STL-Anweisung

Die herkömmliche Programmiermethode mittels Kontaktplan besteht darin, dass ein Ausgang von einem bestimmten, ihm zugeordneten Eingangskontakt (z. B. externer mechanischer Schalter) angesprochen wird und dieser Eingangskontakt parallel oder in Reihe zur Steuerungsaufgabe liegt. Zur Sicherung des Arbeitsprozesses gegen ungewollte Steuerungsabläufe und damit zusammenhängende Fehlfunktionen muss ein solches Programm umfangreiche Verriegelungsmaßnahmen enthalten.

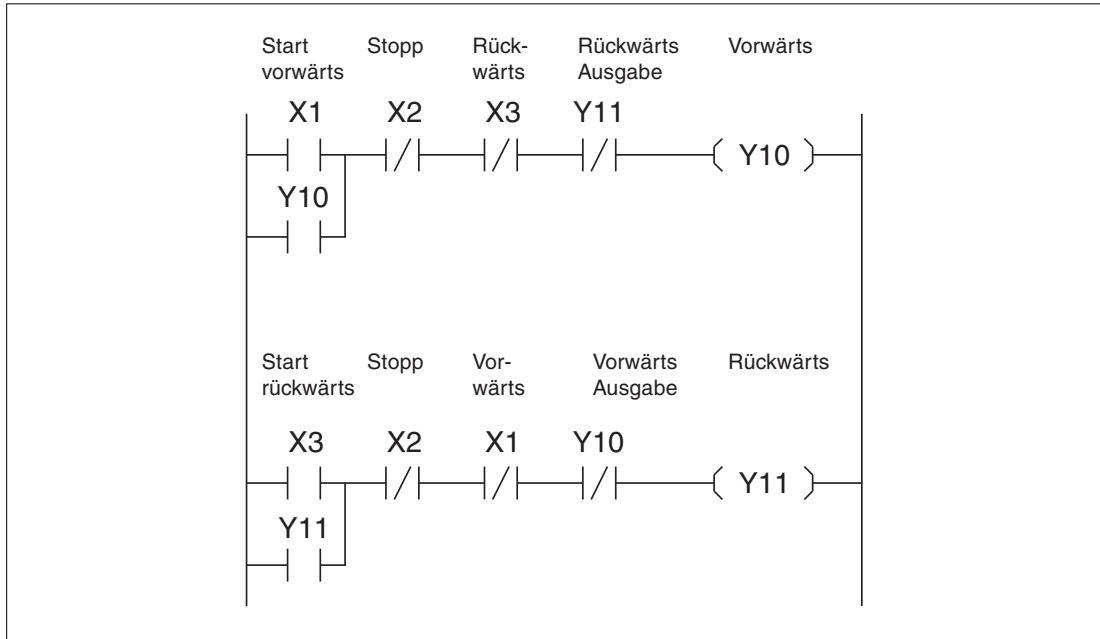


Abb. 5-2: Anwendungsbeispiel mit Verriegelungskontakten

Bei der Verwendung von Schrittsteueranweisungen können die gezeigten Verriegelungskontakte entfallen, da die Steuersignale, wie z. B. „aufwärts“, „abwärts“ usw., über das Programm unter Berücksichtigung bestimmter Grenzwerte erfolgen.

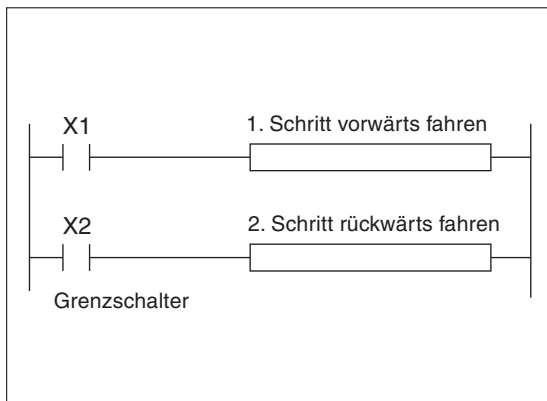


Abb. 5-1: Anwendungsbeispiel ohne Verriegelungskontakte

5.1.2 Schematischer Ablauf einer Schrittsteuerung

Anhand einer kurzen Prozessfolge wird als Beispiel eine Schrittsteuerung mit vier Arbeitsschritten beschrieben. Der vierte Schritt beendet die Schrittsteuerung.

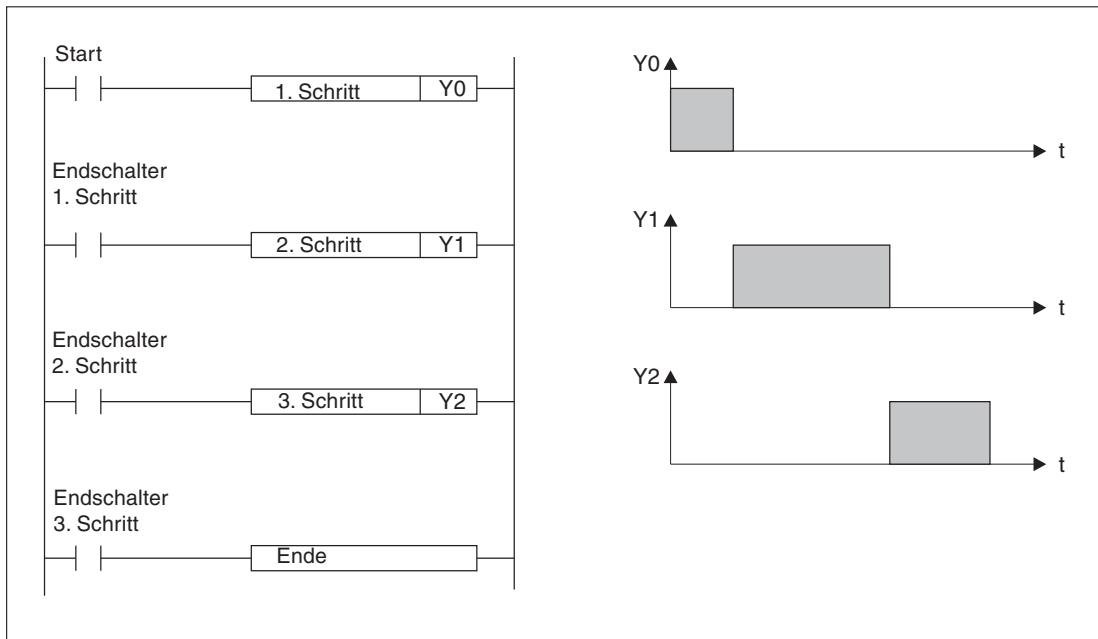


Abb. 5-3: Schematischer Ablauf einer Schrittsteuerung

In der Abb. 5-3 ist zu sehen, dass der 2. Arbeitsgang eingeschaltet wird, sobald der 1. Arbeitsschritt beendet ist und der zugehörige Endschalter eingeschaltet wird. Dies beinhaltet, dass auch alle Zustände der Operanden innerhalb des 1. Arbeitsschrittes zurückgesetzt werden.

Das Ende des 2. Arbeitsschrittes bedeutet gleichzeitig den Start des 3. Arbeitsschrittes. Mit dem Einschalten des 3. Endschalters wird das Ende der Schrittfolge (4. Arbeitsgang) erreicht.

5.1.3 Darstellung einer Ablaufsteuerung in einem Flussdiagramm

In der folgenden Abb. ist die gleiche Ablaufsteuerung in einem Flussdiagramm (IEC-Standard) dargestellt. In einem Flussdiagramm ist die Darstellung einer Ablaufsteuerung erst einmal unabhängig von der späteren Realisierung in einem SPS-Programm.

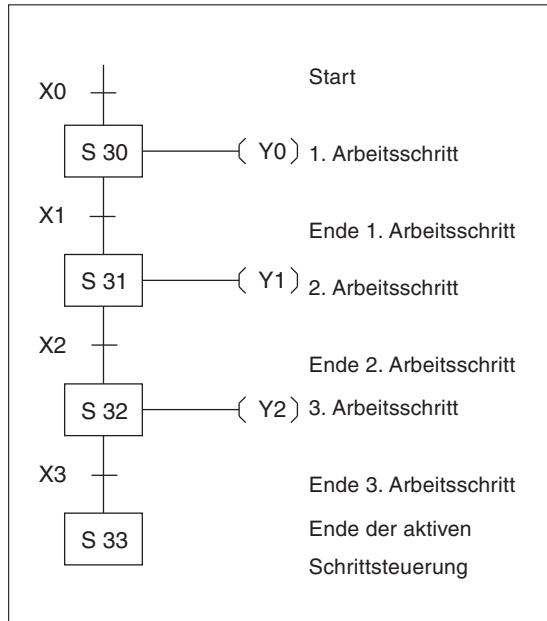


Abb. 5-4:
*Flussdiagramm des
Anwendungsbeispiels*

5.2 STL-Anweisung programmieren

	STL		RET			
	Schrittstatus aktivieren/deaktivieren					
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
	●	●	●	●	●	●
Operanden						
S0–S4095; Die Adressbereiche sind abhängig von der verwendeten MELSEC SPS (siehe Tabelle 5-1).			Puls-Anweisung (P)		Verarbeitung	Programmschritte
			16 Bit	32 Bit	STL	1
					RET	1

Steuerung	Anzahl der Operanden	Adressbereich
FX1S	128	S0 bis S127
FX1N	1000	S0 bis S999
FX2N		
FX2NC		
FX3G	4096	S0 bis S4095
FX3GC		
FX3GE		
FX3S	256	S0 bis S255
FX3U	4096	S0 bis S4095
FX3UC		

Tab. 5-1: Schrittmerker

Funktion

Programmierung von Ablaufsteuerungen

Beschreibung

- Die STL-Anweisung wird zusammen mit dem Schrittstatusoperanden S eingesetzt. Der Schrittstatusoperand S kann mit den folgenden Anweisungen des Grundbefehlssatzes programmiert werden: LD, LDI, AND, ANI, OR, ORI, OUT, SET, RST.
- Bei der FX2N-, der FX2NC-, der FX3G-, der FX3GC-, der FX3GE-, der FX3S-, der FX3U- und der FX3UC-Serie ist die Verwendung von gepulsten Anweisungen (LDP, LDF etc.) und gepulsten Merkern (M2800–M3071) möglich.
- In einem Programm ohne Schrittsteuerung können die Schrittstatusoperanden S auch als herkömmliche Merker verwendet werden.
- Am Anfang eines STL-Programmbereiches (Schrittstatus) muss jeder einzelne Schrittstatusoperand mit der SET-Anweisung gesetzt werden.
- Innerhalb des Kontaktplans tritt der STL-Kontakt an der linken Sammelschiene auf und kann daher als „Hauptkontakt“ angesehen werden.
- Der einer STL-Anweisung folgende Strompfad kann erst bearbeitet werden, wenn der STL-Kontakt gesetzt ist.
- Sobald der STL-Kontakt zurückgesetzt ist, kann der nachstehende Strompfad nicht mehr bearbeitet werden.
- Mit der RET-Anweisung wird der gesamte STL-Programmbereich (Schrittstatus) beendet.

HINWEISE

- | Ein Schrittstatusoperand darf nur einmal pro Programm mit einer STL-Anweisung programmiert werden.
- | Die STL-Anweisung darf nicht in einem Interrupt-Programm eingesetzt werden.
- | Verwenden Sie keine Sprunganweisungen innerhalb eines Schrittstatus.
- | Jede Schrittsteuerung muss mit der RET-Anweisung beendet werden.
- | Der zuletzt aktivierte Schrittstatusoperand sollte mit einer RST-Anweisung zurückgesetzt werden, oder es muss eine Weiterschaltung zurück an den Anfang der Schrittkette erfolgen.

Zulässige Anweisungen innerhalb eines Schrittstatus

In der folgenden Tabelle sind die Anweisungen des Grundbefehlssatzes aufgeführt, die zwischen STL-Anweisungen bzw. zwischen einer STL- und einer RET-Anweisung eingesetzt werden dürfen.

Status		Anweisungen		
		LD, LDI, OUT, NOP, AND, ANI, SET, RST, OR, ORI, PLS, PLF	ANB, ORB, MPS, MRD, MPP	MC, MCR
Initialisierungsstatus		zulässig	zulässig	nicht zulässig
Programmverzweigung	Ausgänge	zulässig	zulässig	nicht zulässig
	Weiterschaltbedingung	zulässig	nicht zulässig	nicht zulässig

Tab. 5-2: Zulässige Anweisungen innerhalb eines Schrittstatus

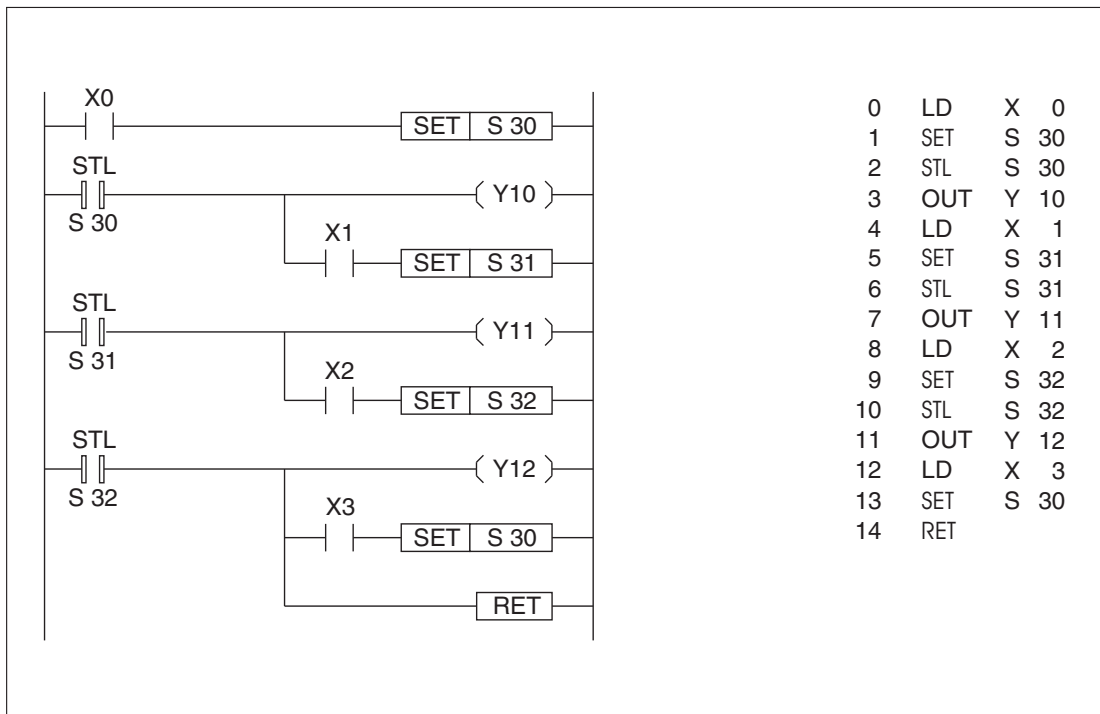


Abb. 5-5: Programmierbeispiel zum Einsatz der Anweisungen STL und RET

Ausgänge mehrfach belegen

Der gleiche Ausgang kann mit verschiedenen STL-Anweisungen bzw. Schrittstatusoperanden angesprochen werden.

Beispiel ▾

Ausgänge mehrfach belegen

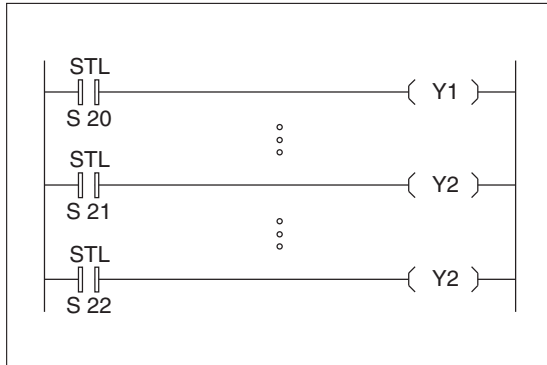


Abb. 5-6:

Mit STL-Anweisungen lassen sich Ausgänge mehrfach belegen

Im obenstehenden Programmausschnitt wird der gleiche Ausgang (Y2) über verschiedene STL-Anweisungen bzw. Schrittstatusoperanden (S21 und S22) angesprochen.

Y2 wird eingeschaltet, wenn S21 oder S22 aktiv ist. Y2 wird ausgeschaltet, wenn S21 und S22 nicht aktiv sind. Die Doppelbelegung ist in diesem Fall unproblematisch, da die Schritte 21 und 22 nicht gleichzeitig aktiv sein können. △

Rücksetzfunktion der Weiterschaltbedingung

Sobald der Status S durch die STL-Anweisung gesetzt ist, wird die Weiterschaltbedingung des vorangegangenen Status zurückgesetzt. Das bedeutet, dass in einem Programmzyklus der aktuelle wie auch der nachfolgende Status für eine sehr kurze Zeit gleichzeitig gesetzt sein können.

HINWEIS

Wenn die aufeinanderfolgenden Operanden nicht gleichzeitig aktiv sein dürfen, empfiehlt es sich, die Operanden durch eine Verriegelung zu schützen.

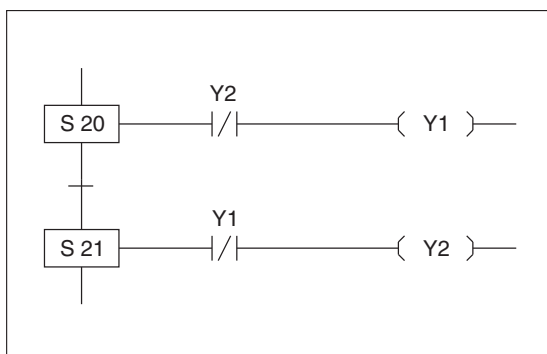


Abb. 5-7:

Verriegelungsmaßnahme, um einen gleichzeitigen Einschaltzustand zu verhindern

Timer mehrfach belegen

In einem Programm kann ein Timer durch Einsatz von Schrittstatusoperanden mehrfach belegt werden. Ein Timer darf jedoch nicht in zwei aufeinanderfolgenden Schritten eingesetzt werden.

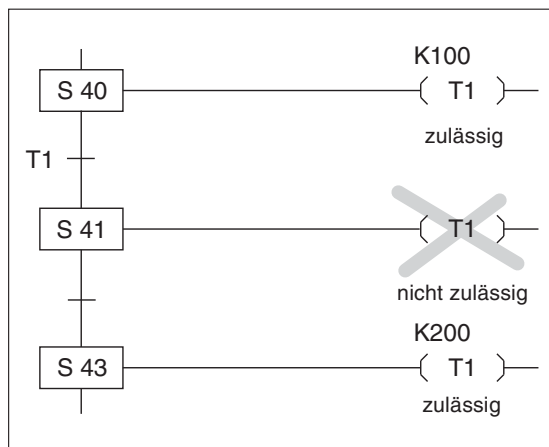


Abb. 5-8:
Timer mehrfach belegen

Weiterschaltbedingung durch ein Impulssignal

Aufeinanderfolgende Schritte können über dieselbe Weiterschaltbedingung aktiviert werden. Hierzu ist der Einsatz einer Impulsanweisung (PLS-Anweisung) erforderlich.

Beispiel ▾

Weiterschaltbedingung durch ein Impulssignal

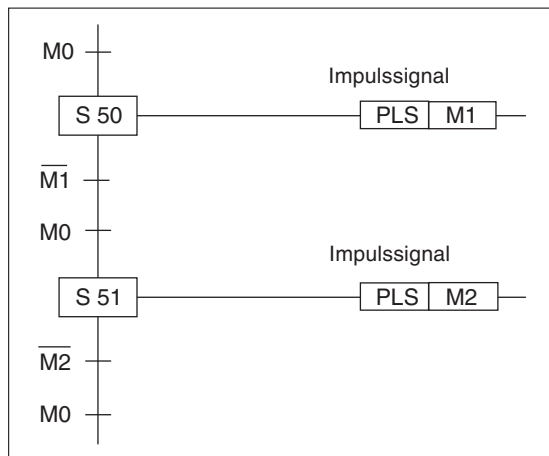


Abb. 5-9:
Weiterschaltbedingung durch ein Impulssignal

Das erste M0-Signal aktiviert den Schrittstatus S50 und schaltet M1 ein. M1 verhindert das unmittelbare Aktivieren des nächsten Schrittstatus. S51 wird erst aktiv, wenn das nächste M0-Signal ansteht. △

Weiterschaltbedingung durch gepulste Anweisungen

Bei Steuerungen der FX2N-, der FX2NC-, der FX3G-, der FX3GC-, der FX3GE-, der FX3S-, der FX3U- und der FX3UC-Serie können die Weiterschaltbedingungen durch die gepulsten Anweisungen (LDP, LDF, ANP etc.) und die gepulsten Merker M2800 bis M3071 realisiert werden.

Beispiel ▾

Weiterschaltbedingung durch Verwendung des gepulsten Merkers M2800:

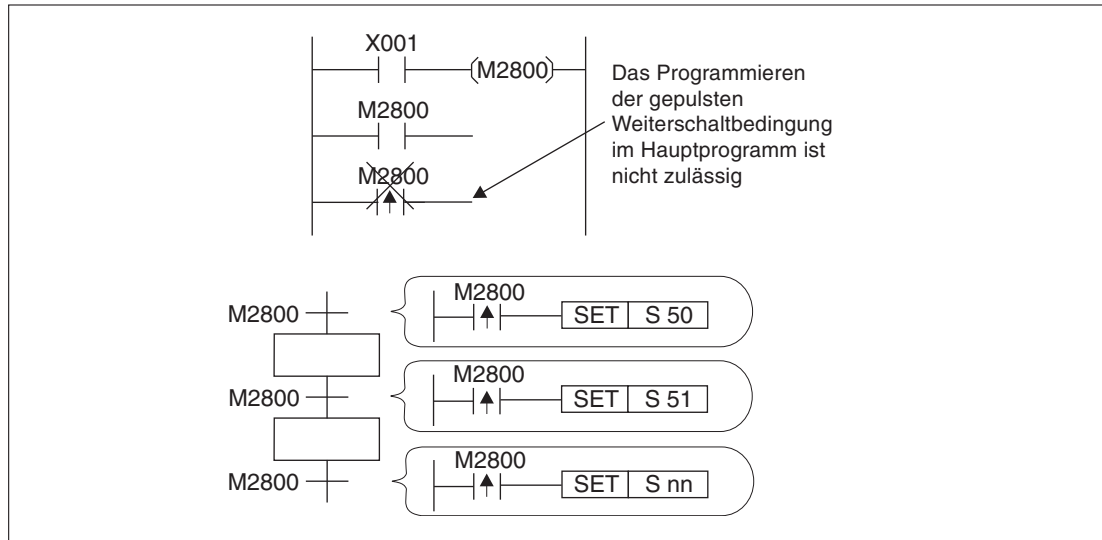


Abb. 5-10 : Verwendung des gepulsten Merkers M2800

Durch Setzen des Merkers M2800 mit X001 wird der Schritt S51 aktiviert. Eine gleichzeitige Aktivierung des Schrittes Snn ist nicht möglich, da M2800 (gepulst) zum 2. Mal programmiert wurde. Beim nächsten Setzen von M2800 durch X001 wird der Schritt Snn aktiviert, da der Schritt S50 inaktiv ist, und der gepulste Merker M2800 somit nur einmal aktiv vorhanden ist.

△

5.3 Schrittstatus initialisieren

Jeder Schrittstatus erfordert eine Initialisierung. Dafür stehen beispielsweise die Initialisierungsoperanden S0 bis S9 zur Verfügung. Mit Hilfe der Initialisierungsoperanden können verschiedene Schrittabläufe innerhalb eines STL-Programms erstellt werden, um z. B. unterschiedliche Betriebsabläufe (Hand-/Automatikbetrieb, Nullpunktfahrt usw.) zu realisieren.

Beispiel ▾

Schrittstatus initialisieren

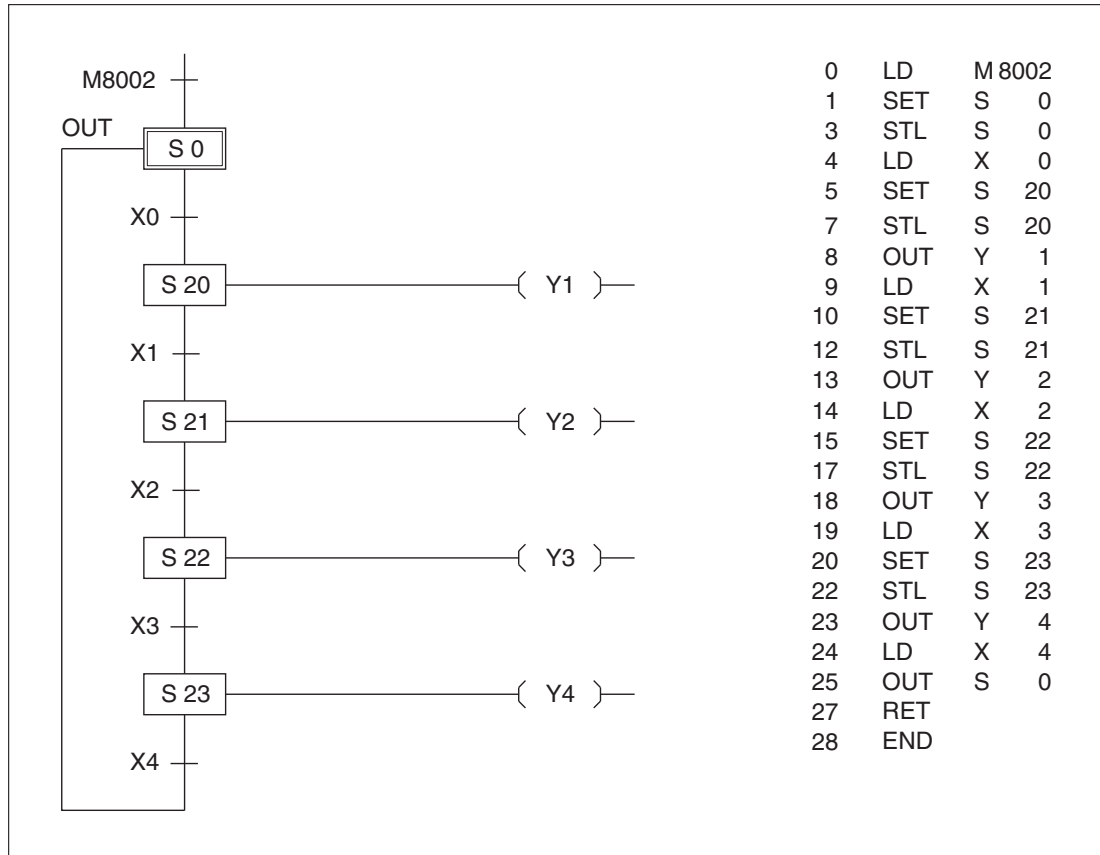


Abb. 5-11: Programmierbeispiel zum Initialisieren eines Schrittstatus

Der Merker M8002 bewirkt beim Einschalten der SPS einen definierten Systemzustand (siehe auch Abschnitt 9.1.1). Es findet eine Initialisierung der Schrittkette statt, indem S0 gesetzt wird.

Die Schrittbedingungen für jeden weiteren Schrittstatus werden in bereits beschriebener Art und Weise ausgeführt.

Um einen Neustart oder eine Wiederholung der Schrittkette zu bewirken, muss zunächst wieder S0 eingeschaltet werden. △

5.4 STL-Verzweigungen

Die speicherprogrammierbaren Steuerungen der FX-Familie können verschiedene, voneinander unabhängige Statusverläufe und Verzweigungen verarbeiten. Es wird unterschieden zwischen:

- Einfachverlauf
- selektive Verzweigung
- parallele Verzweigung
- Sprungverzweigung

5.4.1 Einfachverlauf

Beim Einfachverlauf wird der Schrittstatusverlauf sequentiell (nacheinander) abgearbeitet. Die Reihenfolge der Abarbeitung richtet sich nur nach der Stellung des Schrittstatus im Einfachverlauf und ist dadurch unabhängig von der Schrittstatusadresse.

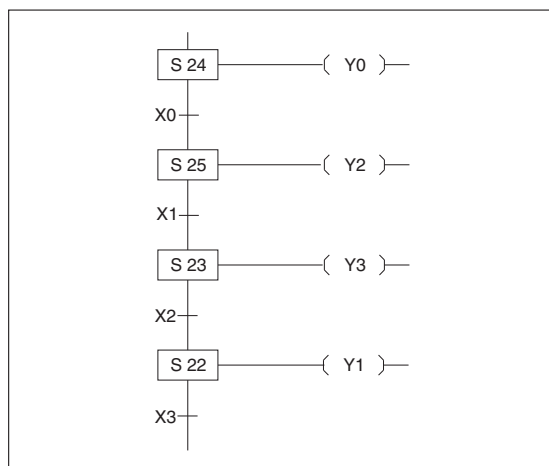


Abb. 5-12:
Beispiel für einen Einfachverlauf

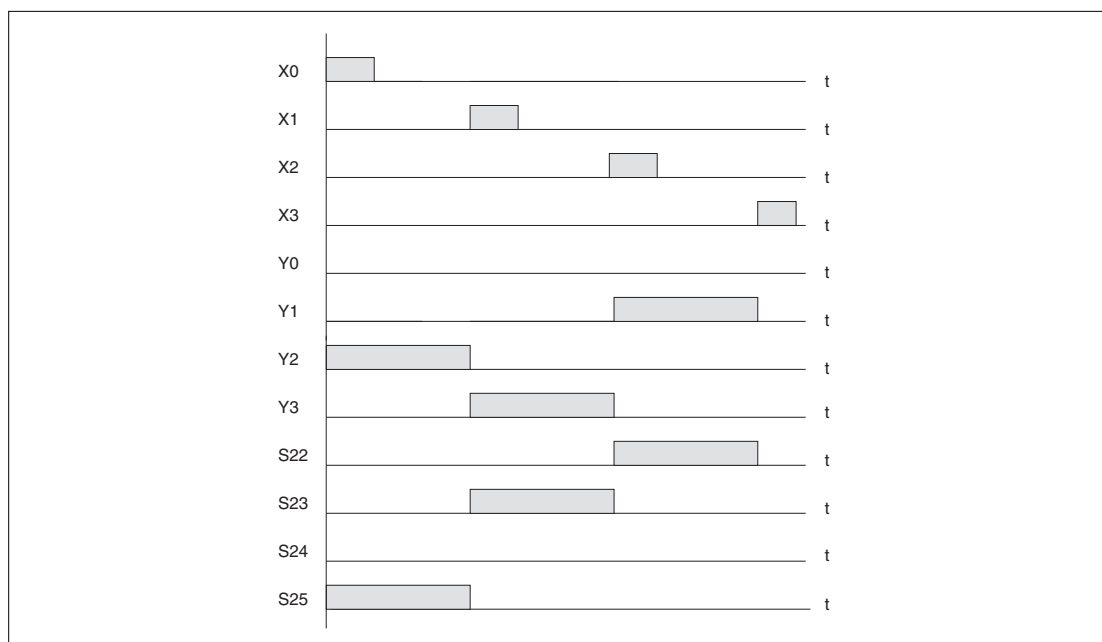


Abb. 5-13: Zeitdiagramm des Einfachverlaufs

5.4.2 Selektive Verzweigung

Bei der selektiven Verzweigung besteht die Möglichkeit, eine **Auswahl** zwischen zwei oder mehreren Statusverläufen während einer Operation zu treffen.

Aus einem Schrittstatus heraus findet eine Verzweigung in mehrere (maximal 8) Statusverläufe statt.

In Abhängigkeit von der jeweils gesetzten Eingangsbedingung erfolgt die Auswahl, welcher Statusverlauf im Programm aktiviert werden soll. Es kann nur jeweils ein Pfad aktiv sein.

HINWEIS

Es können maximal 8 Verzweigungen ausgehend von einem Schrittoperanden programmiert werden. Die Gesamtzahl aller selektiven Verzweigungen darf 16 nicht überschreiten.

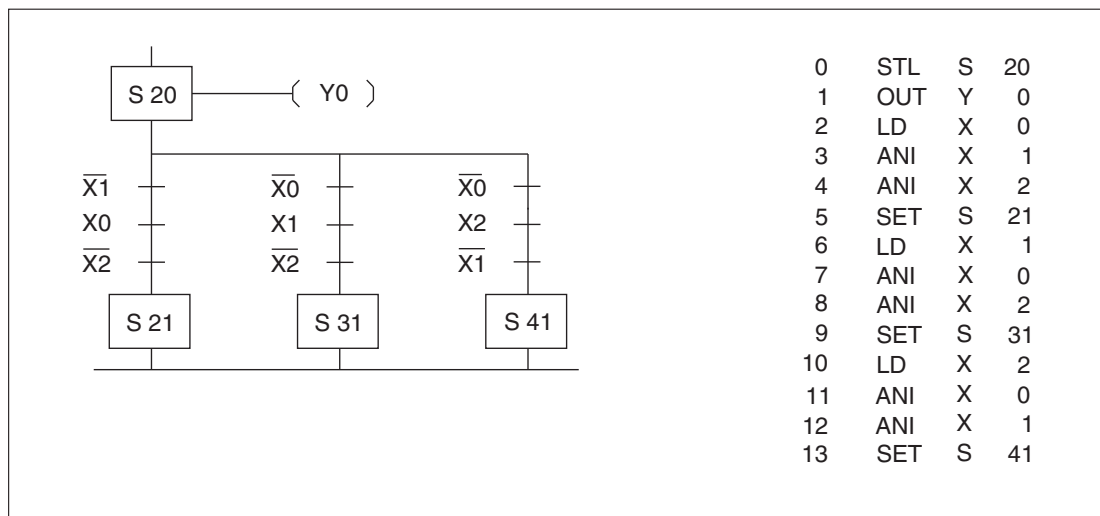


Abb. 5-14: Start einer selektiven Verzweigung

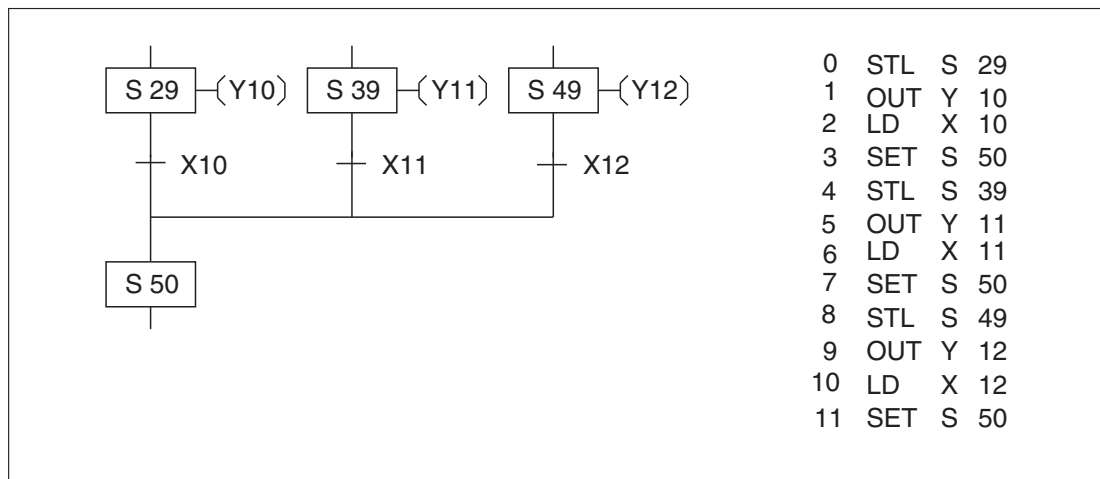


Abb. 5-15: Zusammenführen einer selektiven Verzweigung

Beispiel ▾ Flussdiagramm, Kontaktplan und Anweisungsliste einer selektiven Verzweigung.

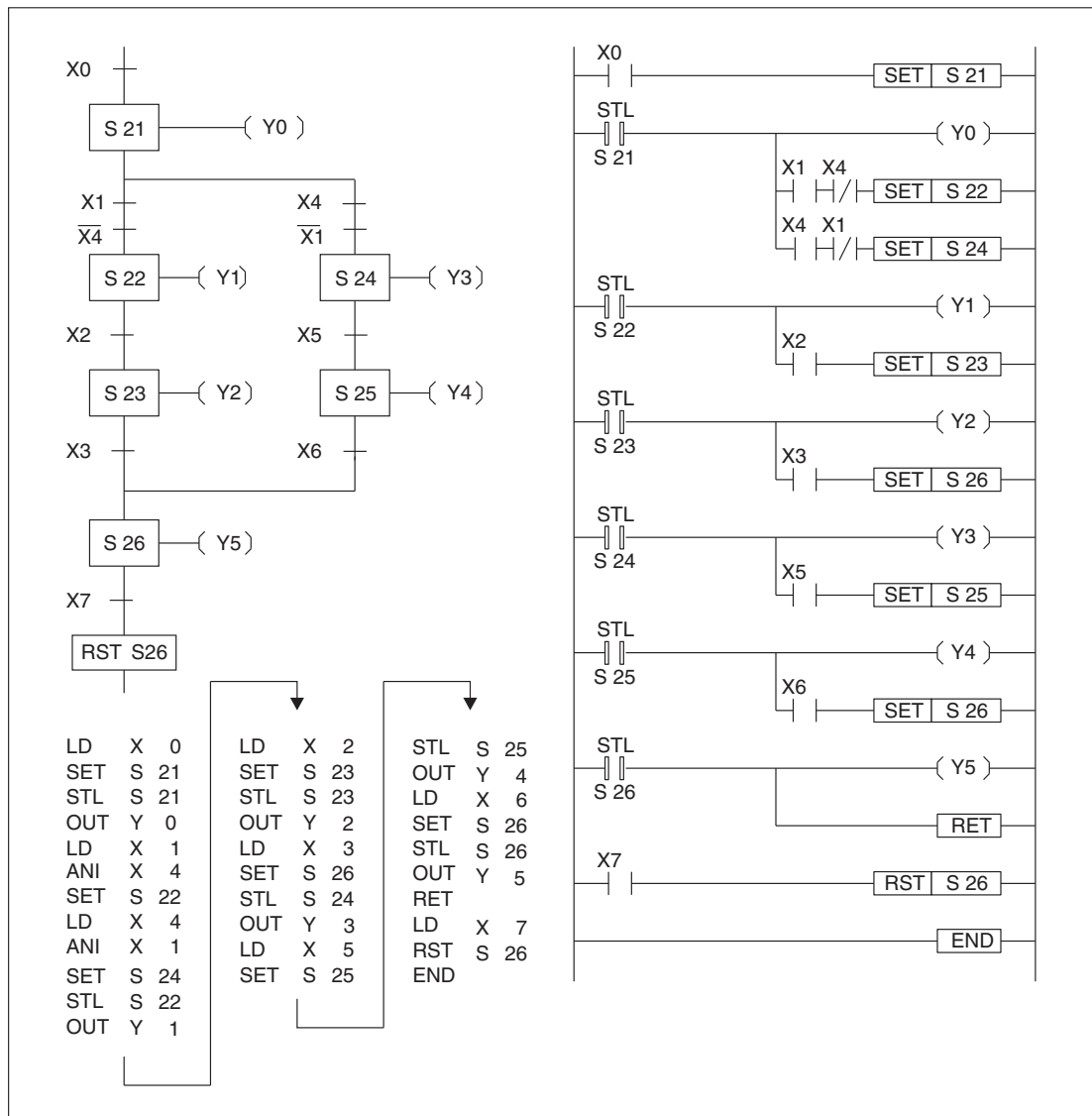


Abb. 5-16: Programmierbeispiel für eine selektive Verzweigung

Es darf hierbei immer nur eine der Funktionen ausgeführt werden. Dies ist sichergestellt, indem S21 automatisch zurückgesetzt wird, wenn entweder S22 oder S24 gesetzt ist.

S26 wird über Schritt S23 und S25 gesetzt. Dementsprechend wird beim Setzen von S26 entweder S23 oder S25 zurückgesetzt. △

5.4.3 Parallele Verzweigung

Bei der parallelen Verzweigung werden zwei oder mehr Statusverläufe **gleichzeitig abgearbeitet**. Aus einem Status heraus findet eine Verzweigung in mehrere (maximal 8) Statusverläufe statt.

HINWEIS | Die Anzahl aller Verzweigungen darf 16 nicht überschreiten.

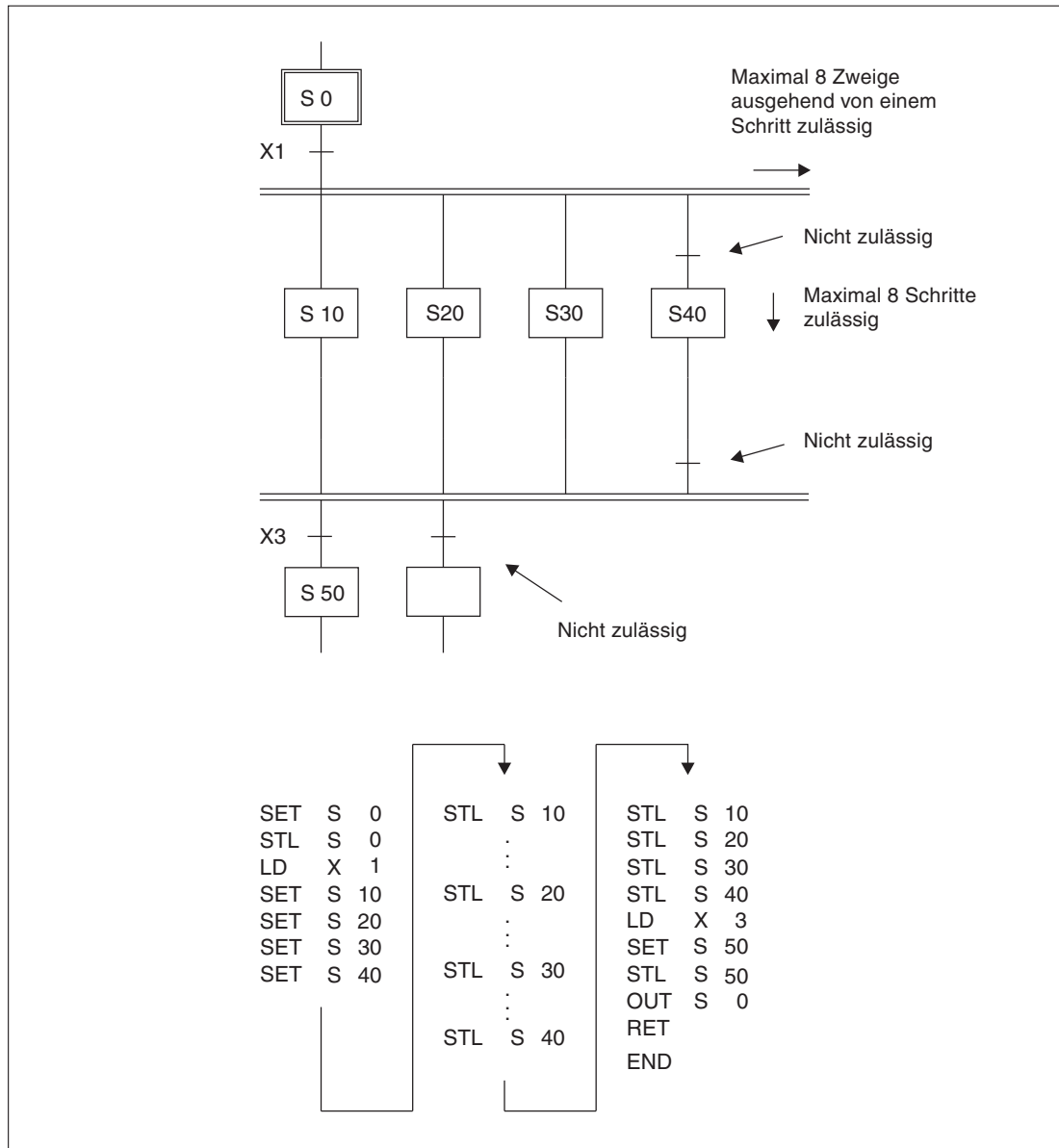


Abb. 5-17: Zulässige parallele Verzweigungen

In Abhängigkeit von der jeweils gesetzten Eingangsbedingung erfolgt die Verzweigung in die einzelnen Pfade. Im Gegensatz zur selektiven Verzweigung können bei der parallelen Verzweigung mehrere Statusverläufe gleichzeitig abgearbeitet werden.

Die geschalteten Operanden der parallelen Schritte werden erst zurückgesetzt, wenn die nach der Zusammenführung liegenden Schritte abgearbeitet werden.

HINWEISE

- | Nach der Verzweigung und vor der Zusammenführung sind keine Verknüpfungen zulässig.
- | Eine parallele Verzweigung darf maximal 8 parallele Zweige enthalten, von denen jeder Zweig aus maximal 8 aufeinanderfolgenden Schritten bestehen darf.
- | Innerhalb einer parallelen Verzweigung darf keine weitere selektive Verzweigung programmiert werden.

Beispiel ▾

Flussdiagramm, Kontaktplan und Anweisungsliste einer parallelen Verzweigung

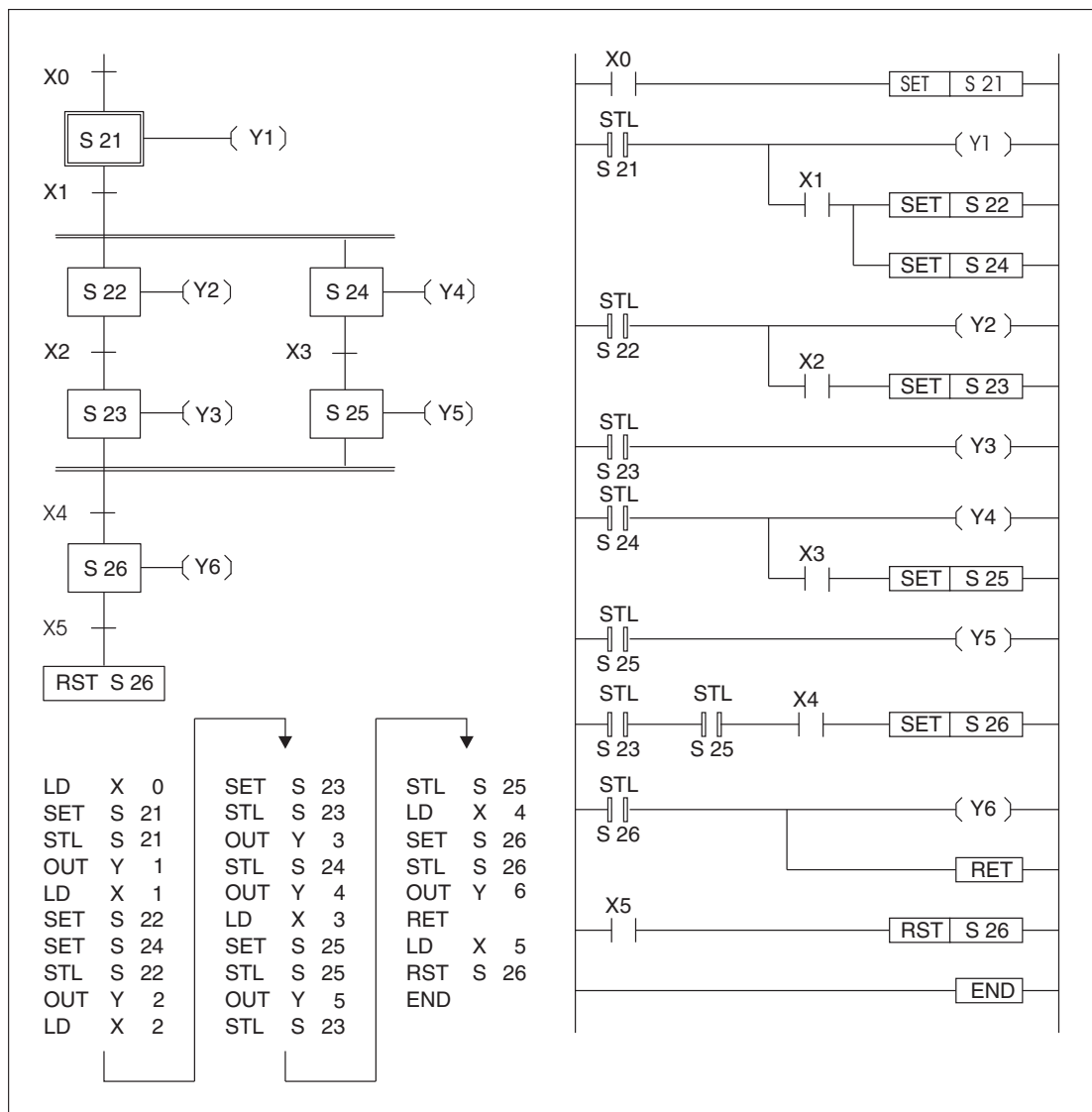


Abb. 5-18: Parallele Verzweigung

Der Schritt S26 ist in Abhängigkeit von X4 erst nach Ausführung der Schritte S23 und S25 möglich. △

5.4.4 Kombination aus selektiver und paralleler Verzweigung

Selektive und parallele Verzweigungen können in einem STL-Programm kombiniert werden.

Beispiel ▾

Kombination aus selektiver und paralleler Verzweigung

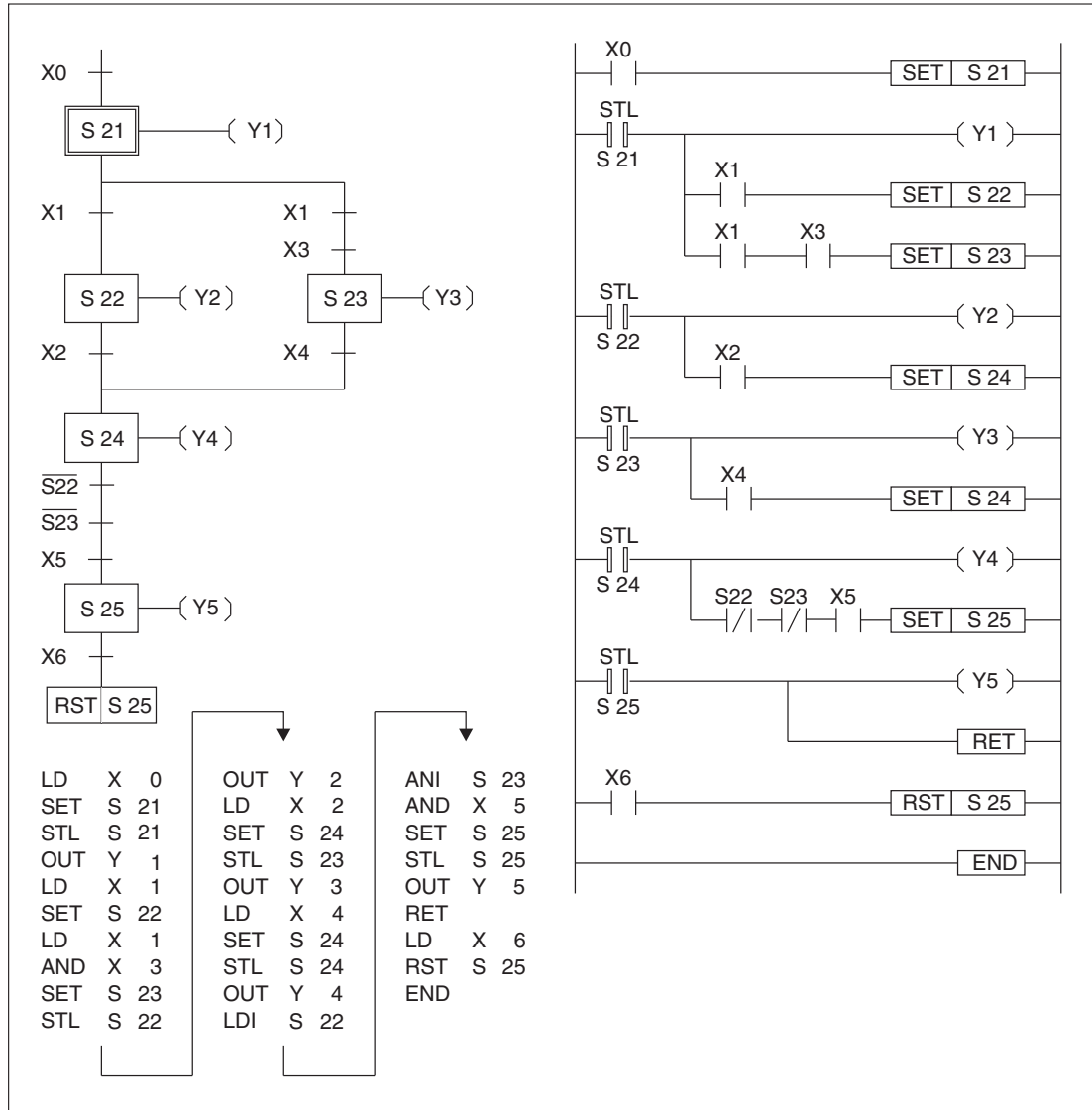


Abb. 5-19: Kombination aus selektiver und paralleler Verzweigung

Wird im Beispiel X3 gesetzt, ist die Bedingung für eine Parallelverzweigung erfüllt. Ist X3 nicht gesetzt, erfolgt eine selektive Programmbearbeitung, d.h., S24 kann nur über S22 gesetzt werden.

S25 wird nur gesetzt, wenn S22 und S23 zurückgesetzt sind. △

5.4.5 Leerstatus programmieren

Für die Realisierung einiger Schrittabläufe ist die Programmierung eines Leerstatus notwendig. Diese Möglichkeit trägt zu einer besseren Übersicht des Programmablaufs sowie zu einer Einsparung von Programmschritten bei.

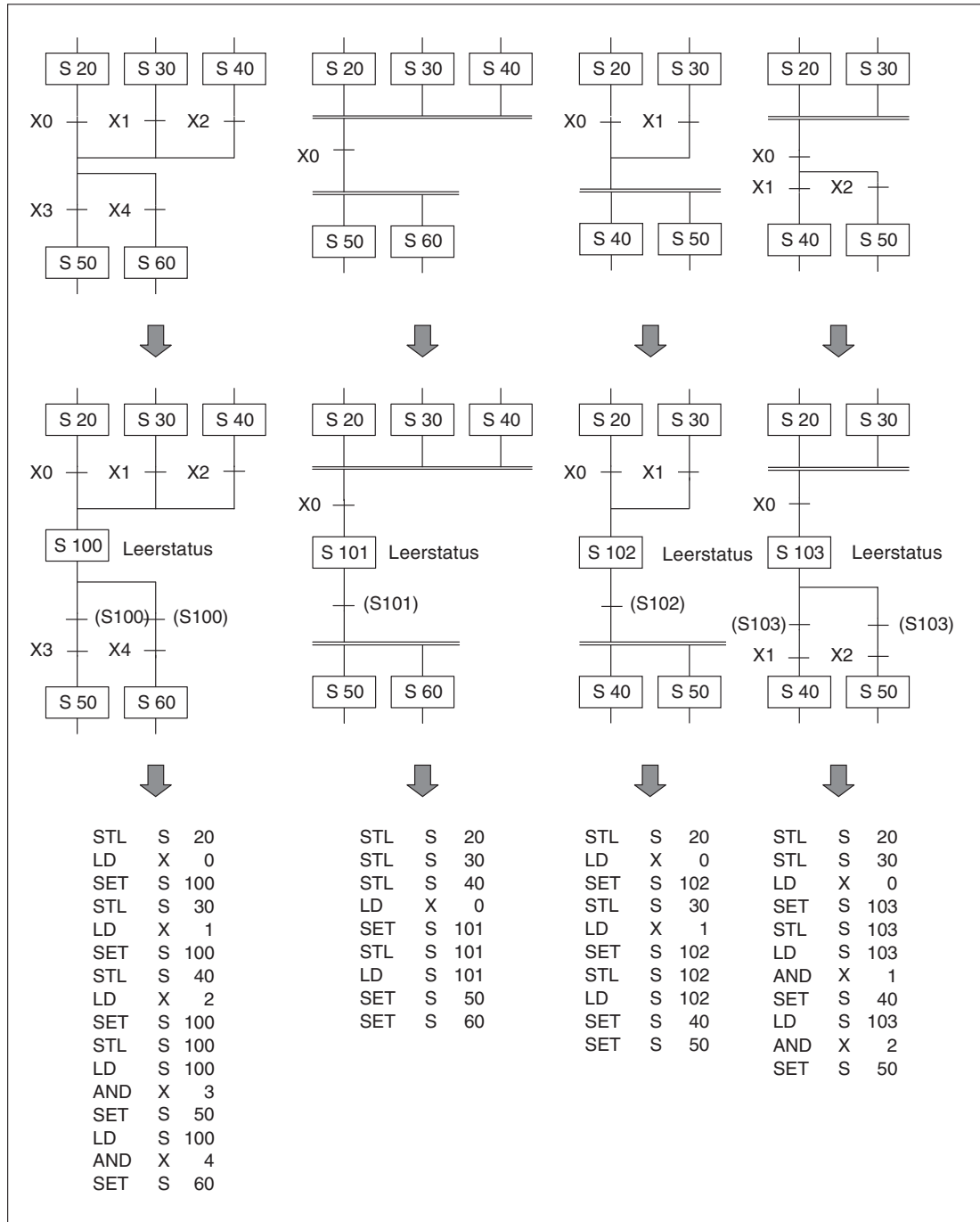


Abb. 5-20: Leerstatus programmieren

5.4.6 Sprungverzweigung

Es besteht die Möglichkeit, Teilbereiche einer Statusfolge zu überspringen oder eine Programmschleife mehrfach zu durchlaufen.

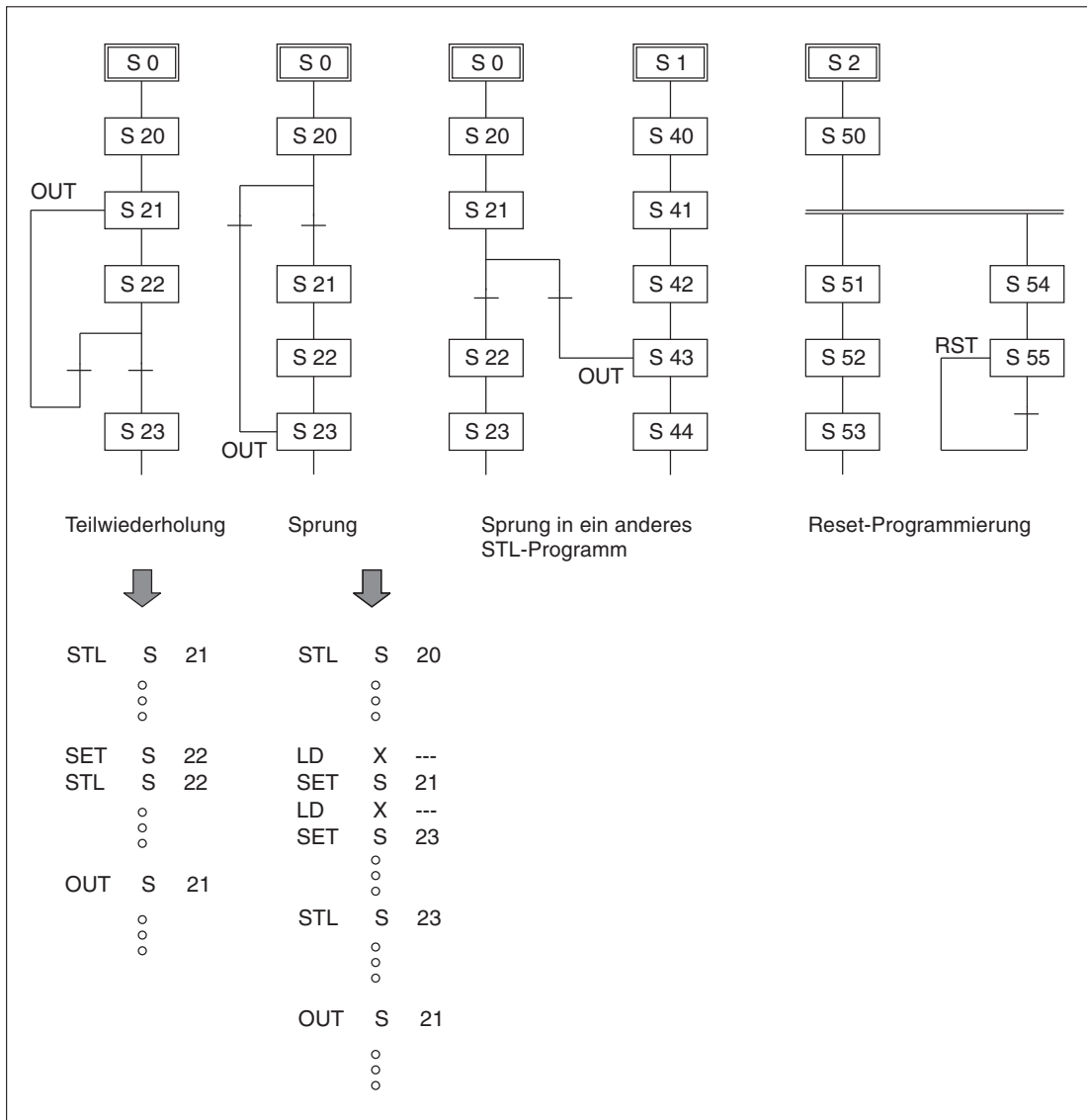


Abb. 5-21: Programmierbeispiele verschiedener Möglichkeiten einer Sprungverzweigung

Weiterschaltung in eine andere Schrittkette

Bei der Weiterschaltung von einer Schrittkette in eine andere Schrittkette kann anstatt einer SET-Anweisung auch eine OUT-Anweisung programmiert werden (siehe OUT S31 im Beispiel zur Schrittkette I). Diese Alternative hat keine Auswirkung auf die interne Programmabarbeitung der Steuerung.

Beispiel ▾

Weiterschaltung in eine andere Schrittkette

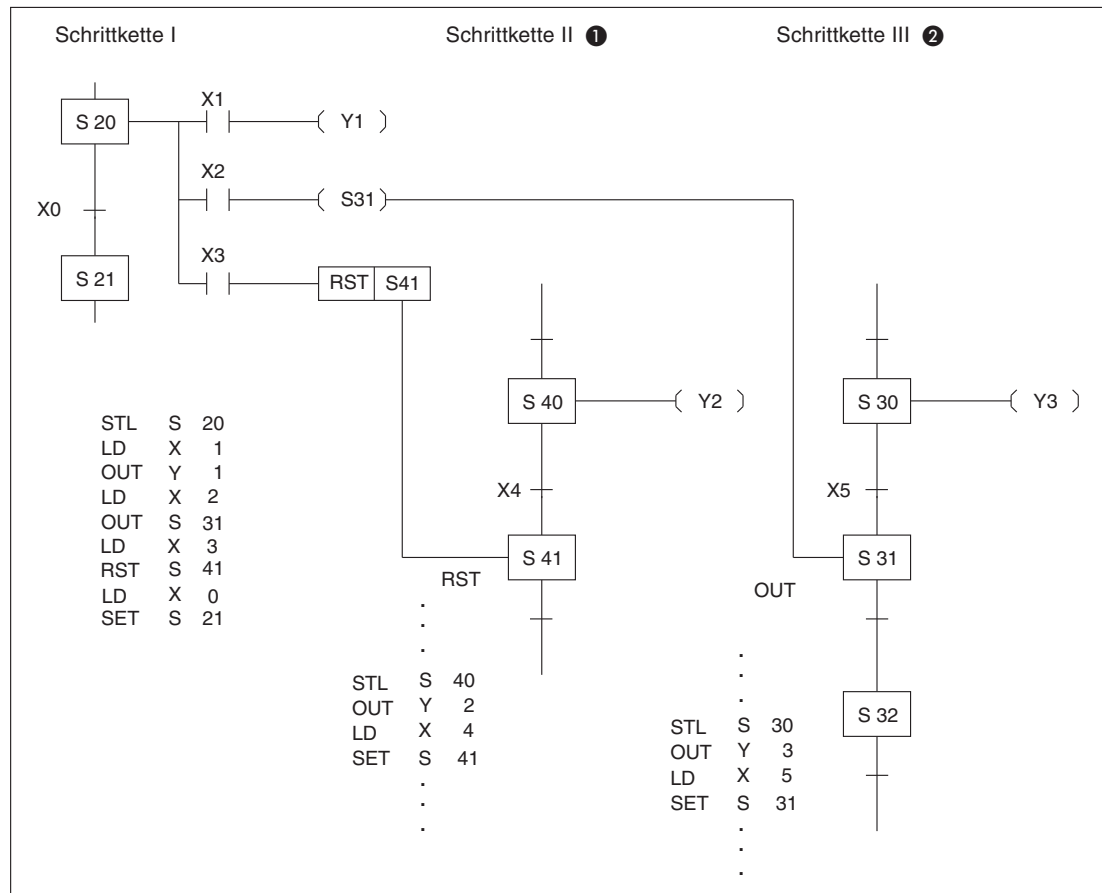


Abb. 5-22: Weiterschaltung in eine andere Schrittkette

- ① Der Schrittstatus S41 in der Schrittkette II wird über S40 und den Eingang X4 gesetzt. Der Schrittstatus S41 wird zurückgesetzt, wenn S20 und der Eingang X3 in der Schrittkette I eingeschaltet sind.
Ist der Rücksetzvorgang abgeschlossen, befindet sich die Schrittkette weiterhin im Schrittstatus S20, der von S41 nicht beeinflusst wird.
- ② Der Schrittstatus S31 in der Schrittkette III wird gesetzt, wenn S20 und der Eingang X2 in der Schrittkette I gesetzt sind. S31 wird zurückgesetzt, nachdem zu S32 weitergeschaltet wurde. Der Schrittstatus S20 wird zurückgesetzt, wenn zum Schrittstatus S31 weitergeschaltet wird.

△

5.5 Beispiel für eine Be- und Entladekontrolle

Beispiel ▾

In diesem Beispiel wird ein Containerfahrzeug für Schüttguttransport auf einem festgelegten Abschnitt verfahren und an vorgegebener Stelle be- oder entladen.

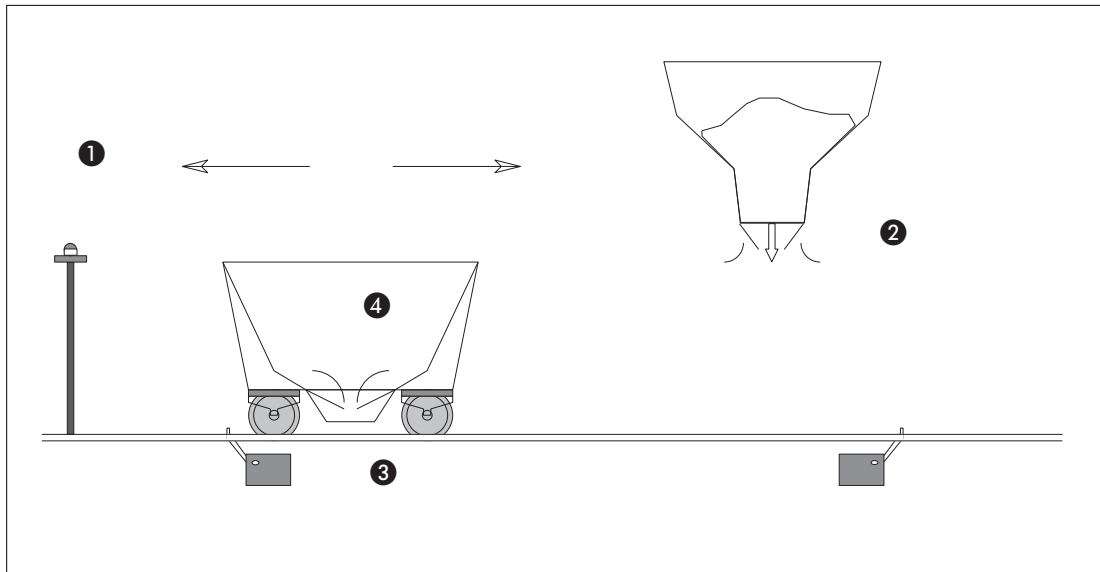


Abb. 5-23: Beispiel für eine Be- und Entladekontrolle eines Containerfahrzeuges

- ① Nach dem Betätigen der Starttaste X0 fährt der Wagen in Richtung Beladestelle und stoppt am Endschalter X1.
- ② Die Silolade öffnet sich für 7 Sekunden (Y1).
- ③ Der Wagen fährt zurück und stoppt am Endschalter X2 an der Entladestelle.
- ④ Die Entladeklappe des Wagens öffnet sich für 5 Sekunden (Y3).

Beispiel ▾

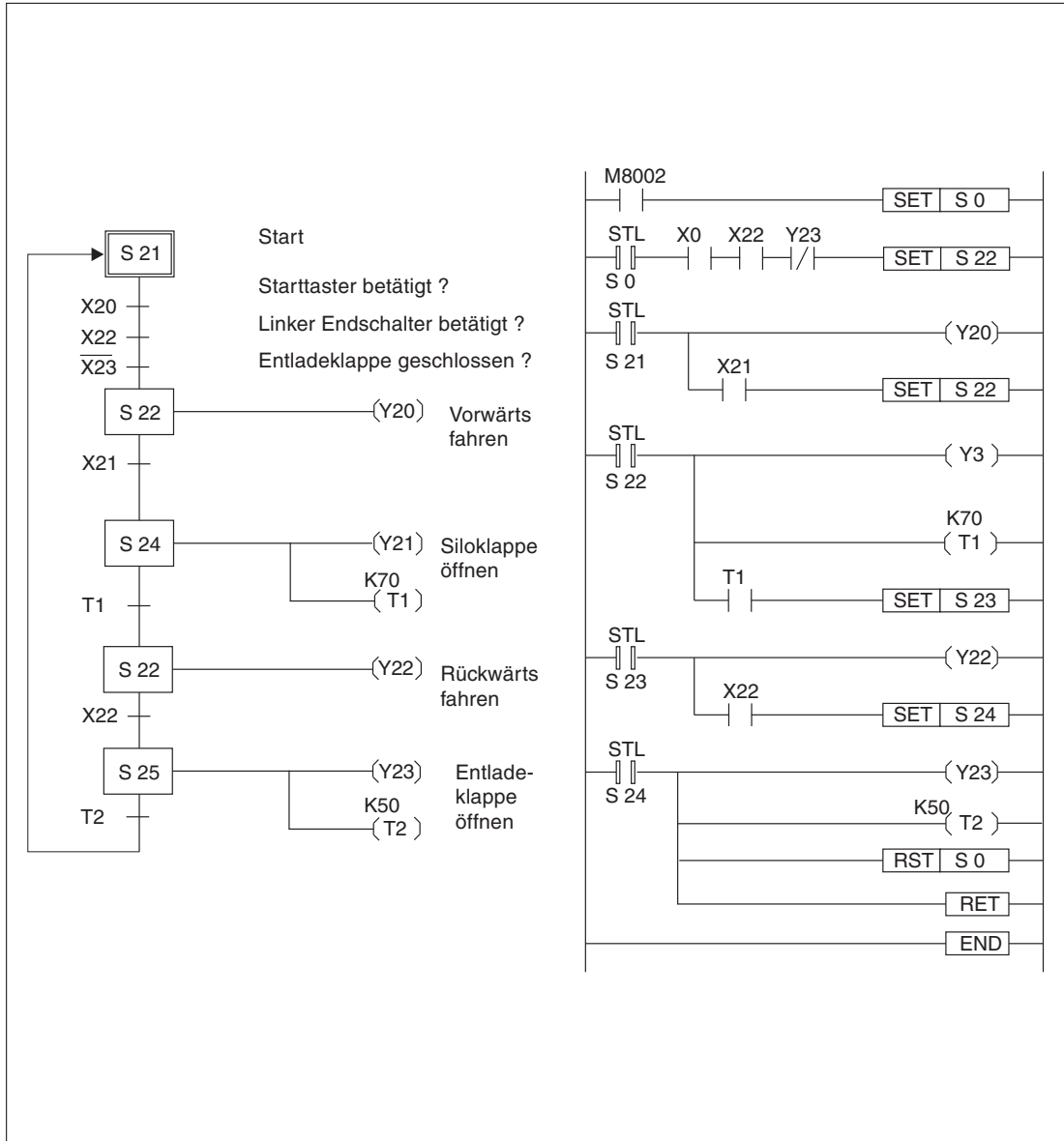


Abb. 5-24: Programmablauf der Be- und Entladekontrolle eines Containerfahrzeuges



5.6 Beispiel für einen Transportier- und Sortiervorgang

Dieses Beispiel zeigt einen Steuerungsmechanismus, bei dem unterschiedlich große Stahlkugeln aus einem Behälter gehoben und über einen Förderweg transportiert werden. Am Ende des Förderweges werden die Kugeln der Größe nach in entsprechende Behälter sortiert.

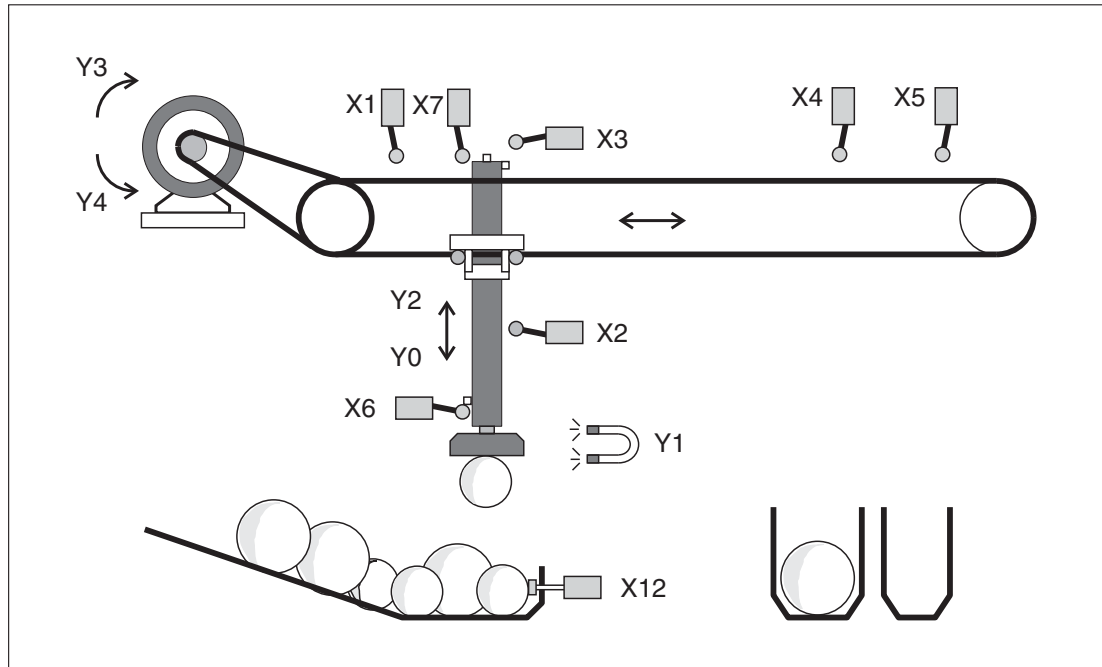


Abb. 5-25: Kugelsortiervorrichtung

- ① Der Hydraulikarm wird aus der Ausgangsposition abgesenkt (Y0=ein).
- ② Wird die untere Grenze nicht erreicht, liegt im Aufnahmeschacht eine große Kugel (X2= aus; X6=ein). Der Kontakt X2 ist geschlossen, wenn im Aufnahmeschacht eine kleine Kugel liegt.
- ③ Der Elektromagnet wird eingeschaltet (Y1=ein) und die Kugel aufgenommen.
- ④ Der Hydraulikarm wird angehoben (Y2=ein). Der Arm stoppt bei Erreichen der oberen Grenze (X3).
- ⑤ Der Hydraulikarm fährt nach rechts (Y3=ein).
- ⑥ Wurde eine kleine Kugel aufgenommen, stoppt der Motor bei Erreichen des Endschalters X4. Wird eine große Kugel aufgenommen, erfolgt der Motorstopp bei Erreichen des Endschalters X5.
- ⑦ Der Hydraulikarm wird abgesenkt (Y0=ein).
- ⑧ Nach Erreichen des Bodens (X6) wird der Magnet ausgeschaltet (Y1=aus).
- ⑨ Der Hydraulikarm wird bis zur oberen Grenze (X3) angehoben (Y2=ein).
- ⑩ Der Hydraulikarm wird in die Ausgangsposition gefahren (Y4=ein).
- ⑪ Die Ausgangsposition ist erreicht (X7=ein).

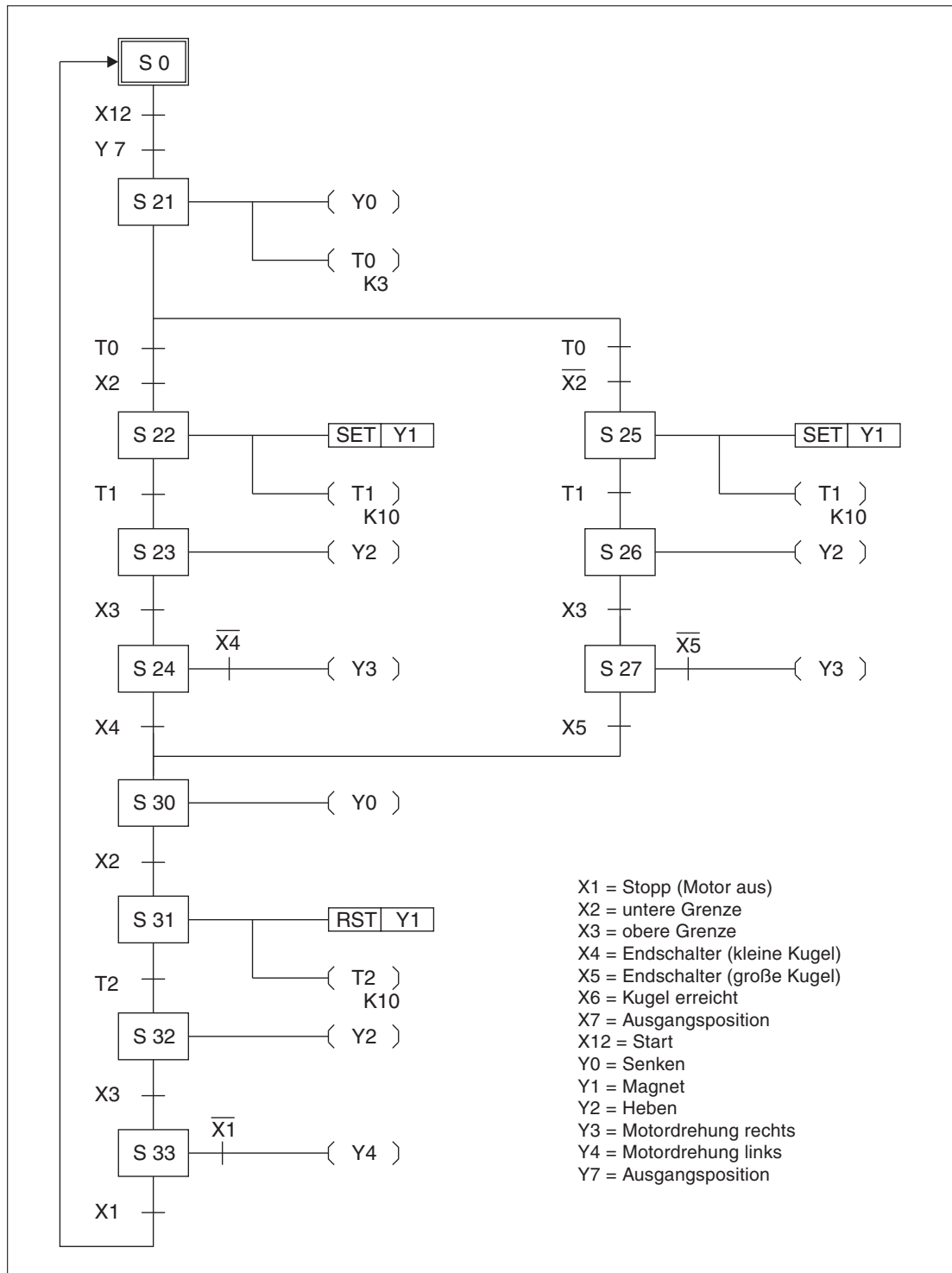


Abb. 5-26: Programmierbeispiel für die vorstehende Sortiervorrichtung

6 Applikationsanweisungen

6.1 Allgemeine Hinweise

Diese Kapitel beschreibt die Applikationsanweisungen der MELSEC FX-Familie. Mit den Applikationsanweisungen lassen sich spezielle Funktionen realisieren, wie zum Beispiel eine Flip-Flop-Funktion oder arithmetische Funktionen. Die Beschreibung einer Applikationsanweisung beginnt jeweils mit einer Übersicht in Form einer Tabelle, in der alle für die Ausführung der Applikationsanweisung wichtigen Informationen aufgeführt sind.

6.1.1 Erläuterung zur Beschreibung der Applikationsanweisungen

Dieser Abschnitt gibt eine einführende Erläuterung zur Struktur der Applikationsanweisungstabellen, die bei jeder Anweisung zu Beginn des jeweiligen Abschnitts zu finden sind.

②			①						
			④						
			CMP		FNC 10				
			③ Numerische Daten vergleichen						
			CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
			●	●	●	●	●	●	
Operanden	S1+, S2+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	Y, M, S	●	16 Bit	32 Bit	CMP	7		
			●	●	●				
⑤			⑥	⑦					

① Anweisung

In der oberen Zeile wird der Name der Anweisung und die zugehörige FNC-Nummer (Function number) angegeben, die bei der Programmierung in Anweisungsliste eingesetzt wird. Sie können je nach Programmiersystem den Anweisungsnamen oder die FNC-Nummer einsetzen.

② Kontaktplansymbol

Das Kontaktplansymbol wird bei der Kontaktplanprogrammierung verwendet. Das Kontaktplansymbol besteht aus der Anweisung und den einsetzbaren Operanden.

③ Bedeutung

Hier finden Sie eine kurze Beschreibung zur Bedeutung der Anweisung.

④ CPU

Die MELSEC FX-Serie, bei der diese Anweisung angewendet werden kann, ist durch einen Punkt (●) gekennzeichnet.

⑤ Operanden

Alle im Zusammenhang mit der Anweisung einsetzbaren Operanden werden in diesem Feld angegeben. Eine eingehende Beschreibung der Operanden und der Datenstruktur enthalten die Abschnitte 6.1.2 bis 6.1.4.

6 Puls-Anweisung

Befindet sich hier ein Punkt (●) , kann die Ausführung der Anweisung auch bei steigender Flanke der Eingangsverknüpfung ausgeführt werden (siehe auch Abschnitt 6.1.5). Der Anweisung muss in diesem Fall ein „P“ nachgestellt werden.

7 Verarbeitung

Hier wird angegeben, ob es sich bei der Anweisung um eine 16-Bit- oder 32-Bit-Anweisung handelt. Bei einer 32-Bit-Anweisung wird dem Anweisungsnamen immer der Buchstabe „D“ vorangestellt (siehe auch Abschnitt 6.1.9).

8 Programmschritte

An dieser Stelle wird die Anzahl der Programmschritte angegeben, die zur vollständigen Ausführung der Anweisung erforderlich sind.

6.1.2 Beschreibung der Operanden**Bit-Operanden**

Ein Bit-Operand kann zwei Signalzustände („0“ und „1“) annehmen. Sein Signalzustand kann demnach mit einem Bit (0 und 1) definiert werden.

Bit-Operanden	Operanden-kennzeichen
Eingang	X
Ausgang	Y
Merker	M
Schrittstatus	S

Tab. 6-1:
Bit-Operanden

Wortoperanden

Wortoperanden können Informationszustände annehmen, die aus mehreren Bits bestehen (numerische Datenwerte). Dabei werden 8 Bits zu einem Byte und 2 Byte zu einem Datenwort zusammengefasst.

Wortoperanden	Operanden-kennzeichen
Timer	T
Counter	C
Datenregister	D
Index-Register	V, Z

Tab. 6-2:
Wortoperanden

6.1.3 Zusammenfassen von Bit-Operanden

Mehrere aufeinanderfolgende Bit-Operanden können in einem Datenwort zusammengefasst werden. Dadurch besteht z. B. die Möglichkeit, die Signalzustände mehrerer Eingänge auf einmal zu verarbeiten.

Die Anzahl der Bit-Operandenadressen, die von einer Applikationsanweisung angesprochen werden sollen, wird durch die Angabe einer Konstanten K festgelegt. Bei 16-Bit-Anweisungen können bis zu 16 und bei 32-Bit-Anweisungen bis zu 32 Operandenadressen in Einheiten von je 4 Operanden vorgegeben werden. Die Anzahl der zusammengefassten Operandenadressen wird als Blocklänge definiert.

Für 16-Bit-Anweisungen liegt die Blocklänge im Bereich von K1 bis K4.

Blocklänge	Anzahl der Adressen
K1	4
K2	8
K3	12
K4	16

Tab. 6-3:
Blocklänge bei 16-Bit-Anweisungen

Für 32-Bit-Anweisungen liegt die Blocklänge im Bereich von K1 bis K8.

Blocklänge	Anzahl der Adressen
K1	4
K2	8
K3	12
K4	16
K5	20
K6	24
K7	28
K8	32

Tab. 6-4:
Blocklänge bei 32-Bit-Anweisungen

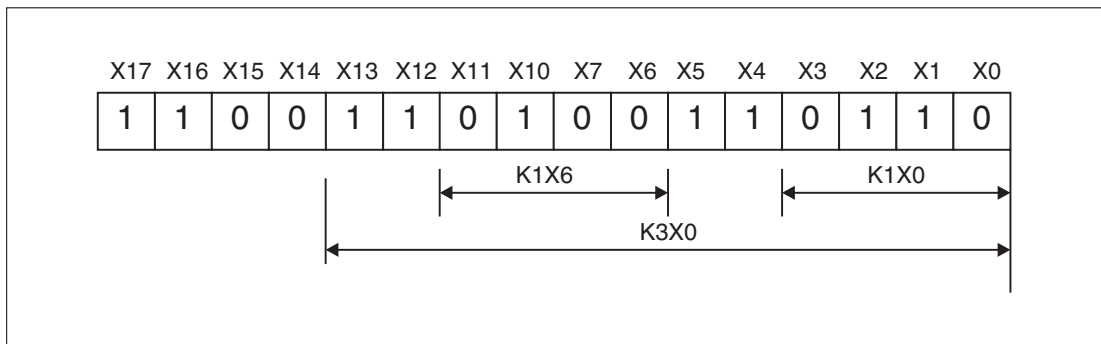
Angabe der Startadresse eines Blocks

Die Angabe der Startadresse legt den Blockbeginn fest. Bei der Angabe der Startadresse kann jede Zahl verwendet werden.

HINWEIS

Verwenden Sie bei der Festlegung von Eingängen X und Ausgängen Y möglichst nur Startadressen, die ein Vielfaches von 10 sind (z. B. X0, X10 usw.).

Bei der Festlegung der Operanden M und S sollten Sie möglichst Startadressen angeben, die ein Vielfaches von 8 sind.

Beispiel ▾ Einteilung der Blocklängen und Startadressen.**Abb. 6-1:** Beispiel zur Einteilung der Blocklängen und Startadressen

K1X0: X0 bis X3 → 4 Eingänge, Startadresse X0

K1X6: X6 bis X11 → 4 Eingänge, Startadresse X6

K3X0: X0 bis X13 → 12 Eingänge, Startadresse X0



6.1.4 Datenstruktur

Quelldaten (S)

Quelldaten sind Daten, die mit einer Applikationsanweisung verarbeitet werden sollen. Die Quelldaten beinhalten eine oder mehrere Operandenadressen und können aus Konstanten und/oder Bit- oder Wortoperanden bestehen.

Konstanten sind numerische Werte, die zur Ausführung einer bestimmten Operation vorgegeben werden. Der Wert einer Konstante wird bei der Programmerstellung festgelegt und kann während der Programmverarbeitung nicht mehr geändert werden.

Mit Bit- oder Wortoperanden wird die Operandenadresse festgelegt, in der die Daten enthalten sind, die verarbeitet werden sollen. Eine Änderung der Daten ist während der Programmverarbeitung jederzeit möglich.

Datengruppe		Kennzeichen
Deutsche Bezeichnung	Englische Bezeichnung	
Quelldaten	Source	(S)
Quelldaten 1	Source 1	(S1)
Quelldaten 2	Source 2	(S2)

Tab. 6-5:
Kennzeichnung der Quelldaten

Zieldaten (D)

Zieldaten sind Daten, die das Operationsergebnis nach Ausführung einer Applikationsanweisung enthalten. Auch Zieldaten bestehen aus einer oder mehreren Operandenadressen und können aus Bit- oder Wortoperanden bestehen. Ein möglicher Adressenbereich der Zieldaten wird über Konstanten festgelegt und muss der Größe des Adressenbereichs der Quelldaten entsprechen.

Datengruppe		Kennzeichen
Deutsche Bezeichnung	Englische Bezeichnung	
Zieldaten	Destination	(D)
Zieldaten 1	Destination 1	(D1)
Zieldaten 2	Destination 2	(D2)

Tab. 6-6:
Kennzeichnung der Zieldaten

6.1.5 Ausführung von Applikationsanweisungen

Es gibt zwei Möglichkeiten, eine Anweisung auszuführen:

- Das Ausführungssignal kann z. B. als statisches Signal anstehen. Wenn das Signal eingeschaltet ist, wird die Anweisung ausgeführt.
- Das Ausführungssignal kann außerdem als ansteigende oder abfallende Flanke anstehen. Die Anweisung kommt nur dann zur Ausführung, wenn ihr Eingangssignal von „0“ nach „1“ bzw. von „1“ nach „0“ wechselt. Dies kann z. B. durch eine vorgeschaltete Impulsfunktion (PLS-, PLF-Anweisung) erreicht werden.

Beispiel ▾

Ausführung durch Merker, der für einen Zyklus gesetzt ist

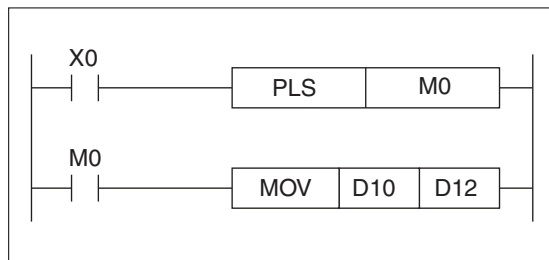


Abb. 6-2:

Ausführung durch Merker, der für einen Zyklus gesetzt ist

△

Beispiel ▾

Ausführung durch Impulssignal

Zusätzlich zu der bereits gezeigten Möglichkeit verfügen die MELSEC-Steuerungen über den Befehlsparameter „P“, der die Flankenerkennung bei steigender Flanke ermöglicht.

Die Funktion des Parameters entspricht der Anweisung „PLS“ des Grundbefehlssatzes.

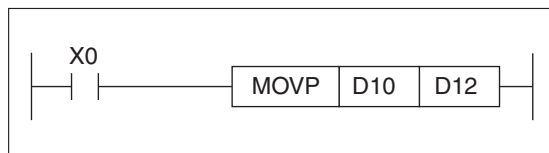


Abb. 6-3:

Ausführung durch Impulssignale

Die Funktion dieses Beispiels ist identisch mit Abb. 6-2.

Die MOV-Anweisung wird ausgeführt, wenn am Eingang X0 ein Signalwechsel von „0“ nach „1“ stattfindet. Die Anweisung wird **nur einmal ausgeführt**. Erst bei einem weiteren Signalwechsel von „0“ nach „1“ wird die Anweisung erneut ausgeführt.

Die Anweisung wird nicht ausgeführt, wenn X0 ausgeschaltet ist.

△

Beispiel ▾

Ausführung durch statisches Signal

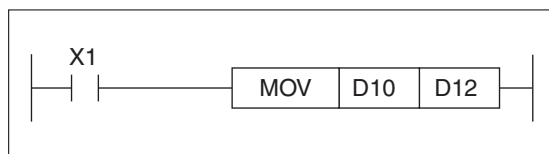


Abb. 6-4:

Ausführung durch statisches Signal

Die MOV-Anweisung wird ausgeführt, wenn am Eingang X1 ein „1“-Signal ansteht. Die Anweisung wird solange in **jedem Programmzyklus ausgeführt**, wie das „1“-Signal ansteht.

Die Anweisung wird nicht ausgeführt, wenn X1 ausgeschaltet ist.

△

6.1.6 Einsatz der Index-Register V, Z

Die Index-Register V und Z werden verwendet, um bei Transfer- und Vergleichsanweisungen zur Operandenadresse einen Indexwert zu addieren.

Die Index-Register V und Z sind 16-Bit-Register.

In 32-Bit-Anweisungen müssen die beiden Index-Register V und Z kombiniert eingesetzt werden. Z speichert die unteren 16 Bit, und V speichert die oberen 16 Bit. Als Zieladresse ist das Index-Register Z anzugeben. Index-Register selbst können nicht indiziert werden.

HINWEIS

Operanden, bei denen eine Indizierung vorgenommen werden kann, sind mit einem Pluszeichen gekennzeichnet: z. B. (S+) und (D+)

Beispiel ▾

Einsatz der Index-Register V, Z

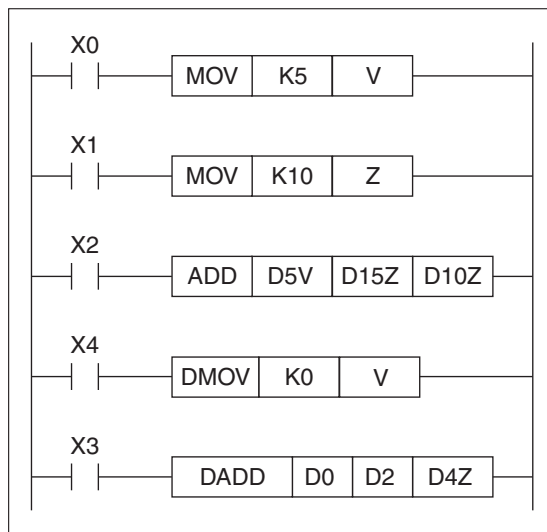


Abb. 6-5:

Programmierbeispiel zum Einsatz der Index-Register V, Z

- Der Wert 5 (K5) wird mit der MOV-Anweisung in das Index-Register V übertragen.
- Der Wert 10 (K10) wird mit der MOV-Anweisung in das Index-Register Z übertragen.
- D5V soll zu D15Z addiert werden. Das Ergebnis soll im Register D10Z abgelegt werden.
- Berechnung der Datenregister:

$$V = 5 \text{ (K5)}$$

$$Z = 10 \text{ (K10)}$$

$$D5V = D10 \text{ (D5 + V = D5 + 5 = D10)}$$

$$D15Z = D25 \text{ (D15 + Z = D15 + 10 = D25)}$$

$$D10Z = D20 \text{ (D10 + Z = D10 + 10 = D20)}$$

- Der Wert 0 (K0) wird mit der MOV-Anweisung in das Index-Register V übertragen. Es findet eine 32-Bit-Operation statt.
- D0 und D1 werden zu D2 und D3 addiert. Das Ergebnis wird in den Datenregistern D14 und D15 gespeichert.

△

6.1.7 Bedeutung der Flags

Bei der Ausführung einiger Applikationsanweisungen werden automatisch von der SPS verschiedene Flags (Sondermerker) gesetzt bzw. zurückgesetzt. Ein gesetztes Flag zeigt einen bestimmten Programmzustand an (z. B. wenn der zulässiger numerischer Datenbereich bei der Ausführung einer Anweisung überschritten wurde).

Diese Flags werden jedesmal gesetzt bzw. zurückgesetzt, wenn die entsprechende Anweisung im Programm aktiviert wird. Das Setzen oder Rücksetzen eines Flags hat jedoch nicht zur Folge, dass im nächsten Programmzyklus, wenn diese Anweisung nicht ausgeführt wird, das Flag seinen Status ändert.

Eine Übersicht über alle Flags und ihre Bedeutung finden Sie im Abschnitt 9.1.3.

6.1.8 Programmablauffehler bei der Ausführung von Applikationsanweisungen

Bei fehlerhaft programmierten Applikationsanweisungen oder Operandenadressen wird eine Fehlermeldung in einem Fehlerdatenregister abgespeichert.

Das Kapitel 10 enthält eine detaillierte Übersicht aller Sondermerker und -register, die Informationen zu Fehlern enthalten sowie sämtlicher Fehlermeldungen.

6.1.9 32-Bit-Anweisungen

Wenn eine Anweisung als 32-Bit-Anweisung ausgeführt werden soll, wird der Anweisung der Parameter „D“ vorangestellt. Ob eine Anweisung als 32-Bit-Anweisung ausführbar ist, kann anhand der Übersichtstabelle zu jeder Applikationsanweisung ersehen werden.

Bei der Verarbeitung von Applikationsanweisungen mit Wortoperanden müssen Sie darauf achten, dass zwei Wortoperanden zu einem 32-Bit-Wort zusammengefasst werden, aber immer das niedrigstwertige Byte adressiert wird (untere 16 Bit).

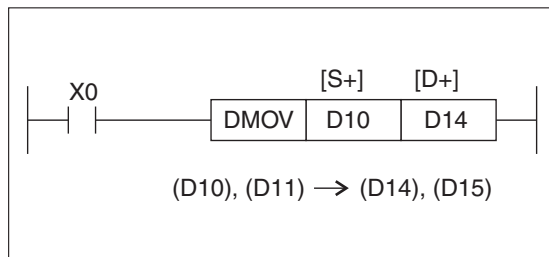
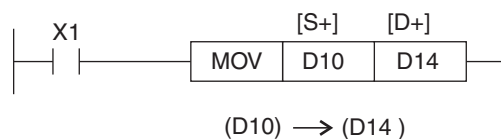


Abb. 6-6:

Mit einer 32-Bit-Anweisung DMOV werden in diesem Beispiel die Inhalte der Datenregister D10 und D11 in die Datenregister D14 und D15 übertragen



HINWEIS

Beim Einsatz der Index-Register in Zusammenhang mit einer 32-Bit-Anweisung darf nur das Index-Register Z adressiert werden.

6.1.10 Übersicht der Applikationsanweisungen

Einteilung	Anweisung	FNC	Bedeutung	Referenz (Abschnitt)	Steuerung					
					FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
Programm-ablauf-anweisungen	CJ	00	Sprung innerhalb eines Programms	6.2.1						
	CALL	01	Aufruf eines Unterprogramms	6.2.2						
	SRET	02	Ende eines Unterprogramms	6.2.3						
	IRET	03	Interrupt-Programm abschließen	6.2.4						
	EI	04	Interrupt-Programm aktivieren							
	DI	05	Interrupt-Programm deaktivieren		●	●	●	●	●	●
	FEND	06	Ende eines Programmbereichs	6.2.5						
	WDT	07	Watch-Dog-Timer auffrischen	6.2.6						
	FOR	08	Anfang einer Programmwiederholung	6.2.7						
NEXT	09	Ende einer Programmwiederholung								
Vergleichs- und Transferanweisungen	CMP	10	Numerische Daten vergleichen	6.3.1	●	●	●	●	●	●
	ZCP	11	Numerische Datenbereiche vergleichen	6.3.2	●	●	●	●	●	●
	MOV	12	Datentransfer	6.3.3	●	●	●	●	●	●
	SMOV	13	Shift-Transfer	6.3.4			●	●	●	●
	CML	14	Kopieren und invertieren	6.3.5			●	●	●	●
	BMOV	15	Block-Transfer	6.3.6	●	●	●	●	●	●
	FMOV	16	Transfer von gleichen Daten	6.3.7			●	●	●	●
	XCH	17	Austausch von Daten	6.3.8			●			●
	BCD	18	BCD-Konvertierung	6.3.9	●	●	●	●	●	●
	BIN	19	Binär-Konvertierung	6.3.10	●	●	●	●	●	●
Arithmetische Anweisungen	ADD	20	Addition numerischer Daten	6.4.1	●	●	●	●	●	●
	SUB	21	Subtraktion numerischer Daten	6.4.2	●	●	●	●	●	●
	MUL	22	Multiplikation numerischer Daten	6.4.3	●	●	●	●	●	●
	DIV	23	Division numerischer Daten	6.4.4	●	●	●	●	●	●
	INC	24	Inkrementieren	6.4.5	●	●	●	●	●	●
	DEC	25	Dekrementieren	6.4.6	●	●	●	●	●	●
	AND	26	Logische UND-Verknüpfung	6.4.7	●	●	●	●	●	●
	OR	27	Logische ODER-Verknüpfung	6.4.8	●	●	●	●	●	●
	XOR	28	Logische Exklusiv-ODER-Verknüpfung	6.4.9	●	●	●	●	●	●
Verschiebe-anweisungen	NEG	29	Negation von Daten	6.4.10			●			●
	ROR	30	Rotation nach rechts	6.5.1			●	●	●	●
	ROL	31	Rotation nach links	6.5.2			●	●	●	●
	RCR	32	Rotieren von Bits nach rechts	6.5.3			●			●
	RCL	33	Rotieren von Bits nach links	6.5.4			●			●
	SFTR	34	Binäre Daten bitweise verschieben, rechts	6.5.5	●	●	●	●	●	●
	SFTL	35	Binäre Daten bitweise verschieben, links		●	●	●	●	●	●
	WSFR	36	Daten wortweise nach rechts verschieben	6.5.6			●	●	●	●
	WSFL	37	Daten wortweise nach links verschieben	6.5.7			●	●	●	●
	SFWR	38	Schreiben in einen FIFO-Speicher	6.5.8	●	●	●	●	●	●
SFRD	39	Lesen aus einem FIFO-Speicher	6.5.9	●	●	●	●	●	●	

Tab. 6-7: Übersicht der Applikationsanweisungen (1)

Einteilung	Anweisung	FNC	Bedeutung	Referenz (Abschnitt)	Steuerung					
					FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
Datenoperationen	ZRST	40	Operandenbereiche zurücksetzen	6.6.1	●	●	●	●	●	●
	DECO	41	Daten decodieren	6.6.2	●	●	●	●	●	●
	ENCO	42	Daten codieren	6.6.3	●	●	●	●	●	●
	SUM	43	Ermittlung gesetzter Bits	6.6.4			●	●	●	●
	BON	44	Überprüfen eines Bits	6.6.5			●	●	●	●
	MEAN	45	Ermittlung von Durchschnittswerten	6.6.6			●	●	●	●
	ANS	46	Starten eines Zeitintervalls	6.6.7			●	●		●
	ANR	47	Rücksetzen von Anzeige-Bits	6.6.8			●	●		●
	SQR	48	Ermittlung der Quadratwurzel	6.6.9			●			●
	FLT	49	Umwandlung des Zahlenformats	6.6.10			●	●	●	●
High-Speed-Anweisungen	REF	50	Ein- und Ausgänge auffrischen	6.7.1	●	●	●	●	●	●
	REFF	51	Einstellen der Eingangsfiler	6.7.2			●			●
	MTR	52	Einlesen einer Matrix (MTR)	6.7.3			●	●	●	●
	DHSCS	53	Setzen durch High-Speed-Counter	6.7.4	●	●	●	●	●	●
	DHSCR	54	Rücksetzen durch High-Speed-Counter		●	●	●	●	●	●
	DHSZ	55	Bereichsvergleich	6.7.5			●	●	●	●
	SPD	56	Geschwindigkeitserkennung	6.7.6	●	●	●	●	●	●
	PLSY	57	Ausgabe einer definierten Anzahl von Impulsen	6.7.7	●	●	●	●	●	●
	PWM	58	Impulsausgabe mit Impulsweitenmodulation	6.7.8	●	●	●	●	●	●
	PLSR	59	Ausgabe einer bestimmten Anzahl von Impulsen	6.7.9	●	●	●	●	●	●
Anwendungs- bezogene Anweisungen	IST	60	Schrittstatus initialisieren	6.8.1	●	●	●	●	●	●
	SER	61	Suchanweisung	6.8.2			●	●	●	●
	ABSD	62	Absoluter Counter-Vergleich	6.8.3	●	●	●	●	●	●
	INCD	63	Inkrementaler Counter-Vergleich	6.8.4	●	●	●	●	●	●
	TTMR	64	Teaching-Timer	6.8.5			●			●
	STMR	65	Sonder-Timer	6.8.6			●			●
	ALT	66	Flip-Flop-Funktion	6.8.7	●	●	●	●	●	●
	RAMP	67	Rampenfunktion	6.8.8	●	●	●	●	●	●
	ROTC	68	Rundtisch-Positionierung	6.8.9			●			●
	SORT	69	Sortieranweisung	6.8.10			●			●

Tab. 6-7: Übersicht der Applikationsanweisungen (2)

HINWEIS

Die Applikationsanweisungen ab FNC 70 werden im Kapitel 7 beschrieben.

6.2 Programmablaufanweisungen

Übersicht der Anweisungen FNC 00 bis 09

Symbol	FNC	Bedeutung	Abschnitt
CJ	00	Sprung innerhalb eines Programms	6.2.1
CALL	01	Aufruf eines Unterprogramms	6.2.2
SRET	02	Ende eines Unterprogramms	6.2.3
IRET	03	Interrupt-Programm abschließen	6.2.4
EI	04	Interrupt-Programm aktivieren	
DI	05	Interrupt-Programm deaktivieren	
FEND	06	Ende eines Programmbereichs	6.2.5
WDT	07	Watch-Dog-Timer auffrischen	6.2.6
FOR	08	Anfang einer Programmwiederholung	6.2.7
NEXT	09	Ende einer Programmwiederholung	

Tab. 6-8: Übersicht der Programmablaufanweisungen

6.2.1 Sprung innerhalb eines Programms (CJ)

	CJ		FNC 00				
	Sprung innerhalb eines Programms						
	CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
		●	●	●	●	●	●
Operanden	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
FX1S, FX1N, FX2N, FX2NC: Pointer P0 bis P63 FX3G, FX3GC, FX3GE: Pointer P0 bis P2047 FX3S: Pointer P0 bis P255 FX3U, FX3UC: Pointer P0 bis P4095 P63 bedeutet bei allen Steuerungen einen Sprung zur END-Anweisung. Pointer können auch indiziert angegeben werden.	●		16 Bit	32 Bit	CJ / CJP	3	
					Pointer P**	1	

Funktionsweise

Mit der CJ-Anweisung können Teile eines Programms übersprungen werden. Die Programmzykluszeit kann durch Einsatz der CJ-Anweisung reduziert werden.

Beschreibung

- Das Sprungziel wird durch Festlegen einer Markierung (Pointer-Markierung) im Programm definiert.
- Die Angabe der Sprungzieladresse (Pointer-Adresse) legt fest, zu welcher Pointer-Markierung der Sprung erfolgen soll.
- Wird innerhalb der Sprungroutine eine Rücksetzanweisung für remanente Counter programmiert, ist der Rücksetzvorgang (Löschen von Istwerten) auch dann noch wirksam, wenn der Strompfad der Counter-Spule übersprungen wird.
- Es besteht die Möglichkeit, Ausgänge doppelt zu belegen.

HINWEIS

Achten Sie bei einer Doppelbelegung von Ausgängen darauf, dass nie beide Ausgänge zur gleichen Zeit aktiv sein können. Dies führt zu Programmablaufstörungen.

Festlegen der Pointer-Markierung im Programm

- Die Pointer-Markierung wird bei der Programmierung in Anweisungsliste direkt vor einem Strompfad (vor einer LD- oder LDI-Anweisung) festgelegt.
- Bei der Programmierung in Kontaktplan wird die Pointer-Markierung links vor dem Strompfad festgelegt.

Beispiel ▾

Ist X0 eingeschaltet, findet ein Sprung zur Pointer-Markierung P20 statt.

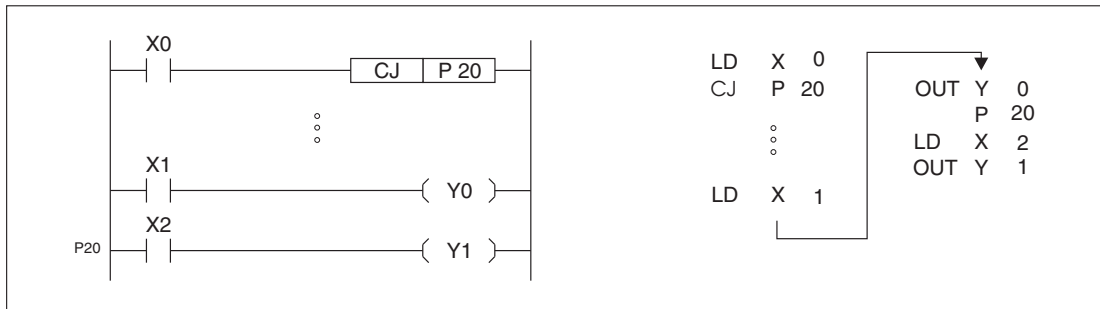


Abb. 6-10: Programmierbeispiel zur CJ-Anweisung



Zweimaliger Einsatz der Sprungzieladresse (Pointer-Adresse) in einem Programm

Beispiel ▾

Zweimaliger Einsatz der Pointer-Adresse P9 in einem Programm.

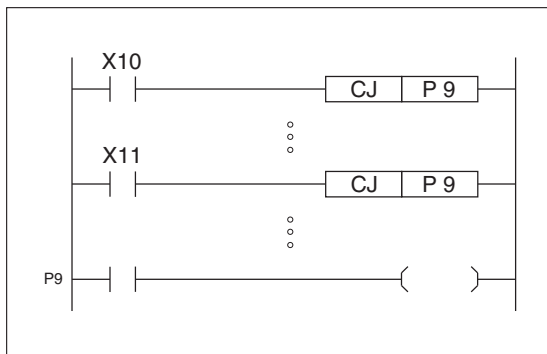


Abb. 6-8: Die Pointer-Adresse P9 wird in einem Programm zwei mal angesprungen.

Ist X10 eingeschaltet, findet ein Sprung zur Pointer-Markierung P9 statt. Ist X10 ausgeschaltet und X11 eingeschaltet, wird ebenso ein Sprung nach P9 ausgeführt.



HINWEIS

Die gleiche Pointer-Markierung darf jedoch nicht mehrfach in einem Programm benutzt werden. Es tritt sonst ein Programmablauffehler auf.

Festlegen der Pointer-Markierung vor der CJ-Sprunganweisung

Innerhalb eines Programms kann auch ein Rücksprung ausgeführt werden.

HINWEIS

Wenn das Eingangssignal für die CJ-Anweisung länger als 200 ms ansteht, tritt ein Watch-Dog-Timer-Fehler auf.

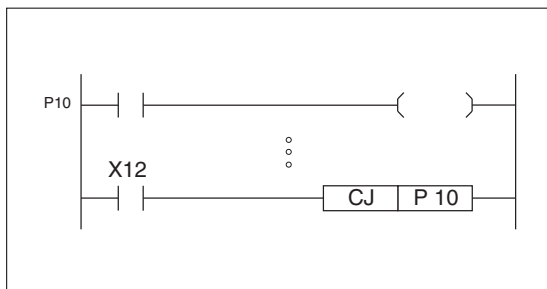


Abb. 6-9: Programmierbeispiel zum Festlegen einer Pointer-Markierung vor der CJ-Sprunganweisung

Sprünge im Master-Control-Bereich

Den Programmablauf beim Einsatz der CJ-Anweisungen in Zusammenhang mit den MC- und MCR-Anweisungen zeigt das folgende Beispiel:

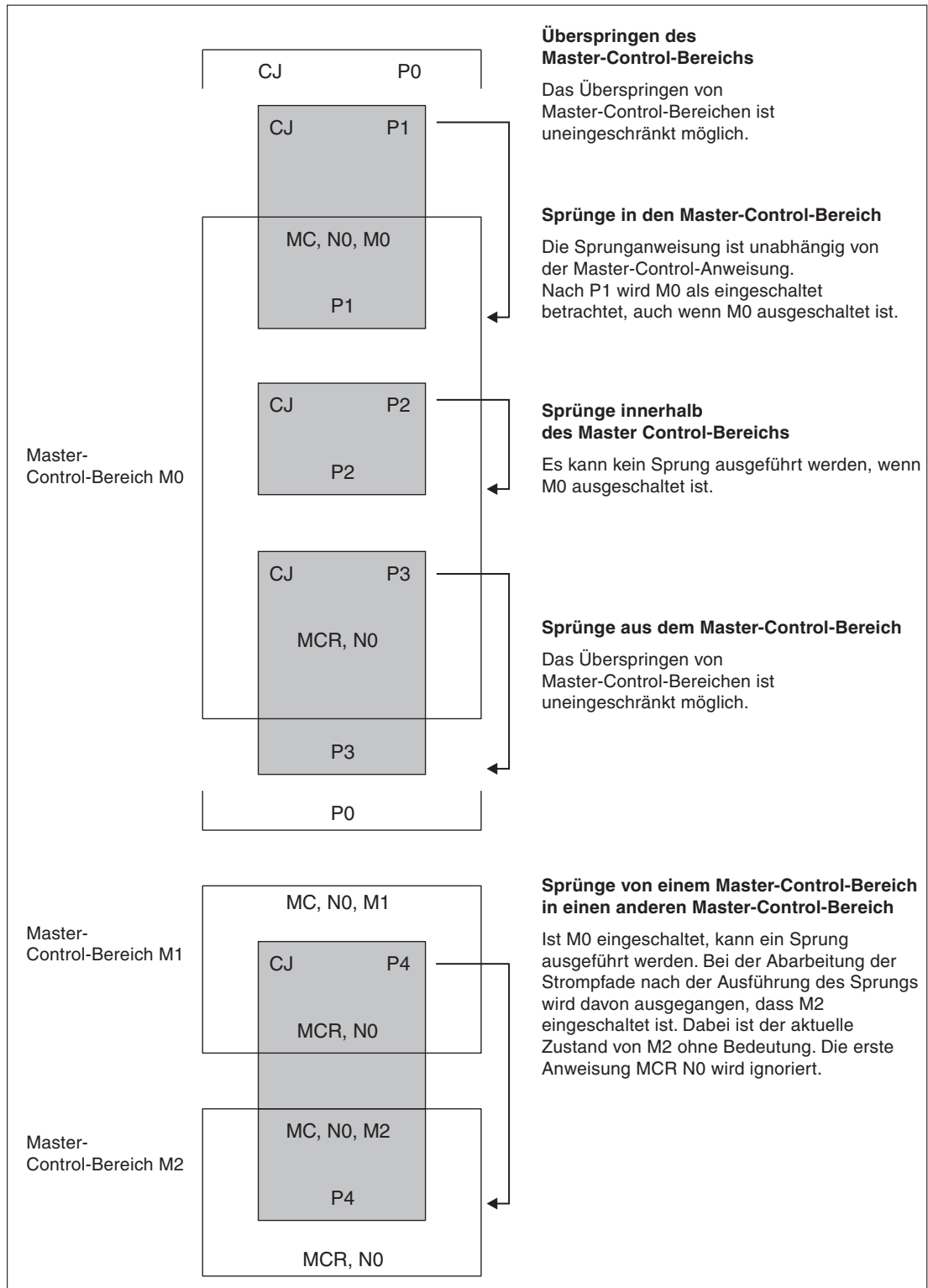


Abb. 6-11: Beispiel zu Sprüngen im Master-Control-Bereich

Verhalten der Kontakte und Spulen innerhalb eines übersprungenen Programmteils

Operanden	Status des Kontaktes und der Spule vor dem Sprung	Status des Kontaktes und der Spule nach dem Sprung	Bemerkung
Ausgänge Y	EIN	EIN	—
Merker M	EIN	EIN	—
Schrittstatus S	EIN	EIN	—
Timer T	EIN	EIN	Der Zeitablauf wird gestoppt. Der aktuelle Zeitwert wird gespeichert. Ist die Sprungbedingung nicht mehr erfüllt, wird der Zeitablauf fortgesetzt.
Counter C	EIN	EIN	Die Zählung wird gestoppt. Der aktuelle Zählerwert wird gespeichert. Ist die Sprungbedingung nicht mehr erfüllt, wird die Zählung fortgesetzt.
Applikationsanweisungen	—	—	Die Ausführung wird gestoppt. Die Anweisungen, wie z. B. RAMP, INC oder DEC, behalten jedoch ihre aktuellen Datenwerte.

Tab. 6-9: Verhalten der Kontakte und Spulen bei der Abarbeitung einer Sprunganweisung

6.2.2 Aufruf eines Unterprogramms (CALL)

		CALL		FNC 01					
		Aufruf eines Unterprogramms							
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
			●	●	●	●	●	●	
Operanden	D		Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	FX1S, FX1N, FX2N, FX2NC: Pointer P0 bis P62 FX3G, FX3GC, FX3GE: Pointer P0 bis P62 und P64 bis P2047		●	16 Bit		32 Bit		CALL/CALLP	3
	FX3S: Pointer P0 bis P62 und P64 bis P255 FX3U, FX3UC: Pointer P0 bis P62 und P64 bis P4095 Pointer können auch indiziert angegeben werden			●				Pointer P**	1

Funktionsweise

Mit der CALL-Anweisung werden Unterprogramme aufgerufen.

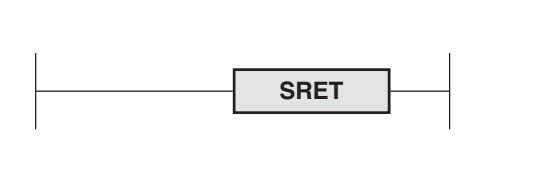
Beschreibung

- Ein Unterprogramm wird mit einem Pointer markiert und über die CALL-Anweisung aufgerufen.
- Am Ende eines Unterprogramms muss eine SRET-Anweisung stehen (Abschnitt 6.2.3).
- Unterprogramme werden hinter der FEND-Anweisung und vor der END-Anweisung programmiert.
- Wenn eine CALL-Anweisung aktiviert wird, findet ein Sprung zu der angegebenen Pointer-Markierung statt. Nach der Bearbeitung der SRET-Anweisung findet ein Rücksprung zu der auf die CALL-Anweisung folgenden Anweisung statt.
- In einem Unterprogramm aktivierte Operanden bleiben nach der Abarbeitung des Unterprogramms bis zur erneuten Bearbeitung des Unterprogramms aktiviert.
- Bei einer FX1S/FX1N/FX2N/FX2NC sollten in einem Unterprogramm die Timer T192 bis T199 und T246 bis T249 genutzt werden.
- Bei einer FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC sollten in einem Unterprogramm die Timer T192 bis T199 genutzt werden.
- Derselbe Pointer kann in beliebig vielen CALL-Anweisungen genutzt werden. Er darf aber nur einmal als Pointer-Markierung programmiert werden.

HINWEIS

Innerhalb eines Unterprogramms können weitere Unterprogramme aufgerufen werden. Es sind maximal 4 Verzweigungsebenen möglich.

6.2.3 Ende eines Unterprogramms (SRET)

		SRET		FNC 02			
		Aufruf eines Unterprogramms					
Operanden	D	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	—	●	16 Bit	32 Bit	SRET	1	
			●				

Funktionsweise

Mit der SRET-Anweisung wird das Ende eines Unterprogramms gekennzeichnet.

Beschreibung

- Ein Unterprogramm wird mit einem Pointer (Pn) markiert und über die CALL-Anweisung (siehe Abschnitt 6.2.2) aufgerufen.
- Am Ende eines Unterprogramms muss eine SRET-Anweisung stehen.
- Unterprogramme werden hinter der FEND-Anweisung und vor der END-Anweisung programmiert.
- Nach der Bearbeitung der SRET-Anweisung findet ein Rücksprung zu der auf die CALL-Anweisung folgenden Anweisung statt.

HINWEIS

Eine SRET-Anweisung kann nur im Zusammenhang mit der CALL-Anweisung programmiert werden.

Beispiel ▾

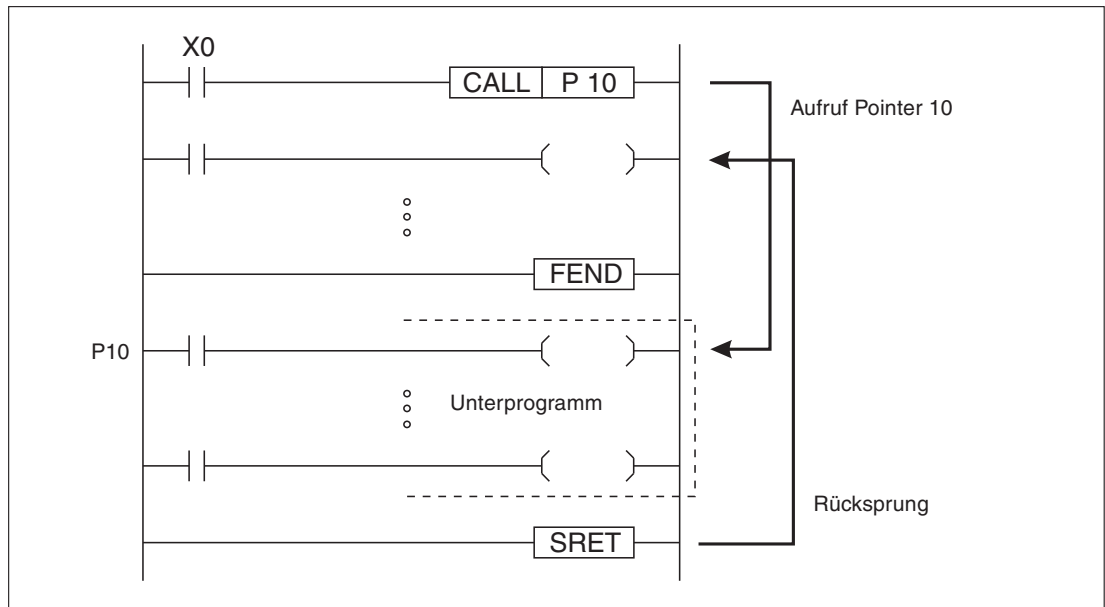


Abb. 6-12: Programmierbeispiel zum Einsatz der CALL- und SRET-Anweisungen



6.2.4 Einsatz eines Interrupt-Programms (IRET, EI, DI)

		IRET		FNC 03				
		Interrupt-Programm abschließen						
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
			●	●	●	●	●	●
Operanden	D	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	—		16 Bit	32 Bit	IRET	1		
					Pointer I***	1		

		EI		FNC 04				
		Pulse-Catch-/Interrupt-Programm aktivieren						
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
			●	●	●	●	●	●
Operanden	D	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	—		16 Bit	32 Bit	EI	1		
					Pointer I***	1		

		DI		FNC 05				
		Interrupt-Programm deaktivieren						
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
			●	●	●	●	●	●
Operanden	D	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	—		16 Bit	32 Bit	DI	1		
					Pointer I***	1		

Funktionsweise

Aufruf, Freigeben und Sperren von Interrupt-Programmen

Interrupt-Programm aufrufen

- Beim Aufruf eines Interrupt-Programms wird das SPS-Programm verlassen, und in das Interrupt-Programm gesprungen. Nachdem das Interrupt-Programm beendet ist, wird in das Hauptprogramm zurück gesprungen.
- Der Beginn eines Interrupt-Programms wird durch Festlegen einer Markierung (Interrupt-Pointer) definiert.
- Das Ende eines Interrupt-Programms wird mit der IRET-Anweisung festgelegt.
- Die Eingänge X0 bis X5 bilden die Interrupt-Eingänge.
- Die Pulsweite der Interrupt-Signale muss mindestens 200 µs betragen.
- Ein Interrupt-Programm muss am Ende eines SPS-Programms hinter der letzten FEND-Anweisung und vor der END-Anweisung programmiert werden.

HINWEISE

Bei einer FX2N, FX2NC, FX3U und FX3UC sind auch Interrupts durch Timer oder Counter möglich. Bei der FX3G-, FX3GC-, FX3GE- und FX3S-Serie können Interrupts auch durch Timer ausgelöst werden.

Die Eingänge X0 bis X5 können nicht gleichzeitig zur Verarbeitung von Interrupt-Signalen und zur Verarbeitung von High-Speed-Zählersignalen eingesetzt werden.

Interrupt-Pointer adressieren

- Die Adressierung eines Interrupt-Pointers wird wie folgt vorgenommen:

Interrupt-Pointer: I ◆ 0 ◇

◆: Adresse 0 bis 5; entspricht Eingang X0 bis X5

◇: 0:= Interrupt bei abfallender Eingangssignalfanke
1:= Interrupt bei ansteigender Eingangssignalfanke

Timer-Interrupt (nur FX2N, FX2NC, FX3G, FX3GC, FX3GE, FX3S, FX3U und FX3UC)

- Das Interrupt-Programm wird jeweils nach Ablauf der vorgegebenen Zeit bearbeitet.

Interrupt-Pointer: I ◆◇◇

◇◇: Intervallzeit 10 bis 99 ms

◆: Adresse 6 bis 8 (T6 bis T8)

Counter-Interrupt (nur FX2N, FX2NC, FX3U und FX3UC)

- Das Interrupt-Programm wird nach Erreichen eines vorgegebenen Zählwertes bearbeitet.

Interrupt-Pointer: I 0 ◇ 0

◇: Interrupt-Nummer 1 bis 6

HINWEIS

Jede Adresse darf nur einmal verwendet werden.

Einsatz der EI- und DI-Anweisung

- Mit der **EI-Anweisung** können Sie ein **Interrupt-Programm freigeben**. Dies bedeutet, dass nach der Abarbeitung der EI-Anweisung, Signalwechsel, die an einem der Eingänge X0 bis X5 auftreten, als Interrupt-Signale im Programm verarbeitet werden.
- Mit der **DI-Anweisung** können Sie ein **Interrupt-Programm sperren**. Dies bedeutet, dass nach Abarbeitung der DI-Anweisung, Signalwechsel, die an einem der Eingänge X0 bis X5 auftreten, nicht mehr als Interrupt-Signale im Programm verarbeitet werden.

HINWEIS

Wenn keine der beiden Anweisungen EI oder DI programmiert wurde, ist der Interrupt-Modus nicht aktiviert, d. h., es können dann keine Interrupt-Signale verarbeitet werden.

Abarbeitung von Interrupt-Programmen

- Während der Ausführung eines Interrupt-Programms können keine anderen Interrupt-Programme aufgerufen werden. Sie können jedoch zwei Verzweigungsebenen programmieren. Die Anweisungen EI und DI müssen Sie dann innerhalb des Interrupt-Programms einsetzen.
- Mehrere aufeinander folgende Interrupt-Programme werden in der Reihenfolge ihres Aufrufs abgearbeitet.
- Werden mehrere Interrupt-Programme gleichzeitig aufgerufen, wird das Interrupt-Programm mit der niedrigsten Pointer-Adresse zuerst abgearbeitet.

- Ein Interrupt-Programm, das in einem Bereich zwischen einer DI-Anweisung und einer EI-Anweisung aufgerufen wurde, wird erst nach der Ausführung der EI-Anweisung abgearbeitet.

Ausschalten beliebiger Interrupts

- Sie können beliebige Interrupts durch Einschalten der zugehörigen Sondermerker zeitweilig oder permanent ausschalten. Die entsprechenden Sondermerker werden im Abschnitt 9.1.6 angegeben. Für alle Steuerungen ist der erste Sondermerker M8050, der den Interrupt I0^{①②} ausschaltet.

HINWEISE

Setzen Sie nie einen Sondermerker, ohne sich seiner Funktion sicher zu sein. Nicht alle Steuerungen arbeiten immer mit den gleichen Sondermerkern.

High-Speed-Counter-Interrupts können immer nur als einzelne Gruppe mit dem Sondermerker M8059 ausgeschaltet werden.

Es können maximal zwei Verzweigungsebenen programmiert werden.

Ein Interrupt-Programm wird nicht ausgeführt, wenn der zugehörige Sondermerker aktiviert ist. So kommt das Interrupt-Programm I□** nicht zur Ausführung, wenn der Sondermerker M805□ (□: 1, 2, 3, 4, 5) auf „1“ gesetzt ist.

Speichern der Signalwechsel der Interrupt-Eingänge

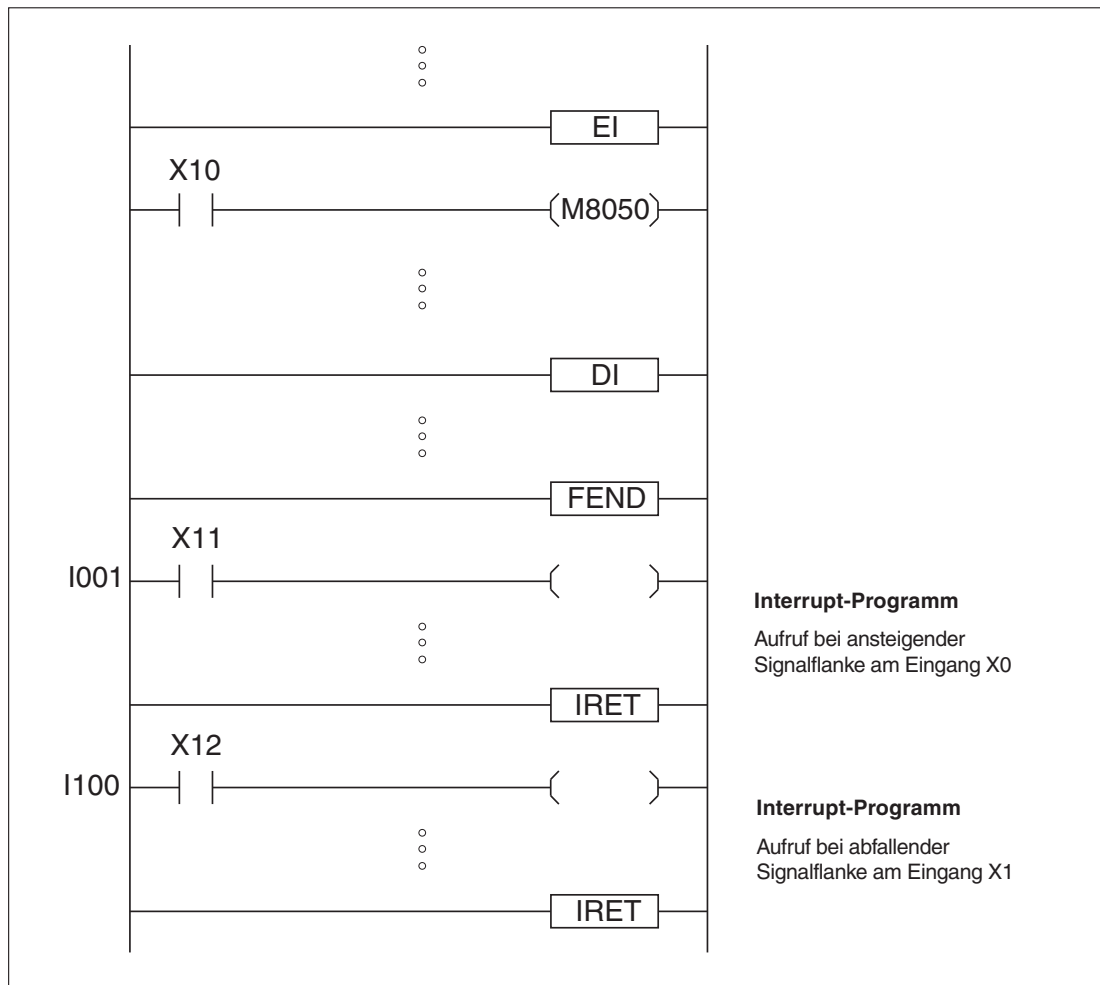
Diese Funktion (Pulse-Catch-Funktion, Abschnitt 8.3) ermöglicht das Speichern der Signalwechsel der Interrupt-Eingänge X0 bis X5 (X0 bis X7 bei FX3U und FX3UC) in den Sondermerkern M8170 bis M8175 bzw. M8170 bis M8177. Diese Speicherfunktion kann nur einmal für einen Eingang gleichzeitig ausgeführt werden. Außer bei der FX3G-, FX3GC-, FX3GE-, und FX3S-Serie wird die Pulse-Catch-Funktion mit einer EI-Anweisung aktiviert.

Beispiel ▽

Interrupt-Pointer adressieren

Pointer: I001

Erklärung: Interrupt-Eingang X0, Interrupt bei ansteigender Eingangssignalfanke (Signalwechsel von „0“ auf „1“) △

Beispiel ▾ Einsatz der Anweisungen EI, DI und IRET**Abb. 6-13:** Programmierbeispiel zum Einsatz der Anweisungen EI, DI und IRET

- Wenn am Eingang X0 ein Interrupt-Signal ansteht, während ein Programmschritt innerhalb des Bereichs von der EI-Anweisung bis zur DI-Anweisung ausgeführt wird, findet ein Sprung zum Interrupt-Programm I001 statt. Das Interrupt-Programm wird ausgeführt, und es erfolgt ein Rücksprung zum SPS-Programm.
- Das Interrupt-Programm I001 wird nicht bearbeitet, wenn der Sondermerker M8050 aktiviert ist (Eingang X10 eingeschaltet).
- Wenn am Eingang X1 ein Interrupt-Signal ansteht, während ein Programmschritt innerhalb des Bereichs von der EI-Anweisung bis zur DI-Anweisung ausgeführt wird, findet ein Sprung zum Interrupt-Programm I100 statt. Das Interrupt-Programm wird ausgeführt und anschließend in das Hauptprogramm zurück gesprungen.
- Wenn beide Signale X0 und X1 gleichzeitig auftreten, wird erst das Interrupt-Programm I001 und danach das Interrupt-Programm I100 bearbeitet.

△

6.2.5 Ende eines Programmbereichs (FEND)

		FEND		FNC 06				
		Ende eines Programmbereichs						
Operanden	D	CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
	—		●	●	●	●	●	●
		Puls-Anweisung (P)	Verarbeitung		Programmschritte			
			16 Bit	32 Bit	FEND		1	

Funktionsweise

Beenden von einzelnen Programmbereichen innerhalb eines SPS-Programms

Beschreibung

- Mit einer FEND-Anweisung wird das Ende eines Programmbereichs festgelegt. Sie können mehrere FEND-Anweisungen innerhalb eines SPS-Programms einsetzen.
- Nach Abarbeitung der FEND-Anweisung erfolgt die Ausgangsbearbeitung. Danach findet ein Rücksprung zum Programmschritt 0 statt. Die Eingangsbearbeitung und der Watch-Dog-Timer werden aufgefrischt.

HINWEISE

Programmieren Sie die Interrupt-Programme zwischen der letzten FEND-Anweisung und der END-Anweisung.

Verwechseln Sie nicht die FEND-Anweisung mit der END-Anweisung. Mit der END-Anweisung wird das gesamte SPS-Programm abgeschlossen (siehe Abschnitt 4.18).

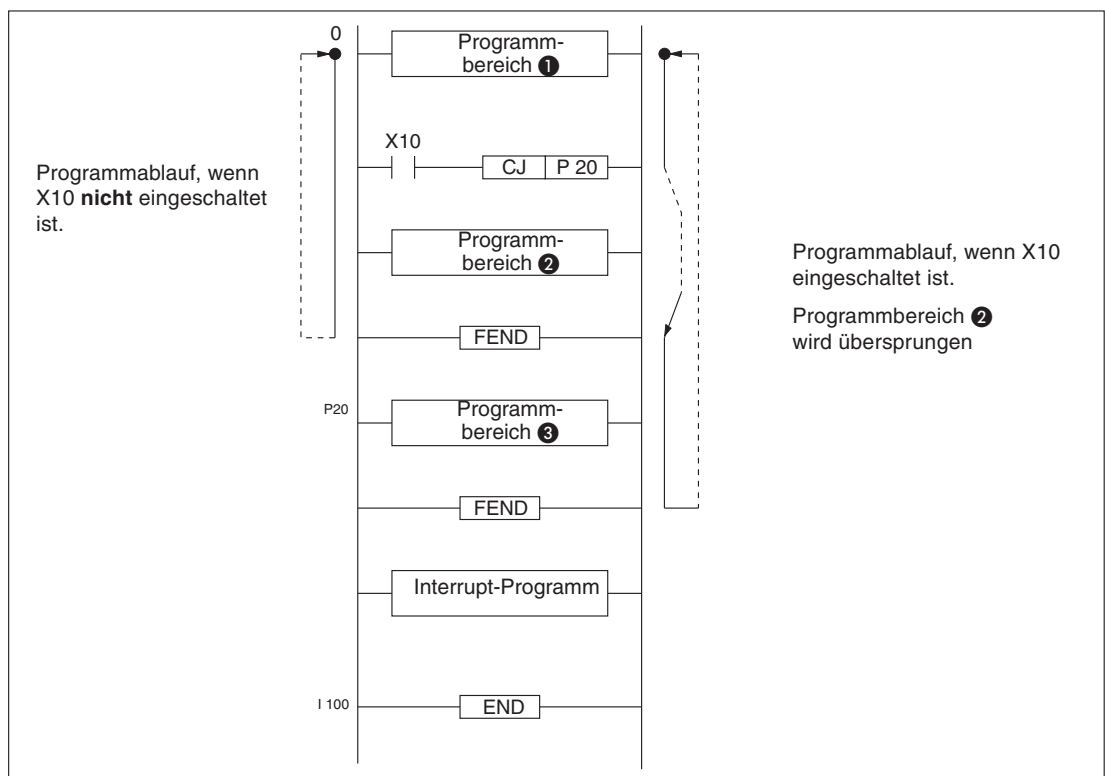


Abb. 6-14: Programmierbeispiel zum Einsatz der FEND-Anweisung



6.2.6 Watch-Dog-Timer auffrischen (WDT)

		WDT		FNC 07				
		Watch-Dog-Timer auffrischen						
Operanden	D	CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
	—	●	●	●	●	●	●	●
Operanden		Puls-Anweisung (P)	Verarbeitung		Programmschritte			
			16 Bit	32 Bit	WDT	1		
		●	●		WDTP	1		

Funktionsweise

Mit Hilfe der WDT-Anweisung können Sie längere Programme in einzelne Programmabschnitte unterteilen. Die Programmzykluszeit wird für jeden einzelnen Programmabschnitt von der Steuerung ermittelt (Watch-Dog-Timer wird nach jedem Programmabschnitt aufgefrischt). Mit der WDT-Anweisung ist es möglich, Programme zu verarbeiten, deren Programmzykluszeit 200 ms überschreitet.

Beschreibung

- Die WDT-Anweisung müssen Sie einsetzen, wenn die Programmzykluszeit vom Programmschritt 0 bis zur END- oder FEND-Anweisung **200 ms** überschreitet.
- Die WDT-Anweisung kann auch nach einer Pointer-Markierung programmiert werden, wenn sich diese im Programm vor der zugehörigen Sprunganweisung (CJ-Anweisung) befindet (siehe Abschnitt 6.2.1).
- Die WDT-Anweisung kann auch innerhalb einer FOR-NEXT-Schleife eingesetzt werden (siehe Abschnitt 6.2.7).

HINWEIS

Der Watch-Dog-Timer wird bei jeder Ausführung der Anweisung END, FEND oder WDT aufgefrischt.

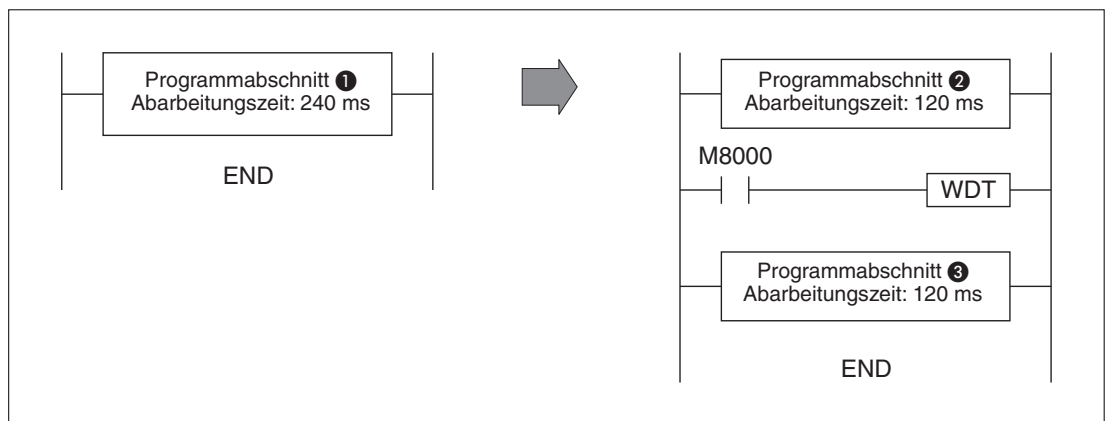


Abb. 6-15: Programmierbeispiel zum Einsatz der WDT-Anweisung

Die Abarbeitungszeit für den Programmabschnitt 1 überschreitet den Wert von 200 ms. Der Programmabschnitt 1 wurde deshalb durch Einsatz der WDT-Anweisung in zwei Programmabschnitte (2, 3) aufgeteilt. Die Programmabschnitte 2 und 3 benötigen jeweils nur 120 ms Abarbeitungszeit. △

Programmzykluszeitwert im Sonderregister D8000 verändern

Überschreitet die Programmzykluszeit wiederholt den Wert von 200 ms, können Sie den Wert der maximal zulässigen Programmzykluszeit im Sonderregister D8000 verändern.

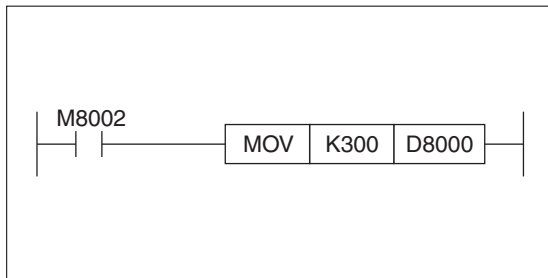


Abb. 6-16:

Programmierbeispiel zur Einstellung der maximal zulässige Programmzykluszeit im Datenregister D8000 auf den Wert 300 ms.

6.2.7 Programmteile wiederholen (FOR, NEXT)

		FOR		FNC 08			
		Anfang einer Programmwiederholung					
Operanden S K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②		Puls-Anweisung (P)		Verarbeitung		Programmschritte	
				16 Bit	32 Bit	FOR	
			●	●	●	●	●

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

		NEXT		FNC 09			
		Ende einer Programmwiederholung					
Operanden S —		Puls-Anweisung (P)		Verarbeitung		Programmschritte	
				16 Bit	32 Bit	NEXT	
			●	●	●	●	●

Funktionsweise

Programmieren von Programmwiederholungen (Programmschleifen)

Beschreibung

- Der Programmteil zwischen der FOR- und der NEXT-Anweisung wird n Mal wiederholt. Anschließend werden die Programmschritte nach der NEXT-Anweisung ausgeführt.
- Der Wert n muss innerhalb des folgenden Bereichs liegen: n: +1 bis +32 767. Wird für n ein Wert zwischen 0 und -32 767 angegeben, wird die FOR-NEXT-Schleife nur einmal abgearbeitet.
- Sie können bis zu fünf FOR-NEXT-Verzweigungsebenen programmieren.

HINWEIS | FOR- und NEXT-Anweisungen dürfen nur paarweise programmiert werden. Zu jeder FOR-Anweisung muss eine passende NEXT-Anweisung programmiert werden.

Fehlerquellen

In den folgenden Fällen tritt ein Programmablauffehler auf:

- Eine NEXT-Anweisung wurde vor einer FOR-Anweisung programmiert.
- Eine NEXT-Anweisung wurde nach der FEND- oder END-Anweisung programmiert.
- Die Anzahl der NEXT-Anweisungen stimmt nicht mit der Anzahl der FOR-Anweisungen überein.

Beispiel ▾ | Einsatz der FOR-, NEXT-Anweisungen

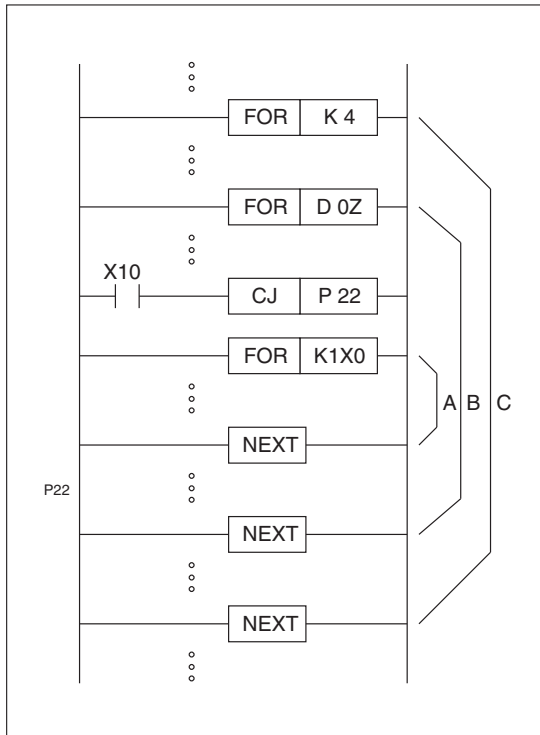


Abb. 6-17:
 Programmierbeispiel zum Einsatz der FOR-,
 NEXT-Anweisungen

In dem Beispiel wurden drei ineinander verschachtelte FOR-NEXT-Verzweigungsebenen programmiert.

- Der Programmabschnitt C wird viermal abgearbeitet. Anschließend werden die Programmschritte nach der dritten NEXT-Anweisung ausgeführt.
- Bei jeder Ausführung des Abschnitts C wird der Programmabschnitt B sechsmal abgearbeitet, wenn im Datenregister D0Z der Wert 6 steht.
- Der Abschnitt B wird demnach 24 Mal abgearbeitet.
- Wenn der Eingang X10 eingeschaltet ist, wird die FOR-NEXT-Schleife A mit Hilfe der CJ-Anweisung übersprungen.
- Wenn X10 ausgeschaltet ist und der Inhalt von K1X0 gleich 7 ist, wird bei jeder Ausführung des Abschnitts B der Programmabschnitt A siebenmal abgearbeitet.
- Der Abschnitt A wird demnach insgesamt 168 (4 x 6 x 7) Mal abgearbeitet.



6.3 Vergleichs- und Transferanweisungen

Übersicht der Anweisungen FNC 10 bis 19

Symbol	FNC	Bedeutung	Abschnitt
CMP	10	Nummerische Daten vergleichen	6.3.1
ZCP	11	Nummerische Datenbereiche vergleichen	6.3.2
MOV	12	Datentransfer	6.3.3
SMOV	13	Shift-Transfer	6.3.4
CML	14	Kopieren und invertieren	6.3.5
BMOV	15	Block-Transfer	6.3.6
FMOV	16	Transfer von gleichen Daten	6.3.7
XCH	17	Austausch von Daten	6.3.8
BCD	18	BCD-Konvertierung	6.3.9
BIN	19	Binär-Konvertierung	6.3.10

Tab. 6-10: Übersicht der Anweisungen FNC 10 bis 19

6.3.1 Numerische Daten vergleichen (CMP, DCMP)

			CMP		FNC 10				
			Numerische Daten vergleichen						
			CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
				●	●	●	●	●	●
Operanden	S1+, S2+		D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②		Y, M, S, D□.b ^②		●	16 Bit	32 Bit	CMP/CMPP	7
						●	●	DCMP/DCMPP	13

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktionsweise

Vergleich zwischen zwei numerischen Datenwerten (größer, kleiner, gleich)

Beschreibung

- Die Daten in den beiden Quellen (S1+) und (S2+) werden miteinander verglichen.
- Das Ergebnis des Vergleichs (größer, gleich, kleiner) wird durch Setzen eines Merkers M, Schrittstatusoperanden S oder Ausgangs Y, bei FX3U und FX3UC auch durch Setzen eines Bits in einem Datenregister, angezeigt. Die Festlegung, welcher Operand gesetzt werden soll, erfolgt mit der Zieladresse (D+).

$(S1+) > (S2+) \rightarrow (D+)$

$(S1+) = (S2+) \rightarrow ((D+)+1)$

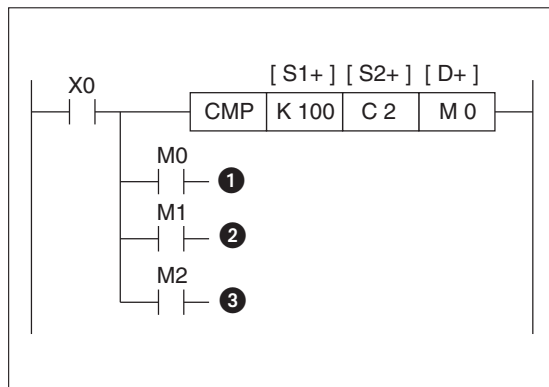
$(S1+) < (S2+) \rightarrow ((D+)+2)$

- Die Daten in S1+ und S2+ werden als binäre Daten behandelt.

Fehlerquellen

- Die CMP-Anweisung erfordert die Angabe von drei Operanden. Werden weniger Operanden angegeben, tritt der Fehler mit dem Fehlercode 6503 auf. Die Ausführung der Anweisung wird unterbrochen.
- Wird ein nicht zulässiger Operand programmiert, wird ein Fehler mit dem Fehlercode 6705 ausgegeben. Als Zieladresse dürfen z. B. keine Eingänge X, Datenregister D, Timer T oder Counter C angegeben werden.
- Wenn der angegebene Operand den für ihn zulässigen Bereich überschreitet, wird ein Fehler mit dem Fehlercode 6706 ausgegeben. Fehler dieser Art treten z. B. bei zu großen Zahlenwerten in Verbindung mit der Index-Adressierung auf.

Eine detaillierte Beschreibung der Fehlercodes enthält Kapitel 10.

Beispiel ▾ Einsatz der CMP-Anweisung**Abb. 6-18:**

Programmierbeispiel zum Einsatz der CMP-Anweisung

In der Zieladresse (D+) ist im Beispiel der Merker M0 angegeben. Entsprechend dem Ergebnis des Vergleichs werden demnach die Merker M0, M1 und M2 wie folgt geschaltet:

- ① M0: EIN, wenn $K100 > \text{Istwert von C2}$
- ② M1: EIN, wenn $K100 = \text{Istwert von C2}$
- ③ M2: EIN, wenn $K100 < \text{Istwert von C2}$

M0, M1, und M2 bleiben unverändert, wenn die Eingangsbedingung X0 ausgeschaltet ist.

△

6.3.2 Numerische Datenbereiche vergleichen (ZCP, DZCP)

			ZCP		FNC 11				
			Numerische Datenbereiche vergleichen						
			CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
				●	●	●	●	●	●
Operanden	S1+, S2+, S3+		D+	Puls-Anweisung (P)		Verarbeitung		Programmschritte	
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②		Y, M, S, D□.b ^②	●		16 Bit	32 Bit	ZCP/ZCPP	9
						●	●	DZCP/DZCPP	17

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktionsweise

Vergleich eines numerischen Datenwertes mit einem numerischen Datenbereich (größer, kleiner, gleich)

Beschreibung

- Die Daten in der Quelle (S3+) werden mit den Daten in den beiden Quellen (S1+) und (S2+) verglichen.
- Das Ergebnis des Vergleichs (größer, kleiner, gleich) wird durch Setzen eines Merkers M, Schrittstatusoperanden S oder Ausgangs Y, bei FX3U und FX3UC auch durch Setzen eines Bits in einem Datenregister, angezeigt. Die Festlegung, welcher Operand gesetzt werden soll, erfolgt mit der Zieladresse (D+).

$(S1+) > (S3+) \rightarrow (D+)$

$(S1+) < (S3+) < (S2+) \rightarrow ((D+)+1)$

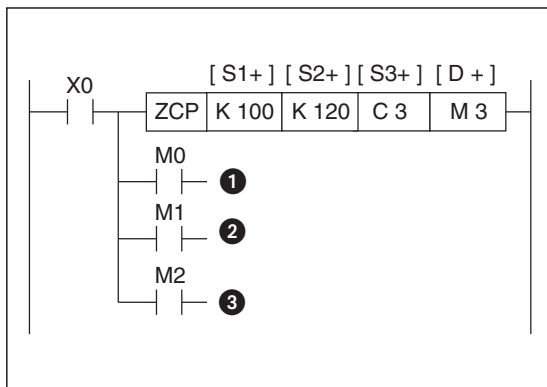
$(S2+) < (S3+) \rightarrow ((D+)+2)$

- Die Daten in (S1+) dürfen nicht größer sein als die Daten in (S2+).

Beispiel ▾

Steht in (S1+) der Wert „K100“ und in (S2+) der Wert „K90“, wird bei der Ausführung der ZCP-Anweisung davon ausgegangen, dass sich in (S2+) ebenfalls der Wert „K100“ befindet.

△

Beispiel ▾ Einsatz der ZCP-Anweisung**Abb. 6-19:**

Programmierbeispiel zum Einsatz der ZCP-Anweisung

In der Zieladresse (D+) ist im Beispiel der Merker M3 angegeben. Entsprechend dem Ergebnis des Vergleichs werden demnach die Merker M3, M4 und M5 wie folgt geschaltet:

- ① M3: EIN, wenn $K100 > \text{Istwert von } C3$
- ② M4: EIN, wenn $K100 \leq \text{Istwert von } C3 \leq K120$
- ③ M5: EIN, wenn $\text{Istwert von } C3 > K120$

M3, M4 und M5 bleiben unverändert, wenn die Eingangsbedingung X0 ausgeschaltet ist.

Wenn sich der Istwert des Zählers C3 im Bereich von 100 bis 120 befindet, wird der Merker M4 eingeschaltet.

△

6.3.3 Datentransfer (MOV, DMOV)

			MOV		FNC 12				
			Datentransfer						
Operanden	S+		D+		Puls-Anweisung (P)	Verarbeitung		Programmschritte	
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□\G□ ^②		KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□\G□ ^②		●	16 Bit	32 Bit	MOV/MOVP	5
						●	●	DMOV/DMOVP	9

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktionsweise

Übertragen von Daten aus einer Datenquelle zu einem Datenziel

Beschreibung

- Die Anweisung dient zur Übertragung von Daten aus einer Datenquelle (S+) zu einem Datenziel (D+).
- Die Daten in der Datenquelle (S+) werden automatisch bei der Ausführung der MOV-Anweisung als binärer Wert interpretiert.

HINWEIS

Die Anweisungen werden in jedem Programmzyklus ausgeführt. Dies können Sie durch Einsatz einer vorgeschalteten Impulsfunktion (PLS- oder PLF-Anweisung bzw. Parameter P) verhindern.

Beispiel ▽

Einsatz der MOV-Anweisung

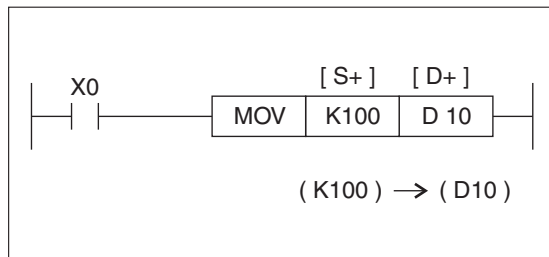


Abb. 6-20:

Transfer des Wertes 100 in das Datenregister D10

Ist die Eingangsbedingung X0 eingeschaltet, findet eine Übertragung der Daten von (S+) nach (D+) statt. Ist X0 ausgeschaltet, findet kein Datentransfer statt.

Die Konstante K100 wird bei der Ausführung der MOV-Anweisung automatisch als binärer Wert interpretiert. △

6.3.4 Shift-Transfer (SMOV)

				SMOV		FNC 13				
				Shift-Transfer						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
							●	●	●	●
Operanden	S+	D+	n, m1, m2	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z 0-9999, U□G□ ^②	K, H, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②	K, H 0-4		●	16 Bit	32 Bit	SMOV, SMOVP	11	

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktionsweise

Transfer von Daten und Verändern der Wertigkeit

Beschreibung

- Die Anweisung wird in 5 Schritten abgearbeitet:
 - 1.) Lesen der binären Daten aus (S+)
 - 2.) Wandeln der Daten in das BCD-Format
 - 3.) Verschieben der BCD-Stellen
 - 4.) Wandeln der Daten in das BIN-Format
 - 5.) Schreiben der binären Daten nach (D+)
- n, m1, m2 legen die Art der Verschiebung der BCD-Stellen fest.
 - m1 = 1. Stelle, die verschoben werden soll
 - m2 = Anzahl der Stellen, die verschoben werden sollen
 - n = 1. Zieladresse

Beispiel ▾

SMOV-Anweisung ohne Sondermerker

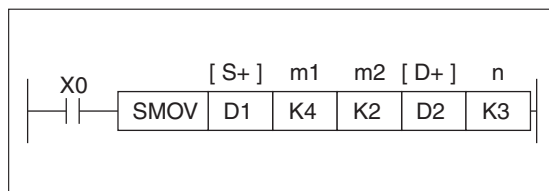


Abb. 6-21

Programmierbeispiel zum Einsatz der SMOV-Anweisung ohne Sondermerker

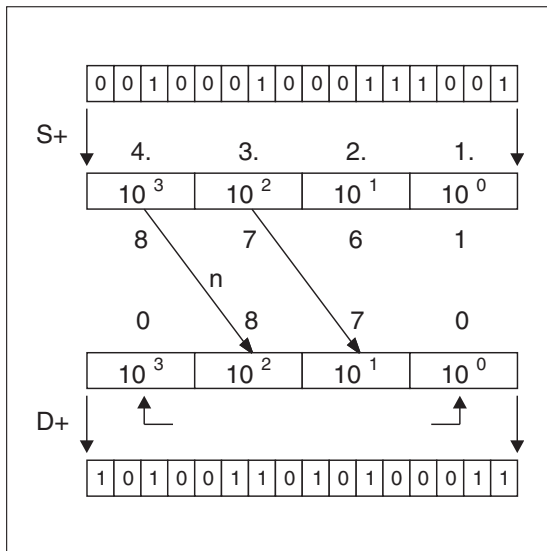


Abb. 6-23:
Konvertierung und Transfer

Beispiel ▾

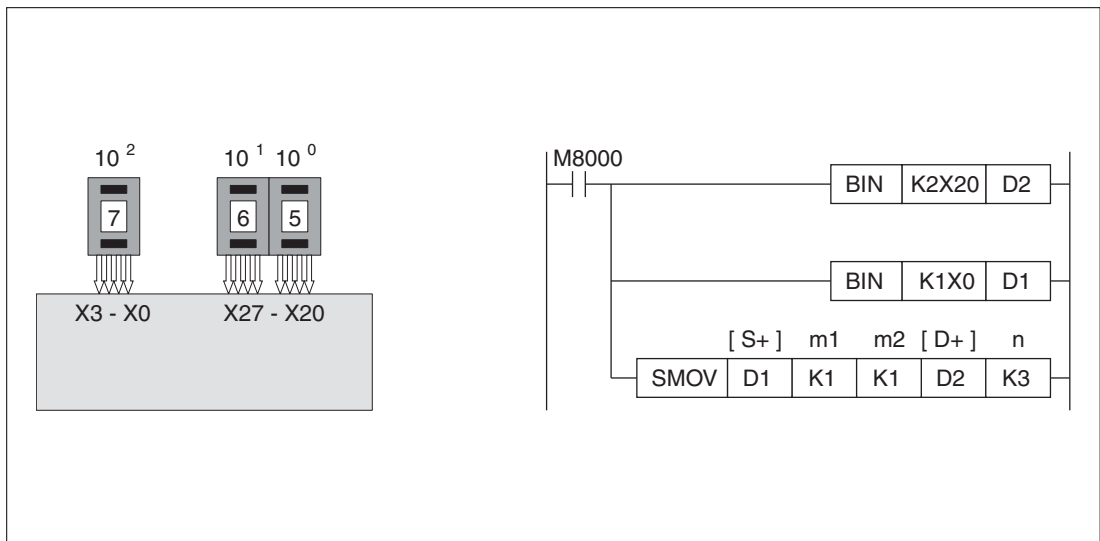


Abb. 6-22: Programmierbeispiel für Binäreingaben

Die Eingangsdaten werden von drei BCD-Schaltern geliefert, die entsprechenden Eingängen der Steuerung zugeordnet sind.

Zunächst werden die BCD-Daten der Eingänge X20 bis X27 (K2X20) in binäre Daten konvertiert und in D2 abgelegt.

Ebenso werden die Daten der Eingänge X0 bis X3 (K1X0) in binäre Daten umgewandelt. Das Ergebnis wird in D1 abgelegt.

Der BCD-Wert des Datenregisters D1 wird an die dritte Position im Zielregister D2 geschrieben. Anschließend werden die BCD-Daten wieder in binäre Daten umgewandelt.

Im Beispiel werden die numerischen Eingangsdaten der drei BCD-Schalter zusammengefasst und im Datenregister D2 als binäre Daten gespeichert.



Funktionsweise mit Sondermerker M8168

Transfer von Daten im Hexadezimal-Format und Verändern der Wertigkeit

Beschreibung

- Die SMOV-Anweisung muss mit einer OUT-Anweisung kombiniert werden.
- Die Anweisung wird in 3 Schritten abgearbeitet:
 - 1.) Lesen der hexadezimalen Daten aus (S+), max. 4 Stellen, max. FFFFH
 - 2.) Verschieben der Stellen
 - 3.) Schreiben der Daten nach (D+)
- n, m1, m2 legen die Art der Verschiebung der Stellen fest.
 - m1 = 1. Stelle, die verschoben werden soll
 - m2 = Anzahl der Stellen, die verschoben werden sollen
 - n = 1. Zieladresse

Beispiel ▾

Einsatz der SMOV-Anweisung mit Sondermerker M8168

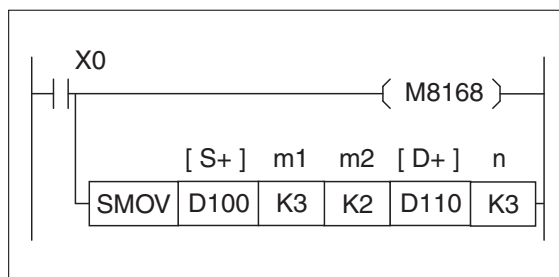


Abb. 6-24:
 Programmierbeispiel zum Einsatz der SMOV-Anweisung

In Datenregister D100 ist die Zahl FFE2H abgelegt, und im Zielregister D110 ist die Zahl 2CD9H abgelegt. Wird X0 gesetzt, wird die SMOV-Anweisung ausgeführt.

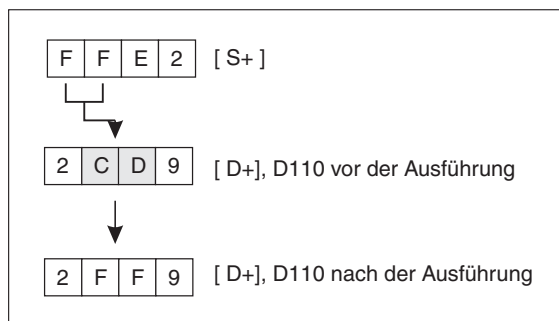


Abb. 6-25:
 Verschiebung in den Datenregistern

Die Ziffern FFH, 1. Stelle Nr. 4, 2 Stellen, von dem Wert in D100 werden kopiert und nach D110 verschoben. Dadurch werden die Ziffern CDH, 1. Zieladresse Nr. 3, mit den Ziffern FFH überschrieben.



6.3.5 Kopieren und invertieren (CML)

		CML		FNC 14			
		Kopieren und invertieren					
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte	
				16 Bit	32 Bit	CML, CMLP	5
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②	KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②	●	●	●	●	●
				●	●	DCML, DCMLP	9

- ① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ② Nur bei FX3U und FX3UC

Funktionsweise

Bilden des 1er-Komplements einer binären Zahl

Beschreibung

Der binäre Zahlenwert in (S+) wird in sein 1er-Komplement gewandelt und nach (D+) geschrieben.

HINWEIS

Wenn die Zieladresse mehr Bits hat als die Quelladresse, werden alle nicht genutzten Bits auf „1“ gesetzt.

Beispiel ▾

CML-Anweisung

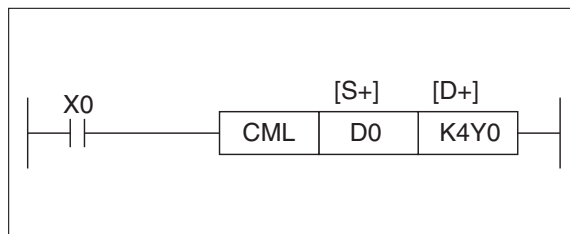


Abb. 6-26: Programmierbeispiel zur CML-Anweisung

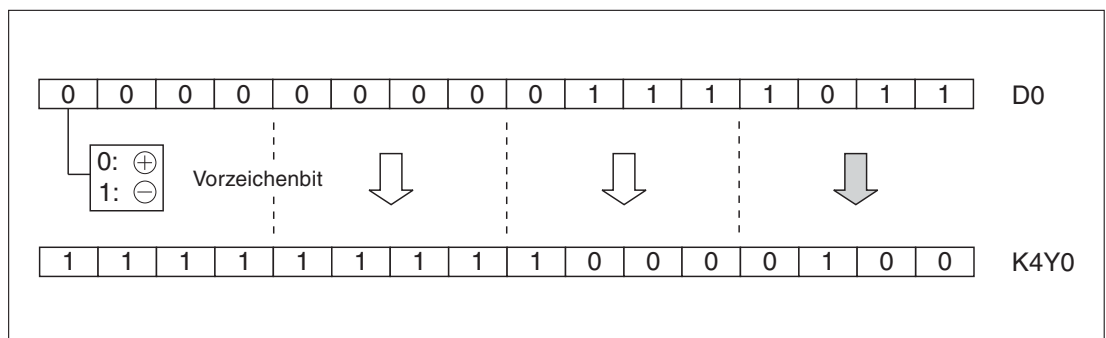


Abb. 6-27: Invertierung und Transfer



6.3.6 Block-Transfer (BMOV)

				BMOV		FNC 15				
				Block-Transfer						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
					●	●	●	●	●	●
Operanden	S+		D+	n	Puls-Anweisung (P)		Verarbeitung		Programmschritte	
	KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□\G□ ^②		KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□\G□ ^②	K, H	●	●	●	●	BMOV/ BMOV P	7

- ① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ② Nur bei FX3U und FX3UC

Funktionsweise

Transferieren von Datenpaketen

Beschreibung

- Es wird eine vorgegebene Menge von Wortoperanden transferiert.
- Für den Transfer werden die Startadresse (S+), die Zieladresse (D+) und die Anzahl der zu transferierenden Worte (n) vorgegeben.
- Mit der BMOV-Anweisung können File-Register der MELSEC FX-Familie gelesen und geschrieben werden.

HINWEISE

Wenn die Größe des Datenpakets die Größe des Ziel- oder des Quellbereichs überschreitet, werden nur die in den Bereich passenden Worte übertragen.

Mit dem Sondermerker M8024 kann bei allen Steuerungen, außer bei der FX1S-Serie, die Transferrichtung der BMOV-Anweisung geändert werden.

Beispiel ▾

BMOV-Anweisung

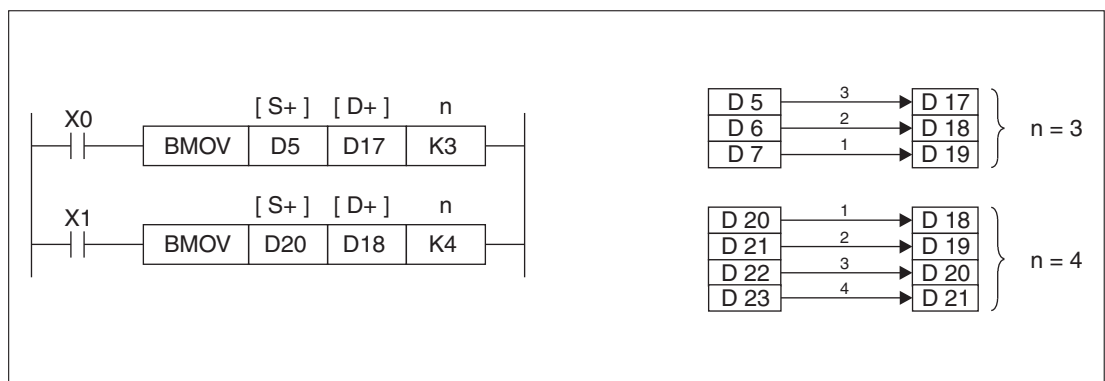


Abb. 6-28: Block-Transfer



6.3.7 Transfer von gleichen Daten (FMOV)

				FMOV		FNC 16				
				Transfer von gleichen Daten						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
							●	●	●	●
Operanden	S+		D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②		KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②	K, H	●	16 Bit	32 Bit	FMOV/ FMOVP	7	
					●	●	DFMOV/ DFMOVP	13		

- ① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ② Nur bei FX3U und FX3UC

Funktionsweise

Übertragen eines Datenwertes in mehrere Zieloperanden

Beschreibung

- Der Dateninhalt von (S+) wird in mehrere Zieloperanden gleichen Typs übertragen.
- Der erste Zieloperand wird in (D+) festgelegt.
- Ausgehend vom Zieloperanden (D+) wird der Datenwert aus (S+) in n Operanden übertragen.

HINWEIS

Wenn n größer als die Anzahl verfügbarer Operanden ist, erfolgt die Übertragung nur bis zum letzten verfügbaren Operanden.

Beispiel ▾

FMOV-Anweisung

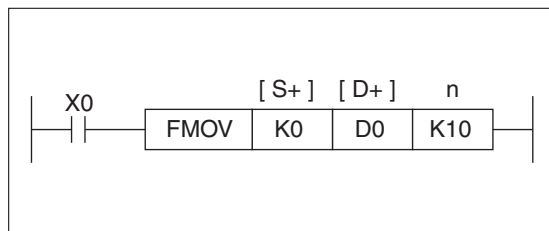


Abb. 6-29:
Programmierbeispiel zur FMOV-Anweisung

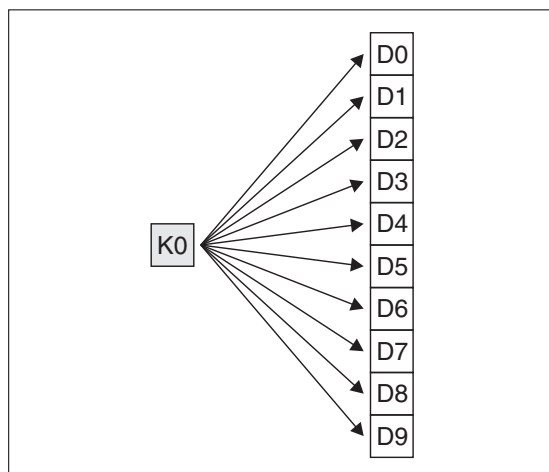
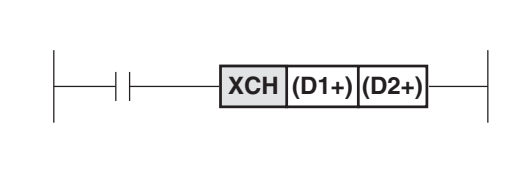


Abb. 6-30:
Transfer des Wertes „0“ in die Datenregister D0 bis D9

6.3.8 Austausch von Daten (XCH)

		XCH		FNC 17			
		Austausch von Daten					
Operanden	D1+, D2+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	KnY, KnM, KnS T, C, D, R*, V, Z, U□\G□*	●	16 Bit	32 Bit	XCH/XCHP	5	
			●	●	DXCH/DXCHP	9	

* Nur bei FX3U und FX3UC

Funktionsweise

Austausch von Daten zwischen zwei Operanden

Beschreibung

Die Daten von (D1+) und (D2+) werden ausgetauscht.

HINWEIS | Wenn keine Flankensteuerung programmiert ist, wird der Austauschvorgang in jedem Zyklus ausgeführt.

Beispiel ▾ XCH-Anweisung ohne Sondermerker M8160

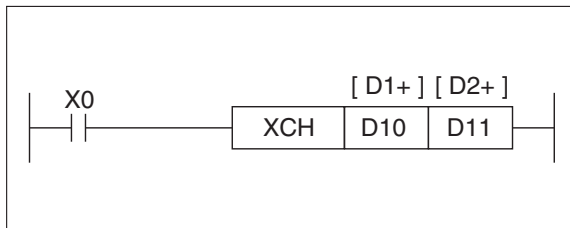


Abb. 6-31:
Programmierbeispiel zur XCH-Anweisung

Werte vor der Ausführung: D10 = 5, D11 = 7

Werte nach der Ausführung: D10 = 7, D11 = 5

△

Funktionsweise mit Sondermerker M8160

Bei gesetztem Sondermerker M8160 erfolgt ein Austausch der oberen und unteren Bytes in (D1+) und (D2+).

Beschreibung

Nach Setzen des Sondermerkers M8160 werden in (D1+) und in (D2+) das obere Byte und das untere Byte ausgetauscht. Wird die XCH-Anweisung ohne Sondermerker M8160 noch einmal in dem Programm verwendet, muss der Sondermerker M8160 wieder zurückgesetzt werden.

- HINWEIS** | Bei Verwendung der XCH-Anweisung mit Sondermerker M8160 müssen (D1+) und (D2+) das gleiche Datenregister angeben, andernfalls kommt es zu einer Fehlermeldung (Der Sondermerker M8067 wird gesetzt).
- | Der Austauschvorgang wird in jedem Zyklus ausgeführt, wenn keine Flankensteuerung programmiert wird.

Beispiel ▽ DXCH-Anweisung mit Sondermerker M8160

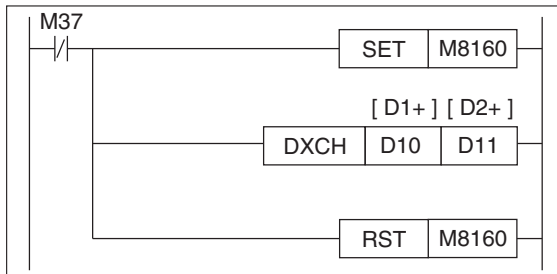


Abb. 6-32:
Programmierbeispiel zur DXCH-Anweisung mit gesetztem Sondermerker M8160

Der Austauschvorgang lässt sich wie folgt darstellen:

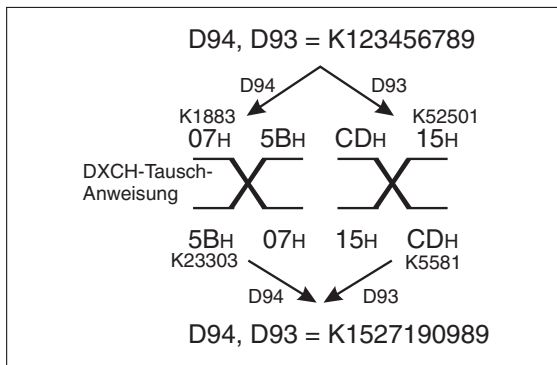


Abb. 6-33:
Darstellung des Austauschvorgangs



- HINWEIS** | Bei Anwendung der 32-Bit-Anweisung DXCH/DXCHP erfolgt der Austausch des oberen und unteren Bytes unabhängig in jedem Einzelwort (16 Bit).

6.3.9 BCD-Konvertierung (BCD, DBCD)

			BCD		FNC 18					
			BCD-Konvertierung							
			CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
				●	●	●	●	●	●	
Operanden	S+		D+		Puls-Anweisung (P)		Verarbeitung		Programmschritte	
	KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□\G□ ^②		KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□\G□ ^②		●		16 Bit	32 Bit	BCD/BCDP	5
							●	●	DBCD/DBCDP	9

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktionsweise

Konvertierung von binären Daten in das BCD-Format

Beschreibung

Innerhalb der Steuerung werden nur binäre Daten verarbeitet. Durch Einsatz der BCD-Anweisung können auch Daten im BCD-Format ausgegeben werden (z. B. zum Ansteuern einer 7-Segment-Anzeige).

- Die binären Daten in der Quelle (S+) werden in BCD-Daten konvertiert und zur Zieladresse (D+) übertragen.
- Das Ergebnis der BCD-Konvertierung muss innerhalb des zulässigen Bereiches liegen:
 16-Bit-Anweisung: 0 bis +9 999
 32-Bit-Anweisung: 0 bis +99 999 999

Fehlerquelle

Liegt das Ergebnis der BCD-Konvertierung außerhalb des zulässigen Bereichs, tritt ein Programmablauffehler auf, und die Anweisung wird nicht ausgeführt.

Beispiel ▾

Die BCD-Anweisung können Sie beispielsweise einsetzen, um binäre Daten aus der SPS zu lesen und auf einer 7-Segment-Anzeige darzustellen.

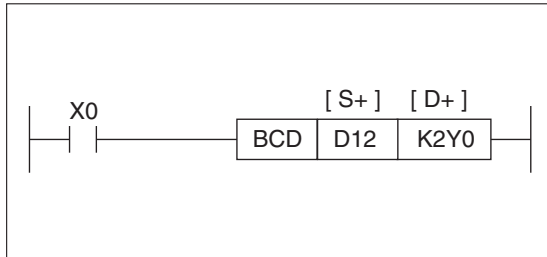


Abb. 6-34:
 Programmierbeispiel zum Einsatz der BCD-Anweisung

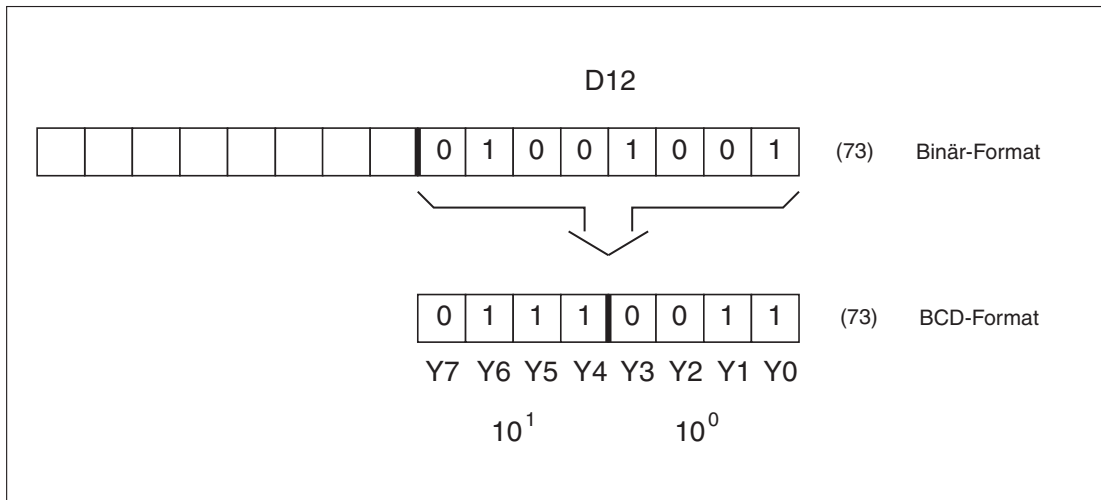


Abb. 6-35: Programmierbeispiel zur Konvertierung von binären Daten in das BCD-Format

Die binären Daten aus dem Datenregister D12 werden in ein BCD-Format konvertiert und danach über die Ausgänge Y0 bis Y7 ausgegeben. In diesem Beispiel: 73 (dezimal).



6.3.10 Binär-Konvertierung (BIN, DBIN)

			BIN		FNC 19				
			Binär-Konvertierung						
			CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
				●	●	●	●	●	●
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□\G□ ^②	KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□\G□ ^②	●	16 Bit	32 Bit	BIN/BINP	5		
				●	●	DBIN/DBINP	9		

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktionsweise

Konvertierung von BCD-Daten in das binäre Format

Beschreibung

Innerhalb der Steuerung werden nur binäre Daten verarbeitet. Durch Einsatz der BIN-Anweisung können auch Daten im BCD-Format über die Eingänge eingelesen werden.

- Die BCD-Daten in der Quelle (S+) werden in binäre Daten konvertiert und zur Zieladresse (D+) übertragen.
- Die Daten in (S+) müssen innerhalb des zulässigen Bereiches liegen:
 16-Bit-Anweisung: 0 bis +9 999
 32-Bit-Anweisung: 0 bis +99 999 999

Fehlerquelle

Sind die Daten in (S+) nicht im BCD-Format, tritt ein Fehler auf. Der Fehler wird durch den eingeschalteten Sondermerker M8067 angezeigt. Der Sondermerker M8068 wird nicht eingeschaltet (siehe auch Abschnitt 9.1.7).

Beispiel ▾

Die BIN-Anweisung können Sie z. B. einsetzen, um BCD-Daten von Schaltern in die SPS zu übertragen.

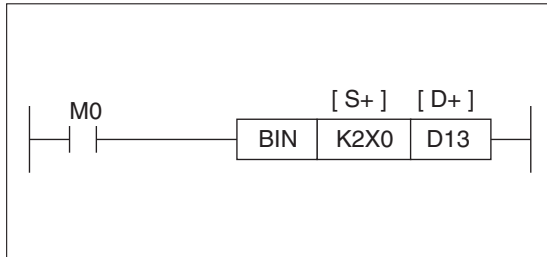


Abb. 6-36:
 Programmierbeispiel für den Einsatz der BIN-Anweisung

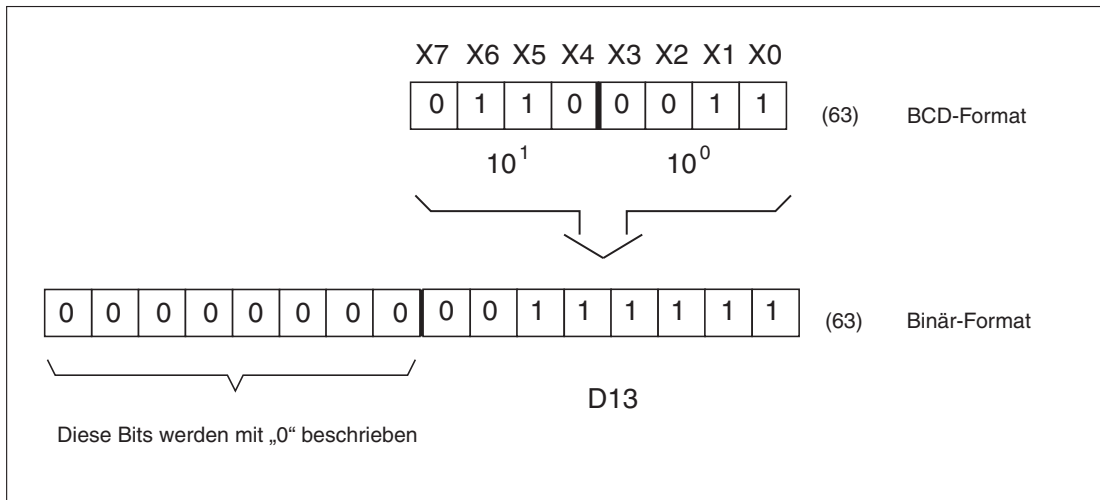


Abb. 6-37: Programmierbeispiel für die Konvertierung von Daten im BCD-Format in das binäre Format

Die BCD-Daten an den Eingängen X0 bis X7 werden in das binäre Format konvertiert. Anschließend werden die Daten zur Zieladresse D13 übertragen.



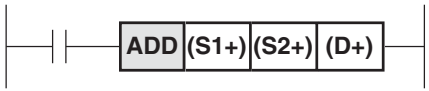
6.4 Arithmetische Anweisungen

Übersicht der Anweisungen FNC 20 bis 29

Symbol	FNC	Bedeutung	Abschnitt
ADD	20	Addition numerischer Daten	6.4.1
SUB	21	Subtraktion numerischer Daten	6.4.2
MUL	22	Multiplikation numerischer Daten	6.4.3
DIV	23	Division numerischer Daten	6.4.4
INC	24	Inkrementieren	6.4.5
DEC	25	Dekrementieren	6.4.6
WAND	26	logische UND-Verknüpfung	6.4.7
WOR	27	logische ODER-Verknüpfung	6.4.8
WXOR	28	logische Exklusiv-ODER-Verknüpfung	6.4.9
NEG	29	Negation von Daten	6.4.10

Tab. 6-11: Übersicht der Anweisungen FNC 20 bis 29

6.4.1 Addition numerischer Daten (ADD, DADD)

		ADD	FNC 20					
		Addition numerischer Daten						
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
			●	●	●	●	●	●
Operanden	S+, S2+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②	KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②	●	16 Bit	32 Bit	ADD/ADDP	7	
				●	●	DADD/DADDP	13	

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktionsweise

Addition von zwei numerischen Daten. Das Ergebnis der Addition wird in einer Zieladresse gespeichert.

Beschreibung

- Die binären Daten in den Quelladressen (S1+) und (S2+) werden addiert. Das Ergebnis der Addition wird in (D+) abgespeichert.
(S1+) + (S2+) = (D+)
- In dem höchstwertigen Bit wird das Vorzeichen der Addition gespeichert:
Bit = 0: positives Vorzeichen
Bit = 1: negatives Vorzeichen
- Bei der Ausführung einer 32-Bit-Anweisung wird der Wortoperand der unteren 16 Bit in der Anweisung angegeben. Der darauffolgende Operand ist der Wortoperand der oberen 16 Bit. Es wird empfohlen, bei der Angabe der Adressen gerade Zahlen zu verwenden, damit nicht versehentlich überschneidende Adressen programmiert werden.
- In der Quelladresse (S+) und der Zieladresse (D+) kann auch der gleiche Operand angegeben werden.

HINWEISE

Bei bestimmten Rechenergebnissen wird nach Ausführung der Anweisung ein Sondermerker (Flag) gesetzt.

Zero Flag M8020

Lautet das Ergebnis der Addition 0, wird das Zero Flag gesetzt.

Borrow Flag M8021

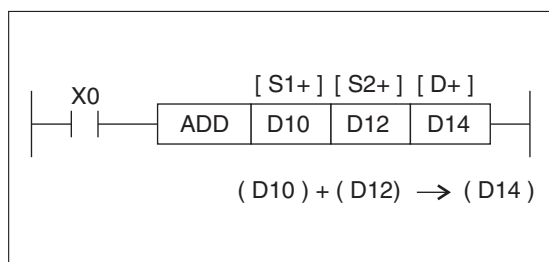
Unterschreitet das Ergebnis der Addition -32 767 (16-Bit-Operation) bzw. -2 147 483 648 (32-Bit-Operation), wird das Borrow Flag gesetzt.

Carry Flag M8022

Überschreitet das Ergebnis den Wert +32 767 (16-Bit-Operation) bzw. +2 147 483 647 (32-Bit-Operation), wird das Carry Flag gesetzt.

Beispiele ▾

Einsatz der ADD-Anweisung

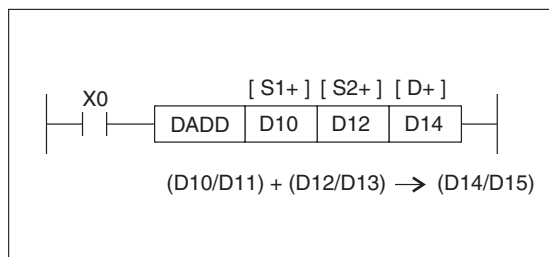
**Abb. 6-38:**

Programmierbeispiel mit einer ADD-Anweisung

Ist X0 eingeschaltet, werden die Datenwerte in den Registern D10 und D12 addiert. Das Ergebnis der Addition wird im Datenregister D14 abgespeichert.

△

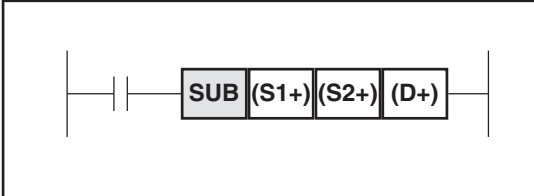
Einsatz der DADD-Anweisung

**Abb. 6-39:**

Programmierbeispiel mit einer DADD-Anweisung

△

6.4.2 Subtraktion numerischer Daten (SUB, DSUB)

			SUB		FNC 21			
			Subtraktion numerischer Daten					
CPU			FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
			●	●	●	●	●	●
Operanden	S+, S2+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□\G□ ^②	KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□\G□ ^②		●	16 Bit	32 Bit	SUB/SUBP	7
				●	●	DSUB/DSUBP	13	

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktionsweise

Subtraktion zweier numerischer Daten

Das Ergebnis der Subtraktion wird in einer Zieladresse abgespeichert.

Beschreibung

- Der Datenwert in (S2+) wird von dem Datenwert in (S1+) subtrahiert. Das Ergebnis der Subtraktion wird in (D+) abgelegt.
 $(S1+) - (S2+) = (D+)$
- In dem höchstwertigen Bit wird das Vorzeichen der Addition gespeichert:
Bit = 0: positives Vorzeichen
Bit = 1: negatives Vorzeichen
- Bei der Ausführung einer 32-Bit-Anweisung wird der Wortoperand der unteren 16 Bit in der Anweisung angegeben. Der darauffolgende Operand ist der Wortoperand der oberen 16 Bit. Es wird empfohlen, bei der Angabe der Adressen gerade Zahlen zu verwenden, damit nicht versehentlich dieselben Adressen programmiert werden.
- In der Quelladresse (S+) und der Zieladresse (D+) kann auch der gleiche Operand angegeben werden.

HINWEISE

Bei bestimmten Rechenergebnissen wird nach Ausführung der Anweisung ein Sondermerker (Flag) gesetzt.

Zero Flag M8020

Lautet das Ergebnis der Subtraktion 0, wird das Zero Flag gesetzt.

Borrow Flag M8021

Unterschreitet das Ergebnis der Subtraktion -32 767 (16-Bit-Operation) bzw. -2 147 483 648 (32-Bit-Operation), wird das Borrow Flag gesetzt.

Carry Flag M8022

Überschreitet das Ergebnis den Wert +32 767 (16-Bit-Operationen) bzw. +2 147 483 647 (32-Bit-Operationen), wird das Carry Flag gesetzt.

Beispiele ▾ Einsatz der SUB-Anweisung

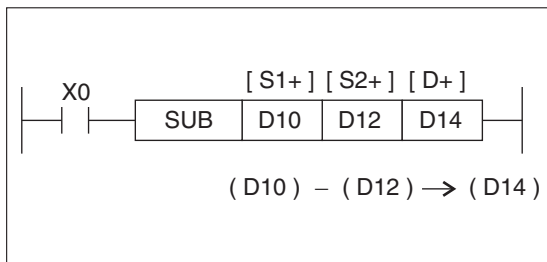


Abb. 6-40:
Programmierbeispiel zum Einsatz der SUB-Anweisung

Ist X0 eingeschaltet, wird der Datenwert im Datenregister D12 von dem Datenwert im Datenregister D10 subtrahiert. Das Ergebnis der Subtraktion wird im Datenregister D14 gespeichert.



Einsatz der DSUB-Anweisung

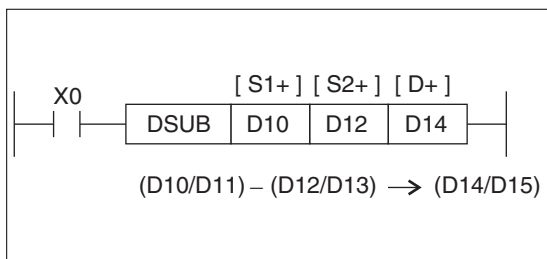


Abb. 6-41:
Programmierbeispiel zum Einsatz der DSUB-Anweisung



6.4.3 Multiplikation numerischer Daten (MUL, DMUL)

			MUL		FNC 22				
			Multiplikation numerischer Daten						
			CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
				●	●	●	●		●
Operanden	S+, S2+		D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②		KnY, KnM, KnS, T, C, D, R ^① , Z (V), U□G□ ^②	●	16 Bit	32 Bit	MUL/MULP	7	
					●	●	DMUL/DMULP	13	

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktionsweise

Multiplikation von zwei numerischen Daten

Das Ergebnis der Multiplikation wird in einer Zieladresse gespeichert.

Beschreibung

- Die Daten in S1+ und S2+ werden miteinander multipliziert. Das Ergebnis der Multiplikation wird in der in D+ angegebenen Operandenadresse und den darauffolgenden Operandenadressen gespeichert.

$$(S1+) \times (S2+) = (D+)$$

- Das höchstwertige Bit zeigt das Vorzeichen des Multiplikationsergebnisses an.

Bit = 0: positives Vorzeichen

Bit = 1: negatives Vorzeichen

- Bei der Ausführung einer 16-Bit-Operation wird das Ergebnis als 32-Bit-Zahl in (D+) und ((D+)+1) abgelegt. Bei der Ausführung einer 32-Bit-Operation wird das Ergebnis als 64-Bit-Zahl in (D+) und den drei darauffolgenden Operanden abgelegt.
- Bei der 32-Bit-Operation kann für (D+) nicht Z(V) verwendet werden.

Multiplikation von 16-Bit-Daten (MUL-Anweisung)

Das Ergebnis einer 16-Bit-Multiplikation ergibt eine 32-Bit-Zahl. Diese Zahl wird als 32-Bit-Datenwert gespeichert. Die unteren 16 Bit werden in der in D+ angegebenen Operandenadresse gespeichert. Die oberen 16 Bit werden in den darauffolgenden Operandenadressen gespeichert.

Beispiel ▾

Einsatz der MUL-Anweisung

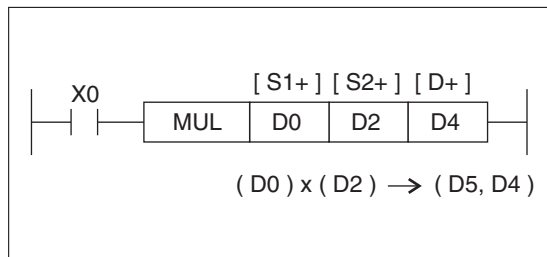


Abb. 6-42:

Programmierbeispiel mit einer MUL-Anweisung

Das Ergebnis der Multiplikation wird als 32-Bit-Datenwert in den Datenregistern D4 und D5 abgelegt. In D4 stehen die unteren 16 Bit, und in D5 stehen die oberen 16 Bit. Das Vorzeichen des Multiplikationsergebnisses steht im 15. Bit von D5.

△

Multiplikation von 32-Bit-Daten (DMUL-Anweisung)

Das Ergebnis einer 32-Bit-Multiplikation wird als 64-Bit-Datenwert gespeichert. Die unteren 16 Bit werden in der in D+ angegebenen Operandenadresse gespeichert. Die höherwertigen Bit werden in den darauffolgenden Operandenadressen gespeichert.

Beispiel ▾

Einsatz der DMUL-Anweisung

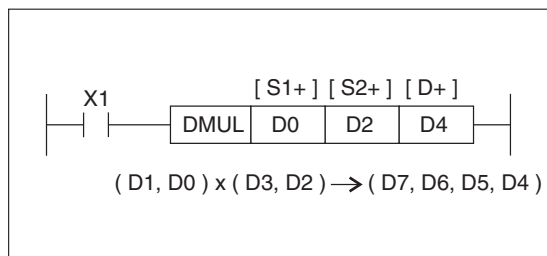


Abb. 6-43:

Programmierbeispiel mit einer DMUL-Anweisung

Das Ergebnis der Multiplikation wird als 64-Bit-Datenwert in den Datenregistern D4, D5, D6 und D7 abgelegt. In D4 stehen die unteren 16 Bit, und in D5, D6 und D7 stehen die höherwertigen Bit.

△

6.4.4 Division numerischer Daten (DIV, DDIV)

			DIV		FNC 23				
			Division numerischer Daten						
			CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
				●	●	●	●	●	●
Operanden	S1+, S2+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②	KnY, KnM, KnS, T, C, D, R ^① , Z (V), U□G□ ^②		●	16 Bit	32 Bit	DIV/DIVP	7	
				●	●	DDIV/DDIVP	13		

- ① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ② Nur bei FX3U und FX3UC

Funktionsweise

Division zwischen zwei numerischen Daten

Das Ergebnis der ganzzahligen Division wird in einer Zieladresse gespeichert.

Beschreibung

- Es findet eine Division zwischen den Daten in (S1+) und den Daten in (S2+) statt. Die Daten in (S1+) entsprechen dem Dividenten. Die Daten in (S2+) entsprechen dem Divisor. Das Ergebnis der Division wird in der in (D+) angegebenen Operandenadresse und in den darauffolgenden Operandenadressen gespeichert. Der Teilungsrest wird in einer der darauffolgenden Operandenadressen gespeichert.

$$(S1+) : (S2+) = (D+)$$

- In dem höchstwertigen Bit wird das Vorzeichen der Division gespeichert.

Bit = 0: positives Vorzeichen
 Bit = 1: negatives Vorzeichen

Das Vorzeichen des Divisionsergebnisses ist abhängig von den Vorzeichen des Dividenten und des Divisors.

- Bei der 32-Bit-Operation kann für (D+) nicht Z(V) verwendet werden.

Divident	Divisor	Quotient	Teilungsrest
+	+	+	+
+	-	-	+
-	+	-	-
-	-	+	-

Tab. 6-12:
Vorzeichen des Divisionsergebnisses

HINWEIS

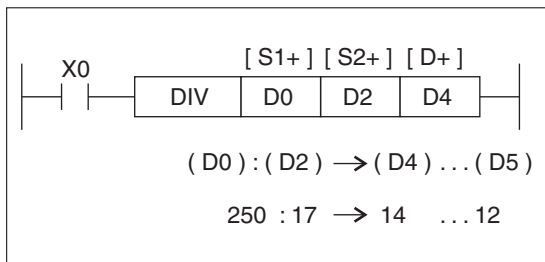
Ist die Zieladresse ein Bit-Operand, kann die Steuerung keinen Teilungsrest ermitteln.

Fehlerquelle

Es kommt zu einem Programmablauffehler, wenn der Wert des Divisors gleich 0 ist. Die Anweisung wird nicht abgearbeitet.

Division von 16-Bit-Daten (DIV-Anweisung)

Das Ergebnis einer 16-Bit-Division wird in der in D+ angegebenen Operandenadresse gespeichert. Der Teilungsrest wird in der darauffolgenden Operandenadresse gespeichert.

Beispiel ▾ Einsatz der DIV-Anweisung**Abb. 6-44:**

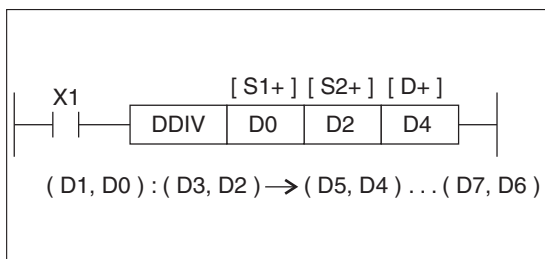
Programmierbeispiel mit einer DIV-Anweisung

Das Ergebnis der Division 14 wird im Datenregister D4 gespeichert. Der Teilungsrest 12 wird im darauffolgenden Datenregister D5 abgelegt.

△

Division von 32-Bit-Daten (DDIV-Anweisung)

Bei einer Division von 32-Bit-Daten stehen für den Dividenten, den Divisor, das Ergebnis und den Teilungsrest jeweils zwei aufeinanderfolgende Datenregister zur Verfügung. In der DDIV-Anweisung müssen Sie jeweils das Datenregister mit der niedrigeren Operandenadresse angeben.

Beispiel ▾ Einsatz der DDIV-Anweisung**Abb. 6-45:**

Programmierbeispiel mit einer DDIV-Anweisung

Das Ergebnis der Division wird in den Datenregistern D4 und D5 gespeichert. Der Teilungsrest wird in den darauffolgenden Datenregistern D6 und D7 gespeichert.

△

6.4.5 Inkrementieren (INC, DINC)

		INC		FNC 24			
		Inkrementieren					
Operanden	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□□ ^②	●	16 Bit	32 Bit	INC/INCP	3	
			●	●	DINC/DINCP	5	

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktionsweise

Zu einem numerischen Datenwert wird die Zahl 1 addiert (inkrementiert).

HINWEIS

Die Anweisung wird in jedem Programmzyklus ausgeführt. Dies können Sie durch Einsatz einer vorgeschalteten Impulsfunktion (PLS- oder PLF-Anweisung) oder den Einsatz des Befehlsparameters P verhindern.

Beim Inkrementieren wird zu dem in D+ gespeicherten Wert die Zahl 1 addiert, sobald die Eingangsbedingung erfüllt ist.

- **16-Bit-Operation (INC-Anweisung)**
Wird bei einer 16-Bit-Operation der Wert 1 zu dem Wert +32 767 addiert, lautet das Ergebnis -32 768. Es wird kein Flag gesetzt.
- **32-Bit-Operation (DINC-Anweisung)**
Wird bei einer 32-Bit-Operation der Wert 1 zu dem Wert +2 147 483 647 addiert, lautet das Ergebnis -2 147 483 648. Es wird kein Flag gesetzt.

Beispiel ▾

Einsatz der INC-Anweisung

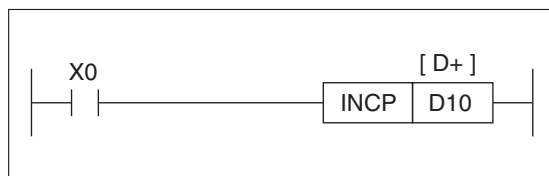


Abb. 6-46:
Programmierbeispiel zum Einsatz der INCP-Anweisung

Der Datenwert im Datenregister D10 wird bei jedem Einschalten des Eingangssignals X0 um den Zahlenwert 1 erhöht.

Die Anweisung wird durch eine vorgeschaltete Impulsfunktion aktiviert. Dies ist wichtig, damit der Additionsvorgang nicht in jedem Programmzyklus stattfindet. △

6.4.6 Dekrementieren (DEC)

		DEC		FNC 25			
		Dekrementieren					
Operanden	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□\G□ ^②	●	16 Bit	32 Bit	DEC/DECP	3	
			●	●	DDEC/DDECP	5	

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktionsweise

Von einem numerischen Datenwert wird die Zahl 1 subtrahiert (dekrementiert).

HINWEIS

Die Anweisung wird in jedem Programmzyklus ausgeführt. Dies können Sie durch Einsatz einer vorgeschalteten Impulsfunktion (PLS- oder PLF-Anweisung) oder den Einsatz des Befehlsparameters P verhindern.

Dekrementieren

Beim Dekrementieren wird die Zahl 1 von dem in D+ gespeicherten Wert subtrahiert, sobald die Eingangsbedingen erfüllt ist.

- **16-Bit-Operation (DEC-Anweisung)**
Wird bei einer 16-Bit-Operation der Wert 1 vom Wert -32 768 subtrahiert, lautet das Ergebnis +32 767. Es wird kein Flag gesetzt.
- **32-Bit-Operation (DDEC-Anweisung)**
Wird bei einer 32-Bit-Operation der Wert 1 vom Wert -2 147 483 648 subtrahiert, lautet das Ergebnis +2 147 483 647. Es wird kein Flag gesetzt.

Beispiel ▾

Einsatz der DEC-Anweisung

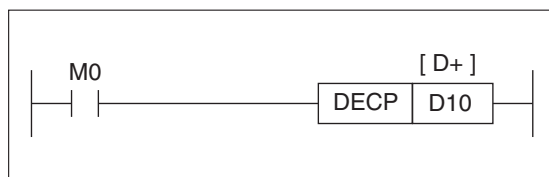
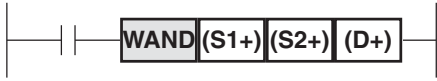


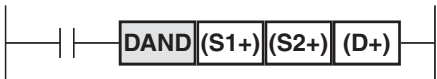
Abb. 6-47:
Programmierbeispiel zum Einsatz der DECP-Anweisung

Der Datenwert im Datenregister D10 wird bei jedem Setzen von M0 um den Zahlenwert 1 vermindert.

Die Anweisung wird durch eine vorgeschaltete Impulsfunktion aktiviert. Dies ist wichtig, damit der Subtraktionsvorgang nicht in jedem Programmzyklus stattfindet. △

6.4.7 Logische UND-Verknüpfung binärer Daten (WAND, DAND)

		WAND		FNC 26						
		Logische UND-Verknüpfung								
Operanden		S1+, S2+		D+		Puls-Anweisung (P)	Verarbeitung		Programmschritte	
		K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②		KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②		●	16 Bit	32 Bit	WAND/WANDP	

		DAND		FNC 26						
		Logische UND-Verknüpfung								
Operanden		S1+, S2+		D+		Puls-Anweisung (P)	Verarbeitung		Programmschritte	
		K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②		KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②		●	16 Bit	32 Bit	DAND/DANDP	

- ① Nur bei FX3G/FX3U/FX3UC
- ② Nur bei FX3U und FX3UC

Funktionsweise

Logische UND-Verknüpfung binärer Daten

Beschreibung

- Es wird eine logische UND-Verknüpfung von einzelnen Bits ausgeführt.
- Die Daten in (S1+) und (S2+) werden bitweise miteinander verknüpft. Das Verknüpfungsergebnis wird in (D+) gespeichert.

(S1+)	(S2+)	(D+)
1	1	1
1	0	0
0	1	0
0	0	0

Tab. 6-13:
Wahrheitstabelle der UND-Verknüpfung

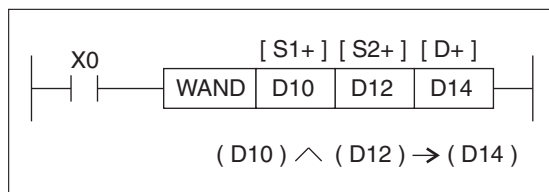


Abb. 6-48:
Programmierbeispiel zum Einsatz der WAND-Anweisung

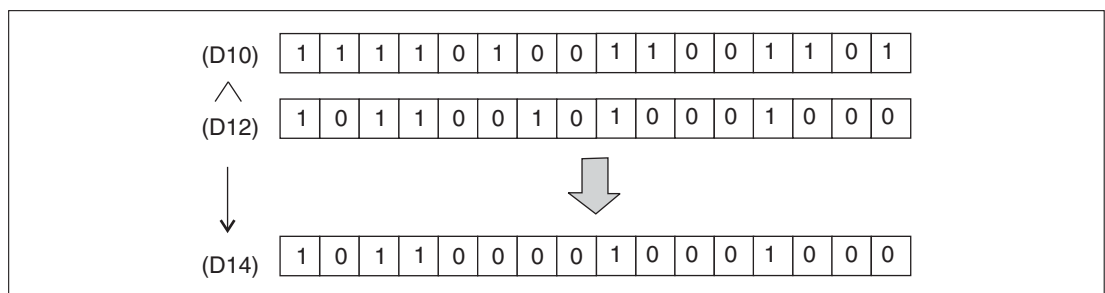
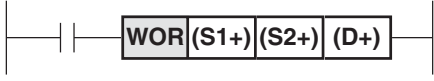
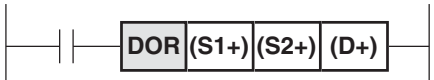


Abb. 6-49: UND-Verknüpfung von einzelnen Bits mit der WAND-Anweisung

6.4.8 Logische ODER-Verknüpfung binärer Daten (WOR, DOR)

		WOR		FNC 27							
		Logische ODER-Verknüpfung									
Operanden		S1+, S2+		D+		Puls-Anweisung (P)		Verarbeitung		Programmschritte	
		K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②		KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②		●		16 Bit 32 Bit		WOR/ WOPP	
		DOR		FNC 27							
		Logische ODER-Verknüpfung									
Operanden		S1+, S2+		D+		Puls-Anweisung (P)		Verarbeitung		Programmschritte	
		K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②		KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②		●		16 Bit 32 Bit		DOR/ DORP	

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
 ② Nur bei FX3U und FX3UC

Funktionsweise

Logische ODER-Verknüpfung binärer Daten

Beschreibung

- Es wird eine logische ODER-Verknüpfung von einzelnen Bits ausgeführt.
- Die Daten in (S1+) und (S2+) werden bitweise miteinander verknüpft. Das Verknüpfungsergebnis wird in (D+) gespeichert.

(S1+)	(S2+)	(D+)
1	1	1
1	0	1
0	1	1
0	0	0

Tab. 6-14:
Wahrheitstabelle der ODER-Verknüpfung

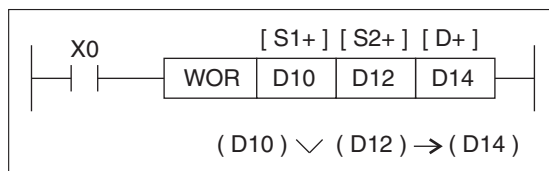


Abb. 6-50:
Programmierbeispiel zum Einsatz der WOR-Anweisung

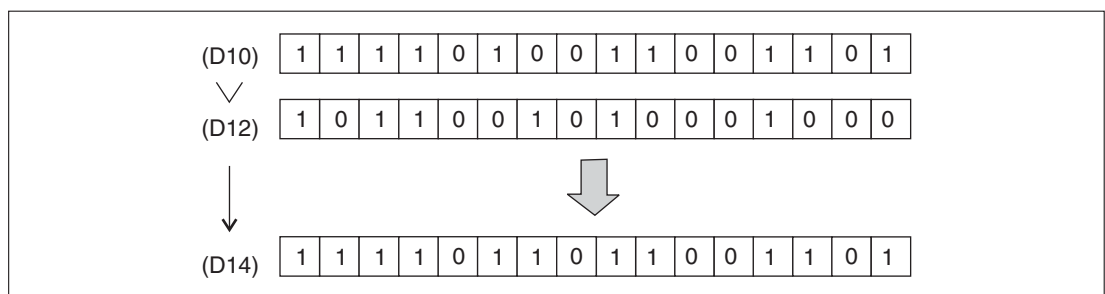
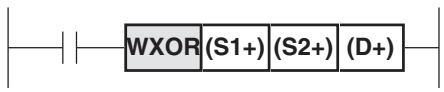
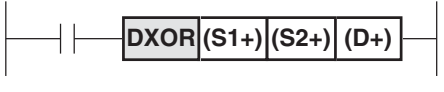


Abb. 6-51: ODER-Verknüpfung von einzelnen Bits mit einer WOR-Anweisung

6.4.9 Logische Exklusiv-ODER-Verknüpfung binärer Daten (WXOR, DXOR)

		WXOR		FNC 28					
		Logische Exklusiv-ODER-Verknüpfung							
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC			
	●	●	●	●	●	●			
Operanden	S1+, S2+	D+		Puls-Anweisung (P)		Verarbeitung		Programmschritte	
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②	KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②		●	16 Bit	32 Bit	WXOR/ WXORP	7	

		DXOR		FNC 29					
		Logische Exklusiv-ODER-Verknüpfung							
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC			
	●	●	●	●	●	●			
Operanden	S1+, S2+	D+		Puls-Anweisung (P)		Verarbeitung		Programmschritte	
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②	KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②		●	16 Bit	32 Bit	DXOR/ DXORP	13	

- ① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ② Nur bei FX3U und FX3UC

Funktionsweise

Logische Exklusiv-ODER-Verknüpfung binärer Daten

Beschreibung

- Es wird eine logische Exklusiv-ODER-Verknüpfung von einzelnen Bits ausgeführt.
- Die Daten in (S1+) und (S2+) werden bitweise miteinander verknüpft. Das Verknüpfungsergebnis wird in (D+) gespeichert.

(S1+)	(S2+)	(D+)
1	1	0
1	0	1
0	1	1
0	0	0

Tab. 6-15:
Wahrheitstabelle der Exklusiv-ODER-Verknüpfung

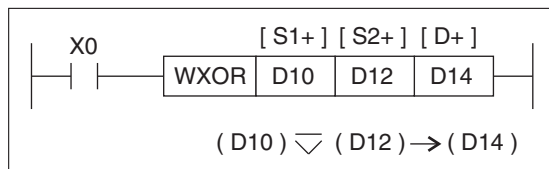


Abb. 6-52:
Programmierbeispiel zum Einsatz der WXOR-Anweisung

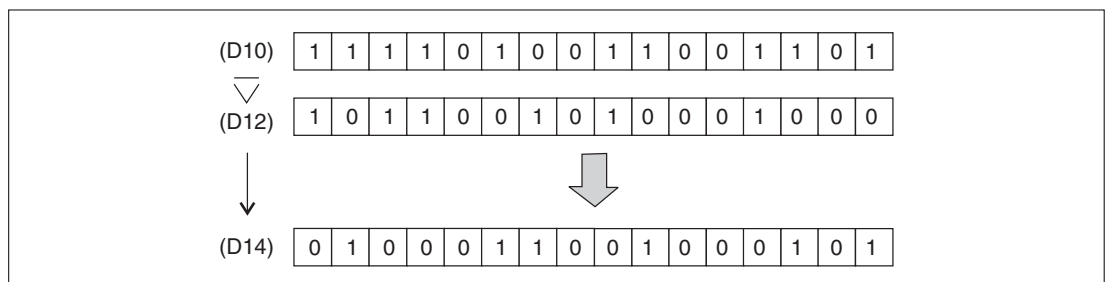


Abb. 6-53: Exklusiv-ODER-Verknüpfung einzelner Bits mit der WXOR-Anweisung

6.4.10 Negation von Daten (NEG)

		NEG		FNC 29			
		Negation von Daten					
Operanden	D	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	KnY, KnM, KnS T, C, D, R*, V, Z, U□\NG□*	●	16 Bit	32 Bit	NEG/NEGP	3	
			●	●	DNEG/DNEGP	5	

* Nur bei FX3U und FX3UC

Funktionsweise

Bilden des 2er-Komplements eines Datenwertes

Beschreibung

- Durch die Anweisung NEG wird das 2er-Komplement des in (D+) angegebenen Datenwertes gebildet und in (D+) gespeichert.

HINWEIS

Wenn keine Flankenerkennung programmiert wird, wird die Komplementbildung in jedem Zyklus wiederholt.

Beispiel ▽

NEG-Anweisung

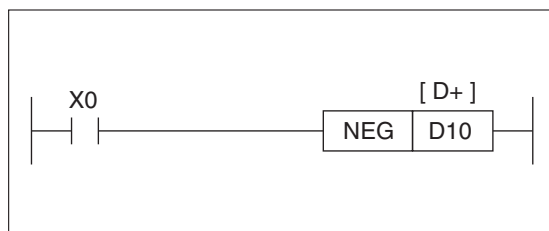


Abb. 6-54

Programmierbeispiel zur NEG-Anweisung

Funktion

Binär: $\overline{D10} + 1 \rightarrow D10$

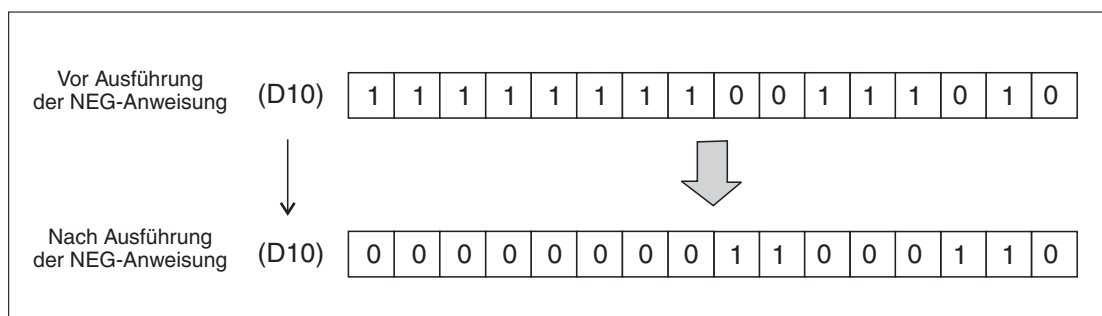


Abb. 6-55: Funktion der NEG-Anweisung



6.5 Verschiebeanweisungen

Übersicht der Anweisungen FNC 30 bis 39

Symbol	FNC	Bedeutung	Abschnitt
ROR	30	Rotation nach rechts	6.5.1
ROL	31	Rotation nach links	6.5.2
RCR	32	Rotieren von Bits nach rechts	6.5.3
RCL	33	Rotieren von Bits nach links	6.5.4
SFTR	34	Binäre Daten bitweise verschieben, rechts	6.5.5
SFTL	35	Binäre Daten bitweise verschieben, links	
WSFR	36	Daten wortweise nach rechts verschieben	6.5.6
WSFL	37	Daten wortweise nach links verschieben	6.5.7
SFWR	38	Schreiben in einen FIFO-Speicher	6.5.8
SFRD	39	Lesen aus einem FIFO-Speicher	6.5.9

Tab. 6-16: Übersicht der Verschiebeanweisungen

6.5.1 Rotation nach rechts (ROR)

		ROR		FNC 30					
		Rotation nach rechts							
Operanden	D+	n	CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
	KnY ^① , KnM ^① , KnS ^① , T, C, D, R ^② , V, Z, U□G□ ^③	K, H ^④	Puls-Anweisung (P)				●	●	●
			Verarbeitung		Programmschritte				
		●	16 Bit	32 Bit	ROR, RORP	5			
			●	●	DROR, DRORP	9			

- ① Kn = K4 (16-Bit-Operation), Kn = K8 (32-Bit-Operation)
- ② Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ③ Nur bei FX3U und FX3UC
- ④ n ≤ 16 (16-Bit-Operation), n ≤ 32 (32-Bit-Operation)

Funktionsweise

Rotieren von Bits um (n) Stellen nach rechts

Beschreibung

- Das Bit-Muster in (D+) wird mit jeder Ausführung von ROR um n Stellen nach rechts rotiert.
- Der Status des zuletzt rotierten Bits wird in M8022 (Carry Flag) kopiert.

HINWEIS | Wird keine Flankenerkennung programmiert, wird das Bit-Muster in jedem Zyklus rotiert.

Beispiel ▽ ROR-Anweisung

Die Bitdaten im Datenregister D0 werden jedesmal um 4 Bits (K4) nach rechts verschoben, wenn der Eingang X0 von AUS nach EIN schaltet. Der Wert des zuletzt rotierenden Bits wird im Carry Flag gespeichert.

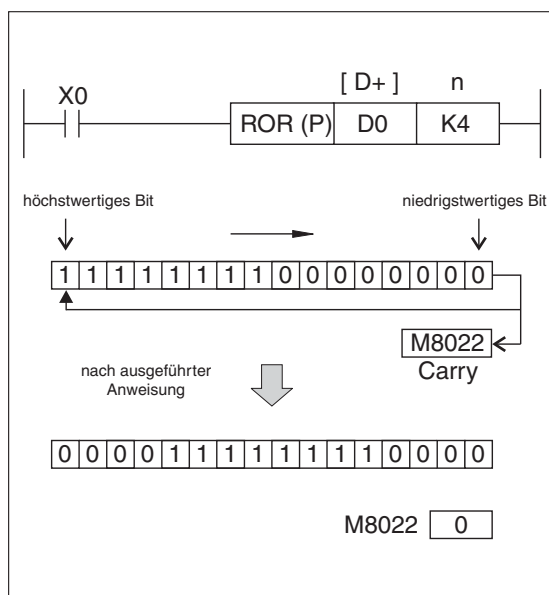


Abb. 6-56:
Programmierbeispiel für eine Rotation nach rechts



6.5.2 Rotation nach links (ROL)

		ROL		FNC 31					
		Rotation nach links							
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
					●	●	●	●	
Operanden	D+	n		Puls-Anweisung (P)		Verarbeitung		Programmschritte	
	KnY ^① , KnM ^① , KnS ^① , T, C, D, R ^② , V, Z, U□G□ ^③	K, H ^④		●		16 Bit	32 Bit	ROL, ROLP	5
						●	●	DROL, DROLP	9

- ① Kn = K4 (16-Bit-Operation), Kn = K8 (32-Bit-Operation)
- ② Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ③ Nur bei FX3U und FX3UC
- ④ n ≤ 16 (16-Bit-Operation), n ≤ 32 (32-Bit-Operation)

Funktionsweise

Rotieren von Bits um n Stellen nach links

Beschreibung

- Das Bit-Muster in (D+) wird mit jeder Ausführung von ROR um n Stellen nach links rotiert.
- Der Status des zuletzt rotierten Bits wird in M8022 (Carry Flag) kopiert.

HINWEIS

| Wird keine Flankenerkennung programmiert, wird das Bit-Muster in jedem Zyklus rotiert.

Beispiel ▽

ROL-Anweisung

Die Bitdaten im Datenregister D0 werden jedesmal um 4 Bits (K4) nach links verschoben, wenn der Eingang X0 von AUS nach EIN schaltet. Der Wert des zuletzt rotierenden Bits wird im Carry Flag gespeichert.

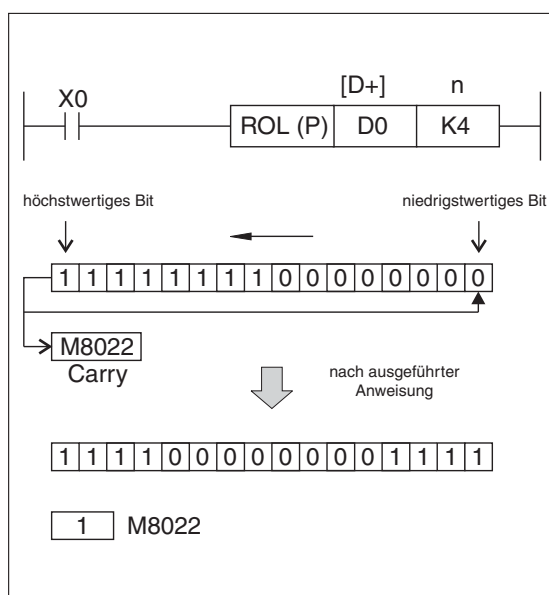


Abb. 6-57:
Programmierbeispiel für eine Rotation nach links



6.5.3 Rotieren von Bits nach rechts (RCR)

		RCR		FNC 32			
		Rotieren von Bits nach rechts					
Operanden	D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte	
				16 Bit	32 Bit	RCR, RCRP	5
	KnY ^① , KnM ^① , KnS ^① , T, C, D, R ^② , V, Z, U□G□ ^②	K, H ^③	●	●	●	DRCR, DCRP	9

- ① Kn = K4 (16-Bit-Operation), Kn = K8 (32-Bit-Operation)
- ② Nur bei FX3U und FX3UC
- ③ n ≤ 16 (16-Bit-Operation), n ≤ 32 (32-Bit-Operation)

Funktionsweise:

Rotieren von Bits um n Stellen nach rechts unter Einbeziehung des Carry-Flags M8022

Beschreibung:

- Das Bit-Muster in (D+) wird um n Stellen nach rechts rotiert.
- Das Carry-Flag M8022 wird in die Rotationschleife eingefügt.
- Wenn in (D+) ein aus Bits zusammengefasster Operand genutzt werden soll, sind für die Zusammenfassung nur die Konstanten K4 (16-Bit-Operation) und K8 (32-Bit-Operation) gültig.

HINWEIS | Wenn keine Flankenerkennung programmiert wird, wird die Rotation in jedem Programmzyklus ausgeführt.

| Der Status des Carry-Bits beim Einschalten der Anweisung wird mit in das zu rotierende Bit-Muster aufgenommen.

Beispiel ▾ RCR-Anweisung

Die Bit-Daten im Datenregister D0 werden jedesmal um 4 Bits (K4) nach rechts verschoben, wenn der Eingang X0 von AUS nach EIN schaltet.

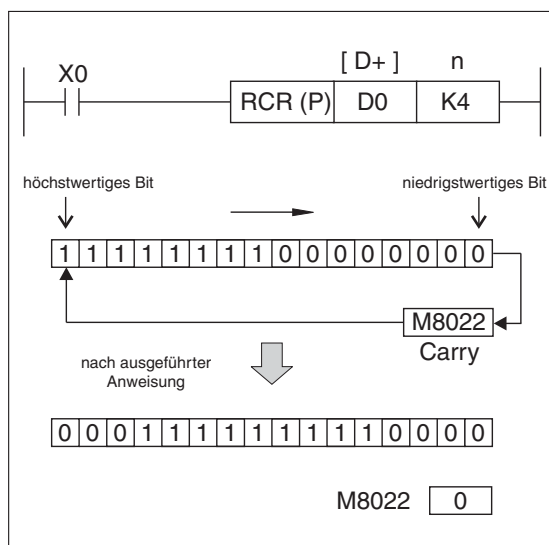


Abb. 6-58: Programmierbeispiel für eine Rotation nach rechts



6.5.4 Rotieren von Bits nach links (RCL)

		RCL		FNC 33				
		Rotieren von Bits nach links						
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
					●			●
Operanden	D+	n		Puls-Anweisung (P)	Verarbeitung		Programmschritte	
	KnY ^① , KnM ^① , KnS ^① , T, C, D, R ^② , V, Z, U□G□ ^②	K, H ^③		●	16 Bit	32 Bit	RCL, RCLP	5
				●	●	DRCL, DRCLP	9	

- ① Kn = K4 (16-Bit-Operation), Kn = K8 (32-Bit-Operation)
- ② Nur bei FX3U und FX3UC
- ③ n ≤ 16 (16-Bit-Operation), n ≤ 32 (32-Bit-Operation)

Funktionsweise

Rotieren von Bits um n Stellen nach links unter Einbeziehung des Carry-Flags M8022

Beschreibung

- Das Bit-Muster in (D+) wird um n Stellen nach links rotiert.
- Das Carry-Flag M8022 wird in die Rotationschleife eingefügt.
- Wenn in (D+) ein aus Bits zusammengefasster Operand genutzt werden soll, sind für die Zusammenfassung nur die Konstanten K4 (16-Bit-Operation) und K8 (32-Bit-Operation) erlaubt.

HINWEIS

- Wenn keine Flankenerkennung programmiert wird, wird die Rotation in jedem Programmzyklus ausgeführt.
- Der Status des Carry-Bits beim Einschalten der Anweisung wird mit in das zu rotierende Bit-Muster aufgenommen.

Beispiel ▾

RCL-Anweisung

Die Bit-Daten im Datenregister D0 werden jedesmal um 4 Bits (K4) nach links verschoben, wenn der Eingang X0 von AUS nach EIN schaltet.

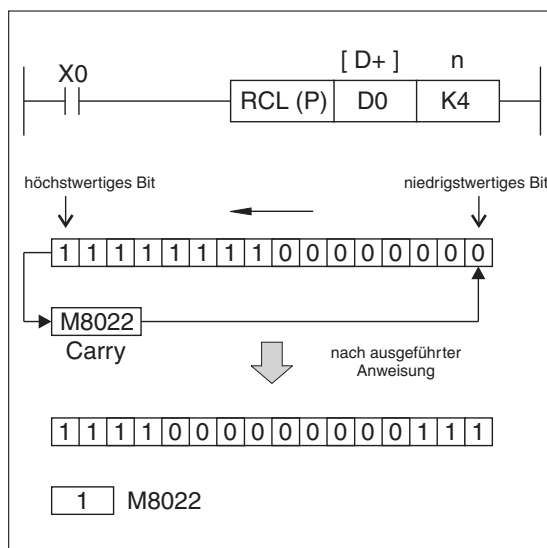


Abb. 6-59: Programmierbeispiel für eine Rotation nach links



6.5.5 Binäre Daten bitweise verschieben (SFTR, SFTL)

				SFTR		FNC 34				
				Binäre Daten bitweise verschieben, rechts						
				CPU	FX1s	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3s	FX3U FX3UC
					●	●	●	●	●	●
Operanden	S+	D+	n1, n2	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	X, Y, M, S, D□.b*	Y, M, S	K, H	●	16 Bit	32 Bit	SFTR/ SFTRP	9		
					●					

				SFTL		FNC 35				
				Binäre Daten bitweise verschieben, links						
				CPU	FX1s	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3s	FX3U FX3UC
					●	●	●	●	●	●
Operanden	S+	D+	n1, n2	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	X, Y, M, S, D□.b*	Y, M, S	K, H	●	16 Bit	32 Bit	SFTL/ SFTLP	9		
					●					

* Nur bei FX3U und FX3UC

Funktionsweise:

Binäre Daten werden bitweise nach rechts oder links verschoben.

Beschreibung:

- Diese Anweisung bezieht sich auf Bitoperanden, die zu einem Datenwort zusammengefasst sind. Die Wortbreite wird mit n1 festgelegt.
- Mit der Ausführung der Anweisung werden die Bits in (D+) um n2 Stellen verschoben, und ausgehend von (S+) werden n2 Bits in (D+) in Abhängigkeit der Verschieberichtung eingefügt.
- n1: Anzahl der Zieladressen, beginnend mit der Startadresse in (D+)
n2: Anzahl der zu verschiebenden Bits
Für n2 darf keine negative Zahl angegeben werden. Weiterhin gilt: (n2 ≤ n1 ≤ 1024) und (n1 ≤ maximal mögliche Adresse des in (D+) angegebenen Operanden)
- Mit der SFTR-Anweisung können Sie Daten bitweise nach rechts verschieben.
- Mit der SFTL-Anweisung können Sie Daten bitweise nach links verschieben.

HINWEIS

Die Anweisungen werden in jedem Programmzyklus ausgeführt. Dies können Sie durch Einsatz einer vorgeschalteten Impulsfunktion (PLS- oder PLF-Anweisung) oder die Verwendung des Befehlsparameters P verhindern.

Beispiel ▾ Einsatz der SFTR-Anweisung

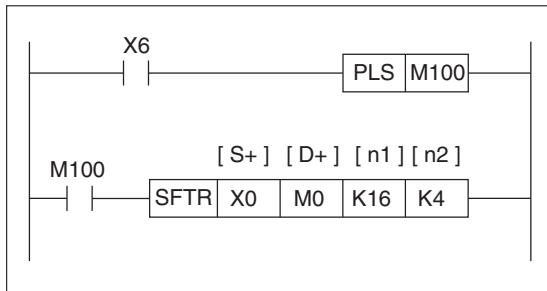


Abb. 6-60:
Programmierbeispiel zum Einsatz der SFTR-Anweisung

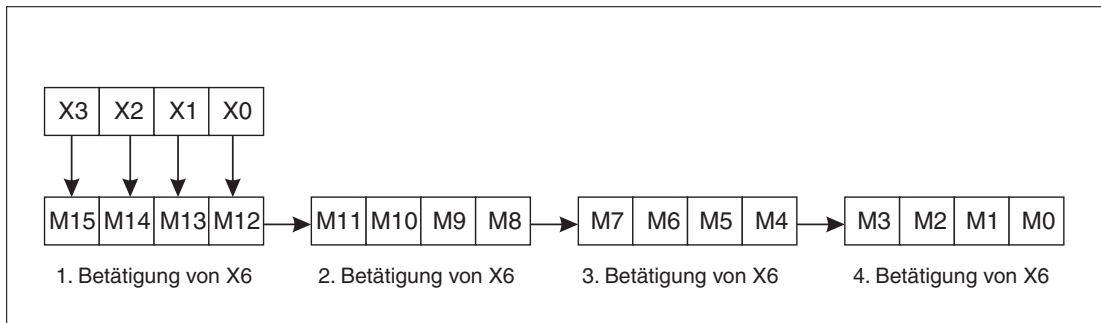


Abb. 6-61: *Beispiel zum bitweisen Verschieben nach rechts*

Mit der Ausführung von X6 werden die an den Eingängen X0 bis X3 anliegenden binären Signale bitweise in den definierten Merkerbereich eingelesen und entsprechend nach rechts verschoben. △

Beispiel ▾ Einsatz der SFTL-Anweisung

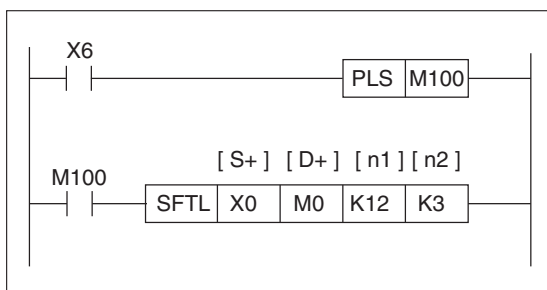


Abb. 6-62:
Programmierbeispiel zum Einsatz der SFTL-Anweisung

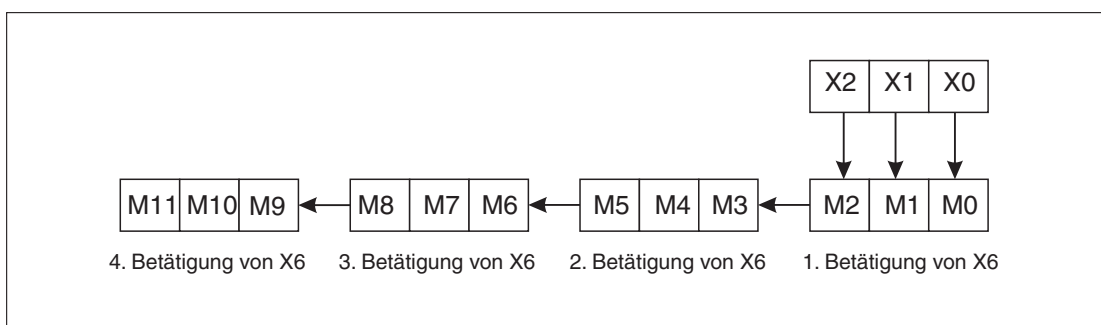


Abb. 6-63: *Beispiel zum bitweisen Verschieben nach links*

Wird X6 betätigt, werden die binären Signale an den Eingängen X0 bis X2 bitweise in den definierten Merkerbereich eingelesen und nach links verschoben. △

6.5.6 Daten wortweise nach rechts verschieben (WSFR)

				WSFR		FNC 36				
				Daten wortweise nach rechts verschieben						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
							●	●	●	●
Operanden	S+	D+	n1, n2	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	KnX, KnY, KnM, KnS, T, C, D, R ^① , U□G□ ^②	KnY, KnM, KnS, T, C, D, R ^① , U□G□ ^②	K, H n2 ≤ n1 ≤ 512	●	16 Bit	32 Bit	WSFR/ WSFRP		9	

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktionsweise

Daten werden wortweise nach rechts verschoben.

Beschreibung

- Die Quelldaten (S+) werden in einen Stapelspeicher (D+) geschrieben und verschoben. Die Tiefe des Stapels beträgt n1 Worte.
- Mit jeder Ausführung der Anweisung werden n2 Worte eingelesen und der Inhalt des Stapels verschoben.
- n2 darf kein negativer Wert sein.

HINWEIS

Bei der Verwendung von zusammengefassten Bit-Operanden müssen (S+) und (D+) die selbe Anzahl Bits haben.

Beispiel ▾

WSFR-Anweisung

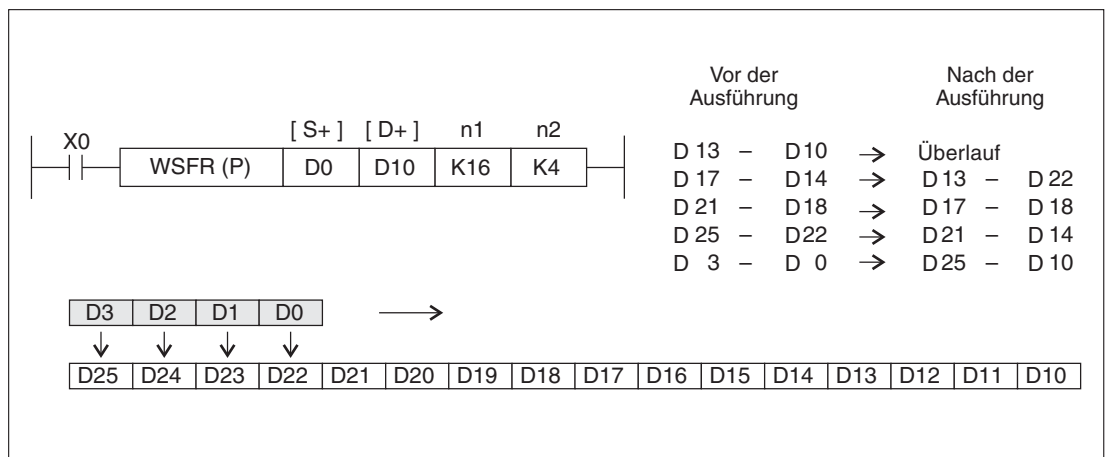


Abb. 6-64: Programmierbeispiel zum Verschieben nach rechts



6.5.7 Daten wortweise nach links verschieben (WSFL)

				WSFL		FNC 37				
				Daten wortweise nach links verschieben						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
							●	●	●	●
Operanden	S+	D+	n1, n2	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	KnX, KnY, KnM, KnS, T, C, D, R ^① , U□G□ ^②	KnY, KnM, KnS, T, C, D, R ^① , U□G□ ^②	K, H n2 ≤ n1 ≤ 512	●	16 Bit	32 Bit	WSFL/ WSFLP		9	
					●					

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktionsweise

Daten werden wortweise nach links verschoben.

Beschreibung

- Die Quelldaten (S+) werden in einen Stapelspeicher (D+) geschrieben und verschoben. Die Tiefe des Stapels beträgt n1 Worte.
- Mit jeder Ausführung der Anweisung werden n2 Worte eingelesen und der Inhalt des Stapels verschoben.
- n2 darf kein negativer Wert sein.

HINWEIS

Bei der Verwendung von zusammengefassten Bit-Operanden müssen (S+) und (D+) die selbe Anzahl Bits haben.

Beispiel ▾

WSFL-Anweisung

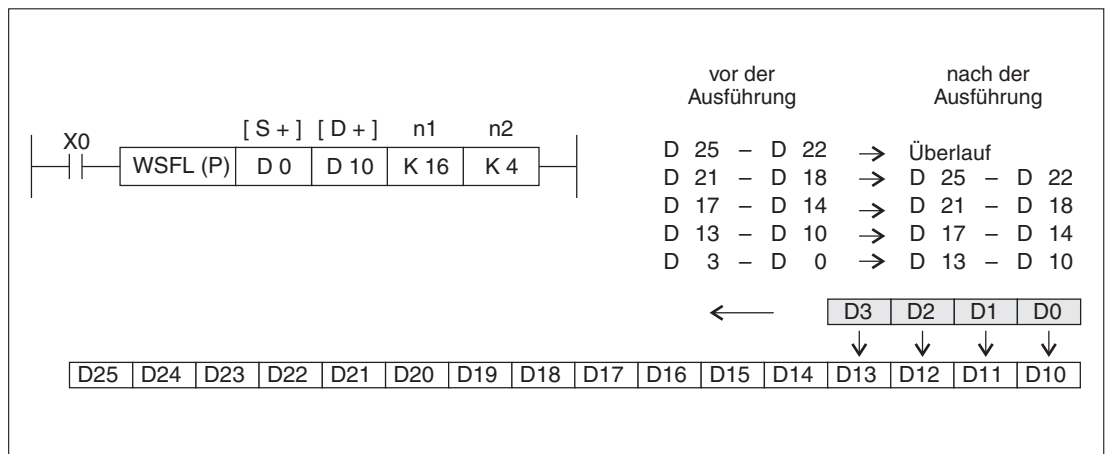


Abb. 6-65: Programmierbeispiel zum Verschieben nach links



6.5.8 Schreiben in einen FIFO-Speicher (SFWR)

				SFWR	FNC 38								
				Schreiben in einen FIFO-Speicher									
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC			
				●	●	●	●	●	●				
Operanden	S+	D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte						
	KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□\G□ ^②	KnY, KnM, KnS, T, C, D, R ^① , U□\G□ ^②	K, H 2 ≤ n ≤ 512		●	16 Bit	32 Bit	SFWR/ SFWRP		7			

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktionsweise

Einlesen von Wörtern in einen durch die Anweisung definierten FIFO-Speicher (FIFO = First-In / First-Out d.h. was zuerst eingetragen wird, wird auch zuerst wieder ausgelesen).

Beschreibung

- Die Daten aus (S+) werden in einen Stapelspeicher geschrieben.
- Die erste Adresse des Stapels ist (D+).
- Die Tiefe des Stapels beträgt (n) Worte.
- Im Stapel können maximal (n-1) Worte abgelegt werden, da (D+) als Zeiger für den Stapel verwendet wird. (D+) sollte vor der ersten Ausführung auf Null gesetzt werden.
- Wenn (n-1) Worte in den Stapel eingetragen wurden, ohne andere Worte auszulesen, ist es nicht möglich, weitere Worte einzutragen. Dieser Zustand wird durch das Einschalten des Carry-Bits (M8022) angezeigt.
- Mit jeder Ausführung der Anweisung wird der Zeiger (D+) inkrementiert.
- Die Anweisung wird zusammen mit der Anweisung SFRD eingesetzt; der Parameter (n) sollte in beiden Anweisungen gleich sein.

Beispiel ▾

SFWR-Anweisung

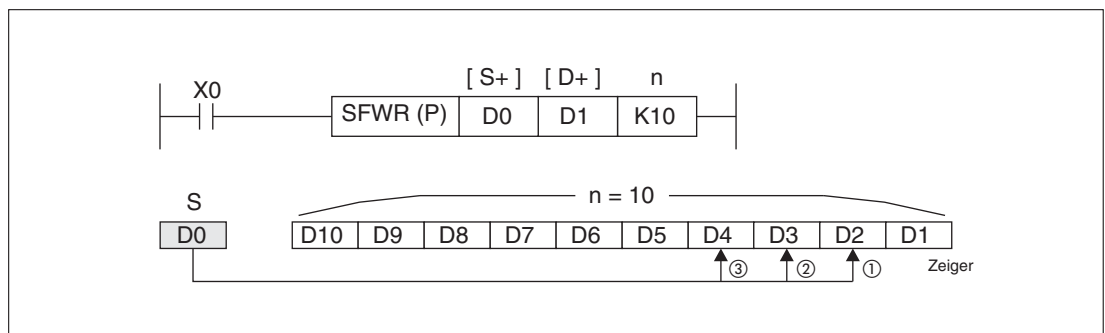


Abb. 6-66: Programmierbeispiel zum Schreiben in einen FIFO-Speicher

Ein Anwendungsbeispiel finden Sie auf der Seite 6-70.



6.5.9 Lesen aus einem FIFO-Speicher (SFRD)

				SFRD		FNC 39				
				Lesen aus einem FIFO-Speicher						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
					●	●	●	●	●	●
Operanden	S+	D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②	KnY, KnM, KnS, T, C, D, R ^① , U□G□ ^②	K, H 2 ≤ n ≤ 512		●	16 Bit	32 Bit	SFRD/ SFRDP	7	

- ① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ② Nur bei FX3U und FX3UC

Funktionsweise

Auslesen von Wörtern aus einem FIFO-Speicher (FIFO = First-In / First-Out d.h. was zuerst eingetragen wurde, wird auch zuerst wieder ausgelesen).

Beschreibung

- Aus dem mit (S+) beginnenden Stapelspeicher wird der Inhalt von ((S+)+1) nach (D+) ausgelesen.
- Der Zeiger des Stapelspeichers (S+) wird mit jeder Ausführung von SFRD dekrementiert.
- Die Werte in ((S+)+2) bis ((S+)+n) werden um eine Position aufwärts geschoben.
- Wenn (S+) den Wert Null annimmt, ist der Stapelspeicher leer. Dies wird durch das Bit M8020 angezeigt.
- Die Anweisung SFRD arbeitet zusammen mit der Anweisung SFWR. Der Parameter n sollte in beiden Anweisungen gleich sein.

Beispiel ▾

SFRD-Anweisung

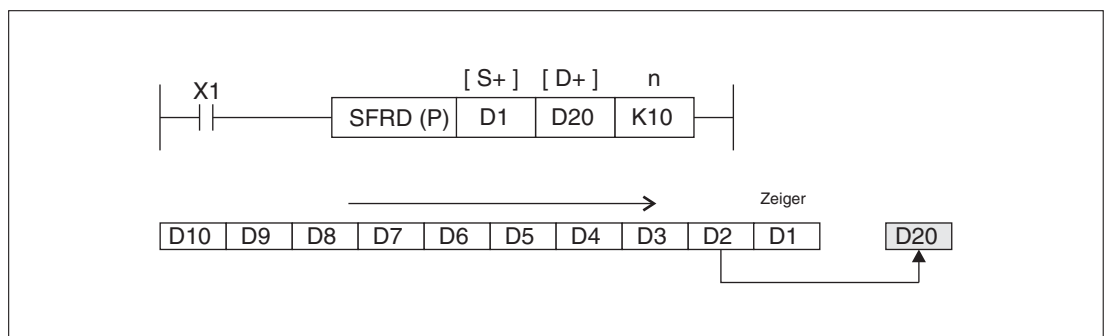


Abb. 6-67: Programmierbeispiel zum Lesen aus einem FIFO-Speicher

Ein Anwendungsbeispiel finden Sie auf der Seite 6-70.



Beispiel ▾ Programmierung eines FIFO-Speichers

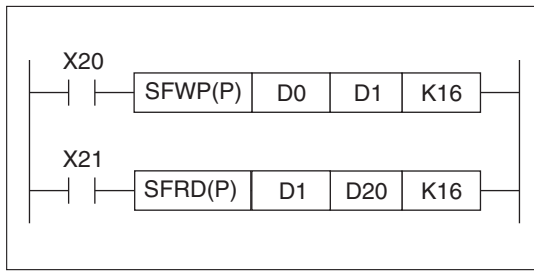


Abb. 6-68:
 Programmierbeispiel zum Schreiben und Lesen eines FIFO-Speichers

Es wird ein Stapelspeicher mit 15 Adressen und einem Pointer definiert.

Betätigen von:	/	X20	X20	X20	X21	X20	X21
D20	0	0	0	0	55	55	66
D0	0	55	66	77	77	88	88
D1	0	1	2	3	2	3	2
D2	0	55	55	55	66	66	77
D3	0	0	66	66	77	77	88
D4	0	0	0	77	0	88	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
D15	0	0	0	0	0	0	0
D16	0	0	0	0	0	0	0

Abb. 6-69: Register des FIFO-Speichers

Mit jeder Betätigung von X20 wird der Zahlenwert von D0 an die erste noch unbelegte Adresse innerhalb des Stapelspeichers geschrieben.

Mit jeder Betätigung von X21 wird der Inhalt von D2 nach D20 ausgelesen und der Inhalt der anderen Adressen innerhalb des Stapelspeichers um eine Position vorgeschoben. △

6.6 Datenoperationen

Übersicht der Anweisungen FNC 40 bis 49

Symbol	FNC	Bedeutung	Abschnitt
ZRST	40	Operandenbereiche zurücksetzen	6.6.1
DECO	41	Daten decodieren	6.6.2
ENCO	42	Daten codieren	6.6.3
SUM	43	Ermittlung gesetzter Bits	6.6.4
BON	44	Überprüfung eines Bits	6.6.5
MEAN	45	Ermittlung eines Durchschnittswertes	6.6.6
ANS	46	Starten eines Zeitintervalls	6.6.7
ANR	47	Rücksetzen von Anzeigebits	6.6.8
SQR	48	Ermittlung der Quadratwurzel	6.6.9
FLT	49	Umwandlung des Zahlenformats	6.6.10

Tab. 6-17: Übersicht der Anweisungen FNC 40 bis 49

6.6.1 Operandenbereiche zurücksetzen (ZRST)

		ZRST		FNC 40			
		Operandenbereiche zurücksetzen					
Operanden	D1+, D2+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	Y, M, S, T, C, D, R ^① , U□G□ ^②	●	16 Bit	32 Bit	ZRST/ ZRSTP	5	

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktionsweise

Mehrere aufeinanderfolgende Operanden (Operandenbereiche) lassen sich mit nur einer ZRST-Anweisung auf den Signalzustand oder Istwert „0“ zurücksetzen.

Beschreibung

- In (D1+) und (D2+) wird der Operandenbereich festgelegt, den Sie zurücksetzen möchten.
- In (D1+) und (D2+) müssen Sie den gleichen Operandentyp angeben.

(D1+): Erste Operandenadresse

(D2+): Letzte Operandenadresse

Es gilt: (D1+) ≤ (D2+)

Wenn (D1+) > (D2+), wird nur der in (D1+) angegebene Operand zurückgesetzt.

HINWEISE

Obwohl es sich um eine 16-Bit-Operation handelt, können in den beiden Zieladressen auch 32-Bit-Counter eingesetzt werden. Der kombinierte Einsatz von 16- und 32-Bit-Countern ist jedoch nicht zulässig. So ist es z. B. nicht erlaubt, in (D1+) einen 16-Bit-Counter und in (D2+) einen 32-Bit-Counter anzugeben.

Einzelne Operanden können mit der RST-Anweisung zurückgesetzt werden (siehe Abschnitt 4.13).

Beispiel ▾

Einsatz der ZRST-Anweisung

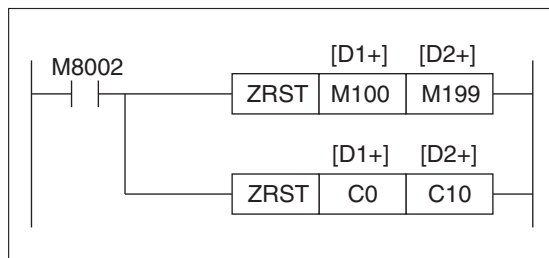


Abb. 6-70:

Programmierbeispiel zum Einsatz der ZRST-Anweisung

Die Bit-Operanden M100 bis M199 werden auf den Signalzustand „0“ zurückgesetzt. Die Wortoperanden C0 bis C10 werden auf den Istwert „0“ zurückgesetzt. Die zugehörigen Spulen und Kontakte werden ausgeschaltet. △

6.6.2 Daten decodieren (DECO)

				DECO		FNC 41				
				Daten decodieren						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
					●	●	●	●	●	●
Operanden	S+	D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	K, H, X, Y, M, S, T, C, D, R ^① , V, Z, U□G□ ^②	Y, M, S, T, C, D, R ^① , U□G□ ^②	K, H		●	16 Bit	32 Bit	DECO/ DECOP	7	
					●					

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktionsweise

Daten decodieren

Beschreibung

Die Daten von n Operanden, ausgehend von der in (S+) angegebenen Startadresse, werden decodiert. In (D+) wird die Startadresse der Zieloperanden festgelegt, in denen das Decodierergebnis abgelegt wird.

n: Anzahl der Operanden, deren Daten decodiert werden sollen.
 Bei der Angabe eines Bit-Operanden in D+ gilt: (1 ≤ n ≤ 8).
 Bei der Angabe eines Wortoperanden in D+ gilt: (1 ≤ n ≤ 4).

(S+): Startadresse der Operanden, deren Daten decodiert werden sollen.

2ⁿ: Anzahl der Zieloperanden

(D+): Startadresse der Zieloperanden

HINWEISE

Die Anweisung wird nicht ausgeführt, wenn n = 0 ist.

Die Anweisung wird nur ausgeführt, wenn die Eingangsbedingung gesetzt ist. Der zugehörige Ausgang bleibt aktiviert, auch wenn die Eingangsbedingung anschließend wieder ausgeschaltet wird.

Fehlerquellen

- Ein Programmablauffehler tritt auf, wenn n nicht im Bereich von 0 bis 8 liegt.
- Ein Programmablauffehler tritt auf, wenn sämtliche Bits der Ausgangsoperanden den Wert 0 aufweisen.

Beispiel ▾ Einsatz der DECO-Anweisung mit Angabe eines Bit-Operanden in D+ ($1 \leq n \leq 8$)

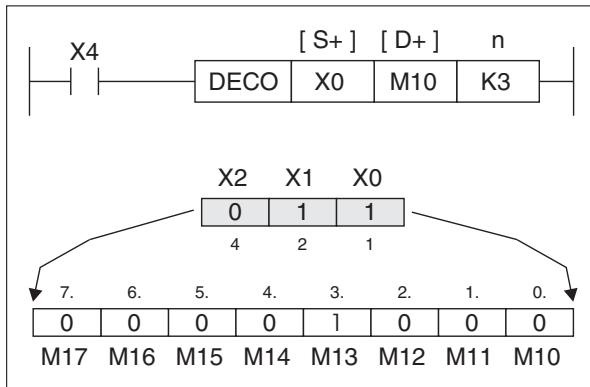


Abb. 6-71: Programmierbeispiel zum Einsatz der DECO-Anweisung mit Angabe eines Bit-Operanden in D+

Bei $n = 3$ lauten die Eingangsoperanden X_0 , X_1 und X_2 . Weil $2^n = 2^3 = 8$, stehen als Zieladressen die Merker M_{10} bis M_{17} zur Verfügung.

Der Wert der Eingangsoperanden ist $1+2=3$. Entsprechend wird das 3. Bit der Zieladressen, d.h. der Merker M_{13} , eingeschaltet. Lautet der Wert der Eingangsoperanden 0, wird der Merker M_{10} eingeschaltet. △

Beispiel ▾ Einsatz der DECO-Anweisung mit Angabe eines Wortoperanden in D+ ($1 \leq n \leq 4$)

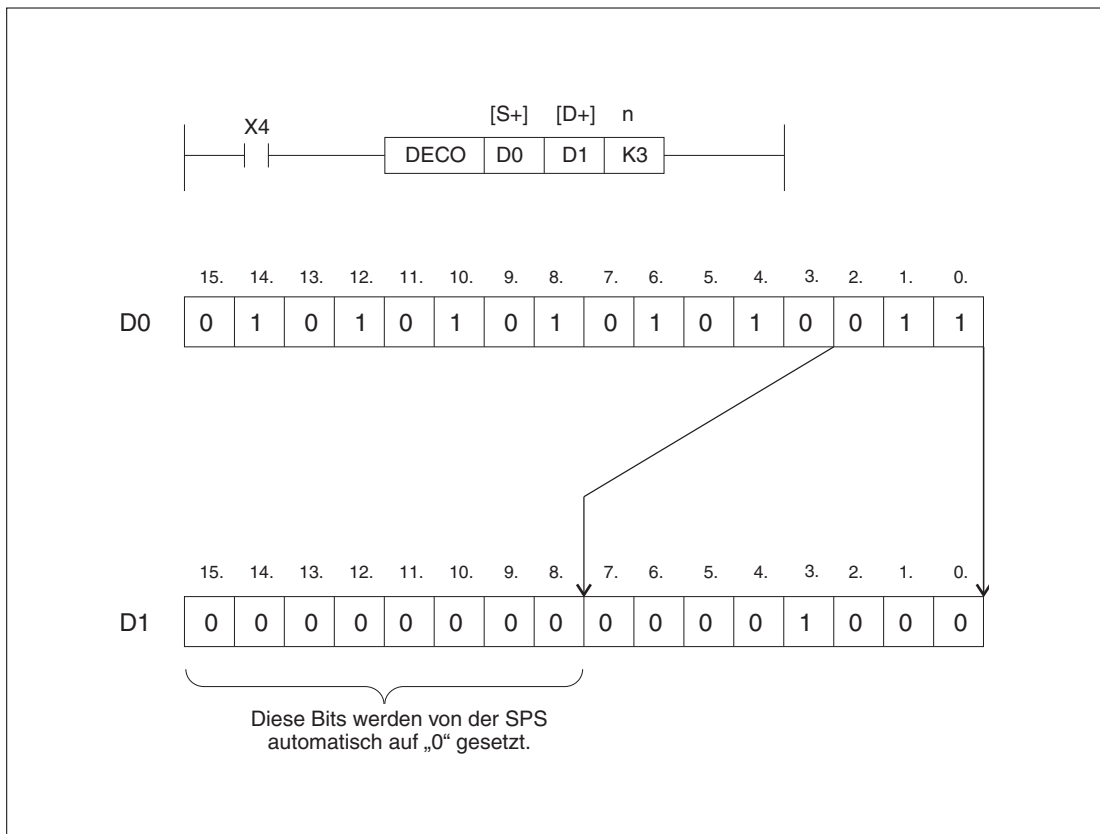


Abb. 6-72: Programmierbeispiel zum Einsatz der DECO-Anweisung mit Angabe eines Wortoperanden in D+

Die niedrigsten 3 Bits aus dem Datenregister D_0 werden decodiert. Das Decodierergebnis $1+2=3$ wird in das Datenregister D_1 übertragen. Im Datenregister D_1 wird das 3. Bit gesetzt.

Ist der Wert für $n \leq 3$, werden alle höherwertigen nicht benötigten Bit in den Zieladressen auf 0 gesetzt.

6.6.3 Daten codieren (ENCO)

				ENCO		FNC 42				
				Daten codieren						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
					●	●	●	●	●	●
Operanden	S+	D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	X, Y, M, S, T, C, D, R ^① , V, Z, U□G□ ^②	T, C, D, V, Z, R ^① , V, Z, U□G□ ^②	K, H		●	16 Bit	32 Bit	ENCO/ ENCOP	7	

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktionsweise

Ermittlung, welche Bits gesetzt sind

Beschreibung

Die Daten von 2^n Operanden, ausgehend von der in (S+) angegebenen Startadresse, werden codiert. In (D+) wird der Zieloperand festgelegt, in dem das Codierergebnis abgelegt wird.

2^n : Anzahl der Operanden, deren Daten codiert werden sollen.

n: Anzahl der Zieloperanden

Bei der Angabe eines Bit-Operanden in (S+) gilt: $(1 \leq n \leq 8)$

Bei der Angabe eines Wortoperanden in (S+) gilt: $(1 \leq n \leq 4)$

(S+): Startadresse der Operanden, deren Daten codiert werden sollen.

(D+): Zieloperand

HINWEISE

Haben mehrere der in (S+) angegebenen Operanden den Wert 1, wird nur das höchste Bit bearbeitet.

Die Anweisung wird nicht ausgeführt, wenn $n = 0$ ist.

Die Anweisung wird nur ausgeführt, wenn die Eingangsbedingung gesetzt ist. Der zugehörige Ausgang bleibt aktiviert, auch wenn die Eingangsbedingung anschließend wieder ausgeschaltet wird.

Fehlerquellen

- Ein Programmablauffehler tritt auf, wenn n nicht im Bereich von 0 bis 8 liegt.
- Ein Programmablauffehler tritt auf, wenn sämtliche Bits der Ausgangsoperanden den Wert 0 aufweisen.

Beispiel ▾

Einsatz der ENCO-Anweisung mit Angabe eines Bit-Operanden in (S+) ($1 \leq n \leq 8$)

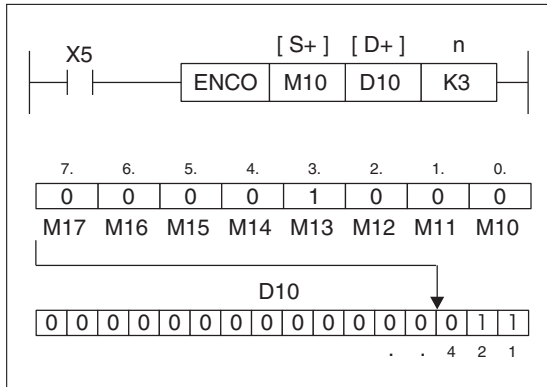


Abb. 6-73: Programmierbeispiel zum Einsatz der ENCO-Anweisung mit Angabe eines Bit-Operanden in (S+)

Wenn $2^n = 2^3 = 8$, stehen als Ausgangsadressen die Merker M10 bis M17 zur Verfügung. Weil bei den Ausgangsoperanden der 3. Operand, d.h. der Merker M13 gesetzt ist, wird in das Datenregister D10 der Wert 3 geschrieben. △

Beispiel ▾

Einsatz der ENCO-Anweisung mit Angabe eines Wortoperanden in (S+) ($1 \leq n \leq 4$)

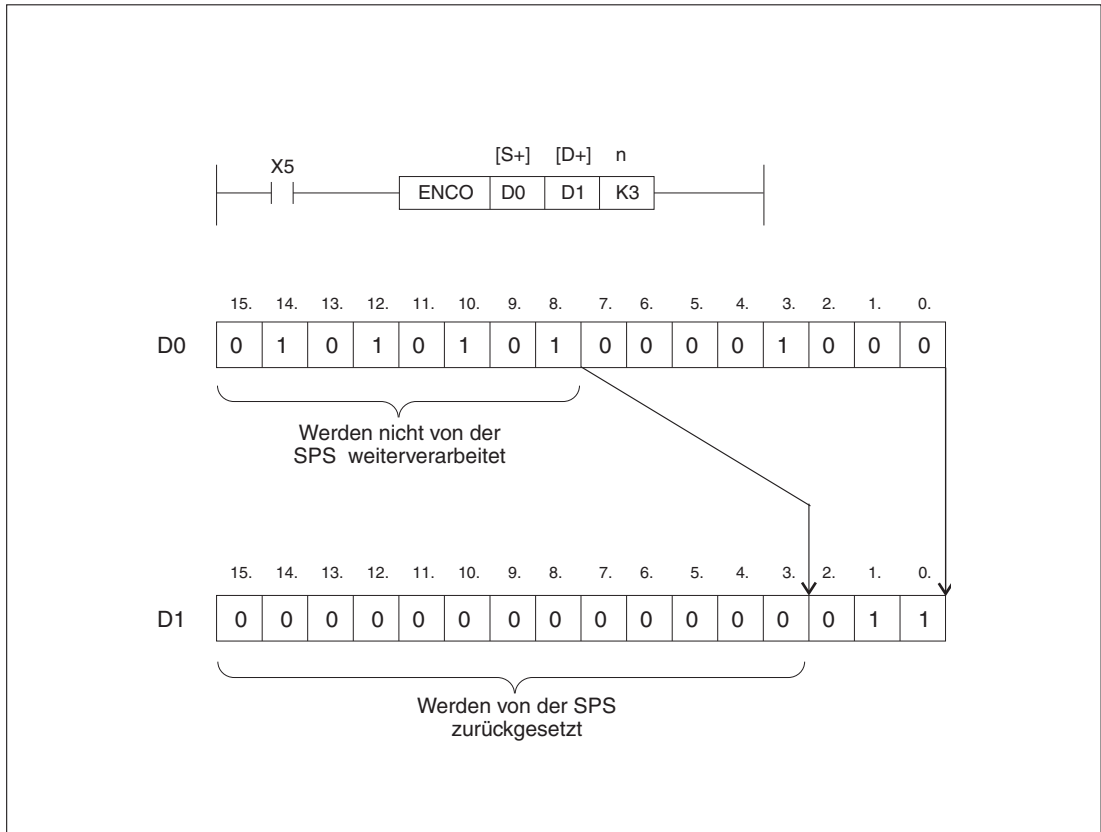


Abb. 6-74: Programmierbeispiel zum Einsatz der ENCO-Anweisung mit Angabe eines Wortoperanden in (S+)

Im Datenregister D0 ist das 3. Bit gesetzt. Es wird also der Zahlenwert 3 codiert und im Datenregister D1 gespeichert. △

6.6.4 Ermittlung gesetzter Bits (SUM)

		SUM		FNC 43			
		Ermittlung gesetzter Bits					
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte	
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②	KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②	●	16 Bit	32 Bit	SUM, SUMP	5/7 ^③
				●	●	DSUM, DSUMP	9

- ① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ② Nur bei FX3U und FX3UC
- ③ 5 Schritte bei FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC, 7 Schritte bei FX2N und FX2NC

Funktionsweise

Ermitteln der Anzahl gesetzter Bits in einem Datenwort.

Beschreibung

- Es wird die Anzahl der eingeschalteten Bits in (S+) ermittelt.
- Der ermittelte Wert wird nach (D+) geschrieben.

HINWEIS

Wenn eine 32-Bit-Operation ausgeführt wird, werden die oberen 16 Bit ((D+)+1) des Zieloperanden (D+) auf Null gesetzt, da die maximale Anzahl eingeschalteter Bits in (S+) 32 beträgt.

Beispiel ▾

SUM-Anweisung

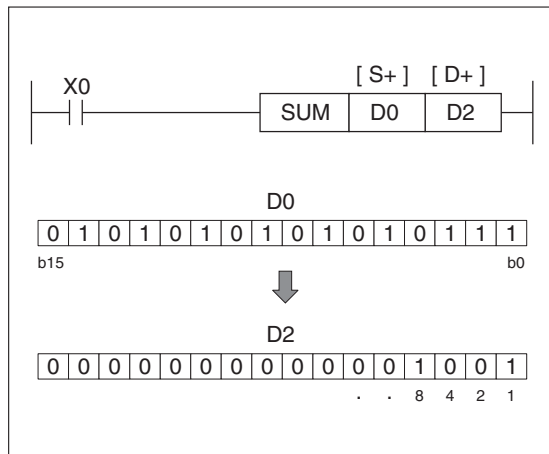


Abb. 6-75:
 Programmierbeispiel zur Ermittlung gesetzter Bits



6.6.5 Überprüfen eines Bits (BON)

				BON		FNC 44				
				Überprüfen eines Bits						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
							●	●	●	●
Operanden	S+		D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	K, H, KnX, KnY, KnM, KnS, S, T, C, D, R ^① , V, Z, U□G□ ^②		Y, M, S, D□.b ^②	K, H ^③		●	16 Bit	32 Bit	BON/BONP	7
						●	●	●	DBON/DBONP	13

- ① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ② Nur bei FX3U und FX3UC
- ③ n = 0 bis 15 (16-Bit-Operation); n = 0 bis 31 (32-Bit-Operation)

Funktionsweise

Überprüfen eines einzelnen Bits innerhalb eines Datenwortes.

Beschreibung

- Wenn das Bit Nummer n innerhalb von (S+) den Zustand „1“ hat, wird (D+) eingeschaltet.

Beispiel ▾

BON-Anweisung

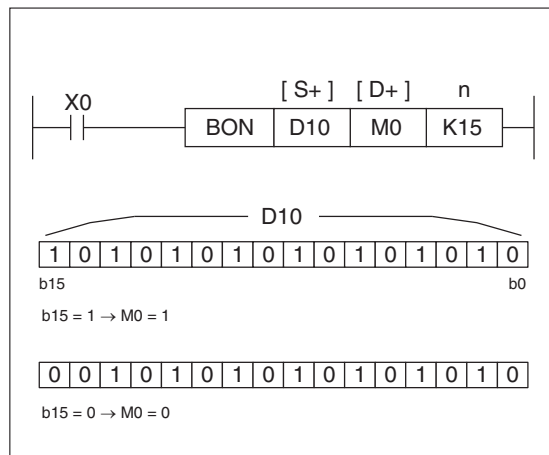


Abb. 6-76:

Programmierbeispiel zum Überprüfen eines Bits



6.6.6 Ermittlung von Durchschnittswerten (MEAN)

				MEAN		FNC 45				
				Ermittlung von Durchschnittswerten						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
							●	●	●	●
Operanden	S+	D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	KnX, KnY, KnM, KnS, T, C, D, R ^① , U□G□ ^②	KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②	K, H, D ^① , R ^① (n = 1 bis 64)	●	16 Bit	32 Bit	MEAN/ MEANP	7		
					●	●	DMEAN/ DMEANP	13		

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktionsweise

Bilden des arithmetischen Mittelwerts aus mehreren Datenwörtern.

Beschreibung

Ausgehend von (S+) werden n Datenwörter addiert und durch n dividiert. Das ganzzahlige Ergebnis wird nach (D+) geschrieben.

HINWEISE

Wird n größer als der verfügbare Operandenbereich, ausgehend von (S+), gewählt, so wird n automatisch auf die verfügbare Anzahl von Operanden angepasst.

Wird in einer 32-Bit-Anweisung für n ein Operand des Typs D oder R angegeben, werden die beiden Operanden n und (n+1) belegt. Beispielsweise werden bei der Programmierung von „DMEAN D0 D100 R0“ für n die Operanden R0 und R1 verwendet.

Fehlerquelle

Wenn (n) außerhalb des Bereichs von 1 bis 64 liegt, tritt ein Fehler auf.

Beispiel ▽

MEAN-Anweisung

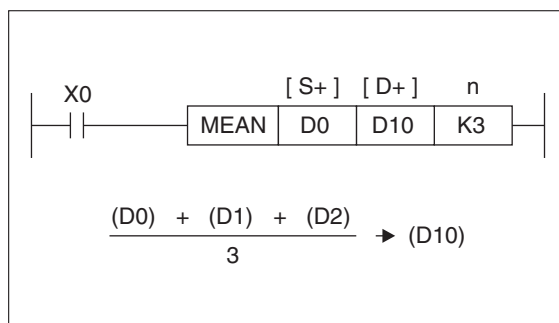


Abb. 6-77:

Programmierbeispiel zur Ermittlung von Durchschnittswerten



6.6.7 Starten eines Zeitintervalls (ANS)

				ANS		FNC 46				
				Starten eines Zeitintervalls						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
							●	●		●
Operanden	S+	D+	m	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	T T0 bis T199	S S900 bis S999	K, H, D*, R* (m = 1 – 32767)		16 Bit	32 Bit	ANS		7	
					●					

* Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

Funktionsweise

Starten eines Zeitintervalls und Schalten eines Anzeige-Bits

Beschreibung

- Die Startoperanden S900 bis S999 können als Anzeige-Bits verwendet werden.
- Mit der Ausführung dieser Anweisung wird eine Zeit von (m x 100 ms) gestartet.
- Nach Ablauf der Zeit wird das Anzeige-Bit (D+) eingeschaltet.
- In (S+) wird ein Timer vorgegeben, der das Zeitintervall bildet.

HINWEIS | Der verwendete Timer darf im weiteren Programm nicht mehr eingesetzt werden.

Beispiel ▾ Programmierung der ANS-Anweisung

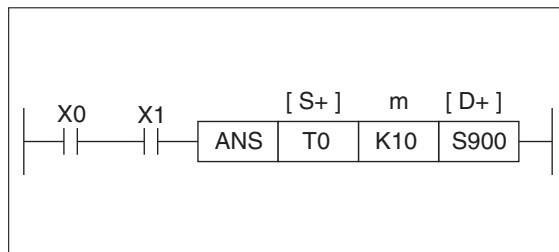


Abb. 6-78:
 Programmierbeispiel zum Starten eines Zeitintervalls von 1 Sekunde

△

6.6.8 Rücksetzen von Anzeige-Bits (ANR)

		ANR		FNC 47				
		Rücksetzen eines Anzeige-Bits						
Operanden	—	CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
	—			●	●		●	
Operanden	—	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	—	●	16 Bit	32 Bit	ANR/ ANRP		1	

Funktionsweise

Rücksetzen von Anzeige-Bits

Beschreibung

Wenn die Anweisung aktiv ist, wird das aktive Anzeige-Bit mit der niedrigsten Adresse zurück-gesetzt.

HINWEIS | Die Anweisung sollte mit der Option „P“ ausgeführt werden.

Beispiel ▾ Programmierung der ANR-Anweisung

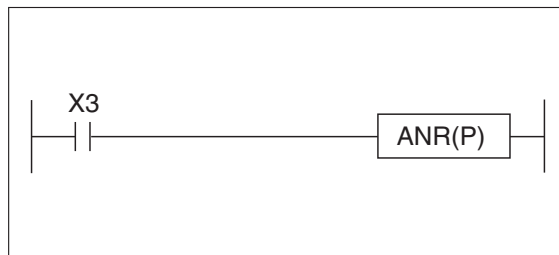


Abb. 6-79:
Programmierbeispiel zum Rücksetzen von Anzeige-Bits

Wird X3 eingeschaltet, wird das gesetzte Anzeigebit zwischen S900 und S999 zurückgesetzt.

Wurden mehrere Anzeigebits gesetzt, wird das Anzeigebit mit der niedrigeren Adresse zurückgesetzt.

Die anderen gesetzten Anzeigebits werden durch nochmaliges Einschalten von X3 in aufstei-gender Reihenfolge zurückgesetzt. △

6.6.9 Ermittlung der Quadratwurzel (SQR)

			SQR		FNC 48		
			Berechnung der Quadratwurzel				
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
			●			●	
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte	
	K, H, D, R*, U□\G□*	D, R*, U□\G□*	●	16 Bit	32 Bit	SQR/SQRP 5 DSQR/DSQRP 9	

* Nur bei FX3U und FX3UC

Funktionsweise

Berechnung der Quadratwurzel, $(D+) = \sqrt{S+}$

Beschreibung

Ausgehend von (S+) wird die Quadratwurzel berechnet und auf einen ganzzahligen Wert abgerundet nach (D+) geschrieben.

Beispiel ▾

Programmierung der SQR-Anweisung

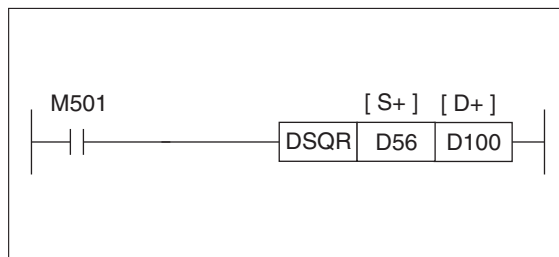


Abb. 6-80:

Programmierbeispiel zur Berechnung der Quadratwurzel

Wird der Merker M501 eingeschaltet, wird vom Wert im Datenregister D56 die Quadratwurzel berechnet, und das Ergebnis als abgerundeter ganzzahliger Wert nach Datenregister D100 geschrieben.

HINWEISE

Die Wurzel aus einer negativen Zahl führt immer zu einem Fehler und der Fehlermerker M8067 wird gesetzt.

Tritt bei der Berechnung der Quadratwurzel eine Nachkommastelle auf, wird der Sondermerker M8021 (Borrow Flag) gesetzt.

Ist das Ergebnis der Quadratwurzelberechnung 0, wird der Sondermerker M8020 (Zero Flag) gesetzt.

In der folgenden Tabelle sind einige Beispielergebnisse für die Wurzelberechnung SQR aufgeführt.

(S+)	Ergebnis	(D+)
25	5,0	5
60	7,746	7
-236	15,36 i	Fehler
147	12,124	12

Tab. 6-18:

Beispielergebnisse zur Wurzelberechnung



6.6.10 Umwandlung des Zahlenformats (FLT)

			FLT		FNC 49			
			Umwandlung des Zahlenformats					
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	D, R ^① , U□\G□ ^②	D, R ^① , U□\G□ ^②		●	16 Bit	32 Bit	FLT/FLTP	5
				●	●	DFLT/DFLTP	9	

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktionsweise

Umwandlung einer Zahl vom Integer-Format in das Gleitkomma-Format

Beschreibung

Ausgehend von (S+) wird die Zahl in eine Gleitkomma-Zahl umgewandelt und nach ((D+)+1) und (D+) geschrieben.

HINWEIS | Das Ergebnis der Zahlenumwandlung wird immer in einem 32-Bit-Datenregister abgelegt.

Beispiel ▾ Programmierung der FLT-Anweisung

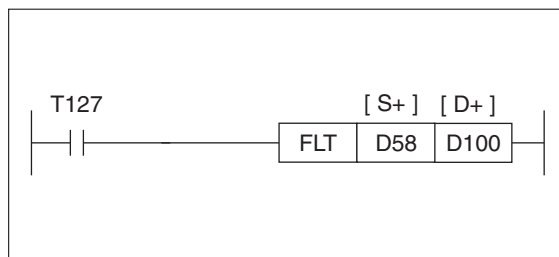


Abb. 6-81:
 Programmierbeispiel zur
 Umwandlung in das Gleitkomma-Format

Sobald der Timer T127 einschaltet, wird der Wert im Datenregister D58 in eine Gleitkomma-Zahl umgewandelt, und der Wert in die Datenregister D101 und D100 geschrieben.

△

6.7 High-Speed-Anweisungen

Übersicht der Anweisungen FNC 50 – 59

Symbol	FNC	Bedeutung	Abschnitt
REF	50	Ein- und Ausgänge auffrischen	6.7.1
REFF	51	Einstellen der Eingangsfiler	6.7.2
MTR	52	Einlesen einer Matrix	6.7.3
DHSCS	53	Setzen durch High-Speed-Counter	6.7.4
DHSCR	54	Rücksetzen durch High-Speed-Counter	
DHSZ	55	Bereichsvergleich	6.7.5
SPD	56	Geschwindigkeitserkennung	6.7.6
PLSY	57	Ausgabe einer definierten Anzahl von Impulsen	6.7.7
PWM	58	Impulsausgabe mit Modulation der Impulsweite	6.7.8
PLSR	59	Ausgabe einer bestimmten Anzahl von Impulsen	6.7.9

Tab. 6-19: Übersicht der High-Speed-Anweisungen

6.7.1 Ein- und Ausgänge auffrischen (REF)

			REF		FNC 50				
			Ein- und Ausgänge auffrischen						
			CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
				●	●	●	●	●	●
Operanden	D	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	X, Y ①	K, H ②	●	16 Bit	32 Bit	REF/ REFP	5		

① Die Operanden sollten ein Vielfaches von 10 sein: X0, X10, X20, etc.

② n muss ein Vielfaches von 8 sein: 8, 16, 24, etc.

Funktionsweise

Ein- und Ausgänge auffrischen (aktualisieren)

Beschreibung

- Die Programmabarbeitung bei den Steuerungen der FX-Familie erfolgt nach dem Prozessabbildverfahren. Vor der Programmabarbeitung liest die Steuerung die Eingangssignalzustände und speichert sie im Prozessabbild der Eingänge. Es werden also nicht die Eingänge, sondern das Prozessabbild der Eingänge verarbeitet.

Nach der Programmabarbeitung werden die Daten des Prozessabbildes der Ausgänge an die Ausgänge übertragen.

- Mit der REF-Anweisung können die Eingänge während eines Programmzyklus abgefragt und das Prozessabbild der Eingänge aufgefrischt (aktualisiert) werden.
- Die REF-Anweisung können Sie verwenden, um aktuelle Eingangsinformationen zu lesen, während eine Operation ausgeführt wird.
- Mit der REF-Anweisung kann das Operationsergebnis unmittelbar nach der Abarbeitung der Operation ausgegeben werden.
- Die REF-Anweisung kann z. B. in einer FOR-NEXT-Anweisung oder zwischen einer CJ-Anweisung (höhere Schrittnummer) und der zugehörigen Pointermarkierung (niedrigere Schrittnummer) eingesetzt werden.

HINWEIS | Der Zustand der Ein- und Ausgänge wird vor jedem Programmzyklus aktualisiert.

Beispiel ▾ Einsatz der REF-Anweisung, Eingänge auffrischen

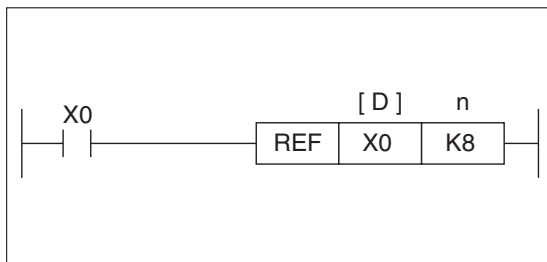


Abb. 6-82:
Programmierbeispiel zum Einsatz der REF-Anweisung; Eingänge auffrischen

Acht Adressen, d.h. die Eingänge X0 bis X7, werden aufgefrischt.

Sind die Eingänge ungefähr 10 ms (Verzögerungszeit) vor der Abarbeitung der REF-Anweisung aktiviert, wird das Prozessbild der Eingänge aktualisiert, wenn die REF-Anweisung ausgeführt wird. △

HINWEIS

Die Signalverzögerungszeit der Eingänge X0 bis X7 (X0 bis X17 bei der FX2N-, FX3U- und FX3UC-Serie, außer FX3U-16M□ und FX3UC-16M□) kann mit einer REFF-Anweisung (siehe Abschnitt 6.7.2) oder direkt im Sonderregister D8020 eingestellt werden .

Beispiel ▾ Einsatz der REF-Anweisung, Ausgänge aktualisieren

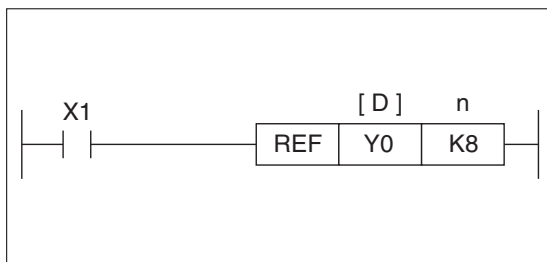


Abb. 6-83:
Programmierbeispiel zum Einsatz der REF-Anweisung; Ausgänge auffrischen

8 Ausgänge, d.h. die Ausgänge Y0 bis Y7, werden aufgefrischt.

Die Ausgangszustände werden nach Ablauf der Reaktionszeit an den Ausgängen ausgegeben. Die Reaktionszeit ist die physikalisch bedingte Schaltzeit des aktivierten Ausganges. △

6.7.2 Einstellen der Eingangfilter (REFF)

		REFF		FNC 51				
		Ein- und Ausgänge auffrischen						
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
					●			●
Operanden	n	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	K, H, D*, R* (n = 0 bis 60)	●		16 Bit	32 Bit	REF/ REFP	5	

* Nur bei FX3U und FX3UC

Funktionsweise

Einstellen der Filterzeiten für die Eingänge X0 bis X7 (X0 bis X17 bei der FX2N-, FX3U- und FX3UC-Serie, außer FX3U-16M□ und FX3UC-16M□) im Sonderregister D8020.

Beschreibung

- Durch die Ausführung der Anweisung REFF werden die Zustände der Eingänge X0 bis X7 (X0 bis X17 bei der FX2N-, FX3U- und FX3UC-Serie, außer FX3U-16M□ und FX3UC-16M□) in das Prozessabbild der Eingänge übernommen und die Eingangfilter auf (n) ms eingestellt.
- (n) kann einen Wert zwischen 0 und 60 [ms] annehmen. Die Einstellung von 0 bewirkt die folgenden Filterzeit:
 - FX2N- und FX2NC-Serie: 50 µs
 - FX3U/FX3UC-Serie
 - X0 bis X5: 5 µs
 - X6 und X7: 50 µs
 - X10 bis X17: 200 µs (Bei einer FX3U-16M□ und FX3UC-16M□ sind diese Eingänge fest auf 10 ms eingestellt.)

HINWEISE

Die Anweisung muss in jedem Zyklus eingeschaltet werden, da die Filtereinstellung ansonsten wieder auf den Standardwert von 10 ms eingestellt wird.

Bei der FX3U- und FX3UC-Serie wird unabhängig von der Einstellung durch die REFF-Anweisung bei den folgenden Funktionen für die Eingänge X0 bis X5 eine Filterzeit von 5 µs und bei den Eingängen X6 und X7 eine Filterzeit von 50 µs aktiviert:

- Verwendung eines Eingangs zur Auslösung eines Interrupts
- Verwendung eines Eingangs im Zusammenhang mit High-Speed-Countern
- Ausführung einer SPD-Anweisung (siehe Abschnitt 6.7.6)

Bei anderen Funktionen ist die im Sonderregister D8020 eingestellte Zeit gültig. Eingangssignale, die kürzer sind als die Filterzeit, werden nicht erfasst.

Beispiel ▾

Ist X10 eingeschaltet, wird der Eingangsstatus der Eingänge X0 bis X7 bei einer Eingangsverzögerung von 1 ms aufgefrischt. Normalerweise beträgt die Eingangsverzögerung 10 ms.

Mit der Anweisung „REFF K20“ wird beim Einschalten der Steuerung die Eingangsverzögerung auf 20 ms eingestellt.

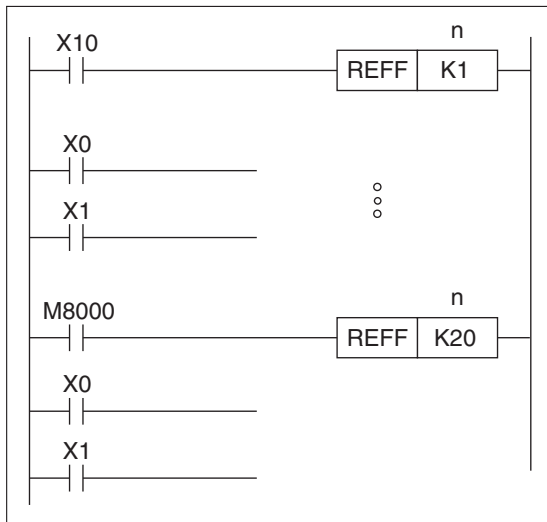


Abb. 6-84:
 Programmierbeispiel zur REFF-Anweisung



6.7.3 Einlesen einer Matrix (MTR)

				MTR		FNC 52			
				Einlesen einer Matrix					
Operanden	S+, D1+	D2+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	H ①	Y, M, S	K, H n = 2 bis 8		16 Bit	32 Bit	MTR	9	
					●	●	●	●	

① Der Operand sollte ein Vielfaches von 10 sein: X0, X10, X20 usw.

Funktionsweise

Einlesen einer 8 x n-Matrix in die SPS

Beschreibung

- Die Schalter einer 8 x n-Matrix werden in einem Multiplex-Verfahren über 8 Eingänge und n Ausgänge eingelesen.
- Den Eingängen (S+) bis ((S+)+8) werden für jede der n Reihen Merker zugeordnet. Die Merker für die erste Reihe beginnen mit (D2+).
- Jeder Reihe ist ein Ausgang zugeordnet; die erste Reihe wird durch den Ausgang (D1+) angesprochen.

HINWEISE

- | Diese Anweisung sollte nur bei einer Steuerung mit Transistorausgängen verwendet werden.
- | Die Anweisung kann nur einmal innerhalb eines Programms genutzt werden.

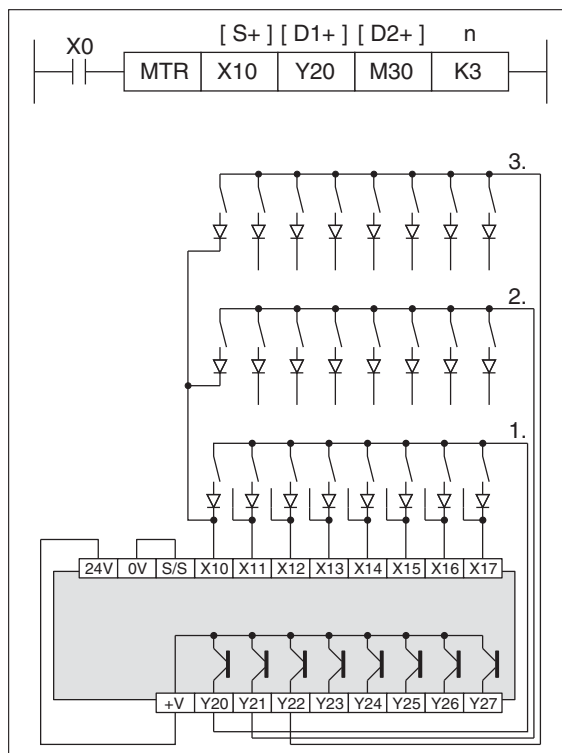
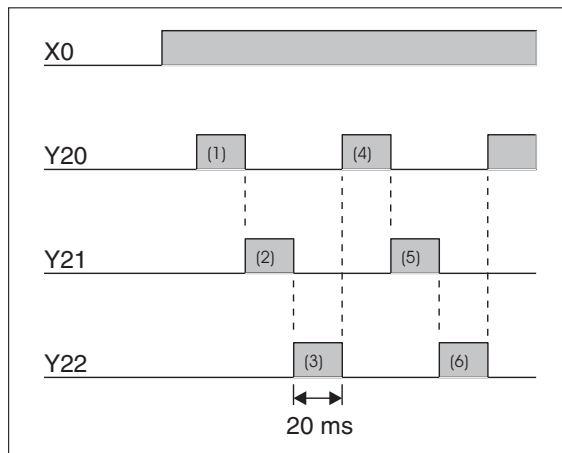


Abb. 6-85:
 Programmier- und Anwendungsbeispiel für eine MTR-Anweisung



Beispiel ▾

Im folgenden Beispiel ist zu sehen, dass die drei Ausgänge Y20, Y21 und Y22 nacheinander eingeschaltet werden. Dieser Vorgang wiederholt sich ständig. Die in der ersten, zweiten und dritten Zeile erfassten Daten werden kontinuierlich nach M30 bis M37, M40 bis M47 und M50 bis M57 übertragen und gespeichert.

**Abb. 6-86:**

Programmierbeispiel zum Schalten der Ausgänge

Die Ein-/Ausgangsbearbeitung für jeden Ausgang wird im Interrupt-Modus in Intervallen von 20 ms ausgeführt, wobei eine Verzögerungszeit der Eingangsfiler von 10 ms berücksichtigt ist.

Mit Hilfe der MTR-Anweisung können 64 Eingangszustände durch den Einsatz von 8 Eingängen und 8 Transistorausgängen erfasst werden. Alle Eingangsdaten werden innerhalb von 160 ms (20 x 8) eingelesen. Die Eingangsdaten können innerhalb von 80 ms aufgenommen werden, wenn die Eingänge X0 bis X7 eingesetzt werden, da für das Einlesen der Daten pro Zeile nur 10 ms benötigt werden.

M30 bis M37 bleiben unverändert, solange die Eingangsbedingung nicht gesetzt ist. M8029 wird gesetzt, sobald die Matrix gefüllt ist. M8029 wird zurückgesetzt, wenn die Eingangsbedingung ausgeschaltet wird.

△

HINWEISE

Zur Vermeidung von Programmkonflikten sollten als Eingänge möglichst nicht die Adressen X0 bis X7 verwendet werden (X0 bis X17 bei der FX3U- und FX3UC-Serie, außer FX3U-16M□ und FX3UC-16M□).

Werden diese Eingänge doch verwendet, ist pro Ausgang entsprechend der folgenden Abbildung ein Pull-Down-Widerstand zwischenschalten.

Beispiel ▾

Zwischenschaltung von Pull-Down-Widerständen bei Verwendung der Eingänge X0 bis X7.

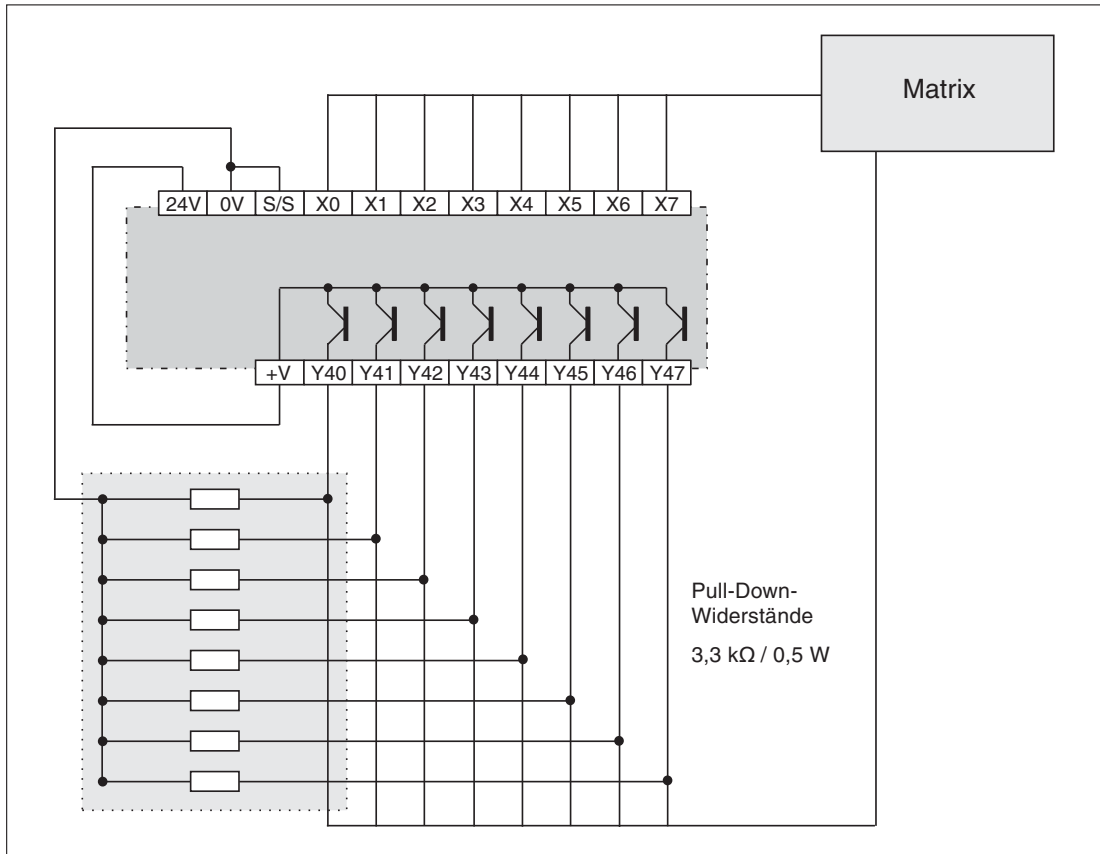
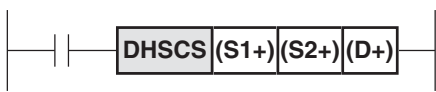



Abb. 6-87: Beispiel zum Einsatz von Pull-Down-Widerständen



6.7.4 Setzen und Rücksetzen durch High-Speed-Counter (DHSCS, DHSCR)

				DHSCS		FNC 53					
				Setzen durch High-Speed-Counter							
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
						●	●	●	●	●	●
Operanden	S1+	S2+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte				
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , Z, U□\G□ ^②	C235 bis C255	Y, M, S, D□.b ^② , P ^③			16 Bit	32 Bit	DHSCS		13	
						●					

				DHSCR		FNC 54					
				Rücksetzen durch High-Speed-Counter							
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
						●	●	●	●	●	●
Operanden	S1+	S2+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte				
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , Z, U□\G□ ^②	C235 bis C255	Y, M, S, D□.b ^② , C235 bis C255			16 Bit	32 Bit	DHSCR		13	
						●					

- ① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ② Nur bei FX3U und FX3UC
- ③ Counter-Interrupt-Pointer I010 bis I060 (siehe unten).

Funktionsweise

Setzen und Rücksetzen von Operanden durch High-Speed-Counter; Die Operanden werden ohne eine Programmzykluszeitverzögerung gesetzt bzw. zurückgesetzt.

Beschreibung

- Ein High-Speed-Counter zählt die Statusänderungen an den Zähl­ein­gängen im Inter­rupt-Modus. Jedem High-Speed-Counter sind feste Zähl­ein­gänge mit festen Funktionen zugeordnet.
- Im Abschnitt 9.1.10 finden Sie eine ausführliche Leistungsbeschreibung aller vorhandenen High-Speed-Counter und deren Einsatzmöglichkeiten in einem SPS-Programm.
- Mit der DHSCS-Anweisung können Sie Operanden durch High-Speed-Counter setzen. Der in D+ angegebene Operand wird gesetzt, sobald der eingestellte Sollwert des Counters erreicht ist.
- Mit der DHSCR-Anweisung können Sie Operanden durch High-Speed-Counter zurücksetzen. Der in D+ angegebene Operand wird zurückgesetzt, sobald der eingestellte Sollwert des High-Speed-Counters erreicht ist.
- Die Operanden werden ohne eine Programmzykluszeitverzögerung direkt nach Ausführung der Anweisung gesetzt.
- Die Anweisung wird ausgeführt, wenn die Daten in S1+ mit den Daten in S2+ übereinstimmen. Dabei muss die Aktivierung entweder durch einen Impuls an einem Zähl­ein­gang oder an einem Reset-Eingang erfolgen. Wenn die Aktivierung über einen Reset-Eingang erfolgen soll, muss der Sondermerker M8025 eingeschaltet sein.
- Die Anweisung wird nicht ausgeführt, wenn die Datenübereinstimmung zwischen S1+ und S2+ durch eine indirekte Änderung der Daten in S1+ erreicht wurde. Steht z.B. in S1+ das Datenregister D0 und wird der Datenwert in D0 durch eine MOV-Anweisung verändert, wird die High-Speed-Anweisung nicht ausgeführt.

HINWEIS

In einem SPS-Programm dürfen nicht mehr als 6 DHSCS- und DHSCR-Anweisungen eingesetzt werden.

Beispiel ▾

Einsatz der DHSCS-, DHSCR-Anweisungen

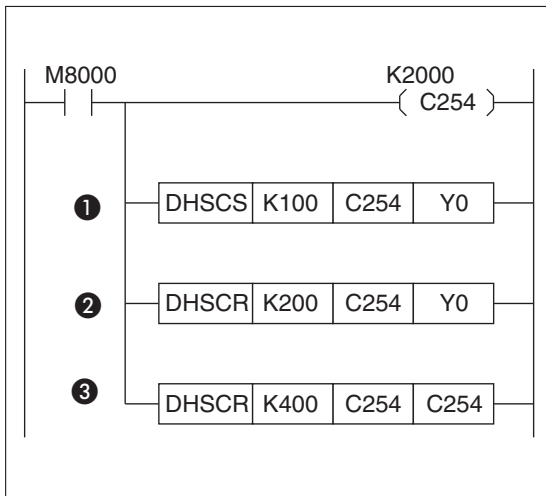


Abb. 6-88:
 Programmierbeispiel zum Einsatz der DHSCS-, DHSCR-Anweisung

Der Zählereingang des High-Speed-Counters C254 ist X0 (A-Phase) und X1 (B-Phase). Der Reset-Eingang ist X2, und der Start-Eingang ist X3.

- ① Wenn sich der Istwert des Counters C254 verändert von 99 auf 100 oder von 101 auf 100, wird der Ausgang Y0 sofort gesetzt.
- ② Wenn sich der Istwert des Counters C254 verändert von 199 auf 200 oder von 201 auf 200, wird der Ausgang Y0 sofort zurückgesetzt.
- ③ Wenn sich der Istwert des Counters C254 verändert von 399 auf 400 oder von 401 auf 400, wird der Counter C254 sofort zurückgesetzt.

△

HINWEIS

Der Ausgang wird durch seine physikalische Schaltzeit verzögert. Interne Operanden werden nach der Ausführung der Anweisung im jeweiligen Image-Register eingetragen.

Einsatz von Counter-Interrupt-Pointern

- Counter-Interrupt: I 0 ① 0
- ① Adresse 1 bis 6
 Counter-Interrupts können als Operanden zum Setzen (HSCS, FNC 53) oder Zurücksetzen (HSCR, FNC 54) durch High-Speed-Counter verwendet werden. Zum Ausschalten des Counter-Interrupts schalten Sie den Sondermerker M8059 ein.

Beispiel ▾

Interrupt-Pointer: I030

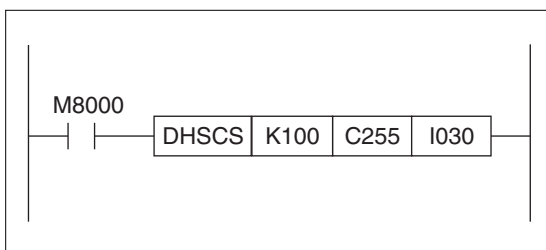


Abb. 6-89:
 Programmierbeispiel zum Einsatz eines Counter-Interrupts

Das über den Interrupt-Pointer I030 aufgerufene Interrupt-Programm wird ausgeführt, sobald der Wert des High-Speed-Counters C255 den in K100 angegebenen Wert erreicht.

6.7.5 Bereichsvergleich (DHSZ)

				DHSZ		FNC 55			
				Bereichsvergleich					
Operanden	S1+, S2+	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
					16 Bit	32 Bit	DHSZ		
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , Z, U□\G□ ^②	C235 bis C255	Y, M, S, D□.b ^②			●	●	●	
						●		17	

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktionsweise ohne Sondermerker

Bereichsvergleich für High-Speed-Counter im durch (S1+ / S2+) vorgegebenen Bereich

Beschreibung

- Die Anweisung DHSZ vergleicht im Interrupt-Betrieb den Istwert eines High-Speed-Counters mit dem in (S1+ / S2+) vorgegebenen Bereich.
- Der Vergleich findet mit jedem Zählimpuls am angegebenen Zähler (S+) statt.
- Das Ergebnis des Vergleichs wird über die Operanden (D+), ((D+)+1), ((D+)+2) dargestellt.
 - (D+) = S+ < S1+ / S2+; Zähler-Istwert unterhalb des Bereichs
 - ((D+)+1) = S+ = S1+ / S2+; Zähler-Istwert innerhalb des Bereichs
 - ((D+)+2) = S+ > S1+ / S2+; Zähler-Istwert oberhalb des Bereichs
- Da es sich um eine High-Speed-Anweisung handelt, werden Ausgänge, die in (D+) angegeben werden, sofort geschaltet.

HINWEIS

In einem SPS-Programm für die Steuerungen der MELSEC FX-Familie dürfen 6 High-Speed-Anweisungen des Typs DHSCS, DHSCR oder DHSZ gleichzeitig aktiv sein.

Beispiel ▾

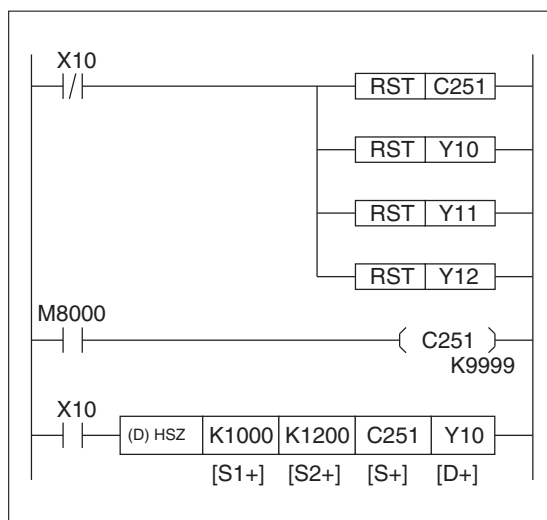
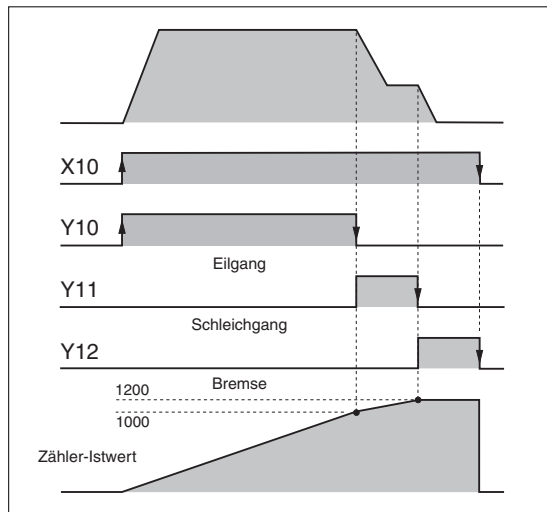


Abb. 6-90:
Programmierbeispiel zur DHSZ-Anweisung

- (S1+): Ende des Eilgangs (Start des Schleichgangs)
- (S2+): Ende des Schleichgangs (Einsetzen der Bremse)
- (S+): Definition des High-Speed-Counters
- (D+): Y10 -> Eilgang, Y11 -> Schleichgang, Y12 -> Bremse

Beispiel

Der Zähl- und Vergleichsvorgang sowie die externe Ausgabe werden im Interrupt-Modus ausgeführt.

**Abb. 6-91:**

Zeitverlauf für das Schalten der Ausgänge Y10, Y11, Y12

Entsprechend dem Istwert des Counters C251 werden im Beispiel die Ausgänge Y10 bis Y12 geschaltet. Ist X10 ausgeschaltet, sind die Ausgänge Y10 bis Y12 durch die RST-Funktion ebenfalls ausgeschaltet. Wird ein neuer Counter-Istwert z. B. mit Hilfe der MOV-Anweisung übertragen, bleibt der Ausgangsstatus bis zur Ausführung der nächsten Zähloperation unverändert. △

Funktionsweise mit Sondermerker M8130

Tabellenvergleich für High-Speed-Counter im Bereich ab S1+ und die folgenden n1

Beschreibung

- Die Anweisung DHSZ mit Sondermerker M8130 vergleicht im Interrupt-Betrieb den Istwert eines High-Speed-Counters mit den in einem Tabellenbereich angegebenen Werten.
- Der Operand (D+) ist bei dieser Spezialfunktion durch den Sondermerker M8130 gegeben.
- Die Tabellenlänge wird über einen Konstantenwert (K, H) angegeben. Die Maximallänge beträgt 128 Einträge. Je Eintrag der Tabelle werden 4 Datenregister belegt. Zu jedem Eintrag müssen die folgenden Angaben gespeichert werden:
 - der zu vergleichende Wert,
 - der zu adressierende Ausgang (hexadezimal),
 - die Setz- bzw. Rücksetz-Anweisung.
- Der Vergleich findet mit jedem Zählimpuls am angegebenen Zähler (S+) statt.

HINWEISE

In einem SPS-Programm für die MELSEC FX-Serie dürfen nur 6 High-Speed-Anweisungen des Typs DHSCS, DHSCR oder DHSZ gleichzeitig aktiv sein.

Es kann immer nur eine DHSZ-Anweisung den Sondermerker M8130 verwenden.

Es muss eine korrekte HSC-Anweisung gesetzt sein.

Die DHSZ-Anweisung mit Sondermerker M8130 wird erstmalig nach der ersten END-Anweisung ausgeführt. Dies erlaubt der Steuerung, die interne Vergleichstabelle aufzubauen.

Der Vergleich in der Tabelle erfolgt immer der Reihenfolge nach. Aus diesem Grund müssen die Vergleichswerte immer in aufsteigender oder absteigender Folge sortiert sein.

Beispiel ▾ Einsatz der DHSZ-Anweisung mit Sondermerker M8130

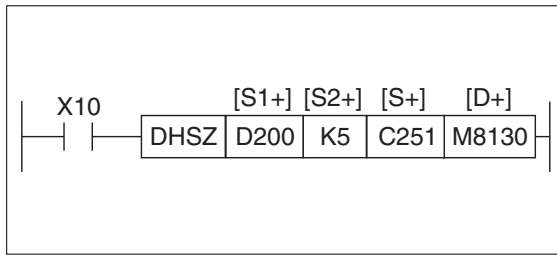


Abb. 6-92:
 Programmierbeispiel zur DHSZ-Anweisung mit Sondermerker M8130

Nach Setzen des Eingangs X10 erfolgt der Tabellenvergleich ab Datenregister D200 in 5 Einträgen der Datentabelle mit dem Wert in C251.

Die Tabelle hat die folgende Form (K1 = Setzen, K0 = Rücksetzen):

Eintrag Nr.	Vergleichswert		Ausgabe-Anweisung	Setzen/Zurücksetzen
	unterer	oberer		
0	D200	D201	D202	D203
	K123		H10 (=Y10)	K1
1	D204	D205	D206	D207
	K234		H10	K0
2	D208	D209	D210	D211
	K345		H23 (=Y23)	K1
3	D212	D213	D214	D215
	K456		H23	K0
4	D216	D217	D218	D219
	K567		H23	K1

Tab. 6-20:
 Vergleichstabelle, Startadresse D200, Länge K5

Der Vergleichswert ist im 32-Bit-Format (Doppelwort) gespeichert. Der Ausgang ist im Hexadezimal-Format angegeben.

Beispiel ▾

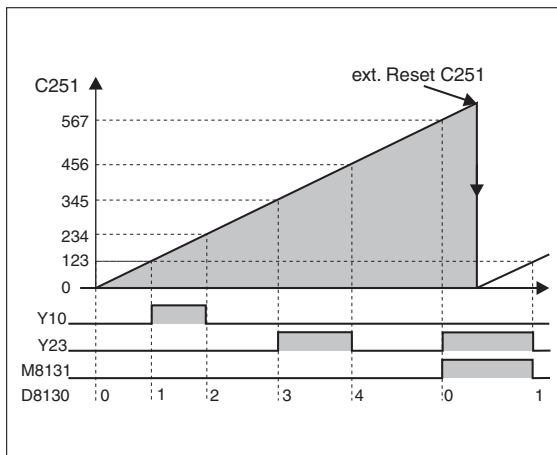


Abb. 6-93:
 Zeitverlauf für das Schalten der Ausgänge Y10, Y23

Wird die DHSZ-Anweisung mit dem Sondermerker M8130 gesetzt, wird Datenregister D8130 als Zähler der Eintragsnummern definiert. Nach jedem Vergleich springt D8130 zur nächsten Eintragsnummer.

Sind alle Einträge der Tabelle abgearbeitet, wird das Vorgangsende-Flag M8131 eingeschaltet und D8130 durch einen externen oder durch einen Programm-Impuls zurückgesetzt. D8130 startet erneut mit der Zählung, wenn der Sondermerker M8131 zurückgesetzt wird.



Funktionsweise mit Sondermerker M8132

Tabellenvergleich für High-Speed-Counter im Bereich ab S1+ und die folgenden n1 mit ergebnisabhängiger Frequenzsteuerung in der DPLSY-Anweisung.

Beschreibung

- Die Anweisung DHSZ mit Sondermerker M8132 vergleicht im Interrupt-Betrieb den Istwert eines High-Speed-Counters mit den in einem Tabellenbereich angegebenen Werten. Bei Übereinstimmung wird ein in der Tabelle angegebener Wert an die folgende DPLSY-Anweisung (FNC 57) zur Frequenzsteuerung ausgegeben.
- Der Operand (D+) ist bei dieser Spezialfunktion durch den Sondermerker M8132 gegeben.
- Die Tabellenlänge wird über einen Konstantenwert (K, H) angegeben. Die Maximallänge beträgt 128 Einträge. Je Eintrag der Tabelle werden 4 Datenregister belegt. Zu jedem Eintrag müssen die folgenden Angaben gespeichert werden:
 - der zu vergleichende Wert (32-Bit-Format),
 - der zu adressierende Ausgang (32-Bit-Format).
- Der Vergleich findet mit jedem Zählimpuls am angegebenen Zähler (S+) statt.

HINWEISE

In einem SPS-Programm für die FX-Serie dürfen nur 6 High-Speed-Anweisungen des Typs DHSCS, DHSCR oder DHSZ gleichzeitig aktiv sein.

Es kann immer nur eine DHSZ-Anweisung den Sondermerker M8132 verwenden.

Es muss eine korrekte HSC-Anweisung gesetzt sein.

Die DHSZ-Anweisung mit Sondermerker M8132 wird erstmalig nach der ersten END-Anweisung ausgeführt. Dies erlaubt der Steuerung, die interne Vergleichstabelle aufzubauen.

Der Vergleich in der Tabelle erfolgt immer der Reihenfolge nach. Aus diesem Grund müssen die Vergleichswerte immer in aufsteigender oder absteigender Folge sortiert sein.

Der letzte Eintrag in der Tabelle sollte auf (K0, K0) gesetzt werden, um sicherzustellen, dass die Impulsausgabe gestoppt wird und D8131 nicht zum Tabellenkopf zurückspringt. Register D8134 und D8135 erhalten den Wert K0 und geben das Ende der Tabelle an.

Beispiel ▾

Einsatz der DHSZ-Anweisung mit Sondermerker M8132

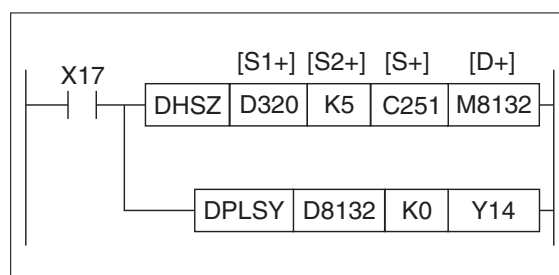


Abb. 6-94:

Programmierbeispiel zur DHSZ-Anweisung mit Sondermerker M8132

Nach Setzen des Eingangs X10 erfolgt der Tabellenvergleich ab Datenregister D200 in 5 Einträgen der Datentabelle mit dem Wert in C251.

Die Tabelle hat die folgende Form:

Eintrag Nr.	Vergleichswert		Ausgangsfrequenz	
	unterer	oberer	unterer	oberer
0	D320	D321	D322	D323
	K20		K300	
1	D324	D325	D326	D327
	K600		K500	
2	D328	D329	D330	D331
	K700		K200	
3	D332	D333	D334	D335
	K800		K100	
4	D336	D337	D338	D339
	K0		K0	

Tab. 6-21:
Vergleichstabelle, Startadresse D320, Länge K5

Der Vergleichswert ist im 32-Bit-Format (Doppelwort) gespeichert, der Ausgangswert gibt die Frequenz an, die gesetzt bleibt, bis das Vergleichsergebnis übereinstimmt.

Beispiel ▾

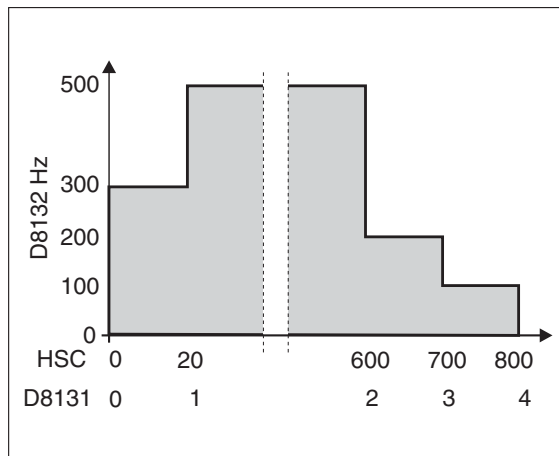


Abb. 6-95:
Frequenzverlauf bei Anwendung der DHSZ-Anweisung mit Sondermerker M8132

Wird die DHSZ-Anweisung mit dem Sondermerker M8132 gesetzt, wird Datenregister D8131 als Zähler der Eintragsnummern definiert. Nach jedem Vergleich springt D8131 zur nächsten Eintragsnummer.

D8132 erhält den jeweiligen Frequenzwert des Tabelleneintrags zur Verwendung in der PLSY-Anweisung. Zur Abarbeitung tragen die Datenregister D8134 und D8135 den zu vergleichenden Wert.

Sind alle Einträge der Tabelle abgearbeitet, wird das Vorgangsende-Flag M8133 eingeschaltet und D8131 durch einen externen oder durch einen Programm-Impuls zurückgesetzt. D8131 startet erneut mit der Zählung, wenn der Sondermerker M8133 zurückgesetzt wird.

Bei Zurücksetzen der DHSZ-Anweisung werden alle Werte, einschließlich der Frequenzausgabe, zurückgesetzt.

△

6.7.6 Geschwindigkeitserkennung (SPD, DSPD)

				SPD		FNC 56				
				Geschwindigkeitserkennung						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE		FX3U FX3UC
				●	●	●	●		●	
Operanden	S1+	S2+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	X0 bis X5 ^①	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^② , V, Z, U□\G□ ^③	T, C, D, R ^②			16 Bit	32 Bit	SPD	7	
					●	● ^②	DSPD	13		

① X0 bis X7 bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

③ Nur bei FX3U und FX3UC

Funktionsweise der 16-Bit-Anweisung

Erfassung der Anzahl der Impulse innerhalb einer vorgewählten Zeit

Beschreibung

- Die Impulse an (S1+) werden für (S2+) in der Einheit „ms“ gezählt, und das Ergebnis wird in (D+) abgelegt.
- Es werden die Operanden (D+), ((D+)+1) und ((D+)+2) belegt.
 (D+): Summe der Impulse nach Ablauf der Zeit
 ((D+)+1): Aktueller Zählwert innerhalb des Zeitintervalls
 ((D+)+2): Verbleibende zu zählende Zeit
- Durch die wiederholte Ausführung der SPD-Anweisung enthält (D+) die Anzahl der in der vorgegebenen Zeit erfassten Impulse. Dieser Wert ist proportional zur Drehzahl.

HINWEISE

Nach Ablauf der Zeit wird der Inhalt von ((D+)+1) nach (D+) übertragen, und ((D+)+1) wird zurückgesetzt.

Die in der Anweisung verwendeten High-Speed-Eingänge dürfen nicht in anderen High-Speed-Operationen verwendet werden.

Für jeden High-Speed-Eingang kann maximal eine SPD-Anweisung verwendet werden.

Beispiel ▾

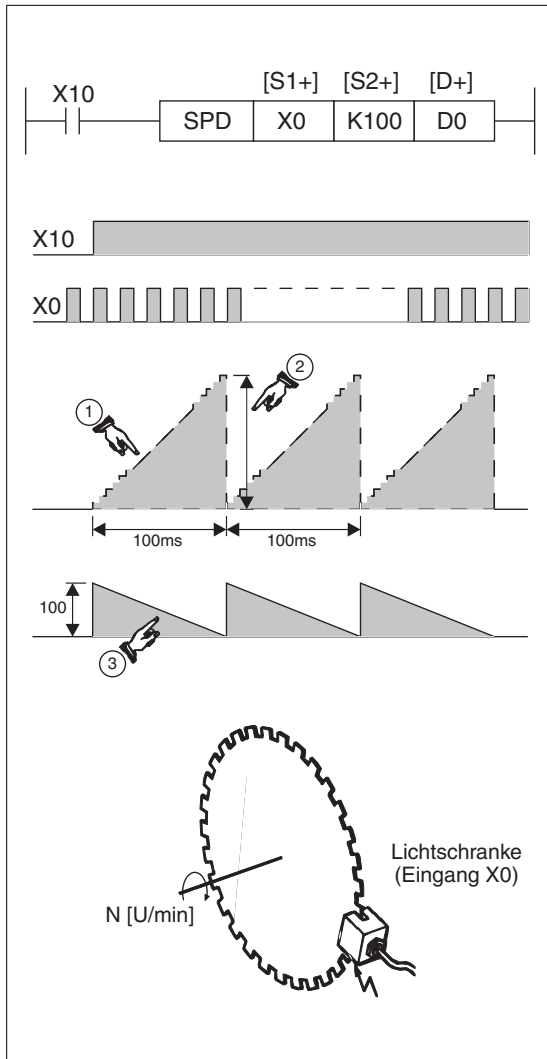


Abb. 6-96:
 Programmierbeispiel zur SPD-Anweisung

- ①: Istwert (D1)
- ②: Zähleristwert (D0)
- ③: Verbleibende Zeit (D2)

Im Beispiel zählt D1 die Anzahl der Einschaltvorgänge von X0. Nach 100 ms wird das Zählergebnis in D0 gespeichert.

D1 wird zurückgesetzt und beginnt erneut mit dem Zählen der Einschaltvorgänge von X0.

In D2 wird die jeweils verbleibende Zeit gemessen.

Mit diesem Wert kann die Drehzahl eines Antriebs ermittelt werden.

$$N = \frac{60 \times D0}{n \times t} \times 10^3 \text{ (U/min.)}$$

n: Impuls/Umdrehung

N: Drehzahl

t: Intervalle (ms), die in S2+ angegeben sind

△

Funktionsweise der 32-Bit-Anweisung (nur bei FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC)

Erfassung der Anzahl der Impulse innerhalb einer vorgewählten Zeit

Beschreibung

- Die Impulse an (S1+) werden für die in ((S2+)+1) und (S2+) angegebene Zeit in der Einheit „ms“ gezählt. Das Ergebnis wird in (D+) abgelegt.
- Es werden die sechs Operanden (D+), ((D+)+1), ((D+)+2), ((D+)+3), ((D+)+4) und ((D+)+5) belegt.
 - ((D+)+1), (D+): Summe der Impulse nach Ablauf der Zeit
 - ((D+)+3), ((D+)+2): Aktueller Zählwert innerhalb des Zeitintervalls
 - ((D+)+5), ((D+)+4): Verbleibende zu zählende Zeit
- Durch die wiederholte Ausführung der SPD-Anweisung enthalten ((D+)+1) und (D+) die Anzahl der in der vorgegebenen Zeit erfassten Impulse. Dieser Wert ist proportional zur Drehzahl.

HINWEISE

Nach Ablauf der Zeit wird der Inhalt von ((D+)+3) und ((D+)+2) nach ((D+)+1) und (D+) übertragen, und ((D+)+3)/((D+)+2) werden zurückgesetzt.

Die in der Anweisung verwendeten High-Speed-Eingänge dürfen nicht in anderen High-Speed-Operationen verwendet werden.

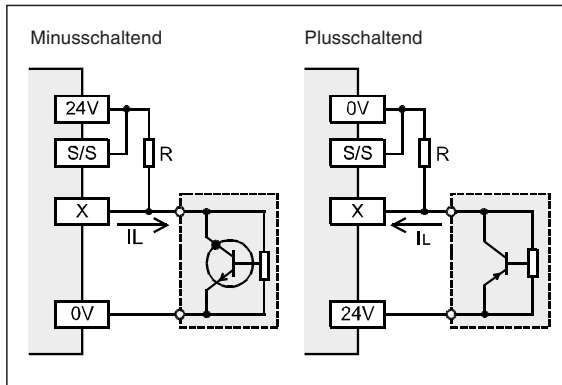
Für jeden High-Speed-Eingang kann maximal eine (D)SPD-Anweisung verwendet werden.

- Maximale Schaltfrequenzen der Eingänge
 - FX3G, FX3GC-, FX3GE-Serie
 - X000, X001, X003, X004: 60 kHz
 - X002, X005, X006, X007: 10 KHz
 - FX3S-Serie
 - X000, X001: 60 kHz
 - X002, X003, X004, X005: 10 KHz
 - FX3U- und FX3UC-Serie
 - X000 bis X005: 100 kHz
 - X006, X007: 10 kHz
- Die Eingänge des an ein Grundgerät der FX3U-Serie anschließbaren Adaptermoduls FX3U-4HSX-ADP können Impulse mit einer Frequenz von bis zu 200 kHz erfassen.

HINWEISE

Bei der Erfassung von Impulsen mit einer Frequenz von 60 bis 100 kHz sollte die Länge der Verdrahtung an den Eingängen 5 m nicht überschreiten.

Bei Frequenzen von 60 bis 100 kHz sollte am Eingang ein zusätzlicher Widerstand von $1,5\text{ k}\Omega$ und mindestens 1 W Belastbarkeit angeschlossen werden, der die Summe aus dem Ausgangsstrom des Open-Kollektor-Ausgangs des angeschlossenen externen Geräts und den Eingangstrom auf mindestens 20 mA erhöht.

**Abb. 6-97:**

Bei 2-Draht-Sensoren muss eventuell ein zusätzlicher Widerstand vorgesehen werden, der den Strom erhöht.

Beispiel ▾

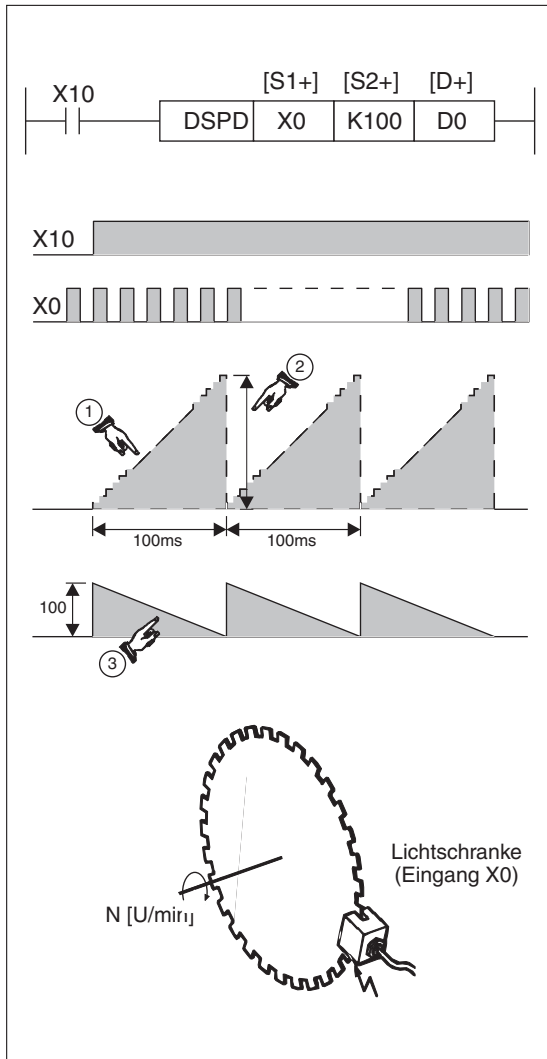


Abb. 6-98:
 Programmierbeispiel zur DSPD-Anweisung

- ①: Istwert (D3, D2)
- ②: Zähleristwert (D1, D0)
- ③: Verbleibende Zeit (D5, D4)

Im Beispiel zählt D3, D2 die Anzahl der Einschaltvorgänge von X0. Nach 100 ms wird das Zählergebnis in D1, D0 gespeichert.

D3 und D2 werden zurückgesetzt und es werden erneut die Einschaltvorgänge von X0 gezählt.

In D5, D4 wird die jeweils verbleibende Zeit gemessen.

Mit diesem Wert kann die Drehzahl eines Antriebs ermittelt werden.

$$N = \frac{60 \times [D1, D0]}{n \times t} \times 10^3 \text{ (U/min.)}$$

n: Impuls/Umdrehung

N: Drehzahl

t: Intervalle (ms), die in S2+ angegeben sind



6.7.7 Ausgabe einer definierten Anzahl von Impulsen (PLSY, DPLSY)

			PLSY		FNC 57				
			Ausgabe einer definierten Anzahl von Impulsen						
			CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
				●	●	●	●	●	●
Operanden	S1+, S2+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②	Nur Y0 oder Y1 ^③		16 Bit	32 Bit	PLSY	7		
				●	●	DPLSY	13		

- ① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ② Nur bei FX3U und FX3UC
- ③ Bei einer FX3U können die Impulse auch an einem Adaptermodul FX3U-2HSY-ADP ausgegeben werden.

Funktionsweise

Definierte Anzahl von Impulsen mit einer festgelegten Frequenz und einem festen Pulsweitenverhältnis von 50 : 50 über einen Ausgang ausgeben

Beschreibung

- Die Anweisung erzeugt eine definierte Anzahl von Impulsen.
- In (S1+) wird die Frequenz festgelegt
 - PLSY-Anweisung: 1 bis 32767 Hz
 - DPLSY-Anweisung: 1 bis 100 kHz (FX1S/FX1N und FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC-Grundgeräte),
2 bis 20 kHz (FX2N/FX2NC),
1 bis 200 kHz (Adaptermodul FX3U-2HSY-ADP)
- In (S2+) wird die Anzahl der zu erzeugenden Impulse angegeben. Dabei dürfen folgende Wertebereiche nicht überschritten werden.
 - PLSY-Anweisung: 1 bis 32 767 Impulse
 - DPLSY-Anweisung: 1 bis 2 147 483 647 Impulse

Wird der Wert 0 eingegeben, werden fortlaufend Impulse erzeugt.
- In (D+) wird die Adresse des Ausgangs festgelegt.
- Das Verhältnis zwischen EIN- und AUS-Zustand beträgt: 50 % EIN, 50 % AUS
Die EIN- und AUS-Zustände werden direkt im Interrupt-Modus ausgegeben.
- Bei Verwendung der DPLSY-Anweisung wird die Anzahl der Impulse in zwei aufeinanderfolgenden Datenregistern angegeben.
- Ist die gewünschte Anzahl von Impulsen erzeugt, wird der Sondermerker M8029 (Anweisung vollständig abgearbeitet) gesetzt. M8029 wird zurückgesetzt, wenn die PLSY-Anweisung deaktiviert wird. Falls im Programm noch andere Anweisungen verwendet werden, die ebenfalls M8029 beeinflussen, sollte der Zustand von M8029 unmittelbar nach der PLSY- bzw. der entsprechenden Anweisung abgefragt werden.

Die Daten in (S1+) (Frequenz) können während der Ausführung der Anweisung verändert werden. Veränderte Daten in (S2+) (Anzahl der Impulse) kommen jedoch erst dann zum Einsatz, wenn die Anweisung abgearbeitet ist.

HINWEISE

Max. zwei PLSY-Anweisungen können gleichzeitig in einem Programm verwendet werden. Die Impulsausgabe erfolgt dann einmal an Y0 und einmal an Y1. Möglich ist auch die Verwendung einer PLSY- und einer PLSR-Anweisung (FNC59), solange gewährleistet ist, dass die Impulsausgabe unabhängig an Y0 und Y1 erfolgt.

Die Impulse können nur über die Ausgänge Y0 und Y1 ausgegeben werden. Die Ausgabe-frequenz wird nicht von der Zykluszeit beeinflusst.

Die Verwendung von Unterprogrammen zur Eingrenzung mehrere PLSY-Anweisungen ist möglich, jedoch muss eine ausgeführte Anweisung gestoppt werden, bevor in einem Unter-programm die nächste PLSY-Anweisung gestartet wird.

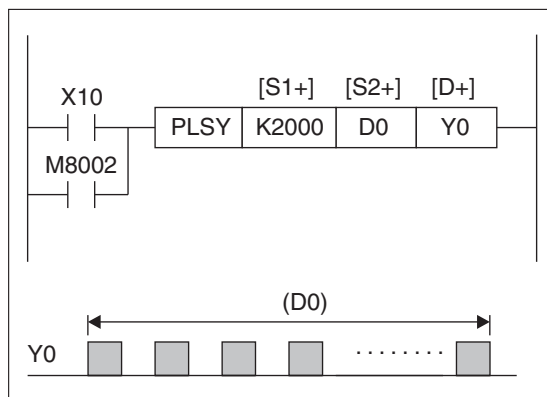
Verwenden Sie eine Steuerung mit Transistorausgängen.

Der Laststrom am Ausgang sollte bei der FX2N-Serie mindestens 200 mA betragen, damit eine einwandfreie Funktion auch bei max. Frequenz gewährleistet ist. Bei der FX1S- und FX1N-Serie sowie der FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC-Serie sollte der Strom zwischen 10 und 100 mA liegen.

Einsatz mit DHSZ-Anweisung siehe Seite 6-94.

Beispiel ▾

Einsatz der PLSY-Anweisung

**Abb. 6-99:**

Programmierbeispiel zum Einsatz der PLSY-Anweisung

Ist X10 eingeschaltet, werden Impulse mit einer Frequenz von 2000 Hz erzeugt. Es werden immer so viele Impulse erzeugt, wie im Datenregister D0 angegeben ist.

Die Erzeugung der Impulse wird eingestellt, wenn X10 ausgeschaltet wird. Wird X10 wieder eingeschaltet, beginnt die Operation erneut. Ist X10 nicht gesetzt, wird Y0 ausgeschaltet.

△

HINWEISE

Im Doppelwort D8140 und D8141 wird die Anzahl der an Y0 ausgegebenen Pulse ange-zeigt. Das Doppelwort D8142 und D8143 zeigt die Pulse von Y1.

In D8136 und D8137 wird bei der FX2N-, der FX2NC- und der FX3G/FX3GC/FX3GE/FX3S/ FX3U/FX3UC-Serie die Summe der Pulse an Y0 und Y1 hinterlegt.

Die Ausgabe der Impulse wird gestoppt, wenn die Eingangsbedingung der PLSY- oder DPLSY-Anweisung ausgeschaltet wird. Ist die Eingangsbedingung wieder gültig, beginnt die Impulsausgabe wieder von vorn.

Die Impulsausgabe kann auch durch Setzen der Sondermerker M8145 oder M8146 (FX1S/FX1N/FX3G/FX3GC/FX3GE/FX3S) bzw. M8349 oder M8359 (FX3G/FX3GC/FX3GE/ FX3S/FX3U/FX3UC) gestoppt werden. Um erneut Impulse auszugeben, muss der entspre-chende Merker zurückgesetzt und die Eingangsbedingung der PLSY- bzw. DPLSY-Anwei-sung wieder aktiviert werden.

6.7.8 Impulsausgabe mit Modulation der Impulsweite (PWM)

		PWM		FNC 58				
		Impulsausgabe mit Modulation der Impulsweite						
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
			●	●	●	●	●	●
Operanden	S1+, S2+	D+	Puls-Anweisung (P)		Verarbeitung		Programmschritte	
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^② (S1 ≤ S2)	Nur Y0, Y1, Y2 oder Y3 ^③			16 Bit	32 Bit	PWM	7
					●			

- ① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ② Nur bei FX3U und FX3UC
- ③ Die Impulsausgabe an Y2 ist nur bei einer FX3U oder FX3UC sowie einer FX3G-40M□ oder FX3G-60M□ möglich. Bei einer FX3U können die Impulse auch an einem Adaptermodul FX3U-2HSY-ADP über die Ausgänge Y0, Y1, Y2 oder Y3 ausgegeben werden.

Funktionsweise

Fortlaufend Impulse mit festgelegter Impulsweite und Periodendauer (bzw. Frequenz) über einen Ausgang ausgeben

Beschreibung

- Die Anweisung erzeugt fortlaufend Impulse. Das Verhältnis zwischen Impulsweite t und Periodendauer T0 wird überwacht.
 - t: Impulsweite [ms]
 - T0: Periodendauer [ms]
 - Frequenz f: 1/T0 [kHz]
- In (S1+) wird die Impulsweite im Bereich von t: 1 bis 32 767 ms festgelegt. Die Impulsweite muss im Bereich von 1 ≤ t ≤ T0 liegen.
- In (S2+) wird die Periodendauer im Bereich von T0: 1 bis 32 767 ms festgelegt.
- In (D+) wird die Adresse des Ausgangs angegeben.
- Die Überwachung des EIN- und AUS-Zustands des Ausgangs wird im Interrupt-Modus ausgeführt.

HINWEISE

- | Die Anweisung kann nur einmal in einem Programm eingesetzt werden.
- | Es können nur die Ausgänge Y0, Y1, Y2 und Y3 angesprochen werden.
- | Die Steuerung sollte über Transistorausgänge verfügen, um einen Kontaktverschleiß zu vermeiden.
- | Der Laststrom der Ausgänge sollte bei der FX2N-Serie mindestens 200 mA betragen, um eine einwandfreie Funktion auch bei der minimalen Periodendauer von T0 (1 ms) zu gewährleisten. Bei der FX1S- und FX1N-Serie sowie der FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC-Serie sollte der Strom zwischen 10 und 100 mA liegen.

Fehlerquelle

Ist der festgelegte Wert für die Impulsweite t in (S1+) größer als der festgelegte Wert für T0 in (S2+), tritt ein Programmablauffehler auf.

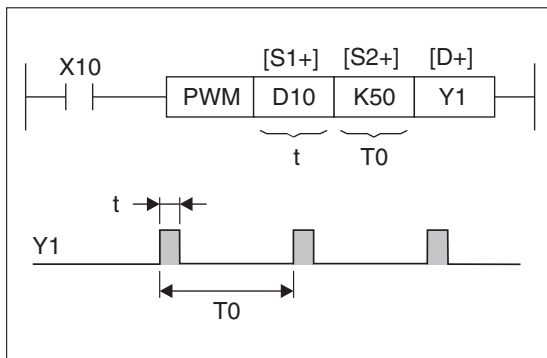
Beispiel ▾ Einsatz der PWM-Anweisung

Abb. 6-100:
Impulsausgabe an Y1 mit einer
PWM-Anweisung

Durch Ändern der Daten im Register D10 in einem Bereich von 0 bis 50 kann die relative Pulsweite T_0 von 0 % bis 100 % variiert werden. Wird der Wert D10 auf 0 gesetzt, wird kein Impuls ausgegeben. Wird der Wert von D10 auf 50 geändert, so ist Y1 für den ganzen Zyklus eingeschaltet.

Y1 wird ausgeschaltet, wenn X10 ausgeschaltet ist.

△

6.7.9 Ausgabe einer bestimmten Anzahl von Impulsen (PLSR)

		PLSR		FNC 59			
		Ausgabe einer bestimmten Anzahl von Impulsen					
CPU		FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
		●	●	●	●	●	●
Operanden	S1+, S2+, S3+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte	
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^② (S1 ≤ S2)	Nur Y0 oder Y1 ^③		16 Bit	32 Bit	PLSR	9
				●	●	DPLSR	17

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

③ Bei einer FX3U können die Impulse auch an einem Adaptermodul FX3U-2HSY-ADP ausgegeben werden.

Funktion

Eine festgelegte Anzahl von Impulsen mit festgelegter Frequenz an einen Ausgang ausgeben.

Beschreibung

- Die PLSR-Anweisung erzeugt an einem Ausgang eine angegebene Anzahl von Impulsen (S2+) mit angegebener Frequenz (S1+). Die Frequenz wird in einer angegebenen Zeit (S3+) in 10 Schritten hoch- und wieder heruntergefahren.
- Die Ausgabefrequenz kann bei einer SPS der FX2N- und FX2NC-Serie zwischen 10 und 20.000 Hz liegen. Bei Steuerungen der FX1S- oder FX1N-Serie können Frequenzen von 10 bis 100.000 Hz ausgegeben werden. Mit Grundgeräten der FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC-Serie sind Frequenzen von 1 bis 100.000 Hz möglich. Über ein an einem FX3U-Grundgerät angeschlossenes Adaptermodul FX3U-2HSY-ADP können Frequenzen von 10 bis 200.000 Hz ausgegeben werden. Die angegebene Frequenz sollte durch 10 teilbar sein. Ist die angegebene Frequenz nicht durch 10 teilbar, wird die Frequenz auf einen entsprechenden Wert aufgerundet.
- Die Schrittweite der Rampen beträgt 1/10 der angegebenen Ausgabefrequenz (bei der Verwendung von Schrittmotoren zu berücksichtigen).
- Maximale Anzahl der Ausgangsimpulse bei FX2N und FX2NC bis Version 3.00 sowie FX1S und FX1N:
 PLSR-Anweisung : 110 bis 32.767 Impulse
 DPLSR-Anweisung : 110 bis 2.147.483.647 Impulse
 Bei einer Angabe unter 110 Impulsen kann eine korrekte Impulsausgabe nicht garantiert werden.
- Maximale Anzahl der Ausgangsimpulse bei der FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC-Serie sowie der FX2N- und FX2NC-Serie ab Version 3.00:
 PLSR-Anweisung : 1 bis 32.767 Impulse
 DPLSR-Anweisung : 1 bis 2.147.483.647 Impulse
- Die Anstiegszeit der Rampe muss den unten beschriebenen Grenzwerten entsprechen.
- Als Ausgänge können nur Y0 und Y1 programmiert werden.
- Das Verhältnis zwischen EIN- und AUS-Zustand beträgt: 50 % EIN, 50 % AUS. Die EIN- und AUS-Zustände werden unabhängig vom Programmzyklus ausgegeben.
- Die Daten in (S1+), (S2+) und (S3+) können während der Ausführung der Anweisung verändert werden. Veränderte Daten kommen jedoch erst dann zum Einsatz, wenn die Anweisung das nächste Mal ausgeführt wird.

HINWEISE

In einem Programm können zur selben Zeit zwei PLSR-Anweisungen verwendet werden, um Impulse an Y0 und Y1 auszugeben. Möglich ist auch die Verwendung von einer PLSY-Anweisung (FNC57) und einer PLSR-Anweisung (FNC59) in einem Zyklus, um Impulse an Y0 und Y1 auszugeben. Eine mehrfache Verwendung kann durch Unterprogramme oder ähnliche Verfahren realisiert werden.

Wenn die Anzahl der angegebenen Impulse nicht ausreicht, um die angegebene Frequenz zu erreichen, wird die Frequenz abgeschnitten.

Der Sondermerker M8029 wird nach Ausgabe der angegebenen Anzahl von Impulsen gesetzt. Das Rücksetzen des Merkers erfolgt mit Rücksetzen der Ausführungsbedingung der PLSR-Anweisung. Falls im Programm noch andere Anweisungen verwendet werden, die ebenfalls M8029 beeinflussen, sollte der Zustand von M8029 unmittelbar nach der PLSY- bzw. der entsprechenden Anweisung abgefragt werden.

Begrenzung der Rampenanstiegszeit

Die Anstiegszeit (S3+) ist auf 5.000 ms begrenzt. Die Grenzwerte der Anstiegszeit in Abhängigkeit der Frequenz und Anzahl der Ausgabeimpulse werden wie folgt berechnet:

- Der Wert in (S3+) muss mindestens 10 mal größer als die Programmzykluszeit sein (D8012). Bei Unterschreitung des Wertes erfolgt der Rampenanstieg in ungeraden Schritten.
- Der Minimalwert für (S3+) errechnet sich nach folgender Gleichung:
$$(S3+) \geq (9.000 / (S1+)) \times 5$$
Bei der FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC-Serie ist der Minimalwert auf 50 ms begrenzt.
- Der Maximalwert für (S3+) errechnet sich nach folgender Gleichung:
$$(S3+) \leq ((S2+) / (S1+)) \times 818$$
- Wenn die Parameter außerhalb der berechneten Grenzen liegen, ist der Wert von (S1+) zu verkleinern.
- Der Anstieg der Ausgabefrequenz erfolgt in 10 Schritten.

Hinweise

Im Doppelwort D8140 und D8141 wird die Anzahl der an Y0 ausgegebenen Pulse angezeigt. Das Doppelwort D8142 und D8143 zeigt die Pulse von Y1.

In D8136 und D8137 wird bei der FX2N-, der FX2NC- und der FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC-Serie die Summe der Pulse an Y0 und Y1 hinterlegt.

Verwenden Sie eine Steuerung mit Transistorausgängen.

Der Laststrom am Ausgang sollte bei der FX2N-Serie mindestens 200 mA betragen, damit eine einwandfreie Funktion auch bei max. Frequenz gewährleistet ist. Bei der FX1S- und FX1N-Serie sowie der FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC-Serie sollte der Strom zwischen 10 und 100 mA liegen.

Nach dem Rücksetzen der Ausführungsbedingung der PLSR-Anweisung werden die angesprochenen Ausgänge zurückgesetzt. Bei erneutem Setzen der Ausführungsbedingung beginnt die Verarbeitung der Anweisung von vorne.

Die Ausgabe der Impulse wird gestoppt, wenn die Eingangsbedingung der PLSY- oder DPLSY-Anweisung ausgeschaltet wird. Ist die Eingangsbedingung wieder gültig, beginnt die Impulsausgabe wieder von vorn.

Die Impulsausgabe kann auch durch Setzen der Sondermerker M8145 oder M8146 (FX1S/FX1N/FX3G/FX3GC/FX3GE/FX3S) bzw. M8349 oder M8359 (FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC) gestoppt werden. Um erneut Impulse auszugeben, muss der entsprechende Merker zurückgesetzt und die Eingangsbedingung der PLSY- bzw. DPLSY-Anweisung wieder aktiviert werden.

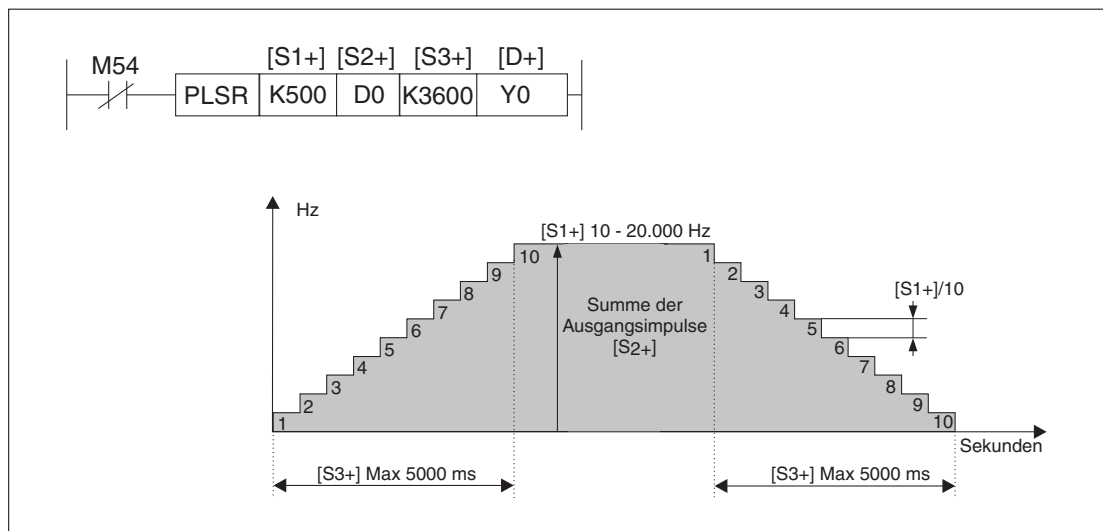
Beispiel ▾

Abb. 6-101: Programmierbeispiel zum Einsatz der PLSR-Anweisung

Mit Rücksetzen des Merkers M54 wird die in D0 (S2+) angegebene Anzahl von Impulsen an Y0 (D+) ausgegeben.

Die Ausgabefrequenz beträgt 500 Hz (S1+).

Der Frequenzanstieg auf 500 Hz (S1+) und die Frequenzabsenkung auf 0 Hz erfolgt in jeweils 3600 ms (S3+) in Schritten zu 50 Hz (S1+ / 10).



6.8 Anwendungsbezogene Anweisungen

Übersicht der Anweisungen FNC 60 bis 69

Symbol	FNC	Bedeutung	Abschnitt
IST	60	Schrittstatus initialisieren	6.8.1
SER	61	Suchanweisung	6.8.2
ABSD	62	Absoluter Counter-Vergleich	6.8.3
INCD	63	Inkrementaler Counter-Vergleich	6.8.4
TTMR	64	Teaching-Timer	6.8.5
STMTR	65	Sonder-Timer	6.8.6
ALT	66	Flip-Flop-Funktion	6.8.7
RAMP	67	Rampenfunktion	6.8.8
ROTC	68	Rundtisch-Positionierung	6.8.9
SORT	69	Sortieranweisung	6.8.10

Tab. 6-22: Übersicht der anwendungsbezogenen Anweisungen

6.8.1 Schrittstatus initialisieren (IST)

		IST		FNC 60				
		Schrittstatus initialisieren						
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
		●	●	●	●	●	●	
		Operanden	S+	D1+, D2+	Puls-Anweisung (P)	Verarbeitung		Programmschritte
	X, Y, M, S ^① , D□.b ^②	FX1S: S20–S127, FX1N: S20–S999 FX2N: S20–S999, FX3G/FX3GC/FX3GE/FX3U/FX3UC: S20–S899 und S1000 bis S4095 FX3S: S20–S255 (D1+ < D2+)		16 Bit	32 Bit	IST	7	
				●				

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktionsweise

Zuweisen von Sonderfunktionen und Reservieren von Schrittstatusoperanden für eine Schrittsteuerung. Mit der IST-Anweisung können verschiedene Schrittketten über ein Schaltpult gekoppelt werden. Es können Schrittketten für Automatikbetrieb, Handbetrieb und Nullpunktfahrt initialisiert werden.

Beschreibung

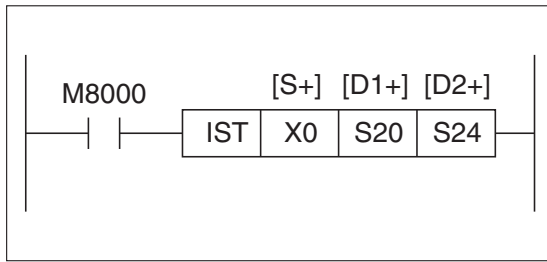
- Die Schrittstatusoperanden S0 bis S2 werden für die Initialisierung der Schrittketten
 - Handbetrieb,
 - Automatikbetrieb und
 - Nullpunktfahrt reserviert.
- Die Schrittstatusoperanden S0 bis S2 benötigen keine SET-Anweisung.
- Die Schrittstatusoperanden S3 bis S9 bleiben frei.
- Die Schrittstatusoperanden S10 bis S19 werden für die Schrittkette der Nullpunktfahrt reserviert.
- Für die Programmierung der übrigen Schrittketten stehen also die verbleibenden Schrittstatusoperanden ab S20 zur Verfügung.
- Die Sondermerker M8040 bis M8043, M8045 und M8047 werden gesteuert.
- In (D1+) und (D2+) wird der Schrittstatusoperandenbereich für die Schrittkette des Automatikbetriebs festgelegt. Dabei gilt: (D1+) < (D2+).
- In (S+) wird der Eingangsbereich der Steuereingänge festgelegt. Als Steuereingänge können die Operanden X, Y oder M verwendet werden. Es wird die Startadresse des Operandenbereichs angegeben.

HINWEIS

| Die IST-Anweisung darf nur einmal in einem Programm verwendet werden.

Beispiel ▾

Einsatz der IST-Anweisung

**Abb. 6-102:**

Programmierbeispiel zum Einsatz der IST-Anweisung

Die Eingänge erhalten in diesem Beispiel folgende Sonderfunktionen:

- X0: Handbetrieb
- X1: Rückkehr in die Ausgangsposition
- X2: Schrittbetrieb
- X3: Einzeloperation
- X4: Automatikbetrieb
- X5: Taster zur Rückkehr in die Ausgangsposition
- X6: Taster zum Starten des Automatikbetriebs
- X7: Taster zum Stoppen des Automatikbetriebs

**HINWEIS**

Diese Steuereingänge dienen der direkten Steuerung des Prozesses. Mit den Steuereingängen werden die Betriebsarten angewählt.

Die folgenden Sondermerker werden durch die IST-Anweisung beeinflusst bzw. steuern die Verarbeitung der Schrittketten:

- M8040: Weiterschalten in einen anderen Schritt möglich
- M8041: Beginn der Weiterschaltung (Wird bei einem STOP der SPS zurückgesetzt.)
- M8042: Startimpuls (Wird bei einem STOP der SPS zurückgesetzt.)
- M8043: Nullpunkt erreicht
- M8045: Rücksetzen aller Ausgänge sperren
- M8047: STL-Status anzeigen (Wird bei der Verarbeitung der END-Anweisung gesetzt.)

Beispiel ▾

Ein Containerfahrzeug für Schüttguttransporte soll im Automatikbetrieb ständig be- und entladen werden.

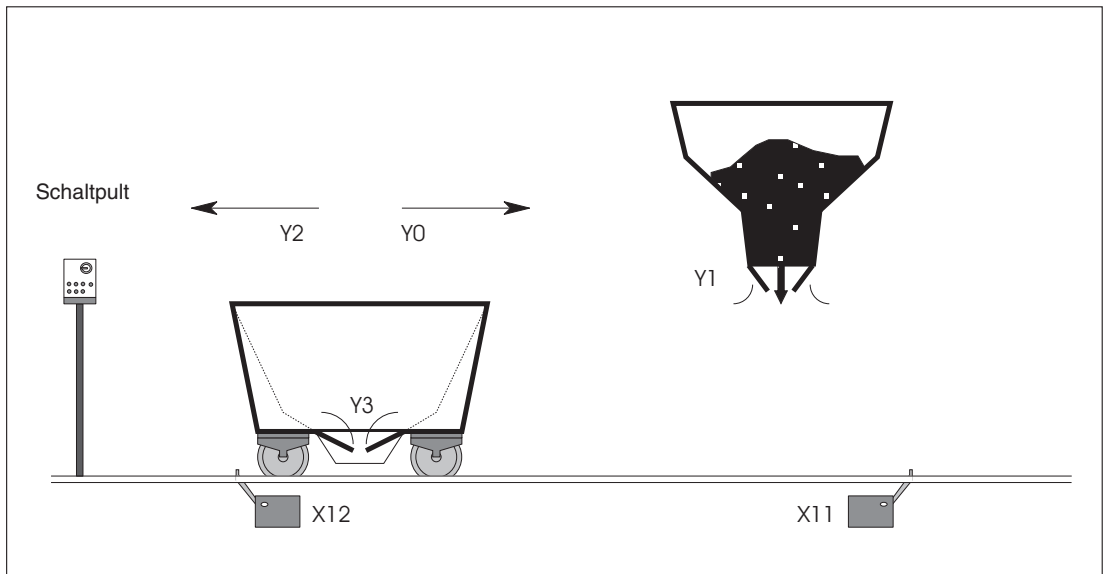


Abb. 6-103: Be- und Entladekontrolle eines Containerfahrzeuges mit Hilfe einer IST-Anweisung

Beschreibung der Signalgeber und Stellglieder

- Endschalter
links: X12
rechts: X11
- Fahrrichtung des Wagens
links: Y2
rechts: Y0
- Silolade öffnet sich für 7 Sekunden: Y1
- Entladeklappe des Wagens öffnet sich für 5 Sekunden: Y3

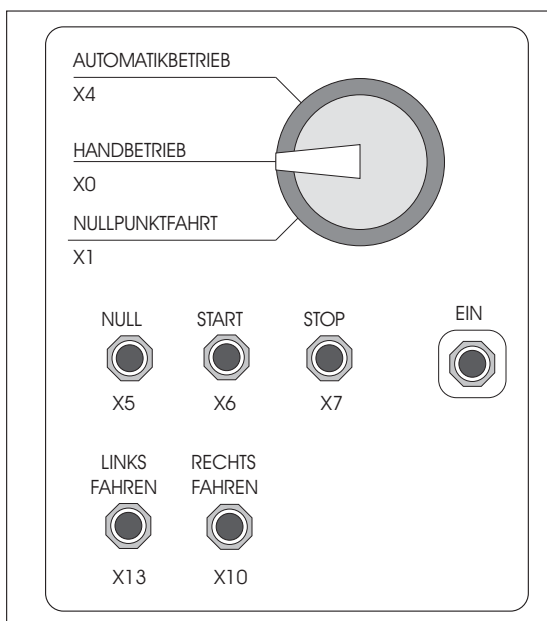


Abb. 6-104: Muster eines Schaltpults

Beispiel ▾ Durch Einsatz der IST-Anweisung vorgegebene Sonderfunktionen:

- X0: Handbetrieb
- X1: Rückkehr in die Ausgangsposition
- X2: Schrittbetrieb
- X3: Einzeloperation
- X4: Automatikbetrieb
- X5: Taster zur Rückkehr in die Ausgangsposition
- X6: Taster zum Starten des Automatikbetriebs
- X7: Taster zum Stoppen des Automatikbetriebs

Funktionsbeschreibung

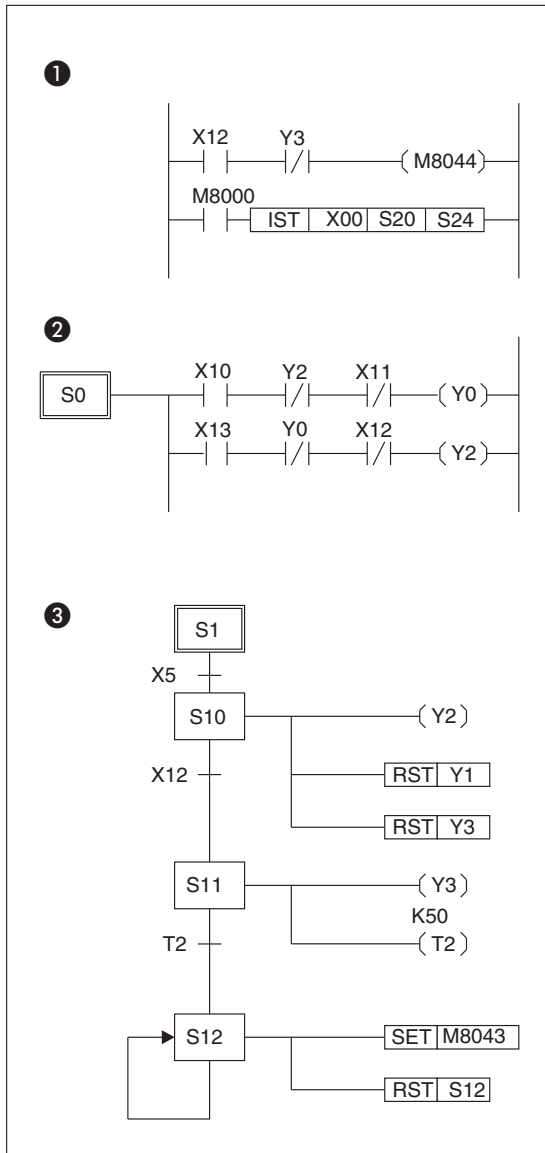
Im Automatikbetrieb soll das Containerfahrzeug ständig zwischen der Be- und Entladestation pendeln. Der Automatikbetrieb wird gestartet, wenn X4 eingeschaltet ist und X6 betätigt wird. Als weitere Einschaltbedingung gilt, dass das Fahrzeug vorher in die Ausgangsposition (Nullpunkt) gefahren wurde (M8043 und M8044 aktiviert). Der Automatikbetrieb wird mit X7 beendet.

Im Handbetrieb kann das Fahrzeug in beiden Bewegungsrichtungen verfahren werden. Hierfür muss X0 eingeschaltet sein. Das Fahrzeug fährt durch Betätigen des Tasters X13 bzw. X10 nach links bzw. nach rechts.

Die Nullpunktfahrt wird durch einen eingeschalteten X1 und einer Betätigung von X5 gestartet. Dadurch wird das Fahrzeug aus jeder beliebigen Position wieder in die Ausgangsposition zurückgefahren.

Die Ausgangsposition ist erreicht, wenn sich das Fahrzeug an der Endschalterposition X12 befindet und entleert ist.

△

Beispiel ▾**Abb. 6-105:**

Programmierbeispiel zur Be- und Entladekontrolle eines Containerfahrzeuges mit Hilfe der IST-Anweisung

Folgende drei Hauptbereiche sind erforderlich:

- ① Initialisierung
- ② Handbetrieb
- ③ Nullpunktfahrt

Die Schrittkette für den Handbetrieb ermöglicht das manuelle Verfahren des Containerfahrzeuges.

Diese Schrittkette für die Nullpunktfahrt lässt das Containerfahrzeug aus einer beliebigen Position zurück in die Ausgangsposition fahren. Dort wird das Containerfahrzeug entleert.

Beispiel ▾

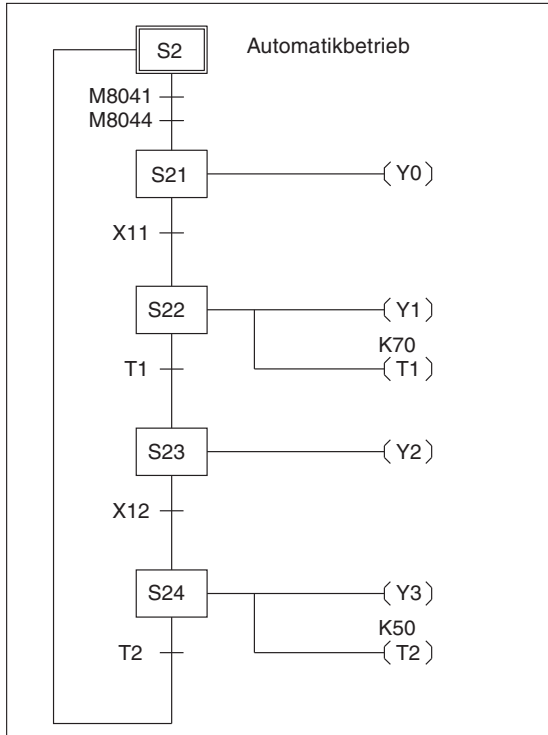


Abb. 6-106:

Programmierbeispiel zur Be- und Entladekontrolle eines Containerfahrzeuges mit Hilfe der IST-Anweisung (Automatikbetrieb)

Initialisierung, Bedingung für erreichten Nullpunkt	Nullposition anfahen	Automatik- betrieb	Hand- betrieb
LD X 12	STL S 1	STL S 2	STL S 0
ANI Y 3	LD X 5	LD M8041	LD X 10
OUT M8044	SET S 10	AND M8044	ANI Y 2
LD M8000	STL S 10	SET S 21	OUT Y 0
IST	RST Y 1	STL S 21	LD X 13
X 0	RST Y 3	OUT Y 0	ANI Y 0
S 20	OUT Y 2	LD X 11	OUT Y 2
S 24	LD X 12	SET S 22	RET
	SET S 11	STL S 22	END
	STL S 11	OUT Y 1	
	OUT Y 3	OUT T 1	
	OUT T 2	K 70	
	K 50	LD T 1	
	LD T 2	SET S 23	
	SET S 12	STL S 23	
	STL S 12	OUT Y 2	
	SET M8043	LD X 12	
	RST S 12	SET S 24	
	RET	STL S 24	
		OUT Y 3	
		OUT T 2	
		K 50	
		LD T 2	
		OUT S 2	
		RET	

Abb. 6-107: Programmierbeispiel einer Anweisungsliste zur Be- und Entladekontrolle eines Containerfahrzeuges mit Hilfe der IST-Anweisung



6.8.2 Suchanweisung (SER)

					SER		FNC 61				
					Suchanweisung						
					CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
								●	●	●	●
Operanden	S1+	S2+	D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	KnX, KnY, KnM, KnS,T, C, D, R ^① , U□\G□ ^②	K, H, KnX, KnY, KnM, KnS,T, C, D, R ^① , V, Z, U□\G□ ^②	KnY, KnM, KnS,T, C, D, R ^① , U□\G□ ^②	K, H, D, R ^①	●		16 Bit	32 Bit	SER/SERP	9	
						●	●	DSER/DSERP	17		

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktionsweise

Durchsuchen eines Datenbereichs nach einem Suchwert

Beschreibung

- Der Datenbereich von (S1+) bis ((S1+) + n) wird nach (S2+) durchsucht, und das Suchergebnis wird in den Datenregistern ab (D+) gespeichert. Gleichzeitig wird der kleinste und der größte Wert im Suchbereich ermittelt und gespeichert.
- Die Länge des Datenbereichs n ist für Daten im 16-Bit-Format auf maximal 256 und für Daten im 32-Bit-Format auf maximal 128 festgelegt.
- Das Suchergebnis wird in 5 Datenregistern, bei 32-Bit-Format in 10 Datenregistern, abgelegt. Es enthält:
 - Anzahl der mit dem Suchwert übereinstimmenden Werte im Suchbereich (0 bei keiner Übereinstimmung)
 - Die Position des ersten übereinstimmenden Wertes (0 bei keiner Übereinstimmung)
 - Die Position des letzten übereinstimmenden Wertes (0 bei keiner Übereinstimmung)
 - Die Position des kleinsten in dem Bereich auftretenden Wertes. Tritt dieser Wert mehrfach auf, wird die letzte Position gespeichert.
 - Die Position des größten in dem Bereich auftretenden Wertes. Tritt dieser Wert mehrfach auf, wird die letzte Position gespeichert.

Beispiel ▾ Einsatz der SER-Anweisung

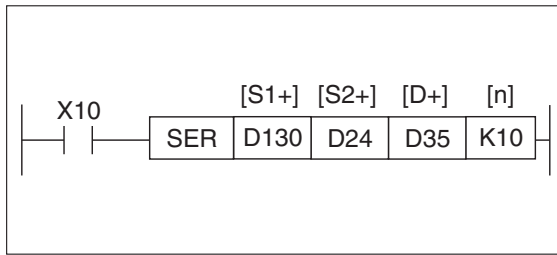


Abb. 6-108:
 Programmierbeispiel zum Einsatz der SER-Anweisung

Das obige Beispiel zeigt die Suche nach D24 = K100 ab D130 mit Bereichlänge K10. Das Ergebnis wird in D35 bis D39 abgelegt.

Der Suchbereich lässt sich wie folgt darstellen:

Position	Suchliste	Suchergebnis „=“	Maximum	Minimum
0	D130 = K100	●		
1	D131 = K111			
2	D132 = K100	●		
3	D133 = K98			
4	D134 = K123			
5	D135 = K66			●
6	D136 = K100	●		
7	D137 = K95			
8	D138 = K78			
9	D139 = K210		●	

Tab. 6-23:
 Suchbereich

Das Suchergebnis lässt sich wie folgt darstellen:

Ergebnisliste	Inhalt	Bedeutung
D35	3	Anzahl Suchergebnis „=“
D36	0	Erste Position Übereinstimmung
D37	6	Letzte Position Übereinstimmung
D38	5	Position kleinster Wert
D39	9	Position größter Wert

Tab. 6-24:
 Ergebnisliste



6.8.3 Absoluter Counter-Vergleich (ABSD)

					ABSD		FNC 62				
					Absoluter Counter-Vergleich						
					CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
					●	●	●	●	●	●	
Operanden	S1+	S2+	D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	KnX, KnY, KnM, KnS ^① , T, C ^② , D, R ^③ , U□G□ ^④	C ^②	Y, M, S, D□.b ^④	K, H n ≤ 64		16 Bit	32 Bit	ABSD	9		
						●	●	DABSD	17		

- ① X, Y, M und S müssen ein Vielfaches von 16 sein. Bei einer ABSD-Anweisung muss Kn = 4 und bei einer DABSD-Anweisung muss Kn = 8 sein.
- ② Bei einer 32-Bit-Anweisung (DABSD) sind High-Speed-Counter nicht zulässig.
- ③ Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ④ Nur bei FX3U und FX3UC

Funktionsweise

Schalten von Bits in Abhängigkeit von Zählerständen

Beschreibung

- Der Istwert des Zählers (S2+) wird mit einer Tabelle von Ein- und Ausschaltwerten verglichen.
- Die Tabelle wird aus Wortoperanden gebildet. Der erste Operand ist (S1+). Die Tabelle enthält n Zeilen.

Einschaltwert	Abschaltwert	zu schaltender Operand
(S1+) +1	(S1+) +1	D
(S1+) +2	(S1+) +3	D + 1
(S1+) +4	(S1+) +5	D + 2
⋮	⋮	⋮
(S1+) + (2n+1)	(S1+) + 2n	D + 3

Abb. 6-109:

Tabelle aus Wortoperanden für die 16-Bit-Anweisung ABSD

- Wenn der Zähler (S2+) einen der in der Tabelle hinterlegten Istwerte erreicht, wird der zugeordnete Operand geschaltet.
- Die hinterlegten Werte in der Tabelle dürfen in einer 16-Bit-Anweisung im Wertebereich von 0 bis 32 767 und bei 32-Bit-Anweisung (DABSD) im Bereich von 0 bis 2.147.483.647 liegen.
- Die Werte können z. B. mit einer MOV-Anweisung in die Tabelle geschrieben werden.

HINWEISE

Es werden immer 2 Zähler belegt, (S2+) und ((S2+)+1). In (S1+) sollten gerade Operandenadressen verwendet werden. Falls in (S1+) Bit-Operanden angegeben werden, muss die Operandenadresse ein Vielfaches von 16 (16, 32, 64 usw.) oder 0 sein. Bei einer ABSD ist in diesem Fall nur der Faktor K4 und bei einer DABSD-Anweisung ist nur der Faktor K8 zulässig.

Eine ABSD-Anweisung darf nur einmal im Programm genutzt werden.

Beispiel ▾

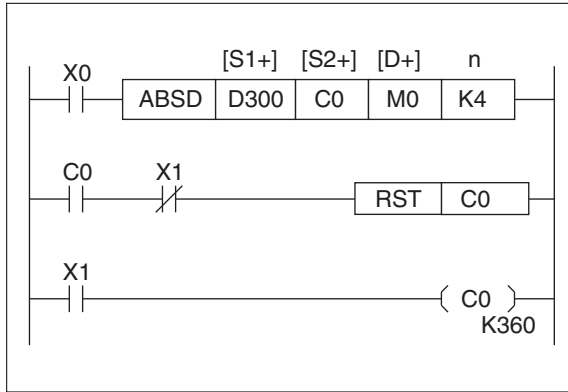


Abb. 6-110:
 Programmierbeispiel zur ABSD-
 Anweisung

Mit dem Programmierbeispiel in der oberen Abbildung wird der EIN-/AUS-Status der Merker M0 bis M3 bei einer Drehung eines Rundtisches kontrolliert (siehe auch Abschnitt 6.8.9).

In (S1+) müssen die Adressen der Operanden X, Y, M und S ein Vielfaches von 8 sein. In (S2+) wird der Counter (C0 bis C199) angegeben. Mit n wird die Anzahl der ein- und auszuschaltenden Zieloperanden (D+) und demzufolge auch die Anzahl der Operanden (S1+), in die die Ein- und Ausschaltwerte geschrieben werden, festgelegt.

Da n = 4, stehen die Merker M0 bis M3 für die Ein- und Ausschaltvorgänge zur Verfügung.

Die vier Einschaltwerte werden in die Datenregister D300, D302, D304 und D306 geschrieben. Die vier Ausschaltwerte werden in die Datenregister D301, D303, D305 und D307 geschrieben.

Für die Einschaltwerte sind die Operanden mit geraden Adressen zu verwenden. Die Ausschaltwerte werden in die Operanden mit ungeraden Adressen geschrieben. Die Ein- und Ausschaltwerte werden mit der MOV-Anweisung in die Datenregister D300 bis D307 geschrieben.

Einschaltwert	Ausschaltwert	Ausgabe
D300 = 40	D301 = 140	M0
D302 = 100	D303 = 200	M1
D304 = 160	D305 = 60	M2
D306 = 240	D307 = 280	M3

Tab. 6-25:
 Ein- und Ausschaltwerte

Wenn X0 eingeschaltet ist, schalten die Merker M0 bis M3 entsprechend der folgenden Abbildung. Ist X0 ausgeschaltet, werden die Merker nicht geschaltet.

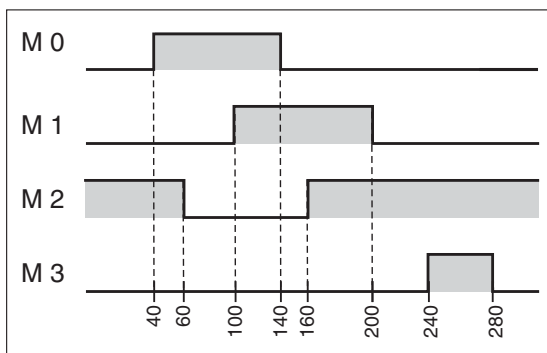


Abb. 6-111:
 Signalerlauf zum Ein- und Ausschalten der
 Merker



6.8.4 Inkrementaler Counter-Vergleich (INCD)

					INCD		FNC 63				
					Inkrementaler Counter-Vergleich						
					CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
						●	●	●	●	●	●
Operanden	S1+		S2+	D+	n	Puls-Anweisung (P)		Verarbeitung		Programmschritte	
	KnX, KnY, KnM, KnS ^① , T, C ^② , D, R ^③ , U□G□ ^④		C ^②	Y, M, S, D□.b ^④	K, H n ≤ 64			16 Bit	32 Bit	INCD	
								●		9	

- ① X, Y, M, S müssen ein Vielfaches von 16 sein. Kn muss „4“ sein.
- ② High-Speed-Counter sind nicht zulässig.
- ③ Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ④ Nur bei FX3U und FX3UC

Funktionsweise

Schalten von Bits in Abhängigkeit von zwei Zählerständen

Beschreibung

- n Bit-Operanden, ausgehend von (D+), werden in Abhängigkeit der Zähler (S2+) und ((S2+)+1) geschaltet.
- In (S1+) werden die Sollwerte für die Einschaltpunkte der Bit-Operanden (D+) vorgegeben.
- Der Zähler (S2+) muss im SPS-Programm programmiert werden; der Sollwert muss größer als der größte Schaltwert in (S1+) sein.
- Der Zähler ((S2+)+1) zählt die Rücksetzvorgänge bei Zähler (S2+).
- Durch das Ausschalten der Einschaltbedingung der Anweisung werden die Zähler (S2+) und ((S2+)+1) sowie die n Bit-Operanden (D+) zurückgesetzt.
- Nachdem der letzte Bit-Operand geschaltet wurde, wird ((S2+)+1) zurückgesetzt und M8029 eingeschaltet.

HINWEISE

Die INCD-Anweisung darf nur einmal im SPS-Programm verwendet werden.

Falls in (S1+) Bit-Operanden angegeben werden, muss die Operandenadresse ein Vielfaches von 16 (16, 32, 64 usw.) oder 0 sein.

Beispiel ▾

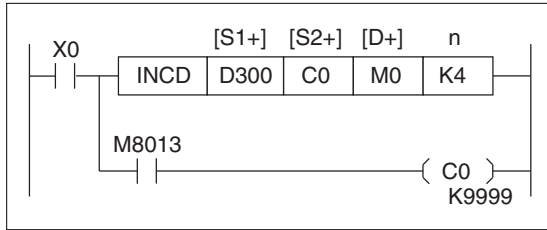


Abb. 6-113:
 Programmierbeispiel zur INCD-Anweisung

Speicher für Vorgabewert	Vorgabewert / Zählwert von C0	Zählwert von C1
D300	20	0
D301	30	1
D302	10	2
D303	40	3

Tab. 6-26:
 Beispiel von Vorgabewerten

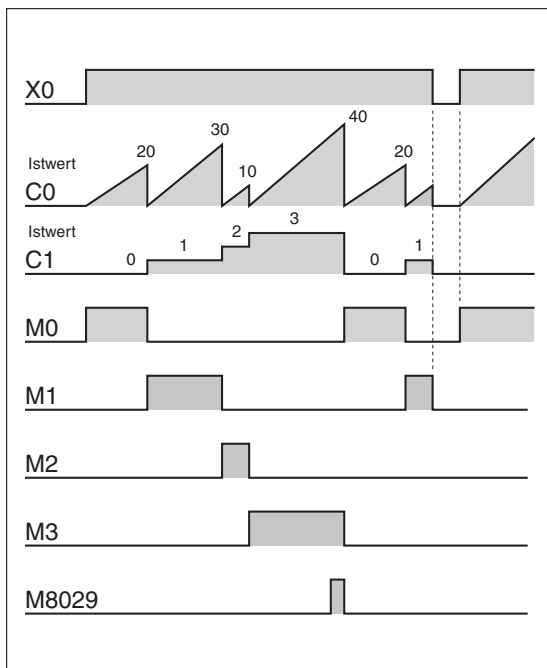


Abb. 6-112:
 Zeitverlauf entsprechend obigen Beispiel zum Ein-/Aus-schalten der Merker

Der Counter C0 wird automatisch zurückgesetzt, wenn der in D300 bis D303 festgelegte Wert erreicht wurde.

Der Counter C1 zählt die Anzahl der Rücksetzvorgänge an C0.

Die Merker M0 bis M3 schalten entsprechend der Werte des Counters C1.

Das Flag M8029 wird gesetzt, wenn der letzte Bit-Operand geschaltet wurde. Anschließend beginnt der Counter-Vergleich erneut.

Die Counter C0 und C1 werden gelöscht, wenn X0 ausgeschaltet wird; M0 bis M3 werden ebenfalls ausgeschaltet. Wird X0 wieder eingeschaltet, beginnt der Vorgang erneut.

△

6.8.5 Teaching-Timer (TTMR)

			TTMR		FNC 64				
			Teaching-Timer						
			CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
						●			●
Operanden	D	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	D, R*	K, H, D*, R* n = 0-2		16 Bit	32 Bit	TTMR	5		
				●					

* Nur bei FX3U und FX3UC

Funktionsweise

Vorgeben eines Timer-Sollwertes über eine Tastenbetätigung

Beschreibung

- Es wird die Einschaltdauer der Anweisung (in Sekunden) gemessen, multipliziert und in das Datenregister ((D+)+1) geschrieben.
- Durch (n) wird der Multiplikator der Zeit festgelegt.

$$\begin{aligned}
 n = 0 &\rightarrow D+ = [(D+) + 1] \times 1 \\
 n = 1 &\rightarrow D+ = [(D+) + 1] \times 10 \\
 n = 2 &\rightarrow D+ = [(D+) + 1] \times 100
 \end{aligned}$$

HINWEISE

Durch die Anweisung TTMR werden 2 Datenregister (D+) und ((D+)+1) belegt.

(D+) enthält die multiplizierte Betätigungszeit (s). Der Operand ((D+)+1) enthält die gemessene Betätigungsdauer (s).

Wenn die Bedingung zur Ausführung der Anweisung nicht mehr erfüllt ist, wird der Inhalt von ((D+)+1) gelöscht. Der Inhalt von (D+) ändert sich danach nicht mehr.

Beispiel ▾

Einsatz der TTMR-Anweisung

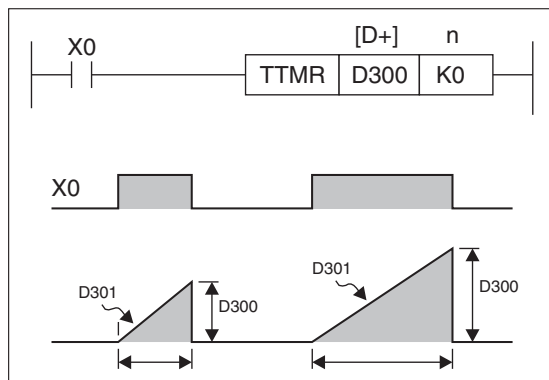


Abb. 6-114:

Programmierbeispiel zur TTMR-Anweisung

Die Betätigungsdauer von X0 wird gemessen.

D300: Betätigungszeit in Sekunden, multipliziert mit 1

D301: Betätigungszeit in Sekunden



6.8.6 Sonder-Timer (STMR)

				STMR		FNC 65				
				Sonder-Timer						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
							●			●
Operanden	S+	D+	n	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	T0 – T199	Y, M, S, D□.b*	K, H, D*, R* n = 1–32767			16 Bit	32 Bit	STMR		
						●		7		

* Nur bei FX3U und FX3UC

Funktionsweise

Generieren von Sonder-Timer-Funktionen

Beschreibung

- Mit Hilfe der STMR-Anweisung können die Timer-Funktionen
 - Ausschaltverzögerung
 - Einmaliger Impuls (wird nach dem Abschalten der Eingangsbedingung ausgegeben)
 - Blinkgeber

gesteuert werden.

- Der Timer in (S+) steuert die Bit-Operanden (D+) in folgender Art und Weise:

(D+): Ausschaltverzögerung
 ((D+)+1): Einmaliger Impuls
 ((D+)+2) / ((D+)+3): Blinktakt

Es werden ausgehend von (D+) vier aufeinanderfolgende Bits belegt.

- Der Sollwert wird dem Timer in (S+) durch (n) in der Einheit 0,1 s vorgegeben.

HINWEIS

Die in dieser Anweisung eingesetzten Timer dürfen nicht an anderer Stelle nochmals verwendet werden.

Beispiel ▾ Einsatz der STMR-Anweisung (1)

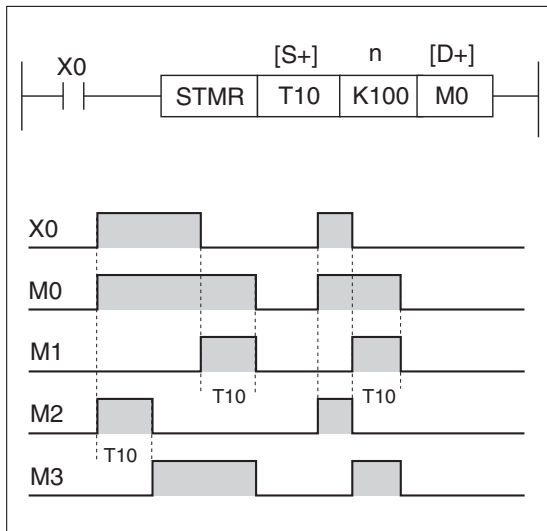


Abb. 6-116: Programmbeispiel zur Generierung einer Ausschaltverzögerung (M0) und eines Impulses (M1) mit einer Zeit von 10 s.



Beispiel ▾ Einsatz der STMR-Anweisung (2)

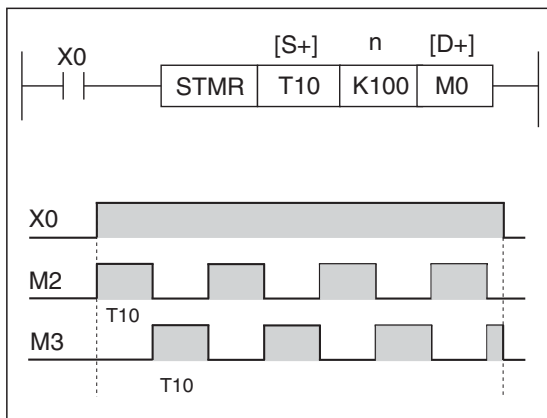


Abb. 6-115: Wenn die Eingangsbedingung fortlaufend erfüllt ist, können M2 und M3 als Blinktakt verwendet werden.



6.8.7 Flip-Flop-Funktion (ALT)

		ALT		FNC 66				
		Flip-Flop-Funktion						
Operanden	D+	CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
			●	●	●	●	●	●
Operanden	Y, M, S, D□.b*	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
		●		16 Bit	32 Bit	ALT/ALTP	3	

* Nur bei FX3U und FX3UC

Funktionsweise

Realisierung der Flip-Flop-Funktion

Beschreibung

- Mit der ALT-Anweisung wird der Zustand eines Bit-Operanden abwechselnd geändert.
- Die ALT-Anweisung wird durch Ansteuern mit einem „1“-Signal aktiviert, und der in (D+) angegebene Operand wird gesetzt.
- Bei der nächsten Ausführung der ALT-Anweisung wird der Operand zurückgesetzt.

HINWEIS

Die Anweisung wird in jedem Programmzyklus ausgeführt. Dies können Sie durch Einsatz einer vorgeschalteten Impulsfunktion (PLS-Anweisung) oder des Parameters „P“ verhindern.

Beispiel ▾

Einsatz der ALT-Anweisung, Operandenstatus invertieren

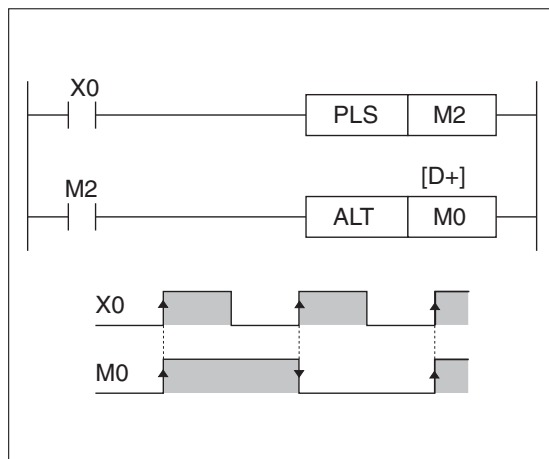


Abb. 6-117:
 Programmierbeispiel zum Einsatz der ALT-Anweisung (Operandenstatus invertieren)

Der Status des Merkers M0 wird bei jedem Einschalten des Eingangs X0 invertiert.



Beispiel ▾ Einsatz der ALTP-Anweisung, Operandenstatus invertieren

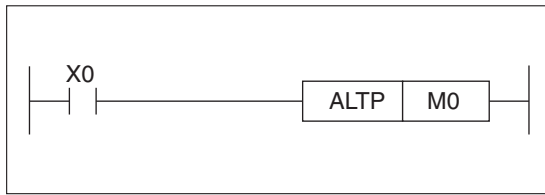


Abb. 6-119:
Bei diesem Programmierbeispiel wird zur Steuerung des Merkers M0 eine ALTP-Anweisung verwendet.

Die Funktion ist identisch mit der in Abb. 6-117.



Beispiel ▾ Einsatz der ALT-Anweisung, Start-Stopp-Funktion

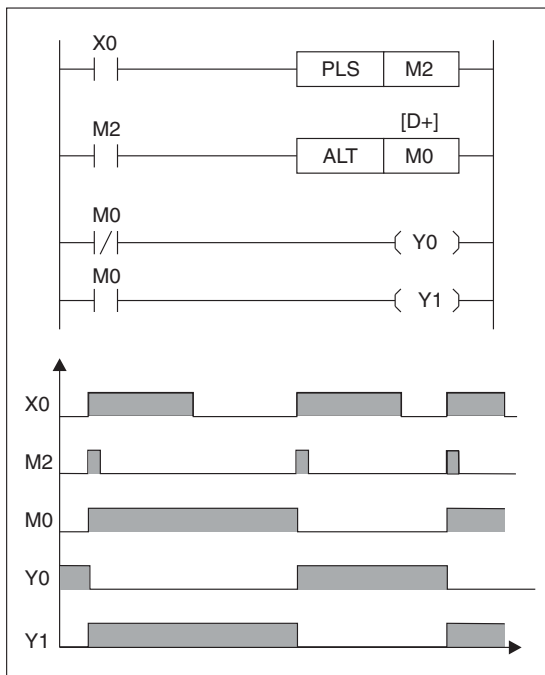


Abb. 6-118:
Programmbeispiel zum Einsatz der ALT-Anweisung (Start-Stopp-Funktion)

Der Start-Ausgang Y1 wird durch Betätigen des Tasters X0 aktiviert. Der Stopp-Ausgang Y0 wird durch nochmaliges Betätigen des Tasters X0 aktiviert.

6.8.8 Rampenfunktion (RAMP)

			RAMP		FNC 67				
			Rampenfunktion						
			CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
				●	●	●	●	●	●
Operanden	S1+, S2+, D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	D, R*	K, H, K, H, D*, R* n=1 bis +32 767		16 Bit	32 Bit	RAMP	9		
				●					

* Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

Funktionsweise

Zeitabhängige Veränderung eines Datenwertes von einem Start- zu einem Zielwert

Beschreibung

- In dem in (S1+) angegebenen Datenregister wird ein Ausgangswert festgelegt.
- In dem in (S2+) angegebenen Datenregister wird ein Zielwert festgelegt.
- In dem in (D+) angegebenen Datenregister wird ein aktueller Rampenfunktionswert abgespeichert. Als Startwert wird in (D+) der Ausgangswert gespeichert. Der Rampenfunktionswert wird n Mal verändert, bis der Zielwert erreicht ist.

(S1+): Ausgangswert

(S2+): Zielwert

(D+): Aktueller Rampenfunktionswert

n: Anzahl der Operationszyklen

- Die benötigte Ausführungszeit T beträgt: $T = (n \times \text{Programmzykluszeit})$
- Ist in (D+) der Zielwert erreicht, wird der Sondermerker M8029 gesetzt. Der Zielwert bleibt weiterhin in (D+) gespeichert.
- Die Anzahl der Operationszyklen wird nach Abarbeitung der Anweisung in dem auf (D+) folgenden Datenregister gespeichert.

HINWEISE

Die Ausführungszeit der RAMP-Anweisung ist von der Programmzykluszeit abhängig. Deshalb sollte die Steuerung mit einer konstanten Programmzykluszeit betrieben werden, um ein definiertes Verhalten der Funktion zu gewährleisten.

Bei FX2N/FX2NC- und FX3U/FX3UC-CPU's kann der Verlauf der RAMP-Funktion mit Hilfe des Sondermerkers M8026 beeinflusst werden. Ist M8026 nicht gesetzt, erfolgt eine ständige Wiederholung der Rampenfunktion. Das bedeutet, wenn der Istwert von D gleich dem Wert in S2 ist, wird die Rampenfunktion automatisch zurückgesetzt und erneut gestartet. Bei gesetztem Sondermerker M8026 wird die Rampenfunktion gehalten. Das heißt, sobald der Istwert von D dem Wert von S2 entspricht, behält die Rampenfunktion den augenblicklichen Zustand bei. M8029 bleibt in diesem Fall solange gesetzt, wie die Rampenfunktion aktiv ist. Der Wert in D wird solange nicht zurückgesetzt, bis die Anweisung reinitialisiert ist.

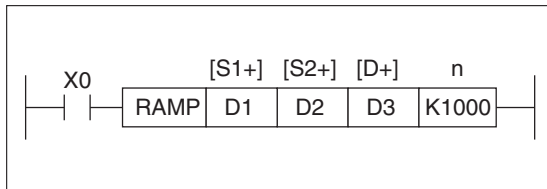
Bei FX1S-, FX1N- und FX3G/FX3GC/FX3GE/FX3S-CPU's kann der Verlauf der RAMP-Funktion nicht beeinflusst werden. Hier verhält sich die Rampenfunktion so, als wenn M8026 gesetzt wäre, d.h. die Rampenfunktion behält den augenblicklichen Zustand bei.

Wird die Rampenfunktion vor Beendigung unterbrochen, wird die Position zum Zeitpunkt der Unterbrechung solange beibehalten, bis das Auslösesignal wieder ansteht. Liegt das Rampensignal wieder an, werden die Register D + D1 zurückgesetzt und der Zyklus beginnt von neuem.

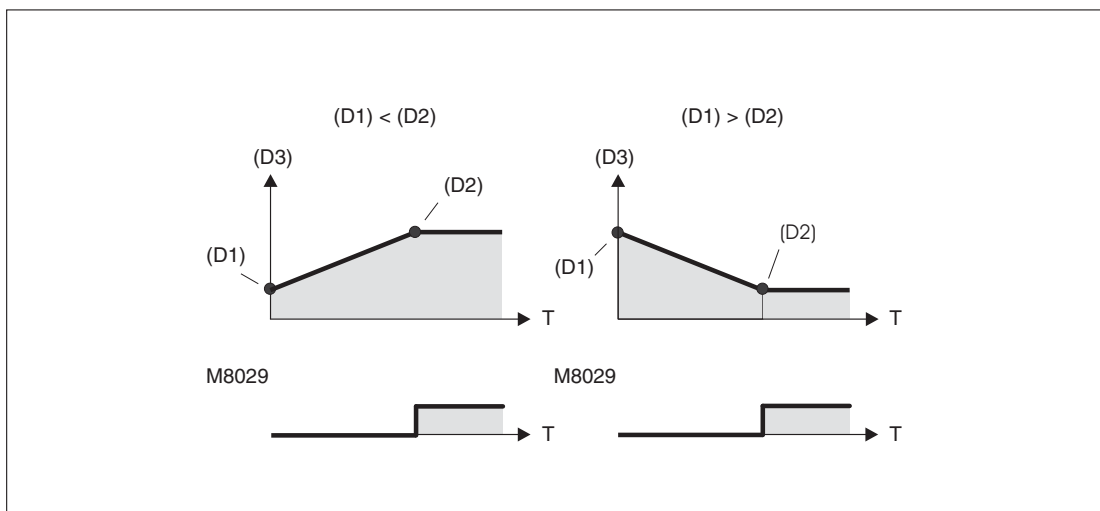
Wird für (D+) ein Latch-Operand verwendet und die SPS in den RUN-Modus geschaltet, während die Eingangsbedingung der RAMP-Anweisung aktiv ist, sollte (D+) vorher gelöscht werden.

Beispiel ▾

Einsatz der RAMP-Anweisung

**Abb. 6-121:**

Programmierbeispiel zum Einsatz der RAMP-Anweisung

**Abb. 6-120:** Zeitverläufe zum obigen Beispiel

- Nach Einschalten von X0 nimmt D3 zunächst den in D1 festgelegten Ausgangswert an. Dieser Wert wird fortlaufend (1000 Mal) verändert, bis der in D2 festgelegte Zielwert erreicht ist.
- Die für diesen Vorgang benötigte Zeit T beträgt: $T = (n \times \text{Programmzykluszeit})$.
- Die Anzahl der Operationszyklen n wird in D4 abgespeichert.
- Wird nach dem Festlegen der Programmzykluszeit (die etwas länger ist als die aktuelle Programmzykluszeit) im Datenregister D8039 der Sondermerker M8039 aktiviert, arbeitet die SPS mit einer konstanten Programmzykluszeit.
Beträgt z. B. der in D8039 festgelegte Wert 20 ms, beansprucht die Änderung im Datenregister D3 vom Ausgangswert bis zum Erreichen des Zielwertes $T = 1000 \times 20 \text{ ms} = 20 \text{ s}$.
- Ist X0 ausgeschaltet, wird die Ausführung der Rampenfunktion abgebrochen. Wird danach X0 wieder eingeschaltet, beginnt die Ausführung der Rampenfunktion erneut mit dem Ausgangswert.
- Ist die Ausführung der Rampenfunktion beendet, wird der Sondermerker (Flag) M8029 gesetzt, und D3 nimmt den in D1 festgelegten Ausgangswert an.
- Stellen Sie sicher, dass D4 gelöscht wird, wenn die SPS nach einem Stillstand wieder in den RUN-Modus geschaltet wird und X0 noch gesetzt ist.

△

6.8.9 Rundtisch-Positionierung (ROTC)

				ROTC		FNC 68				
				Rundtisch-Positionierung						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
							●			●
Operanden	S+	m1 / m2	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	D ^① , R ^{①②}	K ^③ , H ^③	Y ^④ , M ^④ , S ^④ , D□.b ^{④②}		●	16 Bit	32 Bit	ROTC	9	
					●					

- ① Es werden drei aufeinanderfolgende Datenregister belegt ((S+1) ≤ m1).
- ② Nur bei FX3U und FX3UC
- ③ m1= 2 bis 32 767; m2= 0 bis 32 767 (m1 ≥ m2)
- ④ Es werden acht aufeinanderfolgende Bits belegt.

Funktionsweise

Steuern eines Rundtisches

Beschreibung

- Die Position des Rundtisches wird über einen 2-Phasen-Encoder erfasst.
- Alle Zielpositionen auf dem Tisch werden relativ zu einer Nullposition angegeben.
- Die Zielposition wird immer auf dem kürzesten Weg angefahren.

HINWEISE

Folgende Operanden werden durch die Anweisung geschaltet bzw. abgefragt:

- ((D+)+0) A-Phasensignal des Zählers
- ((D+)+1) B-Phasensignal des Zählers
- ((D+)+2) Nullpunkterkennung
- ((D+)+3) Ausgabe: schnelle Tischdrehung vorwärts
- ((D+)+4) Ausgabe: Schleichfahrt vorwärts
- ((D+)+5) Stopp-Ausgang
- ((D+)+6) Ausgabe: Schleichfahrt rückwärts
- ((D+)+7) Ausgabe: Schnelle Tischdrehung rückwärts
- m1 Anzahl Zählimpulse pro Tischumdrehung
- m2 Anzahl Zählimpulse für die in Schleichfahrt zurückzulegende Strecke
- ((S+)+0) Ist-Position (kann nur gelesen werden)
- ((S+)+1) Zielposition
- ((S+)+2) Teilerkennung des zu verfahrenen Teils

Wird während der Ausführung der ROTC-Anweisung der Nullpunkt erkannt (((D+)+2), werden die Inhalte der Operanden ((S+)+0) bis ((S+)+2) gelöscht.

Beispiel ▾

Erfassung der Zählimpulse

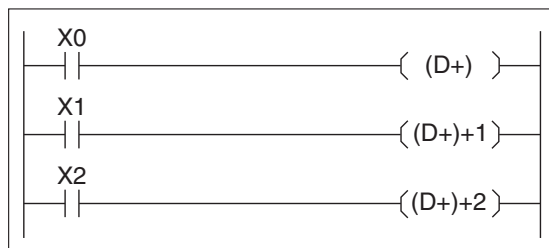


Abb. 6-122:
Programmierbeispiel für Erkennungssignale

Da der Encoder und der Schalter für den Nullpunkt über Eingänge erfasst werden, ist es notwendig, diese Eingänge in die Operanden (D+) zu schreiben. △

HINWEISE

- | Die Parameter ((S+)+1) und ((S+)+2) müssen vor dem Einschalten beschrieben werden.
- | Vor der ersten Inbetriebnahme sollte der Tisch in seine Nullposition gebracht werden.
- | Die Anweisung kann nur einmal im Programm eingesetzt werden.
- | Da die Drehbewegung des Tisches über normale Eingänge erfasst wird, dürfen nicht mehr als ca. 25 Impulse/s ausgegeben werden.

Beispiel ▽

Ein Rundtisch benötigt für eine Umdrehung 500 Impulse. Der Tisch hat 10 Stationen; daraus folgt, dass 50 Impulse zwischen zwei Stationen liegen. Die Station Null wird als „Nullposition“ aufgefasst.

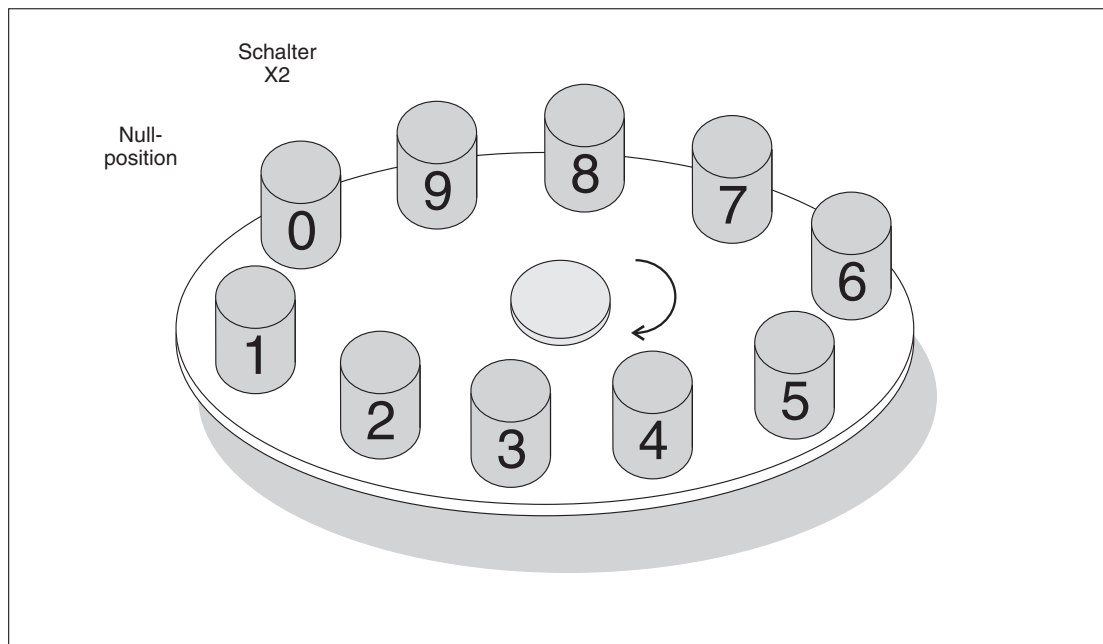


Abb. 6-123: Beispiel für eine Rundtischpositionierung

Wenn das Teil, das an Position 7 auf dem Tisch liegt, an die Position 3 gefahren werden soll, muss die ROTC-Anweisung wie folgt aussehen:

- (S+) = beliebiges Datenregister, z.B. D200
- ((S+)+1) = $3 \times 50 = 150$: Entfernung der Position in Impulsen von der Station 0
(Wert muss vor der Ausführung der Anweisung in D201 gespeichert werden.)
- ((S+)+2) = $7 \times 50 = 350$: Entfernung der Position in Impulsen von der Station 0
(Wert muss vor der Ausführung der Anweisung in D202 gespeichert werden.)
- m1 = 500

Wenn mit zwei Geschwindigkeiten gefahren werden soll, muss der Weg, der langsam zurückgelegt werden soll, in (m2) Impulsen angegeben werden.

- m2 = 0 (keine Schleichfahrt)
- (D+) = Beliebige Bits (M, Y, S, D□.b), über die der Motor des Tisches angesteuert wird

△

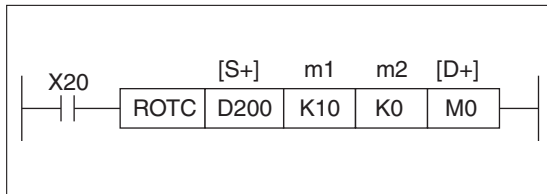
Beispiel ▾

Abb. 6-124:
Programmierbeispiel für eine ROTC-
Anweisung

Wenn X20 eingeschaltet wird, wird der Tisch um 4 Positionen nach rechts verdreht.

Der Motor wird über den Merker M3 gestartet. Wenn die Position erreicht ist, wird der Merker M5 eingeschaltet.

Über die Merker M0, M1 und M2 wird die Drehung des Tisches überwacht. △

6.8.10 Sortieranweisung (SORT)

					SORT		FNC 69				
					Sortieranweisung						
					CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
								●			●
Operanden	S+	n1 / n2	D+	m	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	D ^① , R ^{①②}	K ^③ , H ^③	D, R ^②	K, H, D, R ^②			16 Bit	32 Bit	SORT	11	
						●					

① ((n1) x (n2)) aufeinanderfolgende Datenregister

② Nur bei FX3U und FX3UC

③ n1 = 1 bis 32; n2 = 1 bis 6

Funktionsweise

Sortieren einer Matrix nach den Werten einer Spalte

Beschreibung

Bei Aufruf der SORT-Anweisung wird eine interne Datenmatrix (Tabelle), gekennzeichnet durch das Start-Datenregister (S+) mit der Größe von n1 Zeilen und n2 Spalten, nach den Werten in Spalte m sortiert und ab Datenregister (D+) neu abgelegt.

HINWEISE

Die SORT-Anweisung darf nur **einmal** im Programm verwendet werden.

Bei Ausführung der SORT-Anweisung wird jeder Eintrag entsprechend den Daten im ausgewählten Sortierfeld m in aufsteigender Folge sortiert.

(S+) und (D+) können das gleiche Datenregister angeben, da die gespeicherten Werte nicht verändert werden.

Überschneiden sich bei unterschiedlichen Datenregistern (S+) und (D+) die Datenbereiche, in denen die Matrix gespeichert ist, kann es zu einem Datenverlust kommen.

Nach Ausführung einer SORT-Anweisung wird mit M8029 eine Kennung gesetzt. Ein Sortiervorgang ist erst dann abgeschlossen, wenn die in n1 vorgegebene Anzahl erreicht ist.

Während eines Sortiervorgangs dürfen die Daten in der Sortiermatrix nicht verändert werden, da andernfalls fehlerhafte Daten abgelegt werden.

Mit einer SORT-Anweisung können Daten nur in aufsteigender Reihenfolge sortiert werden. Mit einer SORT2-Anweisung können die Daten in der Tabelle zusätzlich auch in absteigender Reihenfolge sortiert werden (Abschnitt 6.8.10).

Beispiel ▾ Anwendung der SORT-Anweisung auf eine Matrix

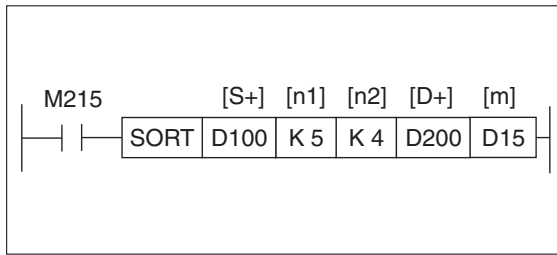


Abb. 6-125:
 Programmierbeispiel zum Einsatz der SORT-Anweisung

Die Datenmatrix könnte die folgende Form aufweisen:

Spalte Nr.	1	2	3	4
Zeile Nr.	ID-Nummer	Höhe	Gewicht	Alter
1	D100	D105	D110	D115
	1	150	45	20
2	D101	D106	D111	D116
	2	180	50	40
3	D102	D107	D112	D117
	3	160	70	30
4	D103	D108	D113	D118
	4	100	20	8
5	D104	D109	D114	D119
	5	150	50	45

Tab. 6-27:
 Unsortierte Matrix

Sortiert nach Spalte (m) = K2 ergibt sich:

Spalte Nr.	1	2	3	4
Zeile Nr.	ID-Nummer	Höhe	Gewicht	Alter
1	D200	D205	D210	D215
	4	100	20	8
2	D201	D206	D211	D216
	1	150	45	20
3	D202	D207	D212	D217
	5	150	50	45
4	D203	D208	D213	D218
	3	160	70	30
5	D204	D209	D214	D219
	2	180	50	40

Tab. 6-28:
 Sortierte Matrix nach Anwendung der SORT-Anweisung

7 Spezielle Anweisungen

7.1 Allgemeine Hinweise

Dieses Kapitel beschreibt die speziellen Anweisungen der MELSEC FX-Familie für besondere Anwendungen. Mit diesen Anweisungen lassen sich Funktionen zur Datenein- und -ausgabe, zur Modul-Kommunikation und zur Steuerung von Sondermodulen realisieren.

Eine einführende Erläuterung zur Struktur der Anweisungstabellen enthält Abschnitt 6.1.1.

7.1.1 Gesamtübersicht der speziellen Anweisungen

Einteilung	Anweisung	FNC	Bedeutung	Referenz (Abschnitt)	Steuerung					
					FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
Ein-/Ausgabe- anweisungen	TKY	70	Zehnertastatur	7.2.1			●			●
	HKY	71	Hexadezimale Tastatur	7.2.2			●			●
	DSW	72	Digitaler Schalter	7.2.3	●	●	●	●	●	●
	SEGD	73	7-Segment-Anzeige	7.2.4			●			●
	SEGL	74	7-Segment-Anzeige mit Latch	7.2.5	●	●	●	●	●	●
	ARWS	75	7-Segment-Anzeige mit zusätzlichen Tasten	7.2.6			●			●
	ASC	76	ASCII-Konvertierung	7.2.7			●			●
	PR	77	Datenausgabe über die Ausgänge	7.2.8			●			●
Datenaustausch mit Sonder- modulen	FROM	78	Auslesen von Daten aus einem Sondermodul	7.3.1		●	●	●		●
	TO	79	Schreiben von Daten in ein Sondermodul	7.3.2		●	●	●		●
Anweisungen für serielle Kommunikation	RS	80	Serielle Datenübertragung	7.4.1	●	●	●	●	●	●
	PRUN	81	Umlegen von Eingängen oder Merkern	7.4.2	●	●	●	●	●	●
	ASCI	82	Umwandlung in ein ASCII-Zeichen	7.4.3	●	●	●	●	●	●
	HEX	83	Umwandlung in einen Hexadezimalwert	7.4.4	●	●	●	●	●	●
	CCD	84	Summen- und Paritätsprüfung	7.4.5	●	●	●	●	●	●
	VRRD	85	Einlesen von Sollwerten vom FX□-8AV-BD	7.4.6	●	●	●	●	●	●
	VRSC	86	Einlesen von Schalterstellungen vom FX□-8AV-BD	7.4.7	●	●	●	●	●	●
	RS2	87	Serielle Datenübertragung	7.4.8				●	●	●
Regelungs- anweisung	PID	88	Programmierung eines Regelkreises	7.5	●	●	●	●	●	●
Index-Register sichern/wieder- herstellen	ZPUSH	102	Inhalt der Index-Register sichern	7.6.1						●
	ZPOP	103	Inhalt der Index-Register wiederherstellen	7.6.2						●

Tab. 7-1: Übersicht der Anweisungen zum Datenaustausch mit peripheren Geräten, zur seriellen Kommunikation, Regelung und zur Handhabung der Index-Register

Einteilung	Anweisung	FNC	Bedeutung	Referenz (Abschnitt)	Steuerung					
					FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
Anweisungen mit Gleitkommazahlen	DECOMP	110	Vergleich von Gleitkommazahlen	7.7.1			●	●	●	●
	DEZCP	111	Vergleich von Gleitkommazahlen mit einem Bereich	7.7.2			●			●
	DEMOV	112	Transfer von Gleitkommazahlen	7.7.3				●	●	●
	DESTR	116	Gleitkommazahl in Zeichenfolge wandeln	7.7.4						●
	DEVAL	117	Zeichenfolge in Gleitkommazahlen wandeln	7.7.5						●
	DEBCD	118	Umwandlung des Gleitkommaformats ins wissenschaftliche Zahlenformat	7.7.6			●			●
	DEBIN	119	Umwandlung des wissenschaftlichen Zahlenformats ins Gleitkommaformat	7.7.7			●			●
	DEADD	120	Addition von Gleitkommazahlen	7.7.8			●	●	●	●
	DESUB	121	Subtraktion von Gleitkommazahlen	7.7.9			●	●	●	●
	DEMUL	122	Multiplikation von Gleitkommazahlen	7.7.10			●	●	●	●
	DEDIV	123	Division von Gleitkommazahlen	7.7.11			●	●	●	●
	DEXP	124	Gleitkommazahl als Exponent zur Basis e	7.7.12						●
	DLOGE	125	Logarithmus-naturalis-Berechnung	7.7.13						●
	DLOG10	126	Berechnung des dekadischen Logarithmus	7.7.14						●
	DESQR	127	Quadratwurzeln aus Gleitkommazahlen	7.7.15			●	●	●	●
	DENEG	128	Vorzeichenumkehr von Gleitkommazahlen	7.7.16						●
	INT	129	Umwandlung von Gleitkomma- in Dezimalformat	7.7.17			●	●	●	●
	DSIN	130	Sinusberechnung mit Gleitkommazahlen	7.7.18			●			●
	DCOS	131	Cosinusberechnung mit Gleitkommazahlen	7.7.19			●			●
	DTAN	132	Tangensberechnung mit Gleitkommazahlen	7.7.23			●			●
	DASIN	133	Arcussinusberechnung	7.7.21						●
	DACOS	134	Arcuscosinusberechnung	7.7.22						●
	DATAN	135	Arcustangensberechnung	7.7.23						●
	DRAD	136	Umrechnung von Grad in Radiant	7.7.24						●
	DDEG	137	Umrechnung von Radiant in Grad	7.7.25						●
Datenverarbeitungsanweisungen	WSUM	140	Summe der Inhalte von Wortoperanden bilden	7.8.1						●
	WTOB	141	Daten in Wortoperanden in Bytes aufteilen	7.8.2						●
	BTOW	142	Wortoperanden aus einzelnen Bytes bilden	7.8.3						●
	UNI	143	Gruppen von 4 Bits zu Wortoperanden zusammenfassen	7.8.4						●
	DIS	144	Wortoperanden in Gruppen von 4 Bit teilen	7.8.5						●
	SWAP	147	High-Low-Byte-Tausch	7.8.6			●			●
	SORT2	149	Daten in Tabelle sortieren	7.8.7						●
Positionieranweisungen	DSZR	150	Referenzpunktfahrt (mit Annäherungsschalter)	7.9.6				●	●	●
	DVIT	151	Positionierung durch Interrupt	7.9.7						●
	TBL	152	Positionierung nach Datentabelle	7.9.8				●		●
	ABS	155	Lesen der absoluten Ist-Position	7.9.9	●	●		●	●	●
	ZRN	156	Referenzpunkt anfahren	7.9.10	●	●		●	●	●
	PLSV	157	Ausgabe von Impulsen variabler Frequenz	7.9.11	●	●		●	●	●
	DRVI	158	Positionieren auf einen Inkrementalwert	7.9.12	●	●		●	●	●
DRVA	159	Positionieren auf einen Absolutwert	7.9.13	●	●		●	●	●	

Tab. 7-2: Übersicht der speziellen Anweisungen zur Verarbeitung von Gleitkommazahlen, Daten und zur Positionierung

Einteilung	Anweisung	FNC	Bedeutung	Referenz (Abschnitt)	Steuerung					
					FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
Operationen mit der integrierten Uhr der SPS	TCMP	160	Vergleich von Uhr-Daten	7.10.1	●	●	●	●	●	●
	TZCP	161	Vergleich von Uhr-Daten mit einem Bereich	7.10.2	●	●	●	●	●	●
	TADD	162	Addition von Uhr-Daten	7.10.3	●	●	●	●	●	●
	TSUB	163	Subtraktion von Uhr-Daten	7.10.4	●	●	●	●	●	●
	HTOS	164	Zeitangabe in der Form „Stunden, Minuten, Sekunden“ in Sekunden wandeln	7.10.5						●
	STOH	165	Zeitangabe in Sekunden in das Format „Stunden, Minuten, Sekunden“ wandeln	7.10.6						●
	TRD	166	Lesen von Datum und Uhrzeit	7.10.7	●	●	●	●	●	●
	TWR	167	Schreiben von Uhr-Daten (Stellen der Uhr)	7.10.8	●	●	●	●	●	●
	HOUR	169	Betriebsstundenzähler	7.10.9	●	●	●	●	●	●
Gray-Code-Anweisungen	GRY	170	Umwandlung Integer in Gray-Code	7.11.1			●	●	●	●
	GBIN	171	Umwandlung Gray-Code in Integer	7.11.2			●	●	●	●
Datenaustausch mit Analogmodulen	RD3A	176	Analogwert lesen	7.12.1		●	●	●	●	●
	WR3A	177	Daten zum FX0N-3A übertragen	7.12.2		●	●	●	●	●
Anweisung aus externen ROM ausführen	EXTR	180	Es wird eine Anweisung ausgeführt, die in einer Speicherkassette abgelegt ist.	7.13.1			●			
Verschiedene Anweisungen	COMRD	182	Operandenkommentar lesen	7.14.1						●
	RND	184	Taktgenerator mit einstellbarem Tastverhältnis	7.14.2						●
	DUTY	186	Impuls mit definierter Länge ausgeben	7.14.3						●
	CRC	188	Daten prüfen (CRC-Prüfung)	7.14.4						●
	DHCMOV	189	Istwert eines High-Speed-Counters transferieren	7.14.5						●
Anweisungen für Daten, die in aufeinanderfolgenden Operanden (Datenblöcke) gespeichert sind	BK+	192	Daten in zwei Datenblöcken addieren	7.15.1						●
	BK-	193	Daten in zwei Datenblöcken subtrahieren	7.15.2						●
	BKCMP=	194	Vergleich auf „gleich“	7.15.3						●
	BKCMP>	195	Vergleich auf „größer“							●
	BKCMP<	196	Vergleich auf „kleiner“							●
	BKCMP<>	197	Vergleich auf „ungleich“							●
	BKCMP<=	198	Vergleich auf „kleiner/gleich“							●
	BKCPA>=	199	Vergleich auf „größer/gleich“							●
Verarbeitungsanweisungen für Zeichenfolgen	STR	200	Binärdaten in Zeichenfolgen wandeln	7.16.1						●
	VAL	201	Zeichenfolgen in Binärdaten wandeln	7.16.2						●
	\$+	202	Zeichenfolgen zusammenfügen	7.16.3						●
	LEN	203	Länge von Zeichenfolgen ermitteln	7.16.4						●
	RIGHT	204	Auszug der Zeichenfolgedaten von rechts	7.16.5						●
	LEFT	205	Auszug der Zeichenfolgedaten von links	7.16.6						●
	MIDR	206	Zeichen aus Zeichenfolge kopieren	7.16.7						●
	MIDW	207	Zeichenfolge ersetzen	7.16.8						●
	INSTR	208	Zeichenfolge suchen	7.16.9						●
	\$MOV	209	Zeichenfolge transferieren	7.16.10						●
Verarbeitungsanweisungen für Datenlisten	FDEL	210	Daten aus Datenliste löschen	7.17.1						●
	FINS	211	Daten in Datenliste einfügen	7.17.2						●
	POP	212	Daten lesen, die zuletzt in eine Datenliste eingetragen wurden	7.17.3						●
	SFR	213	16-Bit-Datenwort nach rechts verschieben	7.17.4						●
	SFL	214	16-Bit-Datenwort nach links verschieben	7.17.5						●

Tab. 7-3: Übersicht der speziellen Anweisungen

Einteilung	Anweisung	FNC	Bedeutung	Referenz (Abschnitt)	Steuerung					
					FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
LADE-verknüpfte Vergleiche	LD=	224	Vergleichsanweisung, gleich	7.18.1	●	●	●	●	●	●
	LD>	225	Vergleichsanweisung, größer							
	LD<	226	Vergleichsanweisung, kleiner							
	LD<>	228	Vergleichsanweisung, ungleich							
	LD≤	229	Vergleichsanweisung, kleiner gleich							
	LD≥	230	Vergleichsanweisung, größer gleich							
UND-verknüpfte Vergleiche	AND=	232	UND-verknüpfter Vergleich, gleich	7.18.2	●	●	●	●	●	●
	AND>	233	UND-verknüpfter Vergleich, größer							
	AND<	234	UND-verknüpfter Vergleich, kleiner							
	AND<>	236	UND-verknüpfter Vergleich, ungleich							
	AND≤	237	UND-verknüpfter Vergleich, kleiner gleich							
	AND≥	238	UND-verknüpfte Vergleichsanweisung, größer gleich							
ODER-verknüpfte Vergleiche	OR=	240	ODER-verknüpfter Vergleich, gleich	7.18.3	●	●	●	●	●	●
	OR>	241	ODER-verknüpfter Vergleich, größer							
	OR<	242	ODER-verknüpfter Vergleich, kleiner							
	OR<>	244	ODER-verknüpfter Vergleich, ungleich							
	OR≤	245	ODER-verknüpfter Vergleich, kleiner gleich							
	OR≥	246	ODER-verknüpfter Vergleich, größer gleich							
Datenkontrollanweisungen	LIMIT	256	Ausgabebereich von Werten begrenzen	7.19.1				●		●
	BAND	257	Eingangs-Offset festlegen	7.19.2				●		●
	ZONE	258	Ausgangs-Offset festlegen	7.19.3				●		●
	SCL	259	Werte skalieren	7.19.4				●		●
	DABIN	260	Zahl in ASCII-Code in Binärwert wandeln	7.19.5				●		●
	BINDA	261	Binärwert in ASCII-Code wandeln	7.19.6				●		●
	SCL2	269	Werte skalieren (Die Wertetabelle ist anders strukturiert als bei der SCL-Anweisung.)	7.19.7				●		●
Anweisungen zur Kommunikation mit Frequenzumrichtern	IVCK	270	Status des Frequenzumrichters prüfen	7.20.1				●	●	●
	IVDR	271	Frequenzumrichter steuern	7.20.2				●	●	●
	IVRD	272	Parameter des Frequenzumrichters lesen	7.20.3				●	●	●
	IVWR	273	Parameter in Frequenzumrichter schreiben	7.20.4				●	●	●
	IVBWR	274	Parameter blockweise in Frequenzumrichter schreiben	7.20.5						●
	IVMC	275	Kommando/Sollfrequenz in Frequenzumrichter schreiben und Status/Istfrequenz (Drehzahl) aus Frequenzumrichter lesen	7.20.6				●	●	●
MODBUS-Kommunikation	ADPRW	276	Datenaustausch des MODBUS-Masters mit Slave-Stationen (lesen und schreiben)	7.21.1				●	●	●
Datenaustausch mit Sondermodulen	RBFM	278	Aus Pufferspeicher von Sondermodulen lesen	7.22.1						●
	WBFM	279	In Pufferspeicher von Sondermodulen schreiben	7.22.2						●
Anweisung für High-Speed-Counter	DHSCT	280	Istwert eines High-Speed-Counters mit Daten in Datenlisten vergleichen	7.23						●

Tab. 7-4: Übersicht der speziellen Anweisungen

Einteilung	Anweisung	FNC	Bedeutung	Referenz (Abschnitt)	Steuerung					
					FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
Anweisungen für erweiterte File-Register	LOADR	290	Daten aus erweiterte File-Register lesen	7.24.1				●		●
	SAVER	291	Daten in erweiterte File-Register schreiben	7.24.2						●
	INTR	292	Erweiterte Register und erweiterte File-Register initialisieren	7.24.3						●
	LOGR	293	Werte von Operanden in erweiterte Register oder erweiterte File-Register speichern	7.24.4						●
	RWER	294	Daten aus erweiterte Register in erweiterte File-Register übertragen	7.24.5				●		●
	ITER	295	Erweiterte File-Register initialisieren	7.24.6						●
Anweisungen für eine CF-Speicherkarte in einem Adaptermodul FX3U-CF-ADP	FLCRT	300	Datei erzeugen/prüfen	7.25.1						●
	FLDEL	301	Datei löschen/CF-Speicherkarte formatieren	7.25.2						●
	FLWR	302	Daten in CF-Speicherkarte schreiben	7.25.3						●
	FLRD	303	Daten aus CF-Speicherkarte lesen	7.25.4						●
	FLCMD	304	Anweisung für FX3U-CF-ADP	7.25.5						●
	FLSTRD	305	Status des FX3U-CF-ADP lesen	7.25.6						●

Tab. 7-6: Übersicht der speziellen Anweisungen

7.2 Ein-/Ausgabeanweisungen

Die Anweisungen FNC 70 bis FNC 77 dienen zum Datenaustausch mit externen Geräten, die an den Ein- oder Ausgängen der SPS-Steuerung angeschlossen sind.

Übersicht der Anweisungen FNC 70 bis 79

Symbol	FNC	Bedeutung	Abschnitt
TKY	70	Zehnertastatur	7.2.1
HKY	71	Hexadezimale Tastatur	7.2.2
DSW	72	Digitaler Schalter	7.2.3
SEGD	73	7-Segment-Anzeige	7.2.4
SEGL	74	7-Segment-Anzeige mit Latch	7.2.5
ARWS	75	7-Segment-Anzeige mit zusätzlichen Tasten	7.2.6
ASC	76	ASCII-Konvertierung	7.2.7
PR	77	Datenausgabe über die Ausgänge	7.2.8

Tab. 7-5: Übersicht der Ein-Ausgabeanweisungen

7.2.1 Zehnertastatur (TKY)

				TKY		FNC 70				
				Zehnertastatur						
	CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC			
				●				●		
Operanden	S1+	D1+	D2+	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	X ^① , Y ^① , M ^① , S ^① , D□.b ^{①②}	KnY, KnM, KnS, T, C, D, R ^② , V, Z, U□.G□ ^②	Y ^③ , M ^③ , S ^③ , D□.b ^{②③}			16 Bit	32 Bit	TKY	7	
						●	●	DTKY	13	

- ① 10 aufeinanderfolgende Bits
- ② Nur bei FX3U und FX3UC
- ③ 11 aufeinanderfolgende Bits

Funktionsweise

Einlesen einer Zehnertastatur über die Eingänge

Beschreibung

- Eine Tastatur mit 10 Tasten wird über die Bits (S1+) bis ((S1+)+9) in die SPS eingelesen.
- Die eingegebenen Werte werden nacheinander im Datenwort (D1+) abgelegt. Bei einer 16-Bit-Operation können 4 Stellen (max. 9.999) und bei einer 32-Bit-Operation 8 Stellen (max. 99.999.999) geschrieben werden.
- Wenn mehr als die möglichen 4 bzw. 8 Stellen eingegeben werden, werden nur die zuletzt eingegebenen 4 bzw. 8 Stellen in (D1+) gespeichert.
- Die Bits (D2+) bis ((D2+)+10) spiegeln den Status der Tasten wieder.

HINWEISE

- | Die TKY-Anweisung darf nur einmal im SPS-Programm eingesetzt werden.
- | Wenn die Anweisung TKY nicht mehr aktiv ist, werden die Bits (D2+) gelöscht. Der Inhalt von (D1+) bleibt erhalten.

Beispiel ▾

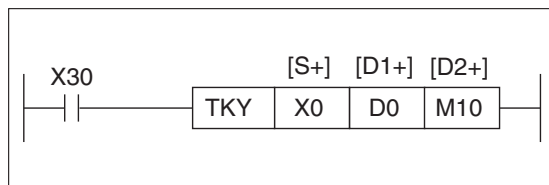


Abb. 7-1:
Programmierbeispiel zur TKY-Anweisung

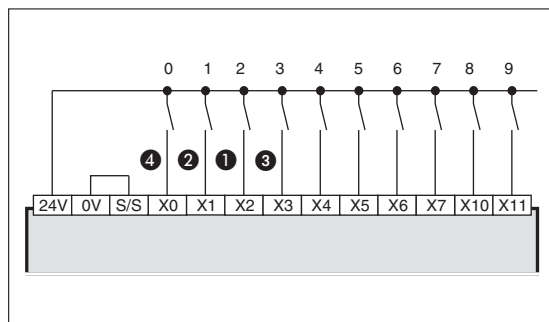


Abb. 7-2:
Zuordnung der Tasten

Im Beispiel werden den numerischen Tasten 0 bis 9 Eingänge X zugeordnet. In (S+) wird die Startadresse X0 angegeben.

Beispiel ▾

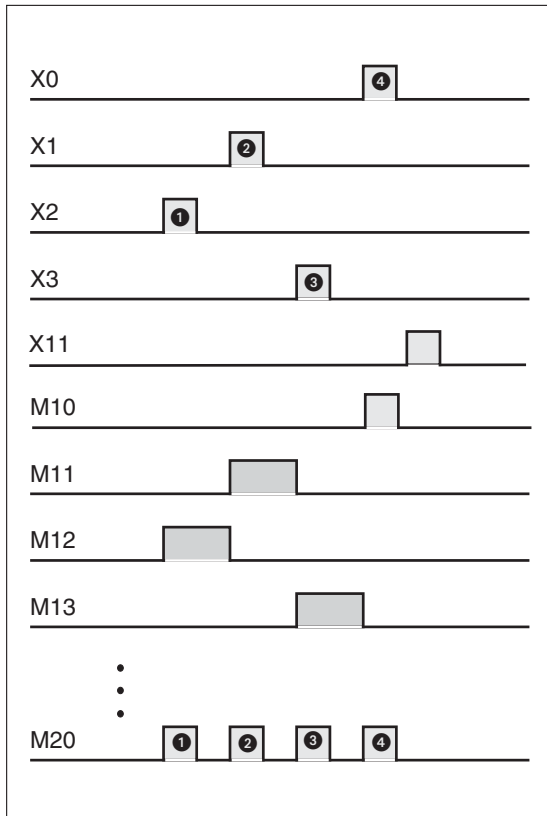


Abb. 7-3:
 Programmierbeispiel zum Ein-/Ausschalten
 der Eingänge und Merker

Wenn die Tasten X0 bis X3 in der Reihenfolge ❶ bis ❹ betätigt werden, steht im Datenregister D0 der Wert 2130. Wenn danach die Taste X11 betätigt wird, fällt die 2 aus dem Register heraus, und der neue Inhalt von D0 beträgt 1309. △

7.2.2 Hexadezimale Tastatur (HKY)

					HKY		FNC 71			
					Hexadezimale Tastatur					
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC				
			●						●	
Operanden	S+	D1+	D2+	D3+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	X ^①	Y ^①	T, C, D, R ^② , V, Z, U□G□ ^②	Y ^③ , M ^③ , S ^③ , D□.b ^{②③}		16 Bit	32 Bit	HKY	9	
						●	●	DHKY	17	

- ① Vier aufeinanderfolgende Ein- bzw. Ausgänge
- ② Nur bei FX3U und FX3UC
- ③ Acht aufeinanderfolgende Bits

Funktionsweise

Einlesen einer hexadezimalen Tastatur (Ziffern 0 bis 9 und Buchstaben A bis F) über die Eingänge

Beschreibung

- Die Tasten werden in einem Multiplex-Verfahren eingelesen. Es werden jeweils 4 Ein- und Ausgänge belegt.
- Die Tasten 0 bis 9 werden als Zahl aufgefasst und in (D2+) eingetragen.
- Die Tasten A bis F schalten die Bits (D3+) bis ((D3+)+5).
- (D3+) gibt den ersten von 8 Merkern zur Speicherung der Funktionstastenbetätigung und der Kontrollsignale an. Die Tasten A bis F schalten die Merker (D3+) bis ((D3+)+5). Merker ((D3+)+6) wird bei Betätigung einer der Tasten A bis F belegt und Merker ((D3+)+7) bei Betätigung einer der Tasten 0 bis 9. Nach jeder Registrierung einer Tastenbetätigung wird Merker M8029 gesetzt.
- Die durch die Tasten 0 bis 9 eingegebene Zahl wird in (D2+) abgelegt. Es können maximal 4 Stellen eingegeben werden (max. 9.999).
Wenn die 32-Bit-Operation ausgeführt wird, können 8 Stellen (max. 99.999.999) eingegeben werden.
- Werden mehr als eine Taste betätigt, wird die zuerst gedrückte Taste berücksichtigt.
- Wenn mehr als 4 bzw. 8 Stellen eingegeben werden, werden nur die zuletzt eingegebenen 4 bzw. 8 Stellen berücksichtigt.

HINWEISE

Die HKY-Anweisung darf nur einmal im Programm eingesetzt werden.

Bei der Verwendung der HKY-Anweisung sollte die Steuerung mit konstanter Zykluszeit betrieben werden.

Beispiel ▾

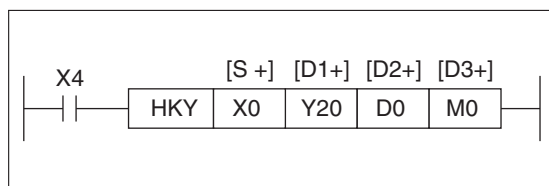


Abb. 7-4:
Programmierbeispiel zur HKY-Anweisung



Beispiel ▾

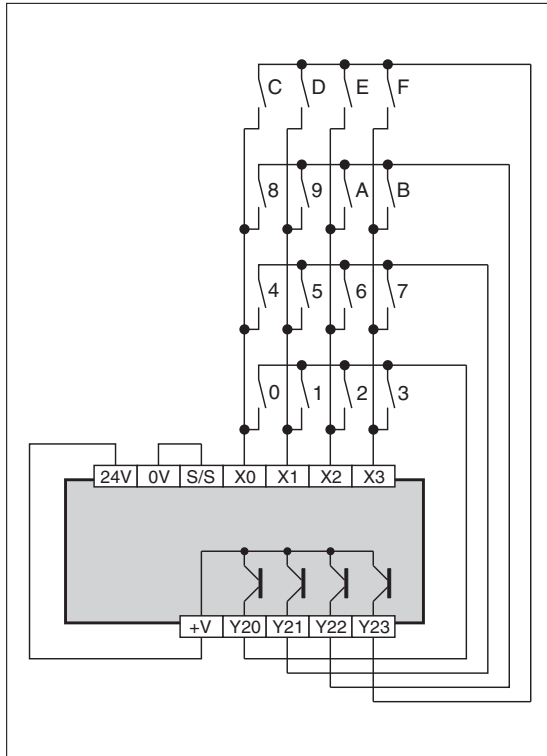


Abb. 7-5:
 Programmierbeispiel zum Ein-/Ausschalten
 der Eingänge und Merker

Funktionstasten

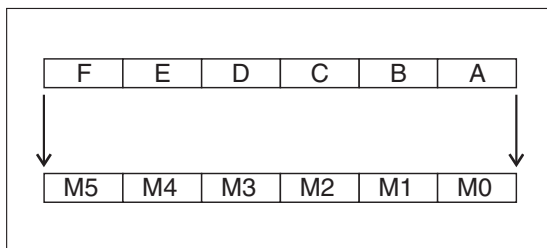


Abb. 7-6:
 Beispiel einer Merker-Zuordnung

Wird die Taste A betätigt, wird der Merker M0 gesetzt. M0 bleibt solange eingeschaltet, bis eine andere Taste betätigt wird. Wird anschließend die Taste D betätigt, wird M0 ausgeschaltet und M3 eingeschaltet.

Werden zwei oder mehr Tasten betätigt, wird die zuerst gedrückte Taste berücksichtigt.

Ausgänge

Solange eine der Tasten A bis F gedrückt gehalten wird, wird M6 aktiviert.

Solange eine der Tasten 1 bis 9 gedrückt gehalten wird, wird M7 aktiviert.

Ist X4 ausgeschaltet, werden die Daten in D0 nicht verändert. Die Merker M0 bis M7 werden ausgeschaltet.

Das Erfassen der Tastenbetätigung erfordert 8 Zyklen.

Datenspeicher

Der eingegebene Wert wird vierstellig im Datenregister D0 abgelegt.



Funktionsweise mit Sondermerker M8167

Einlesen einer hexadezimalen Tastatur über die Eingänge

Beschreibung

- Das Setzen des Sondermerkers M8167 bewirkt, dass die Eingabe über die 16 Tasten (1–9, A–F) als Hexadezimal-Format interpretiert wird.
- Die Tasten werden in einem Multiplex-Verfahren eingelesen. Es werden jeweils 4 Ein- und Ausgänge belegt, wobei (S+) den ersten Eingang und (D1+) den ersten Ausgang angibt.
- Die Tasten werden als Zahl aufgefasst und in (D2+) eingetragen.
- Die durch die Tasten 0 bis 9 und A bis F eingegebene Zahl wird in (D2+) abgelegt. Es können maximal 4 Stellen eingegeben werden (max. FFFF_H).
Wenn die 32-Bit-Operation ausgeführt wird, können 8 Stellen (max. FFFFFFFF_H) eingegeben werden.
- Werden mehr als eine Taste betätigt, wird die zuerst gedrückte Taste berücksichtigt.
- Wenn mehr als 4 bzw. 8 Stellen eingegeben werden, werden nur die zuletzt eingegebenen 4 bzw. 8 Stellen berücksichtigt.

HINWEISE

| Die HKY-Anweisung darf nur einmal im Programm eingesetzt werden.

| Bei der Verwendung der HKY-Anweisung sollte die Steuerung mit einer konstanten Zykluszeit von mehr als 20 ms betrieben werden.

Ist die Zykluszeit zu kurz, sollten Sie mit einem Timer-Interrupt arbeiten.

7.2.3 Digitaler Schalter (DSW)

					DSW		FNC 72				
					Digitaler Schalter						
					CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
						●	●	●	●	●	●
Operanden	S+	D1+	D2+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	X ^①	Y ^②	T, C, D, R ^③ , V, Z, U□\G□ ^④	K, H n = 1 oder 2			16 Bit	32 Bit	DSW		9
						●					

- ① Acht aufeinanderfolgende Eingänge
- ② Vier aufeinanderfolgende Ausgänge
- ③ Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ④ Nur bei FX3U und FX3UC

Funktionsweise

Einlesen von BCD-Schaltern im Multiplex-Verfahren

Beschreibung

- Es können ein oder zwei (n) vierstellige BCD-Schalter in die Steuerung eingelesen werden.
- Die Anweisung kontrolliert 4 Ausgänge und 4 Eingänge. Wenn 2 vierstellige BCD-Schalter eingelesen werden, werden 8 Eingänge benötigt.
- (S+) bestimmt den ersten von vier aufeinanderfolgenden Eingängen.
- (D1+) bestimmt den ersten von vier aufeinanderfolgenden Ausgängen.
- (D2+) bestimmt den Wortoperanden, der den gelesenen Wert enthält.

HINWEISE

Die Anweisung darf bei einer Steuerung der FX2N- oder der FX2NC-Serie nur zweimal in einem SPS-Programm eingesetzt werden. Bei Steuerungen der FX1S-, FX1N-, FX3G-, FX3GC-, FX3GE-, FX3S-, FX3U- und FX3UC-Serie besteht keine Einschränkung in der Zahl der verwendeten DSW-Anweisungen.

Zur korrekten Ausführung der Anweisung muss eine Steuerung mit Transistorausgängen verwendet werden.

Beispiel ▾

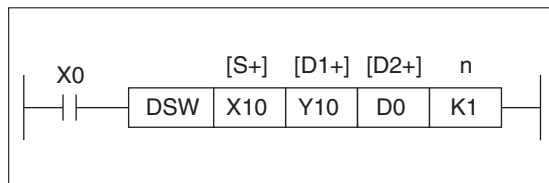


Abb. 7-7:
Programmierbeispiel zur DSW-Anweisung

Beispiel ▾

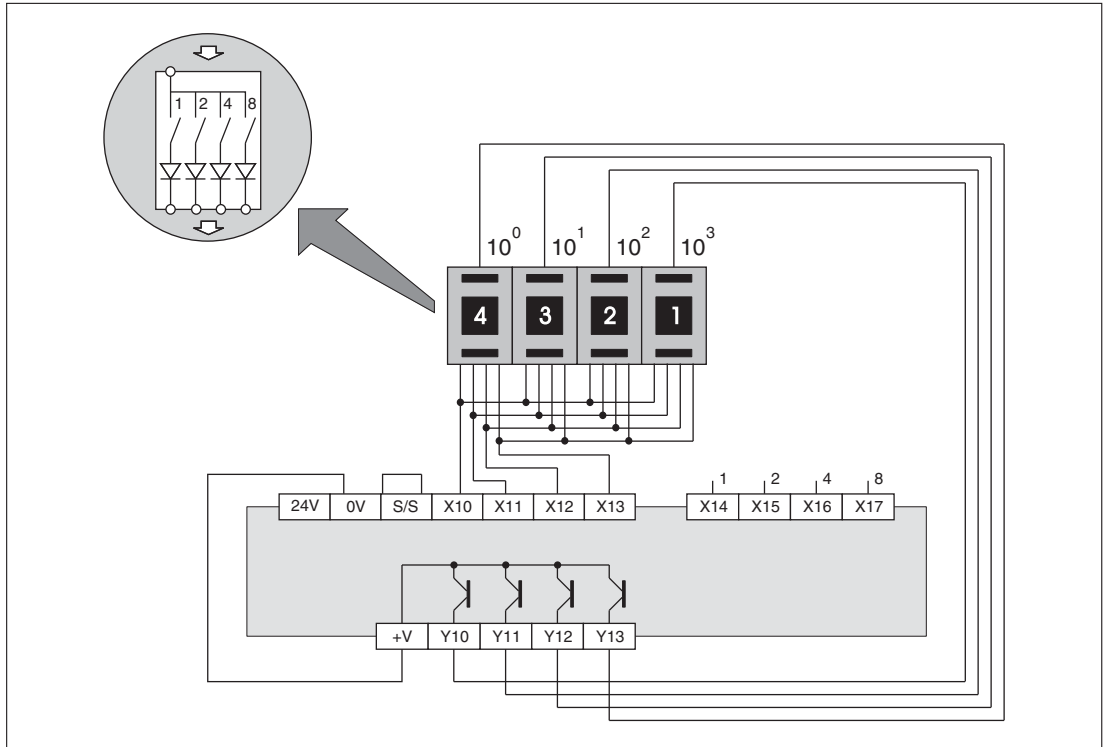


Abb. 7-8: Ein-/Ausgangsbeschaltung

Lesen des ersten Vierblocks

Die Einstellungen des vierstelligen digitalen Schalters (BCD), der mit den Eingängen X10 bis X13 verbunden ist, werden nacheinander von den Ausgängen Y10 bis Y13 gelesen und binär im Datenregister D1 gespeichert.

Die Einstellung von n ist in diesem Fall 1.

Lesen des zweiten Vierblocks

Die Einstellungen des Schalter (BCD), der mit den Eingängen X14 bis X17 verbunden ist, werden nacheinander von den Ausgängen Y10 bis Y13 gelesen und binär im Datenregister D1 gespeichert.

Die Einstellung von n ist hierbei 2.

Ist X0 eingeschaltet, arbeiten die Ausgänge Y10 bis Y13 nacheinander die Zustände der entsprechenden Eingänge X ab.

Ist ein Arbeitsvorgang abgeschlossen, wird der Sondermerker M8029 gesetzt.

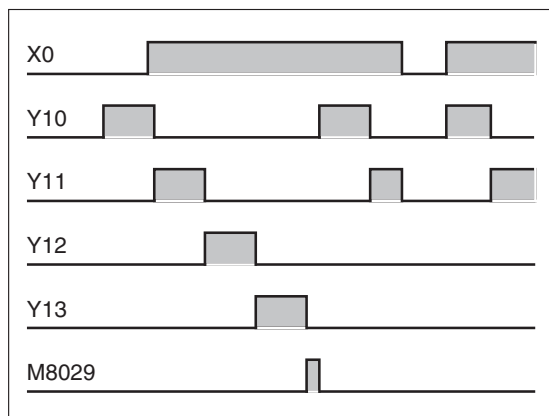


Abb. 7-9: Zeitdiagramm zum Schalten der Ausgänge

7.2.4 7-Segment-Anzeige (SEGD)

		SEGD		FNC 73			
		7-Segment-Anzeige					
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
			●			●	
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte	
	K, H, KnX, KnY, KnM, KnS, T, C, D, R*, V, Z, U□G□*	KnY, KnM, KnS, T, C, D, R*, V, Z, U□G□*	●	16 Bit	32 Bit	SEGD/ SEGDP	5

* Nur bei FX3U und FX3UC

Funktionsweise

Ausgeben einer einstelligen Hexadezimalzahl an eine 7-Segment-Anzeige

Beschreibung

- Die hexadezimale Zahl in (S+) wird in das für eine 7-Segment-Anzeige benötigte Format gewandelt und in (D+) abgelegt.
- Die Bits b0 bis b6 von (D+) entsprechen den Segmenten der 7-Segment-Anzeige:

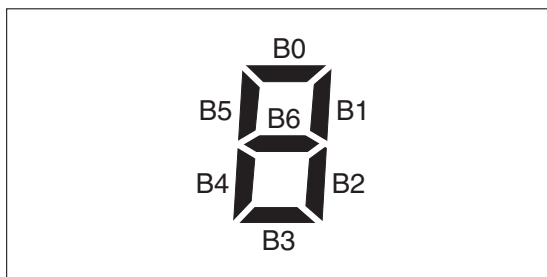


Abb. 7-10:
7-Segment-Anzeige

HINWEIS | Die Bits b15 bis b8 von (D+) werden nicht verändert.

Beispiel ▾

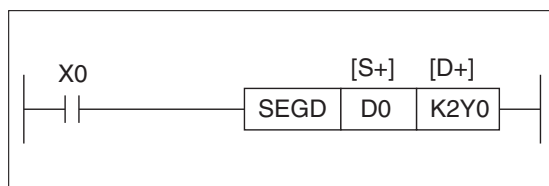


Abb. 7-11:
Programmierbeispiel zur SEGD-Anweisung

Ausgang	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Segment	B0	B1	B2	B3	B4	B5	B6	B7



7.2.5 7-Segment-Anzeige mit Latch (SEGL)

				SEGL		FNC 74																								
				7-Segment-Anzeige mit Latch																										
<table border="1"> <tr> <th>Operanden</th> <th>S+</th> <th>D+</th> <th>n</th> <th>Puls-Anweisung (P)</th> <th colspan="2">Verarbeitung</th> <th colspan="2">Programmschritte</th> </tr> <tr> <td rowspan="2">K, H, KnX, KnY, KnM, KnS, T, C, D, R^①, V, Z, U□G□^②</td> <td rowspan="2">Y^③</td> <td rowspan="2">K, H</td> <td rowspan="2"></td> <td rowspan="2"></td> <td>16 Bit</td> <td>32 Bit</td> <td rowspan="2">SEGL</td> <td rowspan="2">7</td> </tr> <tr> <td>●</td> <td></td> </tr> </table>				Operanden	S+	D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte		K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②	Y ^③	K, H			16 Bit	32 Bit	SEGL	7	●		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
				Operanden	S+	D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte																			
K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^②	Y ^③	K, H			16 Bit	32 Bit	SEGL	7																						
					●																									
					●	●	●	●	●	●																				

- ① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ② Nur bei FX3U und FX3UC
- ③ n = 0 bis 3: 8 aufeinanderfolgende Ausgänge
n = 4 bis 7: 12 aufeinanderfolgende Ausgänge

Funktionsweise

Ansteuern von einer oder zwei vierstelligen 7-Segment-Anzeigen mit speichernder Anzeige

Beschreibung

- Es können bis zu zwei vierstellige 7-Segment-Anzeigen mit dieser Anweisung angesteuert werden. Die Ansteuerung erfolgt in einem Multiplex-Verfahren. Es werden 4 Taktausgänge und für jede vierstellige Anzeige nochmals 4 Datenausgänge belegt.
- Der in (S+) enthaltene Zahlenwert (max. 9999) wird in den BCD-Code gewandelt und über die Ausgänge (D+) bis ((D+)+3) ausgegeben. Wenn zwei vierstellige Anzeigen angesteuert werden sollen, erfolgt die Ausgabe der Daten für die zweite Anzeige über die Ausgänge ((D+)+10) bis ((D+)+13). Auch bei weniger als 4 Stellen können die nicht verwendeten Operanden nicht für andere Zwecke eingesetzt werden.
- Der jeweils an den Datenausgängen anliegende BCD-Code wird über die Taktausgänge ((D+)+4) bis ((D+)+7) der jeweiligen Stelle der Anzeige automatisch zugeordnet.
- Die Einstellung von (n) ist abhängig von vier Faktoren:
 - a) Ausgangslogik der SPS-Ausgänge (plus- oder minusschaltend)
 - b) Logik der Datenleitungen der 7-Segment-Anzeige
 - c) Logik der Takteingänge der 7-Segment-Anzeige
 - d) Anzahl der verwendeten 7-Segment-Anzeigen

Plusschaltende SPS-Ausgänge

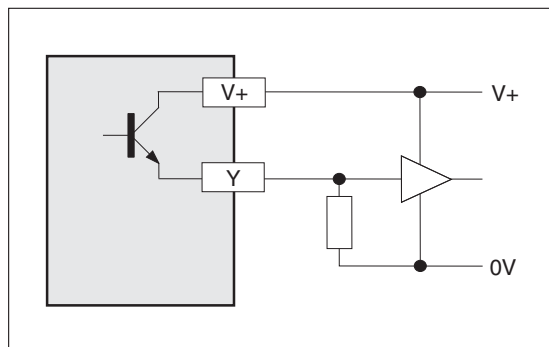


Abb. 7-12:
Bei eingeschaltetem Ausgang wird der „Plus“ der Lastspannung geschaltet.

Taktsignal-Logik: Die Daten werden gespeichert, wenn das Taktsignal HIGH ist.
Datensignal-Logik: Aktive Datenleitungen sind HIGH.

Minusschaltende SPS-Ausgänge

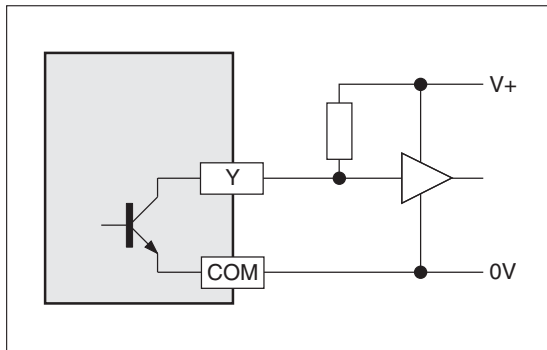


Abb. 7-13:

Bei eingeschaltetem Ausgang wird der „Minus“ der Lastspannung geschaltet.

Taktsignal-Logik: Die Daten werden gespeichert, wenn das Taktsignal LOW ist.

Datensignal-Logik: Aktive Datenleitungen sind LOW.

SPS-Logik	Taktsignal	Datenleitung	n	
			1 Anzeige	2 Anzeigen
Positiv (+)	Positiv (HIGH)	Positiv (HIGH)	0	4
Negativ (-)	Negativ (LOW)	Negativ (LOW)		
Positiv (+)	Positiv (HIGH)	Negativ (LOW)	1	5
Negativ (-)	Negativ (LOW)	Positiv (HIGH)		
Negativ (-)	Positiv (HIGH)	Negativ (LOW)	2	6
Positiv (+)	Negativ (LOW)	Positiv (HIGH)		
Negativ (-)	Positiv (HIGH)	Positiv (HIGH)	3	7
Positiv (+)	Negativ (LOW)	Negativ (LOW)		

Tab. 7-7: 7-Segment-Anzeigelogik

HINWEISE

Zur korrekten Ausführung der Anweisung muss eine Steuerung mit Transistorausgängen verwendet werden.

Es können nur 7-Segment-Anzeigen mit Datenerhaltung verwendet werden.

Die Anweisung darf bei einer Steuerung der FX2N- und FX2NC-Serie nur zweimal in einem SPS-Programm eingesetzt werden. Bei Steuerungen der FX1S-, FX1N-, FX3G-, FX3GC-, FX3GE-, FX3S-, FX3U- und FX3UC-Serie besteht keine Einschränkung in der Zahl der verwendeten SEGL-Anweisungen.

Die SEGL-Anweisung wird synchron zur Programmzykluszeit ausgeführt. Diese sollte größer als 10 ms sein. Betreiben Sie bei kürzeren Zykluszeiten die Steuerung mit einer konstanten Zykluszeit von mindestens 10 ms.

Zur Aktualisierung einer 4-stelligen Anzeige wird der 12-fache Wert der Programmzykluszeit benötigt.

Beispiel ▾

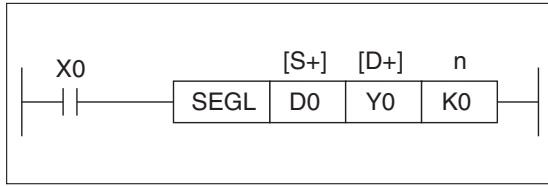


Abb. 7-14:
 Programmierbeispiel für eine
 SEGL-Anweisung

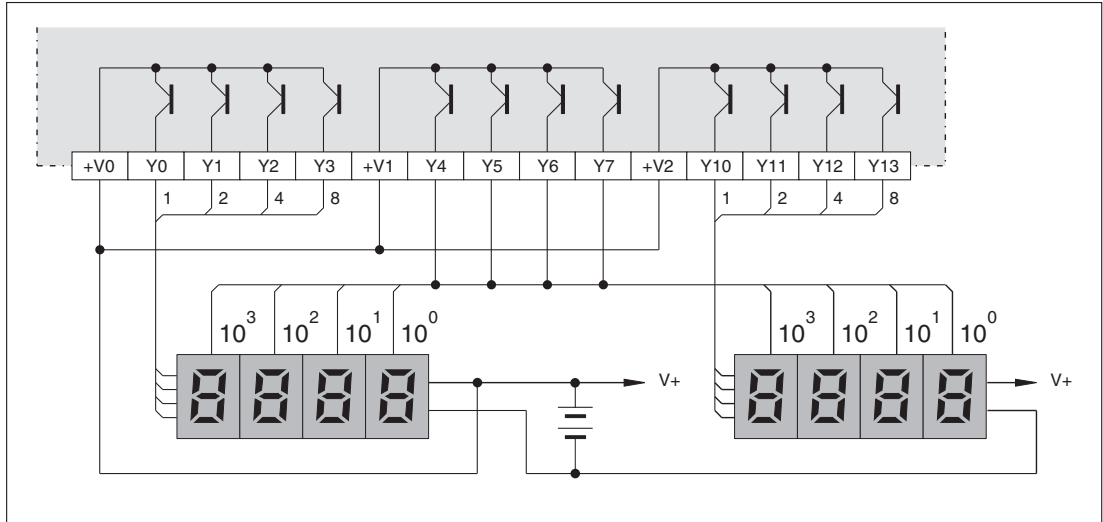


Abb. 7-15: Ausgangsbeschaltung



7.2.6 7-Segment-Anzeige mit zusätzlichen Tasten (ARWS)

					ARWS		FNC 75				
					7-Segment-Anzeige mit zusätzlichen Tasten						
					CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
								●			●
Operanden	S+	D1+	D2+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	X ^① , Y ^① , M ^① , S ^① , D□.b ^{②③}	T, C, D, R ^② , V, Z	Y ^③	K, H n= 0 bis 3		16 Bit	32 Bit	ARWS	9		
						●					

- ① Vier aufeinanderfolgende Operanden
- ② Nur bei FX3U und FX3UC
- ③ Acht aufeinanderfolgende Operanden

Funktionsweise

Auswählen und Verändern einer Stelle einer vierstelligen BCD-Zahl auf einer 7-Segment-Anzeige

Beschreibung

- Es werden vier Tasten (S+) bis ((S+)+3) abgefragt:
 - (S+) = Dekrementieren der ausgewählten Stelle
 - ((S+)+1) = Inkrementieren der ausgewählten Stelle
 - ((S+)+2) = Cursor nach rechts (Niedrigere Stelle auswählen)
 - ((S+)+3) = Cursor nach links (Höhere Stelle auswählen)
- Die in (D1+) abgelegten Daten werden über eine vierstellige 7-Segment-Anzeige angezeigt und mit den Tasten (S+) verändert.
- Die in (D1+) abgelegten Daten sind binäre Daten.
- Mit (D2+) und (n) werden die Ausgänge und die Art der Beschaltung festgelegt, an die die 7-Segment-Anzeige angeschlossen wird (siehe SEGL-Anweisung).

HINWEISE

Zur korrekten Ausführung der Anweisung muss eine Steuerung mit Transistorausgängen verwendet werden.

Die ARWS-Anweisung wird synchron zur Programmzykluszeit ausgeführt. Diese sollte größer als 10 ms sein. Betreiben Sie bei kürzeren Zykluszeiten die Steuerung mit einer konstanten Zykluszeit von mindestens 10 ms.

Die ARWS-Anweisung darf nur einmal im Programm eingesetzt werden. Falls die Funktion mehrmals im Programm benötigt wird, kann eine Indexvergabe (V, Z) verwendet werden.

Beispiel ▾

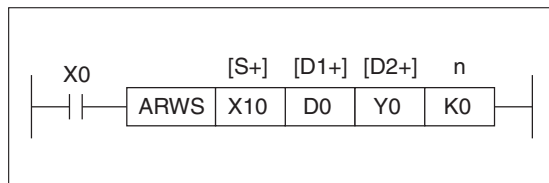


Abb. 7-16: Programmierbeispiel zur ARWS-Anweisung

Beispiel ▾

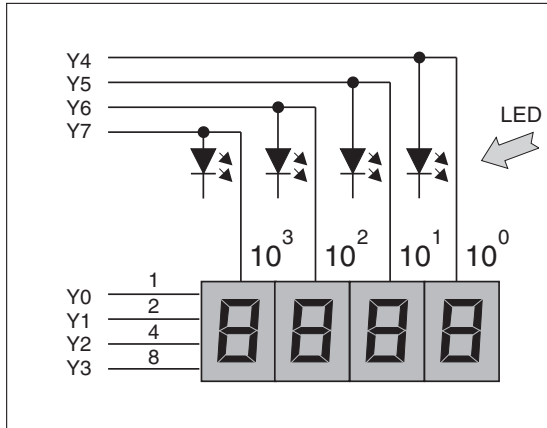


Abb. 7-17:
Beispiel für eine 7-Segment-Anzeige

In dem 16-Bit-Datenregister D0 werden vier BCD-Werte gespeichert. Jeder BCD-Wert belegt vier Bit. Es kann maximal der Wert 9 999 in D0 dargestellt werden.

Über die Tasten bzw. Eingänge X10 bis X13 können die Position und der numerische Wert der Anzeige verändert werden (siehe Abb. 7-18).

- X11: Aufwärtszählen der ausgewählten Stelle 0-1-2-3
- X10: Abwärtszählen der ausgewählten Stelle 0-9-8-7
- X13: Verschieben nach links
- X12: Verschieben nach rechts

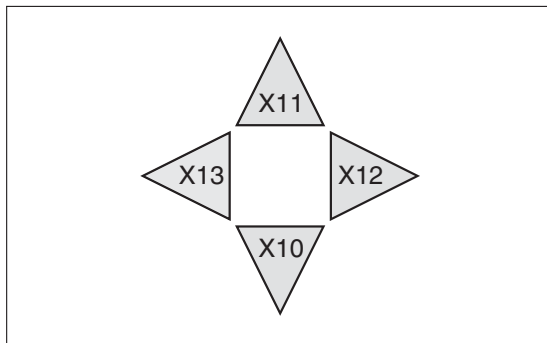


Abb. 7-18:
Beispiel für die Eingänge X10 bis X13

Über die Eingänge X12 und X13 wird die zu verändernde Anzeigenposition festgelegt.

Ist X0 eingeschaltet, wird die Position 10^3 als Anfangsposition betrachtet.

Jedes Betätigen von X12 und X13 bewirkt, dass die Anzeigenposition in einer vorgegebenen Reihenfolge gewechselt wird:

- Betätigen von X12 (Verschieben nach rechts):

$$10^3 - 10^2 - 10^1 - 10^0 - 10^3$$

- Betätigen von X13 (Verschieben nach links):

$$10^3 - 10^2 - 10^1 - 10^0 - 10^3$$

Die über X12 oder X13 festgelegte Position kann von einer zusätzlichen LED in der Leitung des Strobe-Signals angezeigt werden (Y4 bis Y7).

Beispiel ▾

Über die Tasten bzw. Eingänge X10 und X11 wird an der festgelegten Anzeigeposition der numerische Wert verändert.

Mit X10 und X11 wird die Reihenfolge der Dateneingabe festgelegt.

Die Daten im Datenregister D0 ändern sich in der folgenden Reihenfolge:

X11: Aufwärtszählen: 0 - 1 - 2 - ... 8 - 9 - 0 - 1

X10: Abwärtszählen: 0 - 9 - 8 - 7 - ... 1 - 0 - 9

Der aktuell gesetzte Wert wird über die 7-Segment-Anzeige dargestellt.

Mit der ARWS-Anweisung kann ein gewünschter Wert in das Datenregister D0 geschrieben und gleichzeitig auf der 7-Segment-Anzeige dargestellt werden.



7.2.7 ASCII-Konvertierung (ASC)

		ASC		FNC 76				
		ASCII-Konvertierung						
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
					●			●
Operanden	S+	D+	Puls-Anweisung (P)		Verarbeitung		Programmschritte	
	ASCII-Daten z.B. 0-9; A-Z; a-z	T, C, D, R*, V, Z, U□\G□*			16 Bit	32 Bit	ASC	7
					●			

* Nur bei FX3U und FX3UC

Funktionsweise

Konvertieren von alphanummerischen Daten in ASCII-Daten

Beschreibung

- Die in (S+) eingegebenen alphanummerischen Daten werden in ASCII-Zeichen konvertiert und in (D+) abgespeichert. Ab (D+) werden vier aufeinanderfolgende Operanden belegt.
- Es können maximal 8 alphanummerische Daten eingegeben werden.

Beispiel ▾

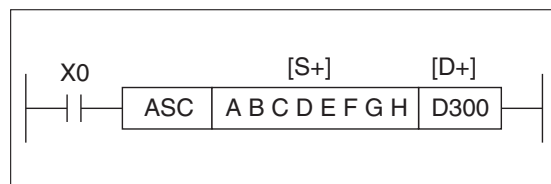


Abb. 7-19:
Programmierbeispiel einer ASC-Anweisung

	obere 8 Bit	untere 8 Bit
D300	42 (B)	41 (A)
D301	44 (D)	43 (C)
D302	46 (F)	45 (E)
D303	48 (H)	47 (G)

Abb. 7-20:
Speichern der Daten A bis H

Bits 3 bis 0		Bits 6 bis 4							
		0	1	2	3	4	5	6	7
		000	001	010	011	100	101	110	111
0	0000	NUL	DLE	SP	0	@	P	'	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	\$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(8	H	X	h	x
9	1001	HT	EM)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[k	{
C	1100	FF	FS	,	<	L	\	l	
D	1101	CR	GS	-	=	M]	m	}
E	1110	SO	RS	.	>	N	×	n	~
F	1111	SI	VS	/	?	O	-	o	DEL

Tab. 7-8: ASCII-Code

Funktion bei gesetztem Sondermerker M8161

Zur Speicherung eines Zeichens im ASCII-Code genügen 8 Bit. Wenn der Sondermerker M8161 nicht gesetzt ist, werden in einem Wortoperanden von (D+) zwei ASCII-Zeichen abgelegt. Ist der Sondermerker M8161 gesetzt, wird in jedem Wortoperanden von (D+) ein ASCII-Zeichen in den unteren 8 Bit gespeichert. Die Bits 8 bis 15 enthalten den Wert „00“.

HINWEISE

Ist der Merker M8161 „0“, entspricht die Anzahl der für (D+) benötigten Operanden der halben Anzahl der zu wandelnden ASCII-Zeichen.

Bei gesetztem Merker M8161 werden für (D+) so viele Operanden benötigt, wie Zeichen in der Zeichenfolge (S+) vorhanden sind.

Der Sondermerker M8161 beeinflusst auch das Verhalten einer RS-, ASCI-, HEX-, CCD- und CRC-Anweisung. Wenn eine dieser Anweisungen im selben Programm wie die ASC-Anweisung verwendet wird, sollte M8161 vor der Ausführung der ASC-Anweisung gesetzt und unmittelbar nach Ausführung der ASC-Anweisung wieder zurückgesetzt werden.

7.2.8 Datenausgabe über die Ausgänge (PR)

			PR		FNC 77				
			Datenausgabe über die Ausgänge						
Operanden	S+	D+	CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
						●			●
	T, C, D, R ^①	Y ^②	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
				16 Bit	32 Bit		PR	5	
				●					

- ① Nur bei FX3U und FX3UC
- ② 10 aufeinanderfolgende Ausgänge

Funktionsweise

Ausgeben von ASCII-Zeichen über die Ausgänge

Beschreibung

- Ausgabe von 8 ASCII-Zeichen in (S+) bis ((S+)+3) oder von 16 ASCII-Zeichen in (S+) bis ((S+)+7) über die Ausgänge (D+)
- Die Ausgänge (D+) bis ((D+) +7) stellen die Bits b0 bis b7 von (S+) dar.
- ((D+)+8) steht als Taktsignal zur Verfügung, ((D+)+9) zeigt den Abschluss der Bearbeitung der Anweisung an.

HINWEISE

Die PR-Anweisung darf nur zweimal im Programm verwendet werden.

Zur korrekten Ausführung der Anweisung muss eine Steuerung mit Transistorausgängen verwendet werden.

Die PR-Anweisung wird synchron zur Programmzykluszeit ausgeführt. Bei zu kurzer Zykluszeit kann die Steuerung mit einer konstanten Zykluszeit betrieben werden. Bei zu langer Zykluszeit kann die PR-Anweisung durch einen Timer-Interrupt ausgeführt werden.

Die Anzahl der ausgegebenen Zeichen kann durch den Sondermerker M8027 beeinflusst werden. Wenn M8027 nicht gesetzt ist („0“), werden 8 Zeichen (8 Bytes) ausgegeben. Ist M8027 gesetzt (auf „1“), werden 16 Zeichen in 16 Bytes ausgegeben. In diesem Fall muss die Eingangsbedingung der PR-Anweisung während der Ausführung nicht gesetzt sein. Die PR-Anweisung kann durch einen kurzen Impuls gestartet werden. Nach der Ausgabe von 16 Zeichen wird der Sondermerker M8029 für die Dauer eines Programmzyklus gesetzt, um das Ende der Ausführung anzuzeigen. Bitte beachten Sie, dass M8029 nur im 16-Byte-Modus (M8027 = „1“) gesetzt wird.

Wenn der Sondermerker M8027 gesetzt ist und in (S+) bis ((S+)+7) das Zeichen „NUL“ (00H) enthalten ist, wird dieses Zeichen nicht ausgegeben. Die PR-Anweisung wird komplett bearbeitet und anschließend der Merker 8029 gesetzt.

Beispiel ▾

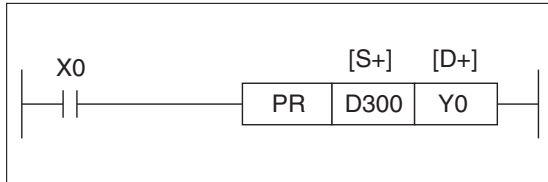


Abb. 7-21:
 Programmierbeispiel zur PR-Anweisung

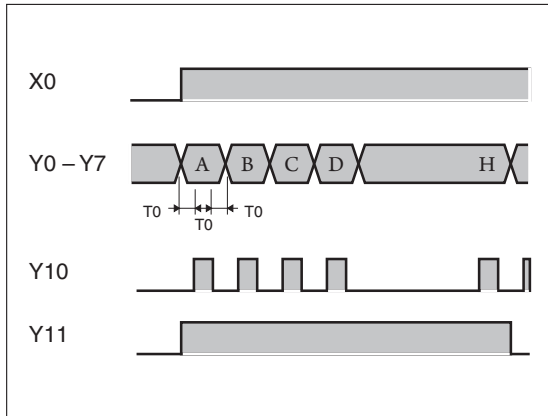


Abb. 7-22:
 Programmbeispiel zum Schalten der Ein-/Ausgänge

$T_0 = \text{Zykluszeit}$

In den Datenregistern D300 bis D303 befinden sich die ASCII-Daten der Beispielprogrammierung in Abschnitt 7.1.7. Es werden demnach die Buchstaben „A“ bis „H“ ausgegeben.

Als Ausgänge stehen Y0 (unteres Bit) bis Y7 (oberes Bit) sowie Y10 (Strobe-Signale) und Y11 (Ausführungs-Flag) zur Verfügung.

Ausgabeformat

Wird X0 während der Abarbeitung der Anweisung ausgeschaltet, wird die Datenübertragung eingestellt.

Der Vorgang beginnt erneut, wenn X0 wieder eingeschaltet wird.



7.3 Datenaustausch mit Sondermodulen

Die Anweisungen TO und FROM

Symbol	FNC	Bedeutung	Abschnitt
FROM	78	Auslesen von Daten aus einem Sondermodul	7.3.1
TO	79	Schreiben von Daten in ein Sondermodul	7.3.2

Tab. 7-9: Anweisungen für den Datenaustausch mit Sondermodulen

Mit Ausnahme der FX1S- und der FX3S-Serie kann die Anzahl der digitalen Ein- und Ausgänge aller Grundgeräte der MELSEC FX-Familie durch den Anschluss von Erweiterungsgeräten erhöht werden. Zusätzlich kann der Funktionsumfang der Steuerung durch die Installation von sogenannten Sondermodule noch weiter gesteigert werden. Sondermodule erfassen zum Beispiel analoge Werte wie Ströme oder Spannungen, regeln Temperaturen oder wickeln die Kommunikation mit externen Geräten ab.

Während für die digitalen Erweiterungen keine besonderen Anweisungen benötigt werden (Die zusätzlichen Ein- und Ausgänge werden genauso wie die Ein- und Ausgänge des Grundgeräts behandelt.), werden für den Datenaustausch zwischen dem Grundgerät und einem Sondermodul zwei Applikationsanweisungen verwendet : Die FROM- und die TO-Anweisung.

Im Sondermodul ist ein Speicherbereich eingerichtet, in dem z. B. analoge Messwerte oder empfangene Daten zwischengespeichert – gepuffert – werden. Wegen dieser Funktion wird dieser Speicherbereich als „Pufferspeicher“ bezeichnet. Auf den Pufferspeicher in einem Sondermodul kann auch das Grundgerät zugreifen und z. B. Messwerte oder empfangene Daten lesen, aber dort auch Daten eintragen, die das Sondermodul dann weiterverarbeitet (Einstellungen für die Funktion des Sondermoduls, Sendedaten etc.).

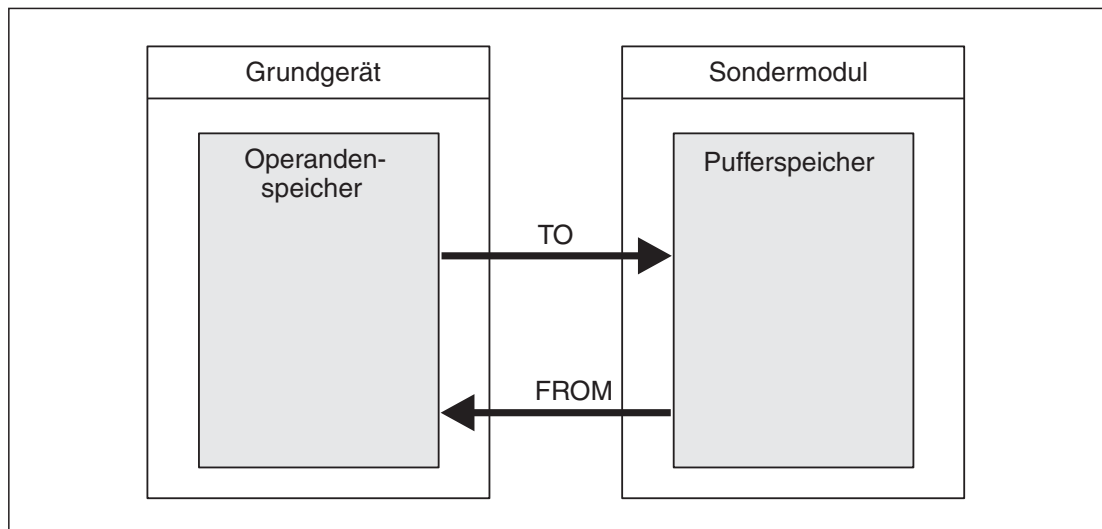


Abb. 7-23: Zwischen dem SPS-Grundgerät und einem Sondermodul können Daten mit FROM-/TO-Anweisungen ausgetauscht werden.

Ein Pufferspeicher kann aus bis zu 32767 einzelnen Speicherzellen bestehen. Jede dieser Pufferspeicheradressen kann 16 Bit an Informationen speichern. Die Funktion einer Pufferspeicheradresse hängt von der Art des Sondermoduls ab und kann den Bedienungsanleitungen der einzelnen Sondermodule entnommen werden.

Für die korrekte Funktion benötigt eine FROM- oder TO-Anweisung bestimmte Angaben:

- Aus welchem Sondermodul sollen Daten gelesen bzw. in welches Sondermodul sollen Daten übertragen werden?

- Wie lautet die erste Pufferspeicheradresse, aus der Daten gelesen oder in die Daten eingetragen werden?
- Aus wie vielen Pufferspeicheradressen sollen Daten gelesen bzw. in wie vielen Adressen sollen Daten eingetragen werden.
- Wo im Grundgerät sollen die Daten aus dem Pufferspeicher abgelegt werden bzw. wo sind die Daten gespeichert, die zum Sondermodul übertragen werden sollen.

Sondermoduladresse

Um bei mehreren Sondermodulen die Daten in das richtige Modul zu transferieren oder aus dem korrekten Modul zu lesen, ist eine besondere Kennzeichnung der Module erforderlich. Dazu erhält jedes Sondermodul automatisch eine Nummer aus dem Bereich 0 bis 7. (Maximal können 8 Sondermodule an die SPS angeschlossen werden.) Die Nummern werden fortlaufend vergeben, und die Nummerierung beginnt mit dem Modul, welches zuerst mit der SPS verbunden wird.

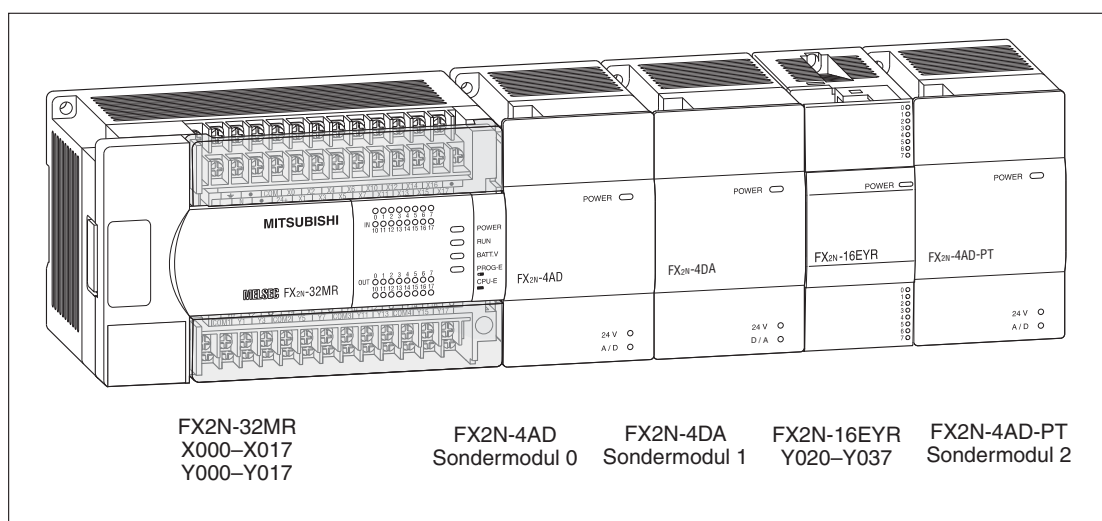


Abb. 7-24: Die mit den FROM- und TO-Anweisungen ansprechbaren Sondermodule werden rechts neben dem Grundgerät angeordnet.

Anfangsadresse im Pufferspeicher

Jede der bis zu 32767 Pufferspeicheradressen kann dezimal im Bereich von 0 bis 32766 adressiert werden (FX1N: 0 bis 31). 32-Bit-Daten werden so im Pufferspeicher abgelegt, dass die Speicherzelle mit der niedrigeren Adresse die niederwertigen 16-Bit und die folgende Pufferspeicheradresse die höherwertigen 16-Bit enthält.

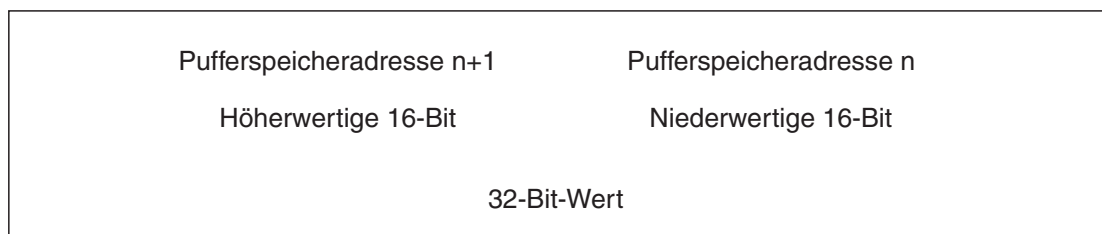


Abb. 7-25: Als Anfangsadresse für 32-Bit-Daten muss immer die Adresse angegeben werden, welche die niederwertigen 16-Bit enthält.

Anzahl der zu übertragenden Daten

Die Anzahl der Daten bezieht sich auf die zu übertragenden Dateneinheiten. Wird eine FROM- oder TO-Anweisung als 16-Bit-Anweisung ausgeführt, entspricht diese Angabe der Anzahl

der Worte, die übertragen wird. Bei einer 32-Bit-Anweisung in der Form DFROM oder DTO wird die Anzahl der zu übertragenden Doppelworte angegeben.

16-Bit-Anweisung Anzahl der Daten: 5		32-Bit-Anweisung Anzahl der Daten: 2	
D100	Adr. 5	D100	Adr. 5
D101	Adr. 6	D101	Adr. 6
D102	Adr. 7	D102	Adr. 7
D103	Adr. 8	D103	Adr. 8
D104	Adr. 9	D104	Adr. 9

Abb. 7-26: Anzahl der Datenworte bei einer 16-Bit- und bei einer 32-Bit-Anweisung

Datenziel oder -quelle im Grundgerät

In den meisten Fällen werden die Daten aus Registern gelesen und an ein Sondermodul übertragen oder aus dessen Pufferspeicher in den Datenregisterbereich des Grundgeräts transferiert. Als Datenziel oder -quelle können aber auch Ausgänge und Merker oder Timer- und Counter-Istwerte dienen.

7.3.1 Auslesen von Daten aus einem Sondermodul (FROM)

			FROM		FNC 78				
			Auslesen von Daten aus einem Sondermodul						
			CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
					●	●	●		●
Operanden	D+	n1, n2, n3	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	KnY, KnM, KnS, T, C, D, R*, V, Z	K, H, D*, R*	●	16 Bit	32 Bit	FROM/FROMP	9	DFROM/DFROMP	17

* Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

Funktionsweise

Auslesen von Daten aus den Pufferspeichern der angeschlossenen Sondermodule

Beschreibung

- Auslesen von n3 Datenworten aus dem Sondermodul mit der Adresse n1.
- Es werden n3 Datenworte, ausgehend von der Pufferspeicheradresse n2 nach (D+) bis [(D+) + (n3 - 1)] geschrieben.

HINWEISE

Zulässige Bereiche: n1 = 0 bis 7
 n2 = 0 bis 31, 0 bis 32767 bei FX2N/FX2NC/FX3G/FX3GC/
 FX3GE/FX3U/FX3UC
 n3 = 1 bis 32 bei einer 16 Bit-Anweisung
 (1 bis 32767 bei FX3G/FX3GC/FX3GE/FX3U/FX3UC)
 1 bis 16 bei einer 32 Bit-Anweisung
 (1 bis 16383 bei FX3G/FX3GC/FX3GE/FX3U/FX3UC)

Wenn mit einer FROM-Anweisung große Datenmengen übertragen werden, kann ein Watchdog-Timer-Fehler auftreten. In diesem Fall sollten die Daten aufgeteilt oder die Einstellung des Watchdog-Timers verändert werden. Bei der FX3U- und FX3UC-Serie kann alternativ eine RBFM-Anweisung verwendet werden.

Mit dem Sondermerker M8028 kann bei der FX2N ab Version 3.0 und einer FX3G/FX3GC/
 FX3GE/FX3U/FX3UC-SPS festgelegt werden, ob der Datentransfer bei einem Interrupt unterbrochen oder vollständig abgeschlossen wird:

Interrupts gesperrt (M8028 = 0)	Interrupts freigegeben (M8028 = 1)
Erst nach Abschluss des Datenaustausches wird in das Interrupt-Programm verzweigt.	Bei einem Interrupt wird der Datenaustausch unterbrochen und in das Interrupt-Programm verzweigt.
Der Sprung ins Interrupt-Programm wird durch die Ausführung der FROM-Anweisung verzögert. Da der Datenaustausch nicht unterbrochen wird, entsteht kein Konflikt mit FROM-Anweisungen, die im Interrupt-Programm verwendet werden.	Nach Beendigung des Interrupt-Programms wird der unterbrochene Datenaustausch fortgesetzt. Falls im Interrupt-Programm ebenfalls eine FROM-Anweisung verwendet wurde, ist das evtl. nicht erwünscht. M8028 sollte nur bei Applikationen gesetzt werden, bei denen eine kurze Verzögerung des Interrupt-Programms im Vordergrund steht.

Beispiel ▾

Daten auslesen

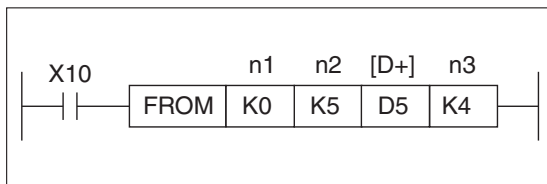


Abb. 7-27:
 Programmierbeispiel zur FROM-Anweisung

Mit dem gezeigten Beispiel wird der Inhalt der Pufferspeicheradressen #5 bis #8 des Sondermoduls mit der Positionsnummer 0 nach D5 bis D8 übertragen. Die Bedeutung der Adressierung ist im Einzelnen in der nachfolgenden Abb. dargestellt.

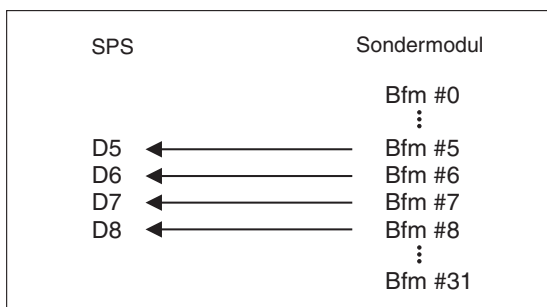


Abb. 7-28:
 Adressierung bei der FROM-Anweisung



HINWEIS

Wenn nur der Inhalt einer Pufferspeicheradresse übertragen werden soll, muss in n3 der Wert 1 eingetragen werden.

7.3.2 Schreiben von Daten in ein Sondermodul (TO)

		TO		FNC 79			
		Schreiben von Daten in ein Sondermodul					
Operanden	S+	n1, n2, n3	Puls-Anweisung (P)	Verarbeitung		Programmschritte	
				16 Bit	32 Bit	TO/TOP	9
	KnY, KnM, KnS, T, C, D, R*, V, Z	K, H, D*, R*		●	●	●	●
				●	●	DT0/DTOP	17

* Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

Funktionsweise

Schreiben von Daten aus der SPS in die Pufferspeicher der anschließbaren Sondermodule

Beschreibung

- Schreiben von n3 Datenworten aus der SPS in das Sondermodul mit der Adresse n1.
- Es werden n3 Datenworte, ausgehend von der Pufferspeicheradresse (S+), nach n2 bis [(n2) + (n3 - 1)] geschrieben.

HINWEISE

Zulässige Bereiche:

- n1 = 0 bis 7
- n2 = 0 bis 31, 0 bis 32767 bei FX2N/FX2NC/FX3G/FX3GC/FX3GE/FX3U/FX3UC
- n3 = 1 bis 32 bei einer 16 Bit-Anweisung (1 bis 32767 bei FX3G/FX3GC/FX3GE/FX3U/FX3UC)
1 bis 16 bei einer 32 Bit-Anweisung (1 bis 16383 bei FX3G/FX3GC/FX3GE/FX3U/FX3UC)

Wenn mit einer TO-Anweisung große Datenmengen übertragen werden, kann ein Watchdog-Timer-Fehler auftreten. In diesem Fall sollten die Daten aufgeteilt oder die Einstellung des Watchdog-Timers verändert werden. Bei der FX3U- und FX3UC-Serie kann alternativ die WBFM-Anweisung verwendet werden.

Mit dem Sondermerker M8028 kann bei der FX2N ab Version 3.0 und einer FX3G/FX3GC/FX3GE/FX3U/FX3UC-SPS festgelegt werden, ob der Datentransfer bei einem Interrupt unterbrochen oder vollständig abgeschlossen wird:

Interrupts gesperrt (M8028 = 0)	Interrupts freigegeben (M8028 = 1)
Erst nach Abschluss des Datenaustausches wird in das Interrupt-Programm verzweigt.	Bei einem Interrupt wird der Datenaustausch unterbrochen und in das Interrupt-Programm verzweigt.
Der Sprung ins Interrupt-Programm wird durch die Ausführung der TO-Anweisung verzögert. Da der Datenaustausch nicht unterbrochen wird, entsteht kein Konflikt mit TO-Anweisungen, die im Interrupt-Programm verwendet werden.	Nach Beendigung des Interrupt-Programms wird der unterbrochene Datenaustausch fortgesetzt. Falls im Interrupt-Programm ebenfalls eine TO-Anweisung verwendet wurde, ist das evtl. nicht erwünscht. M8028 sollte nur bei Applikationen gesetzt werden, bei denen eine kurze Verzögerung des Interrupt-Programms im Vordergrund steht.

Beispiel ▾ Daten schreiben (TO)

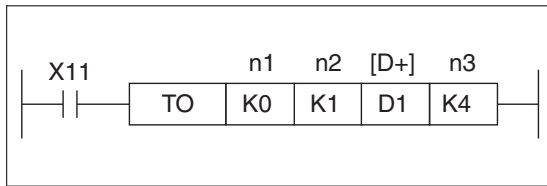


Abb. 7-29:
Programmierbeispiel für eine TO-Anweisung

Mit dem gezeigten Beispiel werden D1 bis D4 in die Pufferspeicheradressen #1 bis #4 des Sondermoduls mit der Positionsnummer 0 übertragen. Die Bedeutung der Adressierung ist im Einzelnen wie folgt:

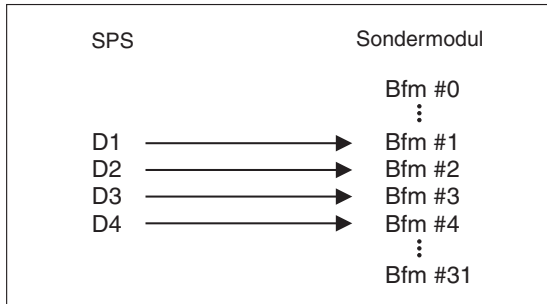


Abb. 7-30:
Adressierung bei der TO-Anweisung



HINWEIS | Wenn nur der Inhalt einer Pufferspeicheradresse übertragen werden soll, muss in n3 der Wert 1 eingetragen werden.

7.4 Serielle Kommunikation

Die Anweisungen FNC80 bis 89 können in Verbindung mit Geräten angewendet werden, die an einer seriellen Schnittstelle der SPS angeschlossen sind.

Übersicht der Anweisungen FNC 80 bis 87

Symbol	FNC	Bedeutung	Abschnitt
RS	80	Serielle Datenübertragung	7.4.1
PRUN	81	Umlegen von Eingängen oder Merkern	7.4.2
ASCI	82	Umwandlung in ein ASCII-Zeichen	7.4.3
HEX	83	Umwandlung in einen Hexadezimalwert	7.4.4
CCD	84	Summen- und Paritätsprüfung	7.4.5
VRRD	85	Einlesen von Sollwerten vom FX□-8AV-BD	7.4.6
VRSC	86	Einlesen von Schalterstellungen vom FX□-8AV-BD	7.4.7
RS2	87	Serielle Datenübertragung	7.4.8

Tab. 7-10: Übersicht der Anweisungen für serielle Kommunikation

7.4.1 Serielle Datenübertragung (RS)

				RS		FNC 80				
				Serielle Datenübertragung						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
					●	●	●	●	●	●
Operanden	S+	D+	n1, n2	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	D, R*	D, R*	K, H, D*, R*			16 Bit	32 Bit	RS	9	
					●					

* Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

Funktionsweise

Übertragung von Daten über eine RS232- oder RS485-Schnittstelle der FX-Steuerung.

Beschreibung

Mit Hilfe der RS-Anweisung ist das Senden und Empfangen von bzw. zu einer Vielzahl von Geräten mit serieller Schnittstelle möglich. Die Kommunikation über die seriellen Schnittstelle wird dabei in vier Teilabschnitten gesteuert:

- 1 Einstellen der Kommunikationsparameter
- 2 Ausgabe der RS-Anweisung, bestehend aus:
 - (S+) = Startadresse des Übertragungspuffers
 - n1 = Anzahl der zu übertragenden Bytes (max. 4096 Bytes, bei FX1S/FX1N max. 256 Bytes)
 - (D+) = Startadresse des Empfangspuffers
 - n2 = Anzahl der zu empfangenden Bytes (max. 4096 Bytes, bei FX1S/FX1N max. 256 Bytes)

Die Summe aus n1 und n2 darf 8000 Bytes nicht überschreiben (512 Bytes bei einer Steuerung der FX1S/FX1N-Serie).

- 3 Meldung übertragen
- 4 Meldung empfangen

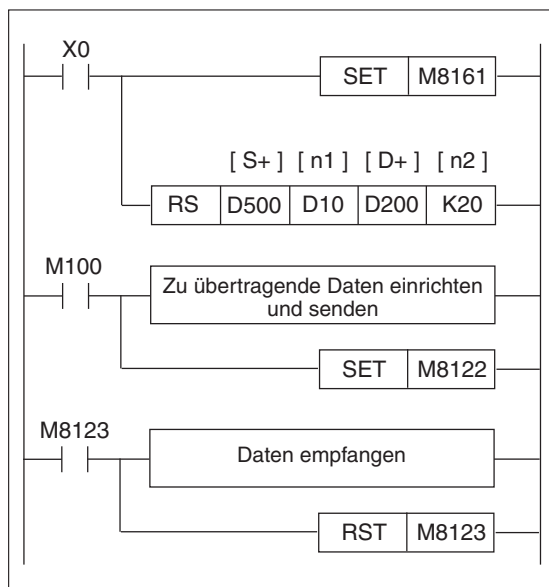


Abb. 7-31:
Programmierbeispiel für die RS-Anweisung

Bei einer Steuerung der FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC-Serie kann auch eine RS2-Anweisung verwendet werden. In der folgenden Tabelle sind die RS- und die RS2-Anweisung gegenübergestellt.

Merkmal	RS-Anweisung	RS2-Anweisung
Größe des Headers	1 Zeichen (Byte)	1 bis 4 Zeichen (Byte)
Größe der Endekennung	1 Zeichen (Byte)	1 bis 4 Zeichen (Byte)
Anfügen einer Prüfsumme	Die Prüfsumme sollte durch das Anwenderprogramm angefügt werden.	Die Prüfsumme kann automatisch angefügt werden*.

Tab. 7-11: Unterschiede zwischen der RS- und der RS2-Anweisung

* Falls bei der RS2-Anweisung die Prüfsumme automatisch mit den Daten übertragen wird, muss auch eine Endekennung verwendet werden.

HINWEISE

Eine RS-Anweisung kann bei einer FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC-Steuerung nur für die 1. serielle Schnittstelle (Ch.1) verwendet werden.

Für eine Schnittstelle darf jeweils nur eine RS- oder RS2-Anweisung aktiviert sein.

Bei einer FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC darf eine RS- oder RS2-Anweisung nicht gleichzeitig mit einer IVCK-, IVDR-, IVRD-, IVWR-, IVBWR-, FLCRT-, FLDEL-, FLWR-, FLRD-, FLCMD- oder FLSTRD-Anweisung für dieselbe Schnittstelle ausgeführt werden.

Kommunikationsparameter

Jedes Protokoll einer seriellen Kommunikation muss zunächst konfiguriert werden, damit Kompatibilität mit dem Peripheriegerät gewährleistet ist. Das Kommunikationsprotokoll für die Module wird mit Hilfe des Sonderregisters D8120 konfiguriert. Dies ist jedoch nur bei inaktiver RS-Anweisung möglich. Die folgende Tabelle zeigt die Zusammensetzung des Sonderregisters D8120 und seiner Bedeutung für die RS232-Kommunikation.

	Beschreibung	0	1
b0	Datenlänge	7 Bit	8 Bit
b1	Parität	(00): keine Parität	
b2		(01): ungerade Parität (11): gerade Parität	
b3	Stoppbit	1 Bit	2 Bit
b4	Übertragungsgeschwindigkeit (Bit/s)	(0011): 300, (0100): 600,	
b5		(0101): 1200, (0110): 2400,	
b6		(0111): 4800, (1000): 9600	
b7		(1001): 19200	
b8	Start-Byte	Kein	Mit D8124
b9	End-Byte	Kein	Mit D8125
b10	Handshake Typ1	Kein	H/W1
b11	Modus Control (Line)	Normal	Single
b12	Handshake Typ2	Kein	H/W2
b13	FX-485 Summenprüfung	Keine Prüfung	Prüfung
b14	Netzwerk	Aus	Aktiv
b15	Protokoll	Format1	Format4

Tab. 7-12:

Bits des Registers D8120 und ihre jeweilige Bedeutung für die RS-232-Kommunikation

Das Sonderregister D8124 enthält den Wert des Start-Bytes, falls eines ausgewählt wurde. Der Grundwert ist ASCII „STX“ oder 02_H. Dieser kann jedoch vom Anwender vor Kommunikationsbeginn geändert werden.

Das Sonderregister D8125 enthält den Wert des End-Bytes, falls eines ausgewählt wurde. Der Default-Wert ist ASCII „ETX“ oder 03_H. Er kann jedoch vom Anwender vor Kommunikationsbeginn geändert werden.

Wenn die periphere Kommunikationseinheit mit Hardware-Handshake arbeitet, sollte dieser Modus gewählt werden. Ist er selektiert, dienen die DSR- und DTR-Anschlüsse (Pin 6 und 20) des Schnittstellenadapters zur Kommunikationssteuerung. Das Anschlussdiagramm finden Sie in der Hardware-Beschreibung zum Adapter.

Detaillierte Informationen finden Sie außerdem im Kommunikationshandbuch zur FX-Familie.

Zusammensetzung der RS-Anweisung

Als Datenspeicherformat steht wahlweise der 16-Bit- oder der 8-Bit-Modus zur Verfügung. Der 16-Bit-Modus verwendet die oberen und unteren Bytes der Übertragungs- und Empfangspufferbereiche, während der 8-Bit-Modus nur die unteren 8 Bits verwendet. Gesteuert wird dies durch den Sondermerker M8161. ON steht hierbei für 8-Bit-Modus.

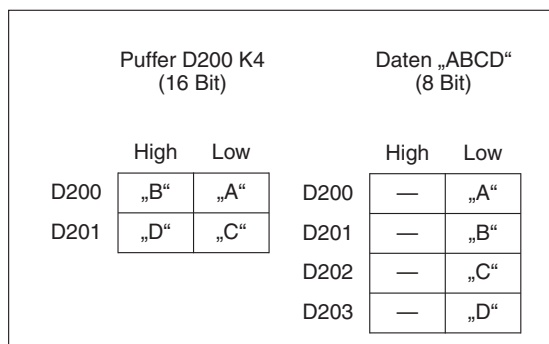


Abb. 7-32:
Adressierung bei der RS-Anweisung

Die RS-Anweisung setzt sich im Einzelnen wie folgt zusammen:

- **Anweisung:**
Ist die RS-Anweisung aktiv, bedeutet das, dass eine Kommunikation möglich ist und Daten gesendet und empfangen werden können. Innerhalb eines Programms ist mehr als eine RS-Anweisung zulässig, wobei jedoch nicht mehr als eine Anweisung gleichzeitig aktiv sein darf.
- **Startadresse des Übertragungspuffers**
Die Startadresse des Übertragungspuffers ist das erste Daten- oder File-Register (D, R) des Bereichs für zu übertragende Meldungen.
- **Länge der zu übertragenden Meldung**
Hier wird die Länge der zu übertragenden Meldung festgelegt. Der Wert kann eine Konstante (K, H) sein; bei variierender Meldungslänge kann aber auch ein Datenregister (D, R) verwendet werden. Bei Verwendung eines Datenregisters lässt sich der Wert zwischen einzelnen Übertragungsanforderungen, nicht jedoch während der Übertragung selbst ändern.
- **Startadresse des Empfangspuffers**
Die Startadresse des Empfangspuffers ist das erste Datenregister (D, R) des Bereichs für empfangene Meldungen.
- **Länge der zu empfangenden Meldung**
Hier wird die maximale Länge der zu empfangenden Meldung bestimmt. Der Wert kann eine Konstante (K, H) sein; bei variierender Meldungslänge kann aber auch ein Datenregister (D, R) verwendet werden. Bei Verwendung eines Datenregisters lässt sich der Wert zwischen einzelnen Empfangsvorgängen, nicht jedoch während des Empfangs selbst ändern.

Meldung übertragen

Die Steuerung der Übertragung bzw. des Sendens einer Meldung erfolgt über den Sondermerker M8122.

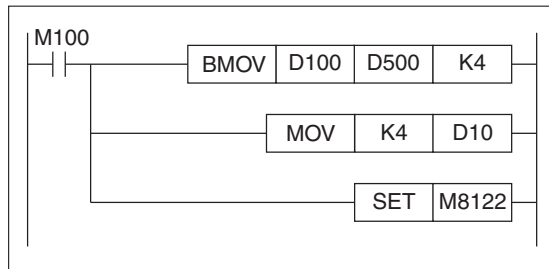


Abb. 7-33:
 Programmierbeispiel zur Übertragung mit Hilfe des Sondermerkers M8122

Zunächst müssen die zu übertragenden Daten im Übertragungspufferbereich enthalten sein. Sie können Sie auf zwei Arten dorthin übertragen:

- ① Kopieren Sie vor der Übertragung die Meldung in den Meldungspufferbereich bzw. erzeugen Sie sie mit Hilfe der MOV- oder der BMOV-Anweisung.
- ② Ändern Sie die Parameter der RS-Anweisung, um auf den entsprechenden Datenregisterbereich zuzugreifen, in dem sich die Meldung befindet. Für jede Meldung ist eine eigene RS-Anweisung erforderlich.

Im oben beschriebenen Beispiel werden die in den Datenregistern D100 bis D103 enthaltenen Daten in den Übertragungspufferbereich, beginnend bei D500, kopiert. Danach wird die Meldungslänge auf 8 Byte festgelegt, indem der Wert für die Länge der zu übertragenden Meldung mit Hilfe des Datenregisters D10 geändert wird.

Sind die Daten definiert und korrekt lokalisiert, kann der Übertragungsmerker M8122 auf ON gesetzt werden. Nun beginnt der Versand der Daten, und der Merker M8122 wird automatisch zurückgesetzt, sobald die Datenübertragung abgeschlossen ist.

Es wird empfohlen, den Merker mit einem Impuls-Signal zu setzen, da der Merker andernfalls nach erfolgter Übertragung wieder auf ON gesetzt wird und sich die Datenübertragung wiederholt.

Bei Verwendung von Start- und/oder Endsignalen (Header/Terminator) werden diese automatisch der Meldung vor der Übertragung hinzugefügt.

Ein Übertragungszähler über das Sonderregister ist ebenfalls möglich. Das Sonderregister D8122 kann während des Sendevorgangs überprüft werden, so dass Sie den Verlauf der Übertragung verfolgen können. Der Wert in D8122 beginnt bei der vollen Meldungslänge und wird pro übertragenem Daten-Byte um jeweils eins heruntergezählt.

HINWEIS

| Die Start- und Endsignale werden im D8122 nicht mitgezählt.

Meldung empfangen

Die RS-Anweisung steuert automatisch den Empfang einer Meldung. Sobald eine Meldung vollständig empfangen ist, werden die Daten im Empfangspufferbereich gespeichert, und der Sondermerker M8123 wird auf ON gesetzt.

Eventuell verwendete Start- und/oder Endsignale werden automatisch entfernt, bevor die Meldung im Pufferbereich abgelegt wird.

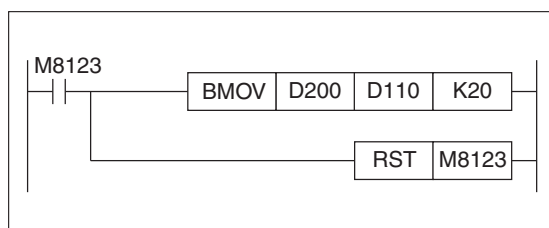


Abb. 7-34:
 Programmierbeispiel zur Übertragung mit Hilfe des Sondermerkers M8123

Sobald der Sondermerker auf ON steht, sollten die im Empfangspuffer enthaltenen Daten weiter verarbeitet und der Merker auf OFF zurückgesetzt werden, damit der Bereich zum Empfang von weiteren Meldungen zur Verfügung steht. Der Merker wird automatisch zurückgesetzt, wenn die RS-Anweisung abgeschaltet wird.

Im oben beschriebenen Programmbeispiel wird Sondermerker M8123 „Meldung empfangen“ überprüft. Wird er aktiviert, werden alle im Empfangspuffer enthaltenen Daten in eine andere Adresse kopiert, und der Empfangsmerker wird zurückgesetzt. Nachdem der Empfangspufferbereich wieder frei ist, können die empfangenen Daten nach Bedarf weiter verarbeitet werden.

HINWEIS

Das gleichzeitige Senden und Empfangen von Daten ist nicht möglich. Der Merker M8121 (Empfangskennung) steht während des Empfangsvorgangs auf ON. Zwar kann der Merker M8122 (Sendekennung) zur selben Zeit auch auf ON stehen, aber der tatsächliche Sendevorgang wird so lange verzögert, bis die Meldung vollständig empfangen ist.

Die Einrichtung eines Empfangszähler ist ebenfalls möglich. Während des Empfangs kann im Sonderregister D8123 überprüft werden, wieviel Bytes aktuell übertragen worden sind. Nach vollständig empfangener Meldung wird die Gesamtlänge der Meldung angezeigt.

Start- und Endsignale (Header, Terminator)**● Beschreibung**

Bei der Datenkommunikation ist es oft erforderlich, den Anfang und das Ende einer Meldung besonders zu kennzeichnen. Dies geschieht üblicherweise mit Hilfe bestimmter Zusätze zur Meldung, sogenannten Start- und Endsignalen. Mit der RS-Anweisung haben Sie die Möglichkeit, der Meldung automatisch ein Start-Byte und/oder ein End-Byte hinzuzufügen.

Ausgewählt werden das Start- und das End-Byte durch Setzen der Bits b8 und b9 im Datenregister D8120 der Kommunikationsparameter.

● Während der Übertragung

Wurde ein Start-Signal gewählt, wird das untere Byte des Sonderregisters D8124 als das erste Byte jeder übertragenen Meldung gesendet.

Wurde ein End-Signal gewählt, wird das untere Byte des Sonderregisters D8125 als das letzte Byte jeder übertragenen Meldung gesendet.

● Während des Empfangs

Wurde ein Start-Signal gewählt, werden alle empfangenen Daten so lange ignoriert, bis das Start-Byte empfangen wurde. Wurde kein Start-Signal gewählt, wird das erste empfangene Byte als Meldungsinhalt gelesen.

Wurde ein End-Signal gewählt, werden beim Lesen alle empfangenen Daten als eine Meldung gelesen, bis das End-Signal empfangen wird oder die Gesamtmeldungslänge erreicht ist, d.h. der Empfangspuffer voll ist.

Wurde kein End-Signal gewählt, dauert der Lesevorgang so lange an, bis der Empfangspuffer voll ist, d.h. die Meldung muss in voller Länge eingegangen sein, bevor sie als vollständig gewertet wird.

Nach Empfang einer vollständigen Meldung wird der Merker M8123 gesetzt. Alle danach empfangenen Daten werden so lange ignoriert, bis dieser Empfangsmerker wieder gelöscht wird.

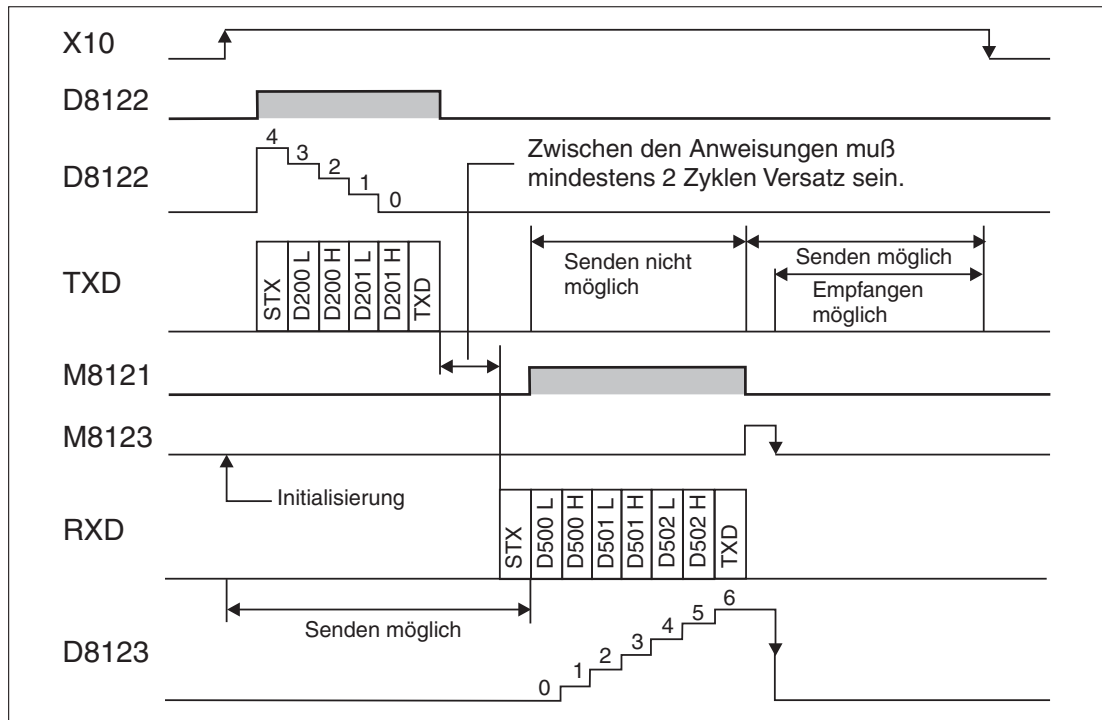


Abb. 7-35: Darstellung des zeitlichen Ablaufs

7.4.2 Umlegen von Eingängen oder Merkern (PRUN)

			PRUN		FNC 81			
			Umlegen von Eingängen oder Merkern					
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	KnX, KnM n = 1-8	KnM, KnY n = 1-8		16 Bit	32 Bit	PRUN/PRUNP	5	
			●	●	DPRUN/DPRUNP	9		

Funktionsweise

Übertragen von Eingängen in einen Merkerbereich oder von Merkern in einen Ausgangsbereich.

Beschreibung

- Die PRUN-Anweisung unterscheidet sich von der MOV-Anweisung darin, dass sie oktal arbeitet.
- Da die Übertragung oktal erfolgt, sollte (S+) mit X / M 0, 10, 20, 30 ... beginnen.

Beispiel ▾

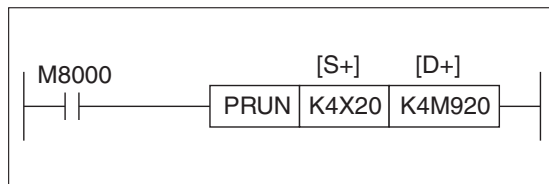


Abb. 7-36:
Programmierbeispiel zur Übertragung der Zustände von Eingängen in Merker

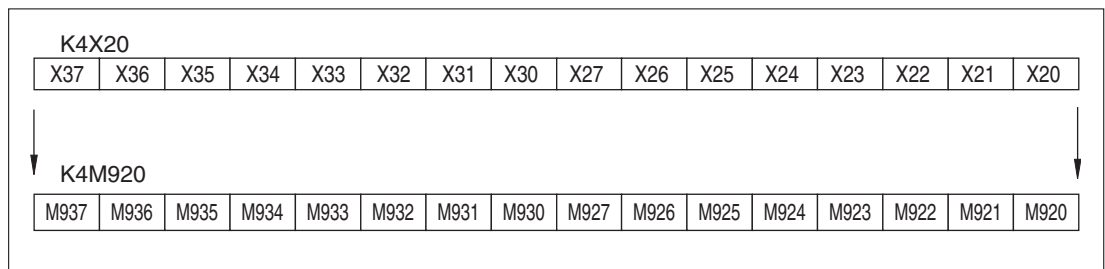


Abb. 7-37: Zuordnung der Merker

Die Merker M929 und M928 werden nicht überschrieben, da die PRUN-Anweisung oktal arbeitet.



7.4.3 ASCII-Umwandlung (ASCII)

				ASCII		FNC 82				
				Umwandlung in ein ASCII-Zeichen						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
				●	●	●	●	●	●	
Operanden	S+	D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	K, H, T, C, D, R ^① , V, Z, U□\G□ ^② , KnX, KnY, KnM, KnS	T, C, D, R ^① , U□\G□ ^② , KnY, KnM, KnS	D ^① , R ^① , K, H		●	16 Bit	32 Bit	ASCII/ASCIP	7	

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktionsweise

Umwandlung eines hexadezimalen Wertes in ein ASCII-Zeichen

Beschreibung

Die ASCII-Anweisung ermöglicht die Umwandlung eines hexadezimalen Wertes aus einem Datenregister in ein ASCII-Zeichen.

Die ASCII-Anweisung setzt sich im einzelnen wie folgt zusammen:

- der Startadresse (S+), wo die hexadezimalen Daten gespeichert sind
- die Zieladresse (D+), wo die umgewandelten ASCII-Zeichen gespeichert werden sollen
- die Angabe über die Anzahl der Zeichen (n), d.h. die Zahl der in ASCII-Zeichen umzuwandelnden hexadezimalen Ziffern

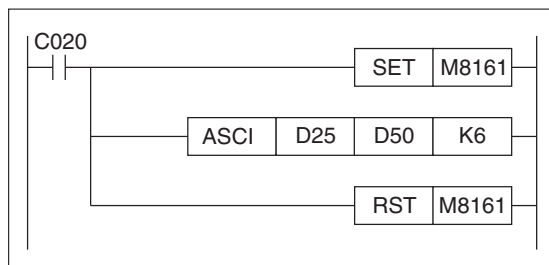


Abb. 7-38:
 Programmierbeispiel für die ASCII-Anweisung

Das dargestellte Programm führt die Umwandlung aus, wenn der Zähler C20 seinen festgelegten Wert erreicht. Während die ASCII-Anweisung aktiv ist, werden die sechs (K6) hexadezimalen Ziffern aus den Datenregistern D25 und D26 in ASCII-Zeichen umgewandelt und in den Datenregistern D50 bis D55 (8-Bit-Format) gespeichert. Dabei ist jedes Zeichen ein Byte.

Als Datenspeicherformat steht wahlweise der 16-Bit- oder der 8-Bit-Modus zur Verfügung. Der 16-Bit-Modus verwendet die oberen und unteren Bytes der Zieleinheiten (D+), während der 8-Bit-Modus nur die unteren 8 Bits verwendet. Gesteuert wird dies durch den Sondermerker M8161 (M8161 = „1“: 8-Bit-Modus).

HINWEISE

Die Anweisung „SET M8161“ ist nur dann erforderlich, wenn der 8-Bit-Modus verwendet werden soll.

Der Sondermerker M8161 beeinflusst auch das Verhalten einer RS-, HEX-, CCD- und CRC-Anweisung. Wird eine dieser Anweisungen im selben Programm wie die ASCII-Anweisung verwendet, sollte M8161 vor der Ausführung der ASCII-Anweisung gesetzt und unmittelbar nach Ausführung der ASCII-Anweisung wieder zurückgesetzt werden.

Zusammensetzung der ASCI-Anweisung

● Startadresse

Der hier definierte Wert bezeichnet den ersten Wortoperanden, der die umzuwandelnden hexadezimalen Zahlen enthält. Sollen mehr als 4 Ziffern umgewandelt werden, werden die folgenden Wortoperanden ebenfalls gelesen, bis alle gewünschten Ziffern umgewandelt sind.

● Zieladresse

Der hier definierte Wert bezeichnet den ersten Wortoperanden, der die ASCII-Zeichen enthalten soll. Jeder Wortoperand kann 2 Zeichen (2 Bytes) enthalten. Die der Startadresse folgenden Wortoperanden werden so lange verwendet, bis alle Zeichen gespeichert sind.

● Anzahl der Zeichen

Der hier angegebene Wert kann nur ein Dezimalwert (K) oder ein hexadezimaler Wert (H) sein. Er bezeichnet die Anzahl der hexadezimalen Ziffern, die umgewandelt werden sollen, und wie viele ASCII-Zeichen gespeichert werden sollen. Die Anzahl der Zeichen kann zwischen 1 und 256 Ziffern betragen.

● Rücksetzung des Datenspeicherformats

Durch Zurücksetzen des Sondermerkers M8161 wird der 16-Bit-Modus aktiviert.

HINWEIS

Die Funktion „RST M8161“ wird nur dann benötigt, wenn in der ASCI-Anweisung das 8-Bit-Datenspeicherformat verwendet wird, während andere Anweisungen in Ihrem Programm das 16-Bit-Datenspeicherformat verwenden.

Anwendungsbeispiel

Entsprechend dem Programmbeispiel auf der vorhergehenden Seite zeigt das folgende Diagramm die Ergebnisse sowohl für das 16-Bit- als auch das 8-Bit-Format.

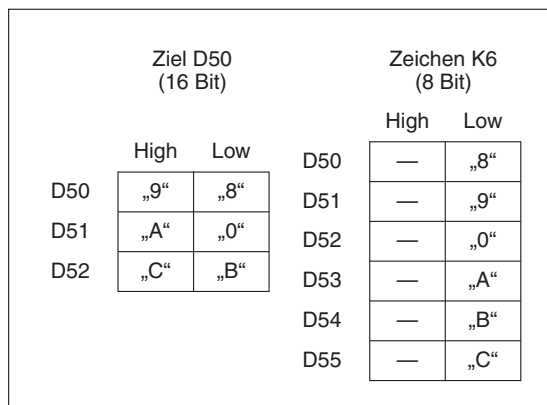


Abb. 7-39:
Grafische Darstellung im 16-Bit- und im 8-Bit-Format

Die folgende Tabelle enthält die ASCII-Codes für alle wandelbaren hexadezimalen Zahlen sowohl in hexadezimalen als auch in dezimalen Ziffern.

HEX	ASCII		Symb.	HEX	ASCII		Symb.	HEX	ASCII		Symb.	HEX	ASCII		Symb.
	HEX	DEZ			HEX	DEZ			HEX	DEZ			HEX	DEZ	
0	30	48	„0“	4	34	52	„4“	8	38	56	„8“	C	43	67	„C“
1	31	49	„1“	5	35	53	„5“	9	39	57	„9“	D	44	68	„D“
2	32	50	„2“	6	36	54	„6“	A	41	65	„A“	E	45	69	„E“
3	33	51	„3“	7	37	55	„7“	B	42	66	„B“	F	46	70	„F“

Tab. 7-13: ASCII-Code

7.4.4 Hexadezimal-Umwandlung (HEX)

				HEX		FNC 83				
				Umwandlung in einen Hexadezimalwert						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
					●	●	●	●	●	●
Operanden	S+	D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	K, H, T, C, D, R ^① , U□G□ ^② , KnX, KnY, KnM, KnS	T, C, D, R ^① , V, Z, U□G□ ^② , KnY, KnM, KnS	D ^① , R ^① , K, H		●	16 Bit	32 Bit	HEX/ HEXP	7	

- ① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ② Nur bei FX3U und FX3UC

Funktionsweise

Umwandlung eines ASCII-Zeichens in einen Hexadezimalwert

Beschreibung

Die HEX-Anweisung ermöglicht die Umwandlung eines ASCII-Zeichens in einen hexadezimalen Wert.

Die HEX-Anweisung setzt sich im einzelnen wie folgt zusammen:

- der Startadresse (S+), wo die ASCII-Zeichen gespeichert sind
- die Zieladresse (D+), in der die gewandelten hexadezimalen Daten gespeichert werden
- die Angabe über die Anzahl der Zeichen (n), d.h. die Zahl der in hexadezimale Ziffern umzuwandelnden ASCII-Zeichen

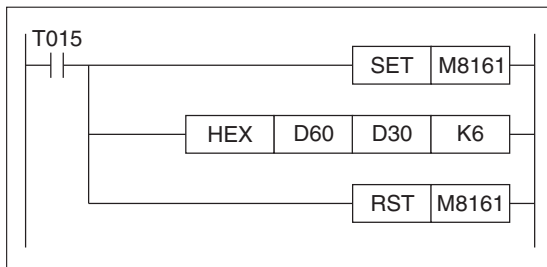


Abb. 7-40:
Programmierbeispiel für die HEX-Anweisung

Das oben dargestellte Programm führt die Umwandlung aus, wenn der Zähler T15 seinen festgelegten Wert erreicht. Während die HEX-Anweisung aktiv ist, werden die sechs (K6) ASCII-Zeichen aus den Datenregistern D60 und D65 in hexadezimale Ziffern umgewandelt und in den Datenregistern D30 und D31 (8-Bit-Format) gespeichert. Dabei ist jede Ziffer ein Byte.

Als Datenspeicherformat steht wahlweise der 16-Bit- oder der 8-Bit-Modus zur Verfügung. Der 16-Bit-Modus verwendet die oberen und unteren Bytes der Quelleinheiten, während der 8-Bit-Modus nur die unteren 8 Bits verwendet. Gesteuert wird dies durch den Sondermerker M8161 (M8161 = „1“: 8-Bit-Modus).

HINWEISE

Die Anweisung „SET M8161“ ist nur dann erforderlich, wenn der 8-Bit-Modus verwendet werden soll.

Der Sondermerker M8161 beeinflusst auch das Verhalten einer RS-, ASCII-, CCD- und CRC-Anweisung. Wenn eine dieser Anweisungen im selben Programm wie die HEX-Anweisung verwendet wird, sollte M8161 vor der Ausführung der HEX-Anweisung gesetzt und unmittelbar nach Ausführung der HEX-Anweisung wieder zurückgesetzt werden.

Zusammensetzung der HEX-Anweisung

- Startadresse

Der hier definierte Wert bezeichnet den ersten Wortoperanden, der die umzuwandelnden ASCII-Zeichen enthält. Sollen mehr als 2 Zeichen (2 Bytes) umgewandelt werden, werden die folgenden Wortoperanden ebenfalls gelesen, bis alle gewünschten Zeichen umgewandelt sind.

- Zieladresse

Der hier definierte Wert bezeichnet den ersten Wortoperanden, der die hexadezimalen Zahlen enthalten soll. Jeder Wortoperand kann 4 Ziffern enthalten. Die der Startadresse folgenden Wortoperanden werden so lange verwendet, bis alle Ziffern gespeichert sind.

- Anzahl der Zeichen

Der hier angegebene Wert kann nur ein Dezimalwert (K) oder ein hexadezimaler Wert (H) sein. Er bezeichnet die Anzahl der ASCII-Zeichen, die umgewandelt werden sollen, und wie viele hexadezimale Ziffern gespeichert werden sollen. Die Anzahl der Zeichen kann zwischen 1 und 256 Ziffern betragen.

- Rücksetzung des Datenspeicherformats

Durch Zurücksetzen des Sondermerkers M8161 wird der 16-Bit-Modus aktiviert.

HINWEIS Die Funktion „RST M8161“ wird nur dann benötigt, wenn in dieser Anweisung das 8-Bit-Datenspeicherformat verwendet wird, während andere Anweisungen in Ihrem Programm das 16-Bit-Datenspeicherformat verwenden.

Anwendungsbeispiel

Entsprechend dem oben abgebildeten Programmbeispiel zeigt das folgende Diagramm die Ergebnisse sowohl für das 16-Bit- als auch das 8-Bit-Format.

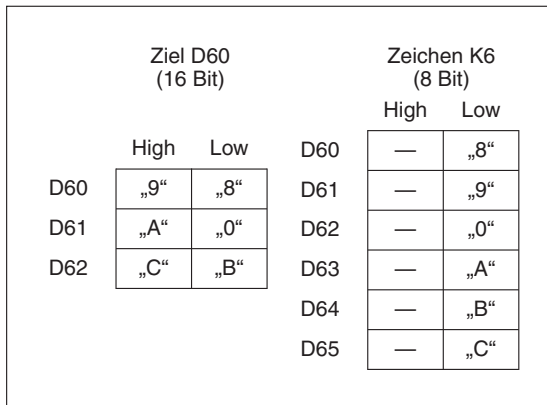


Abb. 7-41: Grafische Darstellung im 16-Bit- und im 8-Bit-Format

Die folgende Tabelle enthält die ASCII-Codes für alle wandelbaren hexadezimalen Zahlen sowohl in hexadezimalen als auch in dezimalen Ziffern.

HEX	ASCII		Symb.	HEX	ASCII		Symb.	HEX	ASCII		Symb.	HEX	ASCII		Symb.
	HEX	DEZ			HEX	DEZ			HEX	DEZ			HEX	DEZ	
0	30	48	„0“	4	34	52	„4“	8	38	56	„8“	C	43	67	„C“
1	31	49	„1“	5	35	53	„5“	9	39	57	„9“	D	44	68	„D“
2	32	50	„2“	6	36	54	„6“	A	41	65	„A“	E	45	69	„E“
3	33	51	„3“	7	37	55	„7“	B	42	66	„B“	F	46	70	„F“

Tab. 7-14: ASCII-Code

7.4.5 Summen- und Paritätsprüfung (CCD)

				CCD		FNC 84				
				Summen- und Paritätsprüfung						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
					●	●	●	●	●	●
Operanden	S+		D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	T, C, D, R ^① , U□G□ ^② , KnX, KnY, KnM, KnS		T, C, D, R ^① , U□G□ ^② , KnY, KnM, KnS	D ^① , R ^① , K, H			16 Bit	32 Bit	CCD/ CCDP	7
						●				

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktionsweise

Paritätskontrolle und Berechnung einer Prüfsumme

Beschreibung

Die CCD-Anweisung ermöglicht die Berechnung einer Prüfsumme und die Paritätskontrolle eines Datenbereiches.

Die CCD-Anweisung setzt sich im Einzelnen wie folgt zusammen:

- Der Startadresse (S+), wo die Daten gespeichert sind
- Die Zieladresse (D+), wo der Wert der Prüfsumme gespeichert werden soll
- Die Anzahl der zu prüfenden Bytes (n), Bereich: 1 bis 256

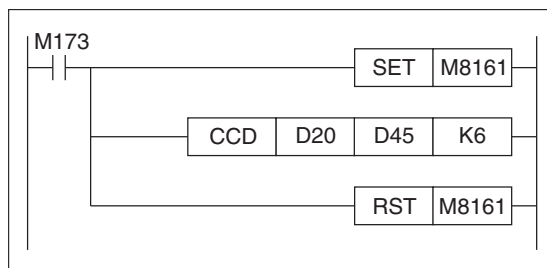


Abb. 7-42:
Programmierschema für die CCD-Anweisung

Das oben dargestellte Programm führt die Summen- und Paritätsprüfung aus, wenn der Merker M173 aktiviert wird. Während die CCD-Anweisung aktiv ist, werden sechs (K6) Bytes aus den Datenregistern D20 bis D25 (8-Bit-Format) summiert und die Summe und die Parität in den Datenregistern D45 bzw. D46 gespeichert.

Als Datenspeicherformat steht wahlweise der 16-Bit- oder der 8-Bit-Modus zur Verfügung. Der 16-Bit-Modus verwendet die oberen und unteren Bytes der Quelleinheiten, während der 8-Bit-Modus nur die unteren 8 Bits verwendet. Gesteuert wird dies durch den Sondermerker M8161 (M8161 = „1“: 8-Bit-Modus).

HINWEISE

Die Anweisung „SET M8161“ ist nur dann erforderlich, wenn der 8-Bit-Modus verwendet werden soll.

Der Sondermerker M8161 beeinflusst auch das Verhalten einer RS-, ASCII-, HEX- und CRC-Anweisung. Wenn eine dieser Anweisungen im selben Programm wie die CCD-Anweisung verwendet wird, sollte M8161 vor der Ausführung der CCD-Anweisung gesetzt und unmittelbar nach Ausführung der CCD-Anweisung wieder zurückgesetzt werden.

7.4.6 Einlesen von Sollwerten vom FX□-8AV-BD (VRRD)

		VRRD		FNC 85			
		Einlesen von Sollwerten vom FX□-8AV-BD					
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
	●	●	●	● ^①	●	●	
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte	
	K, H, D ^② , R ^② S = 0 bis 7	KnY, KnM, KnS, T, C, D, R ^② , V, Z	●	16 Bit	32 Bit	VRRD/ VRRDP	5

① Die Anweisung kann nicht von einem FX3GC-Grundgerät ausgeführt werden.

② Nur bei FX3G/FX3GE/FX3U/FX3UC

Funktionsweise

Einlesen der auf einem FX1N-8AV-BD, FX2N-8AV-BD, FX3G-8AV-BD oder FX3U-8AV-BD eingestellten Sollwerte in die FX-Steuerung

Beschreibung

Mit der Anweisung VRRD wird die Einstellung des Potentiometers (S+) in einen 8-Bit-Wert (0 bis 255) gewandelt und in (D+) abgespeichert.

HINWEISE

Die Potentiometer sind von 0 bis 7 nummeriert.

Die Module FX1N-8AV-BD, FX2N-8AV-BD und FX3G-8AV-BD werden in den Erweiterungssteckplatz des FX1S/FX1N-, FX2N- bzw. FX3G/FX3GE/FX3S-Grundgeräts eingesetzt.

Bei einem Grundgerät FX3U/FX3UC-32MT-LT(-2) wird das Modul FX3U-8AV-BD verwendet.

Falls im Programm für eine FX3U oder FX3UC eine VRRD-Anweisung verwendet wird, steht der Kommunikationskanal 1 des Grundgeräts nicht mehr zur Verfügung.

Beispiel ▾

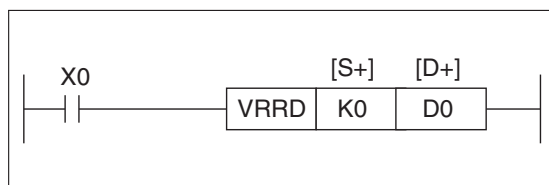



Abb. 7-44:

Programmierbeispiel zur VRRD-Anweisung

Das Potentiometer „0“ wird eingelesen, und der eingestellte Wert wird in dem Datenregister D0 abgespeichert.



7.4.7 Einlesen von Schalterstellungen vom FX□-8AV-BD (VRSC)

		VRSC		FNC 86			
		Einlesen von Schalterstellungen vom FX□-8AV-BD					
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
	●	●	●	● ^①	●	●	
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte	
	K, H, D ^② , R ^② S = 0 bis 7	KnY, KnM, KnS, T, C, D, R ^② , V, Z	●	16 Bit	32 Bit	VRSC/ VRSCP	5

① Die Anweisung kann nicht von einem FX3GC-Grundgerät ausgeführt werden.

② Nur bei FX3G/FX3GE/FX3U/FX3UC

Funktionsweise

Einlesen der auf einem FX1N-8AV-BD, FX2N-8AV-BD, FX3G-8AV-BD oder FX3U-8AV-BD eingestellten Schalter in die FX-Steuerung.

Beschreibung

- Mit der Anweisung VRSC wird die Stellung des Potentiometers (S+) als Schalterstellung interpretiert (0 bis 10) und in (D+) geschrieben.
- Die eingestellten Werte werden auf ganze Zahlen gerundet.

HINWEISE

Die Potentiometer sind von 0 bis 7 nummeriert.

Die Module FX1N-8AV-BD, FX2N-8AV-BD und FX3G-8AV-BD werden in den Erweiterungssteckplatz des FX1S/FX1N-, FX2N- bzw. FX3G/FX3GE/FX3S-Grundgeräts eingesetzt.

Bei einem Grundgerät FX3U/FX3UC-32MT-LT(-2) wird das Modul FX3U-8AV-BD verwendet.

Falls im Programm für eine FX3U oder FX3UC eine VRSC-Anweisung verwendet wird, steht der Kommunikationskanal 1 des Grundgeräts nicht mehr zur Verfügung.

Beispiel ▾

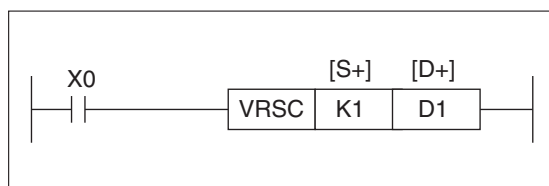


Abb. 7-45:

Programmierbeispiel zur VRSC-Anweisung

Die Schaltstellung von Schalter „1“ wird in das Datenregister D1 eingetragen.



7.4.8 Serielle Datenübertragung (RS2)

					RS2		FNC 80				
					Serielle Datenübertragung						
					CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
									●	●	●
Operanden	S+	D+	n1, n2	n3	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	D, R	D, R	K, H, D, R	K, H		16 Bit	32 Bit	RS2	11		
						●					

Funktionsweise

Übertragung von Daten über eine seriellen Schnittstelle der FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC-Steuerung

Beschreibung

- (S+): Anfangsadresse des Datenregisterbereichs, in dem die zu übertragenden Daten gespeichert sind
- n1: Anzahl der zu übertragenden Bytes (Bereich: 0 bis 4096 Bytes)
- (D+): Startadresse des Empfangspuffers
- n2: Anzahl der zu empfangenden Bytes (Bereich: 0 bis 4096 Bytes)
- n3: Nummer der seriellen Schnittstelle (K0 = Ch.0, K1 = Ch. 1, K2 = Ch 2)
Ch.0 kann nur bei einer FX3G/FX3GC/FX3GE angegeben werden, Ch. 2 steht bei FX3G-Grundgeräten mit 14- oder 24 E/A nicht zur Verfügung.

Die Summe aus n1 und n2 darf 8000 Bytes nicht überschreiben.

Es können nur die folgenden Schnittstellenadapter und Adaptermodule verwendet werden:

- FX3G-232-BD, FX3U-232-BD, FX3U-232ADP(-MB)
- FX3G-485-BD, FX3U-485-BD, FX3U-485ADP(-MB)

Die serielle Übertragung wird durch Sondermerker und -register gesteuert:

Sondermerker			Sonderregister		
Ch. 1	Ch. 2	Bedeutung	Ch. 1	Ch. 2	Bedeutung
			D8400	D8420	Übertragungsformat
M8401	M8421	Auf Datenübertragung warten			
M8402	M8422	Anforderung Datenübertragung	D8402	D8422	Noch zu sendene Bytes
M8403	M8423	Datenempfang beendet	D8403	D8423	Anzahl der empfangenen Bytes
M8404	M824	Trägererkennung			
			D8405	D8425	Anzeige der Kommunikationsparameter
M8409	M8429	Zeitüberlauf	D8409	D8429	Überwachungszeit
			D8410	D8430	Header 1, 2
			D8411	D8431	Header 3, 4
			D8412	D8432	Endekennung 1, 2
			D8413	D8433	Endekennung 3, 4
			D8414	D8434	Empfangene Prüfsumme
			D8415	D8435	Berechnete Prüfsumme
			D8416	D8436	Gesendete Prüfsumme
			D8419	D8439	Anzeige der Betriebsart
M8063	M8438	Fehler bei serieller Kommunikation	D8063	D8438	Fehlercode

Tab. 7-15: Sondermerker- und -register für serielle Kommunikation

Alternativ zur RS2-Anweisung kann für die Schnittstelle Ch. 1 auch eine RS-Anweisung verwendet werden (Abschnitt 7.4.1). In der folgenden Tabelle sind die RS- und die RS2-Anweisung gegenübergestellt.

Merkmal	RS-Anweisung	RS2-Anweisung
Größe des Headers	1 Zeichen (Byte)	1 bis 4 Zeichen (Byte)
Größe der Endekennung	1 Zeichen (Byte)	1 bis 4 Zeichen (Byte)
Anfügen einer Prüfsumme	Die Prüfsumme sollte durch das Anwenderprogramm angefügt werden.	Die Prüfsumme kann automatisch angefügt werden*.

Tab. 7-16: Unterschiede zwischen der RS- und der RS2-Anweisung

* Falls bei der RS2-Anweisung die Prüfsumme automatisch mit den Daten übertragen wird, muss auch eine Endekennung verwendet werden.

HINWEISE

Für eine Schnittstelle darf jeweils nur eine RS- oder RS2-Anweisung aktiviert sein.

Eine RS- oder RS2-Anweisung darf nicht gleichzeitig mit einer IVCK-, IVDR-, IVRD-, IVWR-, IVBWR-, FLCRT-, FLDEL-, FLWR-, FLRD-, FLCMD- oder FLSTRD-Anweisung für dieselbe Schnittstelle ausgeführt werden.

Eine Beschreibung der seriellen Kommunikation finden Sie im Abschnitt 7.4.1 (RS-Anweisung) und im Kommunikationshandbuch zur MELSEC FX-Familie, Art.-Nr. 137315.

7.5 PID-Regelung (PID)

				PID		FNC 88				
				PID-Regelung						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
				● ^①	●	●	●	●	●	
Operanden	S1+, S2+	S3+	D+	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	D, R ^② , U□\G□ ^③	D, R ^②	D, R ^② , U□\G□ ^③		16 Bit	32 Bit	PID	9		

- ① Die Steuerungen der FX1S-Serie können keine Analogwerte einlesen. Die Daten für die Regelung müssen der SPS über eine serielle Verbindung zugeführt werden.
- ② Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ③ Nur bei FX3U und FX3UC

Funktionsweise

Programmierung eines geschlossenen Regelkreises unter Angabe von 29 Regelparametern

Beschreibung

- (S1+) gibt den gewünschten Sollwert (SV) für den Regelkreis an.
Dieser Sollwert wird von der PID-Anweisung nicht verändert.
 - Bei Verwendung der Auto-Tuning-Funktion nach der Grenzyklus-Methode:
Falls der Sollwert für Auto-Tuning sich vom Sollwert für die Regelung unterscheidet, muss zuerst ein Sollwert eingestellt werden, der auch den Bias-Wert enthält. Der eigentliche Sollwert wird dann eingestellt, wenn die Auto-Tuning-Funktion beendet ist.
- (S2+) liest den erfassten Istwert (PV) als Rückmeldung für den Regler ein.
- (S3+) ist die Anfangsadresse des Operandenbereichs, in dem die Regelkreisparameter gespeichert werden.
 - Bei Verwendung der Auto-Tuning-Funktion nach der Grenzyklus-Methode:
Es werden 29 aufeinander folgende Operanden belegt.
 - Bei Verwendung der Auto-Tuning-Funktion nach der Sprungantwort-Methode:
Falls in ((S3+)+1) die Bits 1, 2 und 5 nicht alle auf „0“ gesetzt sind, werden 25 aufeinander folgende Operanden belegt.
Sind in ((S3+)+1) die Bits 1, 2 und 5 auf „0“ gesetzt, werden 20 aufeinander folgende Operanden belegt.

Wird kein Auto-Tuning verwendet, werden 25 aufeinander folgende Operanden belegt.
- (D+) enthält den berechneten Ausgangswert (MV) für die Regelung.
 - Bei der PID-Regelung (Normalbetrieb)
Vor dem Start der PID-Anweisung sollte durch den Anwender ein Anfangsausgangswert eingestellt werden. Danach wird der berechnete Ausgangswert gespeichert.
 - Bei der Auto-Tuning-Funktion nach der Grenzyklus-Methode
Während des Auto-Tuning-Vorgangs wird automatisch der obere oder untere Grenzwert für den Ausgangswert (ULV bzw. LLV) eingetragen. Nach Abschluss des Auto-Tuning wird der berechnete Ausgangswert gespeichert.
 - Bei der Auto-Tuning-Funktion nach der Sprungantwort-Methode
Vor dem Start der PID-Anweisung sollte durch den Anwender ein Anfangsausgangswert eingestellt werden. Während des Auto-Tuning wird der Ausgangswert durch die PID-Anweisung nicht verändert.

Die PID-Anweisung arbeitet mit der folgenden mathematischen Formel zur Berechnung des Ausgangswertes:

$$\text{Ausgangswert} = K_p \left\{ \varepsilon + K_D T_D \frac{d\varepsilon}{dt} + \frac{1}{T_I} \int \varepsilon dt \right\}$$

Hierfür gilt:

- K_p = Proportionalfaktor
- ε = Abweichung
- K_D = Differentialfaktor
- T_D = Differentielle Zeitkonstante
- T_I = Integrierzeitkonstante

Übersicht der Parameter (S3+) bis ((S3+)+28)

(S3+) Parameter Nr.	Bezeichnung	Beschreibung	Wertebereich	
+0	Abtastzeit	Abtastintervall für Prozess-Istwert Die Abtastzeit kann nicht kürzer sein als die Zykluszeit.	1–32767 ms	
+1	Betriebs-einstellungen	Bewegungsrichtung Bit 0: 0 = Vorwärts; 1 = Rückwärts	0 oder 1	
		Bit 1: 0/1 = Istwert-Alarmmeldung AUS/EIN		
		Bit 2: 0/1 = Ausgangswert-Alarmmeldung AUS/EIN ^①		
			Bit 3: Reserviert	—
			Bit 4: Auto-Tuning-Funktion ^② 0 = Aus; 1 = Start	0 oder 1
			Bit 5: Begrenzung des Ausgangswertes ^① 0 = deaktiviert; 1 = Aktiviert	
			Bit 6: Methode des Auto-Tuning 0 = Sprungantwort; 1: Grenzyklus	
	Bit 7–15: Reserviert	—		
+2	Eingangs-Filter (α)	Einstellwert für Eingangs-Filter Bei einem Einstellwert von 0 % ist das Eingangsfilter deaktiviert.	0–99 %	
+3	Proportional-faktor (K_p)	Multiplikationsfaktor für Proportionalregelung	1–32767 %	
+4	Integrierzeit-konstante (T_I)	Faktor für Kehrwertmultiplikation bei Integralregelung. Die Auswahl des Werts Null verhindert die Integralregelung.	0–32767 x 100 ms	
+5	Differential-faktor (K_D)	Multiplikationsfaktor für Differentialregelung Bei einem Einstellwert von 0 % ist der Differentialanteil deaktiviert.	0–100 %	
+6	Differentielle Zeitkonstante (T_D)	Multiplikationsfaktor für Differentialregelung. Die Auswahl des Werts Null verhindert die Differentialregelung.	0–32767 x 10 ms	
+7 – +19	Reserviert	—	—	
+20	Kontrollwert für Istwert-Alarmmeldung (ansteigend)	Alarmausgabe, wenn der Istwert diesen Kontrollwert übersteigt und in ((S3+)+1) das Bit 1 auf „1“ gesetzt ist.	0–32767	
+21	Kontrollwert für Istwert-Alarmmeldung (abfallend)	Alarmausgabe, wenn der Istwert diesen Kontrollwert unterschreitet und in ((S3+)+1) das Bit 1 auf „1“ gesetzt ist.	0–32767	

Tab. 7-17: Übersicht der Regelparameter

(S3+) Parameter Nr.	Bezeichnung	Beschreibung	Wertebereich
+22	Kontrollwert für Ausgangswert-Alarmmeldung (ansteigend)	Alarmausgabe, wenn der Ausgangswert diesen Kontrollwert übersteigt und in ((S3+)+1) das Bit 2 auf „1“ und das Bit 5 auf „0“ gesetzt ist.	0–32767
	Obere Begrenzung des Ausgangswertes	Vom Anwender festgelegte obere Begrenzung des Ausgangswertes (D+). (Aktiv, wenn in ((S3+)+1) das Bit 2 auf „0“ und das Bit 5 auf „1“ gesetzt ist.)	-32768–32767
+23	Kontrollwert für Ausgangswert-Alarmmeldung (abfallend)	Alarmausgabe, wenn der Ausgangswert diesen Kontrollwert unterschreitet und in ((S3+)+1) das Bit 2 auf „1“ und das Bit 5 auf „0“ gesetzt ist.	0–32767
	Untere Begrenzung der Ausgangswertes	Vom Anwender festgelegte untere Begrenzung des Ausgangswertes (D+) (Aktiv, wenn in ((S3+)+1) das Bit 2 auf „0“ und das Bit 5 auf „1“ gesetzt ist.)	-32768–32767
+24	Alarmausgabe	Die Alarmausgabe ist aktiv, wenn in ((S3+)+1) das Bit 1 oder das Bit 2 auf „1“ gesetzt ist.	0: Kein Alarm 1: Alarm
		Bit 0: Istwertalarm (Überschreitung)	
		Bit 1: Istwertalarm (Unterschreitung)	
		Bit 2: Ausgangswertalarm (Überschreitung)	
Die folgenden Parameter sind erforderlich, wenn für das Auto-Tuning die Grenzyklus-Methode verwendet wird. (In diesem Fall ist in ((S3+)+1) das Bit 6 auf „1“ gesetzt.)			
+25	Hysterese für Istwert	Stellen Sie die Hysterese entsprechend der Schwankung des Istwerts (PV) ein.	—
+26	Oberer Grenzwert für Ausgangswert (ULV)	Stellen Sie den Wert ein, der maximal als Ausgangswert (MV) ausgegeben werden soll.	
+27	Unterer Grenzwert für Ausgangswert (LLV)	Stellen Sie den Wert ein, der minimal als Ausgangswert (MV) ausgegeben werden soll.	
+28	Wartezeit nach Auto-Tuning	Wartezeit vom Abschluss des Auto-Tuning bis zum Beginn der Regelung	-50–32717 %

Tab. 7-18: Übersicht der Regelparameter

- ① Bit 2 und Bit 5 dürfen nicht gleichzeitig gesetzt sein.
- ② Die Steuerungen der MELSEC FX-Familie verfügen über eine Auto-Tuning-Funktion. Diese Funktion bestimmt die Startwerte der Regelparameter K_p ((S3+)+3), T_i ((S3+)+4), T_D ((S3+)+6) und die Richtung der Regelung ((S3+)+1), Bit (0). Alle anderen Parameter müssen vom Anwender angegeben werden. Mit Setzen des Bits 4 in ((S3+)+1) wird die Auto-Tuning-Funktion aktiviert. Der Ausgangswert (MV) wird im Bereich des angegebenen Startwertes gehalten und die Antwort des zu regelnden Systems (PV) überwacht. Erreicht dieser Wert $\frac{1}{3}$ des Sollwertes (SV), wird die Auto-Tuning-Funktion abgeschaltet und Bit 4 von ((S3+)+1) zurückgesetzt.

HINWEIS Die Operanden ((S3+)+20) bis ((S3+)+24) werden belegt, wenn in ((S3+)+1) eines der Bits 1, 2 oder 5 auf „1“ gesetzt ist.

Folgende Punkte sind bei der Verwendung der Auto-Tuning-Funktion zu beachten

- Der Unterschied zwischen Ist-Wert (PV) und Soll-Wert (SV) muss 150 % betragen.
- Die Abtastzeit muss ein Vielfaches der Zykluszeit und größer 1 Sekunde sein.
- Vor dem Starten der Autotuning-Funktion muss das System stabil sein.

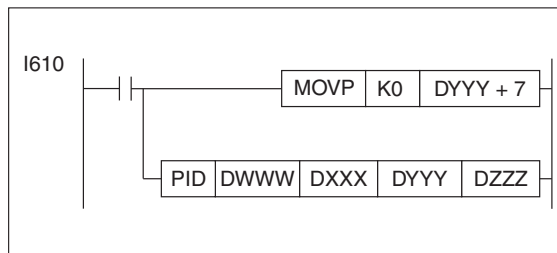
Die Parameterwerte können direkt in die Datenregister geschrieben werden. Sollen die Daten bei Abschaltung der Versorgungsspannung erhalten bleiben, müssen Sie die batter-

iegepufferten Datenregister verwenden. Eine andere Möglichkeit ist, die Parameterdaten in File-Registern abzulegen und über eine BMOV-Anweisung (Abschnitt 6.3.6) in die gewünschten Datenregister zu schreiben. Dies hat den Vorteil, dass Sie mehrere Sätze mit Parameterdaten speichern und diese über einen Programmbefehl je nach Erfordernis austauschen können.

HINWEISE

Die Zahl der PID-Anweisungen in einem Programm ist nicht beschränkt. Achten Sie jedoch darauf, dass sich die Datenbereiche (D3+) nicht überschneiden, um einen Datenkonflikt innerhalb der Steuerung zu vermeiden.

Die PID-Anweisung kann in einem Interrupt, in einem Unterprogramm, in einem STL-Programm oder in Programmsprüngen eingesetzt werden. Dabei ist der PID-Anweisung eine MOVP-Anweisung voranzustellen. In dieser MOVP-Anweisung sollte K0 auf ((S3+)+7) geschrieben werden, um möglichen Programmfehlern vorzubeugen.

**Abb. 7-46:**

Programmierbeispiel zur PID-Anweisung mit vorangestellter MOVP-Anweisung

Die Abtastzeit (T_S) sollte länger als die Programm-Zykluszeit gewählt sein, da es sonst zu Fehlern kommen kann. Ist dies nicht der Fall, wird automatisch die Abtastzeit gleich der Zykluszeit gesetzt. Bei Anwendung der Interrupt-Anweisungen I6XX bis I8XX sollte die Abtastzeit nicht kürzer als die Interrupt-Zykluszeit sein.

Die Abtastzeit (T_S) kann aufgrund der Programmbearbeitung variieren. Der maximale Bereich der Abweichung liegt bei ($T_S - (\text{Programm-Zykluszeit})$) bis ($T_S + (\text{Programm-Zykluszeit})$). Diese Abweichung kann durch Einsatz der PID-Anweisung innerhalb eines getakteten Interrupt-Programms minimiert werden.

Für den Ausgangswert (MV) sollte kein Latch-Operand angegeben werden. Falls ein Latch-Operand verwendet wird, sollte dieser im ersten Zyklus nach dem Start der SPS gelöscht werden.

Die PID-Anweisung erlaubt die Anzeige von Alarmmeldungen bei Störungen im Prozessablauf. Diese Alarmmeldungen können vom Anwender aktiviert und deaktiviert werden. Ebenso kann über anwenderdefinierte Parameter eingestellt werden, bei welchem Zustand eine Alarmmeldung erfolgen soll.

Da die Steuerungen der FX1S-Serie keine Analogwerte einlesen können, müssen der SPS die Daten für die Regelung über eine serielle Verbindung zugeführt werden.

Die PID-Anweisung beinhaltet Fehlermeldungen, die helfen, ein auftretendes Problem zu lösen. Diese Meldungen werden im Datenregister D8067 gespeichert. Tritt ein Fehler auf, wird er durch Setzen von Sondermerker M8067 angezeigt. Die folgende Tabelle enthält eine Übersicht der Fehlermeldungen und ihrer Bedeutung.

Fehlermeldung	Beschreibung	Auswirkung auf PID-Anweisung
K6705	Die PID-Anweisung verweist nicht auf Datenregister.	Die Ausführung der Anweisung wird gestoppt.
K6706	Die angegebenen Datenregister liegen außerhalb des zulässigen Bereichs.	
K6730	Die Sampling-Zeit (T_S) liegt außerhalb des zulässigen Bereichs ($T_S < 0$).	
K6732	Der Einstellwert für den Input-Filter (α) liegt außerhalb des zulässigen Bereichs ($\alpha < 0$ oder $\alpha > 100$).	
K6733	Der Proportionalfaktor (K_P) liegt außerhalb des zulässigen Bereichs ($K_P < 0$).	
K6734	Die Integrierzeitkonstante (T_I) liegt außerhalb des zulässigen Bereichs ($T_I < 0$).	
K6735	Der Differentialfaktor (K_D) liegt außerhalb des zulässigen Bereichs ($K_D < 0$ oder $K_D \geq 101$).	
K6736	Die differentielle Zeitkonstante (T_D) liegt außerhalb des zulässigen Bereichs ($T_D < 0$).	Die Abtastzeit wird gleich der Zykluszeit gesetzt, und die Ausführung wird fortgesetzt.
K6740	Abtastzeit (T_S) \leq Programm-Zykluszeit	
K6742	Die Istwertveränderung liegt außerhalb des zulässigen Bereichs (Δ Istwert < -32768 oder Δ Istwert $> +32767$).	Die betroffenen Daten werden auf den erlaubten Grenzwert herauf-, bzw. herabgesetzt, und die Ausführung wird fortgesetzt.
K6743	Die Abweichung liegt außerhalb des zulässigen Bereichs ($\varepsilon < -32768$ oder $\varepsilon > +32767$).	
K6744	Das Integrierergebnis liegt außerhalb des zulässigen Bereichs ($-32768 - +32767$).	
K6745	Der Differentialfaktor (K_D) liegt über oder der Differenzialwert außerhalb des zulässigen Bereichs.	
K6746	Das Differenzierergebnis liegt außerhalb des zulässigen Bereichs ($-32768 - +32767$).	
K6747	Das PID-Gesamtergebnis liegt außerhalb des zulässigen Bereichs ($-32768 - +32767$).	
K6750	Sollwert - Istwert < 150 oder der Regelkreis ist instabil. (Die Differenz zwischen Soll- und Istwert schwankt zu stark und zu schnell.)	Die Ausführung wird beendet.
K6751	Der Soll-Wert ist deutlich zu groß.	
K6752	Große Schwankungen bei der Auto-Tuning-Funktion	

Tab. 7-19: Übersicht der in Register D8067 gespeicherten Fehlermeldungen

Beispiel ▾ Einsatz der PID-Anweisung

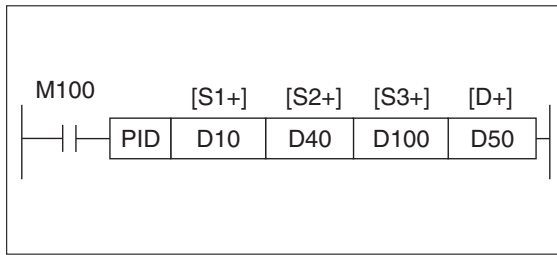


Abb. 7-47:
Programmierbeispiel zur PID-Anweisung

Die PID-Anweisung wird ausgeführt, sobald M100 gesetzt wird. Der Sollwert ist in D10 gespeichert, der Istwert wird nach D40 eingelesen, und der Ausgangswert wird nach D50 geschrieben. Die Regelparameter sind ab dem Datenregister D100 gespeichert.

Sollen analoge Werte geregelt werden, müssen dem SPS-Grundgerät zwei zusätzliche Sondermodule zur Handhabung der von der Steuerung eingelesenen und an die Steuerung auszugebenden Signale hinzugefügt werden.

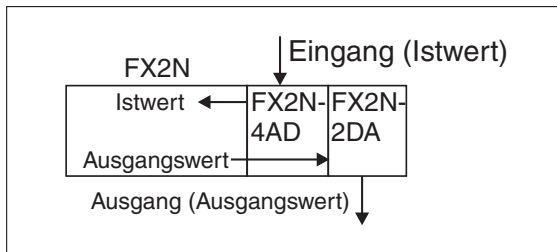


Abb. 7-48:
Beispiel für ein Analogsystem

Bei dem ersten Sondermodul handelt es sich um einen A/D-Wandler vom Typ FX2N-4AD, der die analogen Prozessdaten digitalisiert. Das Grundgerät verwendet eine FROM-Anweisung, um diese Daten in D40 abzulegen. Ein Beispiel für einen analogen Istwert wäre eine wechselnde Temperatur und damit ein wechselndes Spannungssignal.

Bei dem zweiten Sondermodul handelt es sich um einen D/A-Wandler vom Typ FX2N-2DA, der die in D50 gespeicherten digitalen Ausgangswerte analog ausgibt. Ein Beispiel für eine analoge Regelung wäre ein Stellventil, dessen Stellwert abhängig von einem Spannungspegel ist.

Die Regelparameter hängen von den spezifischen Gegebenheiten des Regelsystems ab.

Eine andere Möglichkeit für einen Regelausgangswert ist zum Beispiel die PWM-Anweisung (PWM, FNC 58), die fortlaufende Impulse ausgibt. Deren Pulsweite kann über den Ausgangswert bestimmt werden.

Einstellung der PID-Anweisung

Regelmethode	Auswahl über Datenregister			Beschreibung
	(S3+)+3(K _P)	(S3+)+4(T _I)	(S3+)+6(T _D)	
P	Anwenderwert	Auf Null gesetzt	Auf Null gesetzt	Proportionale Regelung
PI	Anwenderwert	Anwenderwert	Auf Null gesetzt	Proportionale und integrale Regelung
PD	Anwenderwert	Auf Null gesetzt	Anwenderwert	Proportionale und differenzielle Regelung
PID	Anwenderwert	Anwenderwert	Anwenderwert	Volle PID-Regelung

Tab. 7-20: *Einstellung der PID-Anweisung*

Vorwärts- und Rückwärts-Operationen ((S3+)+1, b0)

Die Bezeichnung der Vorwärts- und Rückwärts-Operation erscheint vom Namen her verwirrend. Von der Vorstellung her kommt es der Bewegung eines Punktes in einem Koordinatensystem am nächsten. Für die beiden Parameter kann man sich die folgende Situation vorstellen:

- Der Istwert (CV, Datenregister (S2+)) ist größer als der Sollwert (SP oder Datenregister (S1+)).
- Der Istwert (CV, Datenregister (S2+)) ist kleiner als der Sollwert (SP oder Datenregister (S1+)).

Die folgende Abbildung zeigt ein Koordinatensystem, in dem die Koordinatenachsen für den Sollwert, bzw. den Ausgangswert des Regelkreises, und den Istwert stehen. Die gestrichelte Linie stellt den Verlauf einer Vorwärtsbewegung und die durchgezogene Linie den Verlauf einer Rückwärtsbewegung dar.

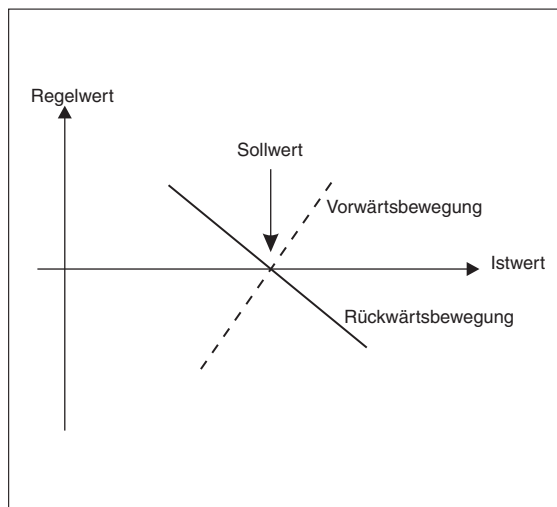


Abb. 7-49:

Darstellung der Vorwärts-/Rückwärtsbewegung in einem Koordinatensystem

Die Vorwärtsbewegung besteht bei zu großem Istwert im Zurückregeln des Istwertes, bzw. bei zu kleinem Istwert im Vorwärtsregeln des Istwertes.

Die Rückwärtsbewegung liegt vor, wenn bei zu großem Istwert der Istwert vorwärts geregelt wird, bzw. bei zu kleinem Istwert zurück geregelt wird.

In der Grafik ist kein Korrekturfaktor P, I oder D, oder eine Kombination aus diesen, berücksichtigt worden.

7.6 Datentransfer mit Index-Register

Mit den Anweisungen ZPUSH und ZPOP können die Inhalte der Index-Register V0 bis V7 und Z0 bis Z7 gesichert bzw. wiederhergestellt werden.

Symbol	FNC	Bedeutung	Abschnitt
ZPUSH	102	Inhalt der Index-Register sichern	7.6.1
ZPOP	103	Inhalt der Index-Register wiederherstellen	7.6.2

Tab. 7-21: Übersicht der Datentransferanweisungen für Index-Register

7.6.1 Inhalt der Index-Register sichern (ZPUSH)

		ZPUSH		FNC 102					
		Inhalt der Index-Register sichern							
Operanden	D+	Puls-Anweisung (P)		Verarbeitung		Programmschritte			
	D, R	●		16 Bit	32 Bit	ZPUSH	ZPUSHP	3	

Funktionsweise

Transfer der Inhalte der Index-Register in einen Datenregisterbereich

Beschreibung

- Die ZPUSH-Anweisung überträgt die Inhalte der Index-Register V0 bis V7 und Z0 bis Z7 in einen Datenregisterbereich, dessen erste Adresse mit (D+) abgegeben wird.
- Zur Übertragung der gesicherten Werte aus dem Datenregisterbereich in die Index-Register verwenden Sie bitte eine ZPOP-Anweisung (Abschnitt 7.6.2).
- Der Operand ((D+)+0) enthält einen Zählwert, der angibt, wie oft die Inhalte der Index-Register gesichert wurden. Bei jeder Ausführung der ZPUSH-Anweisung wird dieser Wert um 1 erhöht und bei jeder Ausführung der ZPOP-Anweisung um 1 vermindert.
- In die Operanden ((D+)+1) bis ((D+)+16) werden die Inhalte der Index-Register eingetragen.
- Bei der Anwendung der Anweisungen auf unterschiedlichen Ebenen (Nesting) wird für jede Ausführung der ZPUSH-Anweisung ein Raum von 16 Registern nach dem mit (D+) angegebenen Operanden benötigt. Aus diesem Grund ist ab (D+) ein den Ausführungen der ZPUSH-Anweisung angepasster Speicherbereich vorzusehen.

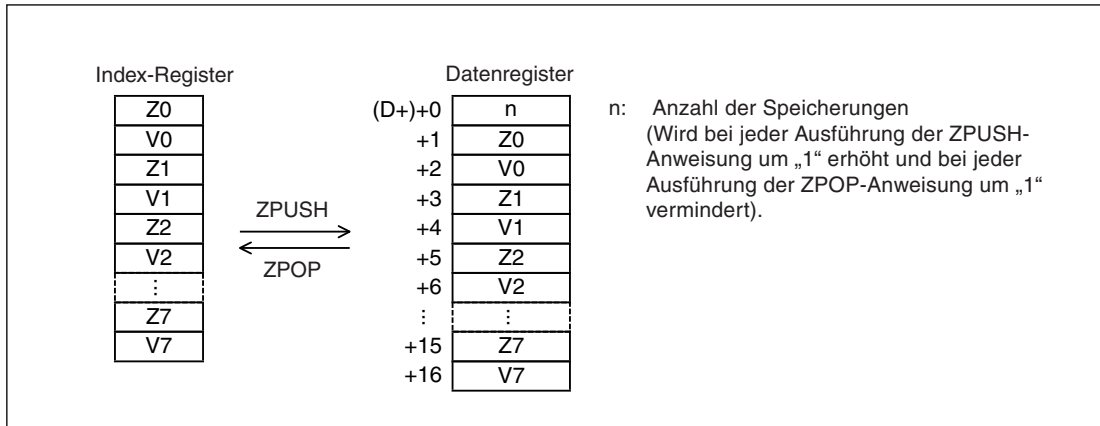


Abb. 7-51: Funktion der ZPUSH-/ZPOP-Anweisungen ohne Nesting

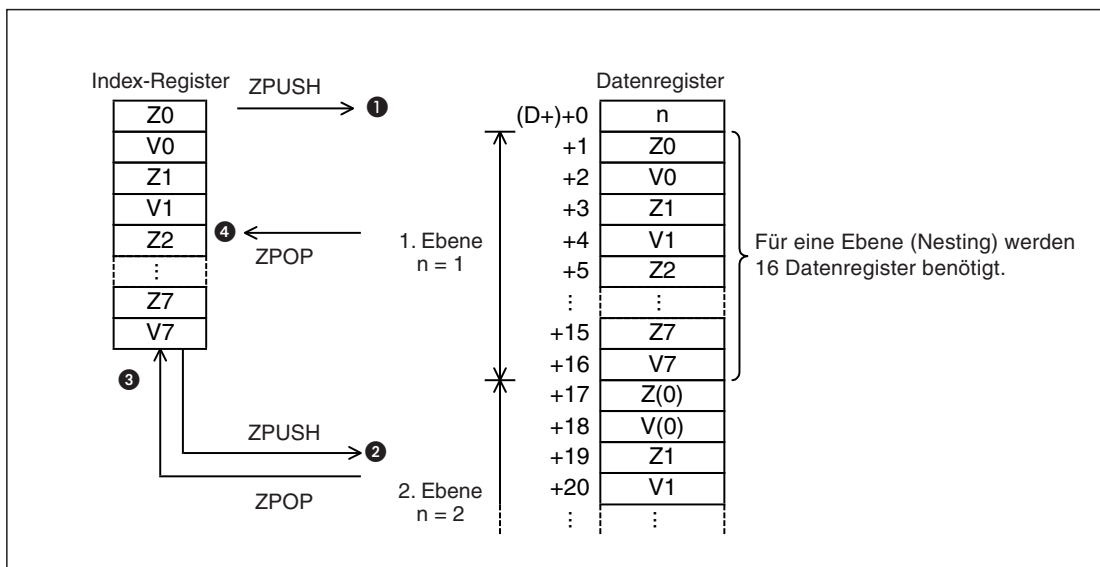


Abb. 7-50: Funktion der ZPUSH-/ZPOP-Anweisungen mit Nesting

- ① Es wird eine ZPUSH-Anweisung für die 1. Nesting-Ebene ausgeführt. In ((D+)+0) wird der Wert „1“ eingetragen.
- ② Für die 2. Nesting-Ebene wird eine ZPUSH-Anweisung ausgeführt. Die Inhalte der Index-Register werden nach den zuvor gespeicherten Daten eingetragen. Der Inhalt von ((D+)+0) wird auf den Wert „2“ erhöht.
- ③ Mit einer ZPOP-Anweisung werden die Daten der 2. Ebene wieder in die Index-Register übertragen. In ((D+)+0) wird der Wert „1“ eingetragen.
- ④ Die Daten der 1. Nesting-Ebene werden mit einer ZPOP-Anweisung aus den Datenregistern in die Index-Register übertragen. Der Inhalt von ((D+)+0) wird zu „0“.

HINWEISE

Wenn keine Nesting-Ebenen verwendet werden, sollte der Inhalt von ((D+)+0) vor dem Aufruf einer ZPUSH-Anweisung gelöscht werden.

Bei der Verwendung von Nesting-Ebenen, sollte der Inhalt von ((D+)+0) vor dem ersten Aufruf einer ZPUSH-Anweisung gelöscht werden.

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermarker M8067 wird gesetzt, und in das Sonderregister D8067 wird ein Fehlercode eingetragen:

- Die Anzahl der Operanden ab (D+) überschreitet den zulässigen Operandenbereich (Fehlercode 6706).
- Bei der Ausführung einer ZPUSH-Anweisung enthält ((D+)+0) (Zahl der Speicherungen) einen negativen Wert (Fehlercode 6707).

Beispiel ▾

Im folgenden Programm wird der Inhalt der Index-Register V0 bis V7 und Z0 bis Z7 vor dem Aufruf eines Unterprogramms in die Datenregister ab D0 eingetragen. Die Index-Register müssen vor der Ausführung des Unterprogramms gerettet werden, weil sie auch im Unterprogramm verwendet werden.

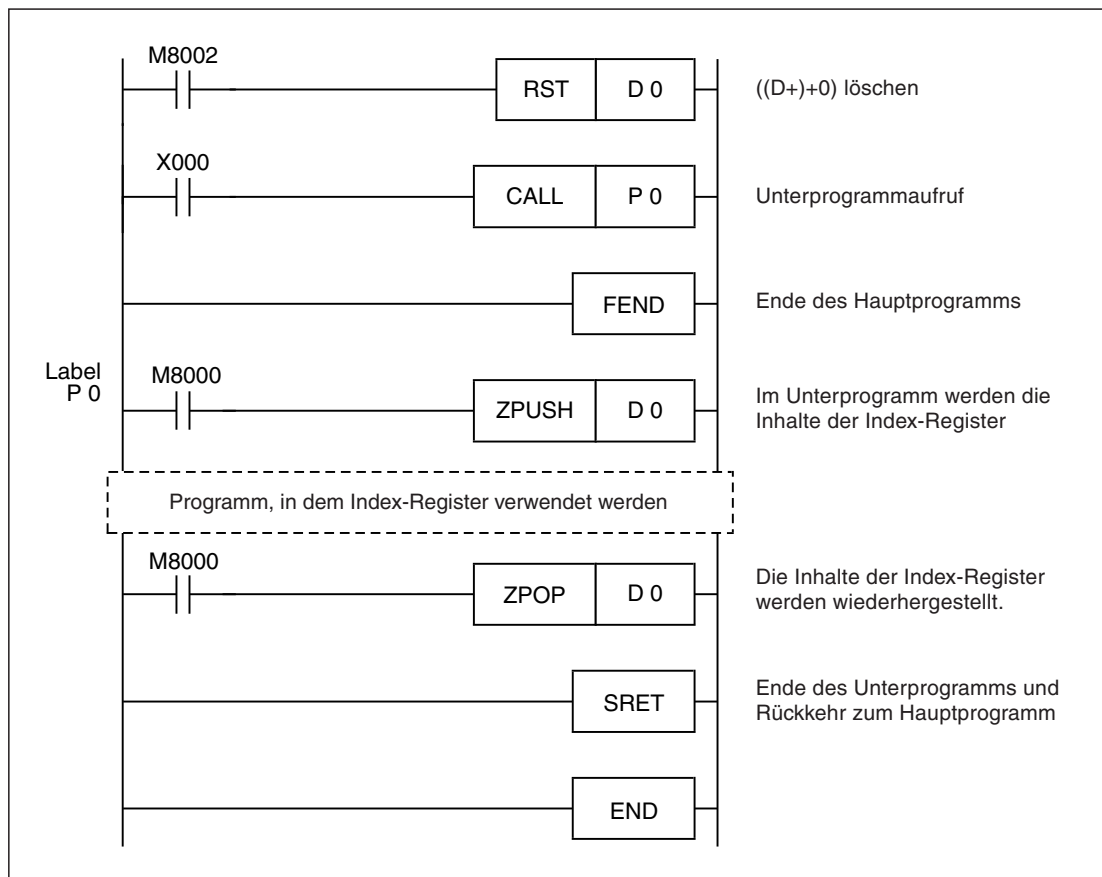


Abb. 7-52: Beispiel für die Anwendung der ZPUSH- und ZPOP-Anweisung



7.6.2 Inhalt der Index-Register wiederherstellen (ZPOP)

		ZPOP		FNC 103				
		Inhalt der Index-Register wiederherstellen						
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
								●
Operanden	D+	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	D, R	●	16 Bit	32 Bit	ZPOP	ZPOPP	3	

Funktionsweise

Transfer der durch eine ZPUSH-Anweisung gesicherten Inhalte der Index-Register aus einen Datenregisterbereich zurück in die Index-Register

Beschreibung

- Die ZPOP-Anweisung überträgt aus einen Datenregisterbereich, dessen erste Adresse mit (D+) abgegeben wird, die dort gespeicherten Inhalte der Index-Register V0 bis V7 und Z0 bis Z7 zurück in die Index-Register.
- Zur Sicherung der Inhalte der Index-Register verwenden Sie bitte eine ZPUSH-Anweisung (Abschnitt 7.6.1).
- Der Operand ((D+)+0) enthält einen Zählwert, der angibt, wie oft die Inhalte der Index-Register gesichert wurden. Bei jeder Ausführung der ZPUSH-Anweisung wird dieser Wert um 1 erhöht und bei jeder Ausführung der ZPOP-Anweisung um 1 vermindert.

Die ZPUSH- und die ZPOP-Anweisung werden paarweise verwendet. Eine Funktionsbeschreibung und ein Programmbeispiel finden Sie bei der Beschreibung der ZPUSH-Anweisung im vorherigen Abschnitt.

Fehlerquelle

In folgenden Fall tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird ein Fehlercode eingetragen:

- Bei der Ausführung einer ZPOP-Anweisung enthält ((D+)+0) (Zahl der Speicherungen) den Wert „0“ oder einen negativen Wert (Fehlercode 6706).

7.7 Anweisungen mit Gleitkommazahlen


Gleitkommazahlen können von den Steuerungen der FX1N- und FX1S-Serie nicht verarbeitet werden.

Übersicht der Anweisungen FNC 110 bis 137

Symbol	FNC	Bedeutung	Abschnitt
DECOMP	110	Vergleich von Gleitkommazahlen	7.7.1
DEZCP	111	Vergleich von Gleitkommazahlen mit einem Bereich	7.7.2
DEMOV	112	Transfer von Gleitkommazahlen	7.7.3
DESTR	116	Gleitkommazahl in Zeichenfolge wandeln	7.7.4
DEVAL	117	Zeichenfolge in Gleitkommazahlen wandeln	7.7.5
DEBCD	118	Umwandlung des Gleitkommaformats ins wissenschaftliche Zahlenformat	7.7.6
DEBIN	119	Umwandlung des wissenschaftlichen Zahlenformats ins Gleitkommaformat	7.7.7
DEADD	120	Addition von Gleitkommazahlen	7.7.8
DESUB	121	Subtraktion von Gleitkommazahlen	7.7.9
DEMUL	122	Multiplikation von Gleitkommazahlen	7.7.10
DEDIV	123	Division von Gleitkommazahlen	7.7.11
DEXP	124	Gleitkommazahl als Exponent zur Basis e	7.7.12
DLOGE	125	Logarithmus-naturalis-Berechnung	7.7.13
DLOG10	126	Berechnung des dekadischen Logarithmus	7.7.14
DESQR	127	Quadratwurzeln von Gleitkommazahlen	7.7.15
DENEG	128	Vorzeichenumkehr von Gleitkommazahlen	7.7.16
INT	129	Umwandlung des Gleitkommaformats ins Dezimal-Format	7.7.17
DSIN	130	Sinusberechnung mit Gleitkommazahlen	7.7.18
DCOS	131	Cosinusberechnung mit Gleitkommazahlen	7.7.22
DTAN	132	Tangensberechnung mit Gleitkommazahlen	7.7.20
DASIN	133	Arcussinusberechnung	7.7.21
DACOS	134	Arcuscosinusberechnung	7.7.22
DATAN	135	Arcustangensberechnung	7.7.23
DRAD	136	Umrechnung von Grad in Radiant	7.7.24
DDEG	137	Umrechnung von Radiant in Grad	7.7.25

Tab. 7-22: Übersicht der Anweisungen für Gleitkommazahlen

7.7.1 Vergleich von Gleitkommazahlen (DECMP)

				DECMP		FNC 110				
				Vergleich von Gleitkommazahlen						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
							●	●	●	●
Operanden	S1+	S2+	D+	Puls-Anweisung (P) ●	Verarbeitung		Programmschritte			
	K, H (Integer-Werte werden automatisch in Gleitkommazahlen konvertiert), Gleitkommakonstante ^① , D, R ^② , U□\G□ ^③ (Gleitkommazahl (32 Bits))				Y, M, S, D□.b ^③ Es werden drei aufeinanderfolgende Adressen des Operanden verwendet	16 Bit	32 Bit	DECMP	13	
							●	DECMP	13	

- ① Nur bei FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC
- ② Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ③ Nur bei FX3U und FX3UC

Funktion

Vergleichen von zwei Gleitkommazahlen mit Ausgabe der Vergleichsergebnisse

Beschreibung

- Die DECMP-Anweisung vergleicht die Gleitkommazahl ab (S1+) mit der Gleitkommazahl ab (S2+).
- Die Vergleichsergebnisse werden in 3 aufeinanderfolgenden Operanden gespeichert.
- Ist die Zahl ab (S2+) kleiner als die Zahl ab (S1+), wird der Bit-Operand (D+) gesetzt.
- Ist die Zahl ab (S2+) gleich der Zahl ab (S1+), wird der Bit-Operand ((D+)+1) gesetzt.
- Ist die Zahl ab (S2+) größer als die Zahl ab (S1+), wird der Bit-Operand ((D+)+2) gesetzt.

HINWEISE

- Die angesprochenen Ausgangsoperanden bleiben nach Abschalten der Ausführungsbedingung der DECMP-Anweisung gesetzt.
- Die Vergleiche werden algebraisch ausgeführt. (Zum Beispiel wird $-1,79 \times 10^{27}$ kleiner als $9,43 \times 10^{-15}$ erkannt.)

Beispiel

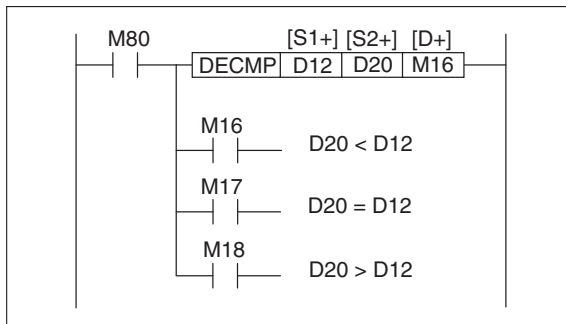


Abb. 7-53
Programmierbeispiel zur DECMP-Anweisung

Mit Setzen des Merkers M80 wird die ab D20 (S2+) angegebene Gleitkommazahl mit der ab D12 (S1+) angegebenen Gleitkommazahl verglichen.
 Ist die Zahl ab D20 kleiner als die Zahl ab D12, wird der Merker M16 gesetzt.
 Ist die Zahl ab D20 gleich der Zahl ab D12, wird der Merker M17 gesetzt.
 Ist die Zahl ab D20 größer als die Zahl ab D12, wird der Merker M18 gesetzt.



7.7.2 Vergleich von Gleitkommazahlen mit einem Bereich (DEZCP)

					DEZCP		FNC 111			
					Vergleich von Gleitkommazahlen mit einem Bereich					
CPU					FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
Operanden S1+ S2+ S3+ D+ K, H (Integer-Werte werden automatisch in Gleitkomma- zahlen konvertiert) Gleitkommakonstante*, D, R*, U□\G□* (Gleitkommazahl (32 Bits)) Y, M, S, D□.b* Es werden drei aufeinanderfolgen- de Adressen des Operanden ver- wendet					Puls-Anweisung (P)		Verarbeitung		Programmschritte	
					●		16 Bit	32 Bit	DEZCP	13
					●		●	DEZCPP	13	

* Nur bei FX3U und FX3UC

Funktion

Vergleich einer Gleitkommazahl mit einem Bereich mit Ausgabe der Vergleichsergebnisse

Beschreibung

- Die DEZCP-Anweisung vergleicht die Gleitkommazahl ab (S3+) mit dem Bereich zwischen (S1+) und (S2+).
- Die Vergleichsergebnisse werden in jeweils 3 aufeinanderfolgenden Operanden gespeichert.
- Ist die Zahl ab (S3+) kleiner als die Zahlen zwischen (S1+) und (S2+), wird der Bit-Operand (D+) gesetzt.
- Ist die Zahl ab (S3+) gleich einer Zahl zwischen (S1+) und (S2+), wird der Bit-Operand ((D+)+1) gesetzt.
- Ist die Zahl ab (S3+) größer als die Zahlen zwischen (S1+) und (S2+), wird der Bit-Operand ((D+)+2) gesetzt.

HINWEISE

Die angesprochenen Ausgangsoperanden bleiben nach Abschalten der Ausführungsbedingung der DEZCP-Anweisung gesetzt.

Die Vergleiche werden algebraisch ausgeführt. (Z. B wird $-1,79 \times 10^{27}$ kleiner als $9,43 \times 10^{-15}$ erkannt.)

Beispiel ▾

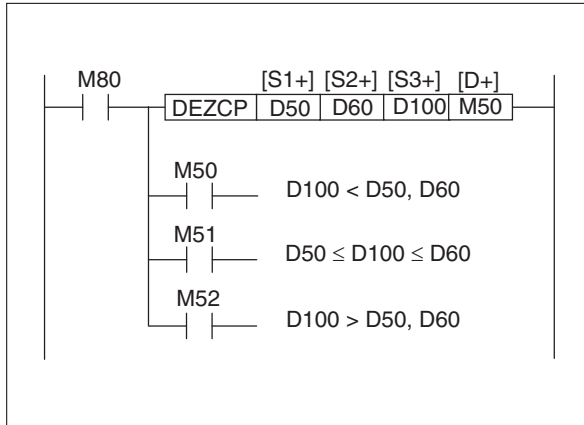


Abb. 7-54:
 Programmierbeispiel zur DEZCP-
 Anweisung

Mit Setzen des Merkers M80 wird die ab D100 (S3+) angegebene Gleitkommazahl mit den Zahlen im Bereich zwischen D50 (S1+) und D60 (S2+) verglichen.

Ist die Zahl ab D100 kleiner als die Zahlen zwischen D50 und D60, wird Merker M50 gesetzt.

Ist die Zahl ab D100 gleich einer Zahl zwischen D50 und D60, wird der Merker M51 gesetzt.

Ist die Zahl ab D100 größer als die Zahlen zwischen D50 und D60, wird Merker M52 gesetzt.

△

7.7.3 Transfer von Gleitkommazahlen (DEMOV)

		DEMOV		FNC 112			
		Transfer von Gleitkommazahlen					
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte	
	D, R, U□/G□*, Gleitkommakonstante	D, R, U□/G□*	●	16 Bit	32 Bit	DMOV	9
Es werden 2 aufeinanderfolgende Adressen der Operanden verwendet.				●		DMOV P	9

* Nur bei FX3U und FX3UC

Funktion

Übertragen einer Gleitkommazahl von einer Datenquelle zu einem Datenziel

Beschreibung

- Bei der Ausführung einer DEMOV-Anweisung wird der Inhalt von (S+) und ((S+)+1) nach (D+) und ((D+)+1) übertragen.

HINWEIS

Eine DMOV-Anweisung wird in jedem Programmzyklus ausgeführt. Eine DMOV P-Anweisung wird nur bei der steigenden Flanke der Eingangsverknüpfung ausgeführt.

Beispiel ▾

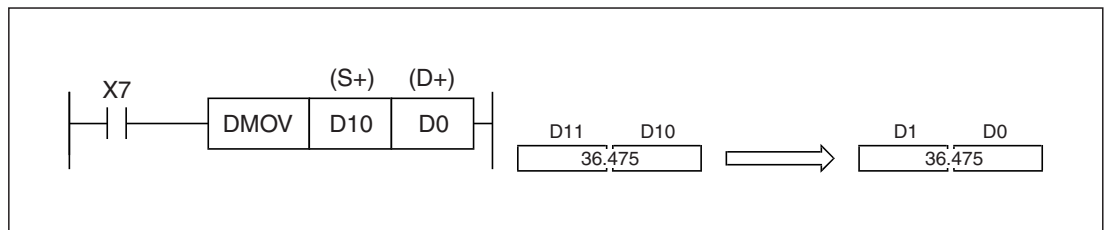


Abb. 7-55: Wenn in diesem Beispiel der Eingang X7 eingeschaltet ist, wird der Inhalt von D10 und D11 nach D0 und D1 transferiert.

△

Beispiel ▾

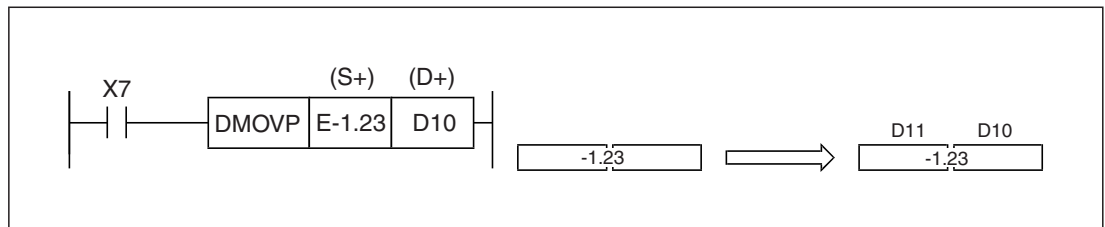


Abb. 7-56: Nur beim Einschalten von X7 wird in diesem Beispiel die Konstante „-1.23“ in die Datenregister D10 und D11 eingetragen.

7.7.4 Gleitkommazahl in Zeichenfolge wandeln (DESTR)

				DESTR		FNC 116				
				Vergleich von Gleitkommazahlen						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE		FX3U FX3UC
Operanden	S1+	S2+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	D, R, U□/G□, Gleitkommakonstante	KnX, KnY, KnM, KnS, T, C, D, R, U□/G□	KnY, KnM, KnS, T, C, D, R, U□/G□		●	16 Bit	32 Bit	DESTR	13	
						●	DESTRP	13		

Funktion

Wandlung einer Gleitkommazahl in eine Zeichenfolge (ASCII-Code) mit einer bestimmten Anzahl von Zeichen

Beschreibung

- Eine in (S1+) und ((S+)+1) gespeicherte Gleitkommazahl oder eine Gleitkommakonstante wird entsprechend den Angaben in (S2+)+0) bis (S2+)+2) in eine Zeichenfolge im ASCII-Code gewandelt und ab dem Operanden (D+) gespeichert.
- In (S2+)+0) wird das Format der Zeichenfolge angegeben:
 (S2+)+0) = 0: Dezimaldarstellung
 (S2+)+0) = 1: Exponentialdarstellung
- (S2+)+1) enthält die Anzahl der Zeichen, die das Wandlungsergebnis haben soll. Es können 2 bis 24 Zeichen angegeben werden.
- In (S2+)+2) wird die Anzahl der Nachkommastellen eingetragen.

Dezimaldarstellung der Zeichenfolge

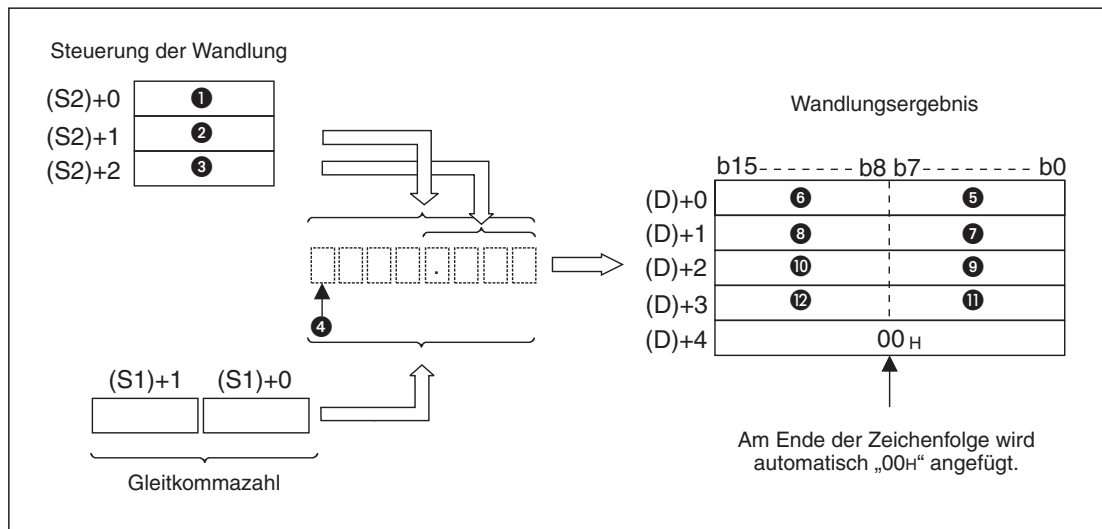


Abb. 7-57:Prinzip der Wandlung einer Gleitkommazahl in eine Zeichenfolge bei dezimaler Darstellung des Ergebnisses

- ① Darstellungsformat; bei Dezimaldarstellung muss (S2+)+0) den Wert „0“ enthalten.
- ② Gesamte Anzahl der Stellen
- ③ Anzahl der Nachkommastellen
- ④ Vorzeichen

- ⑤ ASCII-Code des Vorzeichens
- ⑥ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 1“
- ⑦ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 2“
- ⑧ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 3“
- ⑨ ASCII-Code des Dezimalpunkts (2EH)
- ⑩ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 5“
- ⑪ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 6“
- ⑫ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 7“

Die gesamte Anzahl der Stellen wird in $(S2+)$ +1) angegeben:

- Das Wandlungsergebnis kann maximal 24 Zeichen enthalten.
- Wenn die Anzahl der Nachkommastellen Null beträgt, ist die Anzahl aller Stellen ≥ 2 .
- Ist die Anzahl der Nachkommastellen nicht „0“, ist die Anzahl aller Stellen „3 plus der Anzahl der Nachkommastellen“.

$(S2+)$ +2) enthält die Anzahl der Nachkommastellen:

- Die Anzahl der Nachkommastellen kann im Bereich von 0 bis 7 liegen.
Generell gilt: Anzahl der Nachkommastellen \leq (Anzahl aller Stellen minus 3).

In den Fällen, in denen die tatsächliche Anzahl der Nachkommastellen der Gleitkommazahl größer ist als die eingegebene Anzahl der Nachkommastellen der konvertierten Zahl, werden die überzähligen Stellen abgeschnitten.

Beispiel ▾

Die Zahl -1.23456 wird zu einer Zeichenfolge mit insgesamt 8 Stellen (davon 3 Nachkommastellen) konvertiert. Das Ergebnis wird ab $(D+)$ gespeichert.

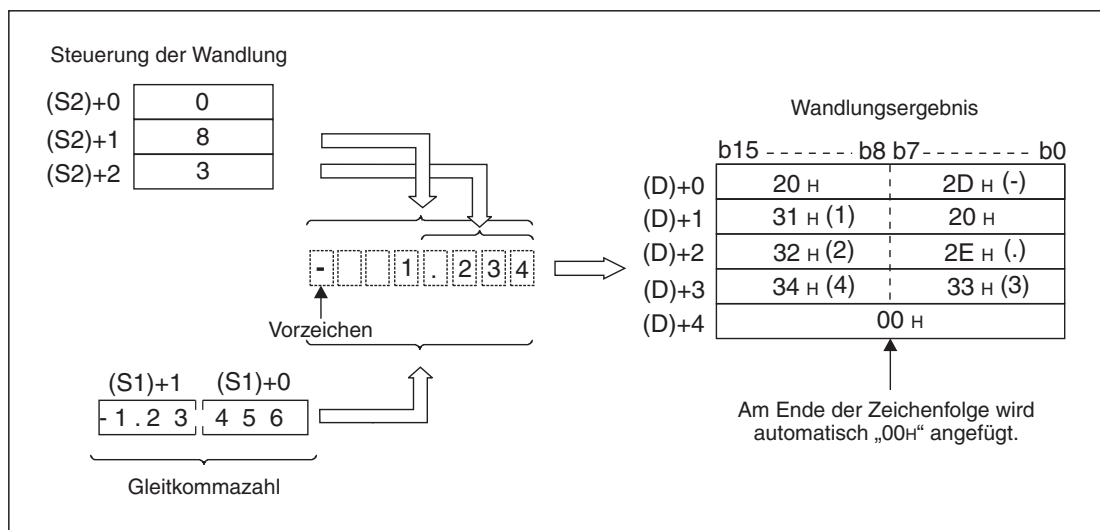


Abb. 7-58: Da die zu wandelnde Gleitkommazahl 5 Nachkommastellen, die Zeichenfolge aber nur 3 Nachkommastellen hat, gehen Stellen verloren.

Im Bereich ab $(D+)$ wird die gewandelte Zeichenfolge abgelegt: △

- Bei einer positiven Zahl wird als Vorzeichen „20H“ (Leerzeichen) in $((D+)+0)$ eingetragen. Eine negative Zahl wird durch ein Minuszeichen (2DH) gekennzeichnet.
- Nachkommastellen der Gleitkommazahl, die nicht als ASCII-Zeichen dargestellt werden können, weil die Anzahl der Nachkommastellen zu gering ist, werden gerundet.

Beispiel ▾

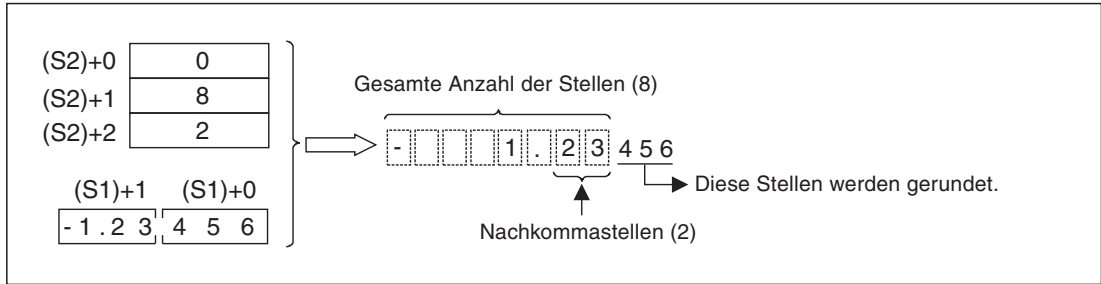


Abb. 7-59: Die restlichen Nachkommastellen erscheinen nicht in der Zeichenkette, werden aber berücksichtigt und gerundet.

△

- Wenn die Anzahl der Nachkommastellen in ((S2+)+2) auf einen anderen Wert als Null eingestellt wird, wird der ASCII-Code „2EH“ für den Dezimalpunkt automatisch an der angegebenen Stelle eingefügt. Wird die Anzahl der Nachkommastellen auf Null gesetzt, wird kein Dezimalpunkt eingefügt.
- Wenn die Anzahl aller darzustellenden Stellen (einschließlich Dezimalpunkt und Vorzeichen) kleiner ist als die angegebene gesamte Anzahl der Stellen, werden die Stellen zwischen Vorzeichen und erster darzustellender Stelle mit dem ASCII-Code „20H“ (Leerzeichen) ausgefüllt.

Beispiel ▾

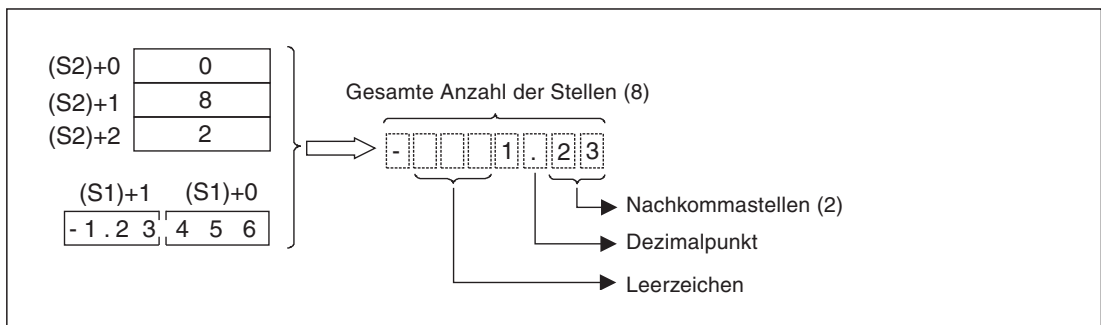


Abb. 7-60: Der Dezimalpunkt und Leerzeichen werden automatisch in die Zeichenkette eingefügt.

△

Exponentialdarstellung

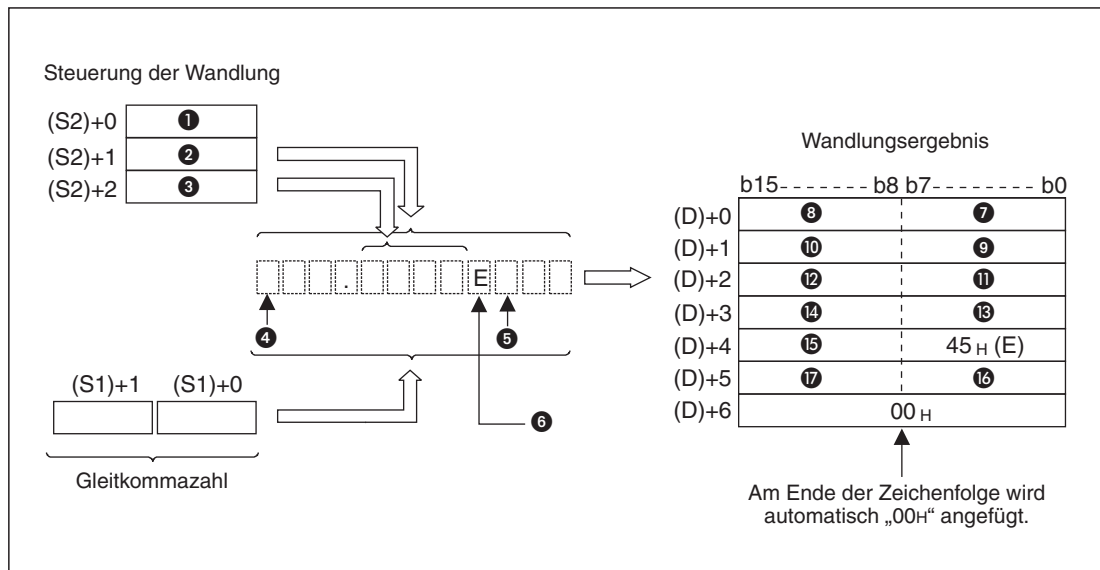


Abb. 7-61: Prinzip der Wandlung einer Gleitkommazahl in eine Zeichenfolge bei Darstellung des Ergebnisses als Exponentialzahl

- ① Darstellungsformat; bei Exponentialdarstellung muss ((S2+)+0) den Wert „1“ enthalten.
- ② Gesamte Anzahl der Stellen
- ③ Anzahl der Nachkommastellen des Dezimalteils
- ④ Vorzeichen des Integerwerts
- ⑤ Vorzeichen des Exponenten
- ⑥ Das „E“ wird automatisch eingefügt.
- ⑦ ASCII-Code des Vorzeichens des Integerwerts
- ⑧ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 1“
- ⑨ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 2“
- ⑩ ASCII-Code des Dezimalpunkts (2EH)
- ⑪ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 4“
- ⑫ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 5“
- ⑬ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 6“
- ⑭ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 7“
- ⑮ ASCII-Code des Vorzeichens des Exponenten
- ⑯ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 10“
- ⑰ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 11“

Die gesamte Anzahl der Stellen wird in $((S2+)+1)$ angegeben:

- Das Wandlungsergebnis kann maximal 24 Zeichen enthalten.
- Wenn die Anzahl der Nachkommastellen Null beträgt, ist die Anzahl aller Stellen ≥ 6 .
- Ist die Anzahl der Nachkommastellen nicht „0“, ist die Anzahl aller Stellen „7 plus der Anzahl der Nachkommastellen“.

$((S2+)+2)$ enthält die Anzahl der Nachkommastellen:

- Die Anzahl der Nachkommastellen des Dezimalteils kann im Bereich von 0 bis 7 liegen. Generell gilt: Anzahl der Nachkommastellen \leq (Anzahl aller Stellen minus 7).

Beispiel ▾

Die Zahl -12.34567 soll in exponentieller Schreibweise dargestellt werden. Die Anzahl aller Stellen beträgt 12. Davon sollen 4 Nachkommastellen im Dezimalteil dargestellt werden. Das Ergebnis wird ab (D+) gespeichert.

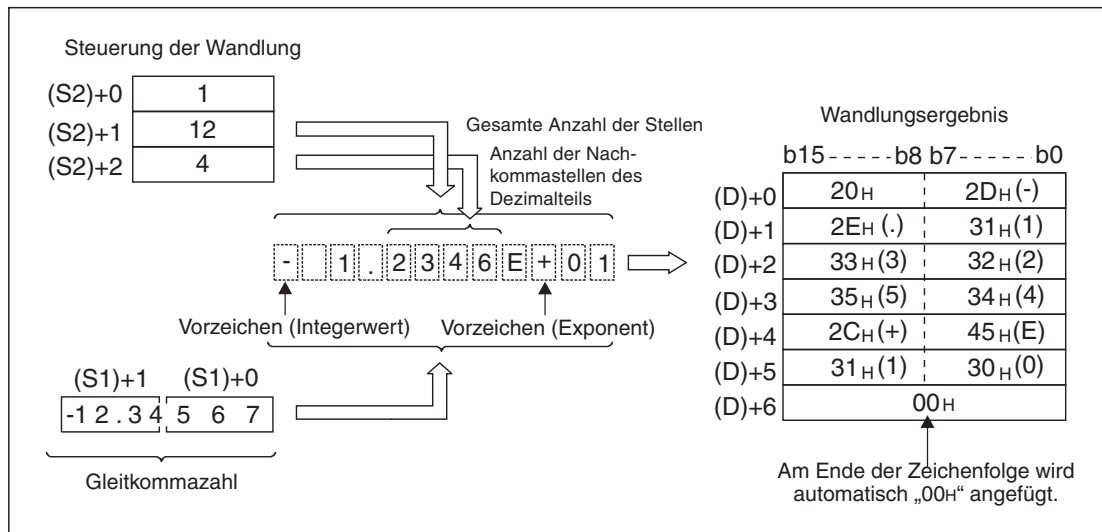


Abb. 7-62: Der Wert -12.34567 wird in exponentieller Form als -1.2345×10^1 dargestellt.



Im Bereich ab (D+) wird die gewandelte Zeichenfolge abgelegt:

- Bei einem positiven Integerwert wird als Vorzeichen „20H“ (Leerzeichen) in $((D+)+0)$ eingetragen. Eine negative Zahl wird durch ein Minuszeichen (2DH) gekennzeichnet.
- Der Integer-Wert umfasst nur eine Stelle. Zwischen dem Vorzeichen und dem Wert wird ein Leerzeichen (20H) eingefügt.

Beispiel ▾

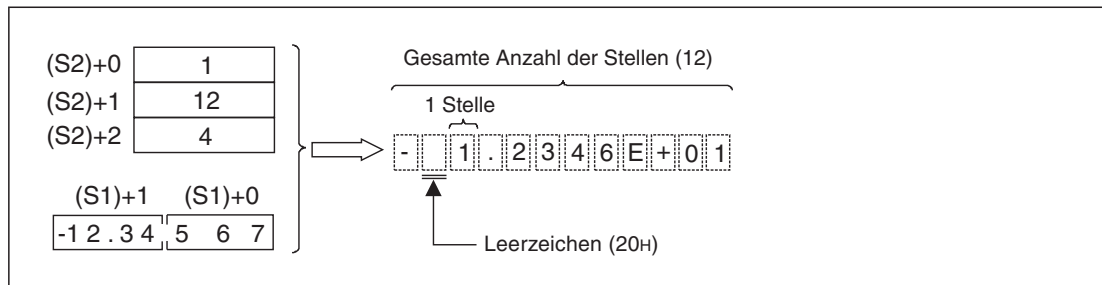


Abb. 7-63: Darstellung des Integerwerts mit Vorzeichen, Leerzeichen und einer Stelle



- Ist der Dezimalbereich der Gleitkommazahl länger als der vorgesehene Speicherbereich, werden die Stellen, die nicht gespeichert werden können, abgeschnitten.

Beispiel ▾

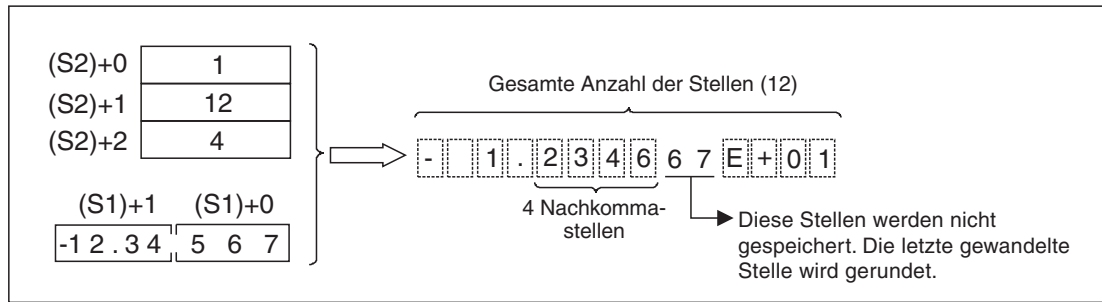


Abb. 7-64:Die restlichen Nachkommastellen erscheinen nicht in der Zeichenkette, werden aber berücksichtigt und gerundet



- Wenn die Anzahl der Nachkommastellen in ((S2+)+2) auf einen anderen Wert als Null eingestellt wird, wird der ASCII-Code „2EH“ für den Dezimalpunkt automatisch an der angegebenen Stelle eingefügt. Wird die Anzahl der Nachkommastellen auf Null gesetzt, wird kein Dezimalpunkt eingefügt.

Beispiel ▾

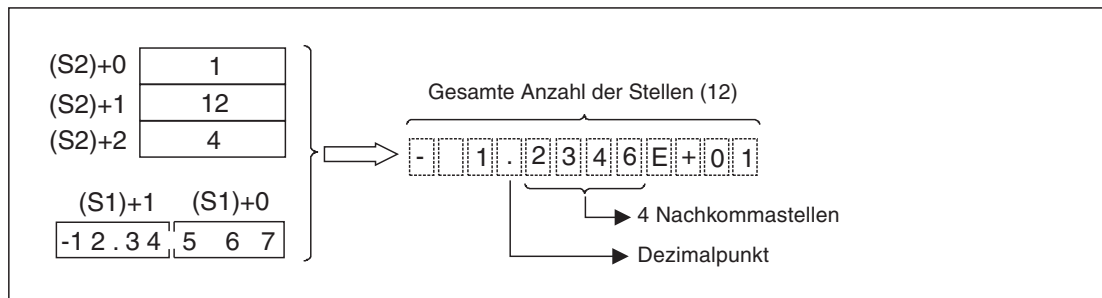


Abb. 7-65:Der Dezimalpunkt wird automatisch in die Zeichenkette eingefügt



- Der Exponent wird zweistellig dargestellt. Bei einem einstelligen Exponenten wird der ASCII-Code „30H“ (0) zwischen Exponentenvorzeichen und Exponent eingefügt.
- Als Vorzeichen des Exponenten wird der ASCII-Code „2BH“ (+) für einen positiven und der ASCII-Code „2DH“ (-) für einen negativen Exponenten eingefügt.

Beispiel ▾

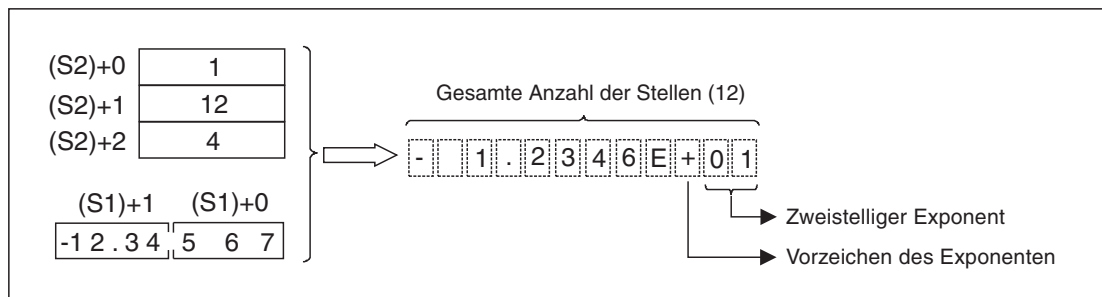


Abb. 7-66:Darstellung des Exponenten

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Der in ((S1)+0) und ((S1)+1) angegebene Wert ist nicht Null oder liegt nicht innerhalb des Wertebereichs von $\pm 2^{-127} \leq s1 < \pm 2^{129}$.
- Der Inhalt von ((S2)+0) ist weder „0“ noch „1“.
- Die Anzahl aller Stellen in ((S2)+1) liegt außerhalb der folgenden Wertebereiche:
 - Im Dezimalformat
 - ≥ 2 (Wenn als Anzahl der Nachkommastellen „0“ vorgegeben wurde)
 - Anzahl aller Stellen \geq (Anzahl der Nachkommastellen + 3)
(Wenn als Anzahl der Nachkommastellen ein anderer Wert als „0“ vorgegeben wurde)
 - Im Exponentialformat
 - ≥ 6 (Wenn als Anzahl der Nachkommastellen „0“ vorgegeben wurde)
 - Anzahl aller Stellen \geq (Anzahl der Nachkommastellen + 7)
(Wenn als Anzahl der Nachkommastellen ein anderer Wert als „0“ vorgegeben wurde)
- Die Anzahl der Nachkommastellen in ((S2)+2) liegt außerhalb der folgenden Wertebereiche:
 - Im Dezimalformat
 - Anzahl der Nachkommastellen \leq (Anzahl aller Stellen minus 3)
 - Im Exponentialformat
 - Anzahl der Nachkommastellen \leq (Anzahl aller Stellen minus 7)
- Der Speicherbereich ab (D+) überschreitet den zulässigen Bereich für diese Operanden.
- Das Wandlungsergebnis überschreitet die angegebene gesamte Anzahl von Zeichen.

Programmbeispiele

Beim Einschalten von X0 wird die Gleitkommazahl in den File-Registern R1 und R0 entsprechend den Einstellungen in R10 bis R12 in eine Zeichenfolge gewandelt und ab D0 gespeichert.

Beispiel ▾

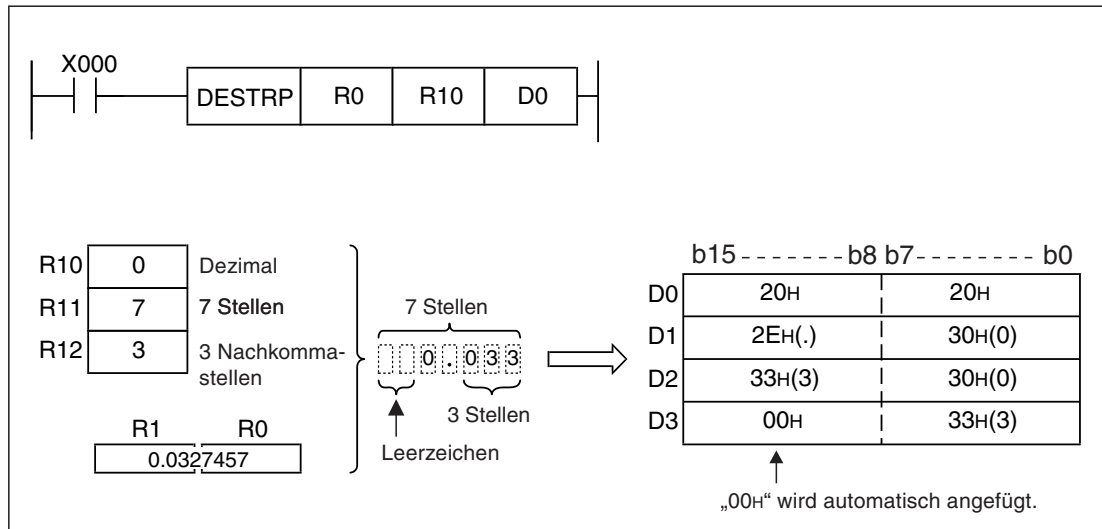


Abb. 7-67: Programmbeispiel zur Wandlung bei Dezimaldarstellung



Beispiel ▾

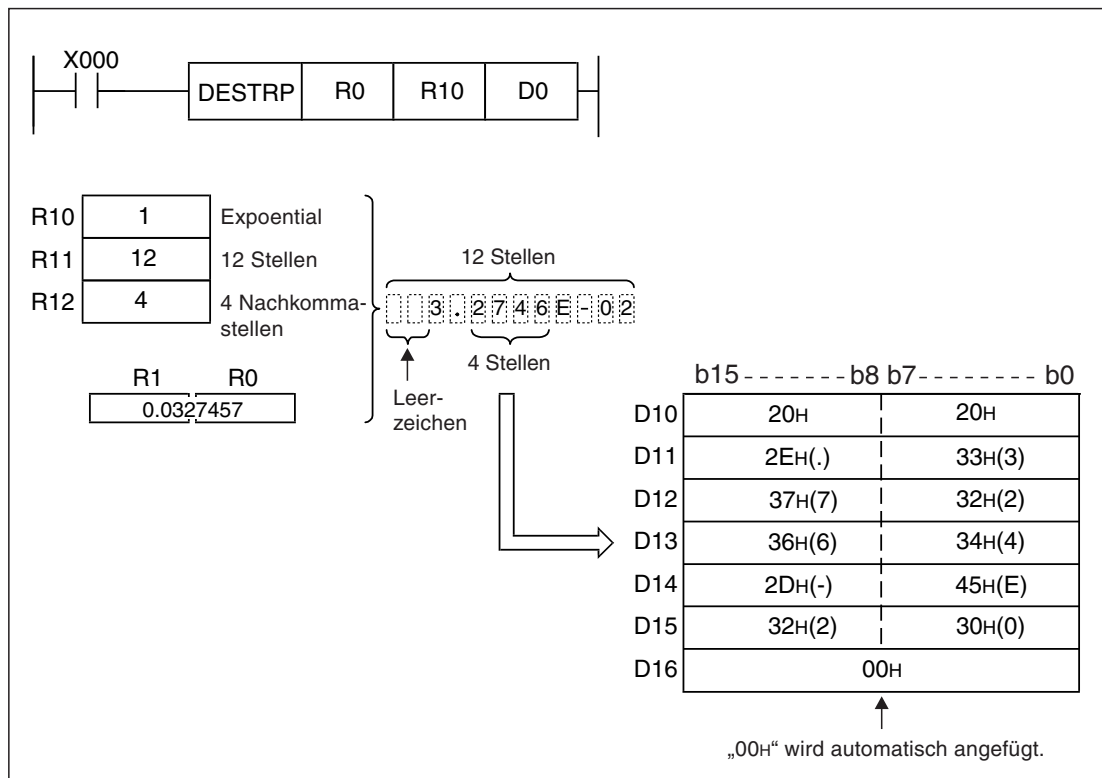


Abb. 7-68: Programmbeispiel zur Wandlung bei Darstellung mit Exponenten



7.7.5 Zeichenfolge in Gleitkommazahl wandeln (DEVAL)

		DEVAL		FNC 117						
		Umwandlung einer Zeichenfolge in eine Gleitkommazahl								
Operanden	S+	D+	Puls-Anweisung (P)	CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
										●
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte				
				16 Bit	32 Bit	DEVAL	9	DEVALP	9	

Funktion

Wandlung einer Zeichenfolge (ASCII-Code) in eine Gleitkommazahl

Beschreibung

- Eine ab (S+) gespeicherte Zeichenfolge im ASCII-Code wird in eine Gleitkommazahl gewandelt und in (D+) und ((D+)+1) gespeichert.
- Die zur Konvertierung bestimmte Zeichenfolge kann in das dezimale Gleitkommaformat oder in das Exponentialformat umgewandelt werden.

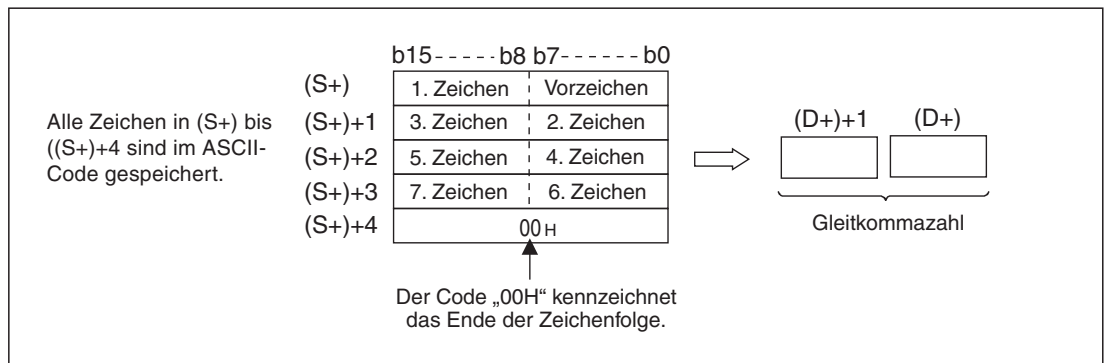


Abb. 7-69:Prinzip der Wandlung einer Zeichenfolge in eine Gleitkommazahl mit einer DVAL-Anweisung

Dezimaldarstellung

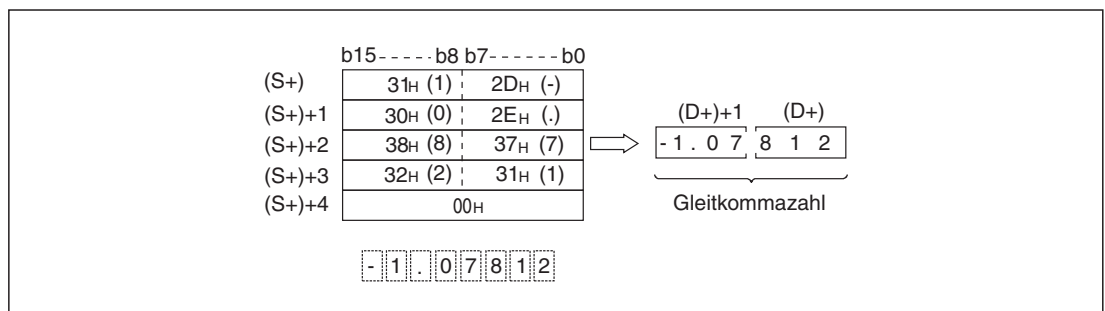


Abb. 7-70:Wandlung einer Zeichenfolge, die eine Gleitkommazahl in dezimaler Darstellung enthält

Exponentialdarstellung

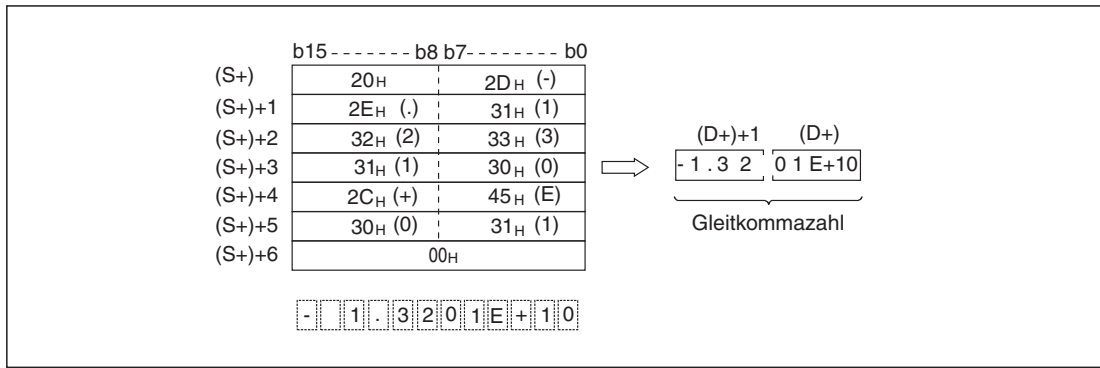


Abb. 7-71:Wandlung einer Zeichenfolge, die eine Gleitkommazahl in Exponentialdarstellung enthält

Es werden 6 Stellen (ohne Vorzeichen, Dezimalpunkt und Exponentenstellen des Resultates) der Zeichenfolge ab (S+) zu einer dezimalen Gleitkommazahl konvertiert. Ab der 7. Stelle werden die weiteren Stellen vom Resultat abgeschnitten. Dadurch können Rundungsfehler auftreten.

Beispiel ▾

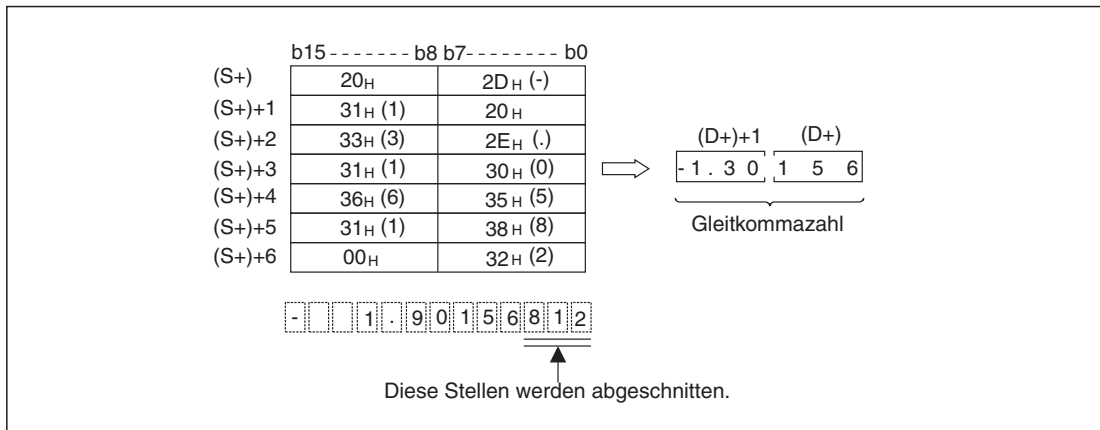


Abb. 7-72:Nach der 6. Stelle werden alle weiteren Stellen nicht mehr gewandelt (Dezimaldarstellung)



Beispiel ▾

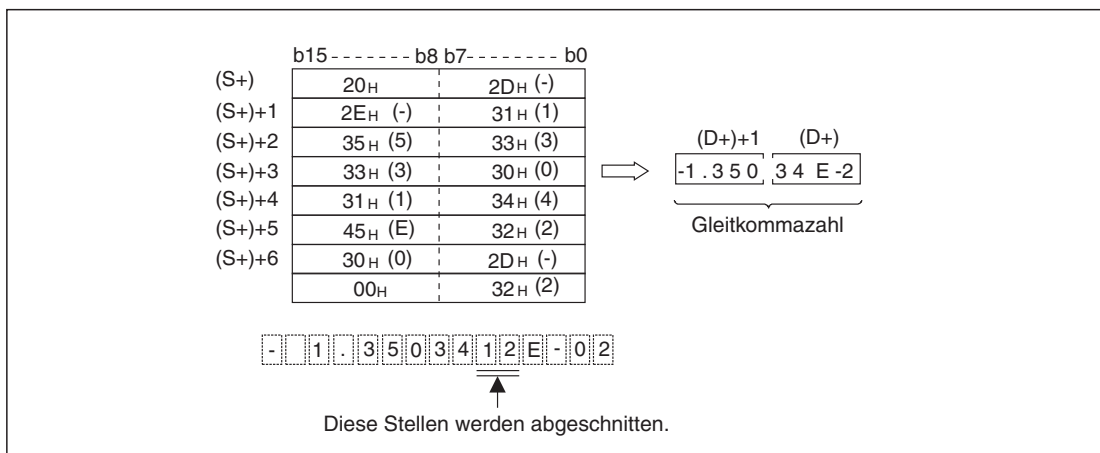


Abb. 7-73:Bei der Exponentialdarstellung werden 6 Stellen des Integer- und Dezimalteils gewandelt.



Wenn in der Zeichenfolge ab (S+) der ASCII-Code für „20H“ (Leerzeichen) oder für „30H“ (Null) vor der ersten darzustellenden Ziffer gesetzt ist, werden diese Zeichen bei der Konvertierung ignoriert.

Beispiel ▾

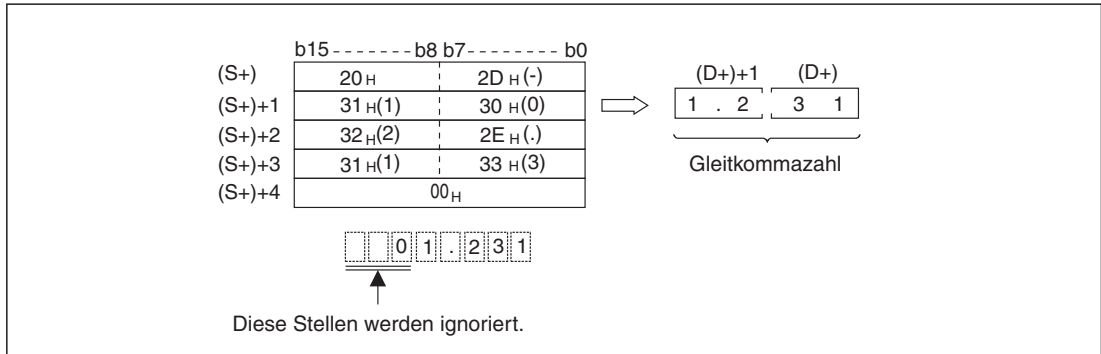


Abb. 7-74: Führende Nullen und Leerzeichen werden nicht gewandelt.



Wenn der ASCII-Code für (Null) „30H“ zwischen dem Zeichen „E“ und der Zeichenfolge für das Exponentialformat gesetzt ist, wird dieses Zeichen bei der Konvertierung ignoriert.

Beispiel ▾

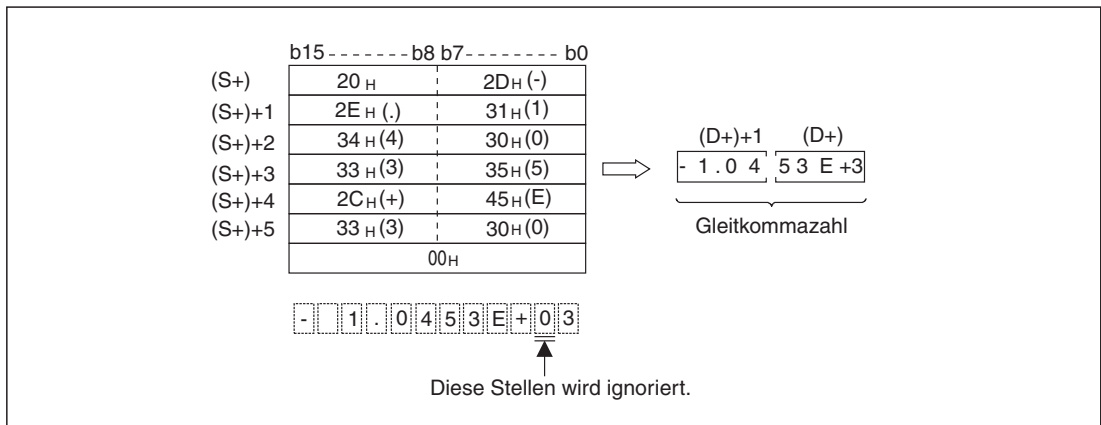


Abb. 7-75: Eine führende Null im Exponenten wird nicht gewandelt.



Eine zu konvertierende Zeichenfolge kann aus maximal 24 Zeichen bestehen.

Die folgenden Sondermerker werden von der DEVAL-Anweisung beeinflusst:

- M8020
Dieser Sondermerker hat den Signalzustand „1“, wenn das Wandlungsergebnis „0“ ist.
- M8021
Ist das Ergebnis der Wandlung kleiner als 2^{-126} wird M8021 gesetzt („1“). In diesem Fall wird als Wandlungsergebnis der minimale Wert (2^{-126}) ausgegeben.
- M8022
M8022 wird gesetzt, wenn das Ergebnis der Wandlung größer als 2^{128} ist. Das Wandlungsergebnis entspricht in diesem Fall dem maximalen Wert (2^{128}).

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Die Stellen vor dem Komma oder die Nachkommastellen enthalten andere Zeichen als die ASCII-Codes für die Ziffern 0 bis 9. („30H“ bis „39H“).
- Das Zeichen „2EH“ (.) wurde mehr als einmal in der Zeichenfolge verwendet.
- Im Exponententeil sind andere Zeichen verwendet worden als „45H“ (E), „2CH“ (+) oder „2DH“ (-), oder es ist mehr als ein Exponent vorhanden
- Im Operandenbereich ab (S+) ist die Endekennung „00H“ nicht vorhanden.
- Die Anzahl der Zeichen in der Zeichenfolge ist Null oder größer als 24.

Programmbeispiele

Beispiel ▾

Beim Einschalten von X0 wird die Zeichenfolge, die ab dem File-Register R0 gespeichert ist, in eine Gleitkommazahl gewandelt und in D1 und D0 gespeichert.

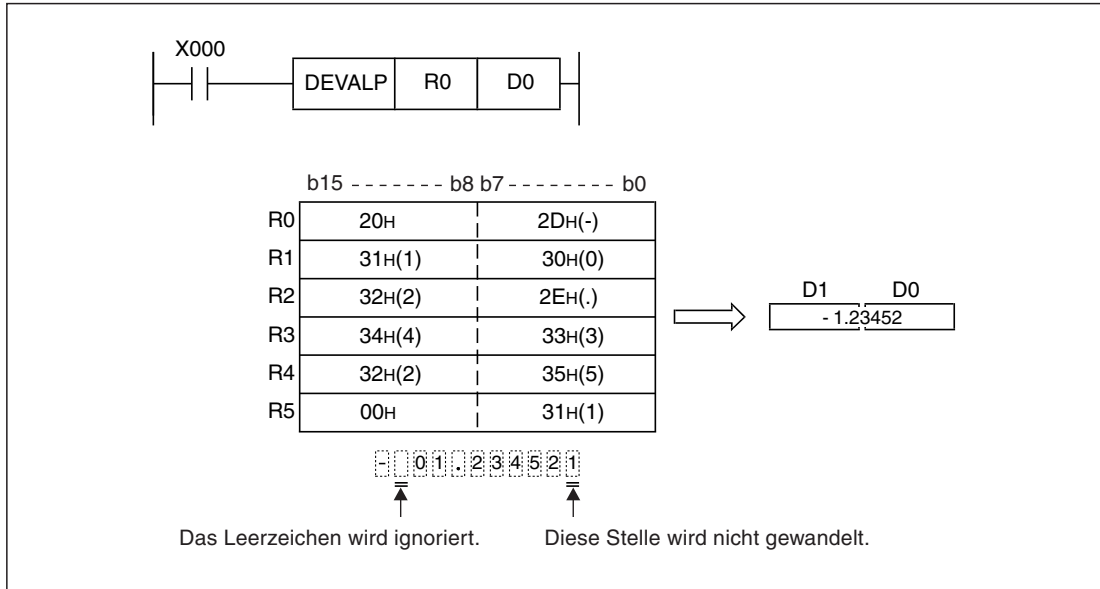


Abb. 7-76:Wandlung einer Zeichenkette in eine dezimal dargestellte Gleitkommazahl



Beispiel ▾

Beim Einschalten von X0 wird die Zeichenfolge, die ab Datenregister D10 gespeichert ist, in eine Gleitkommazahl gewandelt und in D101 und D100 gespeichert.

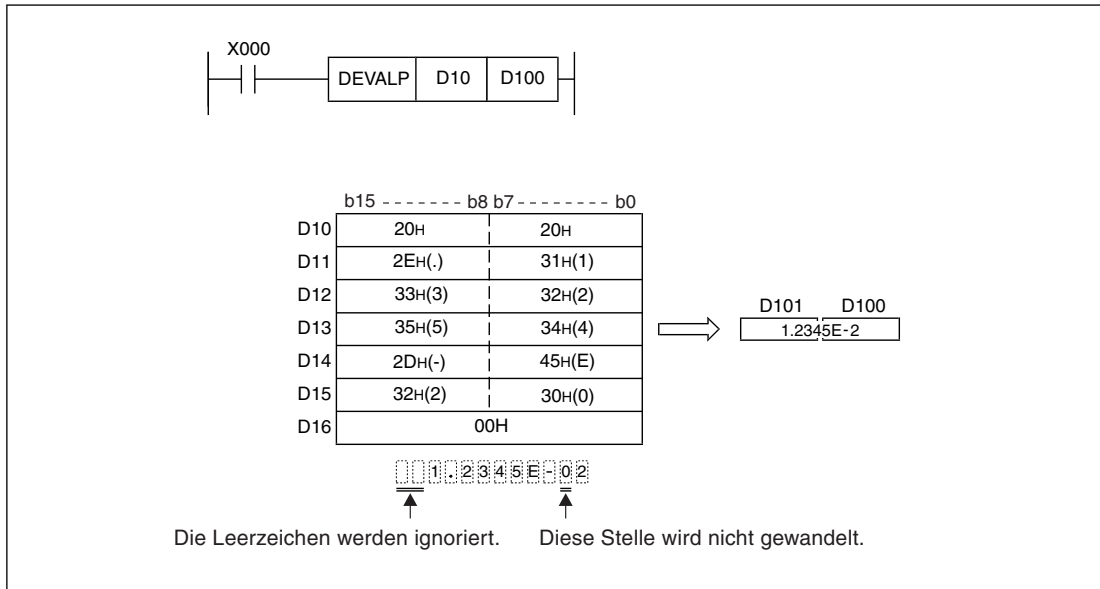


Abb. 7-77:Wandlung einer Zeichenkette in eine Gleitkommazahl mit Exponenten

7.7.6 Umwandlung des Gleitkommaformats ins wissenschaftliche Zahlenformat (DEBCD)

		DEBCD		FNC 118			
		Umwandlung vom Gleitkommaformat in das wissenschaftliche Zahlenformat					
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
			●			●	
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte	
	D, R*, U□\G□* (Gleitkommazahl (32 Bits))	D, R*, U□\G□* Es werden 2 aufeinanderfolgende Adressen der Operanden verwendet		●	16 Bit	32 Bit	DEBCD
					●	DEBCDP	9

* Nur bei FX3U und FX3UC

Funktion

Umwandeln einer Zahl im Gleitkommaformat in eine Zahl im wissenschaftlichen Zahlenformat (siehe Abschnitt 3.8.10).

Beschreibung

- Die im Gleitkommaformat angegebene Zahl ab (S+) wird in das wissenschaftliche Zahlenformat konvertiert und ab (D+) gespeichert.
- Die Mantisse wird in (D+) gespeichert.
- Der Exponent wird in ((D+)+1) gespeichert.

HINWEISE

Um das Konvertierungsergebnis mit maximaler Genauigkeit darzustellen, wird die Mantisse (D+) mit 0 oder einem Wert zwischen 1000 und 9999 angegeben. Die Angabe des Exponenten ((D+)+1) wird dementsprechend korrigiert (z. B. wird $3,4567 \times 10^{-5}$ (S+, (S+)+1) konvertiert und als 3456 (D+) und -8 ((D+)+1) gespeichert).

Beispiel ▾

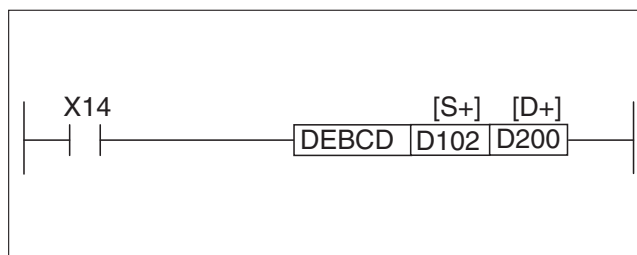


Abb. 7.78:
 Programmierbeispiel zur DEBCD-Anweisung

Mit Setzen des Eingangs X14 wird die in D102 und D103 angegebene Zahl im Gleitkommaformat in das wissenschaftliche Zahlenformat konvertiert und anschließend ab D200 gespeichert.

Die Mantisse wird in D200 gespeichert.

Der Exponent wird in D201 gespeichert.



7.7.7 Umwandlung wissenschaftliches Zahlenformat ins Gleitkommaformat (DEBIN)

		DEBIN		FNC 119			
		Umwandlung vom wissenschaftlichen Zahlenformat in das Gleitkommaformat					
CPU	FX1s	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
			●			●	
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte	
	D, R*, U□\G□* Es werden 2 aufeinanderfolgende Adressen des Operanden verwendet.	D, R*, U□\G□* (Gleitkommazahl mit 32 Bits)		●	16 Bit	32 Bit	DEBIN
					●	DEBINP	9

* Nur bei FX3U und FX3UC

Funktion

Umwandeln einer Zahl im wissenschaftlichen Zahlenformat (siehe Abschnitt 3.8.10) in eine Zahl im Gleitkommaformat

Beschreibung

- Die im wissenschaftlichen Zahlenformat angegebene Zahl ab (S+) wird in das Gleitkommaformat konvertiert und ab (D+) gespeichert.
- Die Mantisse wird in (S+) angegeben.
- Der Exponent wird in ((S+)+1) angegeben.

HINWEISE

Um das Konvertierungsergebnis mit maximaler Genauigkeit darzustellen, muss die Mantisse (S+) mit 0 oder einem Wert zwischen 1000 und 9999 angegeben werden. Die Angabe des Exponenten ((S+)+1) muss entsprechend korrigiert werden (z. B. werden die für Mantisse und Exponent angegebenen Werte 5432 (S+) und 12 ((S+)+1) in die Zahl 5,432 x 10⁹ (D+), (D+)+1) im Gleitkommaformat konvertiert).

Beispiel ▾

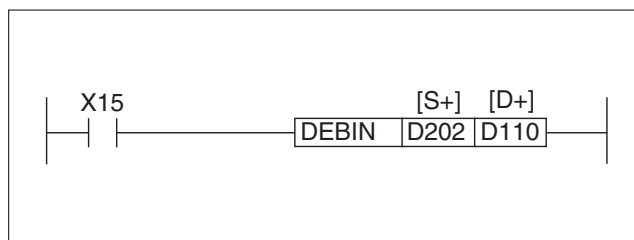


Abb. 7-79:
 Programmierbeispiel zur DEBIN-Anweisung

Mit Setzen des Eingangs X15 wird die in D202 und D203 angegebene Zahl im wissenschaftlichen Zahlenformat in das Gleitkommaformat konvertiert und anschließend ab D110 gespeichert.

Die Mantisse wird in D202 angegeben.

Der Exponent wird in D203 angegeben.



7.7.8 Addition von Gleitkommazahlen (DEADD)

				DEADD		FNC 120				
				Addition von Gleitkommazahlen						
				CPU	FX1s	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3s	FX3U FX3UC
							●	●	●	●
Operanden	S1+	S2+	D+	Puls-Anweisung (P) ●	Verarbeitung		Programmschritte			
	K, H (Integer-Werte werden automatisch in Gleitkommazahlen konvertiert.) Gleitkommakonstante ^① , D, R ^② , U□G□ ^③ (Gleitkommazahl (32 Bits))				D, R ^② , U□G□ ^③ (Gleitkommazahl (32 Bits))	16 Bit	32 Bit	DEADD	13	
							DEADDP	13		

- ① Nur bei FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC
- ② Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ③ Nur bei FX3U und FX3UC

Funktion

Addieren von zwei Gleitkommazahlen und Speichern des Ergebnisses

Beschreibung

- Die ab (S1+) angegebene Gleitkommazahl wird mit der Gleitkommazahl ab (S2+) addiert. Das Ergebnis wird ab (D+) gespeichert.
- Es werden für jeden Operanden jeweils 2 aufeinanderfolgende Register verwendet.
- Konstanten (K, H) werden vor der Addition automatisch in Gleitkommazahlen gewandelt.
- Als Quelle und Ziel kann derselbe Operand verwendet werden. In diesem Fall wird das errechnete Ergebnis wieder in dem Quelloperanden gespeichert und anschließend für die nächste Berechnung genutzt. Dieser Prozess wiederholt sich mit bei jeder Ausführung der DEADD-Anweisung.
- Ist das Additionsergebnis 0, wird das Zero-Flag M8020 gesetzt.
- Ist das Additionsergebnis größer als der maximal zulässige Wert, wird das Carry-Flag M8022 gesetzt.
- Ist das Additionsergebnis kleiner als der minimal zulässige Wert, wird das Borrow-Flag M8021 gesetzt.

HINWEIS

Die Additionen werden nach den geltenden mathematischen Regeln ausgeführt. (z. B. liefert die Addition von $2,3456 \times 10^2 + (-5,6 \times 10^{-1})$ das Ergebnis $2,34 \times 10^2$)

Beispiel ▾

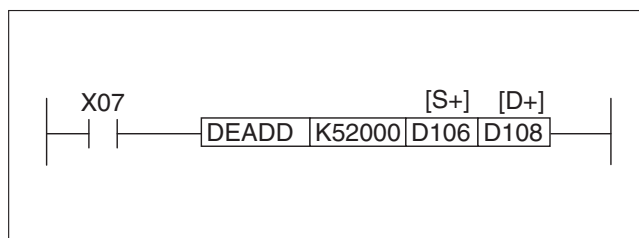


Abb. 7-80:
 Programmierbeispiel zur DEADD-Anweisung

Mit Setzen des Eingangs X7 wird zu der Konstanten K52000 die ab D106 angegebene Gleitkommazahl addiert. Das Ergebnis wird ab D108 gespeichert.



7.7.9 Subtraktion von Gleitkommazahlen (DESUB)

				DESUB		FNC 121				
				Subtraktion von Gleitkommazahlen						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
							●	●	●	●
Operanden	S1+	S2+	D+	Puls-Anweisung (P) ●	Verarbeitung		Programmschritte			
	K, H (Integer-Werte werden automatisch in Gleitkommazahlen konvertiert.) Gleitkommakonstante ^① , D, R ^② , U□G□ ^③ (Gleitkommazahl (32 Bits))				D, R ^② , U□G□ ^③ (Gleitkommazahl (32 Bits))	16 Bit	32 Bit	DESUB	13	
							●	DESUBP		13

- ① Nur bei FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC
- ② Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ③ Nur bei FX3U und FX3UC

Funktion

Subtrahieren von zwei Gleitkommazahlen und Speichern des Ergebnisses

Beschreibung

- Die ab (S2+) angegebene Gleitkommazahl wird von der Gleitkommazahl ab (S1+) subtrahiert. Das Ergebnis wird ab (D+) gespeichert.
- Es werden für jeden Operanden jeweils 2 aufeinanderfolgende Register verwendet.
- Konstanten (K, H) werden vor der Subtraktion automatisch in Gleitkommazahlen gewandelt.
- Als Quelle und Ziel kann derselbe Operand verwendet werden. In diesem Fall wird das errechnete Ergebnis wieder in dem Quelloperanden gespeichert und anschließend für die nächste Berechnung genutzt. Dieser Prozess wiederholt sich bei jeder Ausführung der DESUB-Anweisung.
- Ist das Subtraktionsergebnis 0, wird das Zero-Flag M8020 gesetzt.
- Ist das Subtraktionsergebnis größer als der maximal zulässige Wert, wird das Carry-Flag M8022 gesetzt.
- Ist das Subtraktionsergebnis kleiner als der minimal zulässige Wert, wird das Borrow-Flag M8021 gesetzt.

HINWEIS

Die Subtraktionen werden nach den geltenden mathematischen Regeln ausgeführt. (z. B. liefert die Subtraktion von $2,3456 \times 10^2 - 5,6 \times 10^{-1}$ das Ergebnis $2,34 \times 10^2$.)

Beispiel ▾

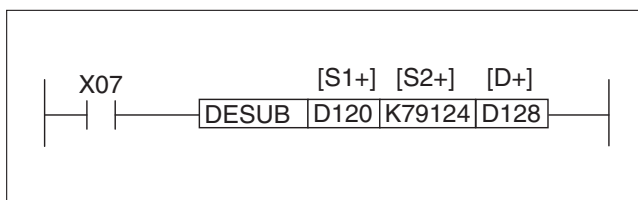


Abb. 7-81:
Programmierbeispiel zur DESUB-Anweisung

Mit Setzen des Eingangs X17 wird von der Gleitkommazahl ab D120 die Konstante K79124 subtrahiert. Das Ergebnis wird ab D128 gespeichert.



7.7.10 Multiplikation von Gleitkommazahlen (DEMUL)

				DEMUL		FNC 122				
				Multiplikation von Gleitkommazahlen						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
							●	●	●	●
Operanden	S1+	S2+	D+	Puls-Anweisung (P) ●	Verarbeitung		Programmschritte			
	K, H (Integer-Werte werden automatisch in Gleitkommazahlen konvertiert.) Gleitkommakonstante ^① , D, R ^② , U□G□ ^③ (Gleitkommazahl (32 Bits))				D, R ^② , U□G□ ^③ (Gleitkommazahl (32 Bits))	16 Bit	32 Bit	DEMUL	13	
							DEMULP	13		

- ① Nur bei FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC
- ② Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ③ Nur bei FX3U und FX3UC

Funktion

Multiplizieren von zwei Gleitkommazahlen und Speichern des Ergebnisses

Beschreibung

- Die ab (S1+) angegebene Gleitkommazahl wird mit der Gleitkommazahl ab (S2+) multipliziert. Das Ergebnis wird ab (D+) gespeichert.
- Es werden für jeden Operanden jeweils 2 aufeinanderfolgende Register verwendet.
- Konstanten (K, H) werden vor der Multiplikation automatisch in Gleitkommazahlen umgewandelt.
- Als Quelle und Ziel kann derselbe Operand verwendet werden. In diesem Fall wird das errechnete Ergebnis wieder in dem Quelloperanden gespeichert und anschließend für die nächste Berechnung genutzt. Dieser Prozess wiederholt sich jeder Ausführung der DMUL-Anweisung.

HINWEIS | Die Multiplikationen werden nach den geltenden mathematischen Regeln ausgeführt.

Beispiel ▾

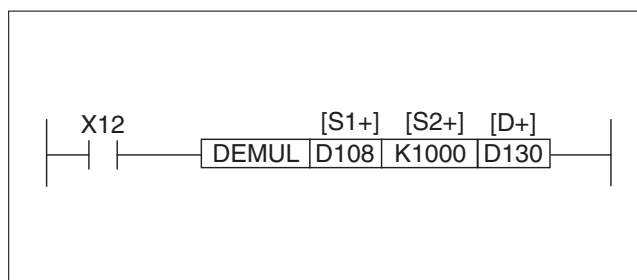


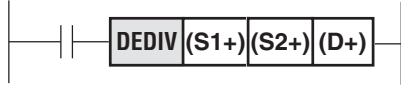
Abb. 7-82:
Programmierbeispiel zur DEMUL-Anweisung

Mit Setzen des Merkers M12 wird die Gleitkommazahl ab D108 mit der Konstanten K1000 multipliziert.

Das Ergebnis wird ab D130 gespeichert.



7.7.11 Division von Gleitkommazahlen (DEDIV)

				DEDIV		FNC 123				
				Division von Gleitkommazahlen						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
							●	●	●	●
Operanden	S1+	S2+	D+	●	Verarbeitung		Programmschritte			
	K, H (Integer-Werte werden automatisch in Gleitkommazahlen konvertiert.) Gleitkommakonstante ^① , D, R ^② , U□G□ ^③ (Gleitkommazahl (32 Bits))				D, R ^② , U□G□ ^③ (Gleitkommazahl (32 Bits))	16 Bit	32 Bit	DEDIV	13	
							DEDIVP	13		

- ① Nur bei FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC
- ② Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ③ Nur bei FX3U und FX3UC

Funktion

Dividieren von zwei Gleitkommazahlen und Speichern des Ergebnisses

Beschreibung

- Die ab (S1+) angegebene Gleitkommazahl wird durch die Gleitkommazahl ab (S2+) dividiert. Das Ergebnis wird ab (D+) gespeichert.
- Es werden für jeden Operanden jeweils 2 aufeinanderfolgende Register verwendet.
- Konstanten (K, H) werden vor der Division automatisch in Gleitkommazahlen umgewandelt.
- Als Quelle und Ziel kann derselbe Operand verwendet werden. In diesem Fall wird das errechnete Ergebnis wieder in dem Quelloperanden gespeichert und anschließend für die nächste Berechnung genutzt. Dieser Prozess wiederholt sich jeder Ausführung der DEDIV-Anweisung.

HINWEIS

Die Divisionen werden nach den geltenden mathematischen Regeln ausgeführt.

Fehlerquelle

Wenn der Wert ab (S2+) mit 0 angegeben wird, wird der Fehler „Division durch 0“ gemeldet und die Verarbeitung wird abgebrochen.

Beispiel

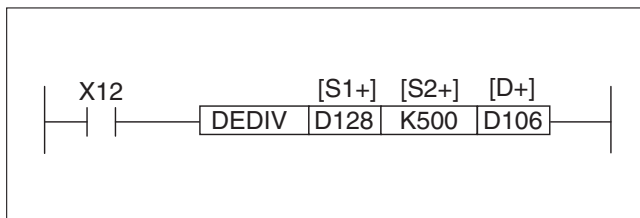


Abb. 7-83:
Programmierbeispiel zur DEDIV-Anweisung

Mit Setzen des Eingangs X10 wird die Gleitkommazahl ab D128 durch die Konstante K500 dividiert.

Das Ergebnis wird ab D106 gespeichert.

7.7.12 Gleitkommazahl als Exponent zur Basis e (DEXP)

			DEXP		FNC 124		
			Gleitkommazahl als Exponent zur Basis e				
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte	
	D, R, U□/G□ Gleitkommakonstante	D, R, U□/G□	●	16 Bit	32 Bit	DEXP	9
Es werden 2 aufeinanderfolgende Adressen der Operanden verwendet.					●	DEXPP	9

Funktion

Eine Gleitkommazahl wird als Exponent zur Basis e ($e = 2.71828$) interpretiert.

Beschreibung

- Die DEXP-Anweisung führt mit der Gleitkommazahl in ((S+)+1) und (S+) die Berechnung des Exponenten zur Basis e durch und speichert das Ergebnis in ((D+)+1) und (D+).

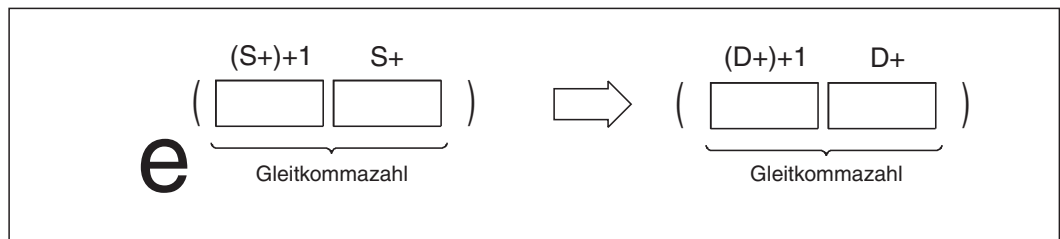


Abb. 7-84: Funktion der DEXP-Anweisung

- In ((S+)+1) und (S+) kann auch eine Gleitkommakonstante angegeben werden.

Fehlerquellen

Im folgenden Fall tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Das Berechnungsergebnis liegt nicht innerhalb des Wertebereichs von 2^{-126} und 2^{128} .

Beispiel ▾

Das folgende Programm berechnet das Ergebnis der Exponentialfunktion zur Basis e mit dem zweistelligen BCD-Wert an X20 bis X27 und speichert das Ergebnis als Gleitkommazahl in D0 und D1.

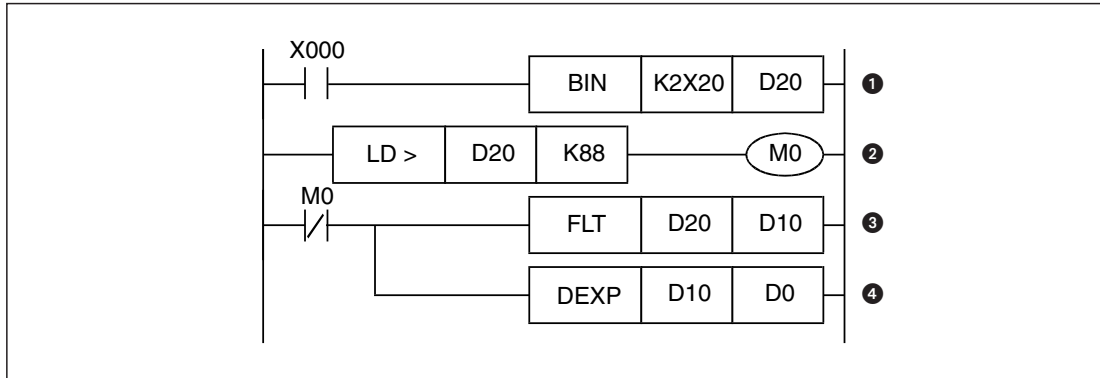


Abb. 7-85: Programmbeispiel für die Anwendung der DEXP-Anweisung

- ① Die an den Eingängen X20 bis X27 z. B. mit einem zweistelligen BCD-Schalter eingegebene BCD-Zahl wird in das binäre Format gewandelt.
- ② Der eingegebene Wert wird geprüft. Das Ergebnis der DEXP-Anweisung ist kleiner als 2^{128} , wenn die BCD-Zahl kleiner oder gleich „88“ ist. (In $2^{128} = 88,7$). Wird ein größerer Wert als 88 eingegeben, wird M0 gesetzt und dadurch die DEXP-Anweisung nicht ausgeführt.
- ③ Der eingegebene Wert wird in eine Gleitkommazahl gewandelt.
- ④ Die DEXP-Anweisung wird ausgeführt und der eingegebene Wert als Exponent zur Basis e verwendet.

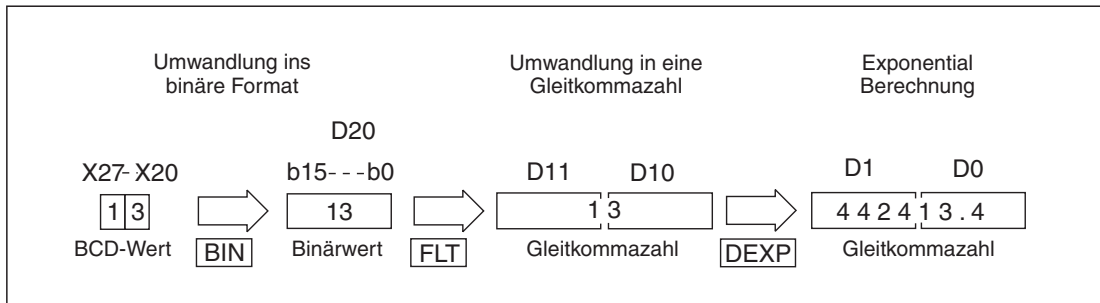


Abb. 7-86: Verarbeitung der Daten, wenn beim oben abgebildeten Beispiel an den BCD-Schaltern der Wert „13“ eingegeben wird.



7.7.13 Logarithmus-naturalis-Berechnung (DLOGE)

			DLOGE		FNC 125		
			Logarithmus-naturalis-Berechnung				
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	●
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte	
	D, R, U□/G□ Gleitkommakonstante	D, R, U□/G□	●	16 Bit	32 Bit	DLOGE	9
Es werden 2 aufeinanderfolgende Adressen der Operanden verwendet.					●	DLOGEP	9

Funktion

Logarithmus-naturalis-Berechnung (natürlicher Logarithmus mit der Basis e, abgekürzt „ln“) mit einer Gleitkommazahl.

Beschreibung

- Die DLOGE-Anweisung berechnet den Logarithmus naturalis aus der Gleitkommazahl in ((S+)+1) und (S+) und speichert das Ergebnis in ((D+)+1) und (D+).

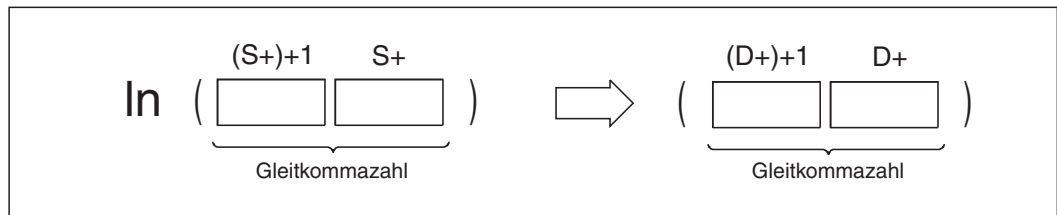


Abb. 7-87: Funktionsweise der DLOGE-Anweisung

- In ((S+)+1) und (S+) können nur positive Werte eingegeben werden. Die Berechnung des natürlichen Logarithmus kann nicht mit negativen Werten ausgeführt werden.

HINWEISE

Zusammenhang zwischen dem natürlichen und dekadischen Logarithmus:

Zur Umrechnung vom dekadischen Logarithmus mit der Basis 10 kann folgende Formel verwendet werden:

$$10^x = e^{\frac{x}{0,4342945}}$$

Mit der DLOG10-Anweisung (siehe Abschnitt 7.7.14) kann der dekadische Logarithmus einer Gleitkommazahl direkt berechnet werden.

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- ((S+)+1) und (S+) enthält einen negativen Wert.
- ((S+)+1) und (S+) enthält den Wert „0“.

Beispiel ▾

Das folgende Programm berechnet den Logarithmus naturalis des Werts „10“ und speichert das Ergebnis in D30 und D31, wenn der Eingang X0 eingeschaltet ist.

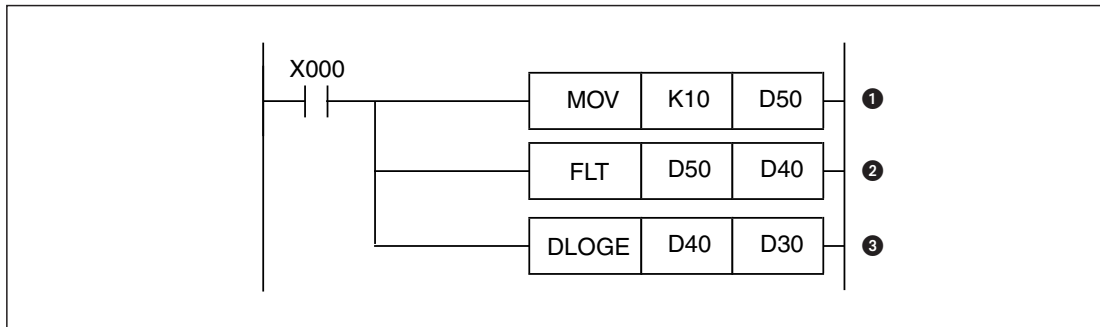


Abb. 7-88: Programmbeispiel für die Anwendung der DLOGE-Anweisung

- ❶ Die Konstante „10“ wird in das Datenregister D50 übertragen.
- ❷ Die Konstante wird in eine Gleitkommazahl gewandelt.
- ❸ Berechnung des natürlichen Logarithmus der Konstanten.

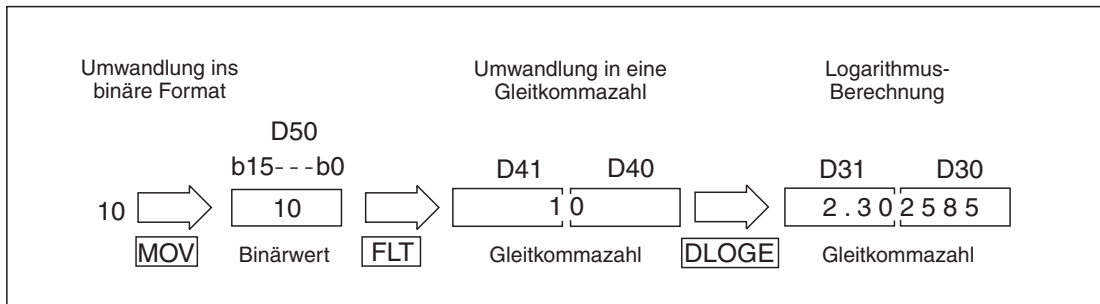
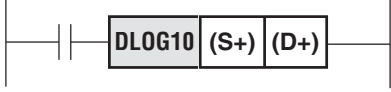


Abb. 7-89: Verarbeitung der Daten bei diesem Programmbeispiel



7.7.14 Berechnung des dekadischen Logarithmus (DLOG10)

			DLOG10		FNC 126			
			Berechnung des dekadischen Logarithmus					
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	D, R, U□/G□ Gleitkommakonstante	D, R, U□/G□	●	16 Bit	32 Bit	DLOG10	9	
Es werden 2 aufeinanderfolgende Adressen der Operanden verwendet.			●		●	DLOG10P	9	

Funktion

Berechnung des dekadischen Logarithmus (Logarithmus mit der Basis 10, abgekürzt „lg“) aus einer Gleitkommazahl.

Beschreibung

- Die DLOG10-Anweisung berechnet den dekadischen Logarithmus aus der Gleitkommazahl in ((S+)+1) und (S+) und speichert das Ergebnis in ((D+)+1) und (D+).

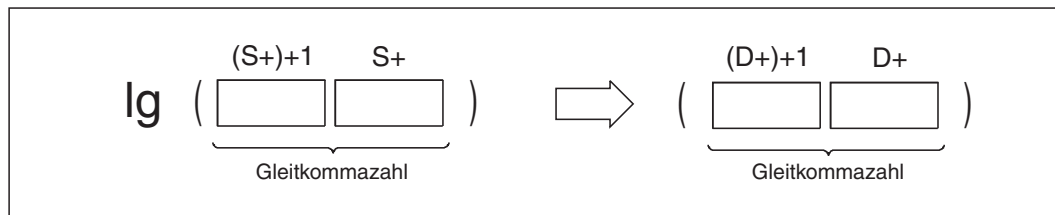


Abb. 7-90: Funktionsweise der DLOG10-Anweisung

- In ((S+)+1) und (S+) können nur positive Werte eingegeben werden. Die Berechnung des dekadischen Logarithmus kann nicht mit negativen Werten ausgeführt werden.

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- ((S+)+1) und (S+) enthält einen negativen Wert.
- ((S+)+1) und (S+) enthält den Wert „0“.

Beispiel ▾

Wenn der Eingang X0 eingeschaltet ist, berechnet das folgende Programm den dekadischen Logarithmus des Wertes „15“ und speichert das Ergebnis in D30 und D31.

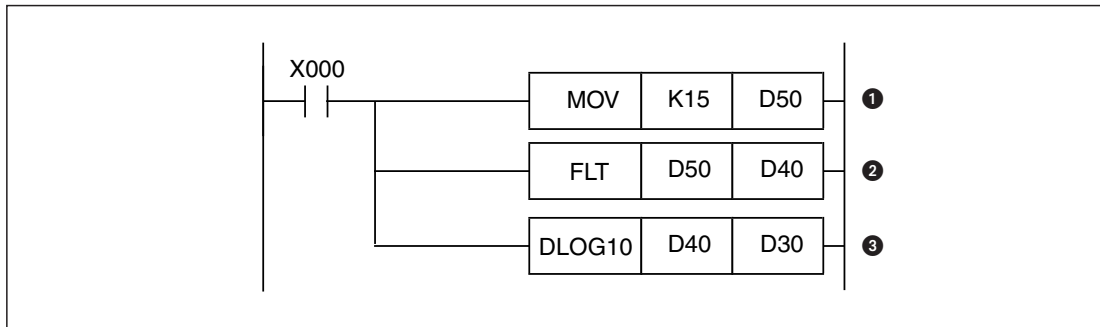


Abb. 7-91: Programmbeispiel für die Anwendung der DLOGE-Anweisung

- ❶ Die Konstante „15“ wird in das Datenregister D50 übertragen.
- ❷ Die Konstante wird in eine Gleitkommazahl gewandelt.
- ❸ Berechnung des dekadischen Logarithmus der Konstanten.

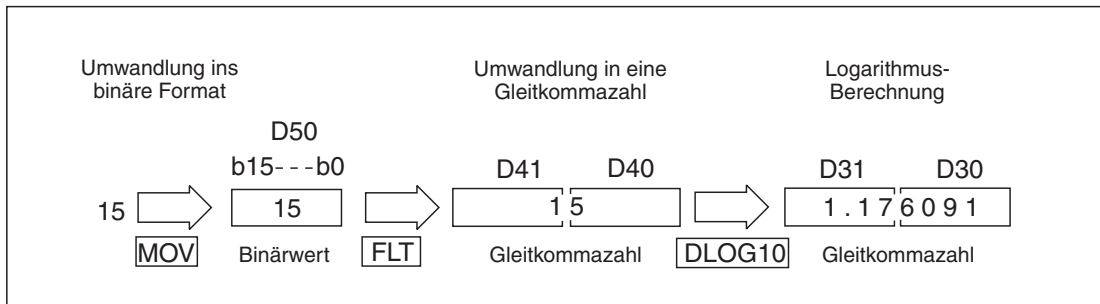


Abb. 7-92: Verarbeitung der Daten bei diesem Programmbeispiel



7.7.15 Quadratwurzel aus Gleitkommazahlen (DESQR)

		DESQR		FNC 127				
		Quadratwurzel aus Gleitkommazahlen						
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
					●	●	●	●
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	K, H (Integer-Werte werden automatisch in Gleitkommazahlen konvertiert.) Gleitkommakonstante ^① , D, R ^① , U□G□ ^② (Gleitkommazahl (32 Bits))	D, R ^① , U□G□ ^② (Gleitkommazahl (32 Bits))		●	16 Bit	32 Bit	DESQR	9
					●	DESQRP	9	

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
 ② Nur bei FX3U und FX3UC

Funktion

Berechnen der Quadratwurzel aus einer Gleitkommazahl und Speichern des Ergebnisses

Beschreibung

- Es wird aus der ab (S+) angegebenen Gleitkommazahl die Quadratwurzel berechnet. Das Ergebnis wird ab (D+) gespeichert.
- Es werden für jeden Operanden jeweils 2 aufeinanderfolgende Register verwendet.
- Konstanten (K, H) werden vor der Wurzelberechnung automatisch in Gleitkommazahlen umgewandelt.
- Als Quelle und Ziel kann derselbe Operand verwendet werden. In diesem Fall wird das errechnete Ergebnis wieder im Quelloperanden gespeichert und anschließend für die nächste Berechnung genutzt. Dieser Prozess wiederholt sich bei jeder Ausführung der (DESQR) mit jedem Zyklus.
- Wenn das Ergebnis der Wurzelberechnung 0 beträgt, wird das Zero-Flag M8020 gesetzt.

HINWEIS

Die Wurzelberechnungen werden nach den geltenden mathematischen Regeln ausgeführt.

Fehlerquelle

Wenn ab (S+) ein negativer Wert eingegeben wird, erfolgt eine Fehlermeldung und das Error-Flag M8067 wird gesetzt. Die Anweisung wird in diesem Fall nicht ausgeführt.

Beispiel ▾

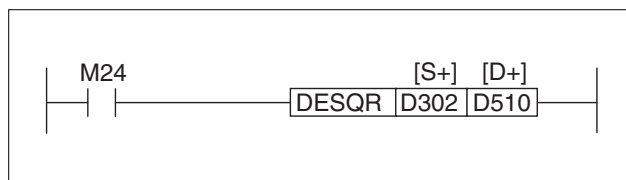


Abb. 7-93:
 Programmierbeispiel zur DESQR-Anweisung

Mit Setzen des Merkers M24 wird aus der Gleitkommazahl in D302 und D303 die Quadratwurzel berechnet.

Das Ergebnis wird in D510 und D511 gespeichert.



7.7.16 Vorzeichenumkehr von Gleitkommazahlen (DENEQ)

		DENEQ		FNC 128			
		Vorzeichenumkehr bei Gleitkommazahlen					
Operanden	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	D, R, U□/G□ Es werden 2 aufeinanderfolgende Adressen der Operanden verwendet.	●	16 Bit	32 Bit	DENEQ	5	
				●	DENEQP	5	

Funktion

Umkehr des Vorzeichens einer Gleitkommazahl

Beschreibung

- Das Vorzeichen der Gleitkommazahl in ((D+)+1) und (D+) wird geändert. Danach wird die Gleitkommazahl mit umgekehrten Vorzeichen wieder in ((D+)+1) und (D+) gespeichert.

HINWEIS

Wird eine DENEQ-Anweisung zyklisch ausgeführt, ändert sich in jedem Programmzyklus das Vorzeichen der Gleitkommazahl. Verwenden Sie die flankengesteuerte Variante der Anweisung (DENEQP), um das Vorzeichen zu einem bestimmten Zeitpunkt zu ändern.

Beispiel

Beim Einschalten des Eingangs X0 wird das Vorzeichen der Gleitkommazahl in D101 und D100 umgekehrt und das Ergebnis wieder in D101 und D100 eingetragen.

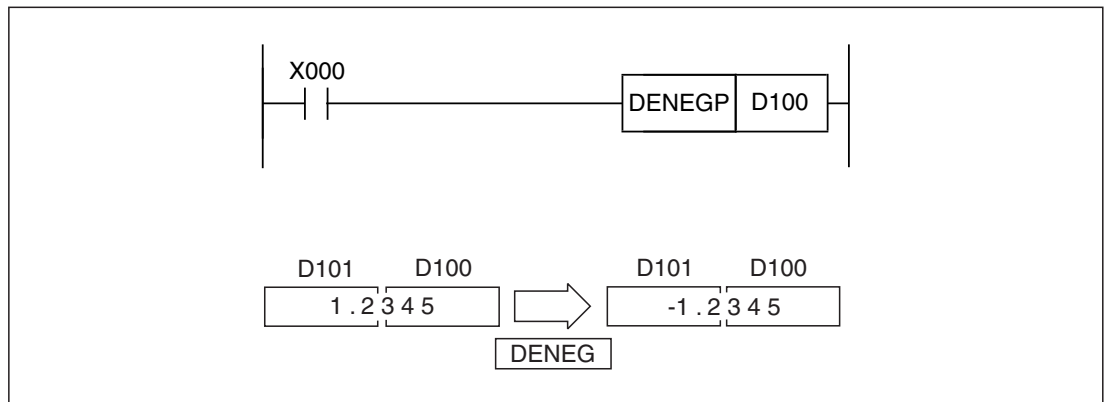


Abb. 7-94: Programmbeispiel zur Vorzeichenumkehr mit der DENEQ-Anweisung



7.7.17 Umwandlung des Gleitkommaformats ins Dezimal-Format (INT)

			INT		FNC 129		
			Umwandlung des Gleitkommaformats ins Dezimal-Format				
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
			●	●	●	●	
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte	
	D, R ^① , U□\G□ ^② (Gleitkommazahl 32 Bits)	D, R ^① , U□\G□ ^② (Dezimal - Format) INT, INTP: 16 Bits DINT, DINTP: 32 Bits	●	16 Bit	32 Bit	INT, INTP	5
				●	●	DINT, DINTP	9

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktion

Konvertierung einer Gleitkommazahl in das Dezimal-Format

Beschreibung

- Die ab (S+) angegebene Gleitkommazahl wird auf den nächst kleineren Integer-Wert abgerundet und ab (D+) gespeichert.
- Der Quelloperand ist immer ein Doppelwortoperand.
- Bei Verwendung der INT-Anweisung ist der Zieloperand ein Wortoperand.
- Bei Verwendung der DINT-Anweisung ist der Ziel-Operand ein Doppelwortoperand.
- Die INT-Anweisung ist die Umkehrfunktion der FLT-Anweisung.
- Wenn das Ergebnis der Konvertierung 0 beträgt, wird das Zero-Flag M8020 gesetzt.
- Wird ab (S+) keine ganze Zahl angegeben, wird diese Zahl auf den nächst kleineren Integer-Wert abgerundet und das Borrow-Flag M8021 gesetzt.
- Wenn der konvertierte Integer-Wert außerhalb des Speicherbereichs des Zieloperanden liegt, tritt ein Überlauf auf, und das Carry-Flag M8022 wird gesetzt.

HINWEIS | Beim Auftreten eines Überlaufs ist das Ergebnis im Zieloperanden fehlerhaft.

Beispiel ▾

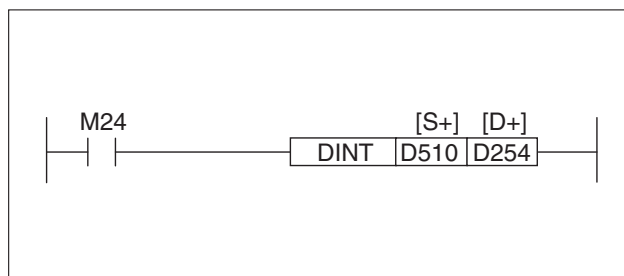


Abb. 7-95:
*Programmierbeispiel zur DINT-
 Anweisung*

Mit Setzen des Merkers M25 wird die Gleitkommazahl in D510 und D511 auf den nächst kleineren Integer-Wert abgerundet und das Borrow-Flag M8021 gesetzt.

Das Ergebnis wird in D254 und D255 gespeichert.



7.7.18 Sinusberechnung mit Gleitkommazahlen (DSIN)

			DSIN		FNC 130		
			Sinusberechnung mit Gleitkommazahlen				
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
			●			●	
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte	
	D, R*, U□\G□* (Gleitkommazahl in Radiant (32 Bits)), Gleitkommakonstante*	D, R*, U□\G□* (Gleitkommazahl (32 Bits))	●	16 Bit	32 Bit	DSIN	9
					●	DSINP	9

* Nur bei FX3U und FX3UC

Funktion

Berechnen des Sinus einer Gleitkommazahl und Speichern des Ergebnisses

Beschreibung

- Es wird der Sinus aus der ab (S+) angegebenen Gleitkommazahl berechnet. Das Ergebnis wird ab (D+) gespeichert.
- Es werden für jeden Operanden jeweils 2 aufeinanderfolgende Register verwendet.
- Die Werte im Quell- und Zieloperanden haben Gleitkommaformat.
- Der ab (S+) angegebene Winkelwert muss ein Wert zwischen 0 und 360° (0 und 2 π rad) sein. Der Wert des Winkels wird im Bogenmaß angegeben ($\text{Grad} \times \pi / 180$ [rad]). Für die Umrechnung zwischen Grad und Radiant können die Anweisungen DRAD und DDEG verwendet werden (Abschnitte 7.7.24 und 7.7.25).

Beispiel ▾

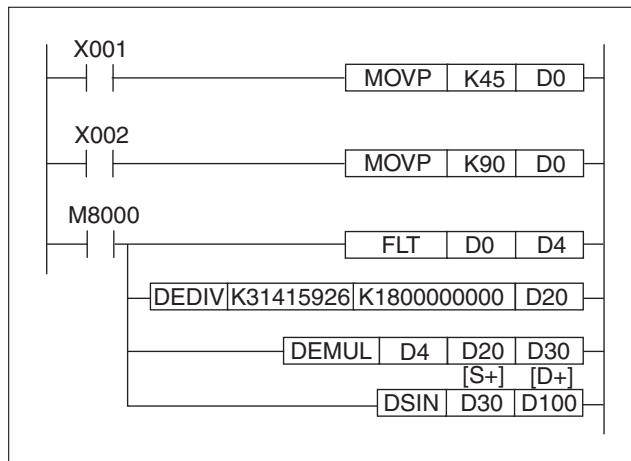


Abb. 7-96:

Programmierbeispiel zur DSIN-Anweisung mit Umwandlung in das Bogenmaß

Mit positiver Flanke des Eingangs X1 wird die Konstante K45 (45°) in D0 geschrieben.

Mit positiver Flanke des Eingangs X2 wird die Konstante K90 (90°) in D0 geschrieben.

Mit Setzen des Merkers M8000 wird der Wert aus D0 in eine Gleitkommazahl konvertiert und in D4 und D5 gespeichert.


Mit den DEDIV- und DEMUL-Anweisungen wird dieser Wert in das Bogenmaß umgerechnet.

Das Ergebnis wird in D30 und D31 gespeichert.

Mit der DSIN-Anweisung wird daraus der Sinus berechnet. Das Ergebnis wird in D100 und D101 gespeichert.



7.7.19 Cosinusberechnung mit Gleitkommazahlen (DCOS)

		DCOS		FNC 131			
		Cosinusberechnung mit Gleitkommazahlen					
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
			●			●	
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte	
	D, R*, U□\G□* (Gleitkommazahl in Radiant (32 Bits)), Gleitkommakonstante*	D, R*, U□\G□* (Gleitkommazahl (32 Bits))	●	16 Bit	32 Bit	DCOS	9
					●	DCOSP	9

* Nur bei FX3U und FX3UC

Funktion

Berechnen des Cosinus einer Gleitkommazahl und Speichern des Ergebnisses

Beschreibung

- Es wird der Cosinus aus der ab (S+) angegebenen Gleitkommazahl berechnet. Das Ergebnis wird ab (D+) gespeichert.
- Es werden für jeden Operanden jeweils 2 aufeinanderfolgende Register verwendet.
- Die Werte im Quell- und Zieloperanden haben Gleitkommaformat.
- Der ab (S+) angegebene Winkelwert muss ein Wert zwischen 0 und 360° (0 und 2 π rad) sein. Der Wert des Winkels wird im Bogenmaß angegeben ($\text{Grad} \times \pi / 180$ [rad]). Für die Umrechnung zwischen Grad und Radiant können die Anweisungen DRAD und DDEG verwendet werden (Abschnitte 7.7.24 und 7.7.25).

Beispiel ▾

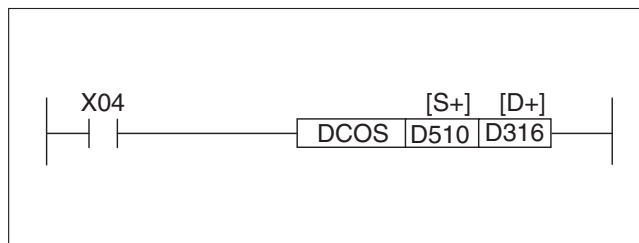


Abb. 7-97:
 Programmierbeispiel zur DCOS-
 Anweisung

Wenn der Eingang X4 eingeschaltet ist, wird der Cosinus des Winkels in Bogenmaß (D510, D511) berechnet (Die Umrechnung von Grad in Radiant wird im Beispiel in Abschnitt 7.7.18 gezeigt).

Das Ergebnis wird in D316 und D317 gespeichert.



7.7.20 Tangensberechnung mit Gleitkommazahlen (DTAN)

		DTAN		FNC 132			
		Tangensberechnung mit Gleitkommazahlen					
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S
					●		●
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte	
	D, R*, U□\G□* (Gleitkommazahl in Radiant (32 Bits)), Gleitkommakonstante*	D, R*, U□\G□* (Gleitkommazahl (32 Bits))	●	16 Bit	32 Bit	DTAN	9
					●	DTANP	9

* Nur bei FX3U und FX3UC

Funktion

Berechnen des Tangens einer Gleitkommazahl und Speichern des Ergebnisses

Beschreibung

- Es wird der Tangens aus der ab (S+) angegebenen Gleitkommazahl berechnet. Das Ergebnis wird ab (D+) gespeichert.
- Es werden für jeden Operanden jeweils 2 aufeinanderfolgende Register verwendet.
- Die Werte im Quell- und Zieloperanden haben Gleitkommaformat.
- Der ab (S+) angegebene Winkelwert muss ein Wert zwischen 0 und 360° (0 und 2π rad) sein. Der Wert des Winkels wird im Bogenmaß angegeben (Grad x π / 180 [rad]). Für die Umrechnung zwischen Grad und Radiant können die Anweisungen DRAD und DDEG verwendet werden (Abschnitte 7.7.24 und 7.7.25).

Beispiel ▾

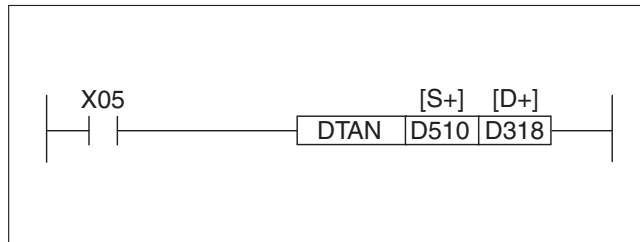



Abb. 7-98:
 Programmierbeispiel zur DTAN-
 Anweisung

Wenn der Eingang X5 eingeschaltet ist, wird der Tangenswert des Winkels im Bogenmaß (D510, D511) berechnet (Umrechnung Grad in Radiant siehe Abschnitt 7.7.24).

Das Ergebnis wird in D318 und D319 gespeichert.



7.7.21 Arcussinusberechnung mit Gleitkommazahlen (DASIN)

			DASIN		FNC 133			
			Arcussinusberechnung					
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC		
							●	
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	D, R, U□/G□, Gleitkommakonstante	D, R, U□/G□	●	16 Bit	32 Bit	DASIN	9	
	Es werden 2 aufeinanderfolgende Adressen der Operanden verwendet.				●	DASINP	9	

Funktion

Berechnen des Arcussinus (\sin^{-1}) einer Gleitkommazahl und Speichern des Ergebnisses

Die Arcussinusberechnung ist die Umkehrfunktion der Sinusberechnung. Mit einer DASIN-Anweisung wird aus einem Sinuswert der zugehörige Winkel im Bogenmaß berechnet.

Beschreibung

- Es wird der Winkel aus dem ab (S+) gespeicherten Sinuswert berechnet. Das Ergebnis wird ab (D+) gespeichert.
- Als (S+) und ((S+)+1) kann auch eine Gleitkommakonstante angegeben werden.
- Die Werte im Quell- und Zieloperanden haben Gleitkommaformat.
- Der ab (S+) angegebene Sinuswert muss innerhalb des Bereichs von -1,0 bis 1,0 liegen.
- Das Ergebnis in ((D+)+1) und (D+) ist ein Winkel, der im Bogenmaß mit der Maßeinheit Radiant (rad) angegeben wird (Wertebereich: $-\pi/2$ bis $\pi/2$). Für die Umrechnung zwischen Grad und Radiant können die Anweisungen DRAD und DDEG verwendet werden (siehe Abschnitte 7.7.24 und 7.7.25).

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- ((S+)+1) und (S+) enthalten einen Wert, der außerhalb des Bereichs von -1,0 bis 1,0 liegt.

Beispiel ▾

Wenn der Eingang X0 eingeschaltet ist, berechnet das folgende Programm den Winkel aus dem Sinuswert, der in D0 und D1 gespeichert ist, und gibt diesen Wert über eine vierstellige BCD-Anzeige aus. Diese Anzeige ist an den 16 Ausgängen Y40 bis Y57 angeschlossen.

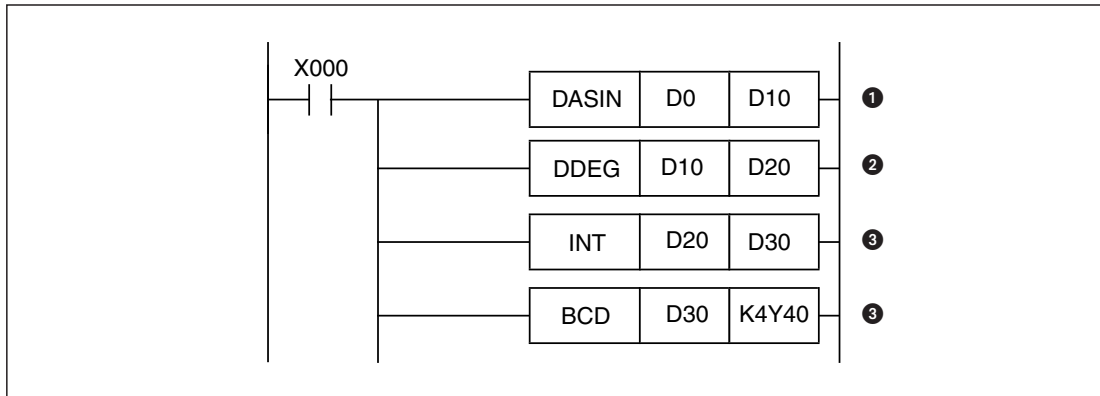


Abb. 7-99: Programmbeispiel zur Arcussinusberechnung und Ausgabe des Winkels an eine BCD-Anzeige

- ① Aus dem Sinuswert wird mit der DASIN-Anweisung der Winkel im Bogenmaß berechnet.
- ② Mit einer DDEG-Anweisung wird der Winkel in die Maßeinheit „Grad“ umgerechnet.
- ③ Die Gleitkommazahl mit der Angabe des Winkels wird in eine ganzzahlige Dezimalzahl gewandelt.
- ④ Der Winkel wird auf der BCD-Anzeige dargestellt.

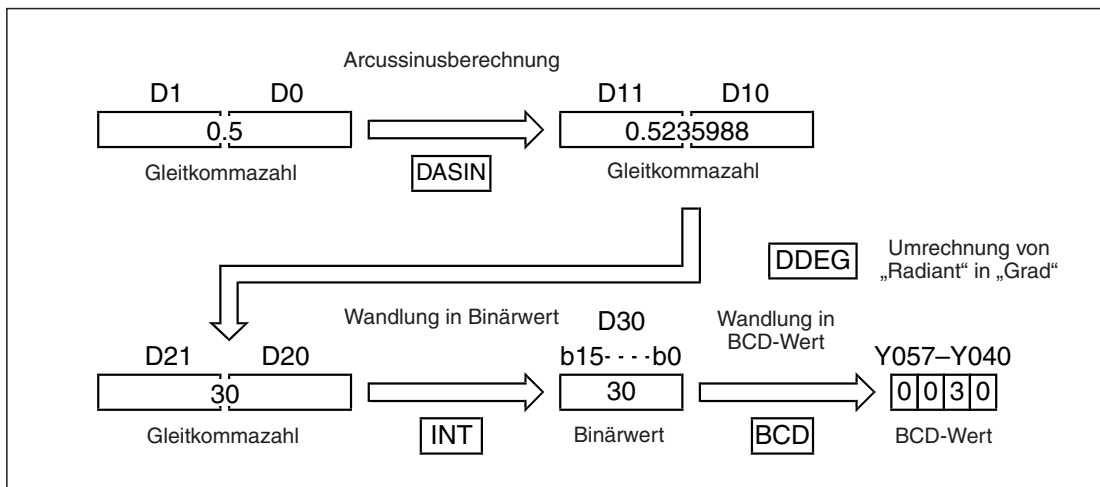



Abb. 7-100: Bei Vorgabe des Sinuswertes „0,5“ wird mit dem oben abgebildeten Programm der Winkel „30°“ angezeigt.



7.7.22 Arcuscoberechnung mit Gleitkommazahlen (DACOS)

			DACOS		FNC 134			
			Arcuscoberechnung					
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	D, R, U□/G□, Gleitkommakonstante	D, R, U□/G□	●	16 Bit	32 Bit	DACOS	9	
Es werden 2 aufeinanderfolgende Adressen der Operanden verwendet.			●	●	DACOSP	9		

Funktion

Berechnen des Arcuscoberechnung (\cos^{-1}) einer Gleitkommazahl und Speichern des Ergebnisses

Die Arcuscoberechnung ist die Umkehrfunktion der Cosinusberechnung. Mit einer DACOS-Anweisung wird aus einem Cosinuswert der zugehörige Winkel im Bogenmaß berechnet.

Beschreibung

- Es wird der Winkel aus dem ab (S+) gespeicherten Cosinuswert berechnet. Das Ergebnis wird ab (D+) gespeichert.
- Als (S+) und ((S+)+1) kann auch eine Gleitkommakonstante angegeben werden.
- Die Werte im Quell- und Zieloperanden haben Gleitkommaformat.
- Der ab (S+) angegebene Cosinuswert muss innerhalb des Bereichs von -1,0 bis 1,0 liegen.
- Das Ergebnis in ((D+)+1 und (D+) ist ein Winkel, der im Bogenmaß mit der Maßeinheit Radiant (rad) angegeben wird (Wertebereich: 0 bis π). Für die Umrechnung zwischen Grad und Radiant können die Anweisungen DRAD und DDEG verwendet werden (siehe Abschnitte 7.7.24 und 7.7.25).

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- ((S+)+1) und (S+) enthalten einen Wert, der außerhalb des Bereichs von -1,0 bis 1,0 liegt.

Beispiel ▾

Wenn der Eingang X0 eingeschaltet ist, berechnet das folgende Programm den Winkel aus dem Cosinuswert, der in D0 und D1 gespeichert ist, und gibt diesen Wert über eine vierstellige BCD-Anzeige aus. Diese Anzeige ist an den 16 Ausgängen Y40 bis Y57 angeschlossen.

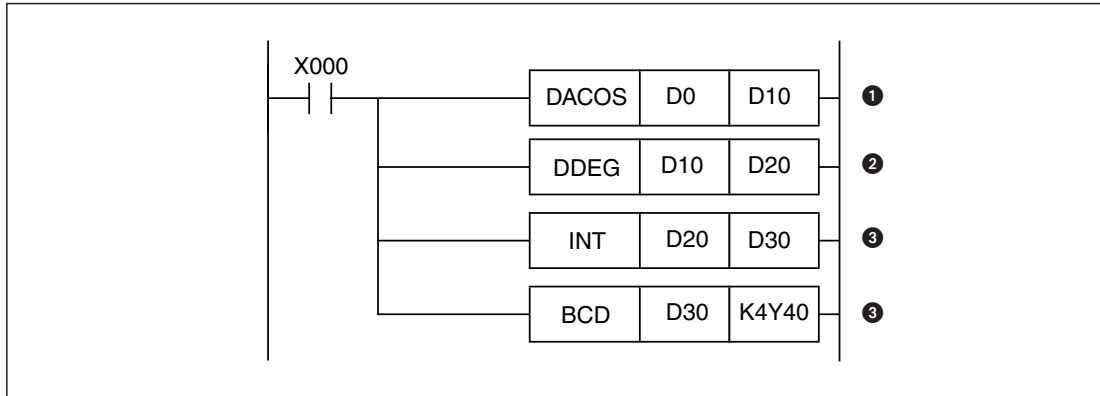


Abb. 7-101: Programmbeispiel zur Arcussinusberechnung und Ausgabe des Winkels an eine BCD-Anzeige.

- ① Aus dem Cosinuswert wird mit der DACOS-Anweisung der Winkel im Bogenmaß berechnet.
- ② Mit einer DDEG-Anweisung wird der Winkel in die Maßeinheit „Grad“ umgerechnet.
- ③ Die Gleitkommazahl mit der Angabe des Winkels wird in eine ganzzahlige Dezimalzahl gewandelt.
- ④ Der Winkel wird auf der BCD-Anzeige dargestellt.

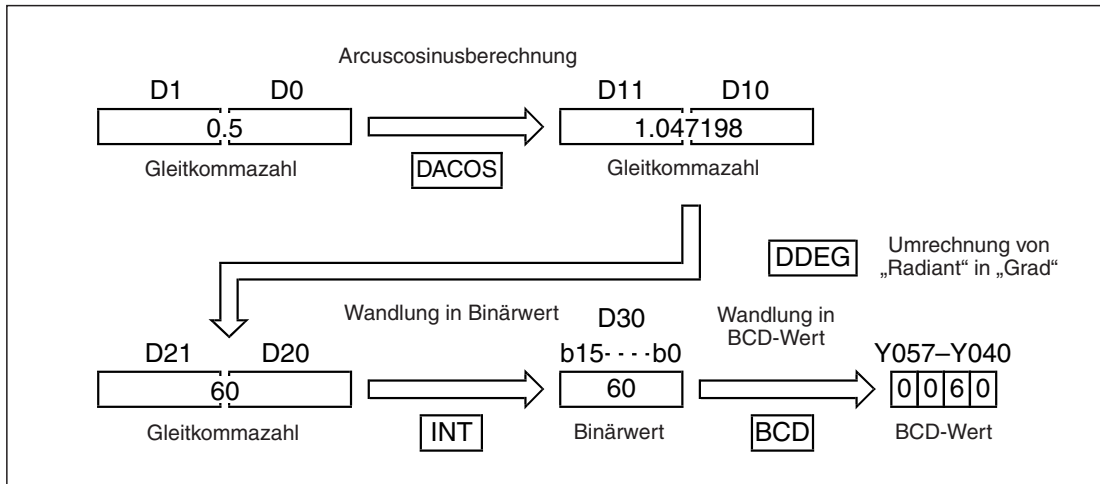



Abb. 7-102: Bei Vorgabe des Cosinuswertes „0,5“ wird mit dem oben abgebildeten Programm der Winkel „60°“ angezeigt.



7.7.23 Arcustangensberechnung mit Gleitkommazahlen (DATAN)

			DATAN		FNC 135			
			Arcuscosinusberechnung					
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC		
							●	
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	D, R, U□/G□, Gleitkommakonstante	D, R, U□/G□	●	16 Bit	32 Bit	DATAN	9	
	Es werden 2 aufeinanderfolgende Adressen der Operanden verwendet.				●	DATANP	9	

Funktion

Berechnen des Arcustangens (\tan^{-1}) einer Gleitkommazahl und Speichern des Ergebnisses

Die Arcustangensberechnung ist die Umkehrfunktion der Tangensberechnung. Mit der DATAN-Anweisung wird aus einem Tangenswert der zugehörige Winkel im Bogenmaß berechnet.

Beschreibung

- Es wird der Winkel aus dem ab (S+) gespeicherten Tangenswert berechnet. Das Ergebnis wird ab (D+) gespeichert.
- Als (S+) und ((S+)+1) kann auch eine Gleitkommakonstante angegeben werden.
- Die Werte im Quell- und Zieloperanden haben Gleitkommaformat.
- Das Ergebnis in ((D+)+1 und (D+) ist ein Winkel, der im Bogenmaß mit der Maßeinheit Radiant (rad) angegeben wird (Wertebereich: $-\pi/2$ bis $\pi/2$). Für die Umrechnung zwischen Grad und Radiant können die Anweisungen DRAD und DDEG verwendet werden (siehe Abschnitte 7.7.24 und 7.7.25).

Beispiel ▾

Wenn der Eingang X0 eingeschaltet ist, berechnet das folgende Programm den Winkel aus dem Tangenswert, der in D0 und D1 gespeichert ist, und gibt diesen Wert über eine vierstellige BCD-Anzeige aus. Diese Anzeige ist an den 16 Ausgängen Y40 bis Y57 angeschlossen.

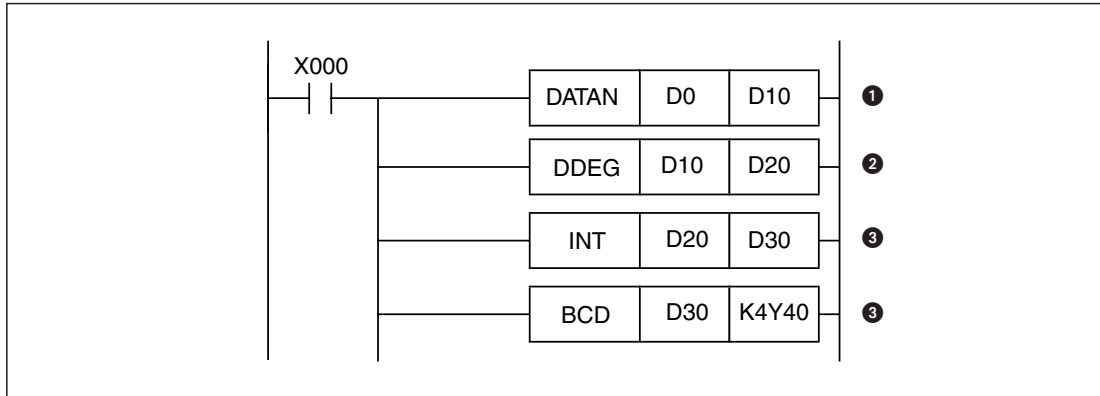


Abb. 7-103: Programmbeispiel zur Arcussinusberechnung und Ausgabe des Winkels an eine BCD-Anzeige.

- ① Aus dem Tangenswert errechnet eine DATAN-Anweisung den Winkel im Bogenmaß.
- ② Mit einer DDEG-Anweisung wird der Winkel in die Maßeinheit „Grad“ umgerechnet.
- ③ Die Gleitkommazahl mit der Angabe des Winkels wird in eine ganzzahlige Dezimalzahl gewandelt.
- ④ Der Winkel wird auf der BCD-Anzeige dargestellt.

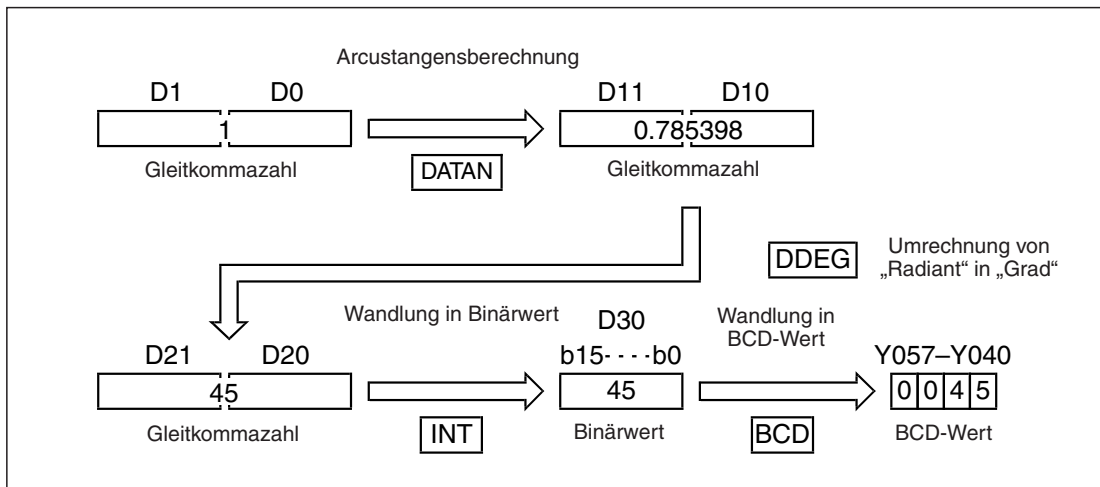


Abb. 7-104: Bei Vorgabe des Tangenswertes „1“ wird mit dem oben abgebildeten Programm der Winkel „45°“ angezeigt.



7.7.24 Umrechnung von Grad in Radiant (DRAD)

			DRAD		FNC 136				
			Umrechnung von Grad in Radiant						
			CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
									●
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	D, R, U□/G□, Gleitkommakonstante	D, R, U□/G□		●	16 Bit	32 Bit	DRAD	9	
Es werden 2 aufeinanderfolgende Adressen der Operanden verwendet.					●	DRADP	9		

Funktion

Umrechnung eines Winkelwerts von Grad in Radiant

Beschreibung

- Die DRAD-Anweisung berechnet aus der Gradangabe (°) in ((S+)+1) und (S+) den entsprechenden Wert im Bogenmaß (rad) und speichert das Ergebnis in ((D+)+1) und (D+).

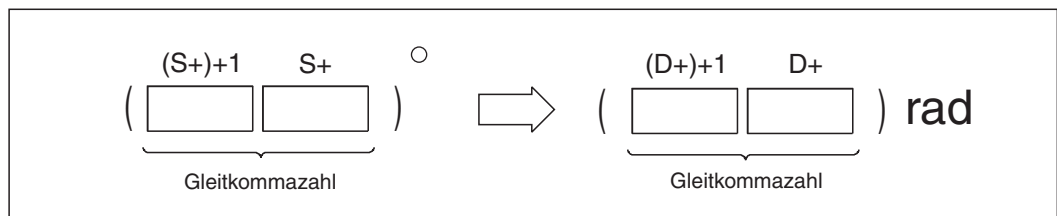


Abb. 7-105: Funktionsweise der DRAD-Anweisung

- Zur Umrechnung von Grad in Radiant wird die folgende Formel verwendet:

$$\text{Radiant} = \text{Grad} \times \frac{\pi}{180}$$

- Als (S+) und ((S+)+1) kann auch eine Gleitkommakonstante angegeben werden.
- Die Werte im Quell- und Zieloperanden haben Gleitkommaformat.

Beispiel ▾

An den Eingängen X020 bis X037 ist ein vierstelliger BCD-Schalter angeschlossen, an dem ein Winkelwert in der Einheit „Grad“ eingegeben werden kann. Wenn der Eingang X0 eingeschaltet ist, rechnet das folgende Programm den eingestellten Wert in das Bogenmaß um und speichert das Ergebnis in D20 und D21.

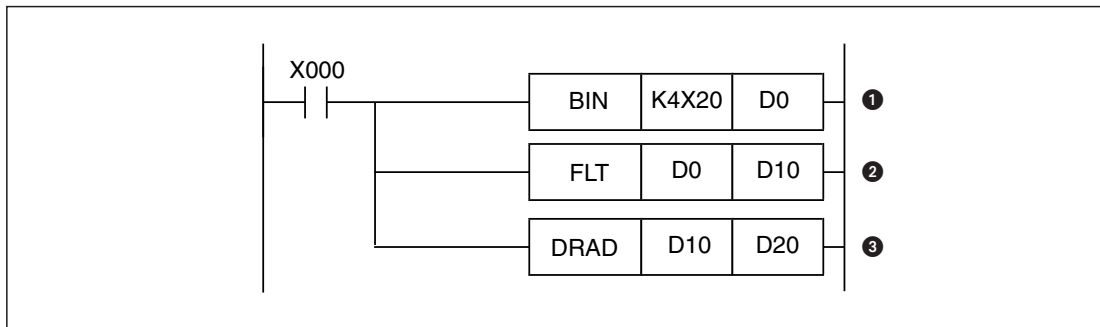


Abb. 7-106: Programmbeispiel für die Anwendung der DRAD-Anweisung

- ① Eingestellten Wert in der Einheit „Grad“ einlesen
- ② Der Winkel wird in eine Gleitkommazahl gewandelt.
- ③ Umrechnung von Grad in Radiant

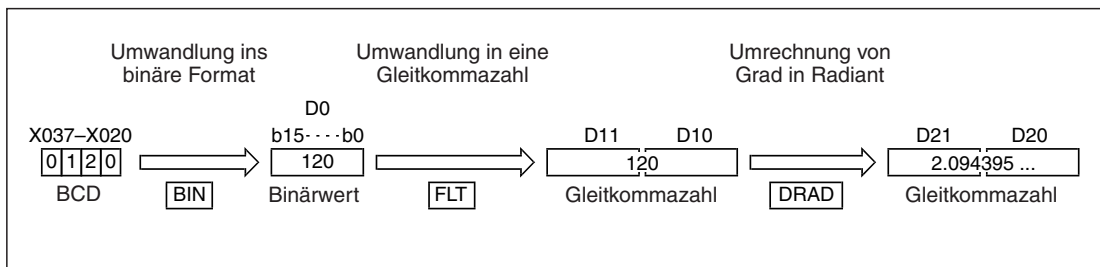



Abb. 7-107: Das Beispielprogramm gibt bei der Eingabe von 120° als Wandlungsergebnis 2,094395 rad aus.



7.7.25 Umrechnung von Radiant in Grad (DDEG)

		DDEG		FNC 136				
		Umrechnung von Radiant in Grad						
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
								●
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	D, R, U□/G□, Gleitkommakonstante	D, R, U□/G□		●	16 Bit	32 Bit	DDEG	9
Es werden 2 aufeinanderfolgende Adressen der Operanden verwendet.			●		●	DDEGP	9	

Funktion

Umrechnung eines Winkelwerts von Radiant in Grad

Beschreibung

- Die DDEG-Anweisung berechnet aus einem in ((S+)+1) und (S+) angegebenen Winkel im Bogenmaß (rad) den entsprechenden Wert in der Einheit „Grad“ (°) und speichert das Ergebnis in ((D+)+1) und (D+).

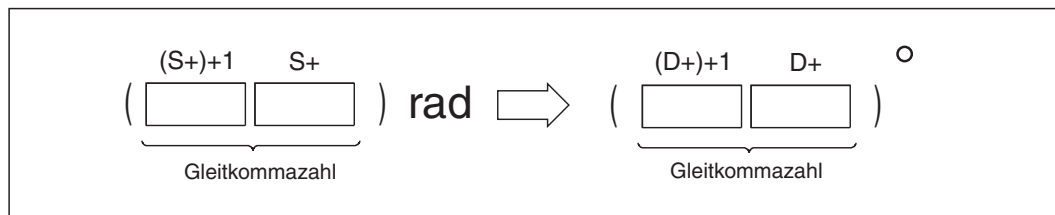


Abb. 7-108: Funktionsweise der DDEG-Anweisung

- Zur Umrechnung von Grad in Radiant wird die folgende Formel verwendet:

$$\text{Grad} = \text{Radiant} \times \frac{180}{\pi}$$

- Als (S+) und ((S+)+1) kann auch eine Gleitkommakonstante angegeben werden.
- Die Werte im Quell- und Zieloperanden haben Gleitkommaformat.

Beispiel ▾

Das folgende Beispielprogramm wandelt einen Winkel, der im Bogenmaß in D20 und D21 gespeichert ist, in einen Winkel mit der Maßeinheit „Grad“ um und gibt diesen Wert über eine vierstellige BCD-Anzeige aus. Diese Anzeige ist an den 16 Ausgängen Y40 bis Y57 angeschlossen.

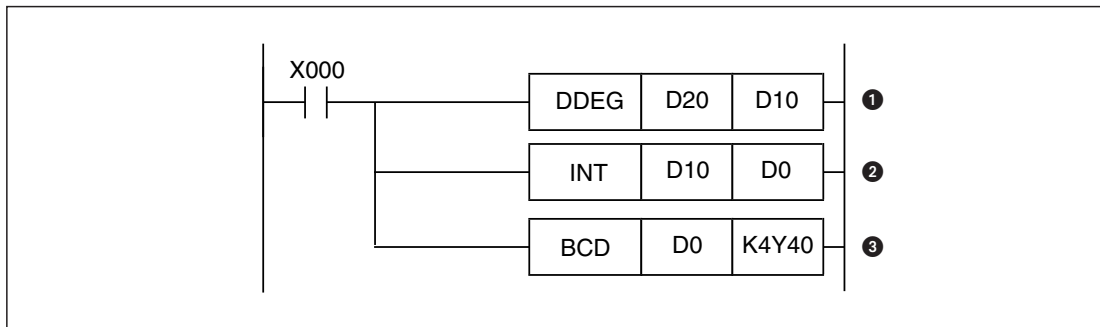


Abb. 7-109: Programmbeispiel für die Anwendung der DDEG-Anweisung

- ① Aus dem Wert des Winkels im Bogenmaß wird der Wert in Grad berechnet.
- ② Die Gleitkommazahl mit der Angabe des Winkels wird in eine ganzzahlige Dezimalzahl gewandelt.
- ③ Der Winkel wird auf der BCD-Anzeige dargestellt.

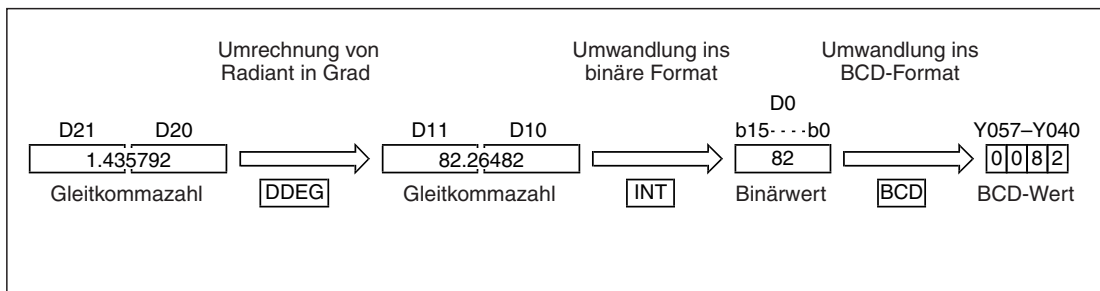


Abb. 7-110: Das Beispielprogramm gibt bei der Eingabe von 120° als Wandlungsergebnis 2,094395 rad aus.



7.8 Datenverarbeitungsanweisungen

Übersicht der Anweisung FNC 140 bis 149

Symbol	FNC	Bedeutung	Abschnitt
WSUM	140	Summe der Inhalte von Wortoperanden bilden	7.8.1
WTOB	141	Daten in Wortoperanden in Bytes aufteilen	7.8.2
BTOW	142	Wortoperanden aus einzelnen Bytes bilden	7.8.3
UNI	143	Gruppen von 4 Bits zu Wortoperanden zusammenfassen	7.8.4
DIS	144	Wortoperanden in Gruppen von 4 Bit aufteilen	7.8.5
SWAP	147	High - Low - Byte - Tausch	7.8.6
SORT2	149	Daten in Tabelle sortieren	7.8.7

Tab. 7-23: Übersicht der Datenverarbeitungsanweisungen

7.8.1 Summe der Inhalte von Wortoperanden bilden (WSUM)

				WSUM		FNC 140												
				Summe der Inhalte von Wortoperanden bilden														
<table border="1"> <tr> <th>Operanden</th> <th>S+</th> <th>D+</th> <th>n</th> </tr> <tr> <td>T, C, D, R, Sondermodule (U□/G□)</td> <td></td> <td></td> <td>D, R, K, H</td> </tr> </table>				Operanden	S+	D+	n	T, C, D, R, Sondermodule (U□/G□)			D, R, K, H	CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
				Operanden	S+	D+	n											
T, C, D, R, Sondermodule (U□/G□)			D, R, K, H															
										●								
				Puls-Anweisung (P)		Verarbeitung		Programmschritte										
				●		16 Bit	32 Bit	WSUM WSUMP	7									
						●	●	DWSUM DWSUMP	13									

Funktion

Summenbildung aus den Inhalten von Wortoperanden

Beschreibung

- Die WSUM-Anweisung bildet die Summe der mit (n) angegebenen Anzahl von 16-Bit- oder 32-Bit-Binärdatenblöcken. In (S+) wird die erste Adresse des Operandenbereich angegeben, in dem die zu addierenden Operanden gespeichert sind. Das Ergebnis wird ab dem in (D+) angegebenen Operanden gespeichert.

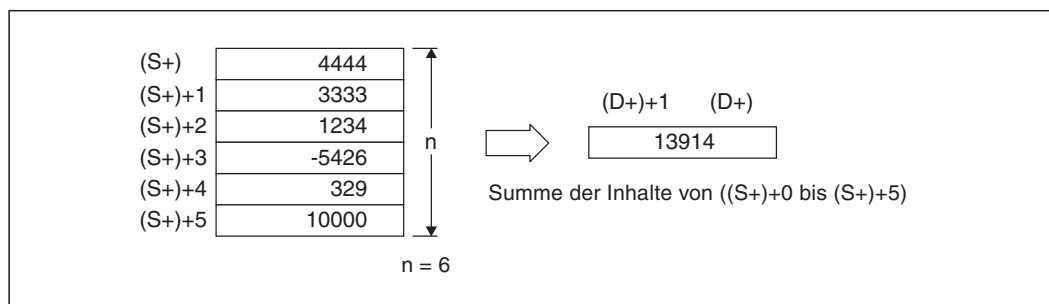


Abb. 7-111: Beispiel für die Ausführung einer WSUM-Anweisung zur Addition von 16-Bit-Daten

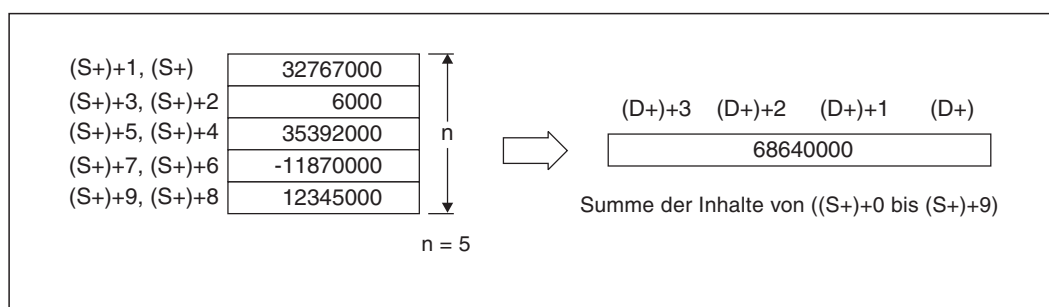


Abb. 7-112: Beispiel für die Ausführung einer DWSUM-Anweisung zur Addition von 32-Bit-Daten

- Für (n) muss ein Wert angegeben werden, der größer als „0“ ist.

HINWEISE

Bei der Addition von 32-Bit-Daten mit einer DWSUM-Anweisung wird das Ergebnis als 64-Bit-Wert gespeichert. 64-Bit-Daten können jedoch von einer SPS der FX3U-Serie nicht verarbeitet werden. Falls die Summe im zulässigen Wertebereich für 32-Bit-Daten liegt (2.147.483.648 bis 2.147.483.647), kann der Inhalt von (D+) und ((D+)+1) ausgewertet und der Inhalt von ((D+)+3) und ((D+)+2) vernachlässigt werden.

Zur Bildung der Summe der Inhalte von Bytes kann die CCD-Anweisung verwendet werden.

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Durch die Angabe von (n) wird der zulässige Bereich für den mit (S+) angegebenen Operanden überschritten.
- Für (n) wurde ein negativer Wert oder der Wert „0“ angegeben.

Beispiel

Wenn der Eingang X10 eingeschaltet wird (steigende Flanke), berechnet das folgende Programm die Summe aus den 16-Bit-Daten, die in D10 bis D14 gespeichert sind und trägt das Ergebnis in D101 und D100 ein.

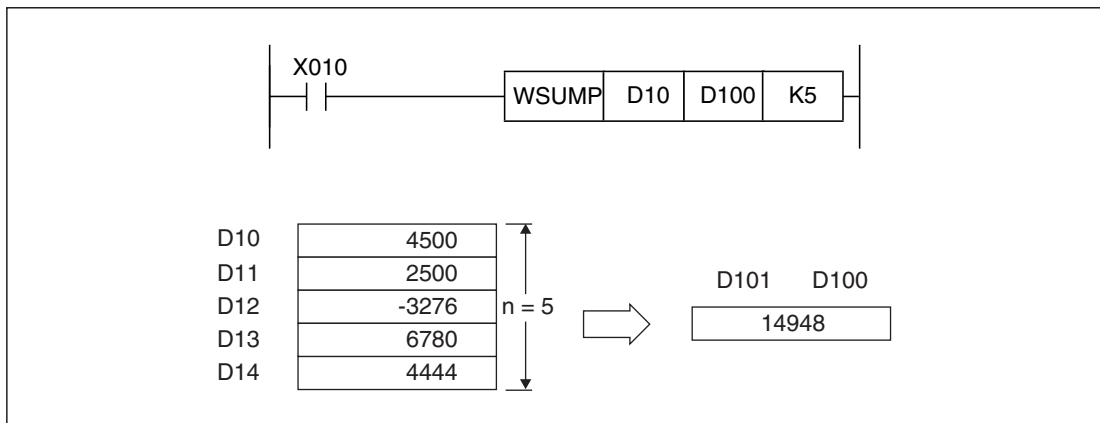


Abb. 7-113: Beispiel für die Summenbildung mit einer WSUMP-Anweisung



7.8.2 Daten in Wortoperanden in Bytes aufteilen (WTOB)

				WTOB		FNC 141			
				Wortoperanden in Bytes aufteilen					
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S
Operanden	S+	D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	T, C, D, R				●	16 Bit	32 Bit	WTOB WTOBP	7

Funktion

Aufteilen von Daten

Beschreibung

- Die WTOB-Anweisung trennt 16-Bit-Datenwerte in Bytes auf und speichert die einzelnen Bytes nacheinander in dem Zielbereich ab, dessen erste Adresse mit (D+) angegeben wurde. Die erste Adresse des Bereichs, in dem die aufzuteilenden Daten gespeichert sind, wird mit (S+) angegeben. (n) gibt die Anzahl der Bytes im Zielbereich an. Zur Speicherung werden nur die niederwertigen Bytes der in (D+) angegebenen Operanden verwendet.

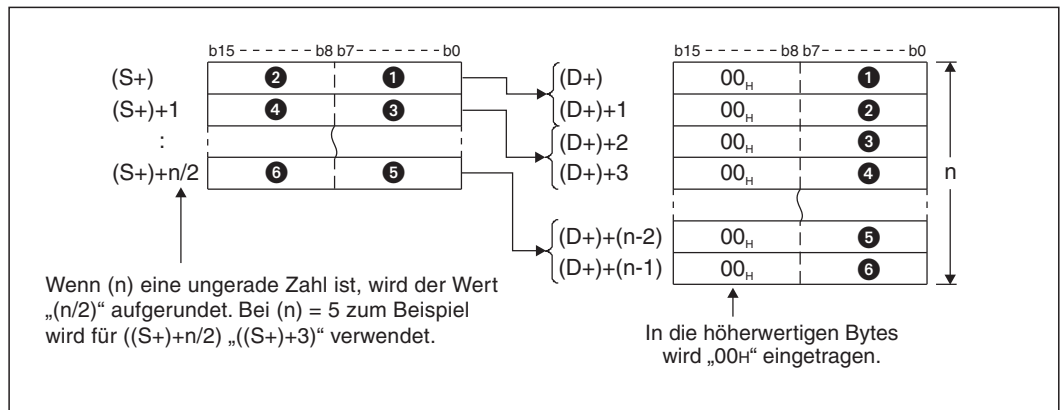


Abb. 7-114: Funktionsweise der WTOB-Anweisung

- Die höherwertigen Bytes der durch (D+) angegebenen Operanden werden mit dem Wert „00H“ beschrieben.
- Wird für (n) ein ungerader Wert angegeben, wird vom letzten Quelloperanden nur das niederwertige Byte erfasst.

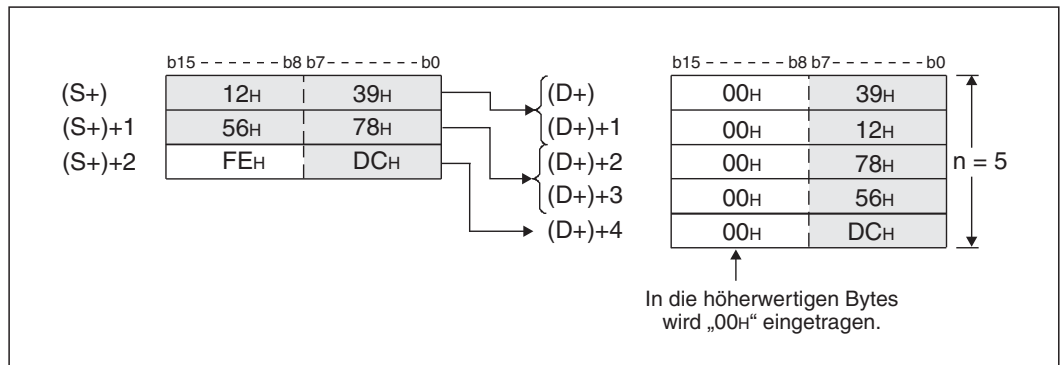


Abb. 7-115: Wenn zum Beispiel für (n) „5“ angegeben wird, werden die Daten von (S+) bis zum niederwertigen Byte von ((S+)+2) erfasst.

- Wird für (n) der Wert „0“ angegeben, wird die WTOB-Anweisung nicht ausgeführt.
- Die Operandenbereiche in (S+) und (D+) dürfen sich überlappen. Wird in diesem Fall aber für (n) ein ungerader Wert angegeben, wird das höherwertige Byte des letzten Quelloperanden durch „00H“ überschrieben.

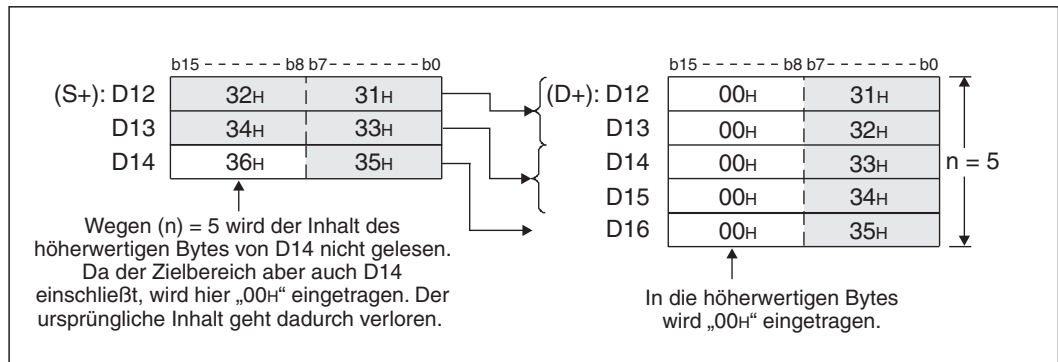


Abb. 7-116: Beispiel für die Nutzung derselben Operanden als Quell- und als Zielbereich.

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Der zulässige Operandenbereich wird durch die mit (S+) bis ((S+)+n/2) angegebenen Operanden überschritten. Falls (n) eine ungerade Zahl ist, wird die Anzahl der Operanden durch den aufgerundeten Betrag von (n/2) bestimmt.
- Der zulässige Operandenbereich wird durch die mit (D+) bis ((D+)+(n-1)) angegebenen Operanden überschritten.

Beispiel ▾

Beim Einschalten des Eingangs X0 werden die Daten, die in D10 bis D12 gespeichert sind, in Bytes aufgeteilt und in D20 bis D25 eingetragen.

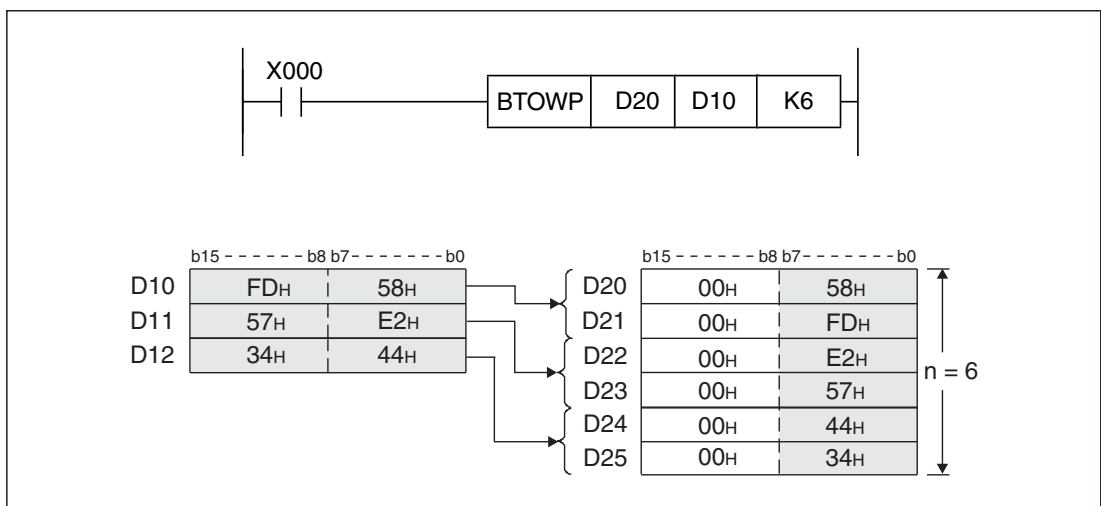


Abb. 7-117: Beispiel für die Aufteilung von Daten mit einer WTOBP-Anweisung



7.8.3 Wortoperanden aus einzelnen Bytes bilden (BTOW)

				BTOW		FNC 142				
				Bytes zu Wortoperanden zusammenfassen						
Operanden	S+	D+	n	CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
										●
				Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	T, C, D, R		D, R, K, H	●	16 Bit	32 Bit	BTOW	BTOWP	7	

Funktion

Zusammenfassen von Daten

Beschreibung

- Die BTOW-Anweisung liest die Inhalte der niederwertigen Bytes der Wortoperanden ab (S+) und speichert die Daten abwechselnd in das nieder- und das höherwertige Byte des Zielbereichs, dessen erste Adresse mit (D+) angegeben wurde. (n) gibt die Anzahl der Bytes und damit die Zahl der Quelloperanden an.

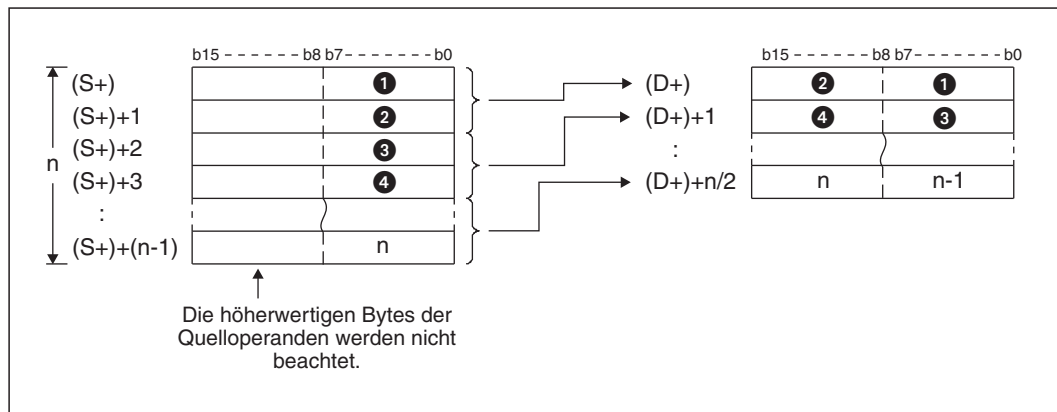


Abb. 7-118: Funktionsweise der BTOW-Anweisung

- Die höherwertigen Bytes der durch (S+) angegebenen Wortoperanden werden ignoriert.
- Wird für (n) ein ungerader Wert angegeben, wird im höherwertigen Byte des letzten Zieloperanden der Wert eingetragen.

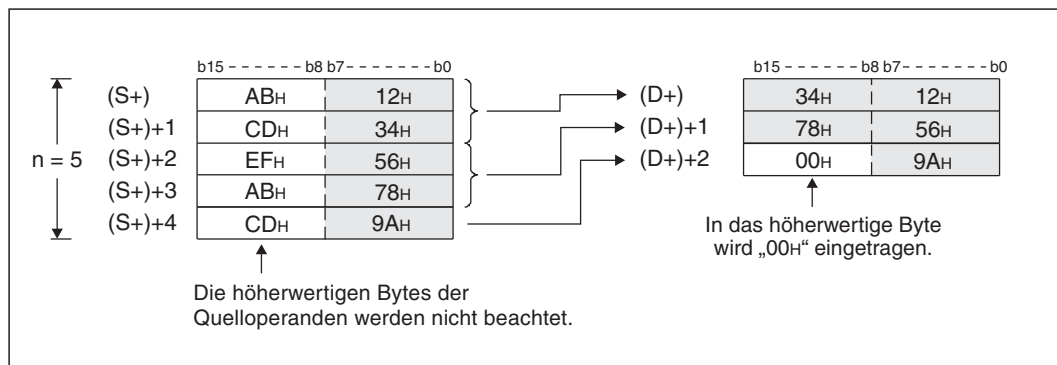


Abb. 7-119: Wenn zum Beispiel für (n) „5“ angegeben wird, werden die Daten von (D+) bis zum niederwertigen Byte von ((D+)+2) eingetragen und das höherwertige Byte von ((D+)+2) gelöscht.

- Wird für (n) der Wert „0“ angegeben, wird die BTOW-Anweisung nicht ausgeführt.
- Die durch (S+) und (D+) definierten Operandenbereiche dürfen sich überlappen. In diesem Fall werden aber die Inhalte der höherwertigen Bytes der Quelloperanden überschrieben, die gleichzeitig auch als Zieloperanden verwendet werden.

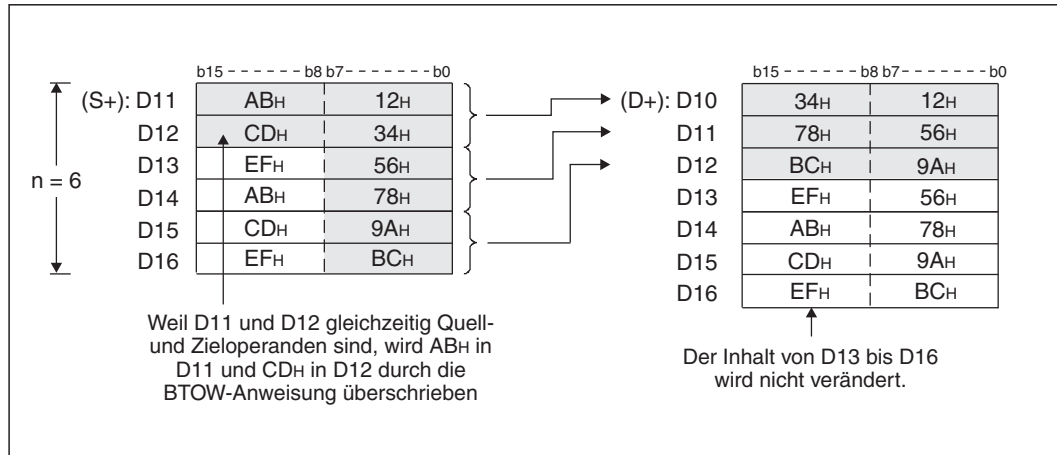


Abb. 7-120: Beispiel für die Nutzung derselben Operanden als Quell- und als Zielbereich.

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermarker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Durch die mit (S+) bis ((S+)+(n-1)) angegebenen Operanden wird der zulässige Operandenbereich überschritten.
- Der zulässige Operandenbereich wird durch die mit (D+) bis ((D+)+n/2) angegebenen Operanden überschritten. Falls (n) eine ungerade Zahl ist, wird die Anzahl der Operanden durch den aufgerundeten Betrag von (n/2) bestimmt.

Beispiel ▽

Beim Einschalten des Eingangs X0 werden die Daten, die in den niederwertigen Bytes der Datenregister D20 bis D25 gespeichert sind, in D10 bis D12 zusammengefasst.

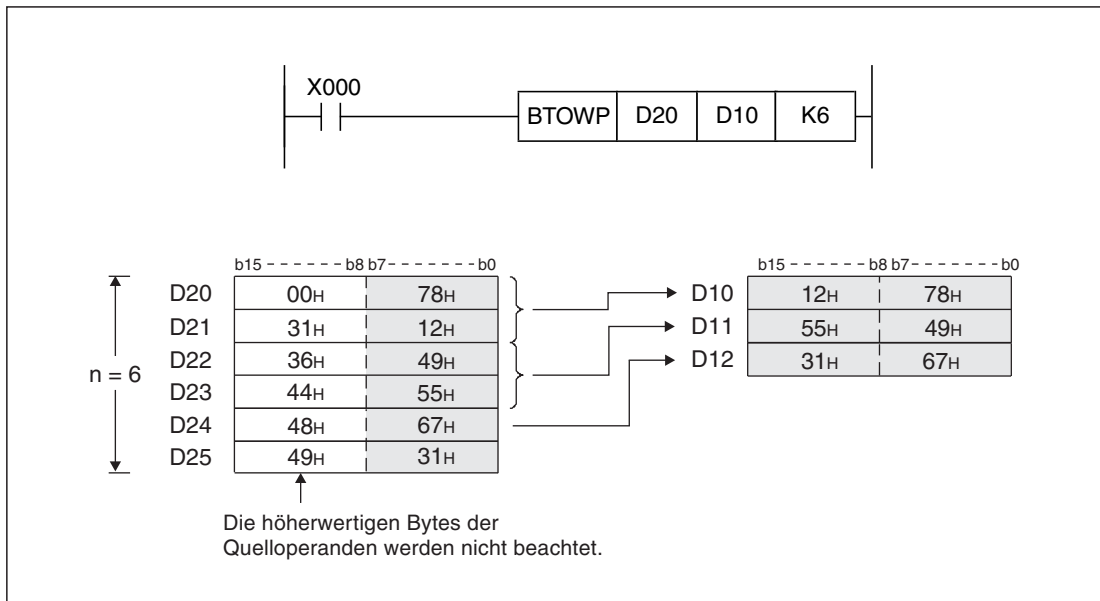


Abb. 7-121: Beispiel für die Zusammenfassung von Daten mit einer BTOWP-Anweisung



7.8.4 Gruppen von 4 Bits zu Wortoperanden zusammenfassen (UNI)

				UNI		FNC 143					
				4-Bit-Gruppen zu Wortoperanden zusammenfassen							
Operanden				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
S+		D+		n		Puls-Anweisung (P)		Verarbeitung		Programmschritte	
T, C, D, R		D, R, K, H				●		16 Bit	32 Bit	UNI UNIP	7
								●			

Funktion

Bis zu vier 4-Bit-Gruppen werden zu einem Wortoperanden zusammengefasst.

Beschreibung

- Die UNI-Anweisung trennt die jeweils 4 niedrigstwertigen Bits von bis zu vier 16-Bit-Datenwerten auf und speichert die Zustände zusammen in einem Wortoperanden (16 Bit) ab. In der Anweisung wird die Startadresse der zusammenzuführenden Datenwerte in (S+), die Anzahl der Operanden in Folge in (n) und die Zieladresse in (D+) festgelegt.

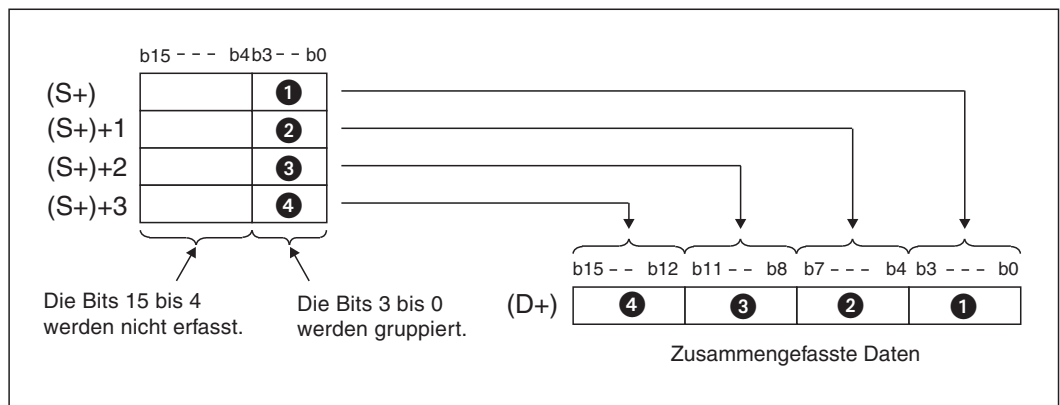


Abb. 7-122: Funktionsweise der UNI-Anweisung

- Für (n) kann ein Wert zwischen 1 und 4 eingestellt werden. Wird für (n) der Wert „0“ angegeben, wird die UNI-Anweisung nicht ausgeführt.
- Wird für (n) ein Wert zwischen 1 und 3 vorgegeben, wird nur die entsprechende Anzahl von 4-Bit-Gruppen in (D+) eingetragen. Die restlichen Bits in (D+) werden auf „0“ zurückgesetzt.

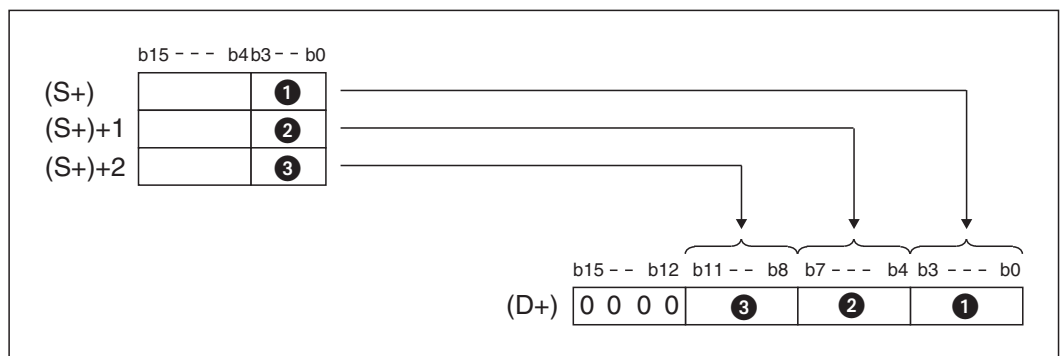


Abb. 7-123: Wenn z. B. für (n) „3“ angegeben wird, werden die Bits 15 bis 12 von (D+) zurückgesetzt.

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Durch die Angabe von (n) wird der zulässige Bereich für den mit (S+) angegebenen Operanden überschritten.
- Für (n) wurde kein Wert zwischen 1 und 4 angegeben.

Beispiel ▽

Beim Einschalten des Eingangs X0 werden die Bits 3 bis 0 der Datenregister D0 bis D2 gelesen und in D10 zusammengefasst. Weil für (n) „3“ angegeben ist, werden die Bits 15 bis 12 von D10 zurückgesetzt.

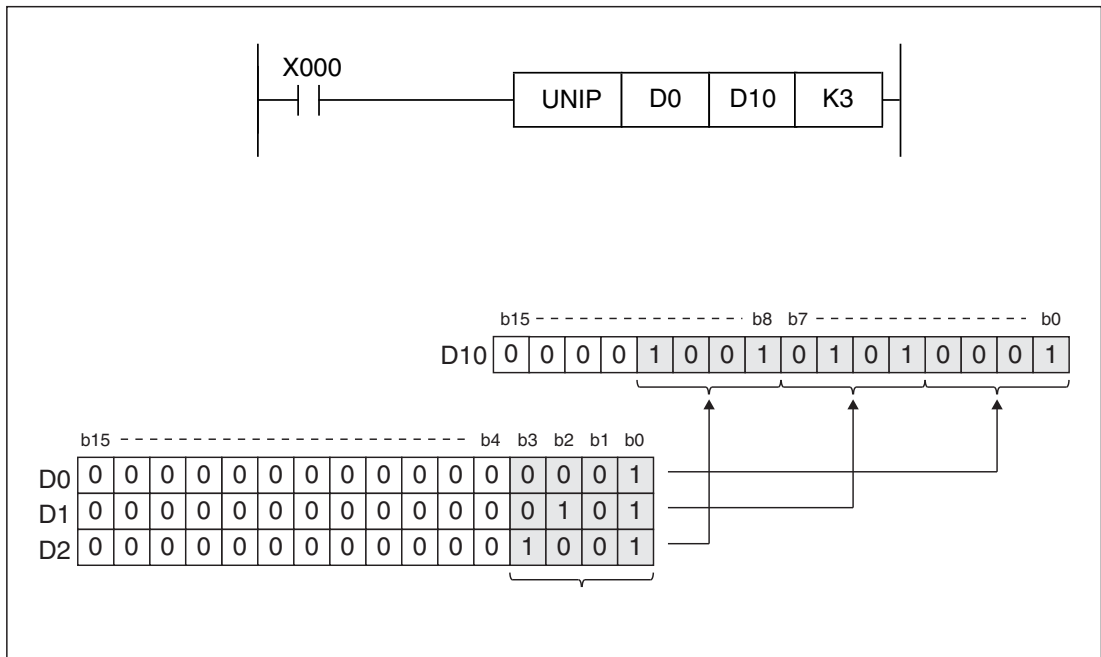


Abb. 7-124: Beispiel für das Gruppieren von 4-Bit-Daten mit einer UNIP-Anweisung



7.8.5 Wortoperanden in Gruppen von 4 Bit aufteilen (DIS)

				DIS		FNC 144				
				Einen Wortoperanden in 4-Bit-Gruppen aufteilen						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
										●
Operanden	S+	D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	T, C, D, R		D, R, K, H	●	16 Bit	32 Bit	DIS DISP		7	

Funktion

Ein Wortoperand wird in bis zu vier 4-Bit-Gruppen aufgeteilt.

Beschreibung

- Die DIS-Anweisung trennt einen Wortoperanden mit 16-Bit-Datenwert in Gruppen zu 4 Bits auf und speichert die Zustände nacheinander in bis zu 4 Zieloperanden ab. In der Anweisung wird der aufzutrennende 16-Bit-Operand in (S+), die Anzahl der 4-Bit-Gruppen in (n) und die erste Zieladresse in (D+) festgelegt. Weitere 4-Bit-Gruppen werden bis zum Operanden ((D+)+n) abgelegt.

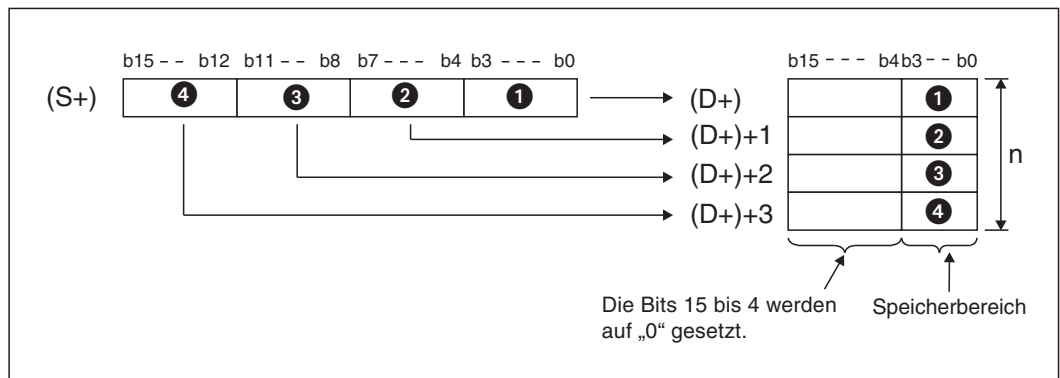


Abb. 7-125: Funktionsweise der DIS-Anweisung

- Für (n) kann ein Wert zwischen 1 und 4 eingestellt werden. Wird für (n) der Wert „0“ angegeben, wird die DIS-Anweisung nicht ausgeführt.
- Die höchstwertigen 12 Bits von n Operanden, beginnend mit der Adresse in (D+), werden auf „0“ gesetzt.

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Durch die Angabe von (n) wird der zulässige Bereich für den mit (D+) angegebenen Operanden überschritten.
- Für (n) wurde kein Wert zwischen 1 und 4 angegeben.

Beispiel ▾

Das folgende Programm trennt beim Einschalten von X0 den 16-Bit-Datenwert aus D0 auf und speichert das Bit-Muster in Gruppen zu 4 Bits nacheinander in D10 bis D13.

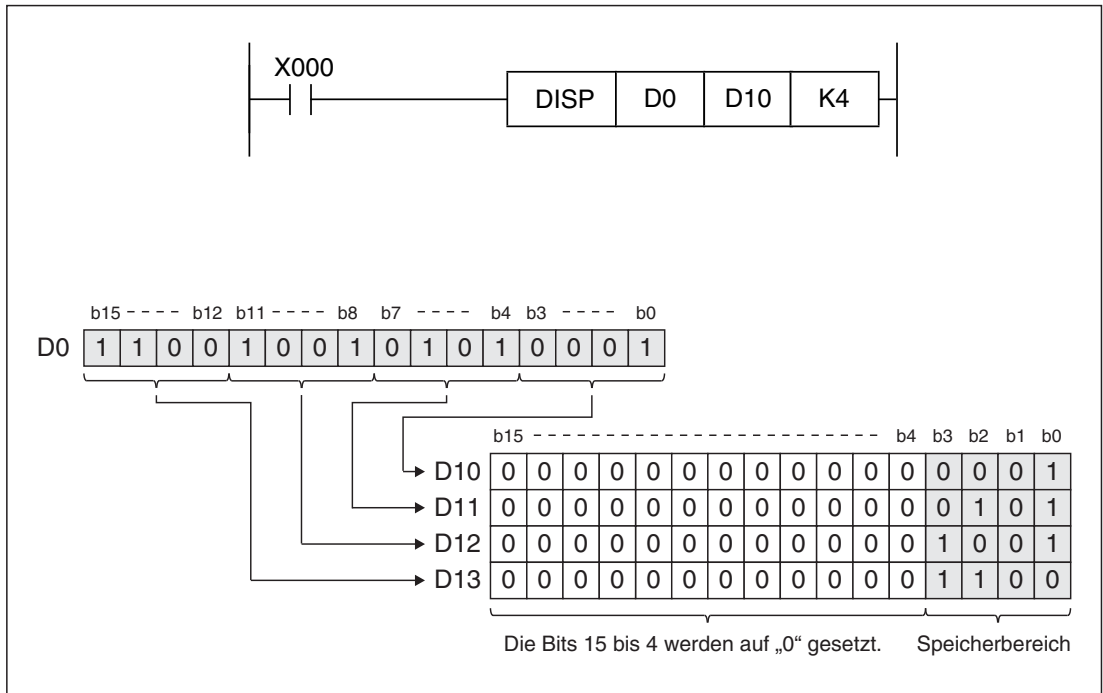


Abb. 7-126: Beispiel zum Trennen von Daten mit einer DISP-Anweisung



7.8.6 High-Low-Byte-Tausch (SWAP)

		SWAP		FNC 147					
		High-Low-Byte-Tausch							
Operanden	S+	Puls-Anweisung (P)	CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
						●			●
Operanden	KnY, KnM, KnS, T, C, D, R ^① , V, Z, U□G□ ^①	●	Verarbeitung		Programmschritte				
			16 Bit	32 Bit	SWAP, SWAPP	5 ^②	DSWAP, DSWAPP	9 ^②	

① Nur bei FX3U und FX3UC

② Bei der FX3U- und FX3UC-Serie werden für die SWAP- und SWAPP-Anweisung 3 Programmschritte und für die DSWAP- und DSWAPP-Anweisung 5 Programmschritte benötigt.

Funktion

Tauschen des High- und Low-Bytes eines Operanden

Beschreibung

- Bei Verwendung der SWAP-Anweisung werden das High- und Low-Byte des Operanden (D+) getauscht.
- Bei Verwendung der DSWAP-Anweisung werden jeweils die High- und Low-Bytes der Operanden (D+) und ((D+)+1) getauscht.
- Diese Operation wird in jedem Programmzyklus erneut ausgeführt. Um eine einmalige Ausführung zu gewährleisten, sind gepulste Anweisungen zu verwenden oder Verriegelungen einzusetzen.

HINWEIS

Die Funktion der SWAP-Anweisung entspricht der XCH-Anweisung bei gesetztem Sondermerker M8160 (siehe Abschnitt 6.3.8).

Beispiel ▾

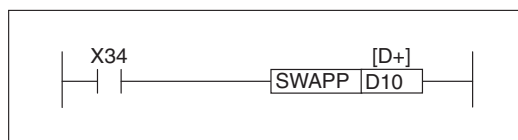


Abb. 7-127:
Programmierbeispiel zur SWAPP-Anweisung

Mit steigender Flanke des Eingangs X34 werden das High- und Low-Byte von D10 getauscht.

Operand	Byte	Vor Ausführen der Anweisung	Nach Ausführen der Anweisung
D10	1	1FH	8BH
	2	8BH	1FH

Tab. 7-24:
Byte-Tausch mit der SWAPP-Anweisung

Wird statt der SWAPP-Anweisung die DSWAPP-Anweisung verwendet, werden mit steigender Flanke von X34 jeweils die High- und Low-Bytes in D10 und D11 getauscht.

Operand	Byte	Vor Ausführen der Anweisung	Nach Ausführen der Anweisung
D10	1	1FH	8BH
	2	8BH	1FH
D11	1	C4H	35H
	2	35H	C4H

Tab. 7-25:
Byte-Tausch bei der DSWAPP-Anweisung



7.8.7 Daten in Tabelle sortieren (SORT2)

						SORT2		FNC149				
						Sortieranweisung						
Operanden						CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
						Puls-Anweisung (P)		Verarbeitung		Programmschritte		
								16 Bit	32 Bit	SORT2	11	
								●	●	DSORT2	21	

- ① ((n1) x (n2)) aufeinanderfolgende Datenregister
- ② (n1) gibt die Anzahl der Zeilen der Tabelle an (1 bis 32).
- ③ (n2) gibt die Anzahl der Spalten der Tabelle an (1 bis 6).
- ④ (m) ist das Sortierkriterium (Nummer der Spalte); Für (m) können Werte von 1 bis zum Wert von (n2) angegeben werden.

Funktionsweise

Sortieren einer Tabelle nach den Werten einer Spalte in auf- oder absteigender Reihenfolge

Die Sortierreihenfolge wird durch den Sondermerker M8156 bestimmt:

- M8156 = 0: Aufsteigende Reihenfolge
- M8156 = 1: Absteigende Reihenfolge

Beschreibung (Ausführung für 16-Bit-Daten)

- Bei der Ausführung der SORT2-Anweisung wird eine interne Datenmatrix (Tabelle), gekennzeichnet durch das Start-Datenregister (S+) mit der Größe von (n1) Zeilen und (n2) Spalten, nach den Werten in Spalte (m) sortiert und ab Datenregister (D+) neu abgelegt.

Beispiel ▾

Die folgende Tabelle mit drei Zeilen und vier Spalten enthält Daten von Personen:

Spalte Nr.	1	2	3	4
Zeile Nr.	ID-Nummer	Größe	Gewicht	Alter
1	(S+)	((S+)+1)	((S+)+2)	((S+)+3)
	1	150	45	20
2	((S+)+4)	((S+)+5)	((S+)+6)	((S+)+7)
	2	180	50	40
3	((S+)+8)	((S+)+9)	((S+)+10)	((S+)+11)
	3	160	70	30

Tab. 7-26: Unsortierte Tabelle

Nach der Ausführung einer SORT2-Anweisung (n1 = K3, n2 = K4, m = K2) wird die Tabelle nach den Werten in Spalte 2 in aufsteigender Reihenfolge sortiert (M8156 = 1):

Spalte Nr.	1	2	3	4
Zeile Nr.	ID-Nummer	Größe	Gewicht	Alter
1	(D+)	((D+)+1)	((D+)+2)	((D+)+3)
	1	150	45	20
2	((D+)+4)	((D+)+5)	((D+)+6)	((D+)+7)
	3	160	70	30
3	((D+)+8)	((D+)+9)	((D+)+10)	((D+)+11)
	2	180	50	40

Tab. 7-27: Sortierte Tabelle nach Ausführung einer SORT2-Anweisung

Beschreibung (Ausführung für 32-Bit-Daten)

- Bei der Ausführung der SORT2-Anweisung wird eine interne Datenmatrix (Tabelle), gekennzeichnet durch die Start-Datenregister ((S+)+1) und (S+) mit der Größe von (n1) Zeilen und (n2) Spalten, nach den Werten in Spalte (m) sortiert und ab ((D+)+1) und (D+) neu abgelegt.

Beispiel ▾

Die folgende Tabelle mit drei Zeilen und vier Spalten enthält Rezepturen:

Spalte Nr.	1	2	3	4
Zeile Nr.	Datensatz	Wasser	Mehl	Zucker
1	((S+)+1), (S+)	((S+)+3), ((S+)+2)	((S+)+5), ((S+)+4)	((S+)+7), ((S+)+6)
	1	56879	27478	16890
2	((S+)+9), ((S+)+8)	((S+)+11), ((S+)+10)	((S+)+13), ((S+)+12)	((S+)+15), ((S+)+14)
	2	44878	21388	15722
3	((S+)+17), ((S+)+16)	((S+)+19), ((S+)+18)	((S+)+21), ((S+)+20)	((S+)+23), ((S+)+22)
	3	23898	11999	18743

Tab. 7-28: Unsortierte Tabelle

Nach der Ausführung einer DSORT2-Anweisung (n1 = K3, n2 = K4, m = K2) wird die Tabelle nach den Werten in Spalte 2 in aufsteigender Reihenfolge sortiert (M8156 = 1):

Spalte Nr.	1	2	3	4
Zeile Nr.	Datensatz	Wasser	Mehl	Zucker
1	((D+)+1), (D+)	((D+)+3), ((D+)+2)	((D+)+5), ((D+)+4)	((D+)+7), ((D+)+6)
	3	23898	11999	18743
2	((D+)+9), ((D+)+8)	((D+)+11), ((D+)+10)	((D+)+13), ((D+)+12)	((D+)+15), ((D+)+14)
	2	44878	21388	15722
3	((D+)+17), ((D+)+16)	((D+)+19), ((D+)+18)	((D+)+21), ((D+)+20)	((D+)+23), ((D+)+22)
	1	56879	27478	16890

Tab. 7-29: Sortierte Tabelle nach Ausführung einer DSORT2-Anweisung

- Wenn für (n1) ein Datenregister (D) oder ein File-Register (R) angegeben wird, beträgt die Datenlänge 32 Bits. Wenn zum Beispiel für (n1) „D0“ angegeben wird, ist die Anzahl der Zeilen in D1 und D2 als 32-Bit-Wert gespeichert.



HINWEISE

Die SORT2-Anweisung darf maximal zweimal in einem Programm verwendet werden.

Die Sortierung beginnt, wenn die Eingangsbedingung der SORT2-Anweisung erfüllt ist und ist nach Ablauf von (n1) Programmzyklen abgeschlossen. Nach Abschluss der SORT2-Anweisung wird der Sondermerker M8029 gesetzt.

Um eine SORT2-Anweisung noch einmal auszuführen, muss zuvor die Eingangsverknüpfung der Anweisung zurückgesetzt werden.

Ein Programm, in dem eine SORT2-Anweisung enthalten ist, kann nicht im RUN-Modus der SPS in die Steuerung übertragen oder geändert werden.

Wenn in (S+) und (D+) dieselben Operanden angegeben werden, werden die Quelldaten durch die sortierten Daten überschrieben.

Während eines Sortiervorgangs dürfen die Daten in der Tabelle nicht verändert werden, da andernfalls fehlerhafte Daten abgelegt werden.

Beispiel ▾ Sortieren einer Tabelle mit 5 Zeilen und 4 Spalten

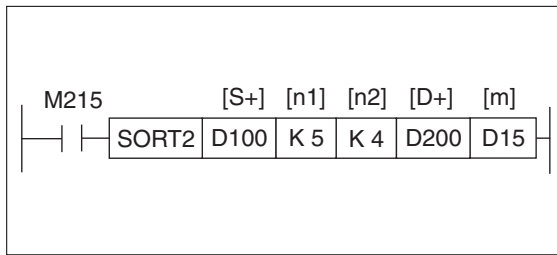


Abb. 7-128:
 Programmierbeispiel zum Einsatz der SORT"-Anweisung
 In D15 wird an anderer Stelle im Programm die Nummer der Spalte eingetragen, nach deren Werte die Tabelle sortiert werden soll.

Spalte	1	2	3	4
Zeile	Nr.	Wert 1	Wert 2	Wert 3
1	D100	D101	D102	D103
	1	150	45	20
2	D104	D105	D106	D107
	2	180	50	40
3	D108	D109	D110	D111
	3	160	70	30
4	D112	D113	D114	D115
	4	100	20	8
5	D116	D117	D118	D119
	5	150	35	45

Tab. 7-30:
 Unsortierte Tabelle
 Zur besseren Übersicht sollte in der ersten Spalte eine Kontroll-Nr. vorgesehen werden, um nach der Sortierung die ursprüngliche Zeile leichter identifizieren zu können.

Spalte	1	2	3	4
Zeile	Nr.	Wert 1	Wert 2	Wert 3
1	D200	D201	D202	D203
	4	100	20	8
2	D204	D205	D206	D207
	1	150	45	20
3	D208	D209	D210	D211
	5	150	35	45
4	D212	D213	D214	D215
	3	160	70	30
5	D216	D217	D218	D219
	2	180	50	40

Tab. 7-31:
 Sortierte Tabelle
 Es wurde in aufsteigender Reihenfolge nach den Inhalten von Spalte 2 sortiert (m = Inhalt von D15 = K2).

Spalte	1	2	3	4
Zeile	Nr.	Wert 1	Wert 2	Wert 3
1	D200	D201	D202	D203
	3	160	70	30
2	D204	D205	D206	D207
	2	180	50	40
3	D208	D209	D210	D211
	1	150	45	20
4	D212	D213	D214	D215
	5	150	35	45
5	D216	D217	D218	D219
	4	100	20	8

Tab. 7-32:
 Sortierte Tabelle
 Es wurde in absteigender Reihenfolge nach den Inhalten von Spalte 3 sortiert (m = Inhalt von D15 = K3).



7.9 Positionieranweisungen

Durch die Verwendung der Positionieranweisungen kann eine SPS der FX1S-/FX1N- oder FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC-Serie in Verbindung mit MITSUBISHI Servoverstärkern zur Positionierung eingesetzt werden.

HINWEIS

Die Positionierung mit einer FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC und die Anwendung der folgenden Anweisungen ist im Positionierhandbuch zur FX3-Serie ausführlich beschrieben. Die Beispiele in diesem Abschnitt beziehen sich daher auf Steuerungen der FX1S- oder FX1N-Serie.

Übersicht der Anweisungen FNC 150 bis 159

Symbol	FNC	Bedeutung	Abschnitt
DSZR	150	Referenzpunktfahrt mit Annäherungsschalter	7.9.6
DVIT	151	Positionierung durch Interrupt	7.9.7
TBL	152	Positionierung nach Datentabelle	7.9.8
ABS	155	Lesen der absoluten Ist-Position	7.9.9
ZRN	156	Referenzpunkt anfahren	7.9.10
PLSV	157	Ausgabe von Impulsen mit variabler Frequenz	7.9.11
DRVI	158	Positionieren auf einen Inkrementalwert	7.9.12
DRVA	159	Positionieren auf einen Absolutwert	7.9.13

Tab. 7-33: Übersicht der Positionieranweisungen

7.9.1 Hinweise zum Einsatz der Positionieranweisungen bei FX1S und FX1N

Die Positionieranweisungen ZRN (FNC 156), PLSV (FNC 157), DRVI (FNC 158) und DRVA (FNC 159) verwenden zur Impulsausgabe die Ausgänge Y0 und Y1.

Diese Anweisungen können beliebig oft im Programm verwendet werden, beachten Sie jedoch bei der Programmierung die folgenden Hinweise:

- Jeweils nur eine der Anweisungen ZRN, PLSV, DRVI und DRVA darf in einem Programmzyklus ausgeführt werden. Falls mehrere dieser Anweisungen gleichzeitig ausgeführt werden, werden die Ausgänge Y0 und Y1 mehrfach angesprochen und die korrekte Funktion ist nicht mehr gewährleistet.
- Verwenden Sie die Sondermerker M8147 und M8148 zur Verriegelung im Programm. Mit diesen Sondermerkern wird der Zustand der Ausgänge Y0 und Y1 überwacht. Eine Positionieranweisung darf erst gestartet werden, wenn die Sondermerker M8147 (Überwachung von Y0) und M8148 (Überwachung von Y1) nach der Ausführung einer Positionieranweisung für mindestens einen Programmzyklus den Zustand „0“ aufgewiesen haben.

Kombination der Positionieranweisungen mit Anweisungen zur Impulsausgabe

- Bei den Anweisungen FNC 57 (PLSY) und FNC 59 (PLSR) werden ebenfalls die Ausgänge Y0 und Y1 zur Impulsausgabe verwendet.
- Werden Anweisungen zur Positionierung und zur Impulsausgabe im einem Programmzyklus gleichzeitig ausgeführt, werden die Ausgänge Y0 und Y1 mehrfach angesprochen und die korrekte Funktion ist nicht mehr gewährleistet.
- Verwenden Sie statt den Anweisungen FNC 57 (PLSY) und FNC 59 (PLSR) die Anweisung FNC 158 (DRVI), wenn zur Positionierung eine Impulsausgabe notwendig ist.

Einsetzbare Steuerungen

Da die Impulse mit hoher Frequenz ausgegeben werden, müssen Steuerungen mit Transistorausgängen verwendet werden. Relaiskontakte nutzen sich in diesem Fall vorzeitig ab und sind daher ungeeignet.

Zur Erzeugung steilflankiger Ausgangssignale sollte der Laststrom der Transistorausgänge zwischen 10 und 100 mA liegen. Eventuell ist die Verwendung von Pull-up-Widerständen notwendig.

Technische Daten der Transistorausgänge Y0 und Y1 bei FX1S und FX1N		
Nennschaltleistung	Spannung	5 bis 24 V DC
	Strom	10 bis 100 mA
Maximale Schaltfrequenz		100 kHz

7.9.2 Impulsausgabe an den Servoverstärker

Die Signale an einen angeschlossenen Servoverstärker werden als Impulskette ausgegeben. Die Drehrichtung wird durch einen zusätzlichen Ausgang festgelegt.

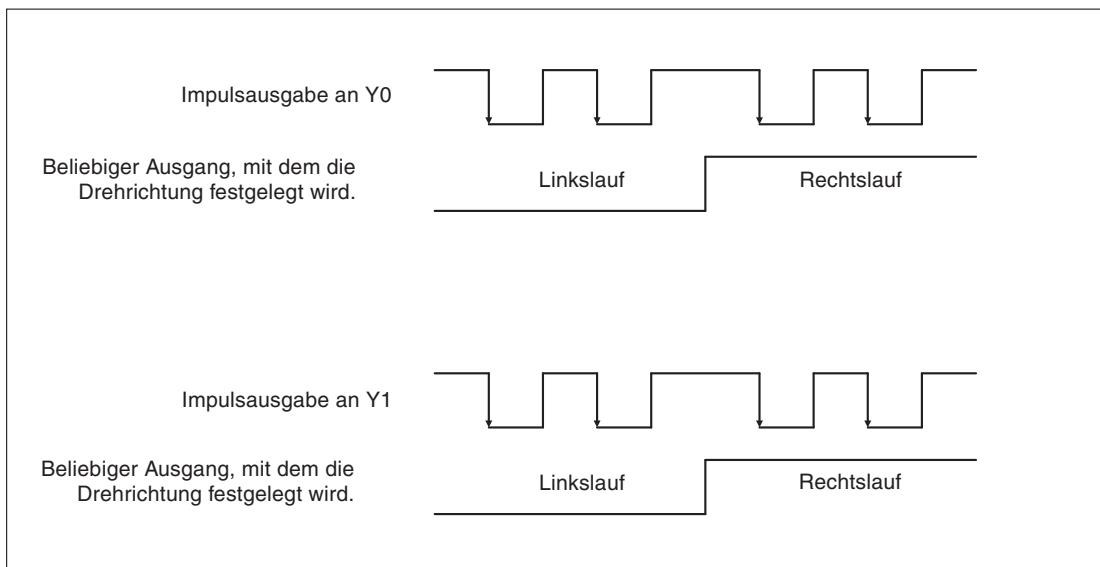


Abb. 7-129: Form der Ausgangssignale an die Servoverstärker

Wählen Sie am Servoverstärker oder Schrittmotor die folgenden Einstellungen:

- Impulsketten-Eingang: Impulskette und Vorzeichen
- Impulsketten-Logik: Negative Logik

7.9.3 Operanden für die Positionierung

Operand	Zugriff	Beschreibung
M8145	Merker können im Anwenderprogramm gesetzt werden.	Impulsausgabe an Y0 sofort stoppen
M8146		Impulsausgabe an Y1 sofort stoppen
M8147	Merker dürfen nur abgefragt werden. Bei zurückgesetztem Merkern ist die Impulsausgabe ausgeschaltet.	Überwachung der Impulsausgabe an Y0
M8148		Überwachung der Impulsausgabe an Y1

Tab. 7-35: Sondermerker für die Positionieranweisungen

Operand		Vorgabewert	Beschreibung
D8140	Niederwertiges Wort	0	Istwert Y0 (32 Bit) ^①
D8141	Höherwertiges Wort		
D8142	Niederwertiges Wort	0	Istwert Y1 (32 Bit) ^①
D8143	Höherwertiges Wort		
D8145		0	Frequenz-Offset bei der Verwendung von FNC 158 oder FNC 159 Wertebereich: max. 1/10 der max. Frequenz (D8146, D8147) Werden größere Werte angegeben, wird der Drehzahl-Offset automatisch auf 1/10 der max. Frequenz begrenzt.
D8146	Niederwertiges Wort	100000	Maximale Frequenz der Ausgangsimpulse bei der Verwendung von FNC 158 oder FNC 159 (32 Bit) Bereich: 100 Hz bis 100 kHz
D8147	Höherwertiges Wort		
D8148		100	Beschleunigungs- und Verzögerungszeit [ms] bei der Verwendung von FNC 156, FNC 158 oder FNC 159 ^② Wertebereich: 50 bis 5000 ms

Tab. 7-34: Sonderregister für die Positionieranweisungen bei FX1S und FX1N

- ① Bei den Anweisungen FNC 157 (PLSV), FNC 158 (DRVI) und FNC 159 (DRVA) wird der Istwert abhängig von der Drehrichtung erhöht oder vermindert.
Von den Anweisungen FNC 57 (PLSY) und FNC 59 (PLSR) werden die Sonderregister D8140/D8141 und D8142/D8143 ebenfalls benutzt. In diesem Fall wird in diesen Registern aber die Summe der an Y0/Y1 während der Ausführung der Anweisung ausgegebenen Impulse abgelegt.
- ② Die Beschleunigungszeit ist die Zeit, die vergeht, wenn vom Frequenz-Offset (D8145) bis zur max. Frequenz (D8146, D8147) beschleunigt wird.
Die Verzögerungszeit ist die Zeit, die vergeht, wenn der max. Frequenz (D8146, D8147) zum Wert des Frequenz-Offset (D8145) verzögert wird.

7.9.4 Anschluss an einem Servoverstärker

Die folgende Abbildung zeigt den Anschluss einer FX1s-30MT (minusschaltend) an einen Servoverstärker MELSERVO MR-J2-□A:

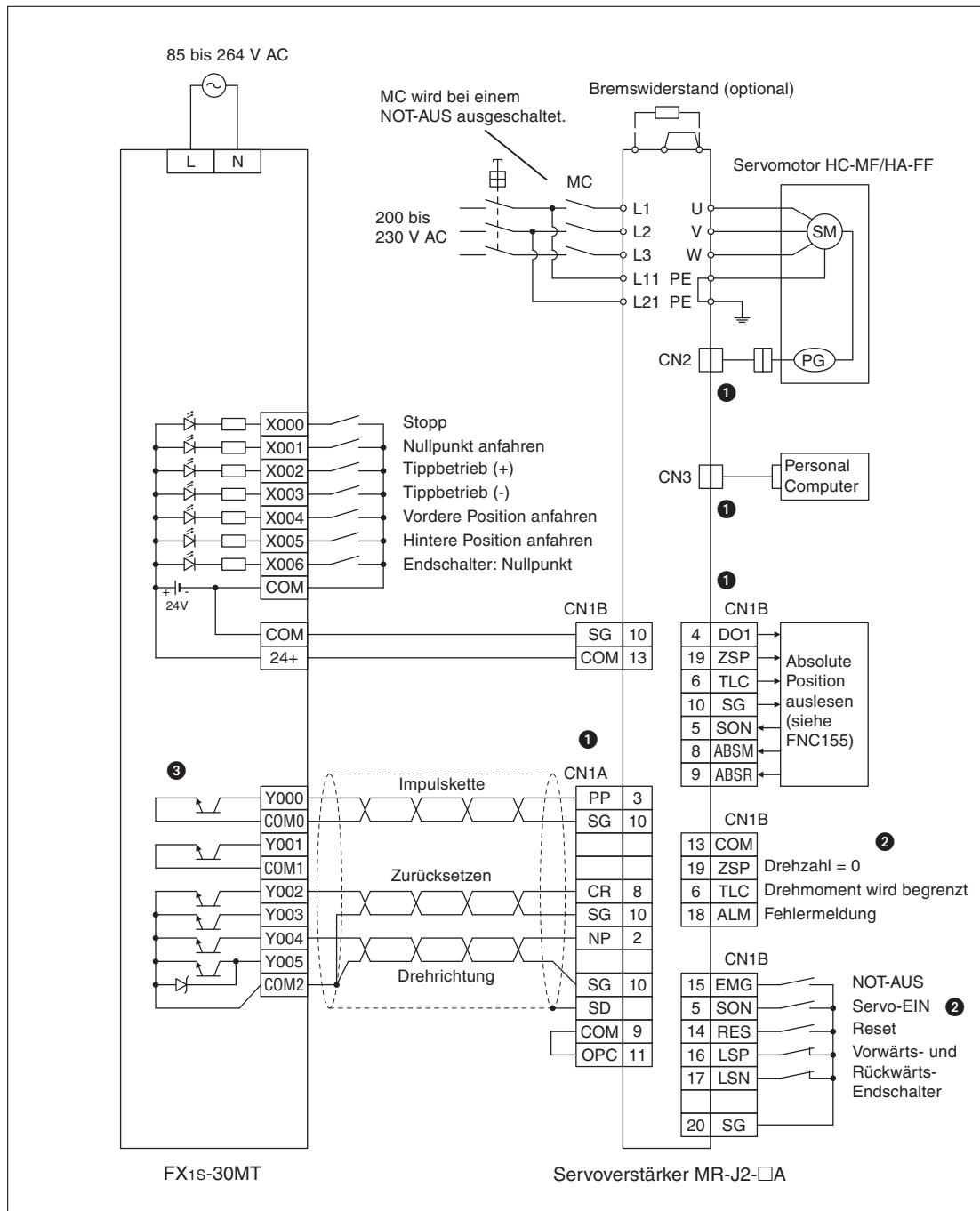


Abb. 7-130: Beispiel zum Anschluss eines Servoverstärkers

- Die äußere Form der Anschlüsse CN1A, CN1B, CN2 und CN3 ist identisch. Bitte verwechseln sie diese nicht.
- Der Anschluss dieser Signale ist für das Anfahren absoluter Positionen erforderlich.
- Verwenden Sie unbedingt eine SPS mit Transistorausgängen.

7.9.5 Beispielprogramm

In diesem Beispiel wird auf Absolutwerte positioniert:

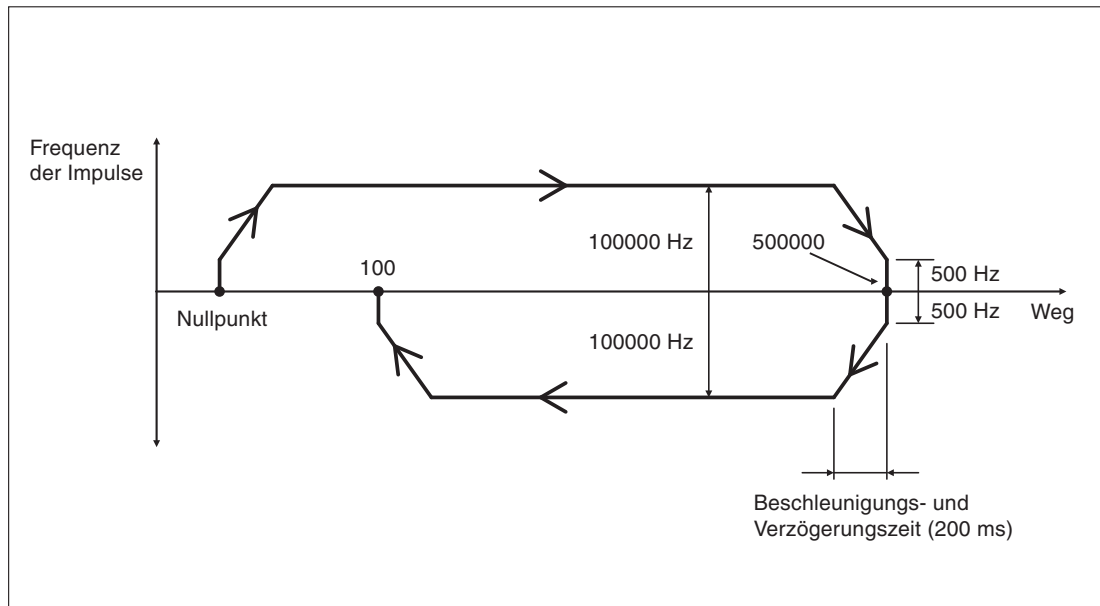


Abb. 7-131: Fahrtdiagramm für das Programmbeispiel

Berechnung der minimalen Frequenz:

$$f_{min} = \sqrt{\frac{f_{max}}{2 \times t_B}}$$

f_{max} ist die in D8146 und D8147 gespeicherte max. Frequenz der Ausgangsimpulse [Hz].
 t_B ist Beschleunigungs- und Verzögerungszeit in der Einheit Sekunden.

Für das Beispiel ergibt sich mit $f_{max} = 100 \text{ kHz}$ und $t_B = 0,2 \text{ s}$ eine minimale Frequenz von

$$f_{min} = \sqrt{\frac{100000}{2 \times 0,2}} = 500 \text{ Hz}$$

Die SPS wird wie in Abschnitt 7.9.4 dargestellt mit dem Servoverstärker verbunden.

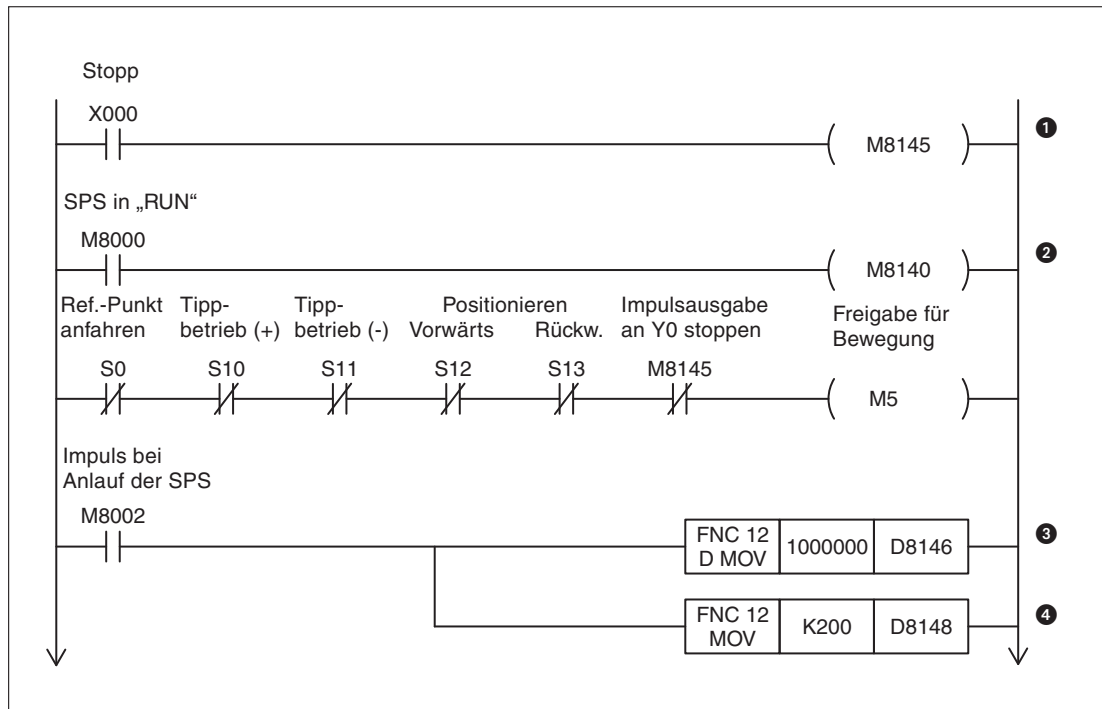


Abb. 7-132: Programmbeispiel (Teil 1: Initialisierung)

- ❶ Die Impulsausgabe für die X-Achse (Y0) wird gestoppt.
- ❷ Wenn M8140 gesetzt ist, wird dem Servoverstärker das Erreichen des Referenzpunktes gemeldet,
- ❸ Die max. Frequenz von 100.000 Hz wird in D8147 und D8146 eingetragen.
- ❹ In D8148 wird die Beschleunigungs- und Verzögerungszeit von 200 ms eingetragen.

HINWEIS

Wenn für die maximale Frequenz, die Beschleunigungszeit und die Verzögerungszeit die Vorgabewerte verwendet werden, ist der Eintrag dieser Werte in die Register D8146 bis D8148 nicht nötig.

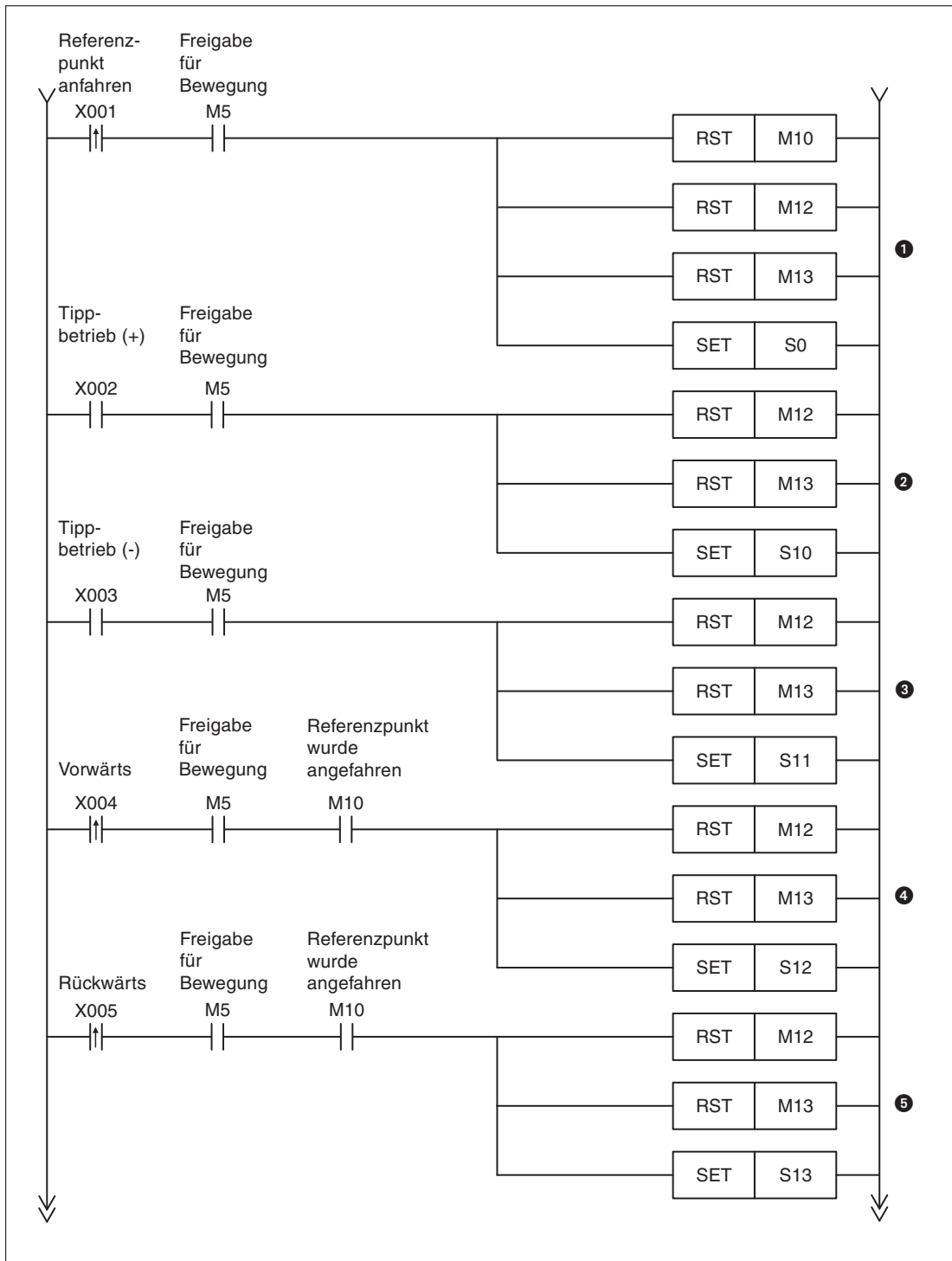


Abb. 7-133: Programmbeispiel (Teil 2: Fahrbefehle generieren)

- ① Bei Betätigung des Tasters „Referenzpunkt anfahren“ werden die Merker M10 („Referenzpunkt wurde angefahren“), M12 („Vorwärts-Positionierung beendet“) und M13 („Rückwärts-Positionierung beendet“) zurückgesetzt. Der Schrittmerker S0 („Referenzpunkt anfahren“) wird gesetzt.
- ② Beim Tippbetrieb in Vorwärtsrichtung werden die Merker M12 und M13, die nach einer Positionierung gesetzt werden, gelöscht, und der Schrittmerker S10 („Tippbetrieb vorwärts“) wird gesetzt.
- ③ Bei Betätigung des Tasters, der am Eingang X3 angeschlossen ist, wird der Schrittmerker S11 („Tippbetrieb rückwärts“) gesetzt. Die Merker M12 und M13, die das Ende einer Positionierung anzeigen, werden gelöscht.
- ④ Beim Start einer Vorwärtsbewegung werden M12 („Positionierung in Vorwärtsrichtung abgeschlossen“) und M13 („Positionierung in Rückwärtsrichtung abgeschlossen“) zurückgesetzt, und der Schrittmerker S12 („Positionieren in Vorwärtsrichtung“) wird gesetzt.
- ⑤ Der Schrittmerker S13 („Positionieren in Rückwärtsrichtung“) wird gesetzt, und die Merker M12 („Vorwärts-Positionierung beendet“) und M13 („Rückwärts-Positionierung beendet“) werden zurückgesetzt, wenn der an X5 angeschlossene Taster betätigt wird.

HINWEIS

Beim Tippbetrieb können mit einer Anweisung max. 999.999 Impulse (max. Wert bei 32-Bit-Operanden für die DRVI-Anweisung) ausgegeben werden. Wenn ein größerer Weg erforderlich ist, muss die DRVI-Anweisung mehrmals ausgeführt werden.

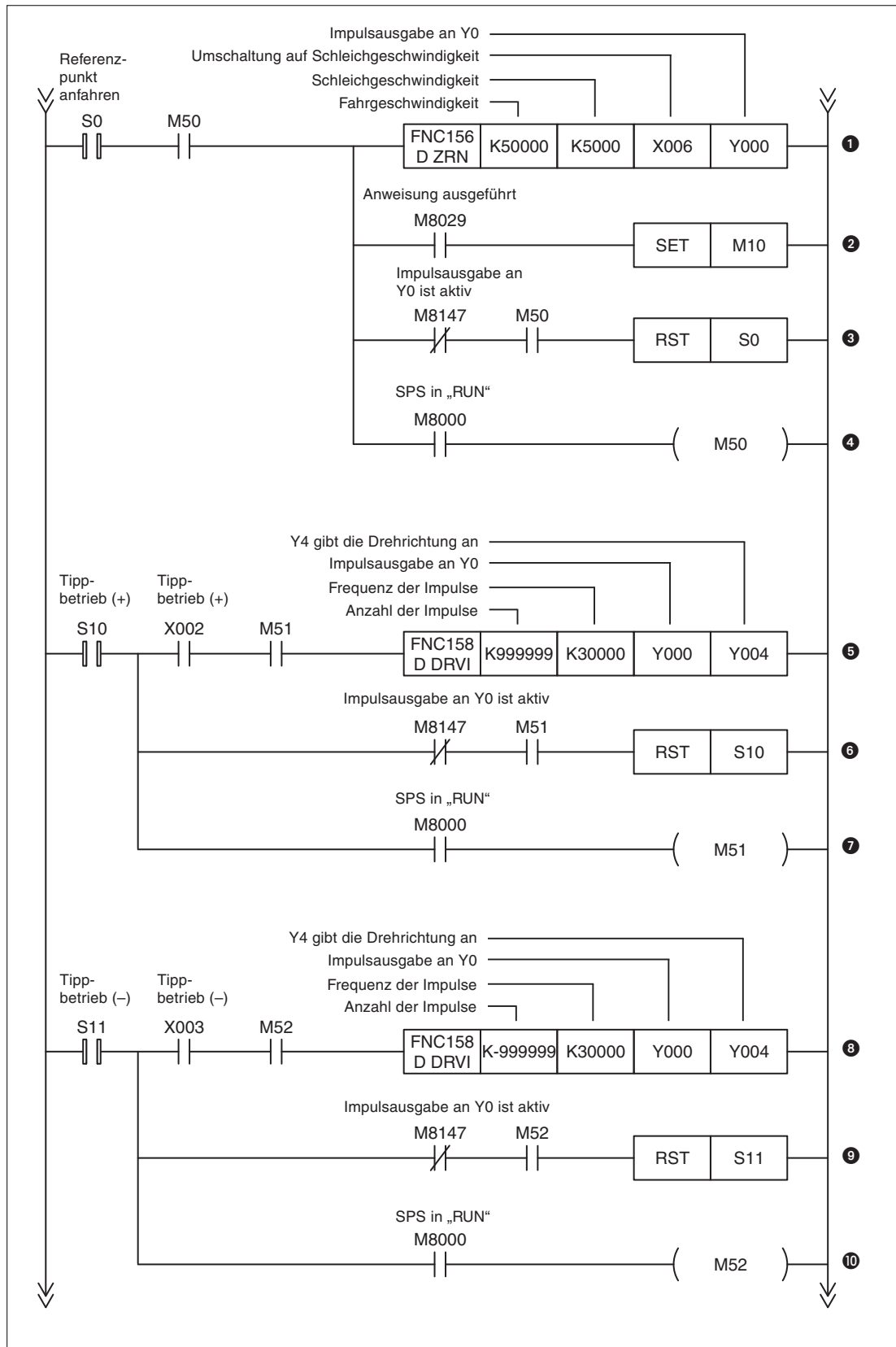


Abb. 7-134: Beispielprogramm (Teil 3: Referenzpunktfahrt und Tippbetrieb)

- ① Der Referenzpunkt (Nullpunkt) wird angefahren, indem in Richtung kleinerer Positionswerte gefahren wird.
- ② M10 zeigt an, dass der Referenzpunkt erreicht wurde.
- ③ Wenn keine Impulse mehr ausgegeben werden, wird S0 zurückgesetzt.
- ④ Nach dem Setzen von S0 wird die ZRN-Anweisung durch M50 für einen Zyklus verzögert, um die gleichzeitige Bearbeitung von mehreren Positionieranweisungen zu verhindern.
- ⑤ Vorwärtsbewegung im Tippbetrieb (die Anzahl der Impulse ist positiv). Y4 ist gesetzt.
- ⑥ S10 wird zurückgesetzt, wenn keine Impulse mehr ausgegeben werden.
- ⑦ Um die gleichzeitige Bearbeitung von mehreren Positionieranweisungen zu verhindern, wird die Ausführung der DRVI-Anweisung einen Zyklus verzögert.
- ⑧ Rückwärtsbewegung im Tippbetrieb (die Anzahl der Impulse ist negativ). Y4 ist nicht gesetzt.
- ⑨ Wenn keine Impulse mehr ausgegeben werden, wird S11 zurückgesetzt.
- ⑩ Verzögerung der DRVI-Anweisung

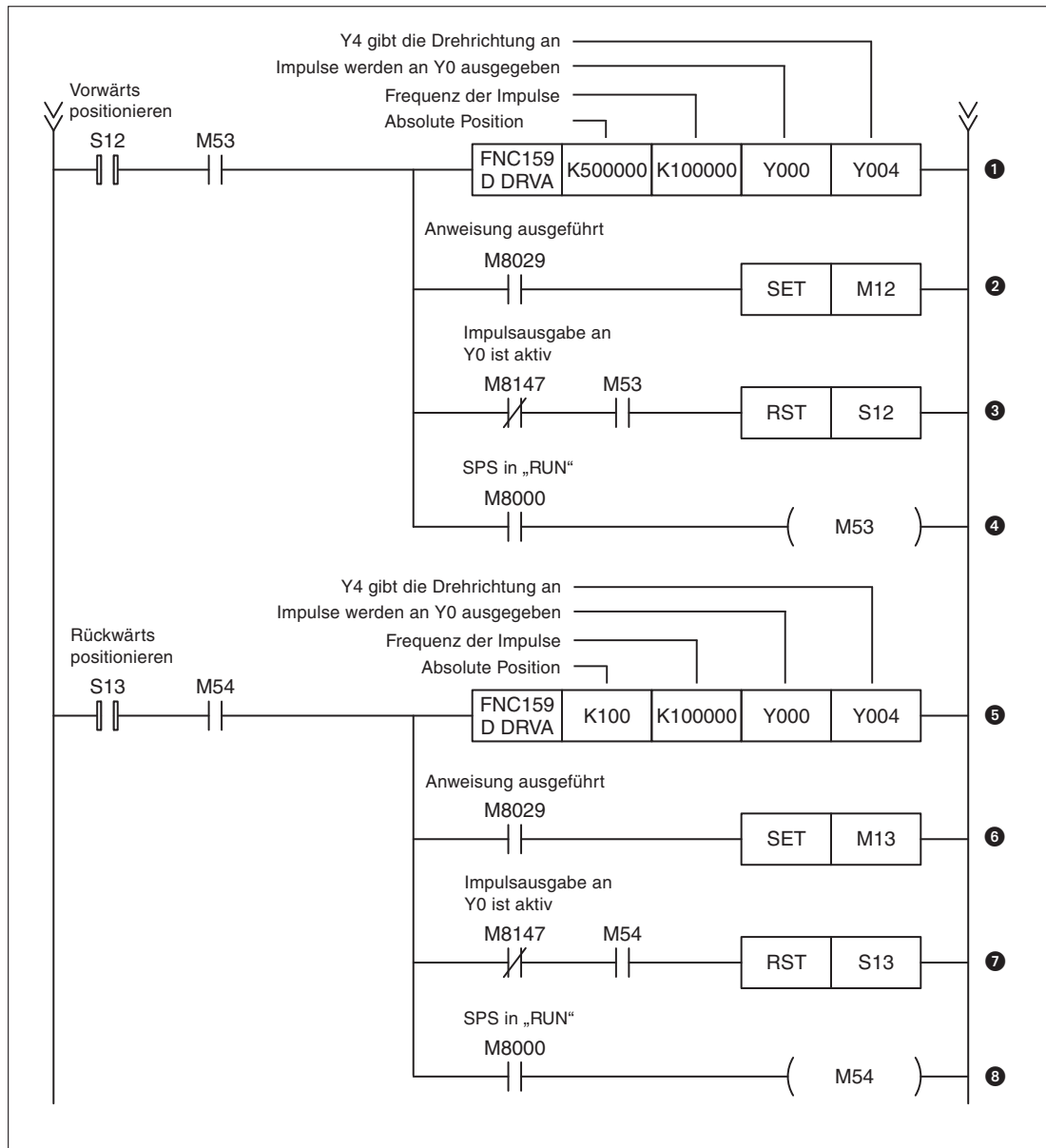


Abb. 7-135: Beispielprogramm (Teil 4, Vorwärts- und Rückwärtsposition anfahren)

- ① Positionierung auf den absoluten Wert „500000“. Y4 ist gesetzt.
- ② M12 zeigt an, dass die Position erreicht wurde.
- ③ Wenn keine Impulse mehr ausgegeben werden, wird S12 zurückgesetzt.
- ④ Nach dem Setzen von S12 wird die DRVA-Anweisung durch M53 für einen Zyklus verzögert, um die gleichzeitige Bearbeitung von mehreren Positionieranweisungen zu verhindern.
- ⑤ Positionierung auf den absoluten Wert „100“. Y4 ist nicht gesetzt.
- ⑥ M13 zeigt an, dass die Position erreicht wurde.
- ⑦ S13 wird zurückgesetzt, wenn keine Impulse mehr ausgegeben werden.
- ⑧ Um die gleichzeitige Bearbeitung von mehreren Positionieranweisungen zu verhindern, verzögert M54 die DRVA-Anweisung für einen Zyklus.

7.9.6 Referenzpunktfahrt mit Annäherungsschalter (DSZR)

					DSZR		FNC 150				
					Referenzpunktfahrt mit Annäherungsschalter						
					CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
									●	●	●
Operanden	S1+	S2+	D1+	D2+	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	X, Y, M, S, D□.b ^①	X ^②	Y ^③	Y ^④ , M, S, D□.b ^①		16 Bit	32 Bit	DSZR		9	
						●					

- ① Nur bei FX3U und FX3UC
- ② Ein Eingang aus dem Bereich X0 bis X7. (Bei einem FX3S-Grundgerät: Ein Eingang aus dem Bereich X0 bis X5.)
- ③ Nur Transistorausgänge Y0, Y1 oder Y2 im Grundgerät oder Y0, Y1, Y2 oder Y3 eines HIGH-SPEED-Ausgabemoduls FX3U-2HSY-ADP (nur an FX3U-Grundgeräte anschließbar). Die Impulsausgabe an Y2 des Grundgeräts ist nur bei einer FX3U oder FX3UC sowie einer FX3G-40M□, FX3G-60M□ oder FX3GE-40M□ möglich. Für die Angabe von Y2 oder Y3 bei einem HIGH-SPEED-Ausgabemodul wird ein zweites HIGH-SPEED-Ausgabemodul benötigt.
- ④ Falls nicht die Ausgänge eines HIGH-SPEED-Ausgabemoduls FX3U-2HSY-ADP verwendet werden, muss ein Transistorausgang angegeben werden.

Funktion

Mit dieser Anweisung wird eine Fahrt zum Referenzpunkt (Nullpunkt) eingeleitet. Zusätzlich zur ZRN-Anweisung (Abschnitt 7.9.10) kann mit der DSZR-Anweisung ein Annäherungsschalter gesucht und der Referenzpunkt durch diesen Schalter und das Nullphasensignal bestimmt werden. Das Nullphasensignal kann jedoch nicht gezählt und dadurch der Nullpunkt ermittelt werden.

Beschreibung

- In (S1+) wird der Operand angegeben, der den Status des Schalters zur Einleitung der Schleichfahrt in der Nähe des Referenzpunktes enthält.
- (S2+) enthält den Eingang, an dem das Nullphasensignal angeschlossen ist.
- (D1+) gibt den Ausgang zur Impulsausgabe an.

Zur Ausgabe der Impulse können nur bestimmte Ausgänge (siehe oben) verwendet werden. Da die Impulse mit hoher Frequenz ausgegeben werden, müssen Steuerungen mit Transistorausgängen verwendet werden. Relaiskontakte nutzen sich in diesem Fall vorzeitig ab und sind daher ungeeignet. Bei der FX3U-Serie können für das erste Ausgabemodul FX3U-2HSY-ADP die Ausgänge Y0 und Y1 und für das Zweite die Ausgänge Y2 und Y3 angegeben werden. Die Zustände von Y4 und Y5 bzw. Y6 und Y7 bestimmen die Drehrichtung.

- In D2+ wird ein Operand angegeben, der dem Servoverstärker die Drehrichtung vorgibt.

HINWEISE

Ein Programm, das eine DSZR-Anweisung enthält, sollte nicht im RUN-Modus der SPS geändert werden, während diese Anweisung ausgeführt wird. Wird dies nicht beachtet, wird der Servoantrieb verzögert und gestoppt.

Die Positionierung mit einer FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC und die Anwendung der DSZR-Anweisung ist im Positionierhandbuch zur FX3-Serie ausführlich beschrieben.

7.9.7 Positionierung durch Interrupt (DVIT)

					DVIT		FNC 151			
					Positionierung durch Interrupt					
	CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC		●	
Operanden	S1+	S2+	D1+	D2+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	K, H, KnX, KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□), V, Z		Y ①	Y ②, M, S, D□.b			16 Bit	32 Bit	DVIT	9
						●	●	DDVIT	17	

- ① Nur Transistorausgänge Y0, Y1 oder Y2 im Grundgerät oder Y0, Y1, Y2 oder Y3 eines HIGH-SPEED-Ausgabemoduls FX3U-2HSY-ADP (nur an FX3U-Grundgeräte anschließbar.) Für die Angabe von Y2 oder Y3 wird ein zweites HIGH-SPEED-Ausgabemodul benötigt.
- ② Falls nicht die Ausgänge eines HIGH-SPEED-Ausgabemoduls FX3U-2HSY-ADP verwendet werden, muss ein Transistorausgang angegeben werden.

Funktion

Mit dieser Anweisung wird nach einem Interrupt eine Position mit einer konstanten Geschwindigkeit angefahren.

Beschreibung

- In (S1+) wird die Anzahl der auszugebenden Impulse vorgegeben. Der Wertebereich umfasst bei Wortoperanden -32767 bis 32767 Inkremente und bei 32-Bit-Operanden -999999 bis 999999 Inkremente. Der Wert „0“ darf nicht vorgegeben werden.
- In (S2+) wird die Frequenz der Ausgangsimpulse angegeben. Bei Wortoperanden umfasst der Bereich 10 Hz bis 32767 Hz, bei 32-Bit-Operanden können zur Ausgabe am Grundgerät Frequenzen von 10 Hz bis 100 kHz und zur Ausgabe an einem HIGH-SPEED-Ausgabemodul FX3U-2HSY-ADP Frequenzen von 10 Hz bis 200 kHz angegeben werden.
- (D1+) gibt den Ausgang zur Impulsausgabe an.

Zur Ausgabe der Impulse können nur bestimmte Ausgänge (siehe oben) verwendet werden. Da die Impulse mit hoher Frequenz ausgegeben werden, müssen Steuerungen mit Transistorausgängen verwendet werden. Relaiskontakte nutzen sich in diesem Fall vorzeitig ab und sind daher ungeeignet. Bei der FX3U-Serie können für das erste Ausgabemodul FX3U-2HSY-ADP die Ausgänge Y0 und Y1 und für das Zweite die Ausgänge Y2 und Y3 angegeben werden. Die Zustände von Y4 und Y5 bzw. Y6 und Y7 bestimmen die Drehrichtung.

- In (D2+) wird ein Operand angegeben, der dem Servoverstärker die Drehrichtung vorgibt.

HINWEISE

Ein Programm, das eine DVIT-Anweisung enthält, sollte nicht im RUN-Modus der SPS geändert werden, während diese Anweisung ausgeführt wird. Wird dies nicht beachtet, wird der Servoantrieb verzögert und gestoppt.

Die Positionierung mit einer FX3U oder FX3UC und die Anwendung der DVIT-Anweisung ist im Positionierhandbuch zur FX3-Serie ausführlich beschrieben.

7.9.8 Positionierung nach Datentabelle (TBL)

			TBL		FNC 152			
			Positionierung nach Datentabelle					
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC		
				●		●		
Operanden	D+	n	Puls-Anweisung (P)		Verarbeitung		Programmschritte	
	Y *	K, H			16 Bit	32 Bit	TBL	9
					●	●	DTBL	17

* Nur Transistorausgänge Y0, Y1 oder Y2 im Grundgerät oder Y0, Y1, Y2 oder Y3 eines HIGH-SPEED-Ausgabemoduls FX3U-2HSY-ADP (nur an FX3U-Grundgeräte anschließbar). Die Impulsausgabe an Y2 des Grundgeräts ist nur bei einer FX3U oder FX3UC sowie einer FX3G-40M□, FX3G-60M□ oder FX3GE-40M□ möglich. Für die Angabe von Y2 oder Y3 bei einem HIGH-SPEED-Ausgabemodul wird ein zweites HIGH-SPEED-Ausgabemodul benötigt.

Funktion

Mit der TBL-Anweisung wird nach Parametern positioniert, die in der SPS in einer Tabelle abgelegt sind.

Beschreibung

- (D+) gibt den Ausgang zur Impulsausgabe an.
 Zur Ausgabe der Impulse können nur bestimmte Ausgänge (siehe oben) verwendet werden. Da die Impulse mit hoher Frequenz ausgegeben werden, müssen Steuerungen mit Transistorausgängen verwendet werden. Relaiskontakte nutzen sich in diesem Fall vorzeitig ab und sind daher ungeeignet. Bei der FX3U-Serie können für das erste Ausgabemodul FX3U-2HSY-ADP die Ausgänge Y0 und Y1 und für das Zweite die Ausgänge Y2 und Y3 angegeben werden. Die Zustände von Y4 und Y5 bzw. Y6 und Y7 bestimmen die Drehrichtung.
- (n) gibt die Nummer der Tabelle an (1 bis 100).

HINWEISE

- Ein Programm, das eine TBL-Anweisung enthält, kann nicht im RUN-Modus der SPS geändert werden.
- Die Positionierung mit einer FX3G/FX3GC/FX3GE/FX3U/FX3UC und die Anwendung der TBL-Anweisung ist im Positionierhandbuch zur FX3-Serie ausführlich beschrieben.

7.9.9 Lesen der absoluten Ist-Position (DABS)

				DABS		FNC 155			
				Lesen der absoluten Ist-Position					
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC			
	●	●	●	●	●	●	●		
Operanden	S+	D1+	D2+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	X, Y, M, S, D□.b ^①	Y ^② , M, S, D□.b ^①	KnY, KnM, KnS, T, C, D, R ^③ , U□/G□ ^② , V, Z			16 Bit	32 Bit	DABS	13
						●			

- ① Nur bei FX3U und FX3UC
- ② Nur Transistorausgänge
- ③ Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

Funktion

Mit dieser Anweisung kann die absolute Position eines angeschlossenen MITSUBISHI Servo-Verstärkers vom Typ MR-H, MR-J2(S), MR-J3 oder MR-J4 erfasst werden. Der Servo-Antrieb muss in der Lage sein, die Position absolut zu erfassen.

Beschreibung

- Die ABS-Anweisung ist eine 32-Bit-Anweisung. Geben Sie aus diesem Grund die Anweisung immer als „DABS“ ein.
- Das Lesen der Position beginnt, wenn die Startbedingung der DABS-Anweisung erfüllt ist. Wenn das Lesen abgeschlossen ist, wird M8029 gesetzt. Wenn die Startbedingung während des Lesens ausgeschaltet wird, wird der Lesevorgang unterbrochen.
- Bei der Projektierung des Antriebes ist zu beachten, dass der Servo-Verstärker vor oder gleichzeitig mit der SPS eingeschaltet wird.
- Die absolute Position wird in den Operanden abgelegt, der in D2+ angegeben wurde. Dies kann ein Wort-Operand sein. Jedoch sollte der Wert später im Programm in die Sonderregister D8141 und D8140 transferiert werden.
- Die Startbedingung der DABS-Anweisung sollte immer, auch nachdem die Position erfasst wurde, eingeschaltet sein. Wenn die Startbedingung nach dem Lesen der Position unwahr wird, wird das Signal „Servo-ON“ (SON) zurückgesetzt und der Antrieb gestoppt.
- Auch wenn der Servoantrieb die absolute Position erfassen kann, sollte nach dem Einschalten des Antriebes eine Referenzfahrt gemacht werden.

Beispiel ▾

ABS-Anweisung

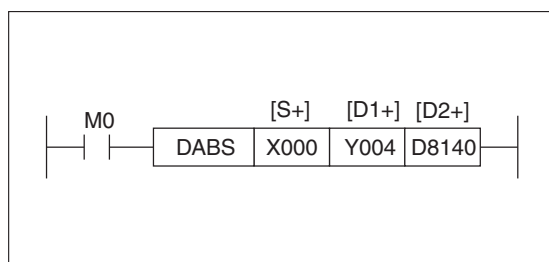


Abb. 7-136:
Programmbeispiel zum Einsatz einer ABS-Anweisung



7.9.10 Referenzpunkt anfahren (ZRN)

					ZRN		FNC 156				
					Referenzpunkt anfahren						
					CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
						●	●		●	●	●
Operanden	S1+	S2+	S3+	D	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , U□/G□ ^② , V, Z,		X, Y, M, S	Y ^③			16 Bit	32 Bit	ZRN	9	
							●	●	DZRN	17	

- ① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ② Nur bei FX3U und FX3UC
- ③ Bei FX1S und FX1N: Nur Y0 oder Y1;
Bei FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC: Nur Transistorausgänge Y0, Y1 oder Y2 im Grundgerät oder Y0, Y1, Y2 oder Y3 eines HIGH-SPEED-Ausgabemoduls FX3U-2HSY-ADP (nur an FX3U-Grundgeräte anschließbar). Die Impulsausgabe an Y2 des Grundgeräts ist nur bei einer FX3U oder FX3UC sowie einer FX3G-40M□, FX3G-60M□ oder FX3GE-40M□ möglich. Für die Angabe von Y2 oder Y3 bei einem HIGH-SPEED-Ausgabemodul wird ein zweites HIGH-SPEED-Ausgabemodul benötigt.

Funktion

Mit dieser Anweisung wird eine Fahrt zum Referenzpunkt (Nullpunkt) eingeleitet. Während der Positionierung wird von der SPS die Ist-Position gespeichert und abhängig von der Bewegungsrichtung erhöht oder erniedrigt. Dadurch ist die Maschinenposition in der SPS stets aktuell. Bei einem Spannungsausfall verliert die SPS diesen Positionswert. Deshalb wird nach dem Einschalten der Steuerung oder zum Einrichten der Maschine eine Referenzfahrt zu einer definierten Position ausgeführt.

Beschreibung

- In (S1+) wird die Geschwindigkeit angegeben, mit der die Referenzfahrt ausgeführt werden soll. Bei Wortoperanden sind 10 bis 32767 Hz möglich, bei 32-Bit-Operanden geht der Bereich von 10 Hz bis 100 kHz bzw. von 10 Hz bis 200 kHz bei einer FX3U und einem HIGH-SPEED-Ausgabemodul FX3U-2HSY-ADP.
- Die Schleichgeschwindigkeit im Bereich von 10 bis 32767 Hz wird in (S2+) angegeben.
- (S3+) enthält den Schalter zur Einleitung der Schleichfahrt in der Nähe des Referenzpunktes. Wenn in (S3+) ein anderer Operand als ein Eingang (X) angegeben wird, wird der Zeitpunkt des Bremsens von der Zykluszeit der SPS beeinflusst und die Abweichung vom Referenzpunkt ist evtl. zu groß.
- Zur Ausgabe der Impulse können nur bestimmte Ausgänge (siehe oben) verwendet werden. Da die Impulse mit hoher Frequenz ausgegeben werden, müssen Steuerungen mit Transistorausgängen verwendet werden. Relaiskontakte nutzen sich in diesem Fall vorzeitig ab und sind daher ungeeignet.

Zur Erzeugung steilflankiger Ausgangssignale sollte der Laststrom der Transistorausgänge mindestens 200 mA betragen. Eventuell ist die Verwendung von Pull-up-Widerständen notwendig.
- Wenn der Sondermerker M8140 gesetzt ist, wird dem Servo-Verstärker signalisiert, dass der Referenzpunkt erreicht wurde.

- Die folgenden Operanden stehen für Positionieraufgaben zur Verfügung:

Operand	Beschreibung
D8141 (höherwertiges Wort) D8140 (niederwertiges Wort)	Istwert von Y0 (32 Bit)
D8143 (höherwertiges Wort) D8142 (niederwertiges Wort)	Istwert von Y1 (32 Bit)
M8145	Impulsausgabe an Y0 sofort stoppen
M8146	Impulsausgabe an Y1 sofort stoppen
M8147	Überwachung der Impulsausgabe an Y0 (BUS/READY)
M8148	Überwachung der Impulsausgabe an Y1 (BUS/READY)

Tab. 7-36: Operanden zur Positionierung bei einer SPS der FX1S- und FX1N-Serie

Die Operanden für eine FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC sind im Positionierhandbuch für die FX3-Serie beschrieben.

- Bei einem MITSUBISHI Servo-Verstärkers vom Typ MR-H, MR-J2(S), MR-J3 oder MR-J4, der in der Lage ist, die absolute Position zu erfassen, kann die Ist-Position mit einer ABS-Anweisung (FNC155) gelesen werden.

HINWEISE

Der Referenzpunkt kann nicht aus jeder Richtung angefahren werden. Bei der Referenzfahrt wird immer rückwärts in Richtung des Referenzpunkts gefahren.

Beachten Sie bei der Programmierung der Startbedingung die zeitlichen Abläufe.

Vermeiden Sie es, das Programm im RUN-Modus der SPS zu ändern, während eine ZRN-Anweisung ausgeführt wird. Während der Aktualisierung eines Programms, das eine ZRN-Anweisung enthält, wird die Impulsausgabe unterbrochen und dadurch der Servoantrieb gestoppt.

Beispiel ▾

ZRN-Anweisung

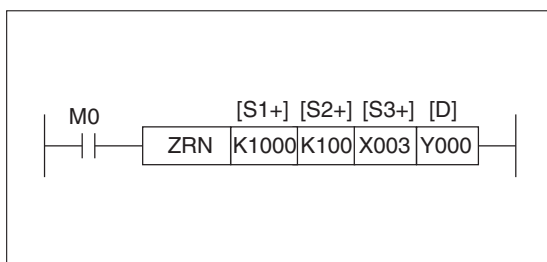


Abb. 7-137: Programmbeispiel zum Einsatz der ZRN-Anweisung



7.9.11 Ausgabe von Impulsen mit variabler Frequenz (PLSV)

				PLSV		FNC 157				
				Impulsausgabe						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
					●	●		●	●	●
Operanden	S+	D1	D2+	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , U□/G□ ^② , V, Z	Y ^③	Y, M, S			16 Bit	32 Bit	PLSV	9	
					●	●	DPLSV	17		

- ① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ② Nur bei FX3U und FX3UC
- ③ Bei FX1S und FX1N: Nur Y0 oder Y1;
Bei FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC: Nur Transistorausgänge Y0, Y1 oder Y2 im Grundgerät oder Y0, Y1, Y2 oder Y3 eines HIGH-SPEED-Ausgabemoduls FX3U-2HSY-ADP (nur an FX3U-Grundgeräte anschließbar). Die Impulsausgabe an Y2 des Grundgeräts ist nur bei einer FX3U oder FX3UC sowie einer FX3G-40M□, FX3G-60M□ oder FX3GE-40M□ möglich. Für die Angabe von Y2 oder Y3 bei einem HIGH-SPEED-Ausgabemodul wird ein zweites HIGH-SPEED-Ausgabemodul benötigt.

Funktion

Ausgabe von Impulsen mit variabler Frequenz an den Ausgängen der SPS und eines Signals zur Festlegung der Drehrichtung

Beschreibung

- In (S1+) wird die Frequenz der Ausgangsimpulse angegeben. Bei Wortoperanden einer FX1S und FX1N umfasst der Bereich 10 bis 32767 Hz, bei 32-Bit-Operanden ist die Ausgabe einer Frequenz von 10 bis 100 kHz möglich.

Bei Wortoperanden der FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC sind -32768 bis -1 und 1 bis 32767 Hz möglich (nicht 0 Hz!), bei 32-Bit-Operanden geht der Bereich von -100 kHz bis -1 Hz und von 1 Hz bis 100 kHz bzw. von -200 kHz bis -1 Hz und von 1 Hz bis 200 kHz bei einer FX3U und einem HIGH-SPEED-Ausgabemodul FX3U-2HSY-ADP.
- Bei einer FX1S oder FX1N können zur Ausgabe der Impulse in (D1) nur die Ausgänge Y0 und Y1 verwendet werden. Da die Impulse mit hoher Frequenz ausgegeben werden, müssen Steuerungen mit Transistorausgängen verwendet werden. Relaiskontakte nutzen sich in diesem Fall vorzeitig ab und sind daher ungeeignet. Zur Erzeugung steilflankiger Ausgangssignale sollte der Laststrom der Transistorausgänge mindestens 200 mA betragen. Eventuell ist die Verwendung von Pull-up-Widerständen notwendig.

Bei der FX3U-Serie können für das erste Ausgabemodul FX3U-2HSY-ADP die Ausgänge Y0 und Y1 und für das Zweite die Ausgänge Y2 und Y3 angegeben werden. Die Zustände von Y4 und Y5 bzw. Y6 und Y7 bestimmen die Drehrichtung.
- In (D2+) wird ein Operand angegeben, der dem Servoverstärker die Drehrichtung vorgibt. Wenn der Operand nicht gesetzt ist, ist Linkslauf angewählt, bei gesetztem Operanden Rechtslauf.
- Die Drehrichtung wird durch das Vorzeichen der in S+ angegebenen Frequenz bestimmt. Ein positiver Wert bedeutet Rechtslauf, bei einem negativen Wert wird der Antrieb im Linkslauf betrieben.
- Während der Ausgabe von Impulsen kann die Frequenz der Impulse (in (S+)) geändert werden.
- Beim Beginn und beim Beenden der Impulsausgabe wird keine Beschleunigungs- oder Verzögerungsrampe durchlaufen. Wenn dies gefordert wird, muss mit der RAMP-Anweisung (FNC67) die Ausgangsfrequenz verändert werden.

- Wenn die Startbedingung der PLSV-Anweisung während der Impulsausgabe nicht mehr erfüllt ist, wird der Antrieb angehalten, ohne dass eine Verzögerungsrampe durchlaufen wird.
- Nach dem Rücksetzen der Startbedingung kann die PLSV-Anweisung nicht erneut gestartet werden, solange die Sondermerker M8147 (Überwachung von Y0) und M8148 (Überwachung von Y1) gesetzt sind.
- Die folgenden Operanden stehen für Positionieraufgaben zur Verfügung:

Operand	Beschreibung
D8141 (höherwertiges Wort) D8140 (niederwertiges Wort)	Istwert von Y0 (32 Bit)
D8143 (höherwertiges Wort) D8142 (niederwertiges Wort)	Istwert von Y1 (32 Bit)
M8145	Impulsausgabe an Y0 sofort stoppen
M8146	Impulsausgabe an Y1 sofort stoppen
M8147	Überwachung der Impulsausgabe an Y0 (BUS/READY)
M8148	Überwachung der Impulsausgabe an Y1 (BUS/READY)

Tab. 7-37: Operanden zur Positionierung bei einer SPS der FX1S- und FX1N-Serie

Die Operanden für eine FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC sind im Positionierhandbuch für die FX3-Serie beschrieben.

HINWEISE

Beachten Sie bei der Programmierung der Startbedingung die zeitlichen Abläufe.

Vermeiden Sie es, das Programm im RUN-Modus der SPS zu ändern, während eine PLSV-Anweisung ausgeführt wird. Ein Programm, das eine PLSV-Anweisung enthält, verhält sich während der Aktualisierung wie folgt:

- Bei einer Beschleunigung/Verzögerung wird verzögert und die Impulsausgabe gestoppt.
- Wenn keine Beschleunigung oder Verzögerung stattfindet, wird die Impulsausgabe sofort gestoppt.

Beispiel ▾

PLSV-Anweisung

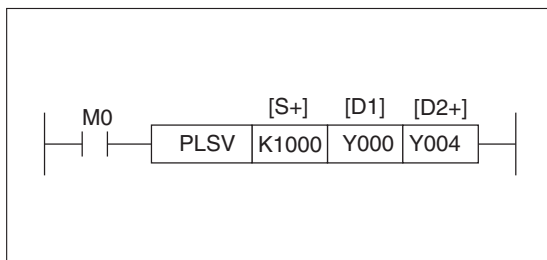


Abb. 7-138:

Programmbeispiel zum Einsatz der PLSV-Anweisung



7.9.12 Positionieren auf einen Inkrementalwert (DRVI)

					DRVI		FNC 158				
					Positionieren auf einen Inkrementalwert						
					CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
						●	●		●	●	●
Operanden	S1+	S2+	D1	D2+	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , U□/G□ ^② , V, Z		Y ^③	Y, M, S, D□.b ^②			16 Bit	32 Bit	DRVI	9	
						●	●	DDRVI	17		

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

③ Bei FX1S und FX1N: Nur Y0 oder Y1;
Bei FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC: Nur Transistorausgänge Y0, Y1 oder Y2 im Grundgerät oder Y0, Y1, Y2 oder Y3 eines HIGH-SPEED-Ausgabemoduls FX3U-2HSY-ADP (nur an FX3U-Grundgeräte anschließbar). Die Impulsausgabe an Y2 des Grundgeräts ist nur bei einer FX3U oder FX3UC sowie einer FX3G-40M□, FX3G-60M□ oder FX3GE-40M□ möglich. Für die Angabe von Y2 oder Y3 bei einem HIGH-SPEED-Ausgabemodul wird ein zweites HIGH-SPEED-Ausgabemodul benötigt.

Funktion

Eine Position, die durch einen Inkrementalwert vorgegeben ist, wird mit einer festen Geschwindigkeit angefahren.

Beschreibung

- In (S1+) wird die Anzahl der Inkremente vorgegeben. Der Wertebereich umfasst bei Wortoperanden -32767 bis 32767 Inkremente und bei 32-Bit-Operanden -999999 bis 999999 Inkremente.
- In (S2+) wird die Frequenz der Ausgangsimpulse angegeben. Bei Wortoperanden umfasst der Bereich 10 Hz bis 32767 Hz, bei 32-Bit-Operanden ist die Ausgabe einer Frequenz von 10 Hz bis 100 kHz möglich (10 Hz bis 200 kHz bei einer FX3U in Verbindung mit einem HIGH-SPEED-Ausgabemodul).
- Bei einer FX1S oder FX1N können zur Ausgabe der Impulse in (D1) nur die Ausgänge Y0 und Y1 verwendet werden. Da die Impulse mit hoher Frequenz ausgegeben werden, müssen Steuerungen mit Transistorausgängen verwendet werden. Relaiskontakte nutzen sich in diesem Fall vorzeitig ab und sind daher ungeeignet. Zur Erzeugung steilflankiger Ausgangssignale sollte der Laststrom der Transistorausgänge mindestens 200 mA betragen. Eventuell ist die Verwendung von Pull-up-Widerständen notwendig.

Bei der FX3U-Serie können für das erste Ausgabemodul FX3U-2HSY-ADP die Ausgänge Y0 und Y1 und für das Zweite die Ausgänge Y2 und Y3 angegeben werden. Die Zustände von Y4 und Y5 bzw. Y6 und Y7 bestimmen die Drehrichtung.
- In (D2+) wird ein Operand angegeben, der dem Servoverstärker die Drehrichtung vorgibt. Wenn der Operand nicht gesetzt ist, ist Linkslauf angewählt, bei gesetztem Operanden Rechtslauf.
- Bei der Positionierung auf einen Inkrementalwert wird die Richtung durch das Vorzeichen der Inkremente vorgegeben.
- Wenn während der Ausführung der Anweisung der Inhalt eines Operanden verändert wird, wird die Positionierung nicht beeinflusst. Die veränderten Operanden werden bei der nächsten Ausführung der Anweisung berücksichtigt.

- Wenn die Startbedingung der DRVI-Anweisung während der Ausführung der Anweisung nicht mehr erfüllt ist, wird eine Verzögerungsrampe durchlaufen und der Antrieb gestoppt. In diesem Fall wird der Sondermerker M8029, der das Ende der Bearbeitung anzeigt, nicht gesetzt.
- Bei einer FX1S oder FX1N kann nach dem Rücksetzen der Startbedingung die DRVI-Anweisung erst erneut gestartet werden, wenn die Sondermerker M8147 (Überwachung von Y0) und M8148 (Überwachung von Y1) nicht mehr gesetzt sind.

Die Bedingungen bei einer FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC sind im Positionierhandbuch für die FX3-Serie beschrieben.

- Die minimale Ausgangsfrequenz kann mit der folgenden Formel berechnet werden:

$$f_{min} = \sqrt{\frac{f_{max}}{2 \times t_B}}$$

f_{max} : maximale Frequenz der Ausgangsimpulse (In D8146/D8147 bei FX1S und FX1N)
 t_B ist die Beschleunigungs- und Verzögerungszeit in der Einheit Sekunden

- Die folgenden Operanden stehen für Positionieraufgaben zur Verfügung:

Operand	Beschreibung
D8145	Drehzahl-Offset bei der Verwendung von FNC158 oder FNC159
D8147 (höherwertiges Wort) D8146 (niederwertiges Wort)	Maximale Frequenz der Ausgangsimpulse bei der Verwendung von FNC158 oder FNC159. Bereich: 100 Hz bis 100000 Hz
D8148	Beschleunigungs- und Verzögerungszeit [ms] bei der Verwendung von FNC158 oder FNC159
M8145	Impulsausgabe an Y0 sofort stoppen
M8146	Impulsausgabe an Y1 sofort stoppen
M8147	Überwachung der Impulsausgabe an Y0 (BUS/READY)
M8148	Überwachung der Impulsausgabe an Y1 (BUS/READY)

Tab. 7-38: Operanden zur Positionierung bei einer SPS der FX1S- und FX1N-Serie

Die Operanden für eine FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC sind im Positionierhandbuch für die FX3-Serie beschrieben.

HINWEISE

Beachten Sie bei der Programmierung der Startbedingung die zeitlichen Abläufe.

Ein Programm, das eine DRVI-Anweisung enthält, sollte nicht im RUN-Modus der SPS geändert werden, während diese Anweisung ausgeführt wird. Wird dies nicht beachtet, wird der Servoantrieb verzögert und gestoppt.

Beispiel ▾

DRVI-Anweisung

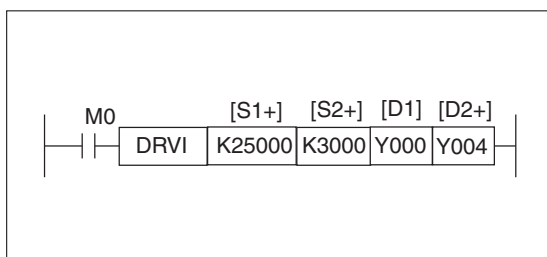


Abb. 7-139:
 Programmbeispiel zum Einsatz der DRVI-Anweisung



7.9.13 Positionieren auf einen Absolutwert (DRVA)

					DRVA		FNC 159				
					Positionieren auf einen Absolutwert						
					CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
						●	●		●	●	●
Operanden	S1+	S2+	D1	D2+	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , U□/G□ ^② , V, Z		Y ^③	Y, M, S, D□.b ^②			16 Bit	32 Bit	DRVA	9	
						●	●	DDRVA	17		

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

③ Bei FX1S und FX1N: Nur Y0 oder Y1;
 Bei FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC: Nur Transistorausgänge Y0, Y1 oder Y2 im Grundgerät oder Y0, Y1, Y2 oder Y3 eines HIGH-SPEED-Ausgabemoduls FX3U-2HSY-ADP (nur an FX3U-Grundgeräte anschließbar). Die Impulsausgabe an Y2 des Grundgeräts ist nur bei einer FX3U oder FX3UC sowie einer FX3G-40M□, FX3G-60M□ oder FX3GE-40M□ möglich. Für die Angabe von Y2 oder Y3 bei einem HIGH-SPEED-Ausgabemodul wird ein zweites HIGH-SPEED-Ausgabemodul benötigt.

Funktion

Eine Position, die durch einen Absolutwert vorgegeben ist, wird mit einer festen Geschwindigkeit angefahren. Bei dieser Art der Positionierung wird ein Nullpunkt festgelegt. Die aktuelle Ist-Position wird als absoluter Wert in Bezug zu diesem Nullpunkt gemessen.

Beschreibung

- In (S1+) wird die absolute Position vorgegeben. Der Wertebereich geht bei Wortoperanden von -32767 bis 32767 und bei 32-Bit-Operanden von -999999 bis 999999.
- In (S2+) wird die Frequenz der Ausgangsimpulse angegeben. Bei Wortoperanden umfasst der Bereich 10 Hz bis 32767 Hz, bei 32-Bit-Operanden ist die Ausgabe einer Frequenz von 10 Hz bis 100 kHz möglich (10 Hz bis 200 kHz bei einer FX3U in Verbindung mit einem HIGH-SPEED-Ausgabemodul FX3U-2HSY-ADP).
- Bei einer FX1S oder FX1N können zur Ausgabe der Impulse in (D1) nur die Ausgänge Y0 und Y1 verwendet werden. Da die Impulse mit hoher Frequenz ausgegeben werden, müssen Steuerungen mit Transistorausgängen verwendet werden. Relaiskontakte nutzen sich in diesem Fall vorzeitig ab und sind daher ungeeignet. Zur Erzeugung steiflankiger Ausgangssignale sollte der Laststrom der Transistorausgänge mindestens 200 mA betragen. Eventuell ist die Verwendung von Pull-up-Widerständen notwendig.

 Bei der FX3U-Serie können für das erste Ausgabemodul FX3U-2HSY-ADP die Ausgänge Y0 und Y1 und für das Zweite die Ausgänge Y2 und Y3 angegeben werden. Die Zustände von Y4 und Y5 bzw. Y6 und Y7 bestimmen die Drehrichtung.
- In (D2+) wird ein Operand angegeben, der dem Servoverstärker die Drehrichtung vorgibt. Wenn der Operand nicht gesetzt ist, ist Linkslauf angewählt, bei gesetztem Operanden Rechtslauf.
- Bei der Positionierung auf einen Absolutwert wird als Ziel der Abstand vom Nullpunkt vorgegeben.
- Wenn während der Ausführung der Anweisung der Inhalt eines Operanden verändert wird, wird die Positionierung nicht beeinflusst. Die veränderten Operanden werden bei der nächsten Ausführung der Anweisung berücksichtigt.

- Wenn die Startbedingung der DRVI-Anweisung während der Ausführung der Anweisung nicht mehr erfüllt ist, wird eine Verzögerungsrampe durchlaufen und der Antrieb gestoppt. In diesem Fall wird der Sondermerker M8029, der das Ende der Bearbeitung anzeigt, nicht gesetzt.
- Bei einer FX1S oder FX1N kann nach dem Rücksetzen der Startbedingung die DRVI-Anweisung erst erneut gestartet werden, wenn die Sondermerker M8147 (Überwachung von Y0) und M8148 (Überwachung von Y1) nicht mehr gesetzt sind.

Die Bedingungen bei einer FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC sind im Positionierhandbuch für die FX3-Serie beschrieben.

- Die minimale Ausgangsfrequenz kann mit der folgenden Formel berechnet werden:

$$f_{min} = \sqrt{\frac{f_{max}}{2 \times t_B}}$$

f_{max} : maximale Frequenz der Ausgangsimpulse (In D8146/D8147 bei FX1S und FX1N.
 t_B ist die Beschleunigungs- und Verzögerungszeit in der Einheit Sekunden

- Die folgenden Operanden stehen für Positionieraufgaben zur Verfügung:

Operand	Beschreibung
D8140 D8141	Istwert Y0 (32 Bit)
D8142 D8143	Istwert Y1(32 Bit)
D8145	Drehzahl-Offset bei der Verwendung von FNC158 oder FNC159
D8147 (höherwertiges Wort) D8146 (niederwertiges Wort)	Maximale Frequenz der Ausgangsimpulse bei der Verwendung von FNC158 oder FNC159. Bereich: 100 Hz bis 100 kHz
D8148	Beschleunigungs- und Verzögerungszeit [ms] bei der Verwendung von FNC158 oder FNC159
M8145	Impulsausgabe an Y0 sofort stoppen
M8146	Impulsausgabe an Y1 sofort stoppen
M8147	Überwachung der Impulsausgabe an Y0 (BUS/READY)
M8148	Überwachung der Impulsausgabe an Y1 (BUS/READY)

Tab. 7-39: Operanden zur Positionierung bei einer SPS der FX1S- und FX1N-Serie

Die Operanden für eine FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC sind im Positionierhandbuch für die FX3-Serie beschrieben.

HINWEISE

Beachten Sie bei der Programmierung der Startbedingung die zeitlichen Abläufe.

Ein Programm, das eine DRVA-Anweisung enthält, sollte nicht im RUN-Modus der SPS geändert werden, während diese Anweisung ausgeführt wird. Wird dies nicht beachtet, wird der Servoantrieb verzögert und gestoppt.

Beispiel

DRVA-Anweisung

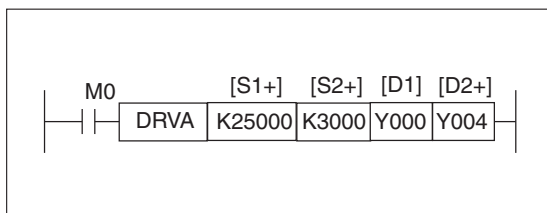


Abb. 7-140

Programmbeispiel zum Einsatz der DRVA-Anweisung



7.10 Anweisungen für die integrierte Uhr der SPS

Übersicht der Anweisungen FNC 160 bis 169

Symbol	FNC	Bedeutung	Abschnitt
TCMP	160	Vergleich von Uhr-Daten	7.10.1
TZCP	161	Vergleich von Uhr-Daten mit einem Bereich	7.10.2
TADD	162	Addition von Uhr-Daten	7.10.3
TSUB	163	Subtraktion von Uhr-Daten	7.10.4
HTOS	164	Zeitangabe in der Form „Stunden, Minuten, Sekunden“ in Sekunden wandeln	7.10.5
STOH	165	Zeitangabe in Sekunden in das Format „Stunden, Minuten, Sekunden“ wandeln	7.10.6
TRD	166	Lesen von Datum und Uhrzeit	7.10.7
TRW	167	Schreiben von Uhr-Daten (Uhr stellen)	7.10.8
HOUR	169	Betriebsstundenzähler	7.10.9

Tab. 7-23: Übersicht der Anweisungen für die integrierte Uhr der SPS

7.10.1 Vergleich von Uhr-Daten (TCMP)

						TCMP		FNC 160			
						Vergleich von Uhr-Daten					
CPU						FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
						●	●	●	●	●	●
Operanden	S1+	S2+	S3+	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , U□/G□ ^② , V, Z			K, H, T, C, D, R ^① , U□/G□ ^②	Y, M, S, D□.b ^②		●	16 Bit	32 Bit	TCMP	11
	Es werden jeweils 3 aufeinanderfolgende Operandenadressen verwendet.					●	●		TCMPP	11	

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktion

Vergleichen von Uhr-Daten mit Ausgabe der Vergleichsergebnisse

Beschreibung

- Die Uhr-Daten „Stunde“ (S1+), „Minute“ (S2+) und „Sekunde“ (S3+) werden mit den in (S+) bis ((S+)+2) gespeicherten Uhr-Daten verglichen.
- Die Vergleichsergebnisse werden in 3 aufeinanderfolgenden Bit-Operanden gespeichert.
- Sind die Uhr-Daten in (S+) bis ((S+)+2) kleiner als die Uhr-Daten in (S1+) bis (S3+), wird der Bit-Operand (D+) gesetzt.
- Sind die Uhr-Daten in (S+) bis ((S+)+2) gleich den Uhr-Daten in (S1+) bis (S3+), wird der Bit-Operand ((D+)+1) gesetzt.
- Sind die Uhr-Daten in (S+) bis ((S+)+2) größer als die Uhr-Daten in (S1+) bis (S3+), wird der Bit-Operand ((D+)+2) gesetzt.

HINWEISE

Die angesprochenen Ausgangsoperanden bleiben nach Abschalten der Ausführungsbedingung der TCMP-Anweisung gesetzt.

In den Operanden (S1+) und (S+) können die Werte 0 bis 23 (Stunden) eingegeben werden.

In den Operanden (S2+) und ((S+)+1) können die Werte 0 bis 59 (Minuten) eingegeben werden.

Die Operanden (S3+) und ((S+)+2) können die Werte 0 bis 59 (Sekunden) enthalten.

Für einen Vergleich der aktuellen Daten der integrierten Uhr kann eine TRD-Anweisung ausgeführt und danach der Inhalt der dort angegebene Zielregister ausgewertet werden. Alternativ können auch die Inhalte der Sonderregister D8015 (Stunden), D8014 (Minuten) und D8013 (Sekunden) verwendet werden.

Beispiel ▾

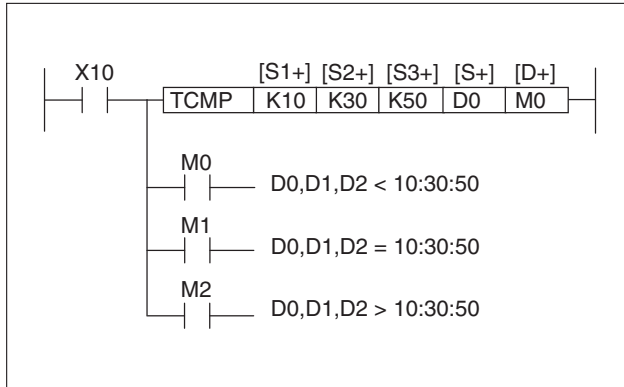


Abb. 7-141:
 Programmierbeispiel zur TCMP-
 Anweisung

Mit Einschalten des Eingangs X10 werden die mit K10, K30 und K50 angegebenen 10 Stunden, 30 Minuten und 50 Sekunden mit den Uhr-Daten in D0 bis D2 verglichen.

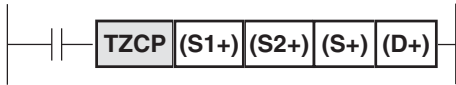
Ist der Wert in D0 bis D2 kleiner als der Wert 10:30:50, wird der Merker M0 gesetzt.

Ist der Wert in D0 bis D2 gleich dem Wert 10:30:50, wird der Merker M1 gesetzt.

Ist der Wert in D0 bis D2 größer als der Wert 10:30:50, wird der Merker M2 gesetzt.



7.10.2 Vergleich von Uhr-Daten mit einem Bereich (TZCP)

					TZCP		FNC 161			
					Vergleich von Uhr-Daten mit einem Bereich					
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC				
		●	●	●	●	●	●			
Operanden	S1+	S2+	S+	D+	Puls-Anweisung (P)		Programmschritte			
	T, C, D, R ^① , U□/G□ ^② , V, Z ((S1+) ≤ (S2+))			Y, M, S, D□.b ^②	●	16 Bit	32 Bit	TZCP	9	
	Es werden jeweils 3 aufeinanderfolgende Operanden- adressen verwendet.					●	●	TZCPP	9	

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktion

Vergleichen von Uhr-Daten mit einem Bereich mit Ausgabe der Vergleichsergebnisse

Beschreibung

- Die Uhr-Daten in (S+) bis ((S+)+2) werden mit den Uhr-Daten im Bereich zwischen (S1+) bis ((S1+)+2) und (S2+) bis ((S2+)+2) verglichen.
- Die Vergleichsergebnisse werden in 3 aufeinanderfolgenden Bit-Operanden gespeichert.
- Sind die Uhr-Daten in (S+) bis ((S+)+2) kleiner als die Uhr-Daten in (S1+) bis ((S1+)+2), wird der Bit-Operand (D+) gesetzt.
- Liegen die Uhr-Daten in (S+) bis ((S+)+2) in dem Bereich zwischen (S1+) bis ((S1+)+2) und (S2+) bis ((S2+)+2), wird der Bit-Operand ((D+)+1) gesetzt.
- Sind die Uhr-Daten in (S+) bis ((S+)+2) größer als die Uhr-Daten in (S2+) bis ((S2+)+2), wird der Bit-Operand ((D+)+2) gesetzt.

HINWEISE

Die angesprochenen Ausgangsoperanden bleiben nach Abschalten der Ausführungsbedingung der TCMP-Anweisung gesetzt.

In den Operanden (S1+), (S2+) und (S+) können die Werte 0 bis 23 (Stunden) eingegeben werden.

Die Operanden ((S1+)+1), ((S2+)+1) und ((S+)+1) können die Werte 0 bis 59 (Minuten) enthalten.

In den Operanden ((S1+)+2), ((S2+)+2) und ((S+)+2) können die Werte 0 bis 59 (Sekunden) eingegeben werden.

Für einen Vergleich der aktuellen Daten der integrierten Uhr kann eine TRD-Anweisung ausgeführt und danach der Inhalt der dort angegebene Zielregister ausgewertet werden. Alternativ können auch die Inhalte der Sonderregister D8015 (Stunden), D8014 (Minuten) und D8013 (Sekunden) verwendet werden.

Beispiel ▾

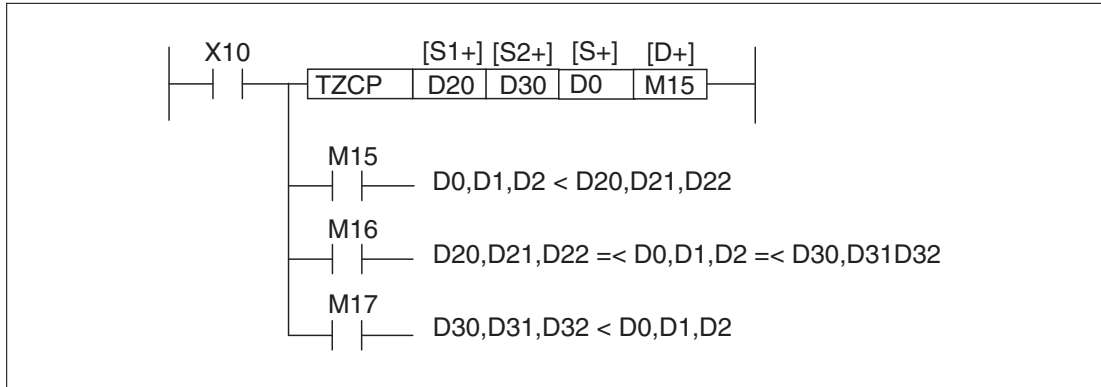


Abb. 7-142: Programmierbeispiel zur TZCP-Anweisung

Mit Einschalten des Eingangs X10 werden die Uhr-Daten in D0 bis D2 mit dem Uhr-Datenbereich zwischen D20 bis D22 und D30 bis D32 verglichen.

Sind die Uhr-Daten in D0 bis D2 kleiner als die Uhr-Daten in D20 bis D22, wird der Merker M15 gesetzt.

Liegen die Uhr-Daten in D0 bis D2 in einem Bereich zwischen den Uhr-Daten in D20 bis D22 und D30 bis D32, wird der Merker M16 gesetzt.

Sind die Uhr-Daten in D0 bis D2 größer als die Uhr-Daten in D30 bis D32, wird der Merker M17 gesetzt.

△

7.10.3 Addition von Uhr-Daten (TADD)

				TADD		FNC 162			
				Addition von Uhr-Daten					
CPU				FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
				●	●	●	●	●	●
Operanden	S1+	S2+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	T, C, D, R ^① , U□/G□ ^②				●	16 Bit	32 Bit	TADD	7
	Es werden jeweils 3 aufeinanderfolgende Operandenadressen verwendet.					●		TADDP	7

- ① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ② Nur bei FX3U und FX3UC

Funktion

Addieren von Uhr-Daten und Speichern des Ergebnisses

Beschreibung

- Die Uhr-Daten in (S1+) bis ((S1+)+2) werden zu den Uhr-Daten in (S2+) bis ((S2+)+2) addiert. Das Ergebnis wird (D+) bis ((D+)+2) gespeichert.
- Bei der Berechnung werden die Überläufe Sekunde - Minute und Minute - Stunde berücksichtigt.

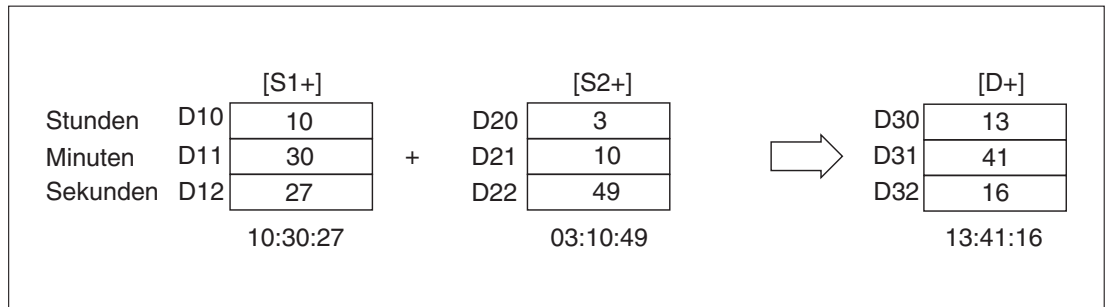


Abb. 7-143: Addition von Uhr-Daten

- Wird das Additionsergebnis größer als 24 Stunden, erfolgt eine Umschaltung auf 0 Stunden („nächster Tag“), und das Carry-Flag M8022 wird gesetzt.

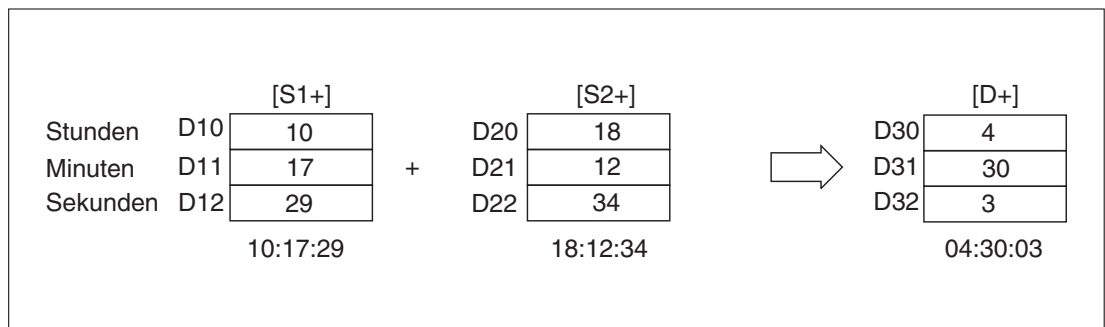


Abb. 7-144: Addition von Uhr-Daten mit Stundenüberlauf

- Ist das Additionsergebnis 0 (00:00:00, 0 Stunden, 0 Minuten, 0 Sekunden), wird das Zero-Flag M8020 gesetzt.

- Als Quelle und Ziel können dieselben Operanden ((S1+) bis ((S1+)+2), (S2+) bis ((S2+)+2)) verwendet werden. In diesem Fall wird das errechnete Ergebnis wieder in dem Quelloperanden gespeichert und anschließend für die nächste Berechnung genutzt. Dieser Prozess wiederholt sich mit bei jeder Ausführung der TADD-Anweisung. Um eine einmalige Ausführung zu gewährleisten, sind gepulste Anweisungen (TADDP) oder Verriegelungen zu verwenden.

HINWEIS Für eine Rechenoperation mit den aktuellen Daten der integrierten Uhr kann eine TRD-Anweisung ausgeführt und danach der Inhalt der dort angegebene Zielregister ausgewertet werden. Alternativ können auch die Inhalte der Sonderregister D8015 (Stunden), D8014 (Minuten) und D8013 (Sekunden) verwendet werden.

Beispiel ▾

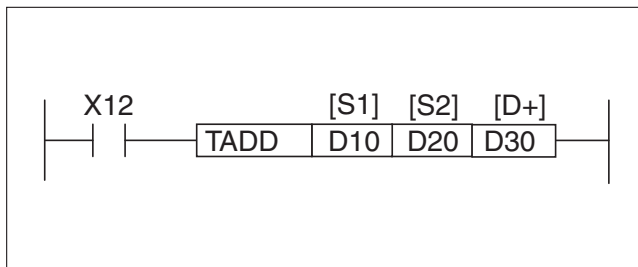


Abb. 7-145:
Programmierbeispiel zur TADD-Anweisung

Bei eingeschaltetem Eingang X12 werden zu den Uhr-Daten in D10 bis D12 die Uhr-Daten aus D20 bis D22 addiert. Das Ergebnis wird in D30 bis D32 gespeichert.

△

7.10.4 Subtraktion von Uhr-Daten (TSUB)

				TSUB		FNC 163			
				Subtraktion von Uhr-Daten					
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC			
	●	●	●	●	●	●			
Operanden	S1+	S2+	D+		Puls-Anweisung (P)	Verarbeitung		Programmschritte	
	T, C, D, R ^① , U□/G□ ^②				●	16 Bit	32 Bit	TSUB	7
	Es werden jeweils 3 aufeinander folgende Operandenadressen verwendet.					●		TSUBP	7

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktion

Subtrahieren von Uhr-Daten und Speichern des Ergebnisses

Beschreibung

- Die Uhr-Daten in (S2+) bis ((S2+)+2) werden von den Uhr-Daten in (S1+) bis ((S1+)+2) subtrahiert. Das Ergebnis wird in (D+) bis ((D+)+2) gespeichert.
- Bei der Berechnung werden die Überläufe Sekunde - Minute und Minute - Stunde berücksichtigt.

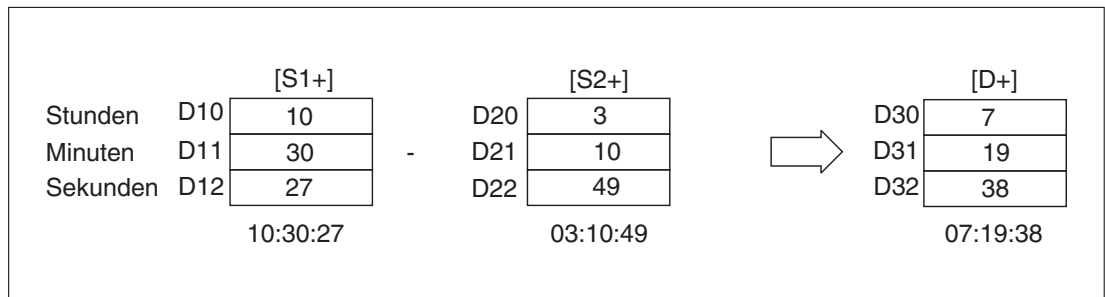


Abb. 7-146: Subtraktion von Uhr-Daten

- Wird das Subtraktionsergebnis kleiner als 0 Stunden (00:00:00), wird der Rest von 24 Stunden abgezogen („vorhergehender Tag“) und das Borrow-Flag M8021 gesetzt.

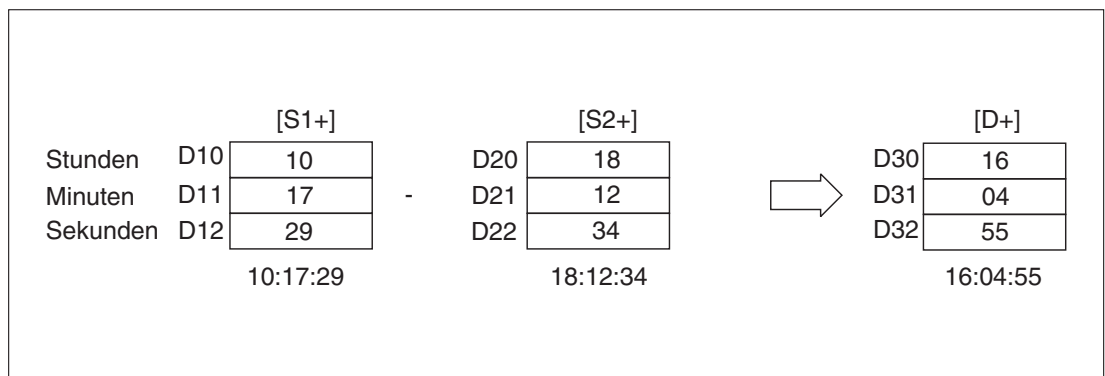


Abb. 7-147: Subtraktion von Uhr-Daten mit Stundenunterlauf

- Ist das Subtraktionsergebnis 0 (00:00:00, 0 Stunden, 0 Minuten, 0 Sekunden), wird das Zero-Flag M8020 gesetzt.
- Als Quelle und Ziel können dieselben Operanden ((S1+) bis ((S1+)+2), (S2+) bis ((S2+)+2)) verwendet werden. In diesem Fall wird das errechnete Ergebnis wieder in dem Quelloperanden gespeichert und anschließend für die nächste Berechnung genutzt. Dieser Prozess wiederholt sich mit bei jeder Ausführung der TSUB-Anweisung. Um eine einmalige Ausführung zu gewährleisten, sind gepulste Anweisungen (TSUBP) oder Verriegelungen zu verwenden.

HINWEIS

Für eine Rechenoperation mit den aktuellen Daten der integrierten Uhr kann eine TRD-Anweisung ausgeführt und danach der Inhalt der dort angegebene Zielregister ausgewertet werden. Alternativ können auch die Inhalte der Sonderregister D8015 (Stunden), D8014 (Minuten) und D8013 (Sekunden) verwendet werden.

Beispiel ▾

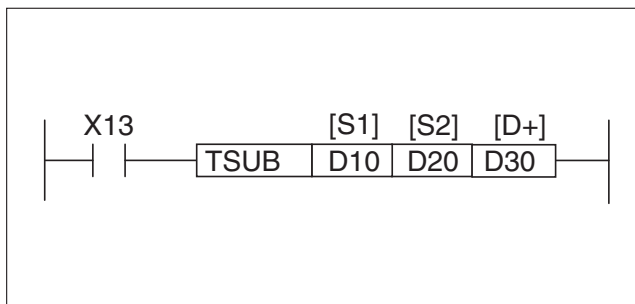


Abb. 7-148:
Programmierbeispiel zur TSUB-Anweisung

Bei eingeschaltetem Eingang X13 werden die Uhr-Daten in D20 bis D22 von den Uhr-Daten in D10 bis D13 subtrahiert. Das Ergebnis wird in D30 bis D32 gespeichert.

△

7.10.5 Wandlung der Einheit „Stunden“ in „Sekunden“ (HTOS)

		HTOS		FNC 164				
		Zeitangabe in der Form „Stunden, Minuten, Sekunden“ in Sekunden wandeln						
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
								●
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	KnX, KnY, KnM, KnS, T, C, D, R, U□/G□ Es werden 3 aufeinander folgende Operandenadressen verwendet.	KnY, KnM, KnS, T, C, D, R, U□/G□	●	16 Bit	32 Bit	HTOS HTOSP	5	
			●	●	DHTOS DHTOSP	9		

Funktion

Eine Zeitangabe in eine andere Einheit umrechnen

Beschreibung (16-Bit-Verarbeitung)

- Eine Zeitangabe in (S+) bis ((S+)+2) (Stunde, Minute und Sekunde) wird mit der HTOS-Anweisung in eine Zeitangabe mit der Einheit „Sekunden“ umgerechnet. Das Ergebnis wird in (D+) gespeichert.

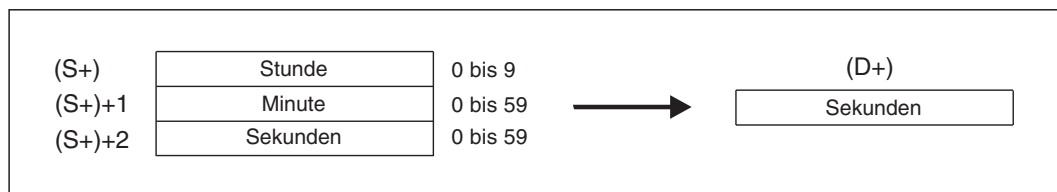


Abb. 7-149: Funktionsweise der HTOS-Anweisung

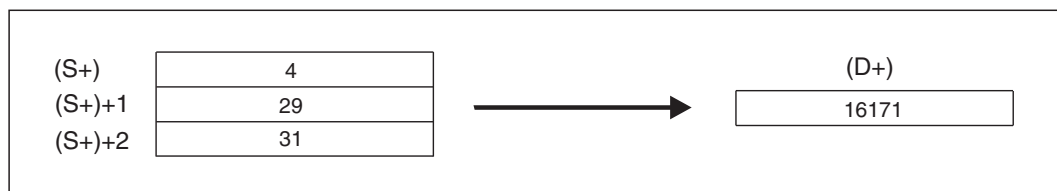


Abb. 7-150: Beispiel für die Umrechnung einer Zeitangabe: 4 Stunden, 29 Minuten und 31 Sekunden entsprechen 16171 Sekunden.

Beschreibung (32-Bit-Verarbeitung)

- Eine Zeitangabe in (S+) bis ((S+)+2) (Stunde, Minute und Sekunde) wird mit der DHTOS-Anweisung in eine Zeitangabe mit der Einheit „Sekunden“ umgerechnet. Das Ergebnis wird in ((D+)+1) und (D+) gespeichert.

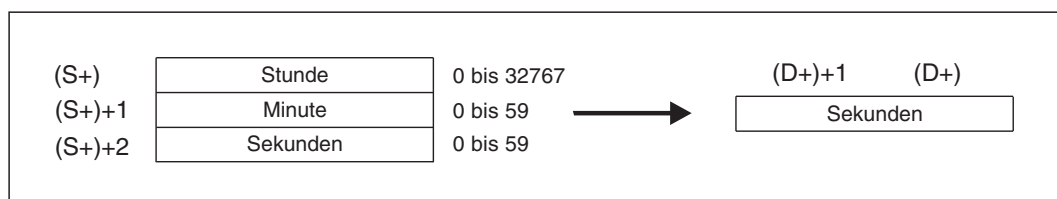


Abb. 7-151: Funktionsweise der DHTOS-Anweisung

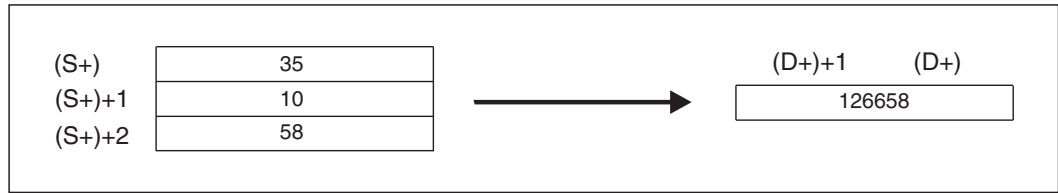


Abb. 7-152: Beispiel für die Umrechnung einer Zeitangabe: 35 Stunden, 10 Minuten und 58 Sekunden entsprechen 126658 Sekunden.

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Ein Wert in (S+), ((S+)+1) oder ((S+)+1) liegt außerhalb des zulässigen Bereichs.

Beispiel ▾

Das folgende Beispielprogramm liest die Uhrzeit und das Datum aus der integrierten Uhr der SPS und trägt diese Daten in die Datenregister D10 bis D16 ein. Anschließend werden diese Daten mit einer DHTOS-Anweisung in eine Zeitangabe mit der Einheit „Sekunden“ gewandelt.

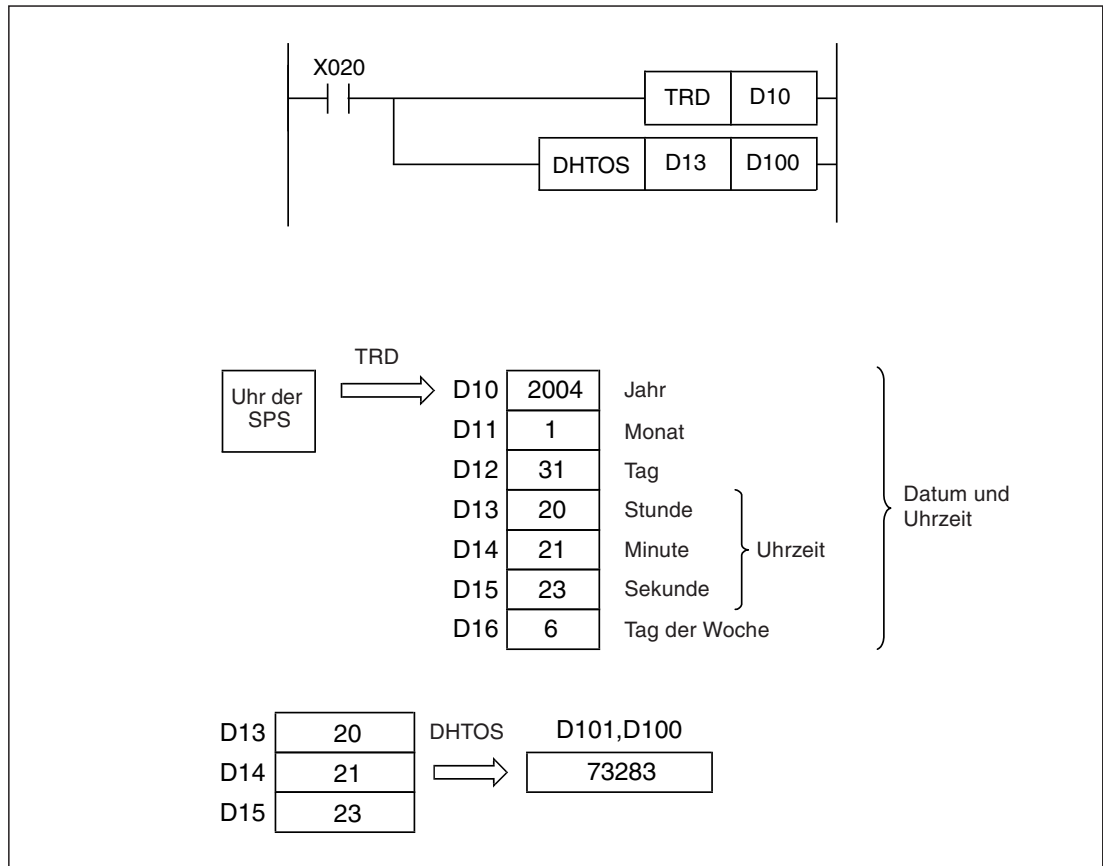
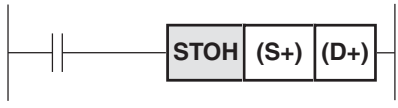


Abb. 7-153: Nachdem das Datum und die Uhrzeit mit einer TRD-Anweisung gelesen wurden, rechnet eine DHTOS-Anweisung die Einheit um.



7.10.6 Wandlung der Einheit „Sekunden“ in „Stunden“ (STOH)

		STOH		FNC 165				
		Zeitangabe in Sekunden in die Form „Stunden, Minuten, Sekunden“ wandeln						
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
								●
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	KnX, KnY, KnM, KnS, T, C, D, R, U□/G□	KnY, KnM, KnS, T, C, D, R, U□/G□ Es werden 3 aufeinander folgende Operanden- adressen verwendet.	●	16 Bit	32 Bit	STOH STOHP	5	
				●	●	DSTOH DSTOHP	9	

Funktion

Eine Zeitangabe in eine andere Einheit umrechnen

Beschreibung (16-Bit-Verarbeitung)

- Mit der STO H-Anweisung wird eine Zeitangabe, die in (S+) in der Einheit „Sekunden“ angegeben wird, in eine Angabe gewandelt, bei der dieser Zeitraum in Stunden, Minuten und Sekunden ausgedrückt wird. Das Ergebnis wird in (D+) bis ((D+)+2) gespeichert.



Abb. 7-154: Funktionsweise der STO H-Anweisung

- Der Wert in (S+) kann innerhalb des Bereichs von 0 bis 32767 [s] liegen.
- Der Wert in (D+) für die Stunden kann innerhalb des Bereichs von 0 bis 9 liegen.
- Die Angaben für die Minuten und Sekunden in ((D+)+1) bzw. ((D+)+2) können Werte von 0 bis 59 annehmen.



Abb. 7-155: Beispiel für die Anwendung einer STO H-Anweisung: 29011 Sekunden entsprechen 8 Stunden, 3 Minuten und 31 Sekunden.

Beschreibung (32-Bit-Verarbeitung)

- Die DSTOH-Anweisung wandelt eine Zeitangabe, die in ((S+)+1) und (S+) in der Einheit „Sekunden“ angegeben wird, in eine Form um, bei der dieser Zeitraum in Stunden, Minuten und Sekunden ausgedrückt wird. Das Ergebnis wird in (D+) bis ((D+)+2) gespeichert.



Abb. 7-156: Funktionsweise der DSTOH-Anweisung

- Der Wert in (S+) kann innerhalb des Bereichs von 0 bis 117964799 [s] liegen.
- Der Wert in (D+) für die Stunden kann innerhalb des Bereichs von 0 bis 32767 liegen.
- Die Angaben für die Minuten und die Sekunden in ((D+)+1) bzw. ((D+)+2) können Werte innerhalb des Bereichs von 0 bis 59 annehmen.



Abb. 7-157: Beispiel für die Anwendung einer DSTOH-Anweisung: 45325 Sekunden entsprechen 12 Stunden, 35 Minuten und 25 Sekunden.

Beispiel ▾

Mit dem folgenden Beispielprogramm wird ein Wert in „Sekunden“, der in D1 und D0 gespeichert ist, in das Format „Stunden, Minuten, Sekunden“ gewandelt und in D100 bis D102 gespeichert.

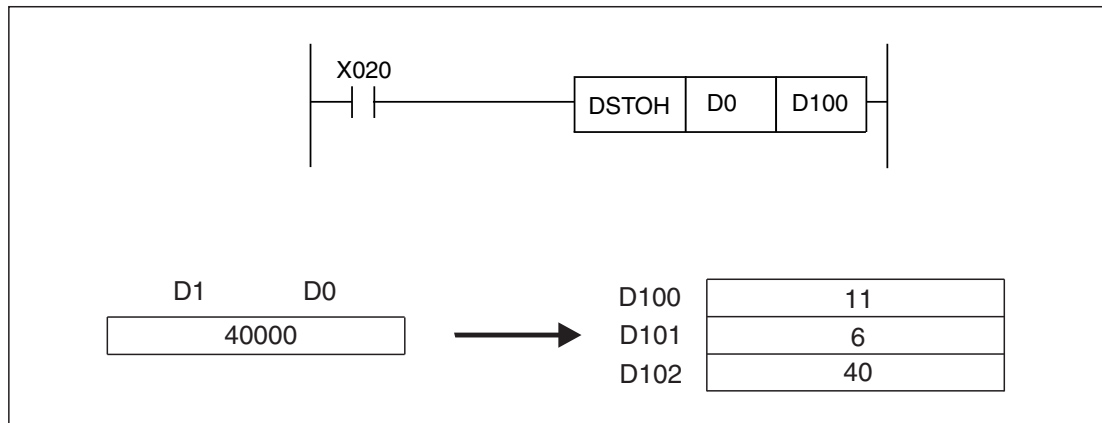


Abb. 7-158: In diesem Beispiel enthalten D1 und D0 den Wert „40000“ [Sekunden], der 11 Stunden, 6 Minuten und 40 Sekunden entspricht.

△

7.10.7 Lesen der Uhrzeit und des Datums (TRD)

		TRD		FNC 166			
		Lesen von Uhr-Daten					
Operanden	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	T, C, D, R ^① , U□/G□ ^② Es werden jeweils 7 aufeinanderfolgende Adressen des Operanden verwendet	●	16 Bit	32 Bit	TRD	3 ^③	
			●		TRDP	3 ^③	

- ① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ② Nur bei FX3U und FX3UC
- ③ Eine FX1N/FX1S/FX2N/FX2NC benötigt für die Ausführung einer TRD/TRDP-Anweisung 5 Schritte.

Funktion

Lesen des Datums und der Uhrzeit aus der internen Uhr der SPS

Beschreibung

- Mit der Anweisung werden die Uhr-Daten Jahr, Monat, Datum, Stunde, Minute, Sekunde und Wochentag aus der Uhr der SPS (Sonderregister D8013 bis D8019) gelesen.
- Diese Daten werden in 7 aufeinanderfolgenden Operanden ab (D+) gespeichert.

Operand	Bedeutung	Wertebereich	⇒	Operand	Bedeutung
D8018	Jahr	00-99	⇒	D+	Jahr
D8017	Monat	01-12	⇒	(D+)+1	Monat
D8016	Datum	01-31	⇒	(D+)+2	Datum
D8015	Stunden	00-23	⇒	(D+)+3	Stunden
D8014	Minuten	00-59	⇒	(D+)+4	Minuten
D8013	Sekunden	00-59	⇒	(D+)+5	Sekunden
D8019	Wochentag	0-6 (Sonntag-Samstag)	⇒	(D+)+6	Wochentag

Tab. 7-40: Lesen der Uhrzeit und des Datums aus den Sonderregistern D8013 bis D8019

HINWEISE

Die Jahreszahl wird normalerweise in der SPS als zweistelliger Wert gespeichert. Eine vierstellige Darstellung wird erreicht, wenn nach dem Start der SPS im ersten Programmzyklus der Wert „2000“ in das Sonderregister D8018 eingetragen wird:

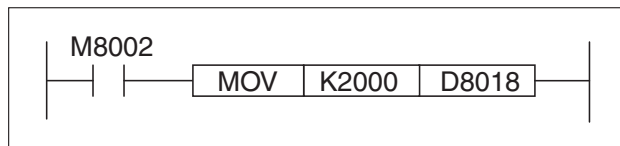


Abb. 7-159: Durch M8002 wird der Wert „2000“ nur im ersten Programmzyklus in D8018 eingetragen.

Die so erreichte vierstellige Darstellung ist der Jahreszahl ist dann nach der Bearbeitung des ersten Programmzyklus aktiv. Der Transfer von „2000“ in D8018 muss bei jedem Anlauf der SPS wiederholt werden. Die aktuelle Uhrzeit und das Datum werden dadurch nicht verändert.

Falls ein Bediengerät FX-10DU-E oder FX-20DU-E an die SPS angeschlossen wird, muss die zweistellige Darstellung der Jahreszahl aktiviert sein. Eine vierstellige Jahreszahl kann von diesen Geräten nicht angezeigt werden.

Beispiel ▾

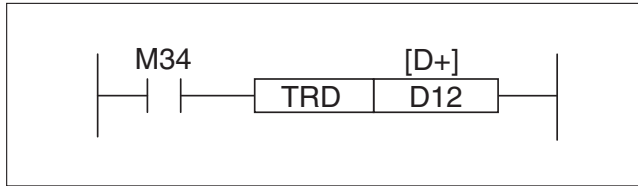


Abb. 7-160:

Bei gesetztem Merker M34 werden die Uhr-Daten ausgelesen und in den Registern D12 bis D18 gespeichert.



7.10.8 Interne Uhr der SPS stellen (TWR)

		TWR		FNC 167				
		Schreiben von Uhr-Daten						
		CPU	FX1S	FX1N	FX2N FX2NC	FX3S	FX3S	FX3U FX3UC
			●	●	●	●	●	●
Operanden	S+	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	T, C, D, R ^① , U□/G□ ^②	●	16 Bit	32 Bit	TWR	3 ^③		
	Es werden jeweils 7 aufeinanderfolgende Adressen des Operanden verwendet		●		TWRP	3 ^③		

- ① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC
- ② Nur bei FX3U und FX3UC
- ③ Eine FX1N/FX1S/FX2N/FX2NC benötigt für die Ausführung einer TWR/TWRP-Anweisung 5 Schritte.

Funktion

Stellen der SPS-Uhr

Beschreibung

- Mit der TWR-Anweisung werden die Uhrdaten Jahr, Monat, Datum, Stunde, Minute, Sekunde und Wochentag in die SPS-Uhr (Sonderregister D8013 bis D8019) übertragen.
- Diese Daten sind in 7 aufeinanderfolgenden Operanden ab (S+) gespeichert.

Operand	Bedeutung	Wertebereich		Operand	Bedeutung
S+	Jahr	00-99*	⇒	D8018	Jahr
(S+)+1	Monat	01-12	⇒	D8017	Monat
(S+)+2	Datum	01-31	⇒	D8016	Datum
(S+)+3	Stunden	00-23	⇒	D8015	Stunden
(S+)+4	Minuten	00-59	⇒	D8014	Minuten
(S+)+5	Sekunden	00-59	⇒	D8013	Sekunden
(S+)+6	Wochentag	0-6 (Sonntag-Samstag)	⇒	D8019	Wochentag

Tab. 7-41: Schreiben von Uhr-Daten

* Bei der vierstelligen Darstellung der Jahreszahl (siehe Abschnitt 7.10.7) geben die Werte „80“ bis „99“ die Jahre 1980 bis 1999 und die Werte „00“ bis „79“ die Jahre 2000 bis 2079 an. „06“ bedeutet zum Beispiel „2006“.

HINWEISE

Bei Verwendung der TWR-Anweisung ist das Setzen des Merkers M8015 (Anhalten der Echtzeituhr) nicht erforderlich.

Wird in (S+) bis ((S+)+6) ein Wert vorgegeben, der außerhalb des zulässigen Wertebereichs liegt, wird die Uhr nicht gestellt.

Beispiel ▾

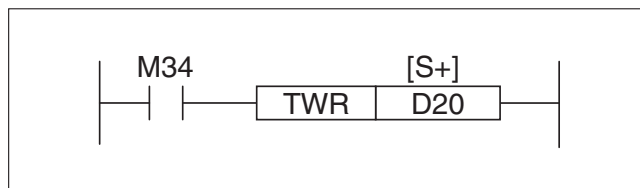


Abb. 7-161:

Mit Setzen des Merkers M34 werden die Uhr-Daten aus den Registern D20 bis D26 in die Uhr übertragen.



7.10.9 Betriebsstundenzähler (HOUR)

				HOUR		FNC 169				
				Betriebsstundenzähler						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
					●	●	●	●	●	●
Operanden	S+	D1+	D2+	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , U□/G□ ^② , V, Z,	D, R ^①	Y, M, S, D□.b ^②			16 Bit	32 Bit	HOUR	7	
					●	●	DHOUR	13		

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktion

Nach Ablauf einer einstellbaren Zeit wird ein Signal ausgegeben. Mit dieser Anweisung kann z. B. die Einschaltdauer von Geräten überwacht werden.

Beschreibung

- In (S+) wird die Zeit in der Einheit Stunden angegeben, nach der der in (D2+) angegebene Operand gesetzt wird.
- 16-Bit-Anweisung: (D1+) enthält die aktuelle Zahl der vollen Stunden. In ((D1+) + 1) wird die restliche Zeit in Sekunden abgelegt.
- 32-Bit-Anweisung: (D1+) und (D1+) enthalten die Angabe der vollen Stunden. In ((D1+) + 2) wird die restliche Zeit in Sekunden gespeichert.
- Der in (D2+) angegebene Operand wird bei der 16-Bit-Anweisung eingeschaltet, wenn der Istwert in (D1+) größer als der Sollwert in (S+) ist. Bei der 32-Bit-Anweisung wird der in (D2+) angegebene Operand eingeschaltet, wenn der Istwert in ((D1+)+1) und (D1+) größer ist als der Sollwert in (S+).

HINWEISE

Um den Istwert der abgelaufenen Zeit auch bei Spannungsausfall zu sichern, sollte für (D1+) ein gepuffertes Register verwendet werden.

Die Zählung wird auch fortgesetzt, nachdem der in (D2+) angegebene Operand eingeschaltet wurde. Die Zählung wird angehalten, wenn der maximale Wertebereich des 16-Bit- oder 32-Bit-Operanden in (D1+) überschritten wird.

Wenn das Signal in (D2+) in festen Intervallen gesetzt werden soll, muss der Istwert in den Registern (D1+) und ((D1+) + 1) (16-Bit) bzw. (D1+) bis ((D1+) + 2) (32-Bit) gelöscht werden, wenn der Sollwert erreicht wurde.

Beispiel ▾

HOUR-Anweisung

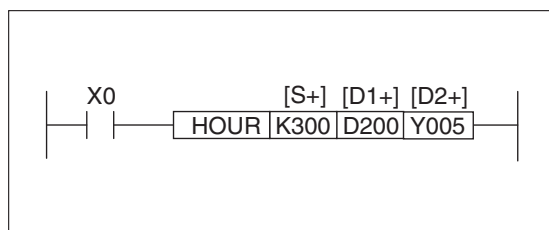


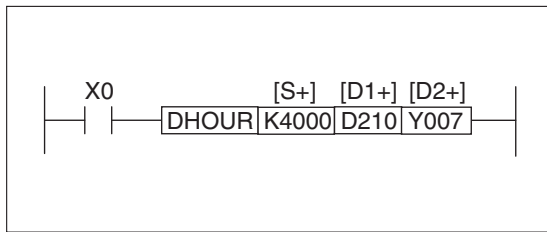
Abb. 7-162:

Programmbeispiel zum Einsatz der HOUR-Anweisung

Nach Ablauf von 300 Stunden und einer Sekunde, nachdem X0 eingeschaltet wurde, wird Y5 eingeschaltet.

Beispiel ▾

D HOUR-Anweisung

**Abb. 7-163:**

Programmbeispiel zum Einsatz der D HOUR-Anweisung

Zwischen dem Einschalten von X0 und Y7 vergehen 4000 Stunden und eine Sekunde.



7.11 Gray-Code-Anweisungen

Encoder, mit deren Hilfe die absolute Position von Maschinenelementen erfasst werden kann, übermitteln diese Information meist in einem besonderen Code an die SPS, dem Gray-Code. Beim Gray-Code ändert sich bei jedem Zifferschnitt nur ein Bit. Dadurch wird eine größere Übertragungssicherheit erreicht.

Übersicht der Anweisungen FNC 170 bis 171

Symbol	FNC	Bedeutung	Abschnitt
GRY	170	Umwandlung Integer in Gray-Code	7.11.1
GBIN	171	Umwandlung Gray-Code in Integer	7.11.2

Tab. 7-42: Übersicht der Anweisungen zur Umwandlung von Gray-Code

7.11.1 Umwandlung Integer in Gray-Code (GRY)

		GRY		FNC 170					
		Umwandlung Integer in Gray-Code							
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
					●	●	●	●	
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , U□/G□ ^② , V, Z	KnY, KnM, KnS, T, C, D, R ^① , U□/G□ ^② , V, Z	●	16 Bit	32 Bit	GRY GRYP	5		
				●	●	DGRY, DGRYP	9		

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktion

Konvertieren eines Integer-Wertes in den Gray-Code

Beschreibung

- Die GRY-Anweisung wandelt einen Integer-Wert ab (S+) in den Gray-Code.
- Das Ergebnis wird ab (D+) gespeichert.

HINWEIS

Durch die Charakteristik des Gray-Codes können numerische Werte durch inkrementieren der Quelldaten ohne ein Synchronisiersignal bei jedem Programmzyklus aktuell ausgegeben werden.

Beispiel ▾

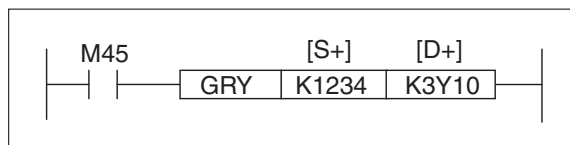


Abb. 7-164:

Programmierbeispiel zur GRY-Anweisung

Mit Setzen des Merkers M45 wird der Integer-Wert 1234 in den Gray-Code konvertiert. Das Ergebnis wird an den Ausgängen Y10 bis Y23 ausgegeben.



7.11.2 Umwandlung Gray-Code in Integer (GBIN)

		GBIN		FNC 171						
		Umwandlung Gray-Code in Integer								
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC				
			●	●	●	●				
Operanden	S+		D+		Puls-Anweisung (P)		Verarbeitung		Programmschritte	
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , U□/G□ ^② , V, Z		KnY, KnM, KnS, T, C, D, R ^① , U□/G□ ^② , V, Z		●		16 Bit	32 Bit	GBIN	5
							●	●	DGBIN	9
								DBBINP	9	

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktion

Konvertieren eines Wertes im Gray-Code in einen Integer-Wert

Beschreibung

- Mit der Anweisung erfolgt das Konvertieren des im Gray-Code codierten Wertes ab (S+) in einen Integer-Wert.
- Das Ergebnis wird ab (D+) gespeichert.

HINWEISE

Diese Anweisung kann zum Lesen von Daten aus einem Gray-Code-Codierer verwendet werden.

Wenn als Quelloperanden die Eingänge X0 bis X17 (X0 bis X7 bei einem FX3G-, FX3GC- oder FX3GE-Grundgerät) verwendet werden, lässt sich die Lesezeit durch Einstellung des Aktualisierungs-Filters (FNC51, REFF) verkürzen.

Beispiel ▾

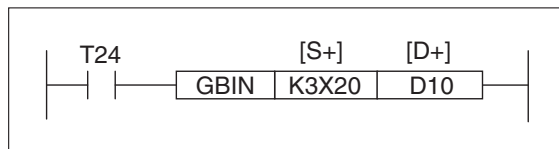


Abb. 7-165:
Programmierbeispiel zur GBIN-Anweisung

Mit Setzen des Timer-Kontakts T24 wird der Wert im Gray-Code an den Eingängen X20 bis X33 in einen Integer-Wert konvertiert. Das Ergebnis wird in D10 gespeichert.



7.12 Datenaustausch mit Analogmodulen

Symbol	FNC	Bedeutung	Abschnitt
RD3A	176	Lesen der analogen Eingangswerte	7.12.1
WR3A	177	Übertragen eines analogen Ausgangswertes	7.12.2

Tab. 7-43: Übersicht der Anweisungen zum Datenaustausch mit Analogmodulen

7.12.1 Lesen der analogen Eingangswerte (RD3A)

				RD3A		FNC 176				
				Lesen der analogen Eingangswerte						
				CPU	FX1s	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
						●	●	●	●	●
Operanden	n1+	n2+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	K, H, KnX, KnY, KnM, KnS, T, C, D, R*, V, Z		KnY, KnM, KnS, T, C, D, R*, V, Z		●	16 Bit	32 Bit	RD3A RD3AP	7	

* Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

Funktion

Lesen der analogen Eingangswerte aus den Analogmodulen FX0N-3A und FX2N-2AD

Beschreibung

- In (n1+) wird die Adresse des Sondermoduls angegeben (K0 bis K7).
- Der analoge Eingangswert des in (n2+) angegebenen Eingangskanals (K1/K21 oder K2/K22) wird in (D+) gespeichert.

Modul FX0N-3A: K1 = Kanal 1, K2 = Kanal 2; Wert in D = 0 bis 255 (8 Bit)

Modul FX2N-2AD: K21 = Kanal 1, K22 = Kanal 2; Wert in D = 0 bis 4095 (12 Bit)

HINWEISE

Vor dem Lesen der Analogwerte sollte die Eingangscharakteristik des Analogmoduls eingestellt werden. Nähere Hinweise hierzu finden Sie in der Bedienungsanleitung des Analogmoduls.

Steuerungen der FX1N-Serie können nur mit dem Modul FX0N-3A kommunizieren.

Ein FX0N-3A kann nur an ein Grundgerät der FX3U/FX3UC-Serie angeschlossen werden.

Beispiel ▾

RD3A-Anweisung

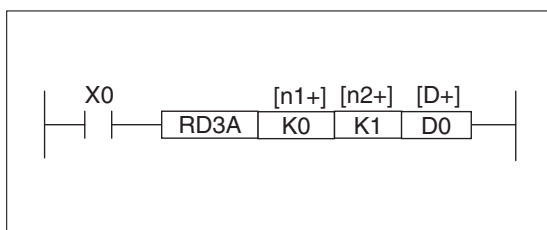


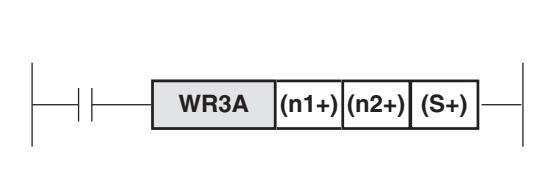
Abb. 7-166:

Programmbeispiel zum Einsatz der RD3A-Anweisung

Der an Kanal 1 gemessene Analogwert des Analogmoduls FX0N-3A mit der Adresse 0 wird in das Register D0 übertragen.



7.12.2 Schreiben eines analogen Ausgangswertes (WR3A)

				WR3A		FNC 177				
				Scheiben eines analogen Ausgangswertes						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
						●	●	●	●	●
Operanden	n1+	n2+	S+	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	K, H, KnX, KnY, KnM, KnS, T, C, D, R*, V, Z		KnY, KnM, KnS, T, C, D, R*, V, Z		●	16 Bit	32 Bit	WR3A WR3AP	7	

* Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

Funktion

Übertragen eines analogen Ausgangswertes an die Analogmodule FX0N-3A und FX2N-2DA

Beschreibung

- In (n1+) wird die Adresse des Sondermoduls angegeben (K0 bis K7).
- Der analoge Ausgangswert (S+) wird an den in (n2+) angegebenen Ausgangskanal (K1/K21 oder K22) übertragen.

Modul FX0N-3A: K1 = Kanal 1; Wert in (S+) = 0 bis 255 (8 Bit)

Modul FX2N-2AD: K21 = Kanal 1, K22 = Kanal 2; Wert in (S+) = 0 bis 4095 (12 Bit)

HINWEISE

Vor dem Übertragen des Ausgangswertes sollte die Ausgangscharakteristik des Analogmoduls eingestellt werden. Nähere Hinweise hierzu finden Sie in der Bedienungsanleitung des Analogmodul.

Steuerungen der FX1N-Serie können nur mit dem Modul FX0N-3A kommunizieren.

Ein FX0N-3A kann nur an ein Grundgerät der FX3U/FX3UC-Serie angeschlossen werden.

Beispiel ▾

WR3A-Anweisung

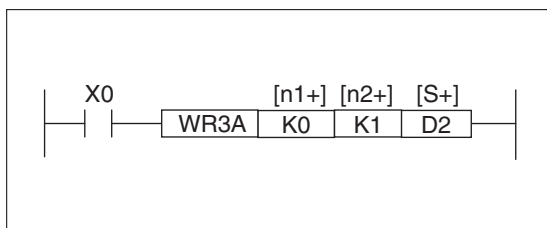


Abb. 7-167:

Programmbeispiel zum Einsatz der WR3A-Anweisung

Der Inhalt des Registers D2 wird zur Ausgabe als analoger Wert an das Analogmodul FX0N-3A mit der Adresse 0 übertragen.



7.13 Anweisungen aus externem Speicher

Symbol	FNC	Bedeutung	Abschnitt
EXTR	180	Ausführung einer Anweisung aus einer Speicherkassette.	7.13.1

Tab. 7-44: Anweisungen aus externem ROM ausführen

HINWEIS

Steuerungen der FX3G-, FX3GC-, FX3GE-, FX3S-, FX3U- und FX3UC-Serie können eine EXTR-Anweisung nicht ausführen. Bei diesen Grundgeräten können statt der Anweisungen EXTR K10 bis EXTR K13 die Anweisungen FNC270 bis FNC273 (IVCK, IVDR, IVRD und IVWR) verwendet werden. Bei einer FX3U/FX3UC steht zusätzlich die IVBWR-Anweisung zur Verfügung.

7.13.1 Anweisung aus externem ROM ausführen (EXTR)

		EXTR		FNC 180						
		Anweisung aus externem ROM ausführen								
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S FX3U FX3UC			
					●					
Operanden	S	SD1+, SD2+, SD3+		Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	K, H	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z, X, Y, M, S		●		16 Bit	32 Bit	EXTR EXTRP	9	
						●	●	DEXTR DEXTRP	17	

Funktion

Aufruf einer Anweisung, die in einer Speicherkassette FX2N-ROM-E1 oder FX2NC-ROM-CE1 abgelegt ist.

Beschreibung

- Durch den in (S+) angegebenen Wert (K0 bis K32767) wird die in der Speicherkassette abgelegte Anweisung ausgewählt.
- In (SD1+), (SD2+) und (SD3+) werden die Quell- oder Zieldaten eingetragen, die für die gewünschte Anweisung erforderlich sind. Ob die Operanden Quell- oder Zieldaten enthalten, hängt von der Art der Anweisung ab.
- Wenn einer der Operanden (SD1+), (SD2+) oder (SD3+) für die gewählte Anweisung nicht benötigt wird, muss an seiner Stelle K0 in die EXTR-Anweisung eingetragen werden.
- Die Datenlänge (16 Bit/32 Bit) und die Ausführungsart (gepulst oder zyklisch) wird durch die gewählte Anweisung bestimmt.

Die Speicherkassetten FX2N-ROM-E1 und FX2NC-ROM-CE dienen bei einer Steuerung der FX2N-Serie zur Erweiterung des Speichers und sind beide mit einem EEPROM (Speicherkapazität 16.000 Schritte) ausgestattet. Die Speicherkassette FX2N-ROM-E1 ist zu der Speicherkassette FX-EEPROM-16 und die Speicherkassette FX2NC-ROM-CE ist zu der Speicherkassette FX2NC-EEPROM-16C kompatibel.

HINWEIS

Die Speicherkassetten FX2N-ROM-E1 und FX2NC-ROM-CE können nur in Steuerungen der FX2N- und der FX2NC-Serie ab Version 3.00 verwendet werden.

Fehlermeldungen bei der Ausführung der EXTR-Anweisung

- Bei einer fehlerhaften Speicherkassette wird der Sondermerker M8061 gesetzt und in D8061 der Fehlercode 6110 eingetragen.
- Wenn die EXTR-Anweisung aufgerufen wird, während keine Speicherkassette installiert ist, wird der Sondermerker M8065 gesetzt und in D8065 der Fehlercode 6512 eingetragen.

EXTR K10 bis EXTR K13 (Kommunikation mit MITSUBISHI Frequenzumrichtern)

Funktion

Die in der Speicherkassette abgelegten Anweisungen K10 bis K13 ermöglichen in Verbindung mit einem Schnittstellenadapter FX2N-485-BD oder einem Kommunikationsmodul FX0N-485-BD den Datenaustausch mit Frequenzumrichtern der A500-, E500- und S500-Serie.

Anweisung	Funktion	Datenrichtung	Bemerkung
EXTR K10	Monitor-Funktion	Frequenzumrichter → SPS	siehe Bedienungsanleitungen der Frequenzumrichter, Kap. "Betrieb mit einem Personal Computer"
EXTR K11	Betrieb steuern	SPS → Frequenzumrichter	
EXTR K12	Parameter lesen	Frequenzumrichter → SPS	siehe detaillierte Beschreibung der Parameter in den Bedienungsanleitungen der Frequenzumrichter
EXTR K13	Parameter schreiben	SPS → Frequenzumrichter	

Tab. 7-45: Anweisungen zum Datenaustausch mit MITSUBISHI Frequenzumrichtern

HINWEIS

Die 6-stelligen Anweisungen der Frequenzumrichter der E500- und S500-Serie werden von der EXTR-Anweisung nicht unterstützt.

Beschreibung

- Wenn die Startbedingung der EXTR-Anweisung erfüllt ist, wird kontinuierlich mit dem Frequenzumrichter kommuniziert.
Die Kommunikation beginnt, wenn die serielle Schnittstelle aktiviert ist. Ist während des Datenaustausches die Startbedingung der EXTR-Anweisung nicht mehr erfüllt, wird trotzdem die Kommunikation fortgesetzt, bis der Datenaustausch abgeschlossen ist.
- Wenn im Programm die Startbedingungen mehrerer EXTR-Anweisungen gleichzeitig erfüllt sind, wird nach der Ausführung einer Anweisung automatisch die nächste EXTR-Anweisung aktiviert. Im Sonderregister D8155 wird die Programm-Schrittnummer der Anweisung abgelegt, die gerade ausgeführt wird.
- Nach der Ausführung einer EXTR-Anweisung wird unabhängig vom Fehler-Status M8029 gesetzt. M8029 wird auch von anderen Anweisungen verwendet. Daher bleibt dieser Merker nur solange gesetzt, bis eine Anweisung, die ebenfalls M8029 beeinflusst, ausgeführt wird.
- Eine EXTR-Anweisung darf nicht innerhalb einer FOR-NEXT-Schleife verwendet werden.
- Innerhalb eines Interrupt-Programms darf keine EXTR-Anweisung aufgerufen werden.
- Eine EXTR-Anweisung benötigt mehrere Programmzyklen zur vollständigen Ausführung. Achten Sie beim Programmieren von Unterprogrammen darauf, dass eine EXTR-Anweisung nicht mehrfach in einem Programmzyklus aufgerufen wird.
- Wird eine EXTR-Anweisung mit einer CJ- oder CJP-Anweisung übersprungen, wird die Kommunikationsschnittstelle nicht geöffnet und dadurch der Datenaustausch gestoppt.
- In Verbindung mit Master-Control-Anweisungen (MC, MCR) kann die EXTR-Anweisung uneingeschränkt genutzt werden.

- Wird in Verbindung mit einer STL-Anweisung ein Schritt mit einer EXTR-Anweisung deaktiviert, wird die Kommunikationsschnittstelle geschlossen und dadurch der Datenaustausch gestoppt.
- Wenn in der Betriebsart RUN Programmänderungen in der SPS vorgenommen werden, darf der Operand S (K10, K11, K12 oder K13) der EXTR-Anweisung nicht verändert und eine EXTR-Anweisung nicht gelöscht werden.

Einstellungen zur Kommunikation in der SPS

Die Anweisungen EXTR K10 bis EXTR K13 tauschen über den Schnittstellenadapter FX2N-485-BD bzw. dem Kommunikationsmodul FX0N-485-BD die Daten in derselben Weise aus wie die RS-Anweisung (FNC 80).

Die serielle Schnittstelle können Sie in der Programmier-Software GX Developer oder GX IEC Developer im Menüpunkt „SPS-Parameter“ parametrieren. Wählen Sie die folgenden Einstellungen:

- 7 Bit Datenlänge
- Gerade Parität
- 1 Stopp-Bit
- Übertragungsgeschwindigkeit entweder 4800, 9600 oder 19200 Baud

Einstellungen zur Kommunikation im Frequenzumrichter

Beim angeschlossenen Frequenzumrichter sind die folgenden Einstellungen erforderlich:

Merkmal		Einstellung					
Übertragungsstandard		RS 485					
Verbindungsart		1:n (n = max. 8 Stationen)					
Übertragungsgeschwindigkeit		wahlweise 4800, 9600 oder 19200 Baud					
Übertragungsart		Asynchron					
Kommunikationsart		Halb-Duplex					
Kommunikation	Datenlänge	7 Bit					
	Anzahl der Stoppbits	1					
	Ende-Zeichen	CR					
	Paritätsprüfung	Gerade Parität					
	Prüfsumme	Fest					
	Wartezeit	<table border="1"> <tr> <td>A500-Serie</td> <td rowspan="3">9999 (Einstellung durch Kommunikationsdaten)</td> </tr> <tr> <td>E500-Serie</td> </tr> <tr> <td>S500-Serie</td> </tr> <tr> <td>A500 + Option A5NR</td> <td>Zeit so kurz wie möglich einstellen</td> </tr> </table>	A500-Serie	9999 (Einstellung durch Kommunikationsdaten)	E500-Serie	S500-Serie	A500 + Option A5NR
A500-Serie	9999 (Einstellung durch Kommunikationsdaten)						
E500-Serie							
S500-Serie							
A500 + Option A5NR	Zeit so kurz wie möglich einstellen						

Tab. 7-46: Kommunikationsdaten der Frequenzumrichter

Anpassung der Parameter der Frequenzumrichter

Parameter	Funktion	Erforderliche Einstellung	Bemerkung
117	Stationsnummer	Die eingestellte Stationsnummer muss mit der im SPS-Programm verwendeten Stationsnummer übereinstimmen.	Beim Anschluss mehrerer Frequenzumrichter an die SPS wird über die Stationsnummer ein bestimmter Umrichter ausgewählt.
118	Übertragungsgeschwindigkeit	192 (19200 Baud) 96 (9600 Baud) 48 (4800 Baud)	Wählen Sie „192“. Bei High-Speed-Anweisungen in der SPS wählen Sie „96“ oder „48“.
119	Datenlänge Anzahl der Stoppbits	10	10 = 7 Datenbits, 1 Stoppbit
120	Paritätsprüfung	2	Prüfung auf gerade Parität
121	Anzahl der Wiederholungsversuche	0 bis 10	Schreiben Sie während der Inbetriebnahme „9999“ in diese Parameter und passen Sie den Wert nach der Inbetriebnahme an.
122	Zeitintervall der Kommunikationsprüfung	0,1 bis 999,8	
123	Wartezeit	9999	Die Wartezeit wird durch die Kommunikationsdaten vorgegeben.
124	Endekennzeichen	1	Als Endekennzeichen wird „CR“ verwendet.

Tab. 7-47: Parametrierung für einen Frequenzumrichter der A500- und E500-Serie

Parameter	Funktion	Erforderliche Einstellung	Bemerkung
331	Stationsnummer	Die eingestellte Stationsnummer muss mit der im SPS-Programm verwendeten Stationsnummer übereinstimmen.	Beim Anschluss mehrerer Frequenzumrichter an die SPS wird über die Stationsnummer ein bestimmter Umrichter ausgewählt.
332	Übertragungsgeschwindigkeit	192 (19200 Baud) 96 (9600 Baud) 48 (4800 Baud)	Wählen Sie „192“. Bei High-Speed-Anweisungen in der SPS wählen Sie „96“ oder „48“.
333	Datenlänge Anzahl der Stoppbits	10	10 = 7 Datenbits, 1 Stoppbit
334	Paritätsprüfung	2	Prüfung auf gerade Parität
335	Anzahl der Wiederholungsversuche	0 bis 10	Schreiben Sie während der Inbetriebnahme „9999“ in diese Parameter und passen Sie den Wert nach der Inbetriebnahme an.
336	Zeitintervall der Kommunikationsprüfung	0 bis 999,8	
337	Wartezeit	9999*	
338	Betriebskommando schreiben	0 oder 1	Wählen Sie die Einstellung entsprechend Ihren Systemanforderungen.
339	Drehzahlkommando schreiben	0 oder 1	
340	Auswahl der Betriebsart	0, 1 oder 2	
341	Endekennzeichen	1	Als Endekennzeichen wird „CR“ verwendet.
342	Auswahl EEPROM-Zugriff	0 oder 1	0 = Parameter werden ins EEPROM eingetragen. 1 = Parameter werden ins RAM eingetragen.

Tab. 7-48: Parametrierung für einen Frequenzumrichter der A500-Serie mit dem Optionsmodul A5NR

* Zur Vermeidung von Gefahren und um den Umrichter bei Kommunikationsstörungen umgehend abzuschalten, sollte die Wartezeit auf den kleinsten möglichen Wert eingestellt werden.

Parameter	Funktion	Erforderliche Einstellung	Bemerkung
n1	Stationsnummer	Die eingestellte Stationsnummer muss mit der im SPS-Programm verwendeten Stationsnummer übereinstimmen.	Beim Anschluss mehrerer Frequenzumrichter an die SPS wird über die Stationsnummer ein bestimmter Umrichter ausgewählt.
n2	Übertragungsgeschwindigkeit	192 (19200 Baud) 96 (9600 Baud) 48 (4800 Baud)	Wählen Sie „192“ Bei High-Speed-Anweisungen in der SPS wählen Sie „96“ oder „48“
n3	Datenlänge Anzahl der Stoppbits	10	10 = 7 Datenbits, 1 Stoppbit
n4	Paritätsprüfung	2	Prüfung auf gerade Parität
n5	Anzahl der Wiederholungsversuche	0 bis 10	Schreiben Sie während der Inbetriebnahme „9999“ in diese Parameter und passen Sie den Wert nach der Inbetriebnahme an.
n6	Zeitintervall der Kommunikationsprüfung	0,1 bis 999,8	
n7	Wartezeit	9999	Die Wartezeit wird durch die Kommunikationsdaten vorgegeben.
n11	Endekennzeichen	1	Als Endekennzeichen wird „CR“ verwendet.

Tab. 7-49: Parametrierung für einen Frequenzumrichter der S500-Serie

Datenformat

Die Daten werden im ASCII-Code übertragen.

ENQ	Stationsadr. 6		Anweisung = 80H		Wartezeit = 0	Daten = 1234				Prüfsumme		CR
05H	30H	36H	38H	30H	30H	31H	32H	33H	34H	43H	38H	0DH

Abb. 7-168: Beispiel für den Datenaustausch zwischen SPS und Frequenzumrichter

Aus der Stationsadresse, der Anweisung, der Wartezeit und den Daten wird die Prüfsumme berechnet:

$$30H + 36H + 38H + 30H + 30H + 31H + 32H + 33H + 34H = 1C8H$$

(„C“ entspricht 43H im ASCII-Code, „8“ entspricht 38H im ASCII-Code)

Beispiel ▾

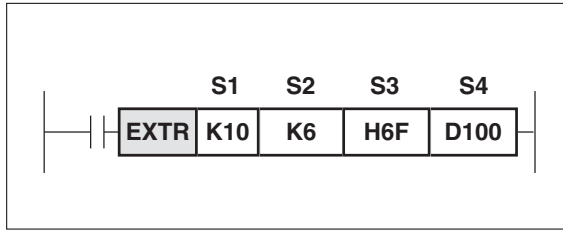


Abb. 7-169:
 Programmbeispiel zur Anweisung
 EXTR K10

In diesem Beispiel wird die Ausgangsfrequenz des Frequenzumrichters mit der Stationsnummer 6 gelesen und in D100 gespeichert.



Anweisung	Funktion	Verarbeitung		
		16 Bit	32 Bit	Puls-Anweisung (P)
EXTR K10	Monitor-Funktion	●		

Operanden		Beschreibung
S1	K, H	K10: Monitor-Funktion
S2	K, H, D	Stationsnummer des Frequenzumrichters (0 bis 31)
S3	K, H, D	Anweisungscode für den Frequenzumrichter (siehe folgende Tabelle)
S4	D, KnY, KnM, KnS	Operand, in dem die vom Frequenzumrichter gesendeten Daten abgelegt werden

Anweisungscode	Bedeutung der vom Frequenzumrichter gesendeten Daten	Frequenzumrichter-Serien, die den Anweisungscode unterstützen			Anzahl der Stellen	Datenformat
		A500	E500	S500		
H7B	Betriebsart	●	●	●	4	B ⇒ E F
H6F	Ausgangsfrequenz	●	●	●	4	B ⇒ E F
H70	Ausgangsstrom	●	●	●	4	B ⇒ E F
H71	Ausgangsspannung	●	●		4	B ⇒ E F
H72	Sonderüberwachung	●			4	B ⇒ E F
H73	Auswahlnummer zur Sonderüberwachung	●			2	B ⇒ E' F
H74	Alarm Definition	●	●	●	4	B ⇒ E F
H75	Alarm Definition	●	●	●	4	B ⇒ E F
H76	Alarm Definition	●	●		4	B ⇒ E F
H77	Alarm Definition	●	●		4	B ⇒ E F
H7A	Frequenzumrichter-Status	●	●	●	2	B ⇒ E' F
H6E	Eingestellte Ausgangsfrequenz (EEPROM)			●	4	B ⇒ E F
H6D	Eingestellte Ausgangsfrequenz (RAM)			●	4	B ⇒ E F

Tab. 7-50: Anweisungscode in S3 bei der EXTR K10-Anweisung

HINWEIS

Ausführliche Informationen zu den Anweisungs-codes und den Datenformaten finden Sie in den Bedienungsanleitungen der Frequenzumrichter.

Beispiel ▾

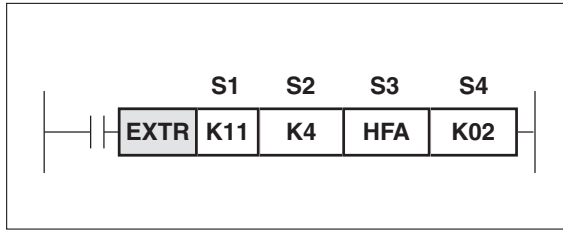


Abb. 7-170:
 Programmbeispiel zur Anweisung
 EXTR K11

Der Frequenzumrichter mit der Stationsnummer 4 erhält das Kommando zum Vorwärtslauf.



Anweisung	Funktion	Verarbeitung		
		16 Bit	32 Bit	Puls-Anweisung (P)
EXTR K11	Betrieb steuern	●		

Operanden		Beschreibung
S1	K, H	K11: Betrieb steuern
S2	K, H, D	Stationsnummer des Frequenzumrichters (0 bis 31)

Anweisungscode	Bedeutung der an den Frequenzumrichter gesendeten Daten	Frequenzumrichter-Serien, die den Anweisungscode unterstützen			Anzahl der Stellen	Datenformat
		A500	E500	S500		
HFB	Betriebsart	●	●	●	4	A ⇒ C D
HF3	Auswahlnummer zur Sonderüberwachung	●			2	A' ⇒ C D
HFA	Betriebssignal	●	●	●	2	A' ⇒ C D
HEE	Ausgangsfrequenz (Eintrag ins EEPROM)	●	●	●	4	A ⇒ C D
HED	Ausgangsfrequenz (Eintrag ins RAM)	●	●	●	4	A ⇒ C D
HFD	Frequenzumrichter zurücksetzen	●	●	●	4	A (ohne Antwort)
HF4	Alarmliste löschen	●		●	4	A ⇒ C D
HFC	Alle Parameter löschen	●	●	●	4	A ⇒ C D
HFC	Benutzerdefiniertes Löschen	●			4	A ⇒ C D

Tab. 7-51: Anweisungscode in S3 bei der EXTR K11-Anweisung

HINWEIS

Ausführliche Informationen zu den Anweisungs-codes und den Datenformaten finden Sie in den Bedienungsanleitungen der Frequenzumrichter.

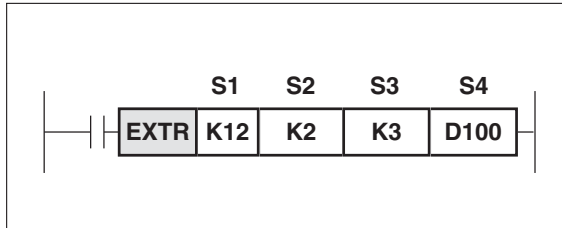
Beispiel ▾

Abb. 7-171:
 Programmbeispiel zur Anweisung
 EXTR K12

Die Basisfrequenz (Parameter 3) des Frequenzumrichters mit der Stationsnummer 2 wird gelesen und in D100 gespeichert.



Anweisung	Funktion	Verarbeitung		
		16 Bit	32 Bit	Puls-Anweisung (P)
EXTR K12	Parameter lesen	●		

Operanden		Beschreibung
S1	K, H	K12: Parameter aus dem Frequenzumrichter lesen
S2	K, H, D	Stationsnummer des Frequenzumrichters (0 bis 31)
S3	K, H, D	Parameter Nummer
S4	D, KnY, KnM, KnS	Operand, in dem die gelesenen Parameter abgelegt werden

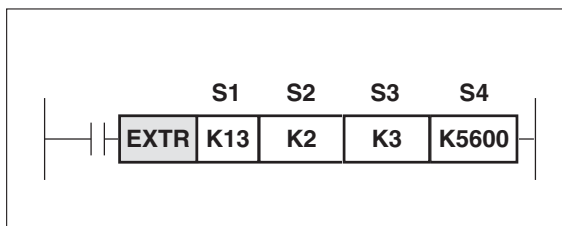
Beispiel ▾

Abb. 7-172:
 Programmbeispiel zur Anweisung
 EXTR K13

Bei diesem Beispiel wird als Basisfrequenz (Parameter 3) der Wert 5600 in den Frequenzumrichter mit der Stationsnummer 2 eingetragen.



Anweisung	Funktion	Verarbeitung		
		16 Bit	32 Bit	Puls-Anweisung (P)
EXTR K13	Parameter schreiben	●		

Operanden		Beschreibung
S1	K, H	K13: Parameter schreiben
S2	K, H, D	Stationsnummer des Frequenzumrichters (0 bis 31)

HINWEIS

Ausführliche Informationen zu den Parametern finden Sie in den Bedienungsanleitungen der Frequenzumrichter.

Zugriff auf erweiterte Parameter mit den Anweisungen EXTR 12 und EXTR 13

Einige Parameter der Frequenzumrichter bestehen aus einem Datensatz, der sich aus drei Einzeldaten zusammensetzt.

Um auf die einzelnen Daten zuzugreifen, wird zu der Parameternummer ein Offset von „0“, „1000“ oder „2000“ addiert. Die folgende Tabelle gibt den erforderlichen Wert in S3 der Anweisungen EXTR 12 und EXTR 13 beim Zugriff auf diese Parameter an.

Programmeinstellungen bei den Frequenzumrichtern der A500-Serie

Parameter	Bedeutung	Wert in S3 zum Lesen/Schreiben von		
		Frequenz	Zeit	Drehrichtung
201	Programmeinstellung 1	201	1201	2201
202		202	1202	2202
203		203	1203	2203
204		204	1204	2204
205		205	1205	2205
206		206	1206	2206
207		207	1207	2207
208		208	1208	2208
209		209	1209	2209
210		210	1210	2210
211	Programmeinstellung 2	211	1211	2211
212		212	1212	2212
213		213	1213	2213
214		214	1214	2214
215		215	1215	2215
216		216	1216	2216
217		217	1217	2217
218		218	1218	2218
219		219	1219	2219
220		220	1220	2220
221	Programmeinstellung 3	221	1221	2221
222		222	1222	2222
223		223	1223	2223
224		224	1224	2224
225		225	1225	2225
226		226	1226	2226
227		227	1227	2227
228		228	1228	2228
229		229	1229	2229
230		230	1230	2230

Tab. 7-52: Werte für S3 beim Zugriff auf erweiterte Parameter

Offset und Verstärkung der Sollwerteingabe bei der A500-, E500- und S500-Serie

Parameter	Bedeutung	Wert in S3 zum Lesen/Schreiben von		
		Offset Verstärkung (schreiben/lesen)	Analogwert (schreiben/lesen)	Analogwert des Eingangs (nur lesen)
902	Offset für Spannungs-Sollwert-eingabe	902	1902	2902
903	Verstärkung für Spannungs-Sollwert-eingabe	903	1903	2903
904	Offset für Strom-Sollwerteingabe	904	1904	2904
905	Verstärkung für Strom-Sollwert-eingabe	905	1905	2905

Tab. 7-53: Werte für S3 beim Zugriff auf Parameter für Sollwerteingabe**Verwendete Sondermerker und -register**

Operand	Beschreibung
M8154	M8145 = 0: Wartezeit 15 ms, M8145 = 1: Wartezeit 1000 ms Normalerweise wird nach der Abwicklung eines Datenaustausches eine Wartezeit von 15 ms eingehalten, bevor erneut eine Kommunikation gestartet wird. Durch Setzen von M8154 wird die Wartezeit auf 1 s verlängert und dadurch eine Fehlerdiagnose, z. B. durch Auswerten von D8155 oder D8156, ermöglicht.
M8155	M8155 = 0: Kommunikation ist beendet, M8155 = 1: Kommunikation läuft
M8156	M8156 = 1: Fehler bei der Kommunikation Werten Sie M8156 sofort nach der Ausführung einer EXTR-Anweisung aus. Bei der Ausführung der nächsten EXTR-Anweisung wird M8156 wieder gelöscht.
M8157	M8157 = 1: Fehler bei der Kommunikation (gespeichert) Beim Übergang der SPS von STOP nach RUN wird M8157 zurückgesetzt.
D8154	Wartezeit für die Antwort vom Frequenzumrichter (FU) Inhalt von D8154 = 0: Wartezeit = 2 s Inhalt von D8154 > 0: Wartezeit x 0,1 s Trifft nach dem Senden einer Kommunikationsanforderung durch die SPS innerhalb der eingestellten Wartezeit keine Antwort des FU ein, wird eine Fehlermeldung ausgegeben.
D8155	Schrittnummer der Anweisung zur Kommunikation mit dem Frequenzumrichter. Wenn nicht kommuniziert wird, enthält D8155 den Wert „-1“.
D8156	Fehlercode (siehe folgende Tabelle) Jeweils wenn bei der Ausführung der EXTR-Anweisung ein Fehler auftritt, wird D8156 aktualisiert. Beim Übergang der SPS von STOP nach RUN wird in D8156 „-1“ eingetragen.
D8157	Schrittnummer, bei der nach dem Start der SPS der erste Kommunikationsfehler aufgetreten ist. Die Schrittnummer bleibt gespeichert, bis beim Übergang der SPS von STOP nach RUN „-1“ in D8157 eingetragen wird.

Tab. 7-54: Sondermerker und -register bei der EXTR-Anweisung

Fehlercodes

Inhalt von D8156	Bedeutung	Verhalten des Frequenzumrichters
0000H	Datenaustausch fehlerfrei beendet	
0001H	Keine Antwort vom Frequenzumrichter	
0002H	Zeitüberschreitung bei der Kommunikation, Überschneidung mit M8129. Der Fehler tritt auf, wenn die Übertragung von Daten zur SPS abgebrochen wurde.	
0003H	Eine undefinierte Station hat geantwortet.	
0004H	Die Anzahl der vom Frequenzumrichter zurückgeschickten Daten ist fehlerhaft.	
0005H	Es wurde versucht, die Parameter 400 bis 899 zu lesen oder zu schreiben. Gleichzeitig wird in D8067 der Fehlercode 6702 eingetragen.	
0006H	Die Kommunikations-Schnittstelle wird für eine andere Funktion verwendet und steht für die EXTR-Anweisung nicht zur Verfügung. Gleichzeitig wird in D8067 der Fehlercode 6702 eingetragen.	
0100H	Der Frequenzumrichter hat den Fehlercode 0H (NAK-Fehler im externen Rechner) übermittelt. Die Anzahl aufeinanderfolgend gefundener Fehler in den Kommunikationsanforderungsdaten übersteigt die zulässige Anzahl der Wiederholungsversuche.	Tritt der Fehler öfter auf, als Wiederholungsversuche vorgesehen sind, kommt es zum Alarmstillstand.
0101H	Der Frequenzumrichter hat den Fehlercode 1H (Paritäts-Fehler) übermittelt. Das Ergebnis der Paritätsprüfung entspricht nicht der vorgegebenen Parität.	
0102H	Der Frequenzumrichter hat den Fehlercode 2H (Prüfsummen-Fehler) übermittelt. Die Prüfsummen in Frequenzumrichter und SPS sind unterschiedlich.	
0103H	Der Frequenzumrichter hat den Fehlercode 3H (Protokoll-Fehler) übermittelt. Das Protokoll der im Frequenzumrichter empfangenen Daten ist falsch, der Datenempfang wurde nicht in der vorgegebenen Zeit abgeschlossen, oder das Endekennzeichen (CR) entspricht nicht der Einstellung.	
0104H	Der Frequenzumrichter hat den Fehlercode 4H (Datenlängen-Fehler) übermittelt. Die Anzahl der Stoppbits weicht vom eingestellten Wert ab.	
0105H	Der Frequenzumrichter hat den Fehlercode 5H (Datenüberlauf) übermittelt. Von der SPS wurden Daten gesendet, bevor der Frequenzumrichter den Empfang der vorangegangenen Daten abgeschlossen hatte.	
0106H	Der Frequenzumrichter hat den Fehlercode 6H übermittelt. Dieser Fehlercode ist zur Zeit noch undefiniert.	
0107H	Der Frequenzumrichter hat den Fehlercode 7H (ungültiges Zeichen) übermittelt. Es wurde ein ungültiges Zeichen empfangen. Zulässig sind die Zeichen 0 bis 9, A bis F und Steuer-codes.	Der Frequenzumrichter akzeptiert die empfangenen Daten nicht, es kommt aber zu keinem Alarmstopp.
0108H	Der Frequenzumrichter hat den Fehlercode 8H übermittelt. Dieser Fehlercode ist zur Zeit noch undefiniert.	
0109H	Der Frequenzumrichter hat den Fehlercode 9H übermittelt. Dieser Fehlercode ist zur Zeit noch undefiniert.	

Tab. 7-55: Fehlercodes (1)

Inhalt von D8155	Bedeutung	Verhalten des Frequenzumrichters
010AH	Der Frequenzumrichter hat den Fehlercode AH (Betriebsart-Fehler) übermittelt. Es wurde versucht, einen Parameter zu schreiben, obwohl sich der Frequenzumrichter nicht im Rechner-Link-Betrieb befand oder während der Frequenzumrichter in Betrieb war.	Der Frequenzumrichter akzeptiert die empfangenen Daten nicht, es kommt aber zu keinem Alarmstopp.
010BH	Der Frequenzumrichter hat den Fehlercode BH (Anweisungs-Code-Fehler) übermittelt. Das angegebene Kommando existiert nicht.	
010CH	Der Frequenzumrichter hat den Fehlercode CH (Datenbereichs-Fehler) übermittelt. Beim Schreiben von Parametern wurden Daten übertragen, die den zulässigen Wertebereich überschreiten.	
010DH	Der Frequenzumrichter hat den Fehlercode DH übermittelt. Dieser Fehlercode ist zur Zeit noch undefiniert.	
010EH	Der Frequenzumrichter hat den Fehlercode EH übermittelt. Dieser Fehlercode ist zur Zeit noch undefiniert.	
010FH	Der Frequenzumrichter hat den Fehlercode FH übermittelt. Dieser Fehlercode ist zur Zeit noch undefiniert.	

Tab. 7-56: Fehlercodes (2)

Kommunikationsfehler

Einschließlich zweier Wiederholungen wird der Datenaustausch dreimal ausgeführt. Wenn auch nach der zweiten Wiederholung die Kommunikation fehlerhaft ist, wird eine Fehlermeldung ausgegeben. Dabei können die Fehler in verschiedene Klassen eingeteilt werden:

Art des Fehlers	Anzeige durch		Eintrag des Fehlercodes in D8156
	M8156	M8157	
Vom Frequenzumrichter wird ein Fehlercode übermittelt	●	●	●
Der Frequenzumrichter antwortet nicht	●	●	●
Eine unbekannte Station antwortet	●	●	●
Ein Empfangsfehler (Datenüberlauf, Paritäts-Fehler, fehlerhafte Datenlänge) ist aufgetreten.		●	
M8063 wird gesetzt und in D8067 wird der Fehlercode 6301 eingetragen		●	
Die Prüfsumme der vom Frequenzumrichter zurückgeschickten Daten stimmt nicht mit der in der SPS ermittelten Prüfsumme überein.		●	

Tab. 7-57: Behandlung der Kommunikationsfehler

Beispiel ▾

Beispielprogramm 1

Die Parameter 0 bis 99 des Frequenzumrichters mit der Stationsnummer 6 werden in die Datenregister D1000 bis D1099 eingetragen.

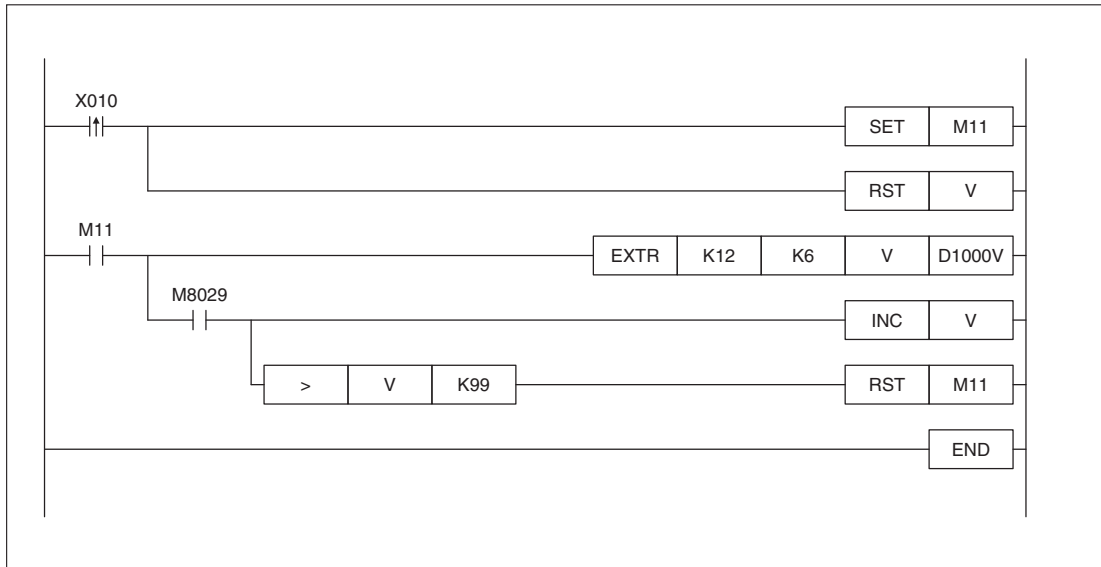


Abb. 7-173: Beispielprogramm zum Lesen von Parametern



Beispiel ▾

Beispielprogramm 2

Aus den Frequenzumrichtern mit den Stationsnummern 6, 7, 8 und 9 werden jeweils die Parameter 0 bis 99 ausgelesen und in die Bereiche D1000 bis D1099, D1100 bis D1199, D1200 bis D1299 bzw. D1300 bis D1399 gespeichert.

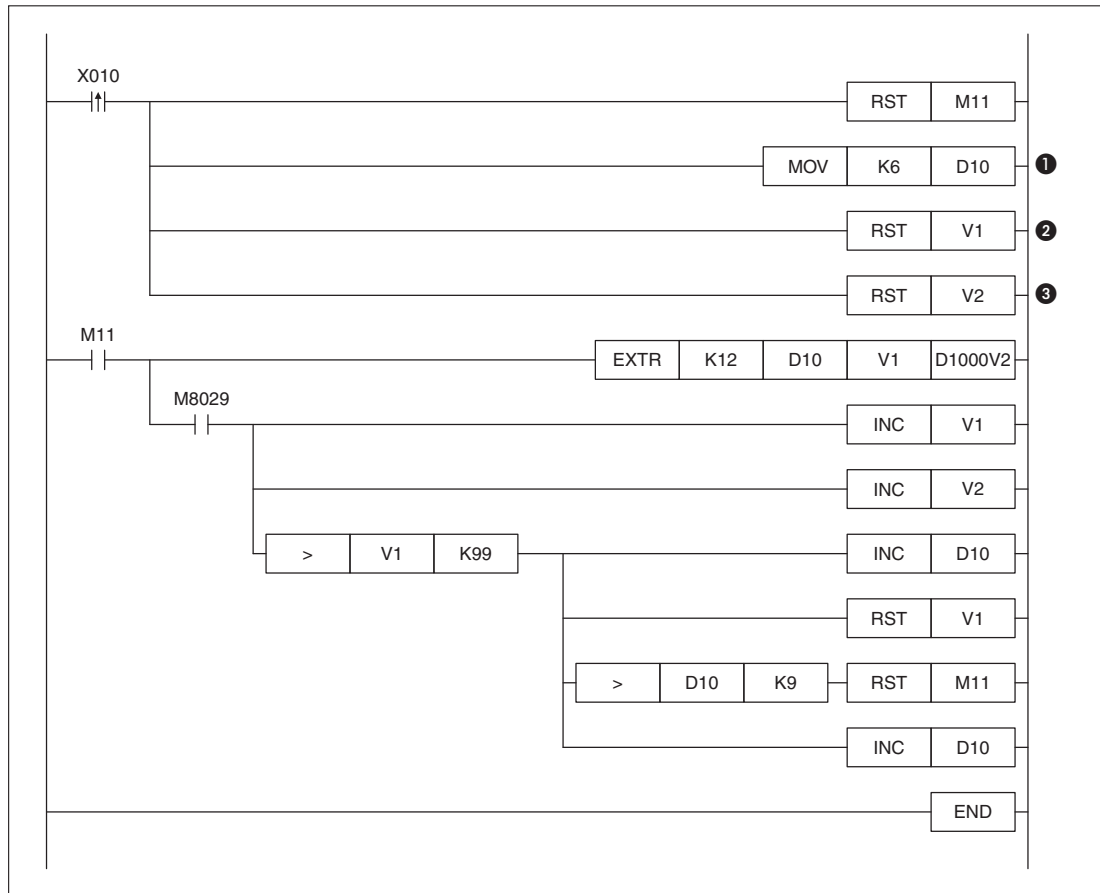


Abb. 7-174: Beispielprogramm zum Lesen der Parameter mehrerer Umrichter

- ① Im Datenregister D10 wird die Stationsnummer abgelegt.
- ② Das Index-Register V1 enthält die Nummer des Parameters (0 bis 99)
- ③ Index-Register V2 gibt an, wo der Parameter in der SPS gespeichert werden soll.

△

Beispiel ▾

Beispielprogramm 3

Funktionen: Übertragen der Grundparameter von der SPS zum Frequenzumrichter, Wahl der Drehrichtung mit X1 (Vorwärts) und X2 (Rückwärts), Änderung der Ausgangsfrequenz, Überwachen der Ausgangsfrequenz und des Ausgangsstromes.

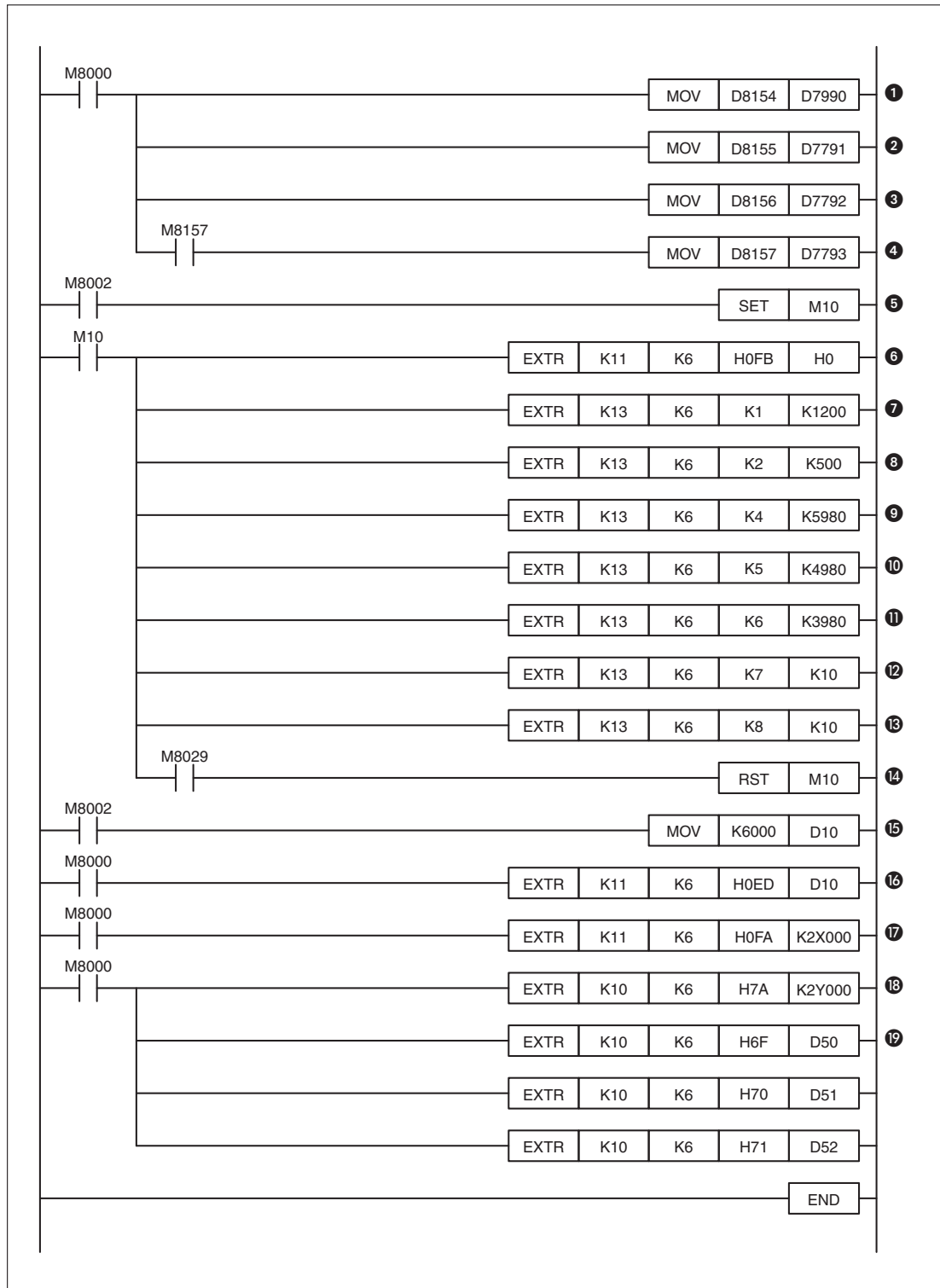


Abb. 7-175: Beispielprogramm zum Steuern und Beobachten der wichtigsten Umrichterparameter

In den Programmpunkten 1 bis 4 werden Daten zur Fehlerdiagnose für eine spätere Auswertung in Datenregister abgelegt.

- ① Wartezeit für eine Antwort des Frequenzumrichters in D7990 ablegen
- ② Schrittnummer der momentan ausgeführten Anweisung lesen
- ③ Fehlercode abspeichern
- ④ Nummer des Programmschrittes, bei dem ein Fehler aufgetreten ist, lesen
- ⑤ Die folgenden Parameter werden beim Anlauf der SPS zum Frequenzumrichter übertragen
- ⑥ Computer-Link aktivieren
- ⑦ Maximale Frequenz
- ⑧ Minimale Frequenz
- ⑨ 3. Geschwindigkeit
- ⑩ 2. Geschwindigkeit
- ⑪ 1. Geschwindigkeit
- ⑫ Beschleunigungszeit
- ⑬ Verzögerungszeit
- ⑭ Nach Ausführung der Anweisungen wird M10 zurückgesetzt.
- ⑮ Nach dem Start der SPS wird der Vorgabewert für die Frequenz in D10 eingetragen.
- ⑯ Die Frequenz wird ständig zum Frequenzumrichter übertragen. Zur Änderung der Frequenz wird der Inhalt von D10 geändert.
- ⑰ Die Eingänge X1 (Vorwärts) und X2 (Rückwärts) der SPS werden zur Steuerung des Umrichters verwendet.
- ⑱ Der Status des Frequenzumrichters wird an Ausgänge des SPS ausgegeben.
- ⑲ Ausgangsfrequenz, -strom und -spannung werden vom Frequenzumrichter gelesen und in D50, D51 und D52 gespeichert.

△

Beispiel ▾

Beispielprogramm 4

Wird beim Beispielprogramm 3 eine Monitor-Funktion (EXTR K10) ausgeführt, wird dadurch eine Frequenzänderung oder ein Kommando zur Drehrichtungsumkehr verzögert zum Frequenzumrichter übertragen.

Im folgendem Programm wird die Monitor-Funktion unterbrochen, wenn Daten zum Umrichter übertragen werden müssen. Nach dem Schreiben in den Frequenzumrichter wird die Monitorfunktion fortgesetzt.

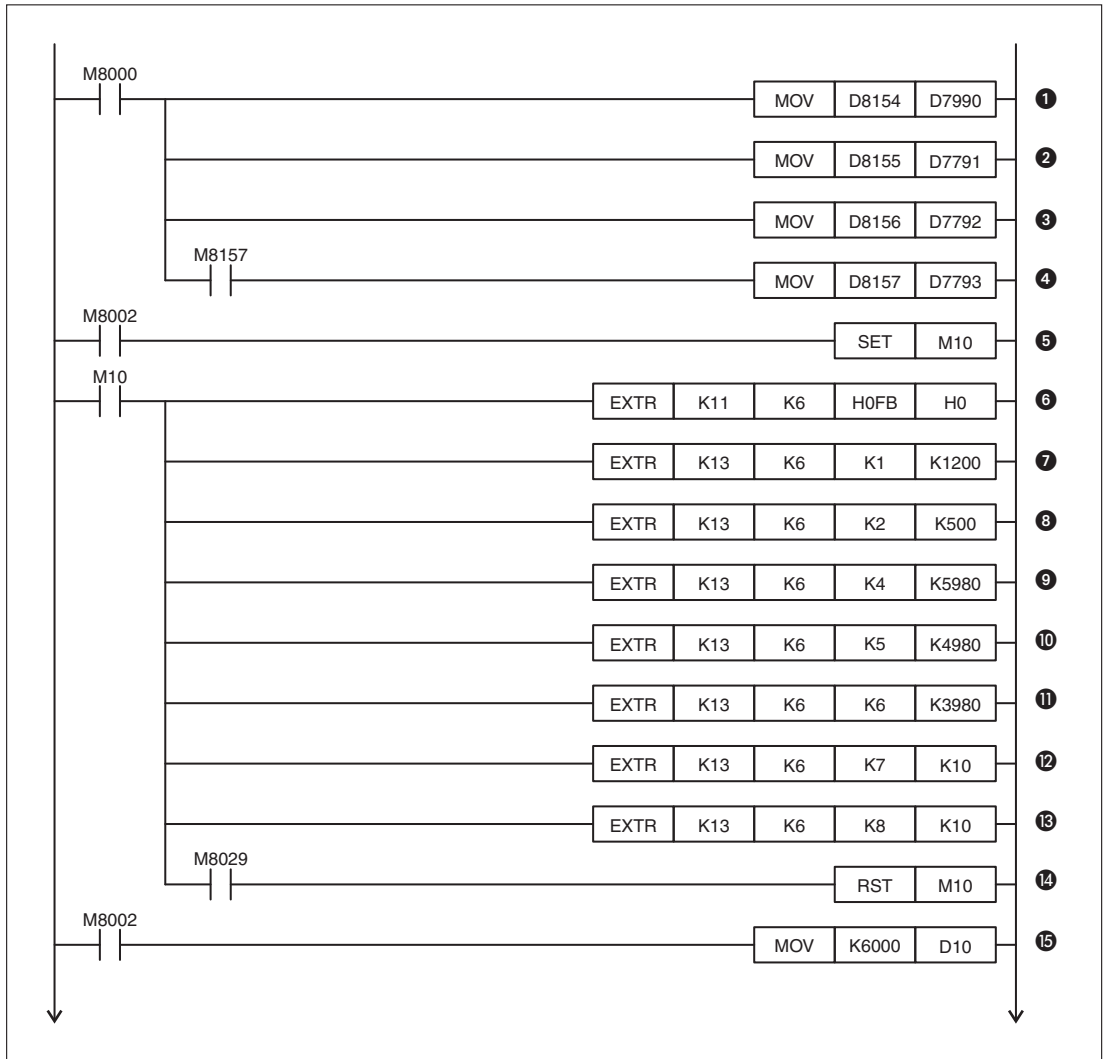


Abb. 7-176: Beispielprogramm mit Unterbrechung der Monitor-Funktion (Teil 1)

In den Programmpunkten 1 bis 4 werden Daten zur Fehlerdiagnose für eine spätere Auswertung in Datenregister abgelegt.

- ① Wartezeit für eine Antwort des Frequenzumrichters in D7990 ablegen
- ② Schrittnummer der momentan ausgeführten Anweisung lesen
- ③ Fehlercode abspeichern
- ④ Nummer des Programmschrittes, bei dem ein Fehler aufgetreten ist, lesen
- ⑤ Die folgenden Parameter werden beim Anlauf der SPS zum Frequenzumrichter übertragen
- ⑥ Computer-Link aktivieren
- ⑦ Maximale Frequenz
- ⑧ Minimale Frequenz
- ⑨ 3. Geschwindigkeit
- ⑩ 2. Geschwindigkeit
- ⑪ 1. Geschwindigkeit
- ⑫ Beschleunigungszeit
- ⑬ Verzögerungszeit
- ⑭ Nach Ausführung der Anweisungen wird M10 zurückgesetzt.

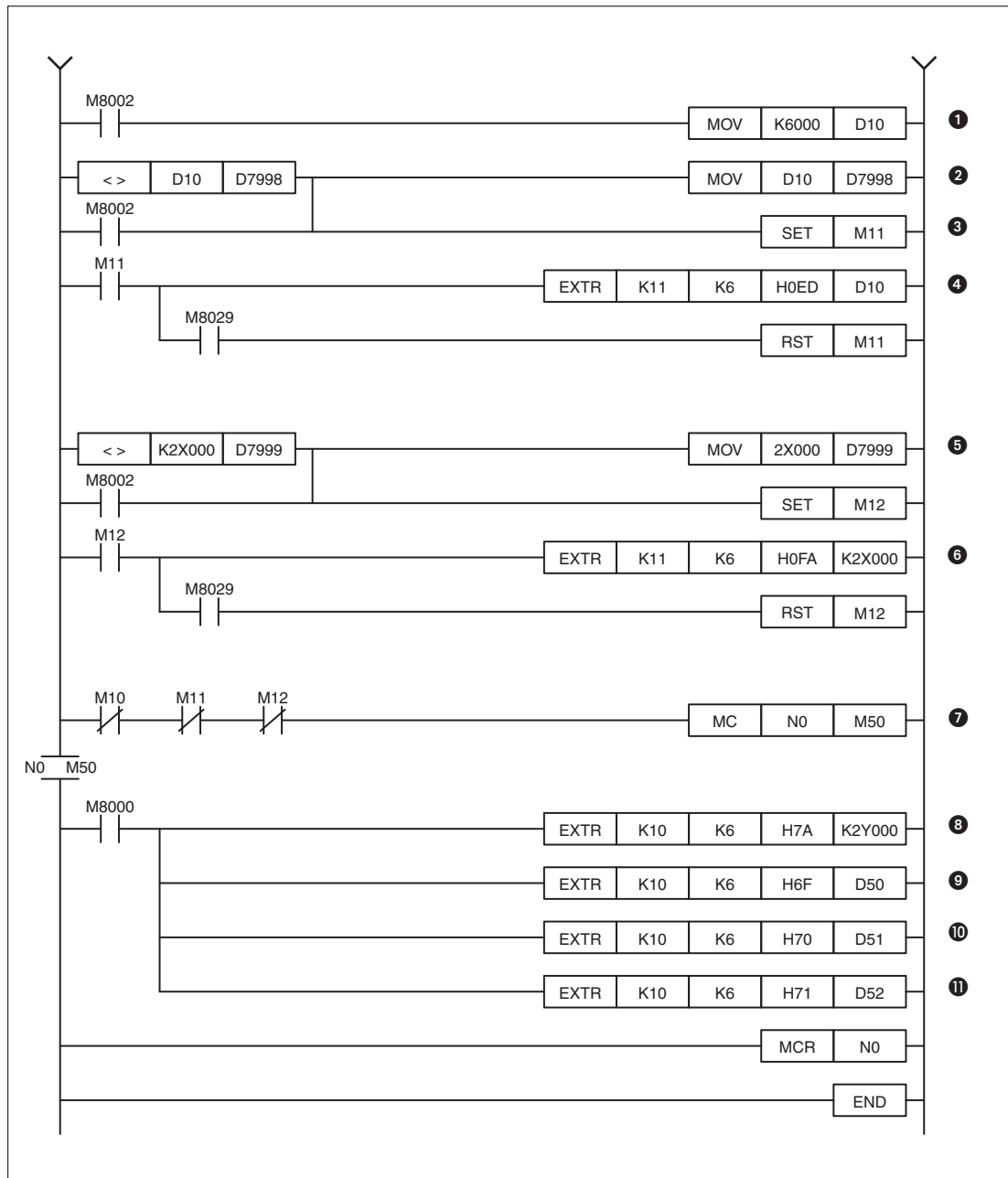


Abb. 7-177: Beispielprogramm mit Unterbrechung der Monitor-Funktion (Teil 2)

- ① Nach dem Start der SPS wird der Vorgabewert für die Frequenz in D10 eingetragen.
- ② Wenn der Inhalt von D10 geändert wurde, wird der aktuelle Inhalt für den nächsten Vergleich in D7998 gespeichert.
- ③ M11 wird als Anforderung zur Übertragung der Frequenz gesetzt, wenn D10 verändert oder die SPS gestartet wurde.
- ④ Der Sollwert der Frequenz wird zum Umrichter übertragen. Nach der Ausführung der Anweisung wird M11 zurückgesetzt.
- ⑤ Wenn sich einer der Eingänge X0 bis X7 ändert, wird der aktuelle Zustand für den nächsten Vergleich in D7999 gespeichert und M12 gesetzt.
- ⑥ Wenn M12 gesetzt ist, werden die Eingänge X1 (Vorwärts) und X2 (Rückwärts) der SPS zur Steuerung des Umrichters verwendet.
- ⑦ Die Monitor-Funktion wird nur aktiviert, wenn keine Parameter oder Steuerdaten zum Umrichter übertragen werden.
- ⑧ Der Status des Frequenzumrichters wird an Ausgänge des SPS ausgegeben.
- ⑨ Die Ausgangsfrequenz wird in D50 gespeichert.
- ⑩ In D51 wird der Ausgangsstrom abgelegt.
- ⑪ D52 enthält die Ausgangsspannung.

△

Beispiel ▾

Beispielprogramm 5

Beispiel zur Verwendung der EXTR-Anweisung in einer Schrittsteuerung (STL-Anweisung).

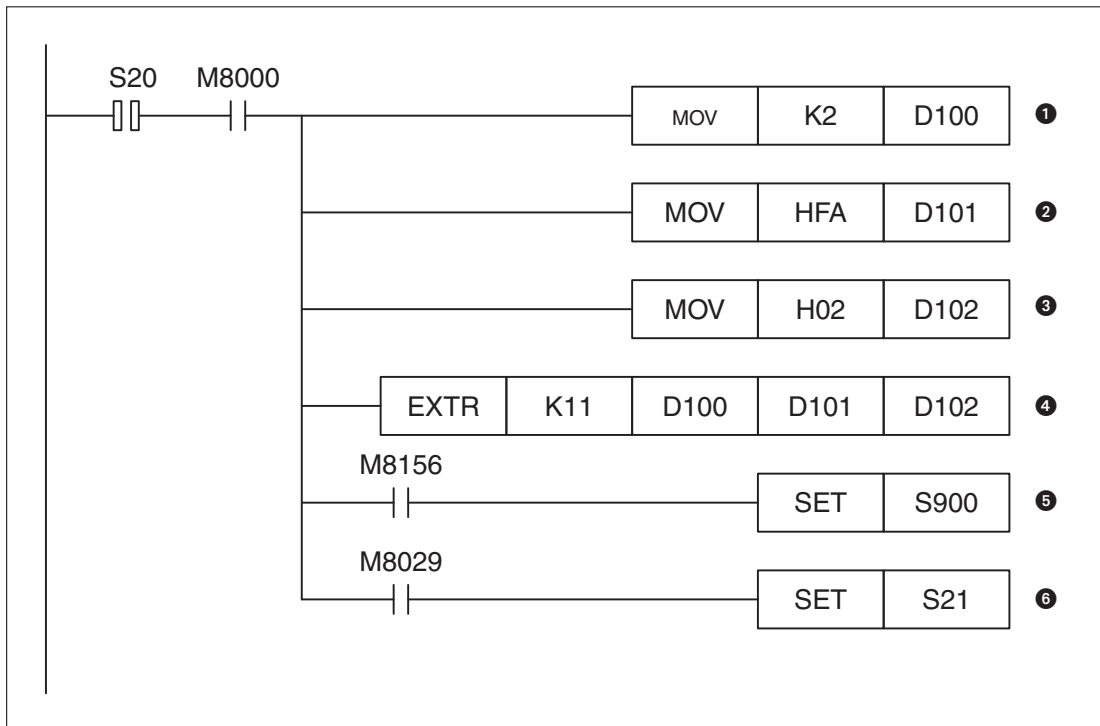


Abb. 7-178: Beispiel zum Aufruf der EXTR-Anweisung in einer Schrittsteuerung

- ① Die Stationsnummer „2“ wird in D100 eingetragen.
- ② In D101 wird der Code („HFA“) für Betriebssignale eingetragen.
- ③ D102 enthält die Anweisung für „Vorwärts“.
- ④ Das Betriebssignal „Vorwärts“ wird zum Frequenzumrichter übertragen.
- ⑤ Bei einem Fehler wird der Fehlerschritt aktiviert.
- ⑥ Wenn die Anweisung fehlerfrei ausgeführt wurde, wird der nächste Schritt aktiviert.



7.14 Verschiedene Anweisungen

Symbol	FNC	Bedeutung	Abschnitt
COMRD	182	Operandenkommentar lesen	7.14.1
RND	184	Zufallszahl generieren	7.14.2
DUTY	186	Taktgenerator mit einstellbarem Tastverhältnis	7.14.3
CRC	188	Daten prüfen (CRC-Prüfung)	7.14.4
HCMOV	189	Istwert eines High-Speed-Counters transferieren	7.14.5

Tab. 7-58: Übersicht der Anweisungen mit verschiedenen Funktionen

7.14.1 Operandenkommentar lesen (COMRD)

		COMRD	FNC 182					
		Operandenkommentar lesen und als Zeichenfolge im ASCII-Code speichern						
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
								●
Operanden	S+	D+	Puls-Anweisung (P)		Verarbeitung		Programmschritte	
	X, Y, M, S, T, C, D, R	T, C, D, R	●		16 Bit	32 Bit	COMRD COMRDP	
				●			5	

Funktion

Die COMRD-Anweisung liest Operandenkommentar und speichert ihn als ASCII-Code. Der Operandenkommentar wird bei der Programmierung eingegeben und kann mit dem Programm in die Steuerung übertragen werden.

Beschreibung

- In (S+) wird der Operand angegeben, dessen Kommentar gelesen werden soll.
- In (D+) wird der erste Operand des Bereichs angegeben, in dem der ASCII-Code abgelegt werden soll.
- Maximal können 16 Zeichen im ASCII-Code gespeichert werden. Ein Zeichen belegt 8 Bit.

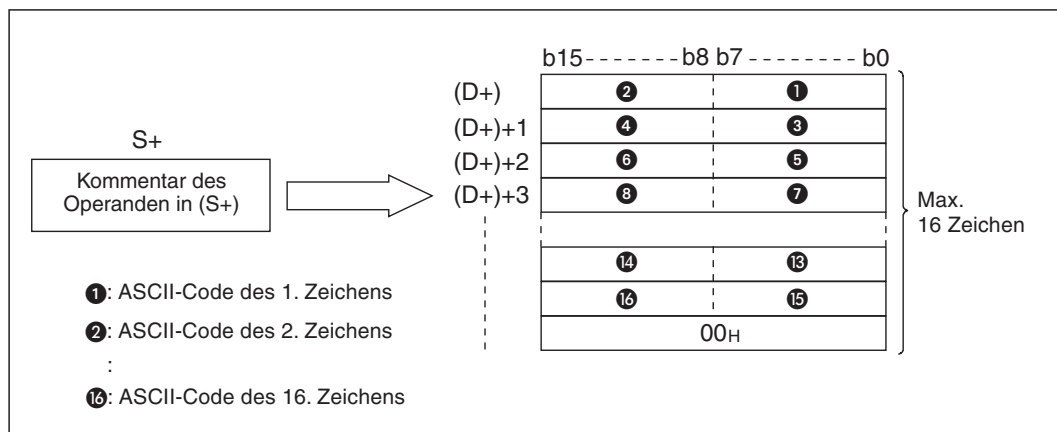


Abb. 7-179: Der Zustand von M8091 entscheidet, ob nach den ASCII-Zeichen der Code „00H“ angefügt wird.

- Der Inhalt des Wortes oder Bytes nach dem letzten ASCII-Zeichen hängt vom Zustand des Sondermerkers M8091 und der Anzahl (gerade oder ungerade) der Zeichen ab:

- M8091 = „0“

Bei einer ungeraden Anzahl von Zeichen wird im höherwertigen Byte des Operanden, in dem das letzte Zeichen geschrieben wurde, „00H“ eingetragen.

Bei einer geraden Anzahl von Zeichen wird im nächsten Operanden, nach dem letzten Zeichen, „00H“ eingetragen.

- M8091 = „1“

Bei einer ungeraden Anzahl von Zeichen wird der Inhalt des höherwertigen Byte des Operanden, in dem das letzte Zeichen geschrieben wurde, nicht verändert.

Bei einer geraden Anzahl von Zeichen wird nach dem letzten ASCII-Zeichen kein „00H“ angefügt.

Die folgende Abbildung zeigt ein Beispiel, bei dem M8091 auf „1“ gesetzt ist und der Kommentar aus einer ungeraden Anzahl Zeichen besteht.

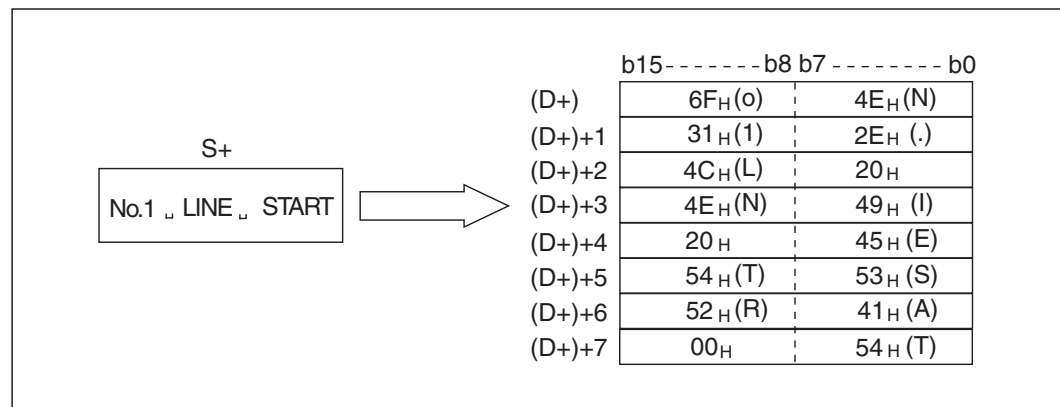


Abb. 7-180: Im Byte nach dem letzten Zeichen wird in diesem Beispiel „00H“ eingetragen.

- Falls für den in (S+) angegebenen Operanden kein Kommentar vorhanden ist, werden in (D+) bis ((D+)+8) 16 Leerzeichen (20H) eingetragen.

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Für den (S+) angegebenen Operanden ist kein Kommentar vorhanden.
- Durch die Speicherung der Zeichen ab (D+) wird der zulässige Bereich für den angegebenen Operanden überschritten.

Beispiel ▾

Für das Datenregister D100 wurde der Operandenkommentar „Target Line A“ eingetragen. Mit der folgenden Programmsequenz wird dieser Kommentar beim Einschalten von X10 gelesen und ab D0 gespeichert. M8091 wird vor dem Lesen des Kommentars zurückgesetzt. Dadurch wird nach dem Kommentar das Zeichen „NUL“ (00H) angefügt.

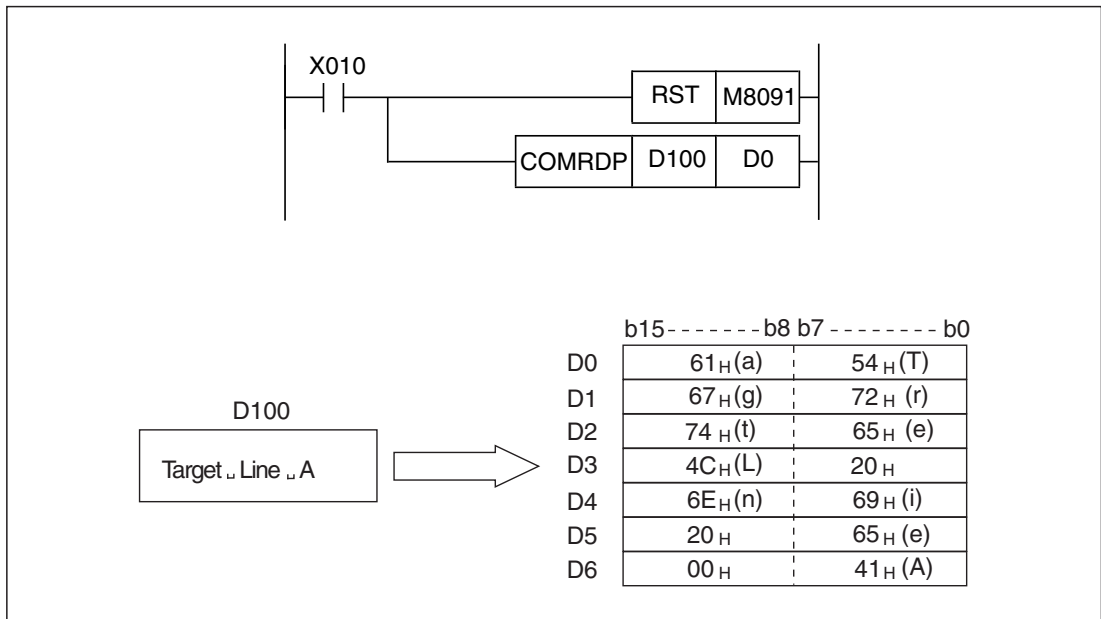


Abb. 7-181: Der Operandenkommentar zu D100 wird als ASCII-Zeichen in D0 bis D6 eingetragen.



7.14.2 Zufallszahl generieren (RND)

		RND		FNC 183			
		Erzeugen einer Zufallszahl					
Operanden	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
			16 Bit	32 Bit	RND RNDP	3	
	KnX, KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□)	●	●				

Funktion

Erzeugen einer Zufallszahl im Bereich von 0 bis 32767

Beschreibung

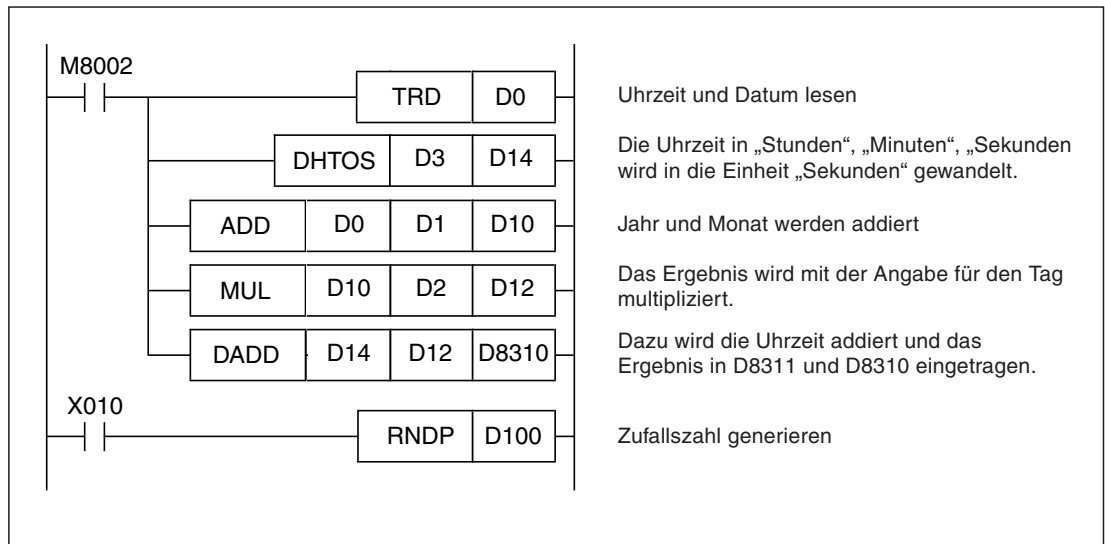
- Mit einer RND-Anweisung wird eine Zufallszahl erzeugt und in dem in (D+) angegebenen Operanden gespeichert.
- Als Quelle zur Berechnung der Zufallszahl werden die Sonderregister D8311 und D8310 verwendet. Nach dem Einschalten der SPS wird in diese Register der Wert „1“ eingetragen. Nach dem Übergang in die Betriebsart „RUN“ kann in D8311 und D8310 ein beliebiger Wert zwischen 0 und 2.147.483.647 eingetragen werden. Dieser Transfer muss nur einmal nach dem Anlauf der Steuerung ausgeführt werden.

Beispiel ▾

Im folgenden Beispielprogramm wird bei jedem Einschalten von X10 eine Zufallszahl erzeugt und in D100 gespeichert.

Nach dem Übergang von „STOP“ nach „RUN“ wird die Uhrzeit und das Datum der SPS-internen Uhr gelesen, gewandelt und nach der Formel „{((Jahr + Monat) x Tag) + Uhrzeit}“ die Grundlage zur Berechnung der Zufallszahl gebildet.

M8002 ist nur im ersten Zyklus nach dem Einschalten der SPS gesetzt.

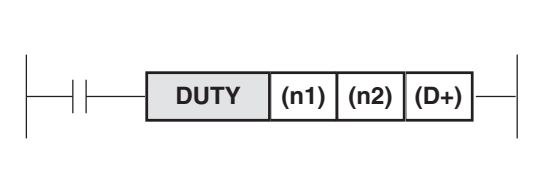


Uhrzeit und Datum lesen
 Die Uhrzeit in „Stunden“, „Minuten“, „Sekunden“ wird in die Einheit „Sekunden“ gewandelt.
 Jahr und Monat werden addiert
 Das Ergebnis wird mit der Angabe für den Tag multipliziert.
 Dazu wird die Uhrzeit addiert und das Ergebnis in D8311 und D8310 eingetragen.
 Zufallszahl generieren

Abb. 7-182: In diesem Beispiel wird die Quelle der Zufallszahl aus der Uhrzeit und dem Datum gebildet.



7.14.3 Taktgenerator mit einstellbarem Tastverhältnis (DUTY)

				DUTY		FNC 186				
				Impulsausgabe						
Operanden	n1	n2	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	T, C, D, R, K, H				M*	16 Bit	32 Bit	DUTY		7
					●				●	

* Es kann nur ein Merker aus dem Bereich M8330 bis M8334 angegeben werden.

Funktion

Sondermerker zyklisch für eine bestimmte Anzahl Programmzyklen setzen und zurücksetzen.

Beschreibung

- Die DUTY-Anweisung schaltet den in (D+) angegebenen Operanden für die in (n1) angegebene Anzahl von Programmzyklen ein und für die in (n2) angegebene Anzahl von Programmzyklen aus.
- In (D+) kann nur ein Merker aus dem Bereich von M8330 bis M8334 angegeben werden.
- Die Anzahl der Programmzyklen für M8330 bis M8334 wird in den Sonderregistern D8330 bis D8334 gezählt. Ein Zähler wird zurückgesetzt, wenn der Wert den Wert „(n1) + (n2)“ erreicht oder wenn die Eingangsbedingung für die DUTY-Anweisung eingeschaltet wird.
- Die Ausführung der DUTY-Anweisung beginnt, wenn die Eingangsbedingung eingeschaltet wird. Der Operand in (D+) wird am Ende des Programmzyklus bei der Bearbeitung der END-Anweisung gesetzt oder zurückgesetzt. Auch wenn die Eingangsbedingung unwahr wird, wird die Ausführung der DUTY-Anweisung nicht beendet. Erst wenn die Steuerung gestoppt oder die Versorgungsspannung ausgeschaltet wird, stoppt auch die Ausführung dieser Anweisung.
- Ist der Wert in (n1) = 0, bleibt der Operand in (D+) ständig ausgeschaltet.
- Ist der Wert in (n2) = 0, wird der Operand in (D+) ständig eingeschaltet.

HINWEISE

Da nur 5 Sondermerker als Operanden einer DUTY-Anweisung zur Verfügung stehen, können in einem Programm maximal 5 DUTY-Anweisungen verwendet werden.

Ein Sondermerker, der in einer DUTY-Anweisung verwendet wird, darf nicht noch einmal in einer anderen DUTY-Anweisung als Operand angegeben werden.

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird ein Fehlercode eingetragen:

- Für (n1) oder (n2) wurde ein negativer Wert angegeben (Fehlercode „6706“).
- Für (D+) wurde ein anderer Operand als ein Merker aus dem Bereich von M8330 bis M8334 angegeben (Fehlercode „6705“).

Beispiel ▾

M8330 im folgenden Beispielprogramm wird für die Dauer eines Programmzyklus eingeschaltet und anschließend für 3 Programmzyklen ausgeschaltet.

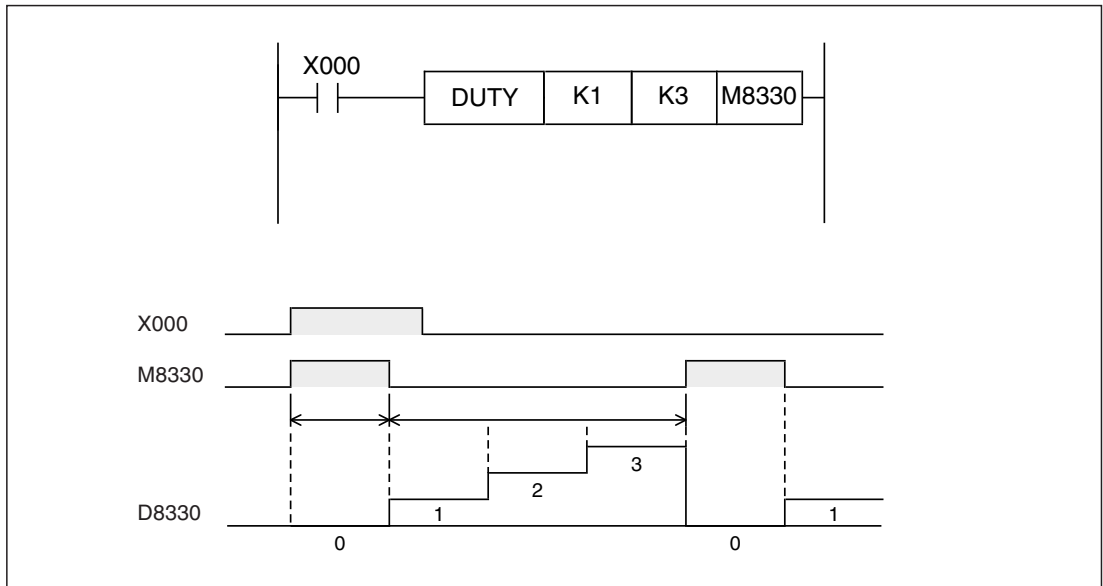


Abb. 7-183: Beispiel zur Anwendung einer DUTY-Anweisung



7.14.4 Daten prüfen (CRC)

				CRC		FNC 188				
				Cyclic Redundancy Check ausführen						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX2N FX2NC
										●
Operanden	S+	D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	KnX*, KnY*, KnM*, KnS*, T, C, D, R, Sondermodule (U□/G□)	KnY*, KnM*, KnS*, T, C, D, R, Sondermodule (U□/G□)	D, R, K, H		●	16 Bit	32 Bit	CRC CRCP	7	

* Bei der Angabe von Bit-Operanden muss der K-Faktor „4“ verwendet werden (z. B. K4M0).

Funktion

CRC-Wert für die Prüfung von Daten berechnen

Beschreibung

- Die CRC-Anweisung berechnet den Prüfwert für den *Cyclic Redundancy Check*, mit dem Fehler bei der Datenübertragung erkannt werden können. Die zu prüfenden 8-Bit-Daten sind in einem Operandenbereich gespeichert, dessen erste Adresse in (S+) angegeben wird. Das Ergebnis der Prüfung wird in dem Operanden abgelegt, der in (D+) angegeben ist. (n) gibt die Anzahl der zu prüfenden 8-Bit-Daten an (1 bis 256).
- Für die Prüfung wird die Formel für CRC-16 verwendet:

$$\text{CRC-Wert} = X^{16} + X^{15} + X^2 + 1$$
- Die CRC-Anweisung kann einem der Operanden in (S+) eine oder zwei 8-Bit-Gruppen zur Prüfung entnehmen. Gesteuert wird dies durch den Sondermerker M8161.

Funktion bei zurückgesetztem Sondermerker M8161 (16-Bit-Modus)

Im 16-Bit-Modus werden die zu prüfenden 8-Bit-Gruppen abwechselnd den niederwertigen und den höherwertigen Bytes der Operanden in (S+) entnommen. Wenn z. B. in (S+) D100 angegeben wird und M8161 „0“ ist werden die Daten dem niederwertigen Byte von D100, dann dem höherwertigen Byte von D100, dem niederwertigen Byte von D101, dann dem höherwertigen Byte von D101 usw. entnommen.

Das Ergebnis wird nur in einem 16-Bit-Operanden abgelegt (D+).

Funktion bei gesetztem Sondermerker M8161 (8-Bit-Modus)

Die zu prüfenden 8-Bit-Gruppen werden im 8-Bit-Modus nur den niederwertigen Bytes der mit (S+) angegebenen Operanden entnommen.

Das Ergebnis wird in zwei 16-Bit-Operanden abgelegt. Das niederwertige Byte von (D+) enthält das niederwertige Byte und das niederwertige Byte von ((D+)+1) enthält das höherwertige Byte des Ergebnisses

HINWEISE

Der Sondermerker M8161 beeinflusst auch das Verhalten einer RS-, ASCII-, HEX- und ASC-Anweisung. Wenn eine dieser Anweisungen im selben Programm wie die CRC-Anweisung verwendet wird, sollte M8161 vor der Ausführung der CRC-Anweisung gesetzt und unmittelbar nach Ausführung der ASC-Anweisung wieder zurückgesetzt werden.

Die CRC-Anweisung berechnet den CRC-Wert nach dem Polynom für CRC-16. Daneben existieren noch andere Standardprüfverfahren, wie zum Beispiel.

CRC-12: $X^{12} + X^{11} + X^3 + X^2 + X + 1$

CRC-32: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

CRC-CCITT: $X^{16} + X^{12} + X^5 + 1$

Bei diesen Prüfverfahren ergeben sich vollkommen andere Werte als bei der CRC-16-Prüfung. Stellen Sie daher sicher, dass für Ihre Anwendung das CRC-16-Polynom verwendet werden kann.

Zusätzlich zum Cyclic Redundancy Check werden zur Fehlererkennung bei der Kommunikation die Paritäts- und die Summenprüfung eingesetzt. In einer MELSEC FX-SPS kann hierfür die CCD-Anweisung (Abschnitt 7.4.5) verwendet werden.

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Bei der Angabe von Bit-Operanden in (S+) oder (D+) wurde nicht der K-Faktor 4 verwendet.
- Für (n) wurde ein Wert angegeben, der außerhalb des zulässigen Bereichs von 1 bis 256 sind.
- Der Operand ((S+)+(n-1)) oder der Operand ((D+)+1) überschreiten den zulässigen Bereich für den angegebenen Operandentyp.

Beispiele ▾

In den folgenden Programmbeispielen wird aus den Zeichen „0123456“, die im ASCII-Code ab dem Datenregister D100 gespeichert sind, der CRC-Wert berechnet. Das Ergebnis wird ab D0 gespeichert.

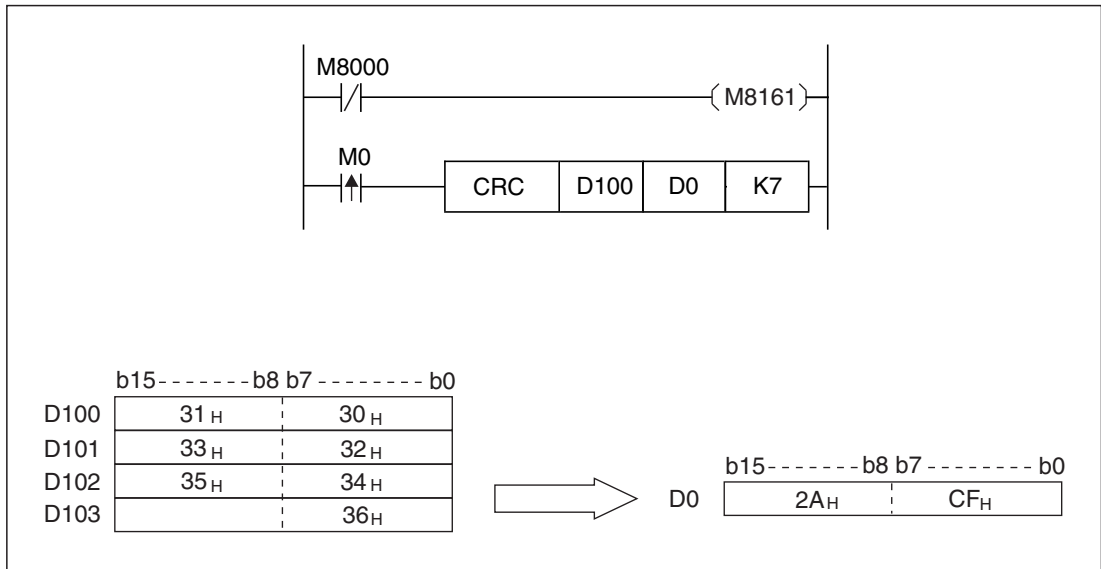


Abb. 7-184: IVerarbeitung der Daten im 16-Bit-Modus (M8161 ist nicht gesetzt)

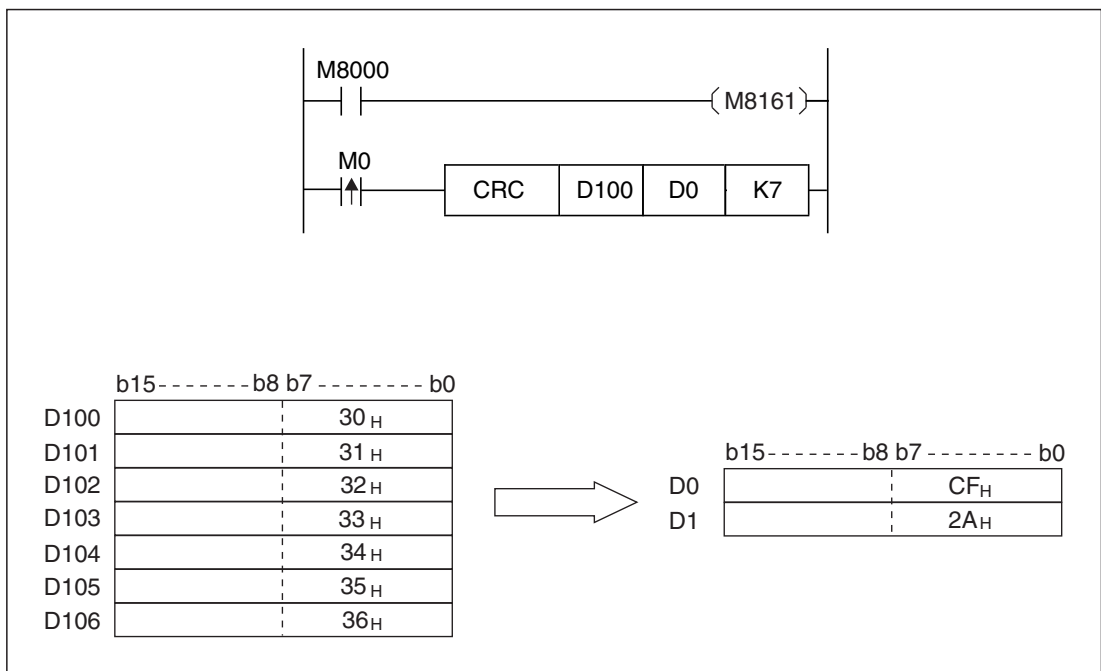


Abb. 7-185: Verarbeitung der Daten im 8-Bit-Modus (M8161 ist gesetzt)



7.14.5 Istwert eines High-Speed-Counters transferieren (DHCMOV)

				DHCMOV		FNC 189				
				High-Speed-Counter-Istwert transferieren						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
										●
Operanden	S+	D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	C*, D*	D, R	K, H		16 Bit	32 Bit	DHCMOV		13	
						●				

* Es können nur die High-Speed-Counter C235 bis C255 oder die Ringzähler D8099 und D8398 angegeben werden.

Funktion

Transfer des Istwerts eines High-Speed-Counters oder Ringzählers

Beschreibung

- Der Istwert des in (S+) angegebenen High-Speed-Counters oder Ringzählers wird in den mit (D+) angegebenen Operanden transferiert.
- Für (S+) können nur die High-Speed-Counter C235 bis C255 sowie die Ringzähler D8099 und D8398 angegeben werden.
- Die Istwerte der Counter C235 bis C255 werden in ((D+)+1) und (D+) eingetragen.
- Der Istwert des 16-Bit-Ringzählers D8099 wird in (D+) eingetragen.
- Wird für (D+) der 32-Bit-Ringzählers D8398 angegeben, enthält ((D+)+1) den Inhalt von D8399 und (D+) den Inhalt von D8398.
- Mit (n) kann angegeben werden, ob der Counter-Istwert nach dem Transfer gelöscht werden soll:
 - (n) = 0: Istwert nicht löschen
 - (n) = 1: Istwert des Counters nach dem Lesen löschen

Anwendung der DHCMOV-Anweisung

Ein High-Speed-Counter (C235 bis C255) zählt bei einem Signal an einem Eingang auf- oder abwärts. Der Istwert wird bei einem Hardware-Counter aktualisiert, wenn eine OUT-Anweisung mit dem Counter ausgeführt wird und ist dadurch abhängig vom Programmzyklus. Der Istwert eines Software-Counters wird bei jedem Eingangsimpuls aktualisiert.

Wird der Istwert mit einer Transferanweisung, wie z. B. einer MOV-Anweisung, gelesen, wird eventuell nicht der aktuelle Istwert erfasst. Durch eine DHCMOV-Anweisung in Verbindung mit einem durch den Zähleringang ausgelösten Interrupt kann der aktuelle Counter-Istwert mit der steigenden oder fallenden Flanke des Eingangssignals gelesen werden.

Wird eine DHCMOV-Anweisung unmittelbar vor einer Vergleichsanweisung (CMP, ZCP oder UND/ODER-verknüpte Vergleichsanweisungen) programmiert, wird beim Vergleich der aktuelle Istwert verwendet. Der Vergleich mit einer CMP- oder ZCP-Anweisung oder UND/ODER-verknüpften Vergleichsanweisungen hat gegenüber Vergleichsanweisungen für High-Speed-Counter den Vorteil, dass ein Hardware-Counter nicht wie ein Software-Counter behandelt wird. Hardware-Counter können höhere Frequenzen erfassen als Software-Counter. Soll mit einem Istwert eines High-Speed-Counters ein Vergleich ausgeführt und ein Ausgang angesteuert werden, sobald sich der Istwert ändert, müssen Vergleichsanweisungen für High-Speed-Counter verwendet werden (HSCS, HSCR, HSZ).

Eine DHCMOV-Anweisung kann beliebig oft im Programm verwendet werden.

Bitte beachten Sie bei der Ausführung einer DHCMOV-Anweisung in einem Interrupt-Programm die folgenden Hinweise:

- Im Hauptprogramm müssen Interrupts mit einer EI-Anweisung (Abschnitt 6.2.4) freigegeben werden. Das Hauptprogramm muss – außer im IEC-Editor der Software GX IEC Developer – mit einer FEND-Anweisung abgeschlossen werden (Abschnitt 6.2.5).
- Wenn eine DHCMOV-Anweisung in der ersten Zeile eines Interrupt-Programms programmiert wird, muss der Sondermerker M8394 verwendet werden:

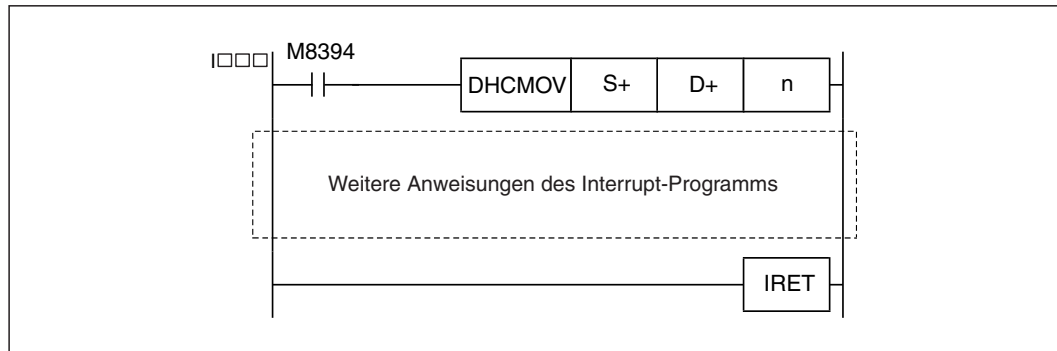


Abb. 7-186: In einem Interrupt-Programm wird eine DHCMOV-Anweisung durch den Sondermerker M8394 gesteuert.

- Falls in einem Interrupt-Programm mehrere DHCMOV-Anweisungen verwendet werden, wird nur die erste DHCMOV-Anweisung, die direkt auf den Interrupt-Pointer folgt, durch den Sondermerker M8394 gesteuert.

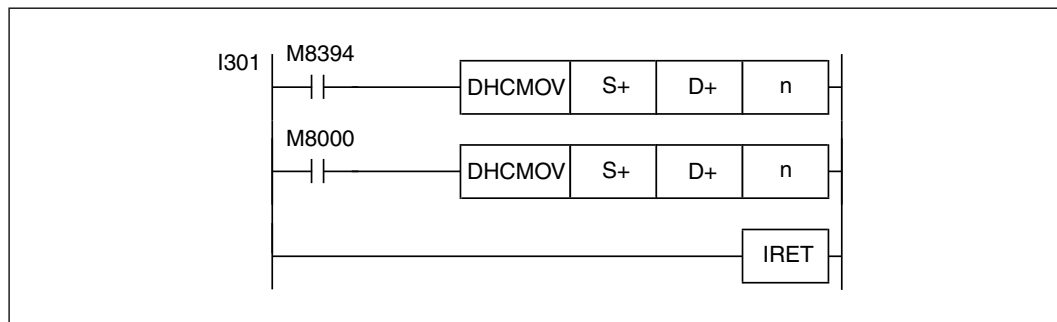


Abb. 7-187: In diesem Beispiel wird wegen des Interrupt-Pointers I301 beim Einschalten des Eingangs X3 zuerst die erste DHCMOV-Anweisung ausgeführt. Anschließend wird die zweite DHCMOV-Anweisung bearbeitet.

- Derselbe Counter darf nicht in mehreren Interrupt-Programmen für DHCMOV-Anweisungen angegeben werden.

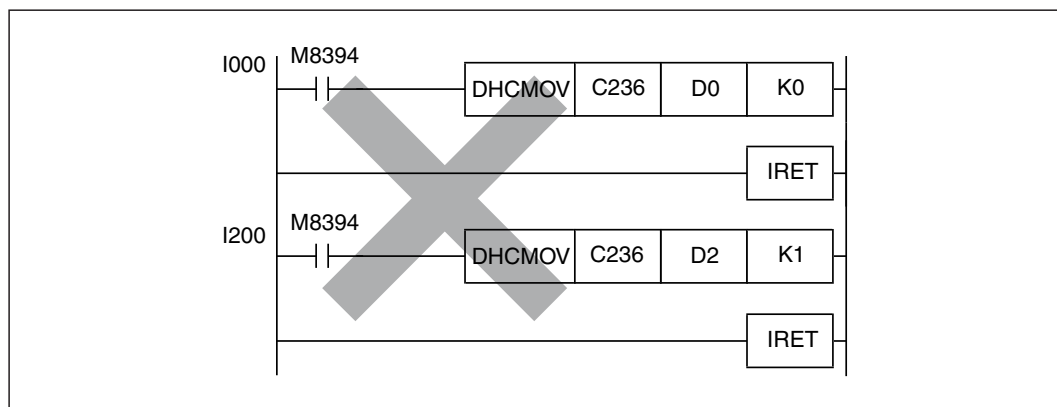


Abb. 7-188: In diesem Beispiel wird mit DHCMOV-Anweisungen zweimal auf den selben Counter zugegriffen. Das ist nicht zulässig.

- Interrupts, die durch Eingänge ausgelöst werden, können durch Sondermerker gesperrt werden. Dadurch werden auch die entsprechenden Interrupt-Programme und die dort programmierten Anweisungen nicht ausgeführt.

Sondermerker	Bedeutung	Eingang
M8050	Interrupt-Programme I000 und I001 sperren	X000
M8051	Interrupt-Programme I100 und I101 sperren	X001
M8052	Interrupt-Programme I200 und I201 sperren	X002
M8053	Interrupt-Programme I300 und I301 sperren	X003
M8054	Interrupt-Programme I400 und I401 sperren	X004
M8055	Interrupt-Programme I500 und I501 sperren	X005

Tab. 7-59: Mit den Sondermerkern M8050 bis M8055 können Interrupt-Programme gesperrt werden.

Zum Sperren eines Interrupt-Programms muss der entsprechende Sondermerker gesetzt werden. Die Sondermerker M8050 bis M8055 werden zurückgesetzt, wenn die SPS in die Betriebsart STOP gebracht wird.

- Falls ein Interrupt ausgelöst wird und die Ausführung des entsprechenden Interrupt-Programms ist durch eine andere Ursache gesperrt als einen gesetzten Sondermerker M8050 bis M8055, wird nur die DHCMOV-Anweisung am Anfang des Interrupt-Programms ausgeführt, aber nicht die weiteren Anweisungen. Dies ist zum Beispiel der Fall, wenn im Programm eine DI-Anweisung ausgeführt wurde. Erst nach Ausführung einer EI-Anweisung (Abschnitt 6.2.4) sind Interrupts wieder freigegeben und Interrupt-Programme werden komplett abgearbeitet.

Fehlerquellen

Im folgenden Fall tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6705“ eingetragen:

- Der Operand in (S+) oder (D+) überschreitet den zulässigen Bereich für den angegebenen Operandentyp.

Beispiel ▾

Im folgenden Programmbeispiel wird der Istwert des High-Speed-Counters C235 in jedem Programmzyklus mit einem Vorgabewert verglichen. Der Ausgang Y000 wird gesetzt, wenn der Istwert größer oder gleich „500“ ist. Der Istwert von C235 wird nach dem Lesen nicht gelöscht ((n) = „0“).

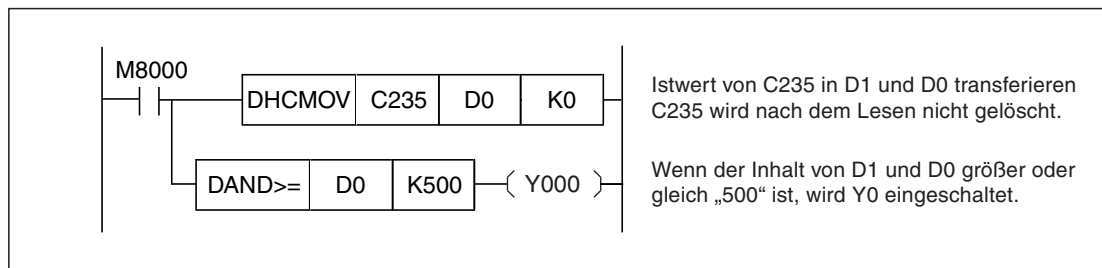


Abb. 7-189: Beispiel für die Ausführung einer DHCMOV-Anweisung im zyklischen Programm



Beispiel ▾

In diesem Beispiel für ein Interrupt-Programm wird der Istwert des High-Speed-Counters C235 in die Datenregister D201 und D200 übertragen und anschließend der Istwert von C235 gelöscht ((n) = „1“).

Das Interrupt-Programm wird ausgeführt, wenn der Eingang X001 eingeschaltet wird (Interrupt-Pointer I101).

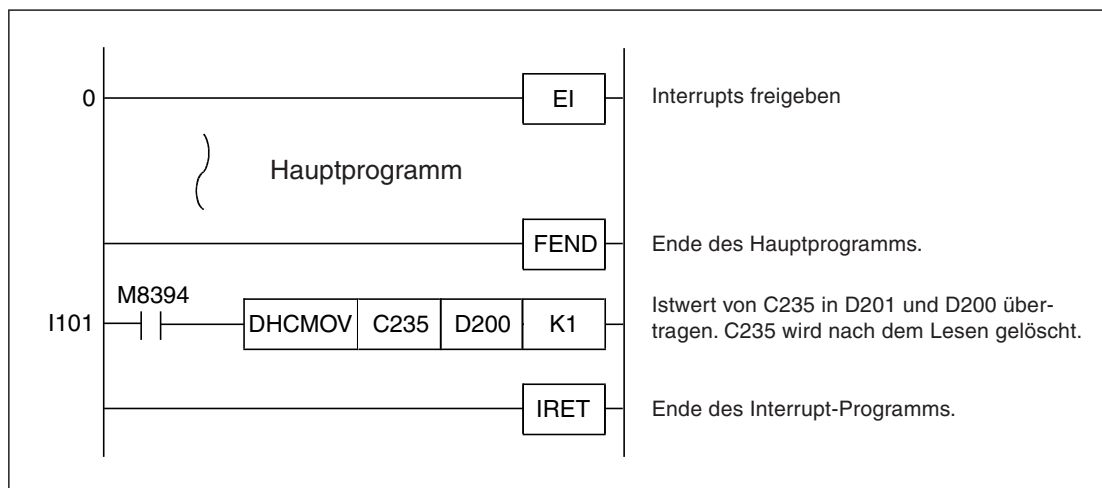


Abb. 7-190: Beispiel für die Ausführung einer DHCMOV-Anweisung in einem Interrupt-Programm



7.15 Anweisungen für Datenblöcke

Symbol	FNC	Bedeutung	Abschnitt
BK+	192	Daten in zwei Datenblöcken addieren	7.15.1
BK-	193	Daten in zwei Datenblöcken subtrahieren	7.15.2
BKCMP=	194	Vergleich der Daten in Datenblöcke auf „gleich“	7.15.3
BKCMP>	195	Vergleich der Daten in Datenblöcke auf „größer“	
BKCMP<	196	Vergleich der Daten in Datenblöcke auf „kleiner“	
BKCMP<>	197	Vergleich der Daten in Datenblöcke auf „ungleich“	
BKCMP<=	198	Vergleich der Daten in Datenblöcke auf „kleiner/gleich“	
BKCMPA>=	199	Vergleich der Daten in Datenblöcke auf „größer/gleich“	

Tab. 7-60: Übersicht der Anweisungen für Datenblöcke

7.15.1 Daten in zwei Datenblöcken addieren (BK+)

					BK+		FNC 192				
					Daten in zwei Datenblöcken addieren						
					CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
											●
Operanden	S1+	S2+	D+	n	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	T, C, D, R	T, C, D, R, K, H	T, C, D, R	D, R, K, H	●		16 Bit	32 Bit	BK+ BK+P	9	
							●	●	DBK+ DBK+P	17	

Funktion

Daten aus zwei zusammenhängenden Bereichen (Datenblöcke) werden addiert und das Ergebnis in einem weiteren Datenblock gespeichert.

Beschreibung

- In (S1+) wird die erste Adresse des 1. Quelloperandenbereichs angegeben.
- In (S2+) wird die erste Adresse des 2. Quelloperandenbereichs oder eine Konstante angegeben.
- Das Ergebnis der Addition wird in dem Bereich eingetragen, dessen 1. Adresse in (D+) angegeben wurde.
- Die Größe der Bereiche (S1+), (S2+) und (D+) wird mit (n) angegeben.
- Diese Anweisung bildet nicht die Summe eines Datenblocks, sondern addiert jeweils zwei Operanden oder den Inhalt eines Operanden und eine Konstante. Bei der 16-Bit-Verarbeitung kann die Konstante im Bereich von -32768 bis 32767 und bei der 32-Bit-Verarbeitung im Bereich von -2.147.483.648 bis 2.147.483.647 liegen.

16-Bit-Verarbeitung

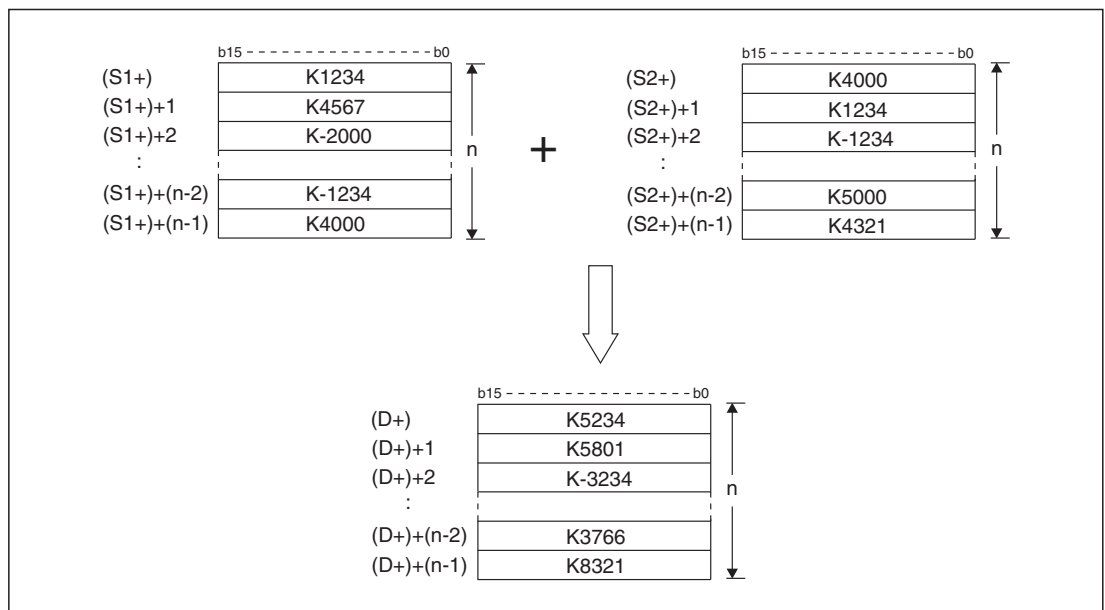


Abb. 7-191: Addition der Inhalte von zwei Datenblöcken

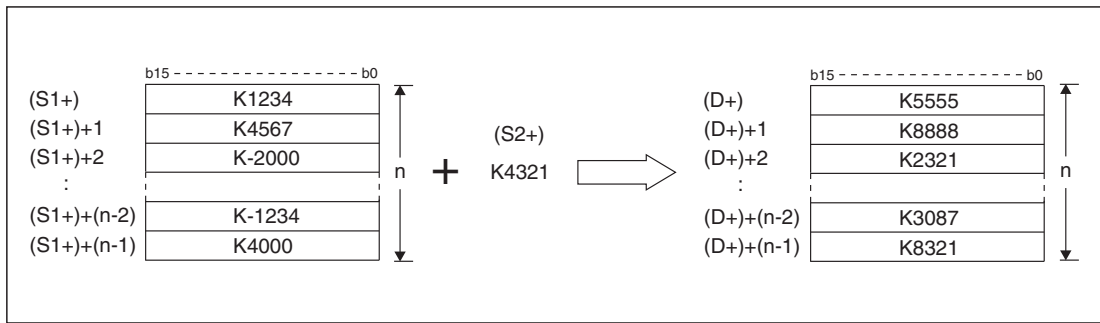


Abb. 7-192: Addition einer Konstanten zu den Inhalten eines Datenblocks

32-Bit-Verarbeitung

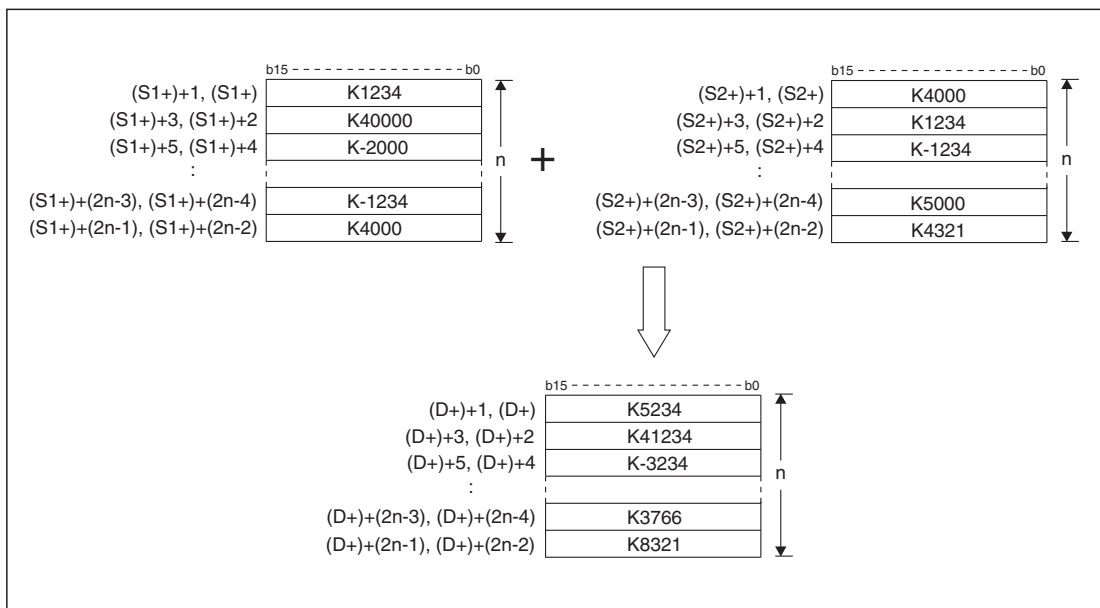


Abb. 7-193: Addition der Inhalte von zwei Datenblöcken, die aus 32-Bit-Operanden bestehen

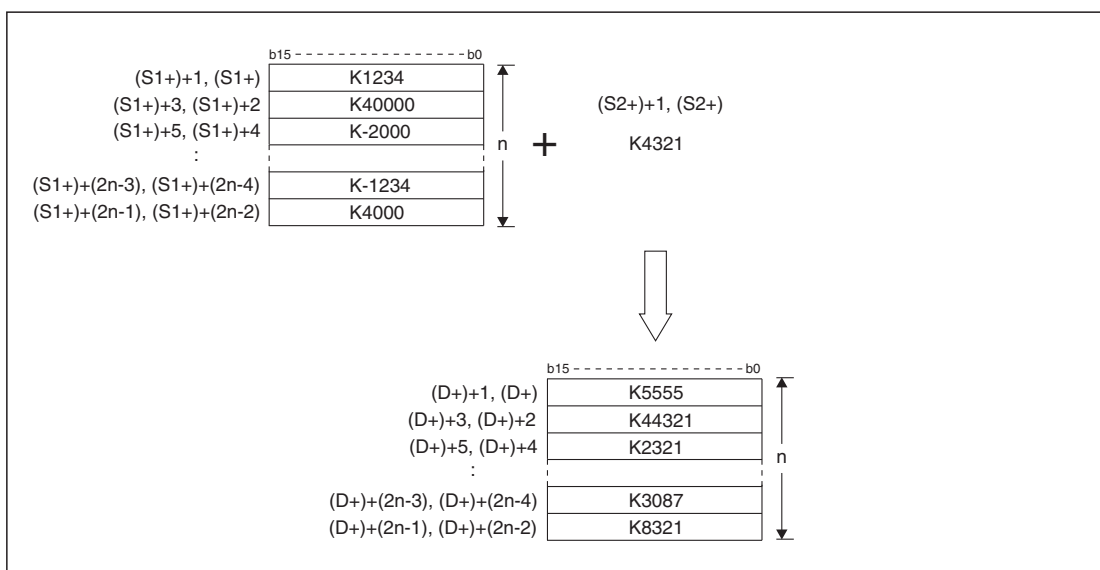


Abb. 7-194: Addition einer Konstanten zum Inhalt eines Datenblocks, der aus 32-Bit-Operanden besteht

HINWEISE

Wenn das Ergebnis der Addition den zulässigen Wertebereich unter- oder überschreitet, werden die folgenden Werte abgelegt. Das Carry-Flag wird in diesen Fällen nicht gesetzt.

16-Bit-Verarbeitung:

32767 (7FFFH) + 2 (0002H) -> -32767 (8001H)

-32767 (8000H) + (-2) (FFFEH) -> 32766 (7FFEH)

32-Bit-Verarbeitung:

2.147.483.647 (7FFFFFFFH) + 2 (00000002H) -> -2.147.483.647 (80000001H)

-2.147.483.648 (80000000H) + (-2) (FFFFFFFEH) -> 2.147.483.646 (7FFFFFFEH)

Wird bei einer 32-Bit-Anweisung (DBK+/DBK+P) für (n) ein Register verwendet (D oder R), werden die Operanden (n) und ((n) + 1) belegt. Zum Beispiel werden durch die Angabe von „DBK+ D0 D100 D200 R0“ für (n) die Register R1 und R0 belegt.

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Durch die Angabe von (n) (2n bei der 32-Bit-Verarbeitung) wird der zulässige Bereich für den in (S1+), (S2+) oder (D+) angegebenen Operandentyp überschritten.
- Die mit (S1+) oder (S2+) und (n) (2n bei der 32-Bit-Verarbeitung) angegebenen Bereiche überlappen sich mit den durch (D+) definierten Bereich.

Beispiel ▾

Durch das folgende Programm werden die Inhalte der Datenregister D100 bis (D100+n) zu den Inhalten der Datenregister D150 bis (D150+n) addiert und die Ergebnisse ab D200 gespeichert. Die Anzahl der Operanden eines Bereichs (n) wird in D0 angegeben.

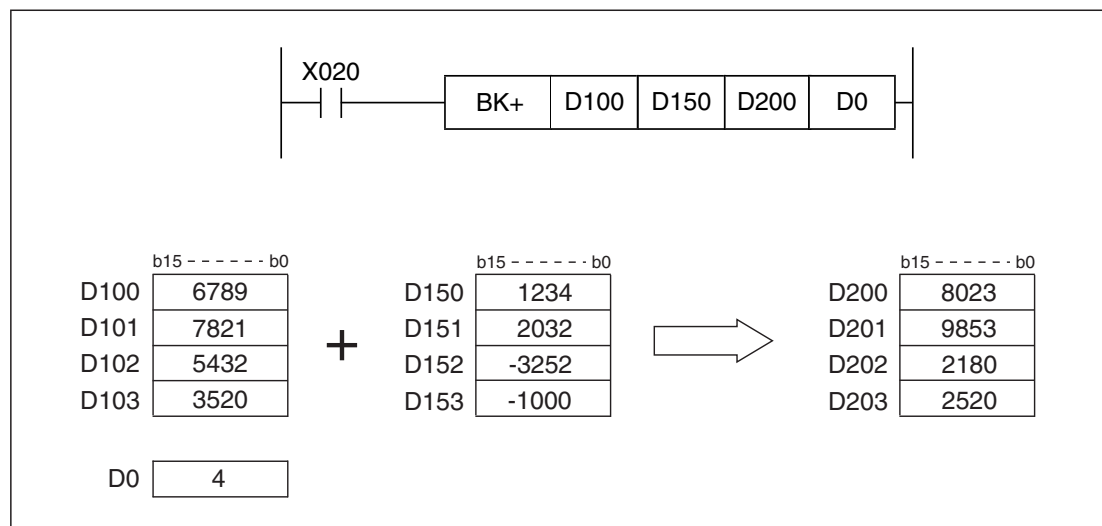


Abb. 7-195: Beispiel zum Addieren der Inhalte von Datenblöcken

△

7.15.2 Daten in zwei Datenblöcke subtrahieren (BK-)

					BK-		FNC 193				
					Daten in zwei Datenblöcken subtrahieren						
Operanden					CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
										●	
					Puls-Anweisung (P)		Verarbeitung		Programmschritte		
					●	16 Bit	32 Bit	BK- BK-P		9	
						●	●	DBK- DBK-P		17	
S1+		S2+	D+	n							
T, C, D, R		T, C, D, R, K, H	T, C, D, R	D, R, K, H							

Funktion

Daten aus zwei zusammenhängenden Bereichen (Datenblöcke) werden subtrahiert und das Ergebnis in einem weiteren Datenblock gespeichert.

Beschreibung

- In (S1+) wird die erste Adresse des Bereichs angegeben, der die Minuenden enthält. (Von diesen Werten wird abgezogen.)
- In (S2+) wird eine Konstante oder die erste Adresse des Bereichs angegeben, der die Subtrahenden enthält.
- Das Ergebnis der Subtraktion wird in dem Bereich eingetragen, dessen 1. Adresse in (D+) angegeben wurde.
- Die Größe der Bereiche (S1+), (S2+) und (D+) wird mit (n) angegeben.
- Falls als Subtrahend eine Konstante angegeben wird, kann diese bei der 16-Bit-Verarbeitung im Bereich von -32768 bis 32767 und bei der 32-Bit-Verarbeitung im Bereich von -2.147.483.648 bis 2.147.483.647 liegen.

16-Bit-Verarbeitung

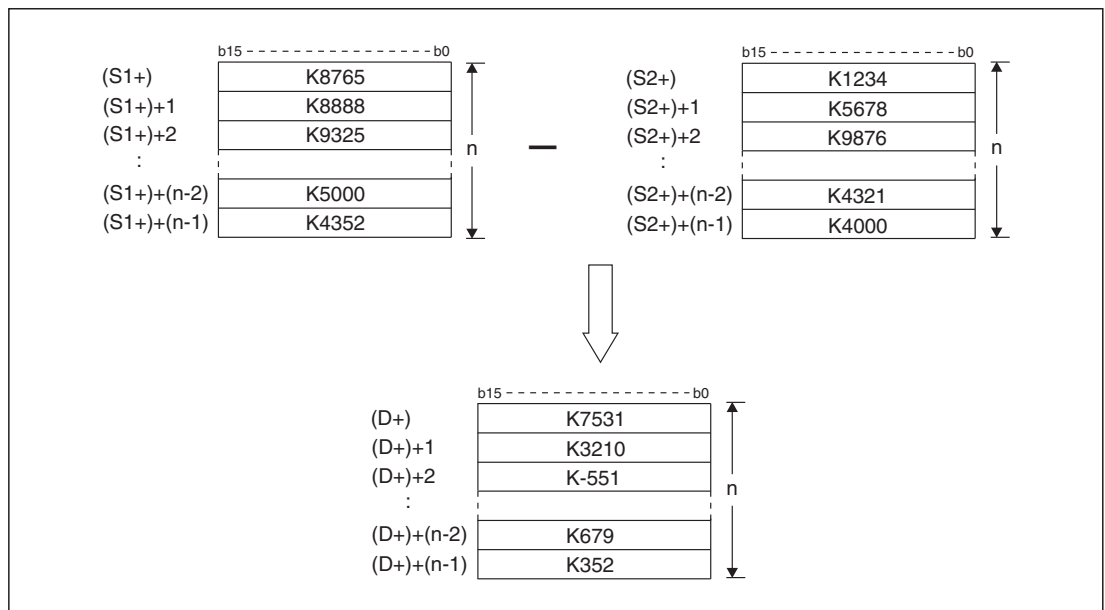


Abb. 7-196: Subtraktion der Inhalte von zwei Datenblöcken

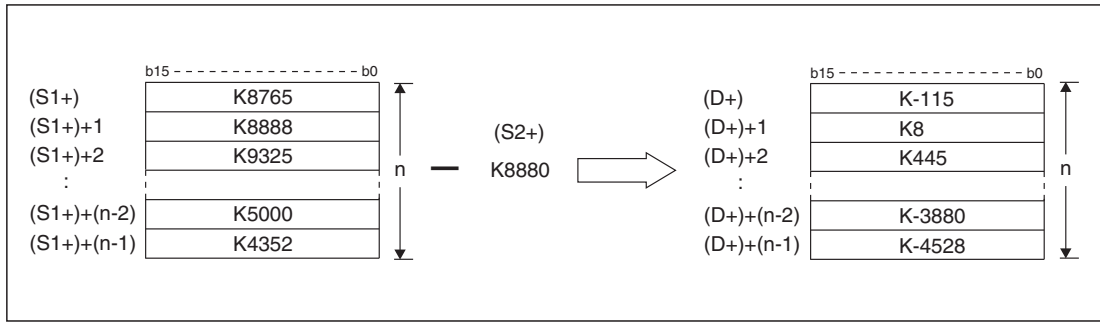


Abb. 7-197: Eine Konstante wird von den Inhalten eines Datenblocks abgezogen

32-Bit-Verarbeitung

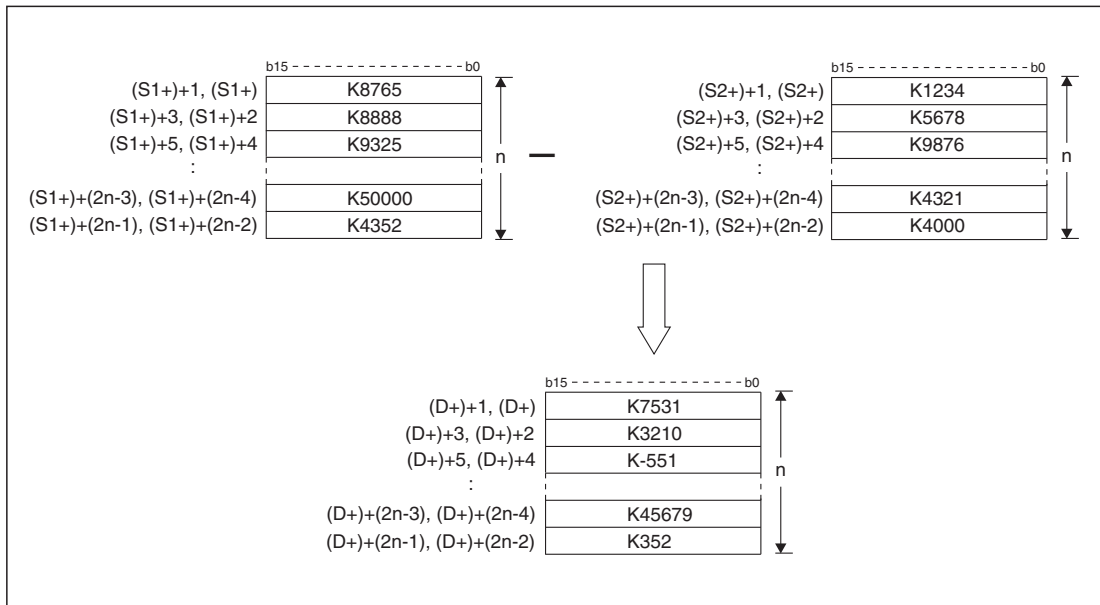


Abb. 7-198: Subtraktion der Inhalte von zwei Datenblöcken, die aus 32-Bit-Operanden bestehen

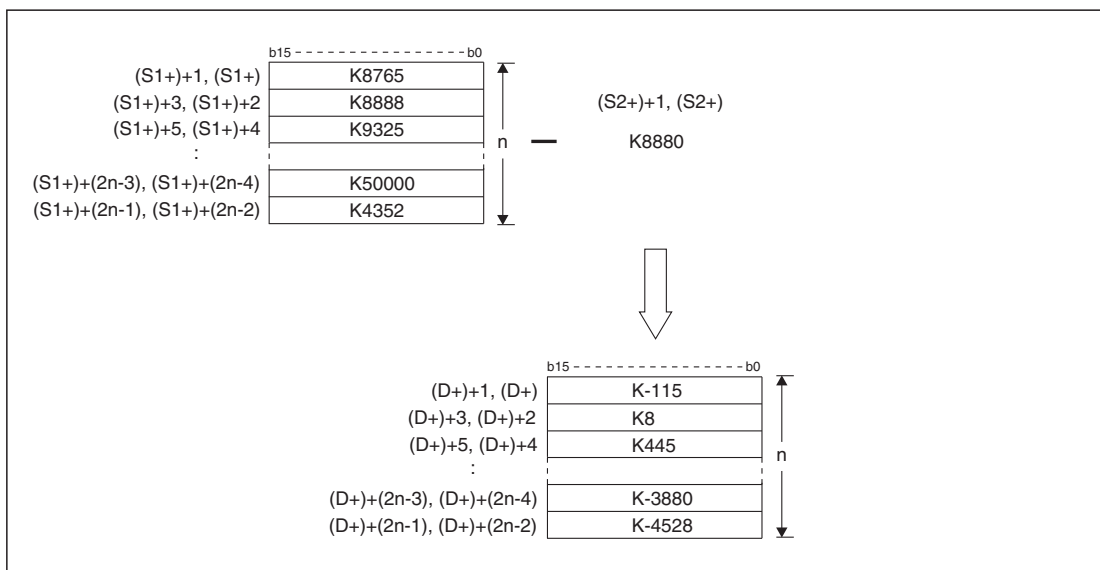


Abb. 7-199: Eine Konstante wird von den Inhalten eines Datenblocks abgezogen, der aus 32-Bit-Operanden besteht

HINWEISE

Wenn das Ergebnis der Subtraktion den zulässigen Wertebereich unter- oder überschreitet, werden die folgenden Werte abgelegt. Das Carry-Flag wird in diesen Fällen nicht gesetzt.

16-Bit-Verarbeitung:

-32767 (8000H) - 2 (0002H) -> 32766 (7FFE_H)
 32767 (7FFF_H) - (-2) (FFFE_H) -> -32767 (8001_H)

32-Bit-Verarbeitung:

-2.147.483.648 (80000000H) - 2 (00000002H) -> 2.147.483.646 (7FFFFFFE_H)
 2.147.483.647 (7FFFFFFFH) - (-2) (FFFFFFFE_H) -> -2.147.483.647 (80000001_H)

Wird bei einer 32-Bit-Anweisung (DBK-/DBK-P) für (n) ein Register verwendet (D oder R), werden die Operanden (n) und ((n) +1) belegt. Zum Beispiel werden durch die Angabe von „DBK- D0 D100 D200 R0“ für (n) die Register R1 und R0 belegt.

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Durch die Angabe von (n) (2n bei der 32-Bit-Verarbeitung) wird der zulässige Bereich für den in (S1+), (S2+) oder (D+) angegebenen Operandentyp überschritten.
- Die mit (S1+) oder (S2+) und (n) (2n bei der 32-Bit-Verarbeitung) angegebenen Bereiche überlappen sich mit den durch (D+) definierten Bereich.

Beispiel ▾

Mit dem folgenden Beispielprogramm wird der Wert „8765“ von den Inhalten der Datenregister D100 bis D102 subtrahiert, wenn der Eingang X10 eingeschaltet wird. Die Ergebnisse werden ab D200 gespeichert.

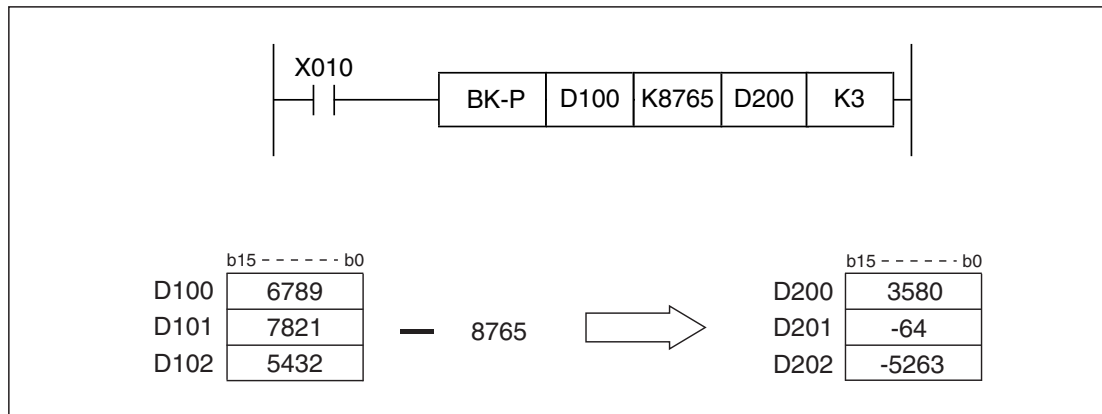


Abb. 7-200: Beispiel zum Subtrahieren der Inhalte von Datenblöcken



7.15.3 Daten in Datenblöcke vergleichen (BKCMP□)

					BKCMP□		FNC 194 – 199				
					Daten in einen Datenblock vergleichen						
					CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
											●
Operanden	S1+	S2+	D+	n	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	T, C, D, R, K, H	T, C, D, R	Y, M, S, D□.b	D, R, K, H	●		16 Bit	32 Bit	BKCMP□ BKCMP□P	9	
							●	●	DBKCMP□ DBKCMP□P	17	

Funktion

Daten in zwei zusammenhängenden Bereichen (Datenblöcke) werden miteinander verglichen und das Vergleichsergebnis in einem weiteren Datenblock gespeichert.

Beschreibung

- In (S1+) wird die erste Adresse des ersten Bereichs, in dem die vergleichenden Werte gespeichert sind, oder eine Konstante angegeben.
- In (S2+) wird die erste Adresse des Bereichs angegeben, der mit dem in (S1+) festgelegten Bereich verglichen wird.
- Das Ergebnis des Vergleichs wird in dem Bereich eingetragen, dessen erste Adresse in (D+) angegeben wurde. Ist der Vergleich wahr, wird als Ergebnis eine „1“ eingetragen. Ist der Vergleich unwahr, lautet das Vergleichsergebnis „0“.
- Die Größe der Bereiche (S1+), (S2+) und (D+) wird mit (n) angegeben.
- Das □ in der BKCMP□-Anweisung steht als Platzhalter für die Vergleichsoperationen =, >, <, <>, ≤ und ≥. Die folgende Tabelle zeigt die Zuordnung der Vergleichsoperationen zu den entsprechenden Funktionsnummern.

FNC	Anweisung	Wahr wenn	Falsch wenn
194	BKCMP =	(S1+) = (S2+)	(S1+) <> (S2+)
195	BKCMP >	(S1+) > (S2+)	(S1+) ≤ (S2+)
196	BKCMP <	(S1+) < (S2+)	(S1+) ≥ (S2+)
197	BKCMP <>	(S1+) <> (S2+)	(S1+) = (S2+)
198	BKCMP ≤	(S1+) ≤ (S2+)	(S1+) > (S2+)
199	BKCMP ≥	(S1+) ≥ (S2+)	(S1+) < (S2+)

Tab. 7-61: Vergleichsanweisungen für 16-Bit-Verarbeitung

FNC	Anweisung	Wahr wenn	Falsch wenn
194	DBKCMP =	((S1+)+1), (S1+) = ((S2+)+1), (S2+)	((S1+)+1), (S1+) <> ((S2+)+1), (S2+)
195	DBKCMP >	((S1+)+1), (S1+) > ((S2+)+1), (S2+)	((S1+)+1), (S1+) ≤ ((S2+)+1), (S2+)
196	DBKCMP <	((S1+)+1), (S1+) < ((S2+)+1), (S2+)	((S1+)+1), (S1+) ≥ ((S2+)+1), (S2+)
197	DBKCMP <>	((S1+)+1), (S1+) <> ((S2+)+1), (S2+)	((S1+)+1), (S1+) = ((S2+)+1), (S2+)
198	DBKCMP ≤	((S1+)+1), (S1+) ≤ ((S2+)+1), (S2+)	((S1+)+1), (S1+) > ((S2+)+1), (S2+)
199	DBKCMP ≥	((S1+)+1), (S1+) ≥ ((S2+)+1), (S2+)	((S1+)+1), (S1+) < ((S2+)+1), (S2+)

Tab. 7-62: Vergleichsanweisungen für 32-Bit-Verarbeitung

- Ergeben alle n Vergleiche das Ergebnis „Wahr“, wird zusätzlich der Sondermarker M8090 gesetzt.
- Eine Konstante kann bei der 16-Bit-Verarbeitung im Bereich von -32768 bis 32767 und bei der 32-Bit-Verarbeitung im Bereich von -2.147.483.648 bis 2.147.483.647 liegen.

HINWEISE

Zum Vergleich der Istwerte von 32-Bit-Zählern (C200 bis C255) müssen Anweisungen für 32-Bit-Verarbeitung verwendet werden (DBKCMP=, DBKCMP>, DBKCMP< etc.). Wird für diese Vergleiche die 16-Bit-Verarbeitung gewählt (BKCMP=, BKCMP> etc.), tritt ein Fehler mit dem Code 6705 auf.

Wird bei einer 32-Bit-Anweisung (DBKCMP□/DBKCMP□P) für (n) ein Register verwendet (D oder R), werden die Operanden (n) und ((n) +1) belegt. Zum Beispiel werden durch die Angabe von „DBKCMP= D0 D100 M0 R0“ für (n) die Register R1 und R0 belegt.

16-Bit-Verarbeitung

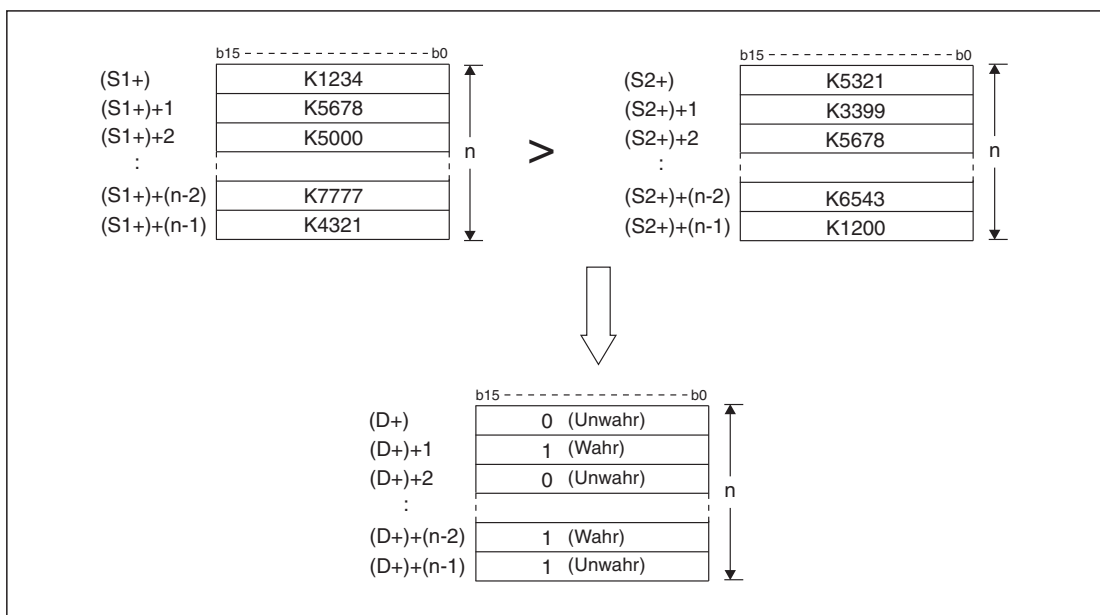


Abb. 7-201: In diesem Beispiel werden die Inhalte von zwei Datenblöcken mit einer BKCMP>-Anweisung auf „größer“ verglichen.

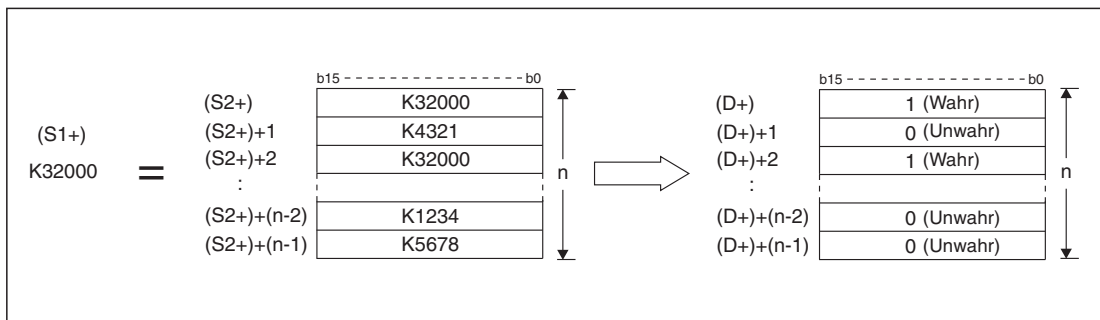


Abb. 7-202: Eine Konstante wird mit dem Inhalt eines Datenblocks auf „gleich“ verglichen (BKCMP=)

32-Bit-Verarbeitung

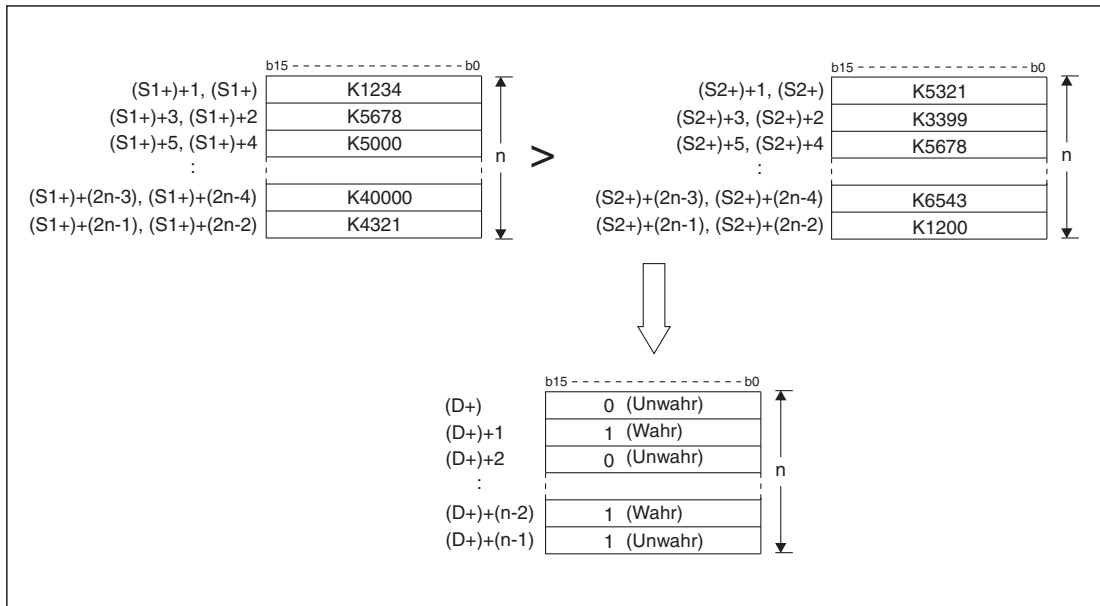


Abb. 7-203: Vergleich der Inhalte von zwei Datenblöcken auf „größer“ (DBKCMP>)

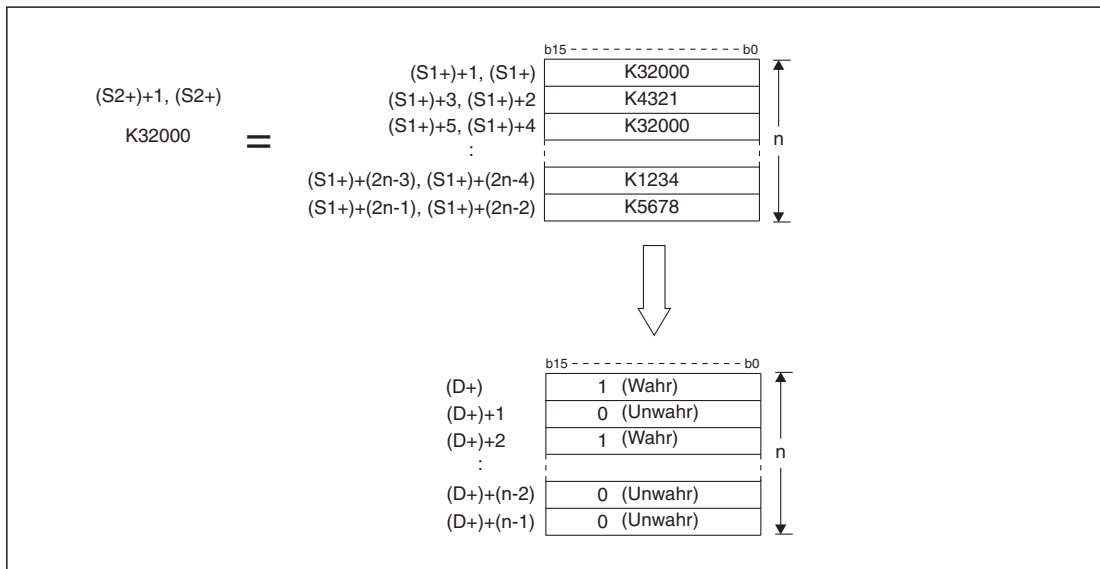


Abb. 7-204: Eine Konstante wird mit dem Inhalt eines Datenblocks auf „gleich“ verglichen (DBKCMP=)

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird ein Fehlercode eingetragen:

- Durch die Angabe von (n) (32-Bit-Verarbeitung: 2n) wird der zulässige Bereich für den in (S1+) oder (S2+) angegebenen Operandentyp überschritten. (Fehlercode „6706“)
- Durch die Angabe von (n) wird der zulässige Bereich für den in (D+) angegebenen Operandentyp überschritten. (Fehlercode „6706“)
- Für (D+) wurden Bits eines Registers angegeben (D□.n) und diese Operanden überlappen sich mit den durch (S1+) oder (S2+) definierten Bereichen. (Fehlercode „6706“)
- Für (S1+) und/oder (S2+) einer Anweisung für 16-Bit-Ausführung wurde ein 32-Bit-Zähler (C200 bis C255) angegeben. (Fehlercode „6705“)

Beispiel ▽

Vier Binärzahlen (16-Bit), die ab D100 gespeichert sind, werden im folgenden Programm mit vier Werten ab D200 auf Gleichheit verglichen, wenn der Eingang X20 eingeschaltet ist. Das Vergleichsergebnis wird ab M10 gespeichert. Falls alle Werte ab D100 und D200 übereinstimmen, wird durch den Sondermerker M8090 der Ausgang Y000 eingeschaltet.

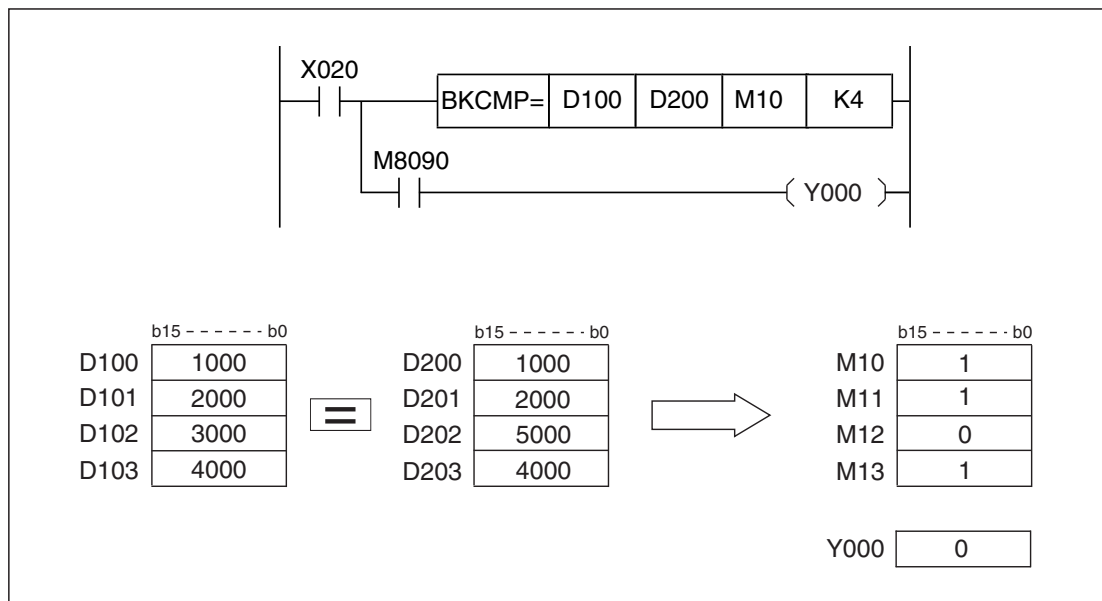


Abb. 7-205: Bei diesem Beispiel stimmen nicht alle Werte überein. Deshalb ist M8090 nicht gesetzt und der Ausgang bleibt ausgeschaltet.



Beispiel ▾

Wenn der Eingang X010 eingeschaltet ist, werden die Inhalte der 4 Datenregister D10 bis D13 mit dem Wert „1000“ auf „ungleich“ verglichen. Die Vergleichsergebnisse werden durch die Bits 4, 5, 6 und 7 von D0 angezeigt.

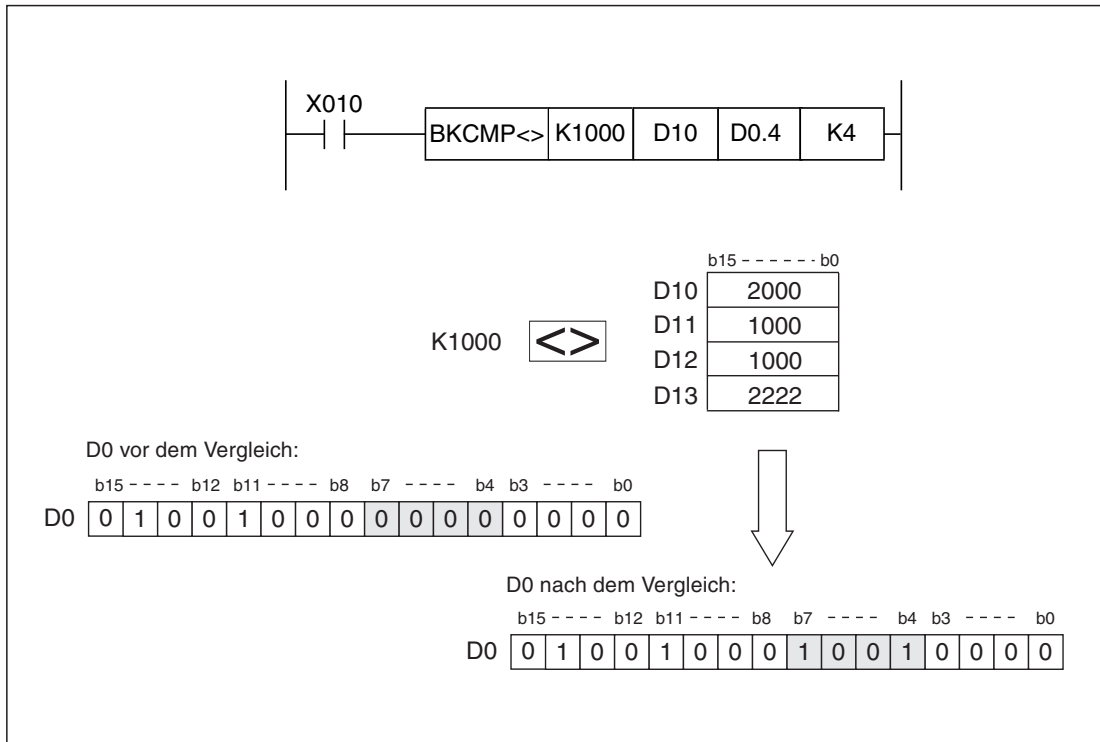


Abb. 7-206: Bei diesem Beispiel werden die Vergleichsergebnisse in einem Datenregister gespeichert. Die restlichen Bits von D0 werden durch den Vergleich nicht beeinflusst.



7.16 Verarbeitungsanweisungen für Zeichenfolgen

Zeichenfolgen bestehen aus mehreren zusammenhängenden Zeichen, die codiert (oft im ASCII-Code) von der SPS gehandhabt werden. Zum Beispiel könnte der Name „MITSUBISHI“ im ASCII-Code in der SPS gespeichert sein und an ein angeschlossenes Gerät übermittelt werden. Eine Übersicht des ASCII-Codes enthält der Anhang C.

Symbol	FNC	Bedeutung	Abschnitt
STR	200	Binärdaten in Zeichenfolgen wandeln	7.16.1
VAL	201	Zeichenfolgen in Binärdaten wandeln	7.16.2
\$+	202	Zeichenfolgen zusammenfügen	7.16.3
LEN	203	Länge von Zeichenfolgen ermitteln	7.16.4
RIGHT	204	Auszug der Zeichenfolgedaten von rechts	7.16.5
LEFT	205	Auszug der Zeichenfolgedaten von links	7.16.6
MIDR	206	Zeichenfolge auswählen	7.16.7
MIDW	207	Zeichenfolge ersetzen	7.16.8
INSTR	208	Zeichenfolge suchen	7.16.9
\$MOV	209	Zeichenfolge transferieren	7.16.10

Tab. 7-63:Übersicht der Anweisungen zur Verarbeitung von Zeichenfolgen

7.16.1 Binärdaten in Zeichenfolgen wandeln (STR)

				STR		FNC 200				
				Binärdaten in Zeichenfolgen wandeln						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
										●
Operanden	S1+	S2+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	T, C, D, R	KnX, KnY, KnM, KnS, T, C, D, R, U□/G□, V, Z, K, H	T, C, D, R	●	16 Bit	32 Bit	STR STRP	7		
					●	●	DSTR DSTRP	13		

Funktion

Umwandlung von 16-/32-Bit-Binärwerten in eine Zahl, die als Zeichenfolge im ASCII-Code mit einer bestimmten Anzahl von Zeichen dargestellt wird

Beschreibung

- Eine in (S2+) gespeicherter Binärwert oder eine Konstante wird entsprechend den Angaben in (S1+)+0 und (S1+)+1 in eine Zeichenfolge im ASCII-Code gewandelt und ab dem Operanden (D+) gespeichert.
- (S1+)+0 enthält die Anzahl der Zeichen, die das Wandlungsergebnis haben soll (einschließlich des Vorzeichens und des Dezimalpunkts). Bei 16-Bit-Verarbeitung können 2 bis 8 und bei 32-Bit-Verarbeitung können 2 bis 13 Zeichen angegeben werden.
- In (S1+)+1 wird die Anzahl der Nachkommastellen der Zeichenfolge eingetragen. (16-Bit-Verarbeitung: 0 bis 5, 32-Bit-Verarbeitung: 0 bis 10 Nachkommastellen)

Generell gilt: Anzahl der Nachkommastellen ≤ (Anzahl aller Stellen minus 3).

- Der binäre Wert in (S2+) kann bei der 16-Bit-Verarbeitung im Bereich von -32768 bis 32767 liegen. Bei der 32-Bit-Verarbeitung wird der zu wandelnde Wert in (S2+)+1 und (S2+)+0 übergeben und kann im Bereich von -2.147.483.648 bis 2.147.483.647 liegen.
- Im Bereich ab (D+) wird die gewandelte Zeichenfolge abgelegt:
 - Bei einer positiven Zahl wird als Vorzeichen „20H“ (Leerzeichen) im niederwertigen Byte von ((D+)+0) eingetragen. Eine negative Zahl wird an dieser Stelle durch ein Minuszeichen („2DH“) gekennzeichnet.
 - Nach der gewandelten Zeichenfolge wird automatisch das Zeichen „NUL“ (00H) eingefügt. Bei einer geraden Anzahl Zeichen wird „00H“ in den Operanden eingetragen, der auf dem Operanden mit dem letzten gewandelten Zeichen folgt. Bei einer ungeraden Anzahl Zeichen wird „00H“ in das höherwertige Byte des Operanden eingetragen, der im niederwertigen Byte das letzte Zeichen enthält.
 - Wenn die Anzahl der Nachkommastellen in (S1+)+1 auf einen anderen Wert als Null eingestellt wird, wird der ASCII-Code „2EH“ für den Dezimalpunkt automatisch an der angegebenen Stelle eingefügt. Wird die Anzahl der Nachkommastellen auf Null gesetzt, wird kein Dezimalpunkt eingefügt.

Wenn die Anzahl aller darzustellenden Stellen (einschließlich Dezimalpunkt und Vorzeichen) kleiner ist als die angegebene gesamte Anzahl der Stellen, werden die Stellen zwischen Vorzeichen und erster darzustellender Stelle mit dem ASCII-Code „20H“ (Leerzeichen) ausgefüllt.

Falls in ((S1+)+0) weniger Stellen angegeben sind, als der Binärwert in (S2+) bzw. (S2+)+1 und (S2+)+0 hat, tritt ein Fehler auf.

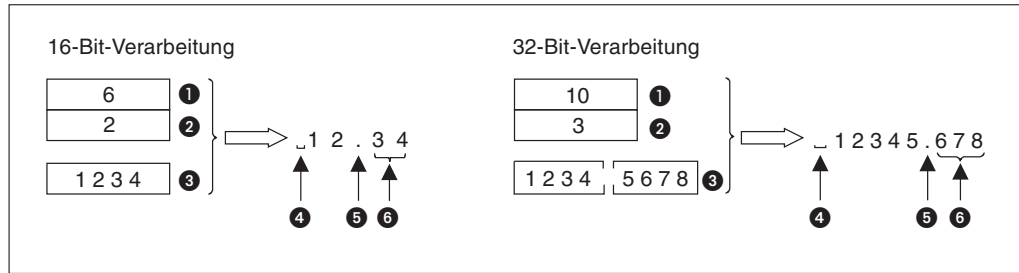


Abb. 7-207: Automatisches Einfügen des Dezimalpunkts und von Leerzeichen

- ①: Gesamte Anzahl der Zeichen in ((S1+)+0)
- ②: Anzahl der Nachkommastellen in ((S1+)+1)
- ③: Binärwert in (S2+) bzw. (S2+)+1) und (S2+)+0)
- ④: Automatisch eingefügtes Leerzeichen (20H).
- ⑤: Automatisch eingefügter Dezimalpunkt
- ⑥: Nachkommastellen

- Falls in ((S1+)+1) mehr Nachkommastellen angegeben werden, als der Binärwert in (S2+) bzw. (S2+)+1) und (S2+)+0) Stellen hat, wird der Wert nach rechts verschoben und es werden automatisch Nullen (30H) eingefügt.

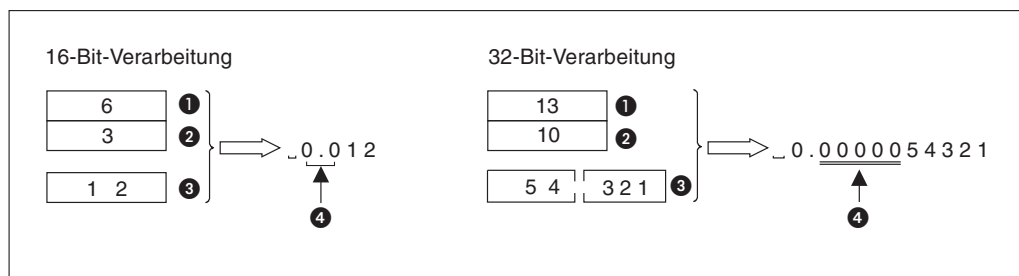


Abb. 7-208: Automatisches Einfügen von Nullen

- ①: Gesamte Anzahl der Zeichen in ((S1+)+0)
- ②: Anzahl der Nachkommastellen in ((S1+)+1)
- ③: Binärwert in (S2+) bzw. (S2+)+1) und (S2+)+0)
- ④: Automatisch eingefügte Nullen (30H).

16-Bit-Verarbeitung

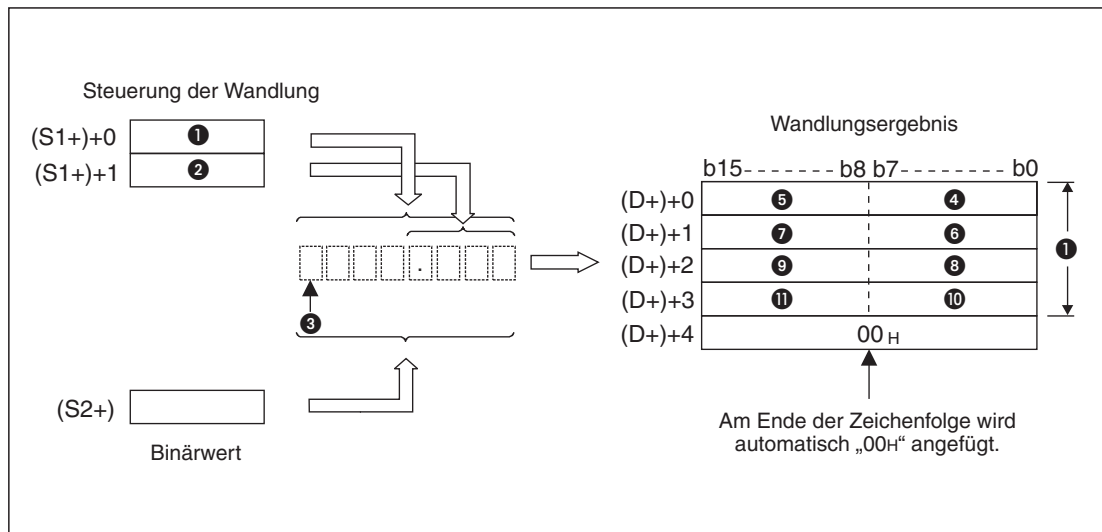


Abb. 7-209: Prinzip der Wandlung eines 16-Bit-Binärwertes in eine Zeichenfolge

- ❶ Gesamte Anzahl der Stellen
- ❷ Anzahl der Nachkommastellen
- ❸ Vorzeichen
- ❹ ASCII-Code des Vorzeichens
- ❺ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 1“
- ❻ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 2“
- ❼ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 3“
- ❽ ASCII-Code des Dezimalpunkts (2EH)
- ❾ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 5“
- ❿ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 6“
- ⓫ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 7“

Beispiel ▾

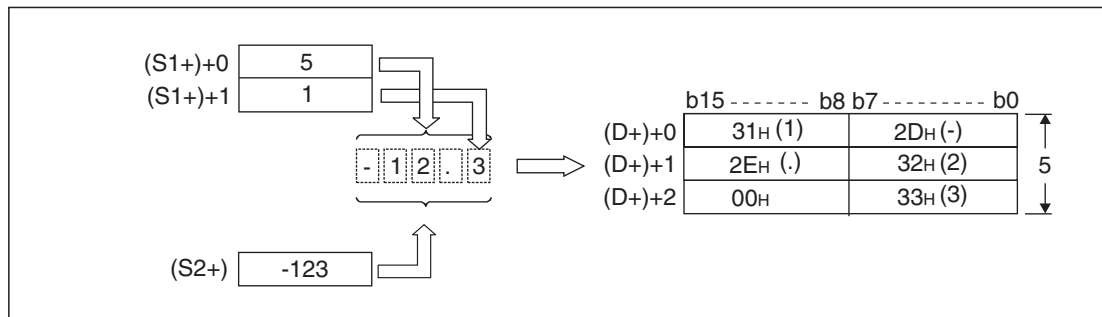


Abb. 7-210: Der binäre Wert -123 wird zu einer Zeichenfolge mit insgesamt 5 Stellen (davon eine Nachkommastelle) konvertiert. Das Ergebnis wird ab (D+) gespeichert.



32-Bit-Verarbeitung

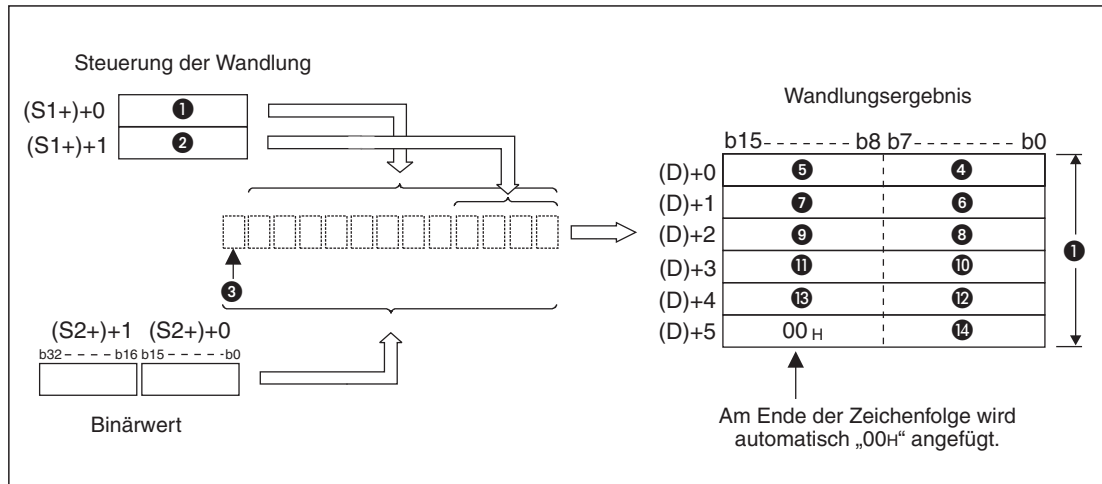


Abb. 7-211: Prinzip der Wandlung eines 32-Bit-Binärwertes in eine Zeichenfolge

- ❶ Gesamte Anzahl der Stellen
- ❷ Anzahl der Nachkommastellen
- ❸ Vorzeichen
- ❹ ASCII-Code des Vorzeichens
- ❺ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 1“
- ❻ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 2“
- ❼ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 3“
- ❽ ASCII-Code des Dezimalpunkts (2EH)
- ❾ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 5“
- ❿ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 6“
- ⓫ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 7“
- ⓬ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 8“
- ⓭ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 9“
- ⓮ ASCII-Code des Zeichens „Gesamte Stellenanzahl - 10“

Beispiel ▾

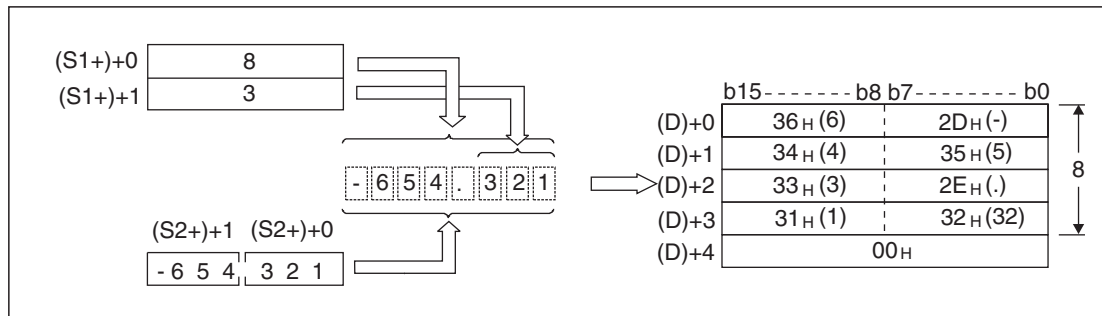


Abb. 7-212: Der binäre Wert -654321 wird zu einer Zeichenfolge mit insgesamt 8 Stellen (davon 3 Nachkommastellen) konvertiert. Das Ergebnis wird ab (D+) gespeichert.



Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Die Anzahl aller Stellen in $((S1)+0)$ liegt bei der 16-Bit-Verarbeitung nicht zwischen 2 und 8 oder bei der 32-Bit-Verarbeitung nicht zwischen 2 und 13.
- Die Anzahl der Nachkommastellen in $((S1)+1)$ liegt bei der 16-Bit-Verarbeitung nicht zwischen 0 und 5 oder bei der 32-Bit-Verarbeitung nicht zwischen 0 und 10.
- Das Verhältnis zwischen der gesamten Anzahl der Stellen in $((S1)+0)$ und der Anzahl der Nachkommastellen in $((S1)+1)$ erfüllt nicht die folgende Bedingung:
 $(\text{Anzahl aller Stellen minus } 3) \geq \text{Anzahl der Nachkommastellen}$
- In $((S1)+0)$ wurden (einschließlich des Vorzeichens und des Dezimalpunkts) weniger Stellen angegeben, als der Binärwert in $(S2+)$ bzw. $(S2+)+1)$ und $(S2+)+0)$ Stellen hat.
- Der Speicherbereich ab $(D+)$ überschreitet den zulässigen Bereich für diese Operanden.

Beispiel ▾

Beim Einschalten des Eingangs X0 wird der zuvor in D10 eingetragene Binärwert in eine Zeichenfolge mit 6 Stellen gewandelt, die anschließend ab D20 gespeichert wird. Die Anzahl aller Stellen (6) wird vor der Ausführung der STRP-Anweisung in D0 und die Anzahl der Nachkommastellen (0) wird in D1 eingetragen

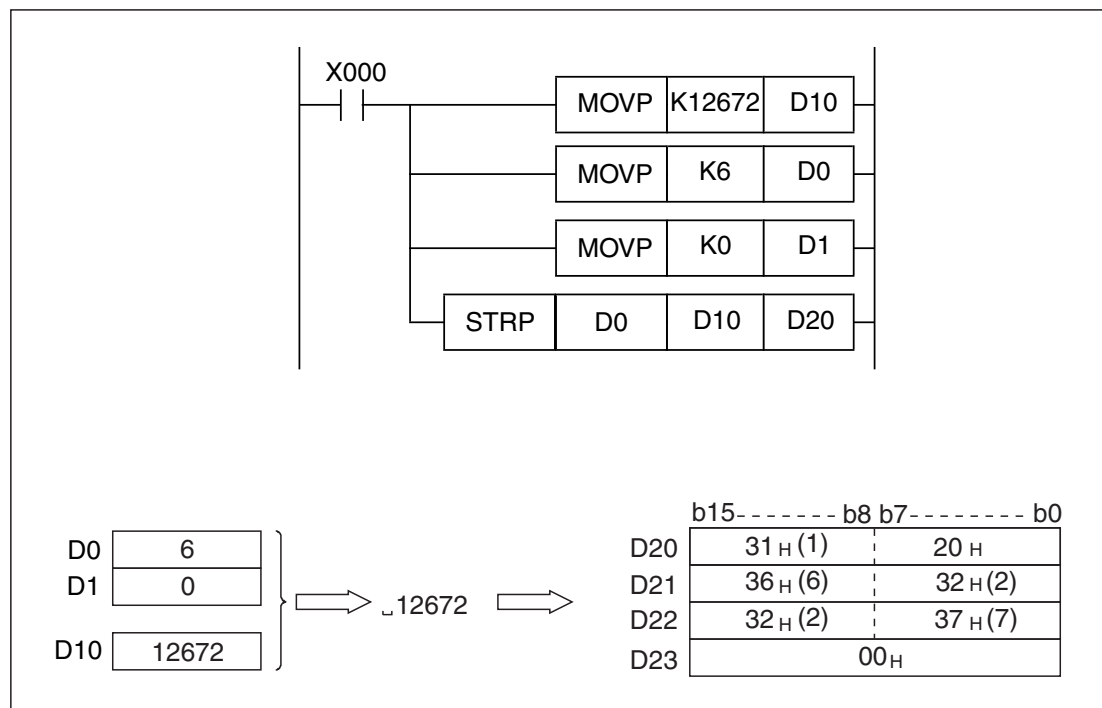


Abb. 7-213: Da der Wert in D10 nur 5 Stellen hat, die Zeichenfolge aber 6 Stellen aufweisen soll, wird automatisch ein Leerzeichen eingefügt.

△

7.16.2 Zeichenfolgen in Binärdaten wandeln (VAL)

				VAL		FNC 201				
				Zeichenfolgen in Binärdaten wandeln						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
										●
Operanden	S+	D1+	D2+	Puls-Anweisung (P) ●	Verarbeitung		Programmschritte			
	T, C, D, R	T, C, D, R	KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□)		●	16 Bit	32 Bit	VAL VALP	7	
						●	●	DVAL DVALP	13	

Funktion

Wandlung einer Zeichenfolge (ASCII-Code) in einen Binärwert

Beschreibung

- Eine ab (S+) gespeicherter Zahlenwert (Zeichenfolge) im ASCII-Code wird entsprechend den Angaben in (D1+) und ((D1+)+1) in Binärdaten gewandelt und ab (D2+) gespeichert. Bei der 16-Bit-Verarbeitung wird nur (D2+) belegt, bei der 32-Bit-Verarbeitung wird die Binärzahl in ((D2+)+1) und (D2+) eingetragen.
- Die ASCII-Daten in (S+) werden solange konvertiert, bis das Endekennzeichen „00H“ erkannt wird. Wenn ((S+)+0) ein Leerzeichen „20H“ enthält, wird die Zeichenfolge als positive Zahl interpretiert. Ein Minuszeichen (2DH) in ((S+)+0) kennzeichnet eine negative Zahl. Die Daten ab (S+) dürfen die folgenden Zeichen enthalten:
 - Leerzeichen (20H)
 - Minuszeichen (2DH)
 - Dezimalpunkt (2EH)
 - Ziffern 0 bis 9 (30H bis 39H)
- (D1+)+0 enthält die Anzahl der Zeichen, die gewandelt werden sollen (einschließlich des Vorzeichens und des Dezimalpunkts). Bei 16-Bit-Verarbeitung können 2 bis 8 und bei 32-Bit-Verarbeitung können 2 bis 13 Zeichen angegeben werden.
- In (D1+)+1 wird die Anzahl der Nachkommastellen der Zeichenfolge eingetragen, die gewandelt werden sollen. (16-Bit-Verarbeitung: 0 bis 5, 32-Bit-Verarbeitung: 0 bis 10 Nachkommastellen)

Generell gilt: Anzahl der Nachkommastellen ≤ (Anzahl aller Stellen minus 3).

Der Dezimalpunkt wird in der Zeichenfolge durch den Code „2EH“ gekennzeichnet. Der gewandelte Binärwert ist immer eine ganze Zahl (ohne Dezimalpunkt).
- Im Bereich ab (D2+) wird der gewandelte Binärwert abgelegt:
 - Der binäre Wert in (D2+) kann bei der 16-Bit-Verarbeitung im Bereich von -32768 bis 32767 liegen. Bei der 32-Bit-Verarbeitung wird der zu wandelnde Wert in (D2+)+1) und (D2+)+0) übergeben und kann im Bereich von -2.147.483.648 bis 2.147.483.647 liegen.
 - Wenn in der Zeichenfolge ab (S+) der ASCII-Code für „20H“ (Leerzeichen) oder für „30H“ (Null) zwischen dem Vorzeichen und der ersten Ziffer gesetzt ist, werden diese Zeichen bei der Konvertierung ignoriert.

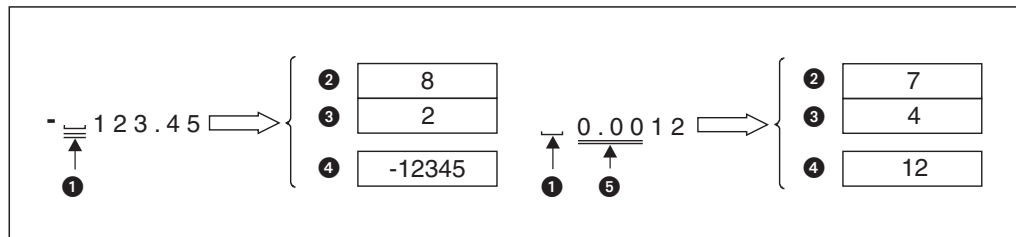


Abb. 7-215: Beispiele zur Ignorierung von Leerzeichen und Nullen bei der 16-Bit-Verarbeitung

- ❶: Leerzeichen (wird nicht gewandelt)
- ❷: Gesamte Anzahl der Zeichen in ((D1+)+0)
- ❸: Anzahl der Nachkommastellen in ((D1+)+1)
- ❹: Binärwert in (D2+)
- ❺: Nullen (werden nicht gewandelt)

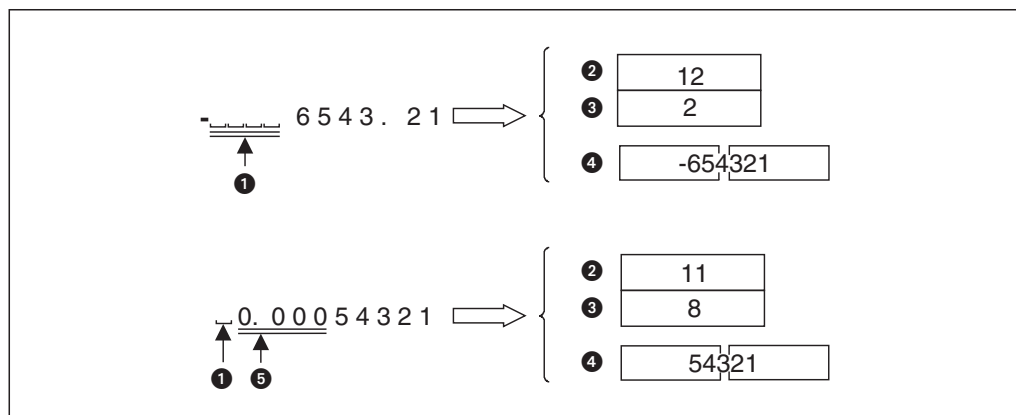


Abb. 7-214: Beispiele zur Ignorierung von Leerzeichen und Nullen bei der 32-Bit-Verarbeitung

- ❶: Leerzeichen (wird nicht gewandelt)
- ❷: Gesamte Anzahl der Zeichen in ((D1+)+0)
- ❸: Anzahl der Nachkommastellen in ((D1+)+1)
- ❹: Binärwert in ((D2+)+1) und (D2+)+0)
- ❺: Nullen (werden nicht gewandelt)

16-Bit-Verarbeitung

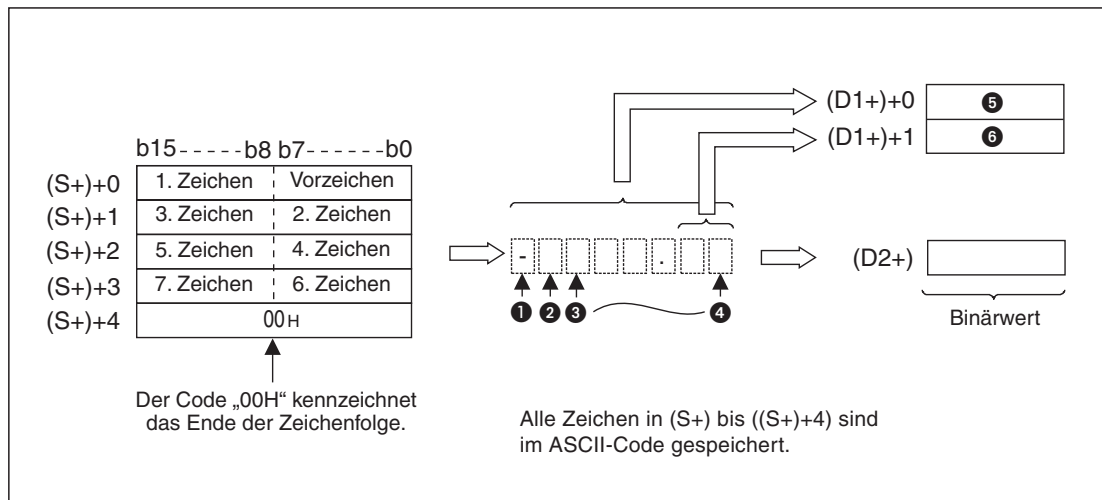


Abb. 7-216: Prinzip der Wandlung einer Zeichenfolge in einen 16-Bit-Binärwert mit einer VAL- oder VALP-Anweisung

- ❶ Vorzeichen
- ❷ 1. Zeichen
- ❸ 2. Zeichen
- ❹ 7. Zeichen
- ❺ Gesamte Anzahl der Zeichen
- ❻ Anzahl der Nachkommastellen

Beispiel ▾

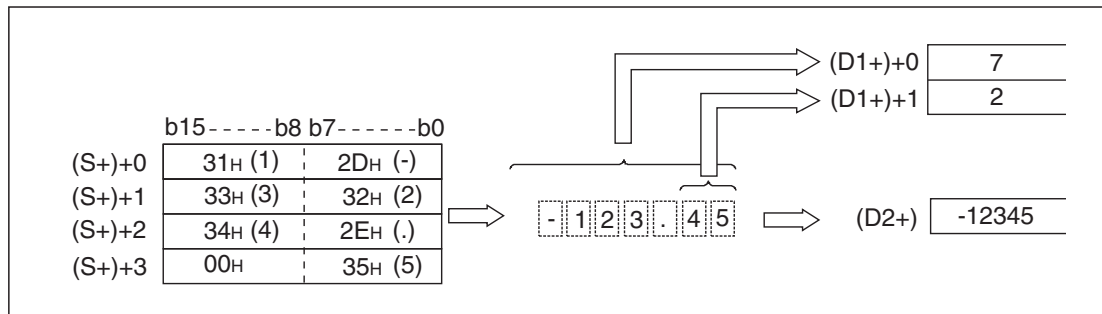


Abb. 7-217: Die Zeichenfolge „-123.45“ wird in eine Binärzahl gewandelt. Das Ergebnis wird in (D2+) gespeichert.



32-Bit-Verarbeitung

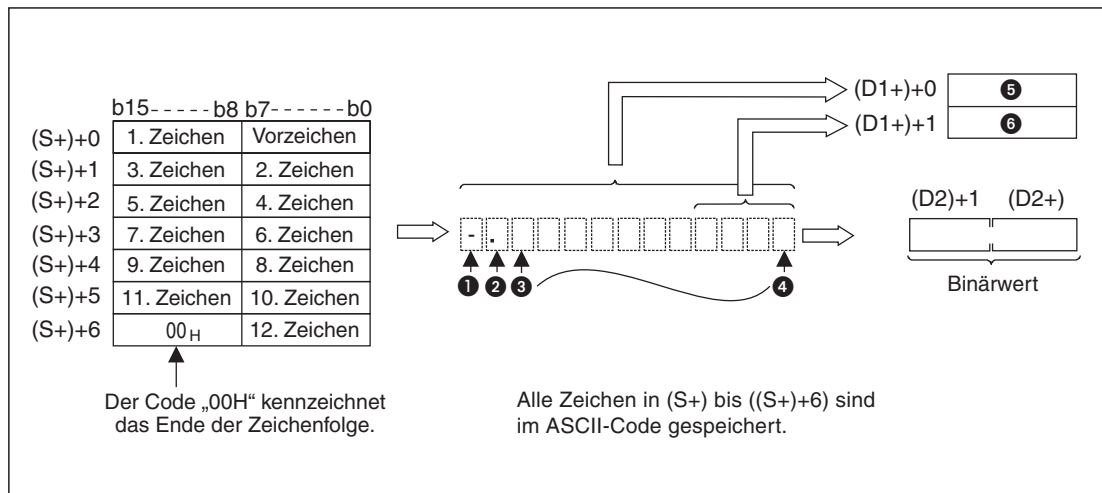


Abb. 7-218: Prinzip der Wandlung einer Zeichenfolge in einen 32-Bit-Binärwert mit einer DVAL- oder DVALP-Anweisung

- ❶ Vorzeichen
- ❷ 1. Zeichen
- ❸ 2. Zeichen
- ❹ 12. Zeichen
- ❺ Gesamte Anzahl der Zeichen
- ❻ Anzahl der Nachkommastellen

Beispiel ▾

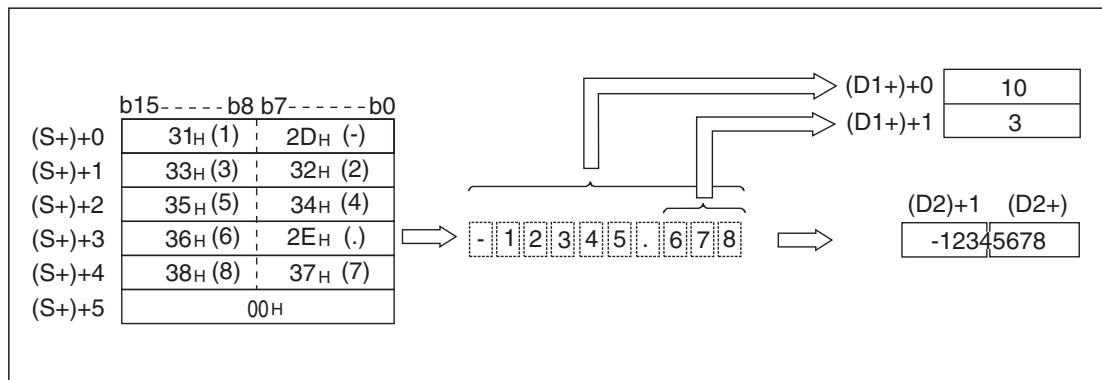


Abb. 7-219: Die Zeichenfolge „-12345.78“ wird in eine Binärzahl gewandelt. Das Ergebnis wird in ((D2+)+1) und (D2+) gespeichert.



Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Die Anzahl aller Stellen in ((D1)+0) liegt bei der 16-Bit-Verarbeitung nicht zwischen 2 und 8 oder bei der 32-Bit-Verarbeitung nicht zwischen 2 und 13.
- Die Anzahl der Nachkommastellen in ((D1)+1) liegt bei der 16-Bit-Verarbeitung nicht zwischen 0 und 5 oder bei der 32-Bit-Verarbeitung nicht zwischen 0 und 10.
- Das Verhältnis zwischen der gesamten Anzahl der Stellen in ((D1)+0) und der Anzahl der Nachkommastellen in ((D1)+1) erfüllt nicht die folgende Bedingung:
(Anzahl aller Stellen minus 3) ≥ Anzahl der Nachkommastellen
- Das Vorzeichen der Zeichenfolge besteht nicht aus den ASCII-Zeichen 20H (Leerzeichen) oder 2DH (Minuszeichen).
- Die Zeichenfolge enthält andere Zeichen als die Ziffern 0 bis 9 (30H bis 39H), das Leerzeichen (20H), den Dezimalpunkt (2EH) oder das Minuszeichen (2DH).
- Die zu konvertierende Zeichenfolge enthält mehr als einen Dezimalpunkt (2EH).
- Der aus der Zeichenfolge gewandelte Binärwert liegt bei der 16-Bit-Verarbeitung außerhalb des Bereichs von -32768 bis 32767 und bei der 32-Bit-Verarbeitung außerhalb des Bereichs von -2.147.483.648 bis 2.147.483.647.
- Die Zeichenfolge wird nicht durch „00H“ abgeschlossen.

Beispiel ▾

Beim Einschalten von X20 wird die ab D20 gespeicherte Zeichenfolge in einen 16-Bit-Binärwert gewandelt und in D0 eingetragen. Das Datenregister D19 enthält die Anzahl aller Stellen (6) und D11 die Anzahl der Nachkommastellen der Zeichenfolge (2).

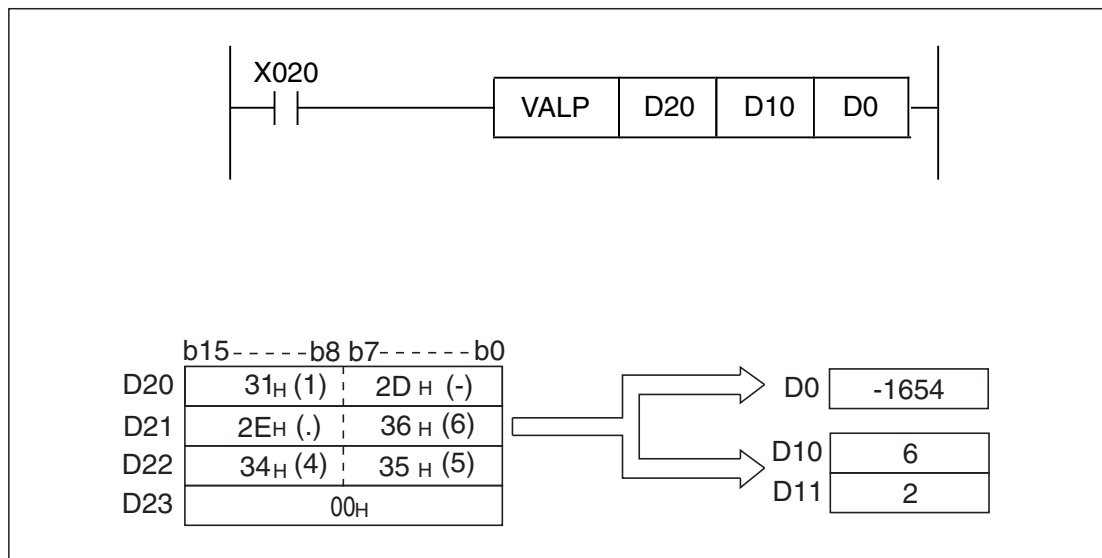


Abb. 7-220: Programmbeispiel zur Wandlung einer Zeichenfolge in eine 16-Bit-Binärzahl



7.16.3 Zeichenfolgen zusammenfügen (\$+)

				\$+		FNC 202				
				Zeichenfolgen zusammenfügen						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
				●						●
Operanden	S1+	S2+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	KnX, KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□), Zeichenfolgen			KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□)	●	16 Bit	32 Bit	\$+ \$+P		7

Funktion

Zwei Zeichenfolgen werden zusammengefügt

Beschreibung

- Die Zeichenfolge ab (S2+) wird an die in (S1+) angegebene Zeichenfolge angehängt.
- Das Ergebnis wird ab (D+) gespeichert.
- Das Ende der Zeichenfolgen in (S1+) und (S2+) wird durch das ASCII-Zeichen NUL (00H) gekennzeichnet.
- Die verknüpfte Zeichenfolge beginnt mit dem Zeichen im niederwertigen Byte in (S1+) angegebenen Zeichenfolge und endet mit dem Code „00H“ der in (S2+) angegebenen Zeichenfolge. Das Endekennzeichen „00H“ der in (S1+) angegebenen Zeichenfolge erscheint nicht in (D+).
- Nach der zusammengeführten Zeichenfolge wird automatisch das Zeichen „NUL“ (00H) eingefügt. Bei einer geraden Anzahl Zeichen wird „00H“ in den Operanden eingetragen, der auf dem Operanden mit dem letzten gewandelten Zeichen folgt. Bei einer ungeraden Anzahl Zeichen wird „00H“ in das höherwertige Byte des Operanden eingetragen, der im niederwertigen Byte das letzte Zeichen enthält.

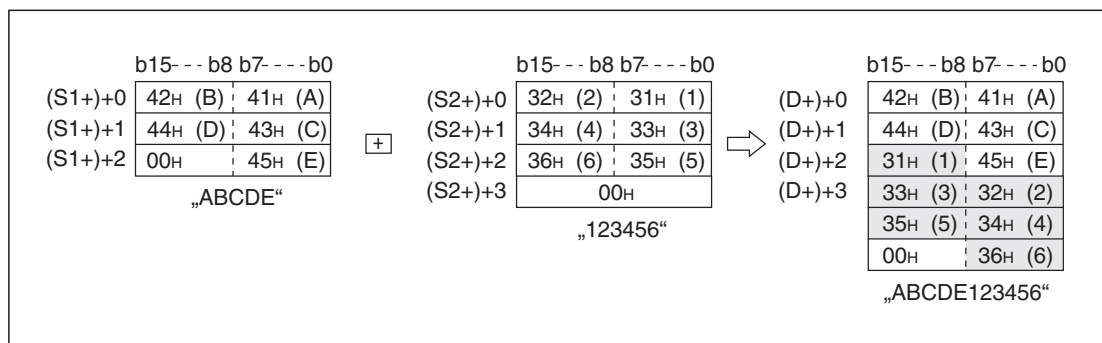


Abb. 7-221: Die Zeichenfolgen in (S1+) und (S2+) werden zusammengefasst und in (D+) gespeichert. (Die Zeichenfolge aus (S2+) ist hier grau hinterlegt.)

- In (S1+) und (S2+) können auch direkt Zeichenfolgen mit bis zu 32 Zeichen angegeben werden. Falls in (S1+) und (S2+) Wortoperanden angegeben werden, besteht keine Einschränkung bei der Anzahl der Zeichen.
- Falls beide Zeichenfolgen in (S1+) und (S2+) mit „00H“ beginnen (in diesem Fall ist die Anzahl der Zeichen „0“), wird ab (D+) „0000H“ eingetragen.

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Die vollständige, verknüpfte Zeichenfolge kann nicht ab (D+) gespeichert werden.
- Die Datenbereiche von (S1+), (S2+) oder (D+) überlappen sich.
- Innerhalb der Operandenbereiche für die in (S1+) oder (S2+) angegebenen Operanden ist nicht der Code „00H“ eingetragen.

Beispiel ▾

Wenn im folgenden Beispiel der Eingang X000 eingeschaltet wird, werden die ab D10 gespeicherte Zeichenfolge „abcde“ und die Zeichenfolge „ABCD“ verbunden. Die zusammengefügte Zeichenfolge wird ab D100 gespeichert.

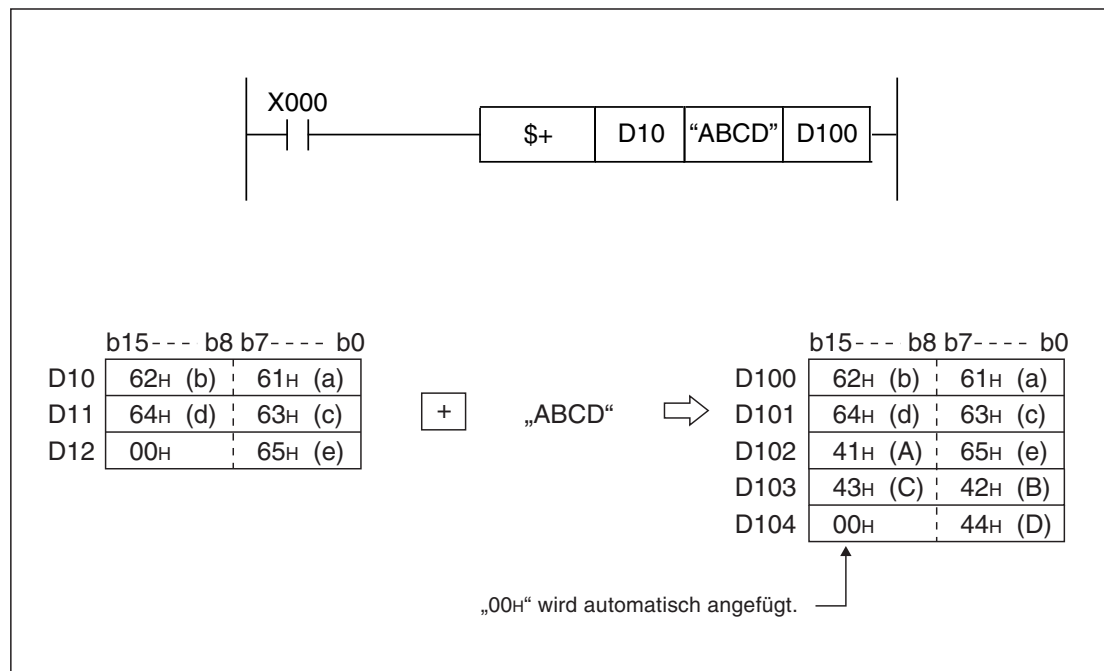


Abb. 7-222: Beispiel für das Zusammenfügen zweier Zeichenfolgen



7.16.4 Länge von Zeichenfolgen ermitteln (LEN)

		LEN		FNC 203					
		Länge von Zeichenfolgen ermitteln							
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
Operanden	S+	D+		Puls-Anweisung (P)		Verarbeitung		Programmschritte	
	KnX, KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□),	KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□)		●		16 Bit	32 Bit	LEN LENP	5

Funktion

Ermittlung der Länge von Zeichenfolgen

Beschreibung

- Die LEN-Anweisung ermittelt die Anzahl der Zeichen der in (S+) angegebenen Zeichenfolge und speichert das Ergebnis in (D+).
- Die Zählung beginnt mit dem Zeichen im niederwertigen Byte der in (S1+) angegebenen Zeichenfolge und endet, wenn der Code „00H“ erkannt wird.
- Da ein ASCII-Zeichen ein Byte belegt, ist die Maßeinheit für die Länge der Zeichenfolge „Byte“.
- Die LEN-Anweisung erkennt auch Zeichen, die nicht im ASCII-Code gespeichert sind. Die Einheit der Zeichen ist aber immer „Byte“. Werden z. B. Zeichen im JIS-Code gezählt (1 Zeichen = 16 Bit), wird für die Länge eines Zeichens „2“ angegeben.

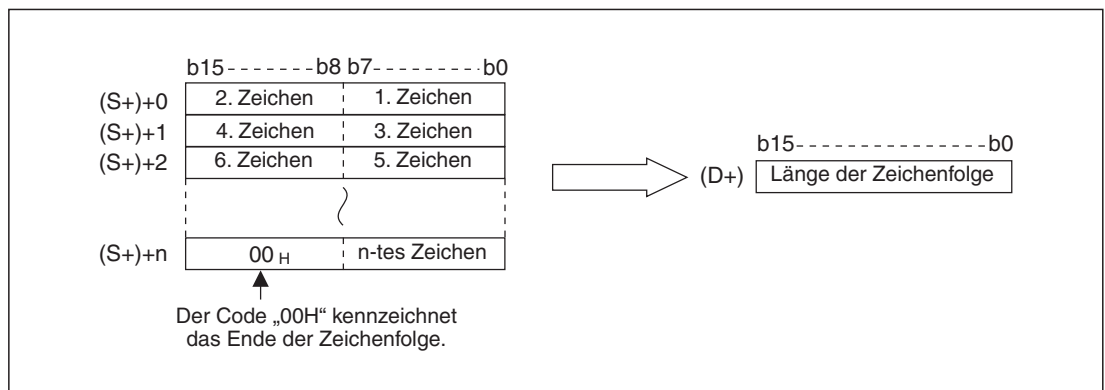


Abb. 7-223: Die LEN-Anweisung zählt die Anzahl der Zeichen einer Zeichenfolge und trägt den Wert in (D+) ein.

Beispiel ▾

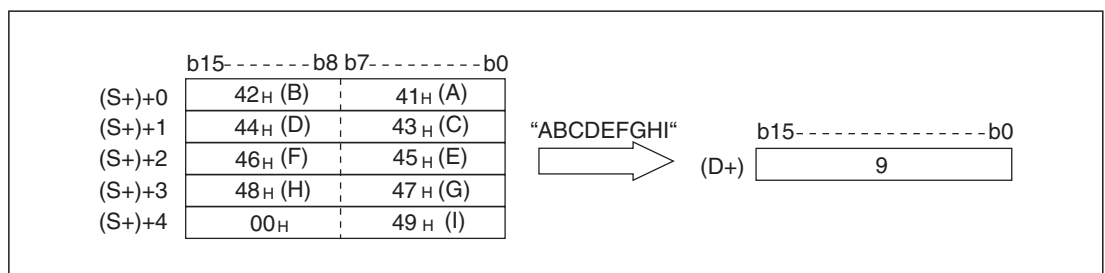


Abb. 7-224: In diesem Beispiel werden die Zeichen in der Zeichenfolge „ABCDEFGHI“ gezählt und in (D+) als Ergebnis „9“ eingetragen.



Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Die Länge der Zeichenfolge überschreitet 32768 Bytes.
- Innerhalb des Operandenbereichs des in (S+) angegebenen Operanden ist nicht der Code „00H“ eingetragen.

Beispiel ▽

Bei eingeschaltetem Eingang X10 wird im folgenden Beispiel die Länge der Zeichenfolge gezählt, die ab D0 gespeichert ist. Das Ergebnis wird in D10 gespeichert. Anschließend wird der ermittelte Wert in eine BCD-Zahl gewandelt und an den Ausgängen Y057 bis Y40 ausgegeben.

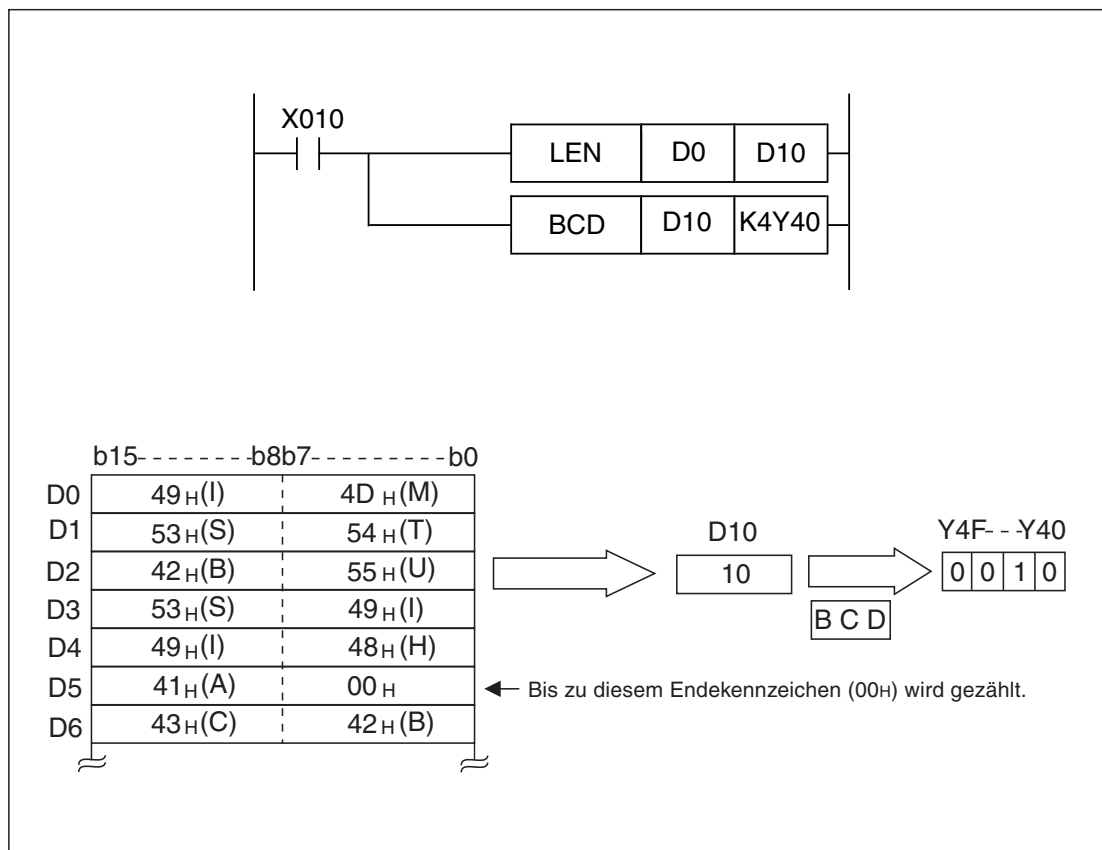


Abb. 7-225: In diesem Beispiel wird die Anzahl der Zeichen im Wort „MITSUBISHI“ gezählt.



7.16.5 Auszug der Zeichenfolgedaten von rechts (RIGHT)

				RIGHT		FNC 204			
				Auszug der Zeichenfolgedaten von rechts					
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S
Operanden	S+	D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	KnX, KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□)	KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□)	D, R, K, H		16 Bit	32 Bit	RIGHT RIGHTP	7	

Funktion

Zeichen aus einer Zeichenfolge im ASCII-Code lesen. Begonnen wird am Ende der Zeichenfolge (rechts).

Beschreibung

- Die RIGHT-Anweisung liest n Zeichen, beginnend beim Ende der Zeichenfolge (rechte Seite der Zeichenfolge).
- In (S+) wird angegeben, wo die Zeichenfolge gespeichert ist. Die Zeichenfolge beginnt mit dem Zeichen im niederwertigen Byte der in (S1+) angegebenen Zeichenfolge und endet mit dem Code „00H“.
- Die extrahierten Zeichen werden ab (D+) gespeichert.
- Nach den gelesenen Zeichen wird automatisch das Zeichen „NUL“ (00H) eingefügt. Bei einer geraden Anzahl Zeichen wird „00H“ in den Operanden eingetragen, der auf dem Operanden mit dem letzten gewandelten Zeichen folgt. Bei einer ungeraden Anzahl Zeichen wird „00H“ in das höherwertige Byte des Operanden eingetragen, der im niederwertigen Byte das letzte Zeichen enthält.
- Die Anzahl der zu lesenden Zeichen wird mit (n) in der Einheit „Byte“ angegeben. Bei (n) = 0 wird in ((D+)+0) der Code NUL (00H) eingetragen.

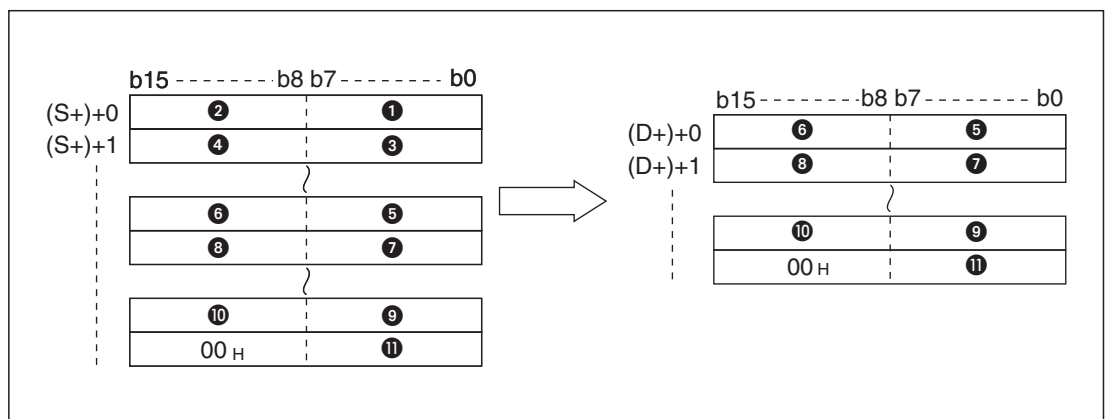


Abb. 7-226: Mit einer RIGHT-Anweisung werden die Zeichen ab dem Ende einer Zeichenfolge erfasst.

- ① ASCII-Code des 1. Zeichens
- ② ASCII-Code des 2. Zeichens
- ③ ASCII-Code des 3. Zeichens
- ④ ASCII-Code des 4. Zeichens

- ⑤ ASCII-Code des Zeichens: Letztes Zeichen - (n+1)
- ⑥ ASCII-Code des Zeichens: Letztes Zeichen - (n+2)
- ⑦ ASCII-Code des Zeichens: Letztes Zeichen - (n+3)
- ⑧ ASCII-Code des Zeichens: Letztes Zeichen - (n+4)
- ⑨ ASCII-Code des Zeichens: Letztes Zeichen - 2
- ⑩ ASCII-Code des Zeichens: Letztes Zeichen - 1
- ⑪ ASCII-Code des letzten Zeichens

Beispiel ▾

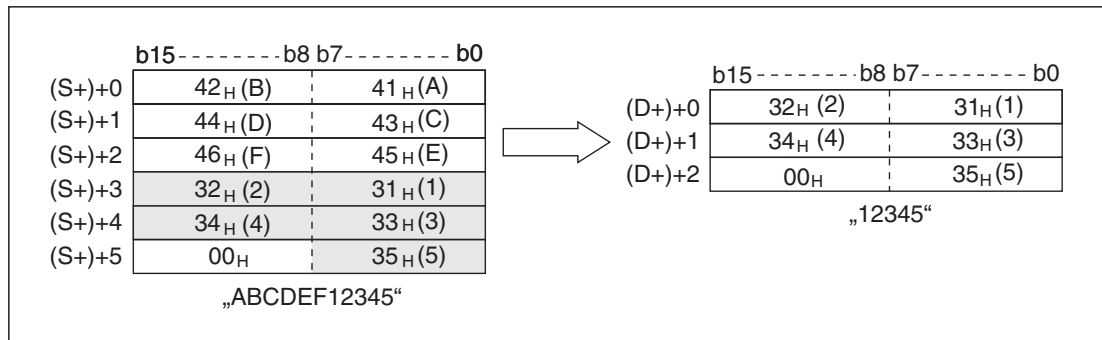


Abb. 7-227: Beispiel für die Anwendung einer RIGHT-Anweisung: Die letzten 5 Zeichen der Zeichenfolge „ABCDEF12345“ werden gelesen und ab (D+) gespeichert



HINWEIS

Die Angabe der Anzahl der Zeichen in (n) erfolgt in der Einheit „Byte“. Dies muss berücksichtigt werden, falls Zeichen gelesen werden sollen, die nicht im ASCII-Code gespeichert sind. Werden z. B. Zeichen im JIS-Code erfasst werden sollen (1 Zeichen = 16 Bit), muss die Anzahl der gewünschten Zeichen mit 2 multipliziert und dieser Wert in (n) eingetragen werden.

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Die Angabe in (n) für die Anzahl der zu lesenden Zeichen ist größer als die Anzahl der ab (S+) gespeicherten Zeichen.
- Für (n) wurde eine negative Zahl angegeben.
- Innerhalb des Operandenbereichs des in (S+) angegebenen Operanden ist nicht der Code „00H“ eingetragen.
- Durch die Angabe für (n) wird der zur Speicherung der Zeichen ab (D+) erforderliche Operandenbereich überschritten.

Beispiel ▾

Wenn der Eingang X000 eingeschaltet wird, werden aus der Zeichenfolge, die ab R0 gespeichert ist, die letzten 4 Zeichen gelesen und ab D0 gespeichert.

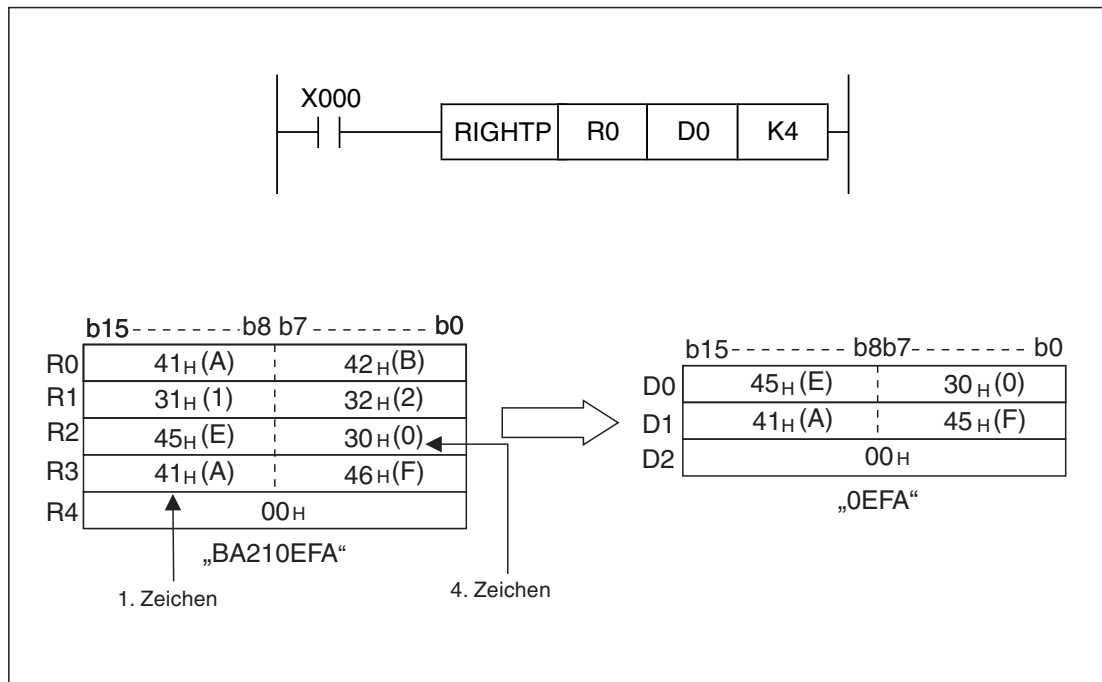


Abb. 7-228: Programmbeispiel zur Anwendung einer `RIGHT(P)`-Anweisung



7.16.6 Auszug der Zeichenfolgedaten von links (LEFT)

				LEFT		FNC 205				
				Auszug der Zeichenfolgedaten von links						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
										●
Operanden	S+	D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	KnX, KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□)	KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□)	D, R, K, H	●	16 Bit	32 Bit	LEFT LEFTP	7		

Funktion

Zeichen aus einer Zeichenfolge im ASCII-Code lesen. Begonnen wird am Anfang der Zeichenfolge (links).

Beschreibung

- Die LEFT-Anweisung liest n Zeichen, beginnend beim Anfang der Zeichenfolge (linke Seite der Zeichenfolge).
- In (S+) wird angegeben, wo die Zeichenfolge gespeichert ist. Die Zeichenfolge beginnt mit dem Zeichen im niederwertigen Byte der in (S1+) angegebenen Zeichenfolge und endet mit dem Code „00H“.
- Die extrahierten Zeichen werden ab (D+) gespeichert.
- Nach den gelesenen Zeichen wird automatisch das Zeichen „NUL“ (00H) eingefügt. Bei einer geraden Anzahl Zeichen wird „00H“ in den Operanden eingetragen, der auf dem Operanden mit dem letzten gewandelten Zeichen folgt. Bei einer ungeraden Anzahl Zeichen wird „00H“ in das höherwertige Byte des Operanden eingetragen, der im niederwertigen Byte das letzte Zeichen enthält.
- Die Anzahl der zu lesenden Zeichen wird mit (n) in der Einheit „Byte“ angegeben. Bei (n) = 0 wird in ((D+)+0) der Code NUL (00H) eingetragen.

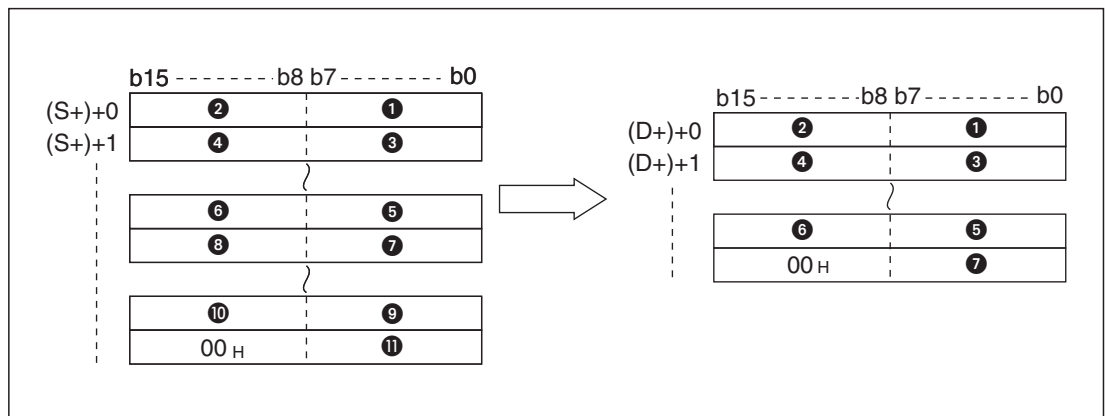


Abb. 7-229: Mit einer LEFT-Anweisung werden die Zeichen ab dem Anfang einer Zeichenfolge erfasst.

- ① ASCII-Code des 1. Zeichens
- ② ASCII-Code des 2. Zeichens
- ③ ASCII-Code des 3. Zeichens
- ④ ASCII-Code des 4. Zeichens

- ⑤ ASCII-Code des Zeichens: Letztes Zeichen - (n+1)
- ⑥ ASCII-Code des Zeichens: Letztes Zeichen - (n+2)
- ⑦ ASCII-Code des Zeichens: Letztes Zeichen - (n+3)
- ⑧ ASCII-Code des Zeichens: Letztes Zeichen - (n+4)
- ⑨ ASCII-Code des Zeichens: Letztes Zeichen - 2
- ⑩ ASCII-Code des Zeichens: Letztes Zeichen - 1
- ⑪ ASCII-Code des letzten Zeichens

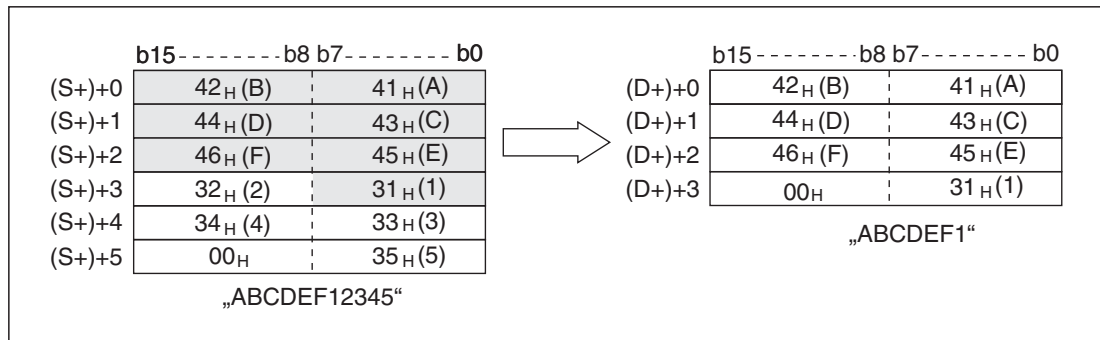
Beispiel ▾

Abb. 7-230: Beispiel für die Anwendung einer LEFT-Anweisung: Die ersten 7 Zeichen der Zeichenfolge „ABCDEF12345“ werden gelesen und ab (D+) gespeichert

**HINWEIS**

Die Angabe der Anzahl der Zeichen in (n) erfolgt in der Einheit „Byte“. Dies muss berücksichtigt werden, falls Zeichen gelesen werden sollen, die nicht im ASCII-Code gespeichert sind. Werden z. B. Zeichen im JIS-Code erfasst werden sollen (1 Zeichen = 16 Bit), muss die Anzahl der gewünschten Zeichen mit 2 multipliziert und dieser Wert in (n) eingetragen werden.

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Die Angabe in (n) für die Anzahl der zu lesenden Zeichen ist größer als die Anzahl der ab (S+) gespeicherten Zeichen.
- Für (n) wurde eine negative Zahl angegeben.
- Innerhalb des Operandenbereichs des in (S+) angegebenen Operanden ist nicht der Code „00H“ eingetragen.
- Durch die Angabe für (n) wird der zur Speicherung der Zeichen ab (D+) erforderliche Operandenbereich überschritten.

Beispiel ▾

Im folgenden Programmbeispiel werden beim Einschalten des Eingangs X010 die ersten 6 Zeichen der Zeichenfolge, die ab D100 gespeichert ist, gelesen und ab R10 gespeichert. Die Anzahl der Zeichen wird im Datenregister D0 angegeben.

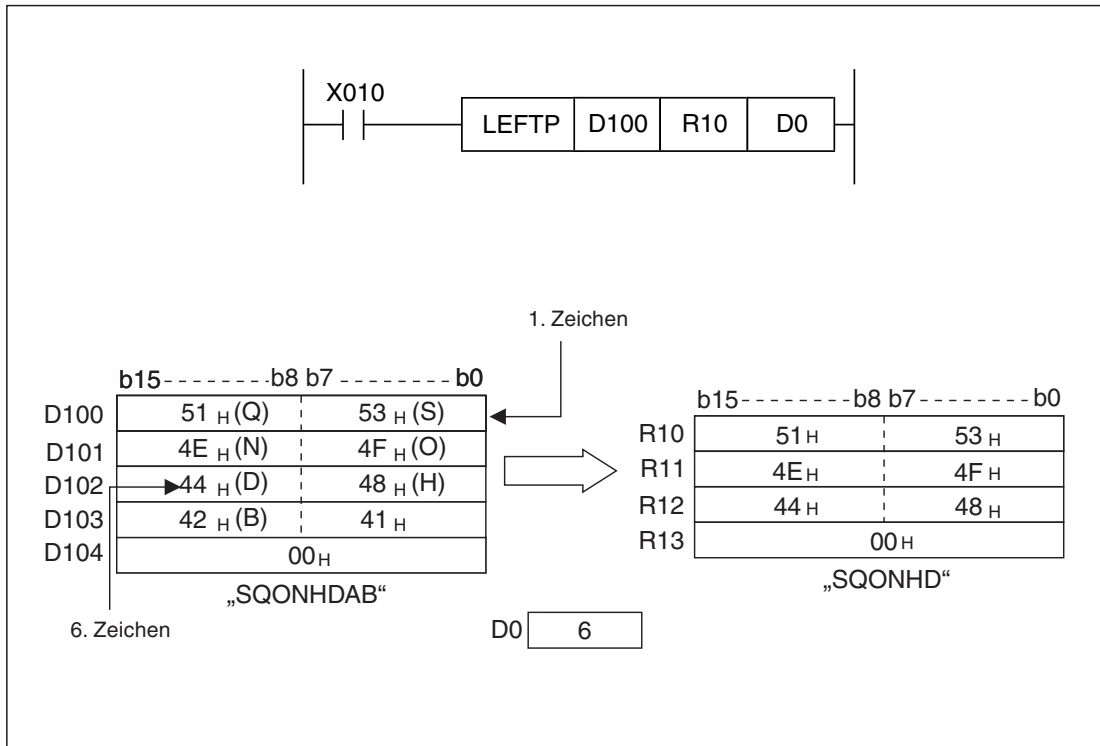


Abb. 7-231: Programmbeispiel zur Anwendung einer `LEFT(P)`-Anweisung



7.16.7 Zeichen aus Zeichenfolge kopieren (MIDR)

				MIDR		FNC 206				
				Zeichen aus Zeichenfolge kopieren						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
										●
Operanden	S1+	D+	S2+	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	KnX, KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□)	KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□)	KnX, KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□)		●	16 Bit	32 Bit	MIDR MIDRP	7	

Funktion

Zeichen ab einer beliebigen Position aus einer Zeichenfolge kopieren

Beschreibung

- Die MIDR-Anweisung liest einen durch die Angabe der Position des ersten Zeichens und der zu lesenden Anzahl der Zeichen definierten Bereich einer Zeichenfolge.
- In (S1+) wird angegeben, wo die Zeichenfolge gespeichert ist. Die Zeichenfolge beginnt mit dem Zeichen im niederwertigen Byte der in (S1+) angegebenen Zeichenfolge und endet mit dem Code „00H“.
- In ((S2+)+0) wird angegeben, ab welchem Zeichen der Zeichenfolge gelesen werden soll.
- ((S2+)+1) enthält die Anzahl der Bytes (Zeichen), die kopiert werden sollen. Wird hier der Wert „0“ eingetragen, wird die MIDR-Anweisung nicht ausgeführt. Beim Wert „-1“ werden ab dem ersten abgegebenem Zeichen alle restlichen Zeichen der ab (S1+) gespeicherten Zeichenkette kopiert (siehe Abb. 7-233).
- Die extrahierten Zeichen werden ab (D+) gespeichert.
- Nach den kopierten Zeichen wird automatisch das Zeichen „NUL“ (00H) eingefügt. Bei einer geraden Anzahl Zeichen wird „00H“ in den Operanden eingetragen, der auf dem Operanden mit dem letzten gewandelten Zeichen folgt. Bei einer ungeraden Anzahl Zeichen wird „00H“ in das höherwertige Byte des Operanden eingetragen, der im niederwertigen Byte das letzte Zeichen enthält.

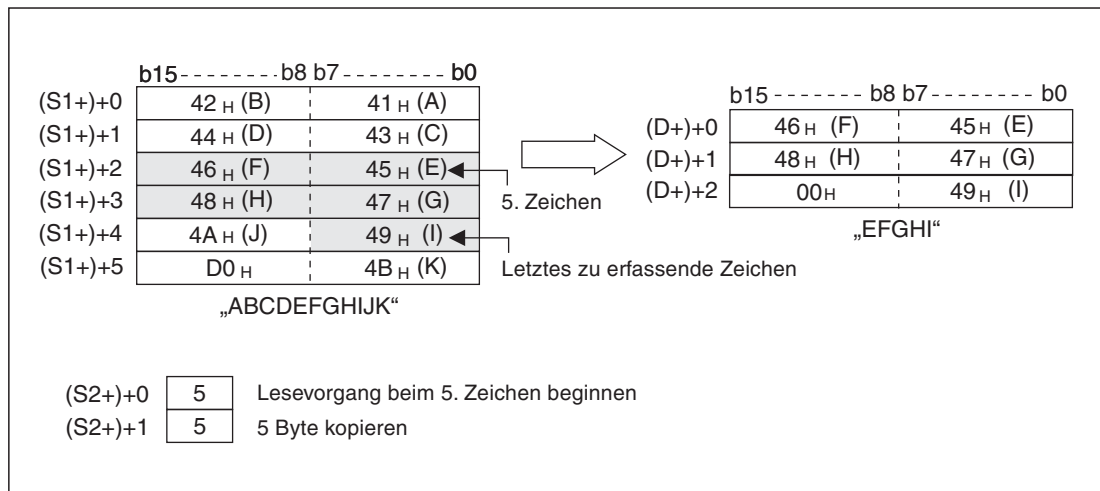


Abb. 7-232: Ab welchem Zeichen und wie viele Zeichen kopiert werden sollen, wird bei der MIDR-Anweisung in ((S2+)+0) und ((S2+)+1) eingetragen

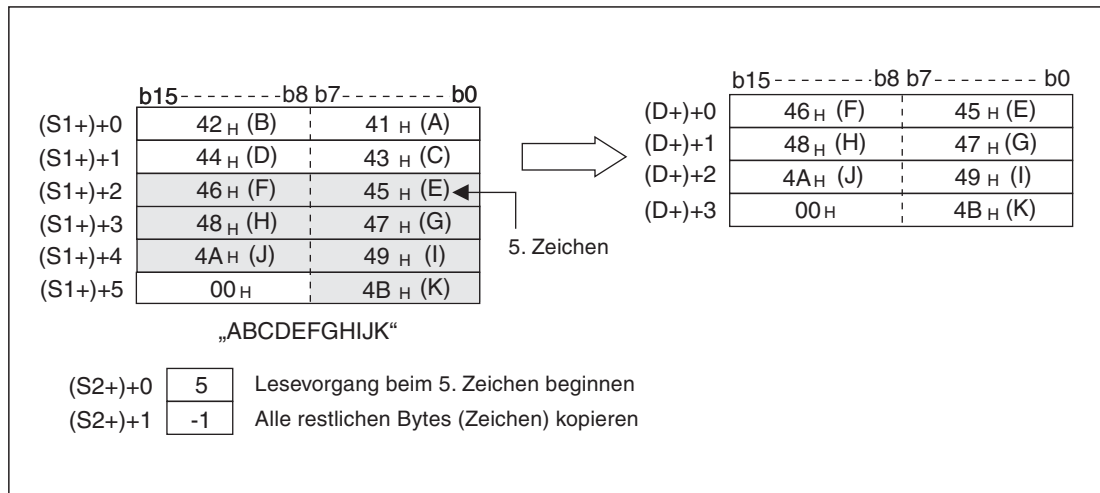


Abb. 7-233: Alle restlichen Zeichen der Zeichenfolge werden kopiert, wenn in ((S2+)+1 der Wert „-1“ eingetragen wird.

HINWEIS

Die Angabe der Anzahl der Zeichen in ((S2+)+1) erfolgt in der Einheit „Byte“. Dies muss berücksichtigt werden, falls Zeichen gelesen werden sollen, die nicht im ASCII-Code gespeichert sind. Werden z. B. Zeichen im JIS-Code erfasst werden sollen (1 Zeichen = 16 Bit), muss die Anzahl der gewünschten Zeichen mit 2 multipliziert und dieser Wert in ((S2+)+1) eingetragen werden.

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Die Angabe in ((S2+)+1) für die Anzahl der zu kopierenden Zeichen ist größer als die Anzahl der ab (S1+) gespeicherten Zeichen.
- Innerhalb des Operandenbereichs des in (S1+) angegebenen Operanden ist nicht der Code „00H“ eingetragen.
- Durch die Angabe für ((S2+)+1) wird der zur Speicherung der Zeichen ab (D+) erforderliche Operandenbereich überschritten.
- Für ((S2+)+0) wurde eine negative Zahl angegeben.
- Für ((S2+)+1) wurde ein kleinerer Wert als -1 angegeben.

Beispiel ▾

Im folgenden Programmbeispiel werden beim Einschalten des Eingangs X000 aus der Zeichenfolge, die ab D10 gespeichert ist, sechs Zeichen ab dem dritten Zeichen gelesen und ab R0 gespeichert. Die Position des ersten zu kopierenden Zeichens wird in R0 und die Anzahl der Zeichen in R1 angegeben.

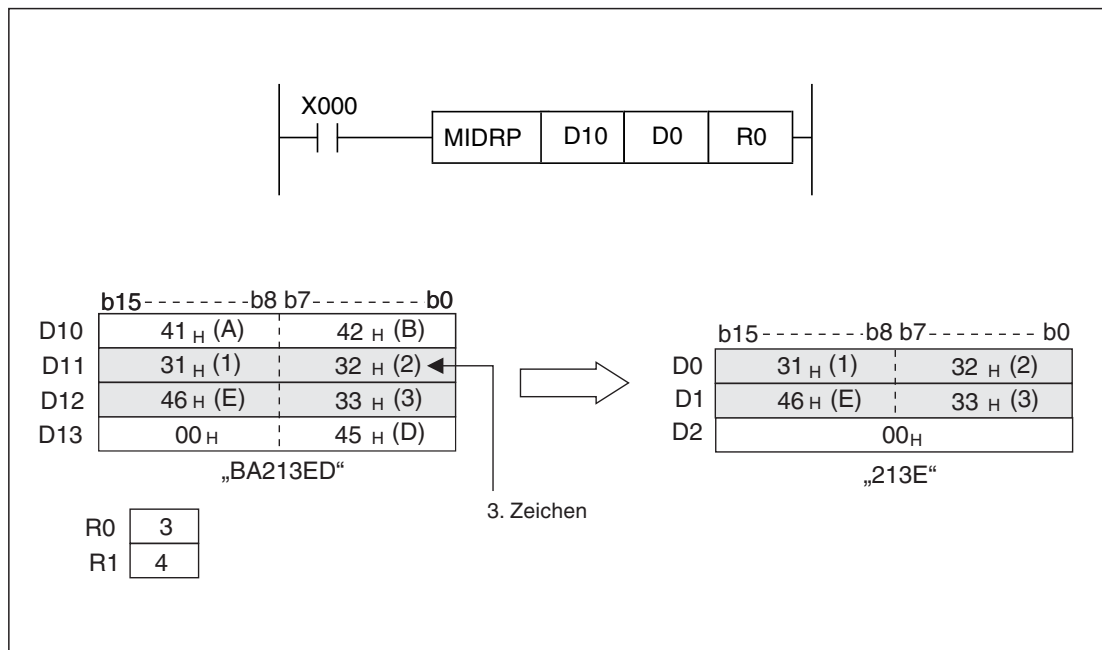


Abb. 7-234: Aus der Zeichenfolge „BA213ED“ werden die Zeichen „213E“ herauskopiert.



7.16.8 Zeichenfolge ersetzen (MIDW)

				MIDW		FNC 207				
				Zeichenfolge ersetzen						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
										●
Operanden	S1+	D+	S2+	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	KnX, KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□)	KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□)	KnX, KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□)		●	16 Bit	32 Bit	MIDW MIDWP	7	

Funktion

Zeichen in einer Zeichenfolge durch andere Zeichen ersetzen

Beschreibung

- Die MIDW-Anweisung schreibt Zeichen aus der mit (S1+) angegebenen Zeichenfolge in die mit (D+) angegebene Zeichenfolge.
- Die Zeichen werden ab dem Anfang (linkes Ende) der in (S1+) angegebenen Zeichenfolge entnommen.
- Die Zeichenfolgen in (S1+) und (D+) beginnen mit dem Zeichen im niederwertigen Byte und enden mit dem Code „00H“.
- In ((S2+)+0) wird angegeben, ab welchem Zeichen der Zeichenfolge in (D+) die Zeichen eingefügt werden sollen.
- ((S2+)+1) enthält die Anzahl der Bytes (Zeichen), die überschrieben werden sollen.

Wird hier der Wert „0“ eingetragen, werden keine Zeichen ersetzt. Überschreitet der Wert in ((S2+)+1) die Anzahl der in (D+) speicherbaren Zeichen, werden nur so viele Zeichen übertragen, wie möglich sind (Abb. 7-236). Beim Wert „-1“ wird die gesamte Zeichenfolge aus (S1+) in die Zeichenfolge in (D+) übertragen (Abb. 7-237).

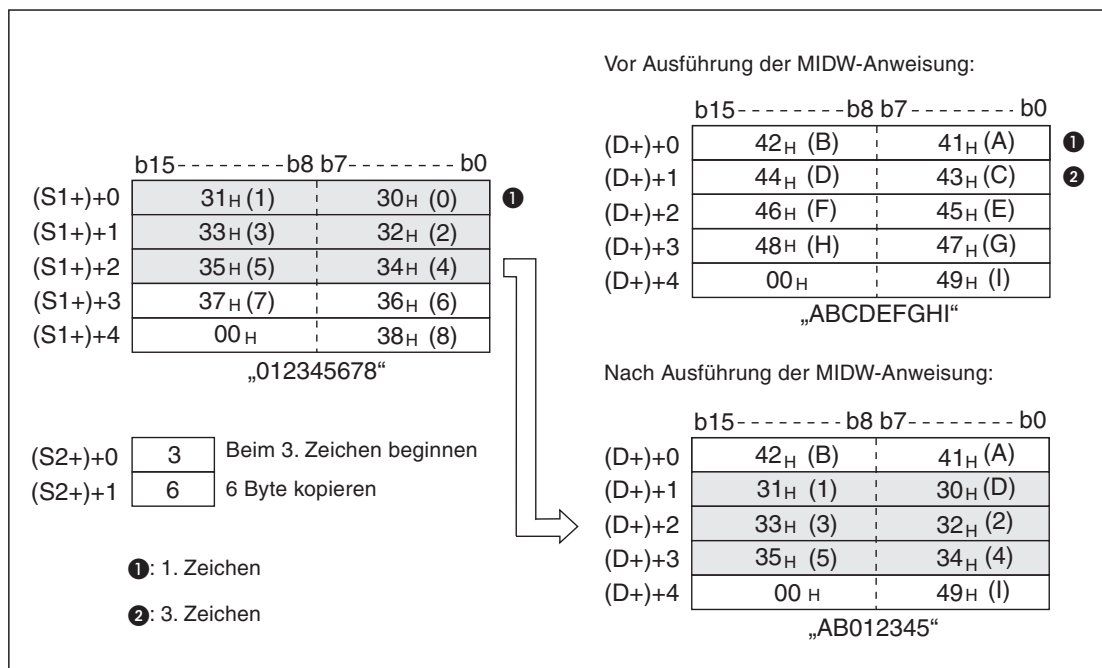


Abb. 7-235: Mit einer MIDW-Anweisung können Zeichen in einer Zeichenfolge ersetzt werden.

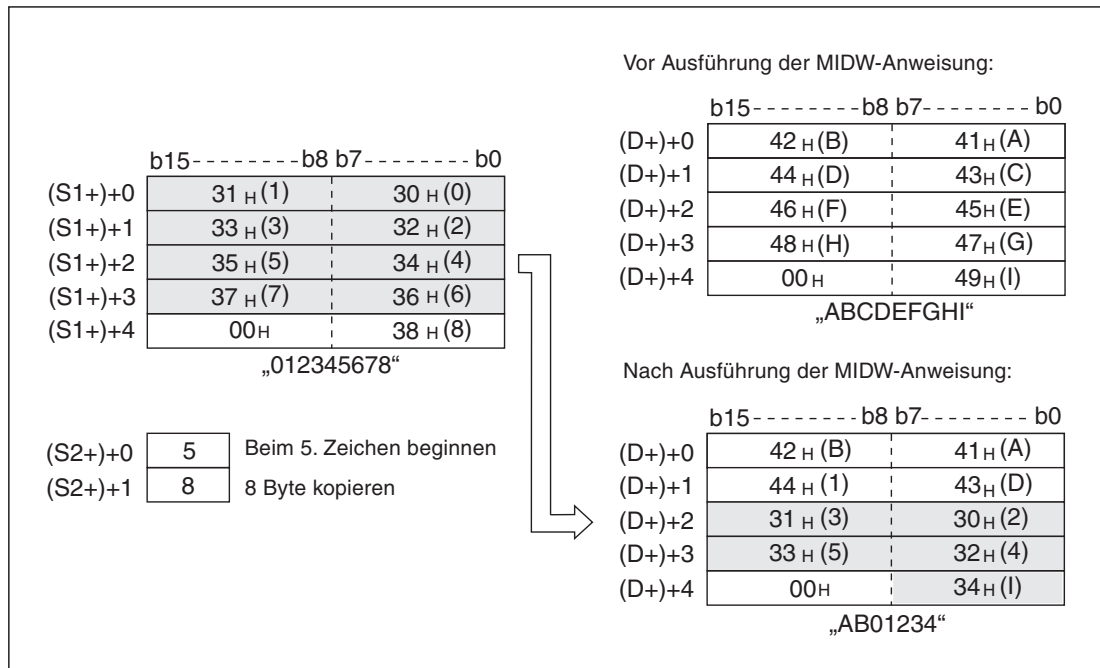


Abb. 7-236: In die Zeichenfolge in (D+) werden nur Zeichen bis zum Code „00H“ eingefügt. In diesem Beispiel sollen ab dem 5. Zeichen der Zeichenkette in (D+) 8 Zeichen eingefügt werden. Ab dem 5. Zeichen in ((D+)+2) können aber nur 5 Zeichen ersetzt werden. Die restlichen Zeichen der Zeichenfolge in (S1+) werden abgeschnitten.

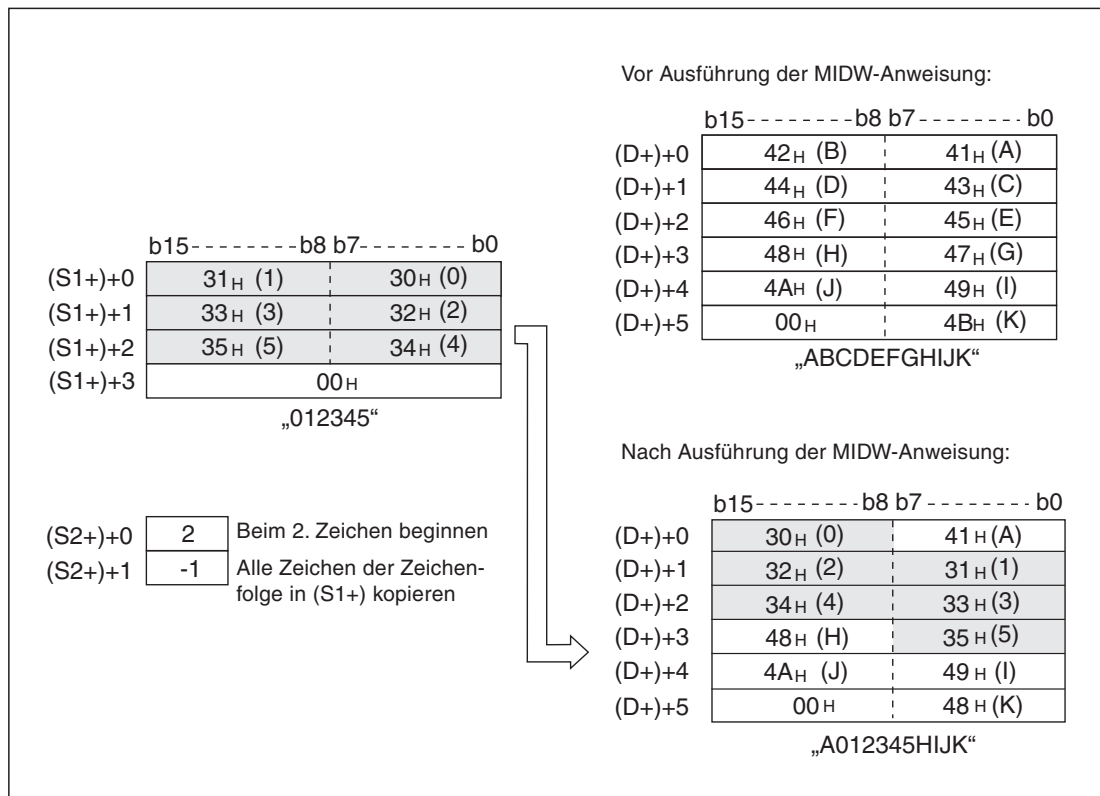


Abb. 7-237: Beim Wert „-1“ in ((S2+)+1) werden alle Zeichen der Zeichenfolge in (S1+) in die in (D+) angegebene Zeichenfolge ab dem in ((S2+)+0) angegebenen Zeichen kopiert.

HINWEIS

Die Angabe der Anzahl der Zeichen in $((S2+)+1)$ erfolgt in der Einheit „Byte“. Dies muss berücksichtigt werden, falls Zeichen gelesen werden sollen, die nicht im ASCII-Code gespeichert sind und mehr als ein Byte belegen. Werden z. B. Zeichen im JIS-Code erfasst werden sollen (1 Zeichen = 16 Bit), muss die Anzahl der gewünschten Zeichen mit 2 multipliziert und dieser Wert in $((S2+)+1)$ eingetragen werden.

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Die Angabe in $((S2+)+1)$ für die Anzahl der zu kopierenden Zeichen ist größer als die Anzahl der ab $(S1+)$ gespeicherten Zeichen.
- Mit der Angabe in $((S2+)+0)$ wird auf die Position eines Zeichens verwiesen, das in der ab $(D+)$ gespeicherten Zeichenfolge nicht vorhanden ist.
- Die Anzahl der Zeichen in $((S2+)+1)$ ist größer als die Länge der Zeichenfolge in $(S1+)$.
- Für $((S2+)+0)$ wurde eine negative Zahl angegeben.
- Für $((S2+)+1)$ wurde ein kleinerer Wert als -1 angegeben.
- Innerhalb des Operandenbereichs der in $(S1+)$ und $(D+)$ angegebenen Operanden ist nicht der Code „00H“ eingetragen.

Beispiel ▾

Beim Einschalten des Eingangs X010 werden aus der Zeichenfolge, die ab D0 gespeichert ist, die ersten vier Zeichen gelesen und ab dem 3. Zeichen in die Zeichenfolge eingetragen, die ab D100 gespeichert ist. Die Position des ersten zu überschreibenden Zeichens wird in R0 und die Anzahl der Zeichen in R1 angegeben.

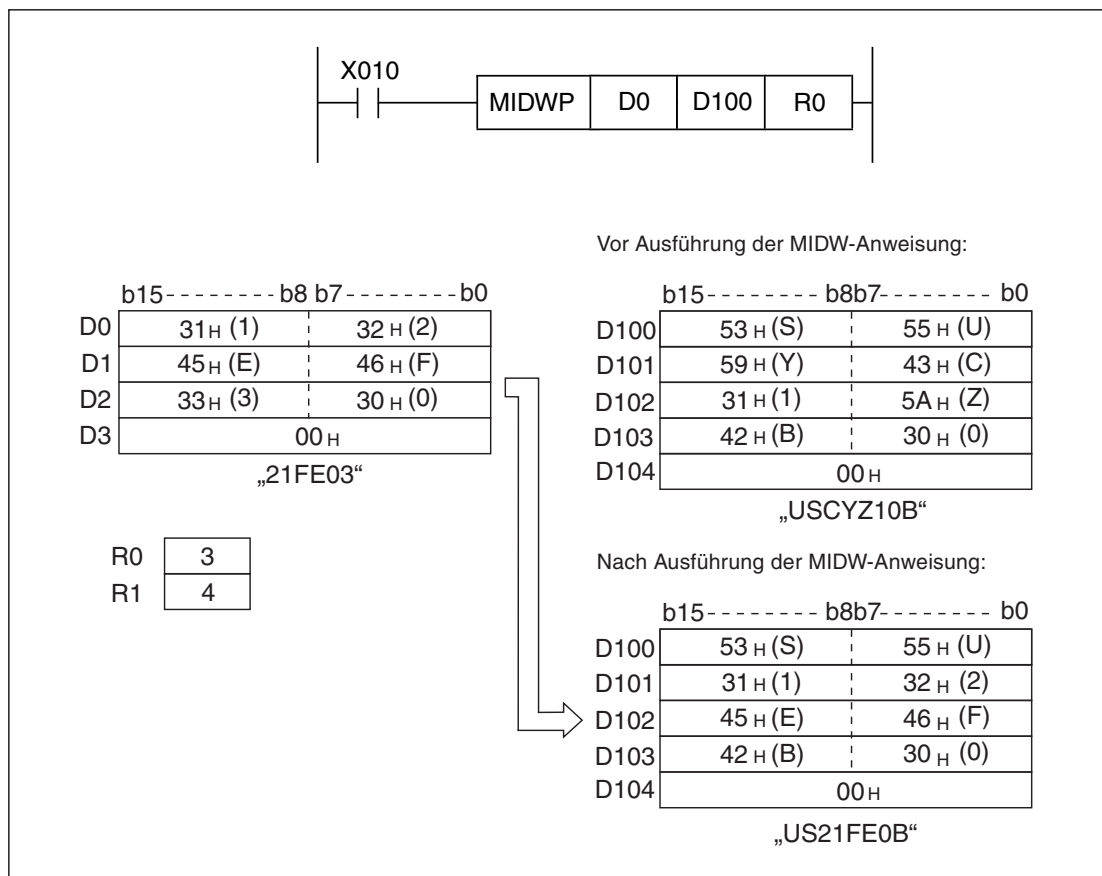


Abb. 7-238: In diesem Beispiel werden in der Zeichenfolge ab D100 das 3. bis 6. Zeichen ersetzt.



7.16.9 Zeichenfolge suchen (INSTR)

					INSTR		FNC 208				
					Zeichenfolge suchen						
					CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
Operanden	S1+	S2+	D+	S2+	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	T, C, D, R, Zeichenfolge	T, C, D, R		D, R, K, H	●		16 Bit	32 Bit	INSTR INSTRP	9	

Funktion

Zeichen in einer Zeichenfolge suchen

Beschreibung

- Die INSTR-Anweisung sucht in der Zeichenfolge ab (S2+) die in (S1+) angegebene Zeichenfolge.
- Die Suche beginnt an der mit (n) angegebenen Position.
- In (D+) wird die Position eingetragen, an der die gesuchte Zeichenfolge gefunden wurde. Angegeben wird die Nummer des Zeichens. Die Zählung beginnt am Anfang (linker Teil) der Zeichenfolge. (Das erste Zeichen der durchsuchten Zeichenfolge ist im niederwertigen Byte von ((S2+)+0) gespeichert.)

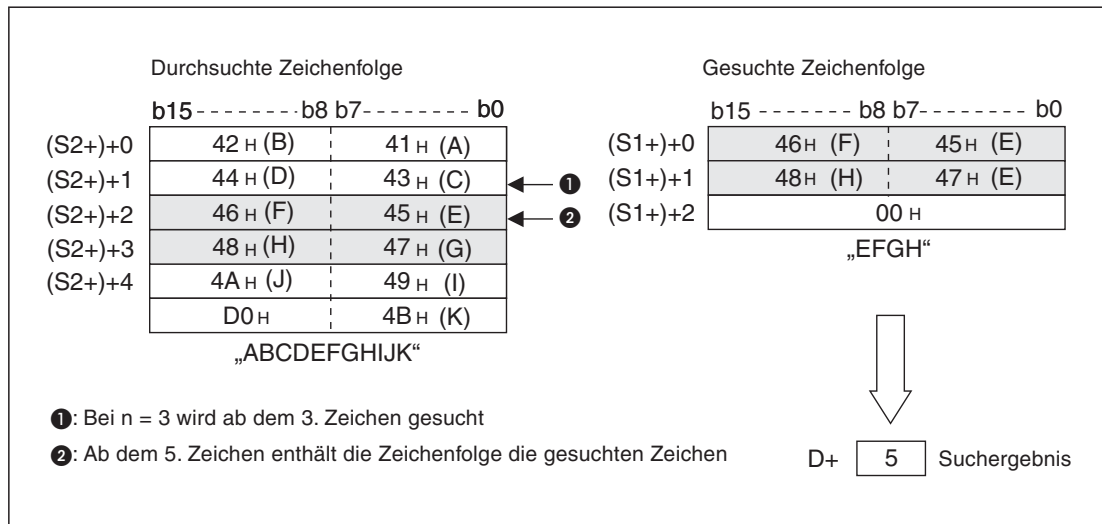


Abb. 7-239: Das Suchergebnis zeigt die Position des ersten Zeichens der gesuchten Zeichenfolge.

- Wenn die gesuchte Zeichenfolge nicht gefunden wurde, wird in (D+) eine Null gespeichert.
- Wenn der in (n) angegebene Wert negativ oder Null ist, wird die INSTR-Anweisung nicht ausgeführt.

- In (S1+) kann auch eine Zeichenfolge direkt angegeben werden:

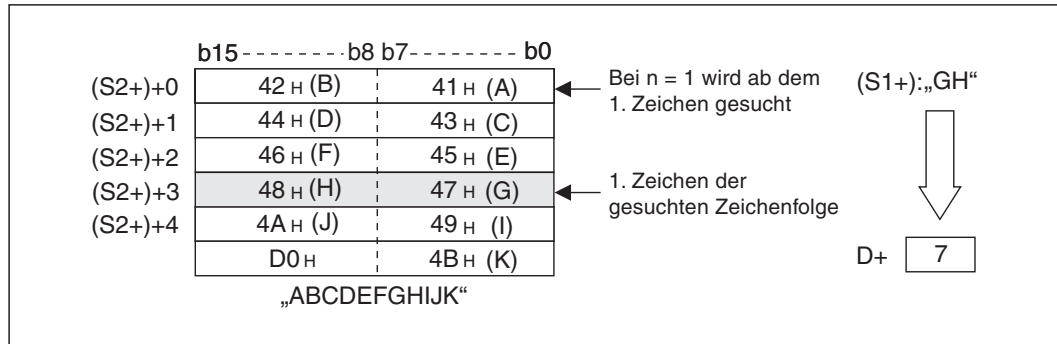


Abb. 7-240: In diesem Beispiel ist die direkt angegebene Zeichenfolge „GH“ ab dem 7. Zeichen gespeichert.

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Die Angabe in (n) für den Beginn der Suche ist größer als die Anzahl der ab ((S1+)+0) gespeicherten Zeichen.
- Innerhalb des Operandenbereichs der in (S1+) und (S2+) angegebenen Operanden ist nicht der Code „00H“ eingetragen.

Beispiel ▾

Das folgende Programm sucht bei eingeschaltetem Eingang X0 in der Zeichenfolge, die ab R0 gespeichert ist, nach der in D0 bis D2 angegebenen Zeichenfolge. Die Suche beginnt ab dem 5. Zeichen der Zeichenfolge ab R0. Das Suchergebnis wird in D100 gespeichert.

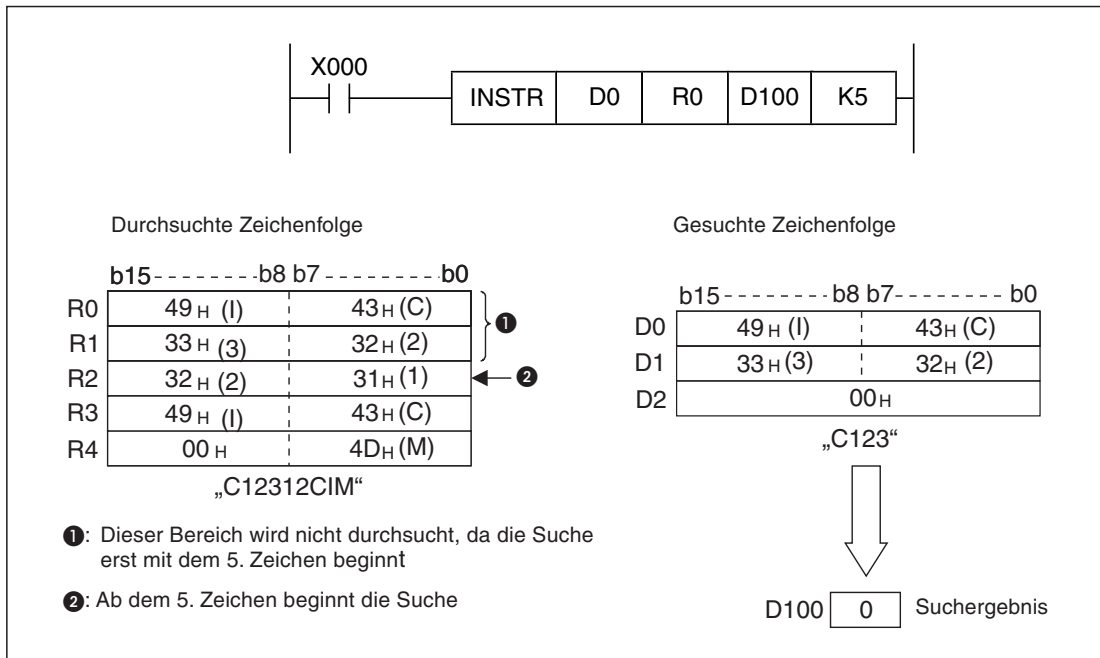


Abb. 7-241: In diesem Beispiel wurde die gesuchte Zeichenfolge nicht gefunden. Das Suchergebnis ist daher „0“.



7.16.10 Zeichenfolge transferieren (\$MOV)

		\$MOV		FNC 209				
		Zeichenfolgen transferieren						
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
								●
Operanden	S+	D+	Puls-Anweisung (P)		Verarbeitung		Programmschritte	
	KnX, KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□)	KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□)	●	16 Bit	32 Bit	\$MOV \$MOV P	5	

Funktion

Übertragen von Zeichenfolgen

Beschreibung

- Die \$MOV-Anweisung überträgt die in (S+) gespeicherten Bytes der Zeichenfolge nach (D+). Bei diesem Transfer wird die gesamte Zeichenfolge, beginnend bei dem ersten Zeichen (Byte) bis zu dem mit „00H“ beschriebenen Byte (Ende der Zeichenfolge) in einem Arbeitsgang übertragen.

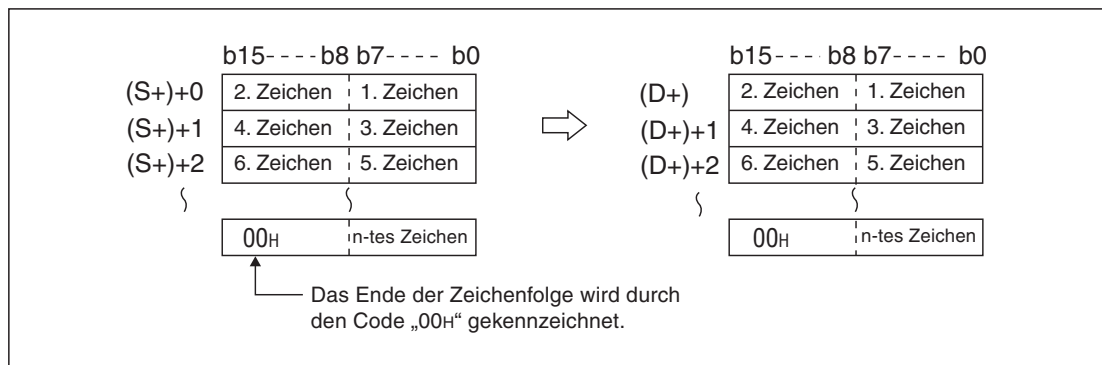


Abb. 7-242: Prinzip der Datenübertragung mit einer \$MOV-Anweisung

- Die \$MOV-Anweisung wird auch ausgeführt, wenn die für die Speicherung vorgesehenen Datenbereiche von ((S+)+0) bis ((S+)+n) mit denen von ((D+)+0) bis ((D+)+n) überlappen. Das folgende Ergebnis tritt auf, wenn die in D10 bis D13 gespeicherte Zeichenfolge nach D11 bis D14 übertragen wird.

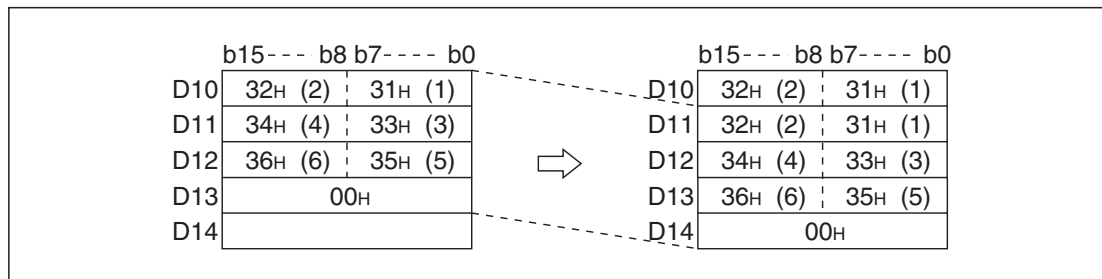


Abb. 7-243: In diesem Beispiel wird der Inhalt von D10 nicht verändert.

- Befindet sich in der Zeichenfolge der Code „00H“ im niederwertigen Byte von ((S+)+n), so wird das darauffolgende Zeichen bei der Übertragung nicht berücksichtigt und „00H“ in beiden Bytes von ((D+)+n) eingetragen.

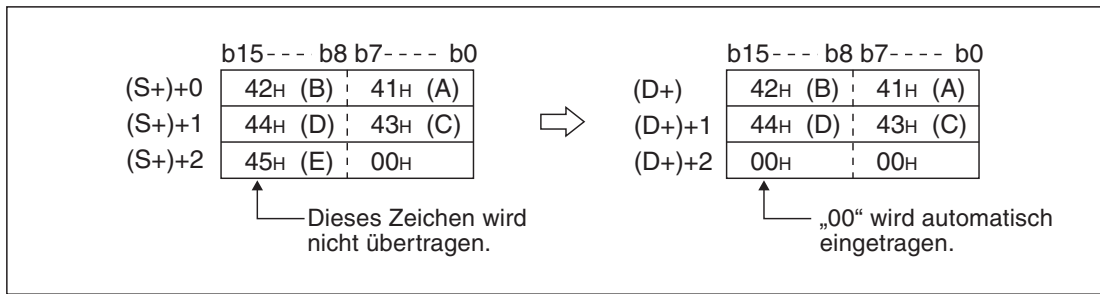


Abb. 7-244: Alle Zeichen nach dem Code „00“ werden ignoriert.

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Die vollständige Zeichenfolge kann nicht nach (D+) übertragen werden.
- Innerhalb des Operandenbereichs des in (S+) angegebenen Operanden ist nicht der Code „00H“ eingetragen.

Beispiel ▾

Das folgende Programm überträgt bei eingeschaltetem Eingang X0 die ab D10 bis D12 abgelegte Zeichenfolge in die Datenregister ab D20.

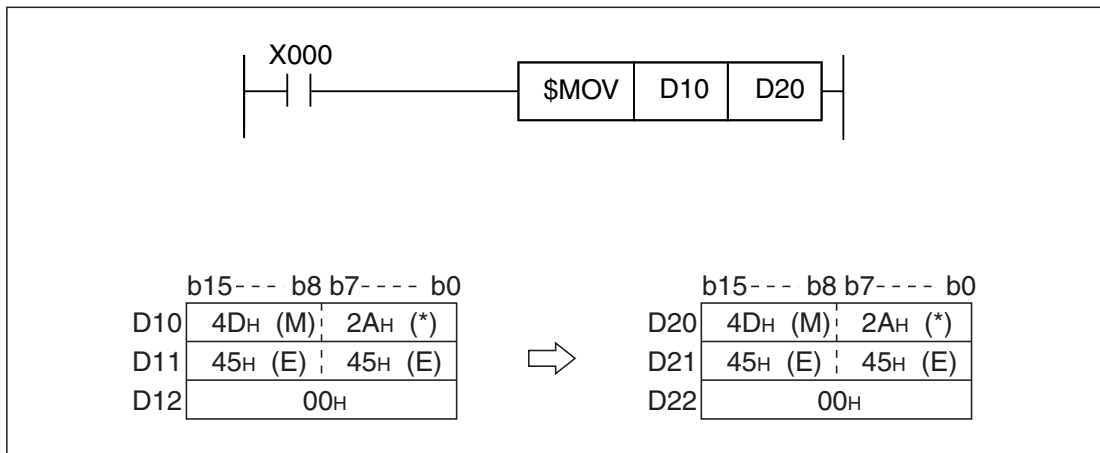


Abb. 7-245: Programmbeispiel zur Anwendung einer \$MOV-Anweisung



7.17 Verarbeitungsanweisungen für Datenlisten

Symbol	FNC	Bedeutung	Abschnitt
FDEL	210	Daten aus Datenliste löschen	7.17.1
FINS	211	Daten in Datenliste einfügen	7.17.2
POP	212	Daten lesen, die zuletzt in eine Datenliste eingetragen wurden	7.17.3
SFR	213	16-Bit-Datenwort nach rechts verschieben	7.17.4
SFL	214	16-Bit-Datenwort nach links verschieben	7.17.5

Tab. 7-64: Übersicht der Anweisungen zur Datenverarbeitung

7.17.1 Daten aus Datenliste löschen (FDEL)

				FDEL		FNC 210				
				Daten aus Liste löschen						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
										●
Operanden	S+	D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	T, C, D, R	T, C, D, R	D, R, K, H	●	16 Bit	32 Bit	FDEL FDELP	7		

Funktion

Löschen von Datenblöcken aus einer Datenliste

Beschreibung

- Die FDEL-Anweisung löscht die n-ten Daten in der in (D+) angegebenen Datenliste und speichert den Inhalt des gelöschten Blocks in dem in (S+) angegebenen Operanden.
- Die Daten der Datenliste werden nach dem Löschen der Daten zusammengeschoben. ((D+)+0) enthält die Anzahl der Elemente der Datenliste. Nach dem Löschen wird dieser Wert um 1 vermindert. Ab ((D+)+1) folgt die entsprechende Anzahl Daten.

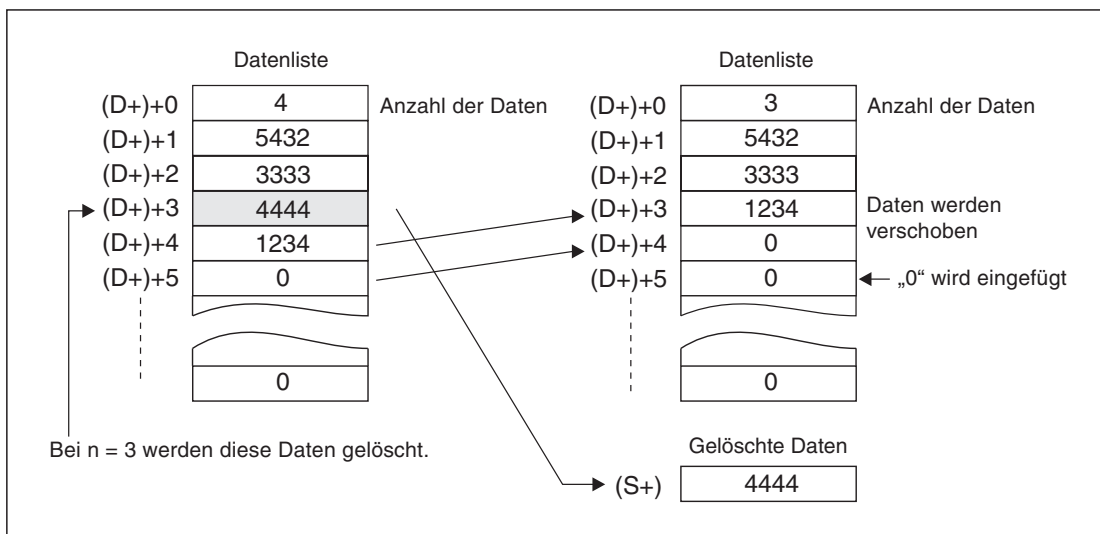


Abb. 7-246: Durch eine FDEL-Anweisung werden Daten einer Datenliste gelöscht und verschoben.

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Die mit (n) angegebene Position ist größer als die in ((D+)+0) angegebene Anzahl der Elemente der Datenliste.
- Der durch (n) angegebene Operand überschreitet den Bereich des in (D+) angegebenen Operanden.
- Der in (n) angegebene Wert ist negativ oder „0“.
- Die in ((D+)+0) angegebene Anzahl der Elemente der Datenliste ist „0“.
- Die Länge der Datenliste überschreitet den Operandenbereich.

Beispiel ▾

Das folgende Programm löscht mit positiver Flanke von X20 das zweite Element aus der Datenliste, die in D100 bis D107 gespeichert ist. Die gelöschten Daten werden in D0 gespeichert. Die FDEL-Anweisung wird nur ausgeführt, wenn mindestens 1 Element oder maximal 7 Elemente in der Datenliste vorhanden sind.

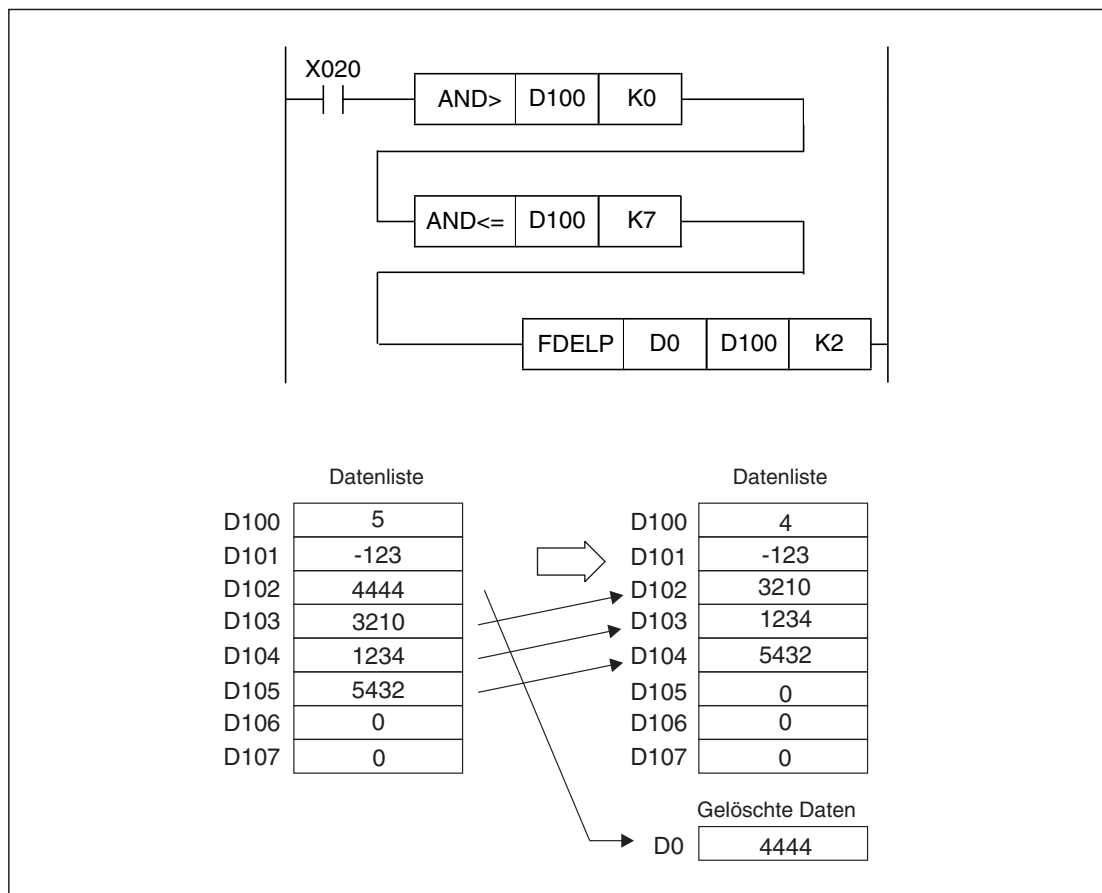


Abb. 7-247: Programmbeispiel zur FDEL-Anweisung



7.17.2 Daten in Datenliste einfügen (FINS)

				FINS		FNC 211				
				Daten in Liste einfügen						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
										●
Operanden	S+	D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	T, C, D, R, K, H	T, C, D, R	D, R, K, H	●	16 Bit	32 Bit	FINS FINSP	7		

Funktion

Löschen von Datenblöcken aus einer Datenliste

Beschreibung

- Die FINS-Anweisung fügt die in (S+) angegebenen 16-Bit-Daten an der n-ten Stelle in der in (D+) angegebenen Datenliste ein.
- Die auf die Einfügestelle folgenden Daten werden um eine Adresse weiterverschoben. ((D+)+0) enthält die Anzahl der Elemente der Datenliste. Nach dem Einfügen wird dieser Wert um 1 erhöht. Ab ((D+)+1) folgt die entsprechende Anzahl Daten.

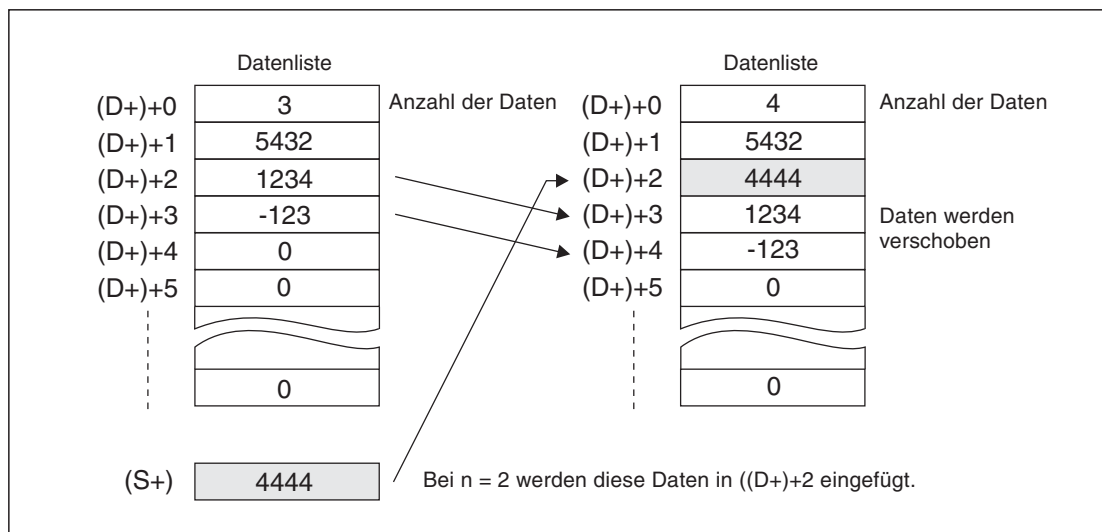


Abb. 7-248: Durch eine FINS-Anweisung werden Daten einer Datenliste eingefügt. Vorhandene Daten werden verschoben.

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Die mit (n) angegebene Position ist größer als die in ((D+)+0) angegebene Anzahl der Elemente der Datenliste.
- Der durch (n) angegebene Operand überschreitet den Bereich des in (D+) angegebenen Operanden.
- Der in (n) angegebene Wert ist negativ oder „0“.
- Die in ((D+)+0) angegebene Anzahl der Elemente der Datenliste ist „0“.
- Die Länge der Datenliste überschreitet den Operandenbereich.

Beispiel ▾

Das folgende Programm fügt mit positiver Flanke von X10 die in D100 enthaltenden Daten an der 3. Position der in D0 bis D7 gespeicherten Datenliste ein. Die FINS-Anweisung wird nicht ausgeführt, wenn schon 7 Elemente in der Datenliste vorhanden sind.

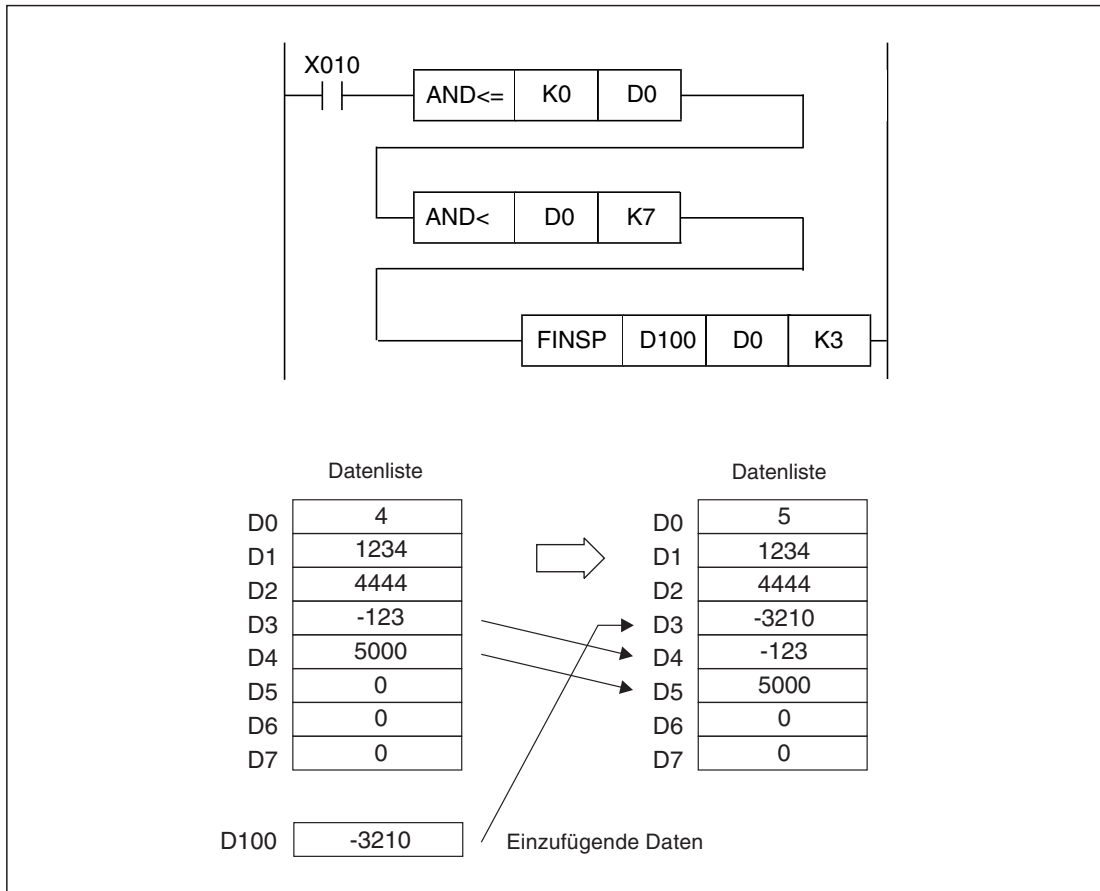
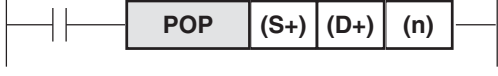


Abb. 7-249: Programmbeispiel zur FINS-Anweisung



7.17.3 Daten lesen, die zuletzt in eine Datenliste eingetragen wurden (POP)

				POP		FNC 212				
				Zuletzt gespeicherte Daten aus Liste lesen						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
										●
Operanden	S+	D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□)	KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□), V, Z	K, H	●	16 Bit	32 Bit	POP POPP	7		

Funktion

Daten lesen, die zuletzt mit einer SFWR-Anweisung (Abschnitt 6.5.8) in einen FIFO-Speicher eingetragen wurden

Beschreibung

- In (S+) wird die erste Adresse des Bereichs angegeben, in dem die Datenliste gespeichert ist. ((S+)+0) enthält einen Datenzeiger, der die Anzahl der Elemente der Datenliste angibt. Ab ((S+)+1) folgt die entsprechende Anzahl Daten.

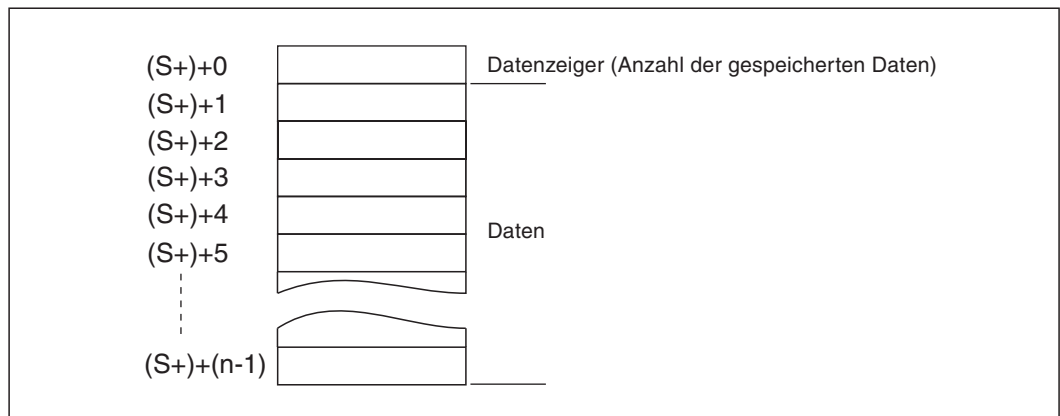


Abb. 7-250: Struktur einer Datenliste

- Die aus der Datenliste gelesenen 16-Bit-Daten werden in (D+) gespeichert.
- In (n) wird die Länge der Datenliste angegeben. Der Wert in (n) kann zwischen 2 und 512 liegen. (Zur Anzahl der in ((S+)+0) gespeicherten Daten muss 1 addiert werden, weil ((S+)+0) mitgezählt wird.)

- Nach dem Lesen der Daten wird der Inhalt des Datenzeigers um 1 vermindert. Der Inhalt der Datenliste wird durch die POP-Anweisung nicht verändert.

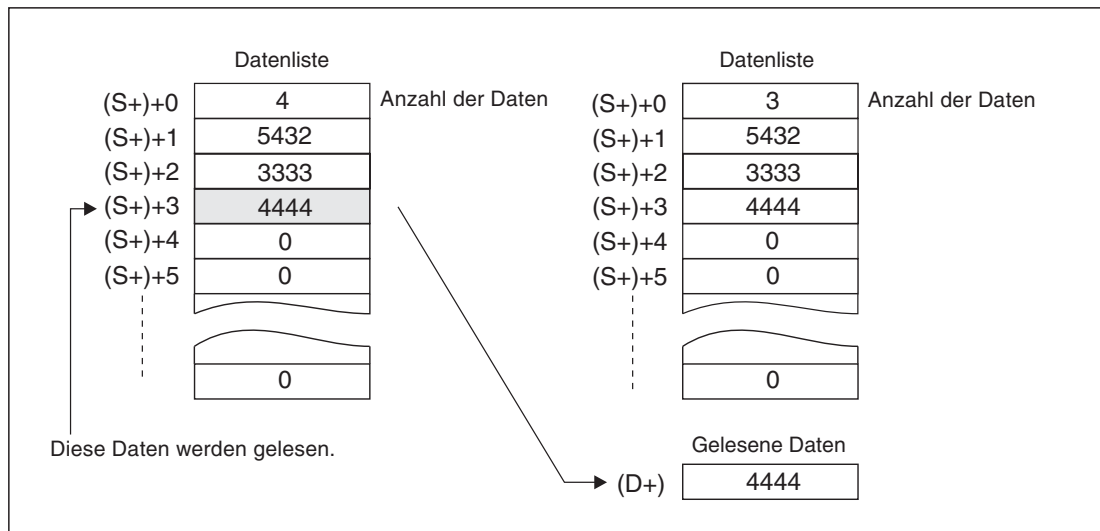


Abb. 7-251: Beim Lesen der Daten wird der Inhalt der Datenliste nicht verändert.

HINWEISE

Wird eine POP-Anweisung zyklisch ausgeführt, erreicht der Datenzeiger nach einigen Programmzyklen den Wert „0“. Verwenden Sie deshalb die impulsgesteuerte Variante der POP-Anweisung (POPP).

Wenn der Datenzeiger in ((S+)+0) den Wert „0“ erreicht, wird der Sondermerker M8020 gesetzt. In diesem Fall wird die POP-Anweisung nicht ausgeführt. Prüfen Sie vor der Ausführung der POP-Anweisung mit einer Vergleichsanweisung, ob der Datenzeiger in ((S+)+0) mindestens den Wert „1“ und maximal den Wert „n-1“ aufweist.

Hat der Datenzeiger in ((S+)+0) den Wert „1“, weist er nach Ausführung der POP-Anweisung den Wert „0“ auf und der Sondermerker M8020 wird gesetzt.

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Der Wert in ((S+)+0) ist größer als „n-1“.
- Der Wert in ((S+)+0) ist negativ.

Beispiel ▾

Im folgenden Programm wird beim Einschalten von X020 der Inhalt von D20 in die in D100 bis D106 gespeicherte Datenliste eingetragen. (Bei jeder Ausführung der SFWR-Anweisung werden Daten in das nächste freie Datenregister eingetragen und der Inhalt von D100 um 1 erhöht).

Wird der Eingang X021 eingeschaltet, wird der Wert gelesen, der zuletzt in die Datenliste eingetragen wurde, und in D10 gespeichert. Der Inhalt von D100 wird bei jeder Ausführung der POP-Anweisung um 1 vermindert.

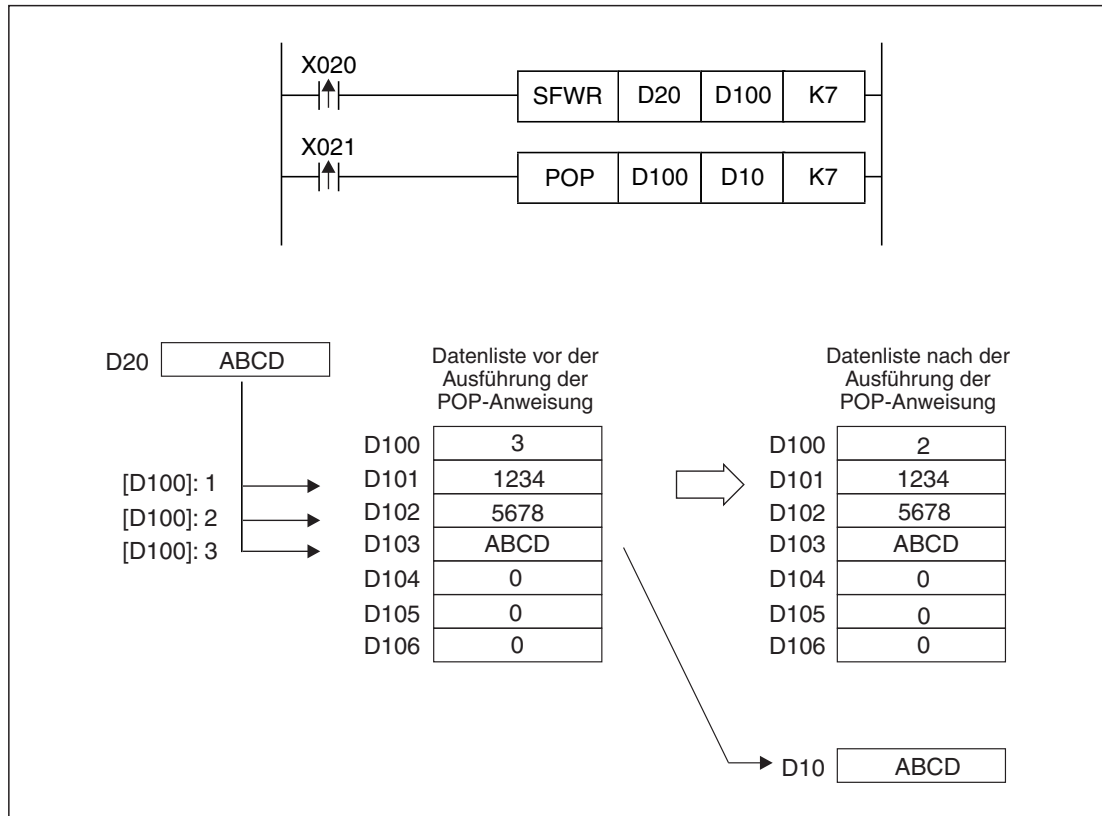


Abb. 7-252: Die Daten, die mit einer SFWR-Anweisung in eine Datenliste eingetragen wurden, können mit einer POP-Anweisung extrahiert werden.



7.17.4 16-Bit-Datenwort nach rechts verschieben (SFR)

		SFR		FNC 213				
		Daten nach rechts verschieben (mit Übertrag)						
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
								●
Operanden	D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□), V, Z	KnX, KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□), V, Z	●	16 Bit	32 Bit	SFR SFRP	5	

Funktion

Verschiebung eines 16-Bit-Datenwortes um n Bit nach rechts

Beschreibung

- Die SFR-Anweisung verschiebt das in (D+) vorgegebene 16-Bit-Datenwort bitweise um (n) Bits nach rechts.
- Für (n) kann ein Wert zwischen 0 und 15 angegeben werden. Wird für (n) ein größerer Wert als 15 angegeben, ergibt sich die Verschiebung aus dem Rest der Division „n/16“. Ist z. B. n = 18, werden die Daten um 2 Bits verschoben (18/16 = 1, Rest 2).
- Die höchstwertigen n Bits werden, beginnend mit Bit 15, auf 0 gesetzt.
- Das n-te zu verschiebende Bit (Bit (n-1)) wird in das Carry Flag M8022 geschoben

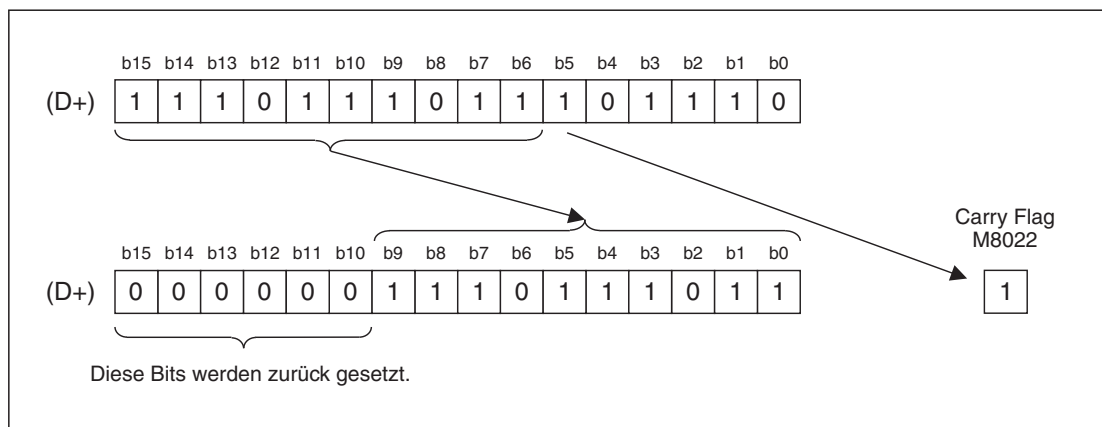


Abb. 7-253: Bei n = 6 werden die Daten um 6 Stellen nach rechts verschoben

- Es können auch Bit-Operanden verschoben werden, die durch einen K-Faktor zusammengefasst worden sind.

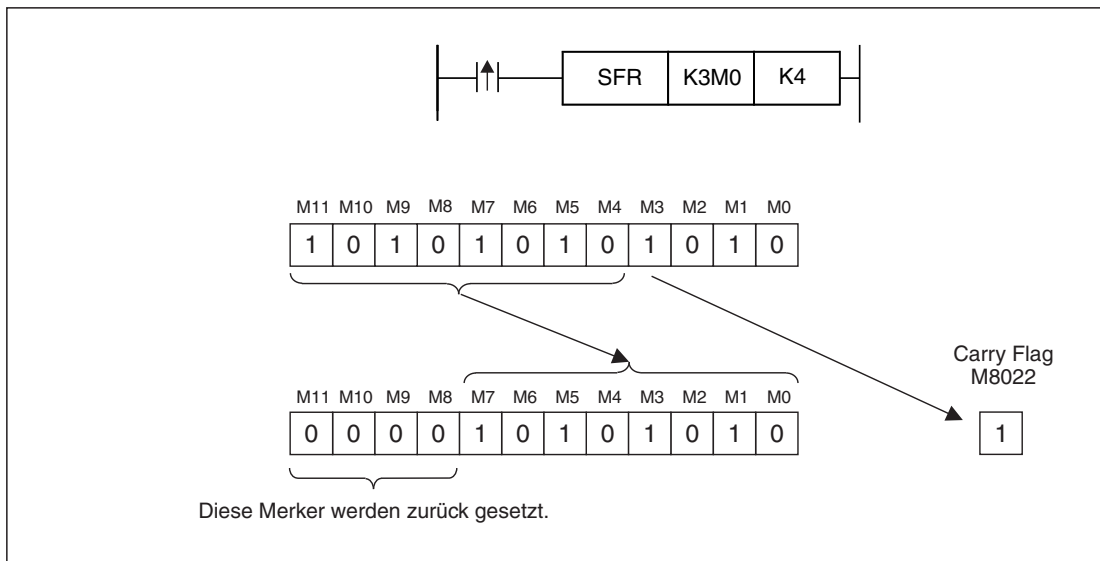


Abb. 7-254: In diesem Beispiel wird der Inhalt der Merker M0 bis M11 verschoben.

Fehlerquellen

Im folgenden Fall tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Der Wert in (n) ist negativ.

Beispiel ▾

Im folgenden Programm werden mit der steigenden Flanke von X20 die Zustände von Y10 bis Y1B um die in D0 angegebene Anzahl Bits nach rechts geschoben. Der Zustand von Y13 wird im Carry Flag gespeichert.

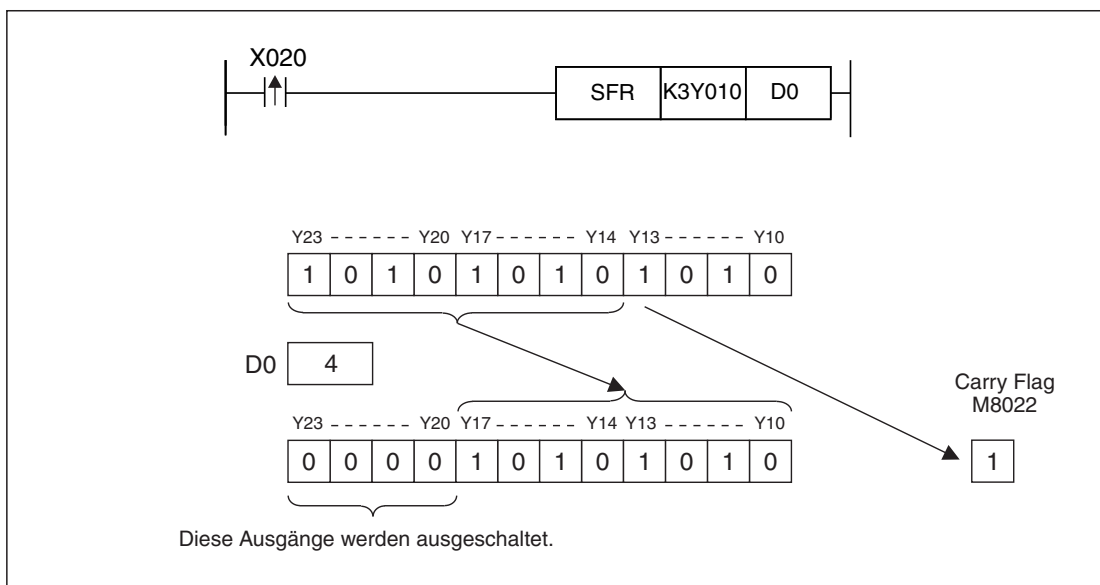


Abb. 7-255: Programmbeispiel zur SFR-Anweisung; Die Daten werden um 4 Bit nach rechts verschoben.



7.17.5 16-Bit-Datenwort nach links verschieben (SFL)

		SFL		FNC 214			
		Daten nach links verschieben (mit Übertrag)					
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S FX3UC
							●
Operanden	D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte	
	KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□), V, Z	KnX, KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□), V, Z	●	16 Bit	32 Bit	SFL SFLP	5

Funktion

Verschiebung eines 16-Bit-Datenwortes um n Bit nach links

Beschreibung

- Die SFL-Anweisung verschiebt das in (D+) vorgegebene 16-Bit-Datenwort bitweise um (n) Bits nach links.
- Für (n) kann ein Wert zwischen 0 und 15 angegeben werden. Wird für (n) ein größerer Wert als 15 angegeben, ergibt sich die Verschiebung aus dem Rest der Division „n/16“. Ist z. B. n = 18, werden die Daten um 2 Bits verschoben (18/16 = 1, Rest 2).
- Die niedrigstwertigen n Bits werden, beginnend mit Bit 0, auf 0 gesetzt.
- Das (n+1)-te zu verschiebende Bit (Bit n) wird in das Carry Flag M8022 geschoben.

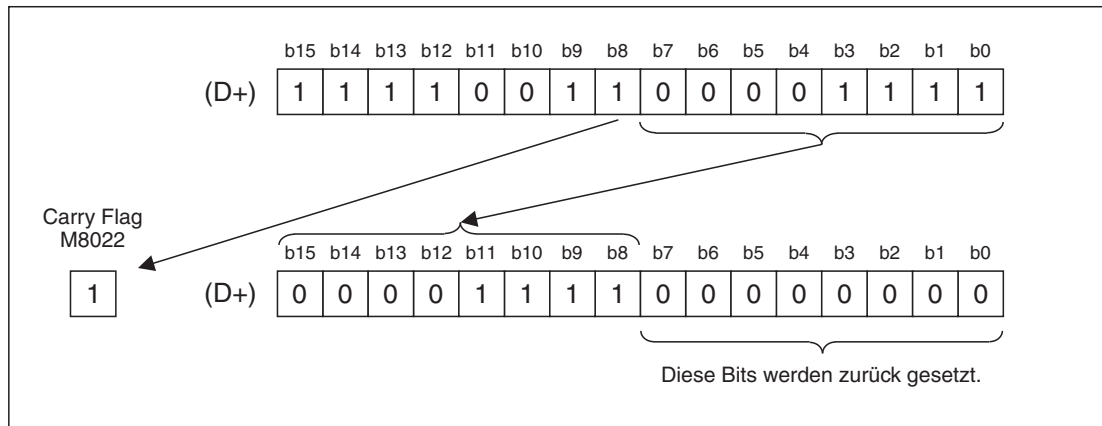


Abb. 7-256: Bei n = 8 werden die Daten um 8 Stellen nach links verschoben

- Es können auch Bit-Operanden verschoben werden, die durch einen K-Faktor zusammengefasst worden sind.

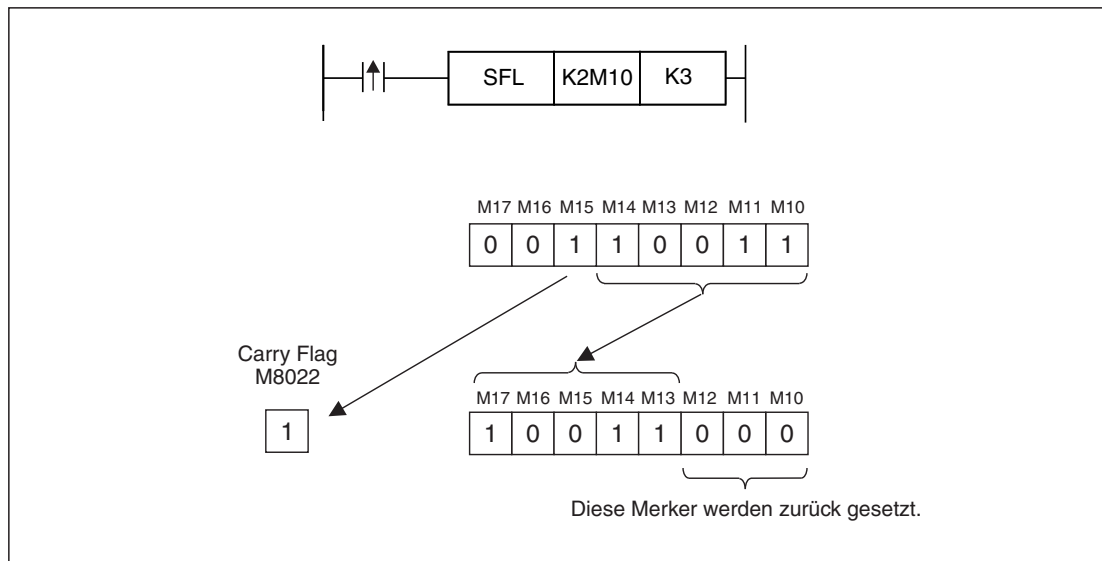


Abb. 7-257: In diesem Beispiel wird der Inhalt der Merker M10 bis M17 verschoben.

Fehlerquellen

Im folgenden Fall tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Der Wert in (n) ist negativ.

Beispiel ▾

Das folgende Programm verschiebt die Zustände der Ausgänge Y10 bis Y17 mit der steigenden Flanke von X20 um 3 Bits nach links. Der Zustand von Y15 wird im Carry Flag gespeichert.

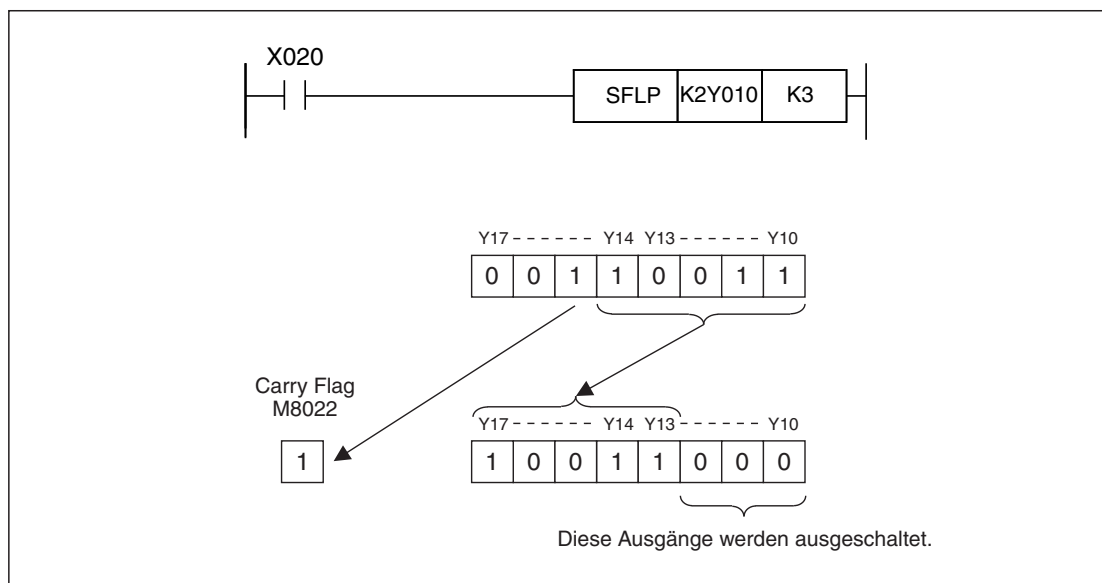


Abb. 7-258: Programmbeispiel zur SFL-Anweisung; Die Daten werden um 3 Bit nach links verschoben.



7.18 Vergleichsanweisungen (2)

Übersicht der Anweisungen FNC 224 bis 246

Symbol	FNC	Bedeutung	Abschnitt
LD =	224	Vergleichsanweisung, gleich	7.18.1
LD >	225	Vergleichsanweisung, größer	
LD <	226	Vergleichsanweisung, kleiner	
LD <>	228	Vergleichsanweisung, ungleich	
LD ≤	229	Vergleichsanweisung, kleiner gleich	
LD ≥	230	Vergleichsanweisung, größer gleich	
AND =	232	UND-verknüpfte Vergleichsanweisung, gleich	7.18.2
AND >	233	UND-verknüpfte Vergleichsanweisung, größer	
AND <	234	UND-verknüpfte Vergleichsanweisung, kleiner	
AND <>	236	UND-verknüpfte Vergleichsanweisung, ungleich	
AND ≤	237	UND-verknüpfte Vergleichsanweisung, kleiner gleich	
AND ≥	238	UND-verknüpfte Vergleichsanweisung, größer gleich	
OR =	240	ODER-verknüpfte Vergleichsanweisung, gleich	7.18.3
OR >	241	ODER-verknüpfte Vergleichsanweisung, größer	
OR <	242	ODER-verknüpfte Vergleichsanweisung, kleiner	
OR <>	244	ODER-verknüpfte Vergleichsanweisung, ungleich	
OR ≤	245	ODER-verknüpfte Vergleichsanweisung, kleiner gleich	
OR ≥	246	ODER-verknüpfte Vergleichsanweisung, größer gleich	

Tab. 7-65: Übersicht der Vergleichsanweisungen

7.18.1 Lade Vergleiche (LD□)

		LD□		FNC 224 – 230					
		Lade Vergleiche							
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
			●	●	●	●	●	●	
Operanden	S1+	S2+	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , U□/G□ ^② , V, Z				16 Bit	32 Bit	LD□	5	
				●	●	DLD□	9		

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktion

Laden der Vergleichsergebnisse LD□

Beschreibung

- Mit der Anweisung erfolgt das Laden der Vergleichsergebnisse der ab (S1+) und (S2+) angegebenen Werte.
- Ist das Vergleichsergebnis wahr, ist das Verknüpfungsergebnis „1“.
- Ist das Vergleichsergebnis falsch, ist das Verknüpfungsergebnis „0“.

- Das □ in der LD□-Anweisung steht als Platzhalter für die Vergleichsoperationen =, >, <, <>, ≤ und ≥. Die folgende Tabelle gibt die Zuordnung der Vergleichsoperationen zu den entsprechenden Funktionsnummern wieder.

FNC Nummer	Anweisungen		Wahr wenn	Falsch wenn
	16 Bit	32 Bit		
224	LD =	DLD =	(S1+) = (S2+)	(S1+) <> (S2+)
225	LD >	DLD >	(S1+) > (S2+)	(S1+) ≤ (S2+)
226	LD <	DLD <	(S1+) < (S2+)	(S1+) ≥ (S2+)
228	LD <>	DLD <>	(S1+) <> (S2+)	(S1+) = (S2+)
229	LD ≤	DLD ≤	(S1+) ≤ (S2+)	(S1+) > (S2+)
230	LD ≥	DLD ≥	(S1+) ≥ (S2+)	(S1+) < (S2+)

Tab. 7-66: Übersicht der LD□-Anweisungen

HINWEIS | Die LD□-Anweisung kann wie eine LD-Anweisung verwendet werden.

Beispiel ▽

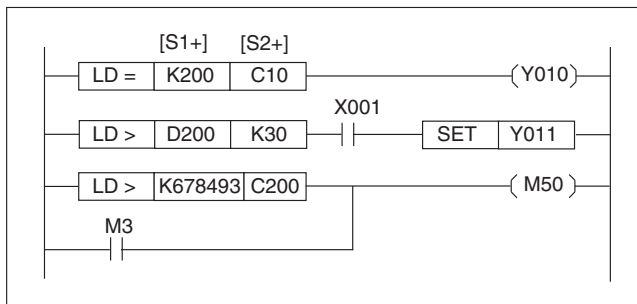


Abb. 7-259:
 Programmierbeispiel zu den LD□-Anweisungen

Ist der Wert K200 gleich dem Counter-Wert C10, wird der Ausgang Y10 eingeschaltet.

Ist der Wert in D200 größer als der Wert K30 und der Eingang X1 eingeschaltet, wird der Ausgang Y11 durch die SET-Anweisung gesetzt. Ist der Wert K678493 größer als der Zählwert des Counters C200 oder der Merker M3 eingeschaltet, wird der Merker M50 eingeschaltet.

△

7.18.2 UND-verknüpfte Vergleiche (AND□)

		AND□		FNC 230 – 238			
		UND-verknüpfte Vergleiche					
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
	●	●	●	●	●	●	
Operanden	S1+	S2+	Puls-Anweisung (P)	Verarbeitung		Programmschritte	
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , U□/G□ ^② , V, Z			16 Bit	32 Bit	AND□	5
				●	●	DAND□	9

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktion

UND-Verknüpfen eines Vergleichsergebnisses

Beschreibung

- Mit der Anweisung erfolgt eine UND-Verknüpfung der Vergleichsergebnisse der ab (S1+) und (S2+) angegebenen Werte.
- Ist das Vergleichsergebnis wahr, ist das Verknüpfungsergebnis „1“.
- Ist das Vergleichsergebnis falsch, ist das Verknüpfungsergebnis „0“.
- Das □ in der AND□-Anweisung steht als Platzhalter für die Vergleichsoperationen =, >, <, <>, ≤ und ≥. Die folgende Tabelle zeigt die Zuordnung der Vergleichsoperationen zu den Funktionsnummern.

FNC Nummer	Anweisungen		Wahr wenn	Falsch wenn
	16 Bit	32 Bit		
232	AND =	DAND =	(S1+) = (S2+)	(S1+) <> (S2+)
233	AND >	DAND >	(S1+) > (S2+)	(S1+) ≤ (S2+)
234	AND <	DAND <	(S1+) < (S2+)	(S1+) ≥ (S2+)
236	AND <>	DAND <>	(S1+) <> (S2+)	(S1+) = (S2+)
237	AND ≤	DAND ≤	(S1+) ≤ (S2+)	(S1+) > (S2+)
238	AND ≥	DAND ≥	(S1+) ≥ (S2+)	(S1+) < (S2+)

Tab. 7-67: Übersicht der AND□-Anweisung

HINWEIS | Die AND□-Anweisung kann wie eine AND-Anweisung verwendet werden.

Beispiel ▾

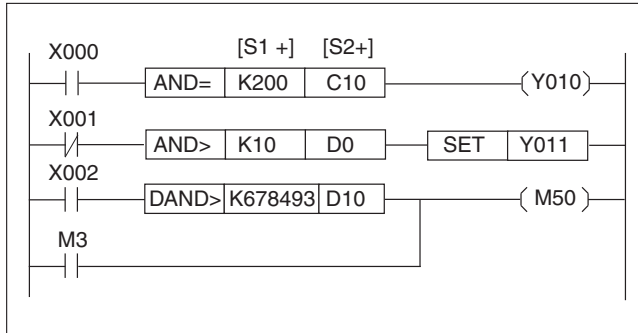


Abb. 7-260:
 Programmierbeispiel zu den AND□-
 Anweisungen

Ist der Wert K200 gleich dem Zählwert des Counters C10 und der Eingang X0 eingeschaltet, wird der Ausgang Y10 eingeschaltet.

Ist der Wert K10 größer als der Wert in D0 und der Eingang X1 nicht eingeschaltet, wird der Ausgang Y11 durch die SET-Anweisung gesetzt.

Ist der Wert K678493 größer als der Wert in D10 und D11 und der Eingang X2 eingeschaltet, wird der Merker M50 eingeschaltet. Der Merker M50 wird auch eingeschaltet, wenn M3 eingeschaltet ist.



7.18.3 ODER-verknüpfte Vergleiche (OR□)

			OR□		FNC 240 – 246				
			ODER-verknüpfte Vergleiche						
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC			
	●	●	●	●		●			
Operanden	S1+	S2+		Puls-Anweisung (P)	Verarbeitung		Programmschritte		
	K, H, KnX, KnY, KnM, KnS, T, C, D, R ^① , U□/G□ ^② , V, Z				16 Bit	32 Bit	OR□	5	
					●	●	DOR□	9	

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktion

ODER-verknüpfen eines Vergleichsergebnisses

Beschreibung

- Mit der Anweisung erfolgt eine ODER-Verknüpfung der Vergleichsergebnisse der ab (S1+) und (S2+) angegebenen Werte.
- Ist das Vergleichsergebnis wahr, ist das Verknüpfungsergebnis „1“.
- Ist das Vergleichsergebnis falsch, ist das Verknüpfungsergebnis „0“.
- Das □ in der OR□-Anweisung steht als Platzhalter für die Vergleichsoperationen =, >, <, <>, ≤ und ≥. Die folgende Tabelle gibt die Zuordnung der Vergleichsoperationen zu den entsprechenden Funktionsnummern wieder.

FNC Nummer	Anweisungen		Wahr wenn	Falsch wenn
	16 Bit	32 Bit		
240	OR =	DOR =	(S1+) = (S2+)	(S1) <> (S2+)
241	OR >	DOR >	(S1+) > (S2+)	(S1) ≤ (S2+)
242	OR <	DOR <	(S1+) < (S2+)	(S1) ≥ (S2+)
244	OR <>	DOR <>	(S1+) <> (S2+)	(S1) = (S2+)
245	OR ≤	DOR ≤	(S1+) ≤ (S2+)	(S1) > (S2+)
246	OR ≥	DOR ≥	(S1+) ≥ (S2+)	(S1) < (S2+)

Tab. 7-68: Übersicht der OR□-Anweisung

HINWEIS

Die OR□-Anweisung kann wie eine OR-Anweisung verwendet werden.

Beispiel ▾

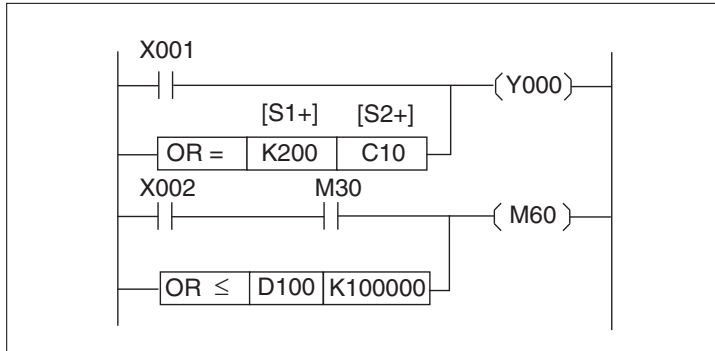


Abb. 7-261:
 Programmierbeispiel zu den
 OR□-Anweisungen

Ist der Wert K200 gleich dem Zählwert des Counters C10 oder der Eingang X1 eingeschaltet, wird der Ausgang Y0 eingeschaltet. Ist der Wert in D100 kleiner oder gleich dem Wert K100000 oder sind der Eingang X2 und der Merker M30 eingeschaltet, wird der Merker M60 eingeschaltet.

△

7.19 Datenkontrollanweisungen

Symbol	FNC	Bedeutung	Abschnitt
LIMIT	256	Ausgabebereich von Werten begrenzen	7.19.1
BAND	257	Eingangs-Offset festlegen	7.19.2
ZONE	258	Ausgangs-Offset festlegen	7.19.3
SCL	259	Werte skalieren	7.19.4
DABIN	260	Zahl im ASCII-Code in Binärwert wandeln	7.19.5
BINDA	261	Binärwert in ASCII-Code wandeln	7.19.6
SCL2	269	Werte skalieren	7.19.7

Tab. 7-69: Übersicht der Datenkontrollanweisungen

7.19.1 Ausgabebereich von Werten begrenzen (LIMIT)

					LIMIT		FNC 256				
					Begrenzung von Werten						
					CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
										●	
Operanden	S1+	S2+	S2+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	KnX, KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□), K, H	KnX, KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□)	KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□)	●		16 Bit	32 Bit	LIMIT LIMITP	9		
						●	●	DLIMIT DLIMITP	17		

Funktion

Begrenzung des Ausgangswertebereichs von 16- und 32-Bit-Binärdaten

Beschreibung für 16-Bit-Verarbeitung (LIMIT, LIMITP)

- Die LIMIT-Anweisung überprüft die im Operanden (S+) angegebenen Daten darauf, ob sie innerhalb des in (S1+) angegebenen unteren Grenzwertes bzw. des in (S2+) angegebenen oberen Grenzwertes liegen. Abhängig von dem Prüfergebnis werden die Werte wie folgt in dem in (D+) angegebenen Operanden gespeichert:
 - Wenn der in (S3+) angegebene Datenwert kleiner als der in (S1+) angegebene untere Grenzwert ist, wird der untere Grenzwert in dem in (D+) angegebenen Operanden gespeichert.
 - Wenn der in (S3+) angegebene Datenwert größer als der in (S2+) angegebene obere Grenzwert ist, wird der obere Grenzwert in dem in (D+) angegebenen Operanden gespeichert.
 - Wenn der in (S3+) angegebene Datenwert zwischen dem unteren und oberen Grenzwert liegt, wird der Wert in (S3+) in dem in d angegebenen Operanden gespeichert.
- Der in (S1+), (S2+) und (S3+) angegebene Wert kann zwischen -32768 und 32767 liegen.
- Wenn nur der obere Grenzwert kontrolliert werden soll, muss für den unteren Grenzwert in (S1+) der Wert -32768 eingetragen werden.
- Wenn nur der untere Grenzwert kontrolliert werden soll, muss für den oberen Grenzwert in (S2+) der Wert 32767 eingetragen werden.

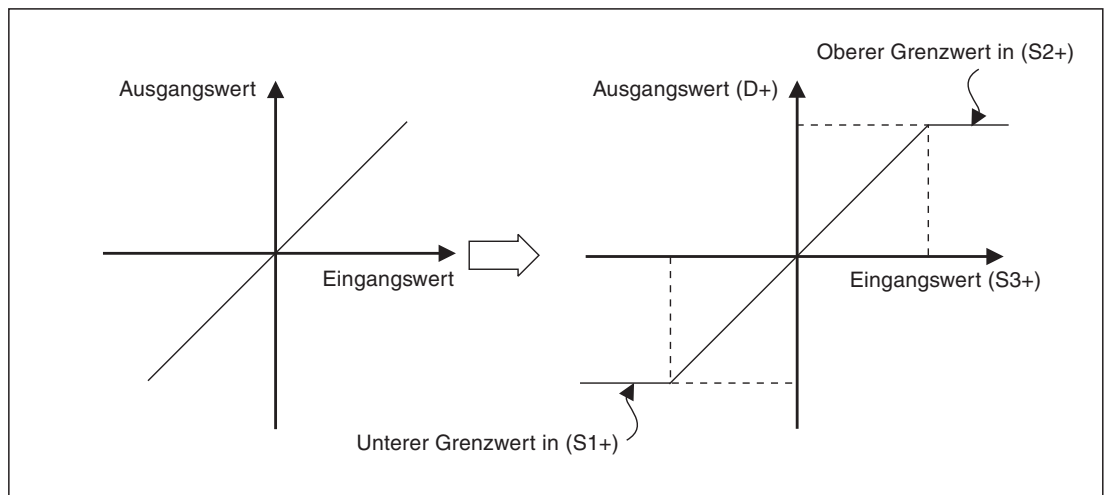


Abb. 7-262: Zusammenhang zwischen Ein- und Ausgangswert ohne (links) und Begrenzung durch die LIMIT-Anweisung (rechts)

Beschreibung für 32-Bit-Verarbeitung (DLIMIT, DLIMITP)

- Die DLIMIT-Anweisung überprüft die in den Operanden $((S3+)+1)$ und $((S3)+0)$ angegebenen Daten darauf, ob sie innerhalb des in $((S1+)+1)$ und $((S1)+0)$ angegebenen unteren Grenzwertes bzw. des in $((S2+)+1)$ und $((S2)+0)$ angegebenen oberen Grenzwertes liegen. Abhängig von dem Prüfergebnis werden die Werte wie folgt in den in $((D+)+1)$ und $((D+)+0)$ angegebenen Operanden gespeichert:
 - Wenn der in $((S3+)+1)$ und $((S3)+0)$ angegebene Datenwert kleiner als der in $((S1+)+1)$ und $((S1)+0)$ angegebene untere Grenzwert ist, wird der untere Grenzwert in den in $((D+)+1)$ und $((D+)+0)$ angegebenen Operanden gespeichert.
 - Wenn der in $((S3+)+1)$ und $((S3)+0)$ angegebene Datenwert größer als der in $((S2+)+1)$ und $((S2)+0)$ angegebene obere Grenzwert ist, wird der obere Grenzwert in den in $((D+)+1)$ und $((D+)+0)$ angegebenen Operanden gespeichert.
 - Wenn der in $((S3+)+1)$ und $((S3)+0)$ angegebene Eingangswert zwischen dem unteren und oberen Grenzwert liegt, wird der Eingangswert in den in $((D+)+1)$ und $((D+)+0)$ angegebenen Operanden gespeichert.
- Der in $((S1+)+1)$ und $((S1)+0)$, $((S2+)+1)$ und $((S2)+0)$ sowie in $((S3+)+1)$ und $((S3)+0)$ angegebene Wert kann zwischen -2147483648 und 2147483647 liegen.
- Wenn nur der obere Grenzwert kontrolliert werden soll, muss für den unteren Grenzwert in $((S1+)+1)$ und $((S1)+0)$ der Wert -2147483648 eingetragen werden.
- Wenn nur der untere Grenzwert kontrolliert werden soll, muss für den oberen Grenzwert in $((S2+)+1)$ und $((S2)+0)$ der Wert 2147483647 eingetragen werden.

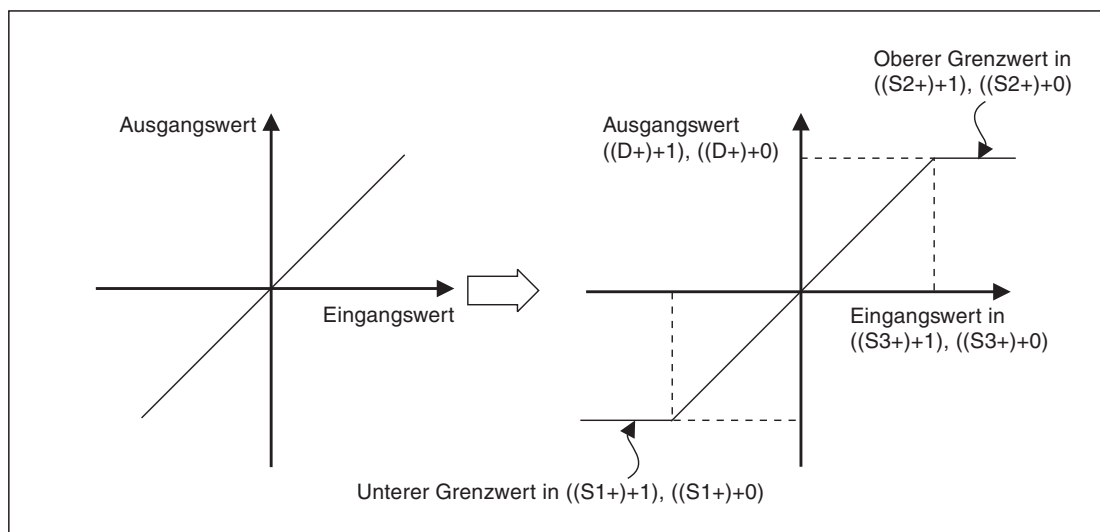


Abb. 7-263: Zusammenhang zwischen Ein- und Ausgangswert ohne (links) und Begrenzung durch die LIMIT-Anweisung (rechts)

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Bei der 16-Bit-Verarbeitung ist der in $(S1+)$ angegebene untere Grenzwert größer als der in $(S2+)$ angegebene obere Grenzwert.
- Bei der 32-Bit-Verarbeitung ist der in $((S1+)+1)$ und $((S1)+0)$ angegebene untere Grenzwert größer als der in $((S2+)+1)$ und $((S2)+0)$ angegebene obere Grenzwert.

Beispiel ▾

Im folgenden Programm wird der Wert der BCD-Daten an den Eingängen X20 bis X37 auf den unteren Wert „500“ und den oberen Wert „5000“ begrenzt. Der Ausgabewert wird in D1 gespeichert.

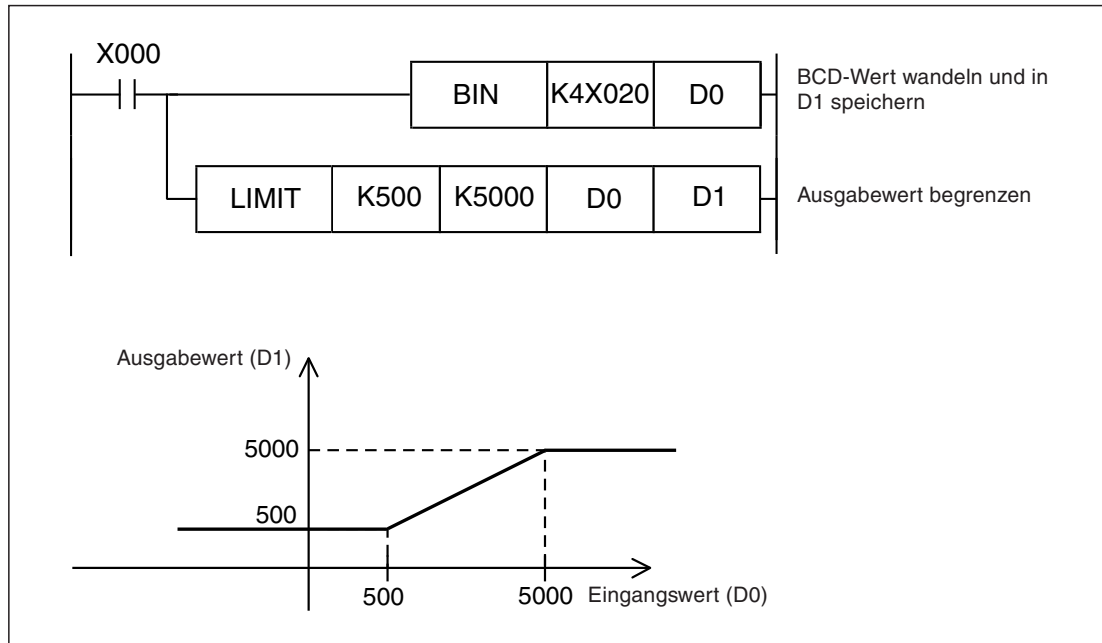


Abb. 7-264: Der Wert in D1 wird durch dieses Programm auf Werte zwischen 500 und 5000 begrenzt.



Beispiel ▾

Im folgenden Programm wird der Wert der BCD-Daten an den Eingängen X20 bis X57 auf den unteren Wert „10000“ und den oberen Wert „1000000“ begrenzt. Der Ausgabewert wird in D11 und D10 gespeichert.

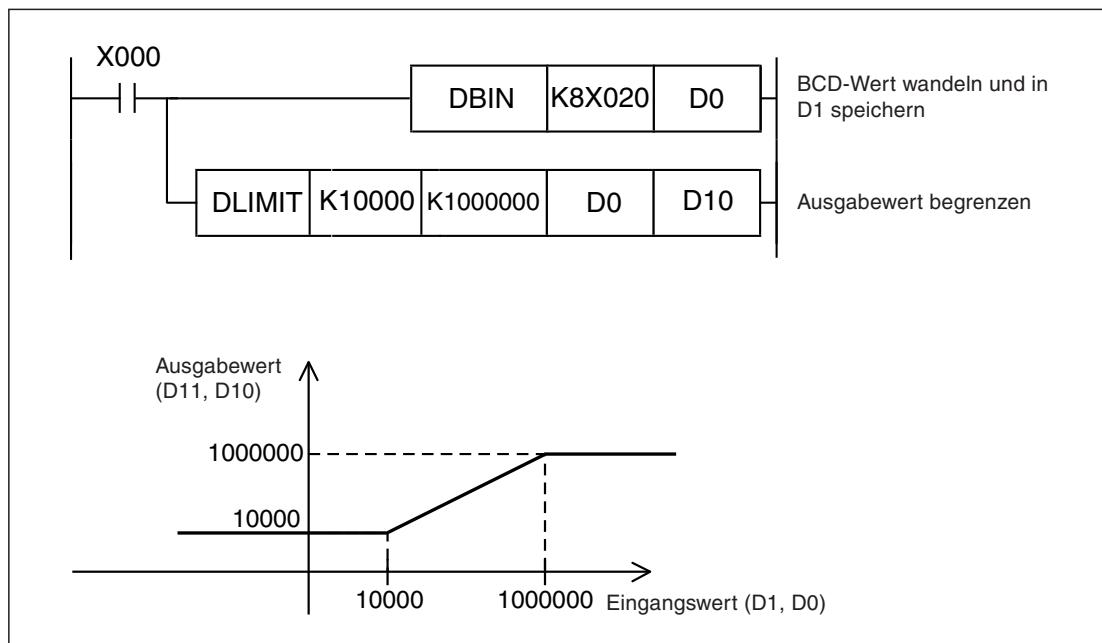
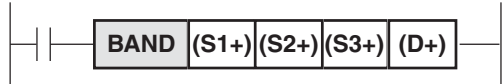


Abb. 7-265: Der Wert in D11 und D10 wird durch dieses Programm auf Werte zwischen 10000 und 1000000 begrenzt.



7.19.2 Eingangs-Offset festlegen (BAND)

					BAND		FNC 257					
					Eingangs-Offset festlegen						CPU	FX1S
Operanden	S1+	S2+	S2+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte				
	KnX, KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□), K, H		KnX, KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□)	KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□)	●	16 Bit	32 Bit	BAND BANDP	9			
						●	●	DBAND DBANDP	17			

Funktion

Eingangsoffset von 16- oder 32-Bit-Binärdaten

Beschreibung für 16-Bit-Verarbeitung (BAND, BANDP)

- Die BAND-Anweisung subtrahiert von der in dem in (S3+) angegebenen Eingangswert (16-Bit-Binärzahl) einen unteren (negativen) und oberen (positiven) Offsetwert. Der untere Offsetwert ist in (S1+) und der obere Offsetwert ist in (S2+) gespeichert. Das Ergebnis wird in Abhängigkeit vom Eingangswert in dem in (D+) angegebenen Operanden wie folgt gespeichert:
 - Wenn der Datenwert in (S3+) kleiner als der untere Offsetwert in (S1+) ist, wird das Ergebnis aus der Subtraktion (S3+) - (S1+) in dem in (D+) angegebenen Operanden eingetragen.
 - Wenn der Datenwert in (S3+) größer als der obere Offsetwert in (S2+) ist, wird das Ergebnis der Subtraktion (S3+)- (S2+) in dem in (D+) angegebenen Operanden gespeichert.
 - Wenn sich der Datenwert in (S3+) innerhalb der beiden Offsetbereiche befindet, wird der Wert 0 in dem in (D+) angegebenen Operanden eingetragen.

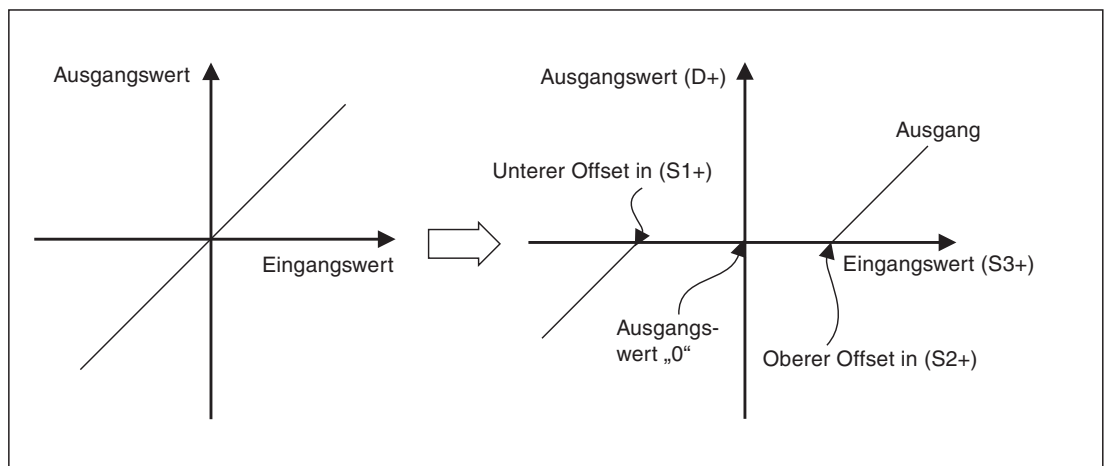


Abb. 7-266: Zusammenhang zwischen Ein- und Ausgangswert ohne (links) und mit Begrenzung durch die BAND-Anweisung (rechts)

- Der in (S1+), (S2+) und (S3+) angegebene Wert kann zwischen -32768 und 32767 liegen.

- Wenn das Subtraktionsergebnis außerhalb des Bereichs von -32768 bis 32767 liegt, findet folgender Vorgang statt:
 - Bei Unterschreiten des Wertes -32768 wird der Rest der Subtraktion bei 32767 beginnend ausgeführt. Wenn in ((S3+)) zum Beispiel der Wert -32768 gespeichert ist und ((S1+)) mit dem Wert 10 davon subtrahiert wird, ergibt sich:
 $-32768 - 10 = 8000_{\text{H}} - A_{\text{H}} = 7\text{FF}6_{\text{H}} = 32758$.
 - Bei Überschreiten des Wertes 32767 wird der Rest der Subtraktion beginnend bei -32768 ausgeführt.

Beschreibung für 32-Bit-Verarbeitung (DBAND, DBANDP)

- Die DBAND-Anweisung subtrahiert von dem in ((S3+)+1) und ((S3)+0) angegebenen Eingangswert (32-Bit-Binärzahl) einen unteren (negativen) und oberen (positiven) Offsetwert. Der untere Offsetwert ist in ((S1+)+1) und ((S1)+0) und der obere Offsetwert in ((S2+)+1) und ((S2)+0) gespeichert. Das Ergebnis wird in Abhängigkeit von dem Eingangswert in den in ((D+)+1) und ((D)+0) angegebenen Operanden wie folgt gespeichert:
 - Wenn der Datenwert in ((S3+)+1) und ((S3)+0) kleiner als der untere Offsetwert in ((S1+)+1) und ((S1)+0) ist, wird das Ergebnis aus der Subtraktion $[(S3+)+1, (S3)+0] - [(S1+)+1, (S1)+0]$ in dem in ((D+)+1) und ((D)+0) angegebenen Operanden eingetragen.
 - Wenn der Datenwert in ((S3+)+1) und ((S3)+0) größer als der obere Offsetwert in ((S2+)+1) und ((S2)+0) ist, wird das Ergebnis der Subtraktion $[(S3+)+1, (S3)+0] - [(S2+)+1, (S2)+0]$ in dem in ((D+)+1) und ((D)+0) angegebenen Operanden gespeichert.
 - Wenn sich der Datenwert in ((S3+)+1) und ((S3)+0) innerhalb der beiden Offsetbereiche befindet, wird in dem in ((D+)+1) und ((D)+0) angegebenen Operanden der Wert 0 gespeichert.

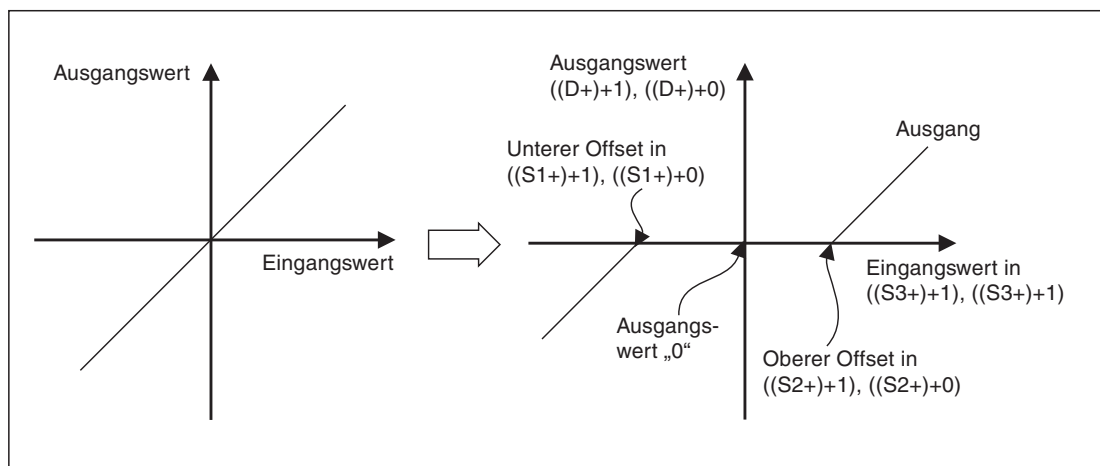


Abb. 7-267: Zusammenhang zwischen Ein- und Ausgangswert ohne (links) und mit Begrenzung durch die DBAND-Anweisung (rechts)

- Der in ((S1+)+1) und ((S1)+0), ((S2+)+1) und ((S2)+0) sowie in ((S3+)+1) und ((S3)+0) angegebene Wert kann zwischen -2147483648 und 2147483647 liegen.

- Wenn das Subtraktionsergebnis außerhalb des Bereichs von -2147483648 bis 2147483647 liegt, findet folgender Vorgang statt:
 - Bei Unterschreiten des Werts -2147483648 wird der Rest der Subtraktion bei 2147483647 beginnend ausgeführt. Wenn in ((S3+)+1) und ((S3)+0) zum Beispiel der Wert -2147483648 gespeichert ist und ((S1+)+1) und ((S1)+0) mit dem Wert 1000 davon subtrahiert werden, ergibt sich:
 $-2147483648 - 1000 = 80000000_H - 3E8_H = 7FFFC18_H = \underline{2147482648}$.
 - Bei Überschreiten des Wertes 2147483647 wird der Rest der Subtraktion beginnend bei dem Wert -2147483648 ausgeführt.

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Bei der 16-Bit-Verarbeitung ist der in (S1+) angegebene untere Offset größer als der in (S2+) angegebene obere Offset.
- Bei der 32-Bit-Verarbeitung ist der in ((S1+)+1) und ((S1)+0) angegebene untere Offset größer als der in ((S2+)+1) und ((S2)+0) angegebene obere Offset.

Beispiel ▾

Im folgenden Programm werden bei eingeschaltetem Eingang X0 von den BCD-Daten an X20 bis X37 der untere (negative) Offsetwert -1000 und der obere (positive) Offsetwert 1000 subtrahiert. Das Ergebnis wird in D1 gespeichert.

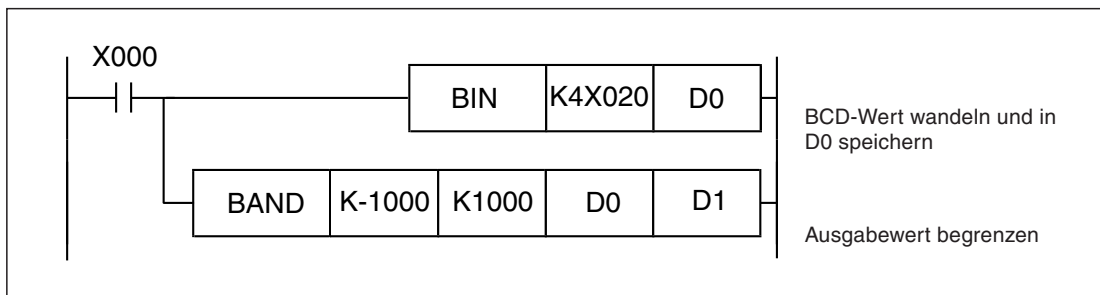


Abb. 7-268: Durch dieses Programm werden alle Eingangswerte zwischen -1000 und 1000 nicht in D1 gespeichert.

Beispiel ▾

Im folgenden Programm werden bei eingeschaltetem Eingang X0 von den BCD-Daten an X20 bis X57 der untere (negative) Offsetwert -10000 und der obere (positive) Offsetwert 10000 subtrahiert. Das Ergebnis wird in D10 und D11 gespeichert

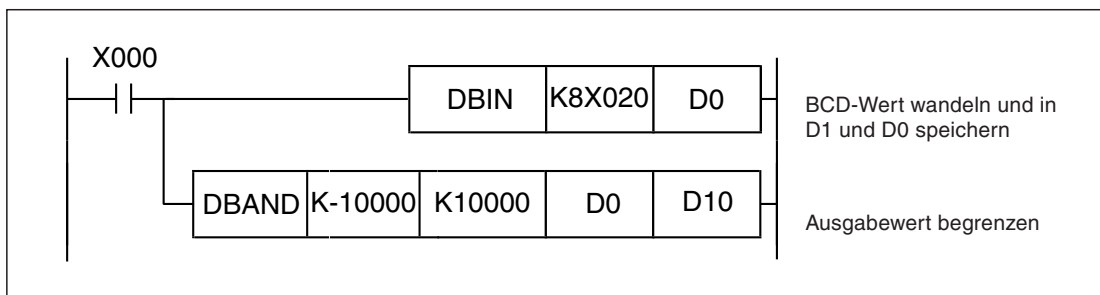
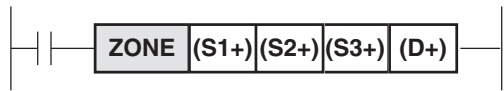


Abb. 7-269: Durch dieses Programm werden alle Eingangswerte zwischen -10000 und 10000 nicht in D11 und D10 gespeichert.



7.19.3 Ausgangs-Offset festlegen (ZONE)

					ZONE		FNC 258				
					Ausgangs-Offset festlegen						
					CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
Operanden	S1+	S2+	S2+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	KnX, KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□), K, H	KnX, KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□)	KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□)	KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□)		●	16 Bit	32 Bit	ZONE ZONEP	9	
						●	●	DZONED DZONEP	17		

Funktion

Ausgangsoffset von 16- oder 32-Bit-Binärdaten

Beschreibung für 16-Bit-Verarbeitung (ZONE, ZONEP)

- Die ZONE-Anweisung addiert zu der 16-Bit-Binärzahl in dem in (S3+) angegebenen Operanden einen negativen und einen positiven Offsetwert. Der negative Offsetwert ist in (S1+) und der positive Offsetwert in (S2+) gespeichert. Das Ergebnis wird in Abhängigkeit von dem Eingangswert in dem in (D+) angegebenen Operanden wie folgt gespeichert:
 - Wenn der Datenwert in (S3+) kleiner als 0 ist, wird in dem in (D+) angegebenen Operanden das Ergebnis aus der Addition (S3+) + (S1+) eingetragen.
 - Wenn der Datenwert in (S3+) größer als 0 ist, wird in dem in (D+) angegebenen Operanden das Ergebnis der Addition (S3+) + (S2+) gespeichert.
 - Wenn der Datenwert in (S3+) gleich 0 ist, wird in dem in (D+) angegebenen Operanden auch der Wert 0 eingetragen.

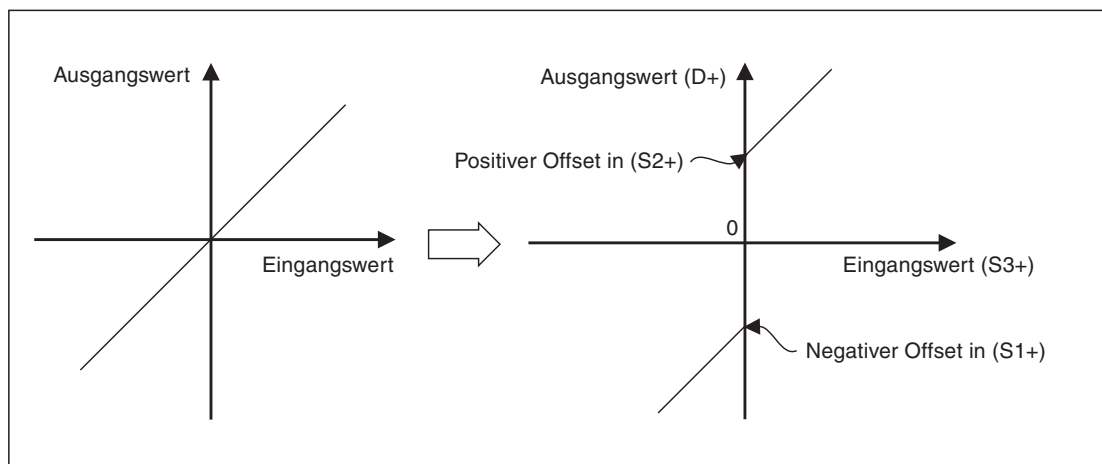


Abb. 7-270: Zusammenhang zwischen Ein- und Ausgangswert ohne (links) und mit Begrenzung durch die ZONE-Anweisung (rechts)

- Liegt das Additionsergebnis außerhalb des Bereichs von -32768 bis 32767, findet folgender Vorgang statt:
 - Bei Unterschreiten des Wertes -32768 wird der Rest der Operation bei 32767 beginnend ausgeführt. Wenn in (S3+) z. B. der Wert -32768 gespeichert ist und (S1+) mit dem Wert -100 addiert wird, ergibt sich:
 $-32768 + (-100) = 8000H + FF9CH = 7F9CH = \underline{32668}$

- Bei Überschreiten des Wertes 32767 wird der Rest der Addition beginnend bei -32768 ausgeführt.

Beschreibung für 32-Bit-Verarbeitung (DZONE, DZONEP)

- Die ZONE-Anweisung addiert zu der 16-Bit-Binärzahl in ((S3+)+1) und ((S3)+0) einen negativen und einen positiven Offsetwert. Der negative Offsetwert ist in ((S1+)+1) und ((S1)+0) und der positive Offsetwert in ((S2+)+1) und ((S2)+0) gespeichert. Das Ergebnis wird in Abhängigkeit von dem Eingangswert in ((D+)+1) und ((D)+0) gespeichert:
 - Wenn der Datenwert in ((S3+)+1) und ((S3)+0) kleiner als 0 ist, wird in ((D+)+1) und ((D)+0) das Ergebnis aus der Addition [((S3+)+1), ((S3)+0) + ((S1+)+1), ((S1)+0)] eingetragen.
 - Wenn der Datenwert in ((S3+)+1) und ((S3)+0) größer als 0 ist, wird in ((D+)+1) und ((D)+0) das Ergebnis der Addition [((S3+)+1), ((S3)+0) + ((S2+)+1), ((S2)+0)] gespeichert.
 - Wenn der Datenwert in ((S3+)+1) und ((S3)+0) gleich 0 ist, wird in ((D+)+1) und ((D)+0) ebenfalls der Wert 0 eingetragen.

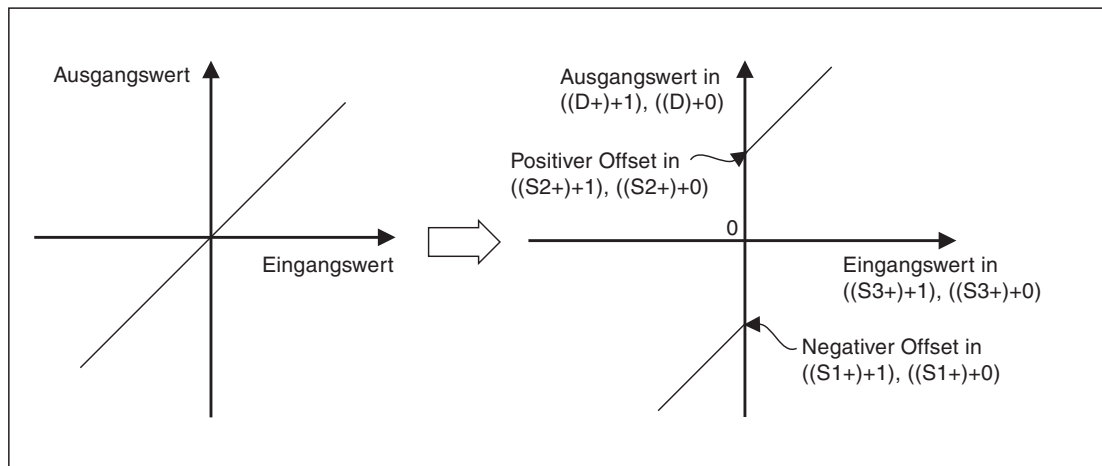


Abb. 7-271: Zusammenhang zwischen Ein- und Ausgangswert ohne (links) und mit Begrenzung durch die DZONE-Anweisung (rechts)

- Wenn das Additionsergebnis außerhalb des Bereichs von -2147483648 bis 2147483647 liegt, findet folgender Vorgang statt:
 - Bei Unterschreiten des Wertes -2147483648 wird der Rest der Operation bei 2147483647 beginnend ausgeführt. Wenn in ((S3+)+1) und ((S3)+0) zum Beispiel der Wert -2147483648 gespeichert ist und ((S1+)+1) und ((S1)+0) mit dem Wert -1000 addiert werden, ergibt sich:

$$-2147483648 + (-1000) = 80000000H + FFFFFFFC18H = 7FFFFFFC18H = \underline{2147482648}.$$
 - Bei Überschreiten des Wertes 2147483647 wird der Rest der Addition beginnend bei dem Wert -2147483648 ausgeführt.

Beispiel ▾

Im folgenden Programm werden beim Einschalten des Eingangs X0 zu den BCD-Daten an X20 bis X37 der negative Offsetwert -100 und der positive Offsetwert 100 addiert. Das Ergebnis wird in D1 gespeichert.

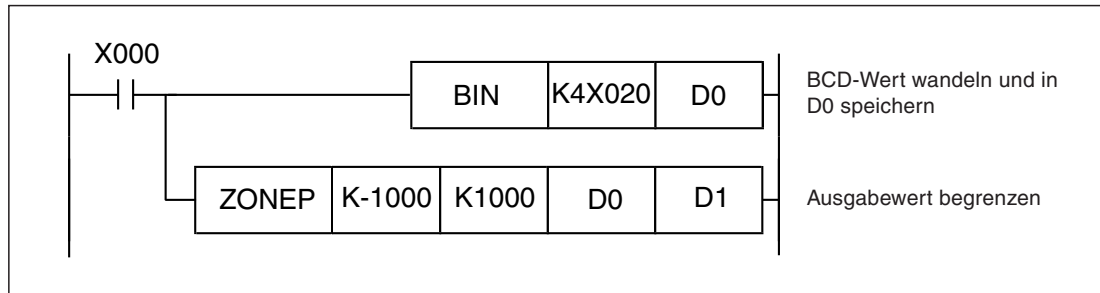


Abb. 7-272: Durch dieses Programm werden keine Ausgangswerte zwischen -1000 und 1000 ausgegeben.

Ist der Wert in D0 größer als 0, wird in D1 der Wert „D0 + 100“ eingetragen.

Ist der Wert in D0 kleiner als 0, wird in D1 der Wert „D0 + (-100)“ eingetragen.

Ist der Wert in D0 gleich 0, wird der Wert „0“ in D1 gespeichert.

△

Beispiel ▾

Das folgende Programm addiert beim Einschalten des Eingangs X0 zu den BCD-Daten an X20 bis X57 den negativen Offsetwert -10000 und den positiven Offsetwert 10000. Das Ergebnis wird in D10 und D11 gespeichert.

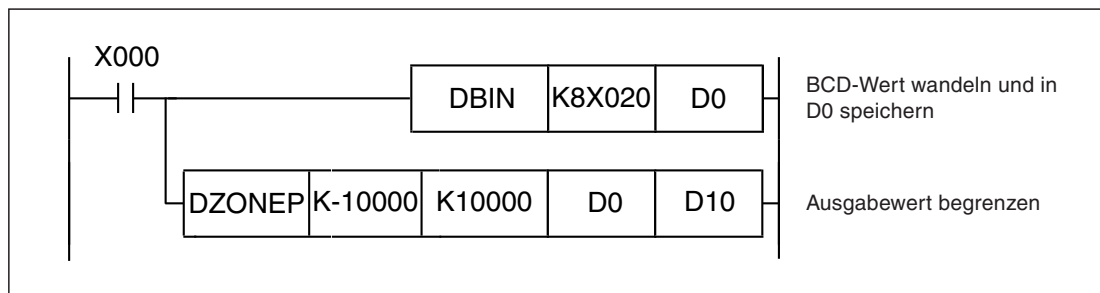


Abb. 7-273: Durch dieses Programm werden keine Ausgangswerte zwischen -10000 und 10000 ausgegeben.

Ist der Wert in D1 und D0 größer als 0, wird in D11 und D10 der Wert „D1, D0 + 10000“ eingetragen.

Ist der Wert in D1 und D0 kleiner als 0, wird in D11 und D10 der Wert „D0, D1 + (-10000)“ eingetragen.

Ist der Wert in D1 und D0 gleich 0, wird der Wert 0 in D11 und D10 gespeichert.

△

7.19.4 Werte skalieren (SCL)

				SCL		FNC 259				
				Skalierung						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
										●
Operanden	S1+	S2+	D+	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	KnX, KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□), K, H	D, R	KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□)	●		16 Bit	32 Bit	SCL SCLP	7	
						●	●	DSCL DSCLP	13	

Funktion

Werte mit Hilfe einer Tabelle skalieren, in der Koordinatenpaare abgelegt sind

Beschreibung für 16-Bit-Verarbeitung (SCL, SCLP)

- Der Eingangswert in (S1+) wird anhand einer vorgegebenen Wandlungscharakteristik skaliert und in (D+) gespeichert.

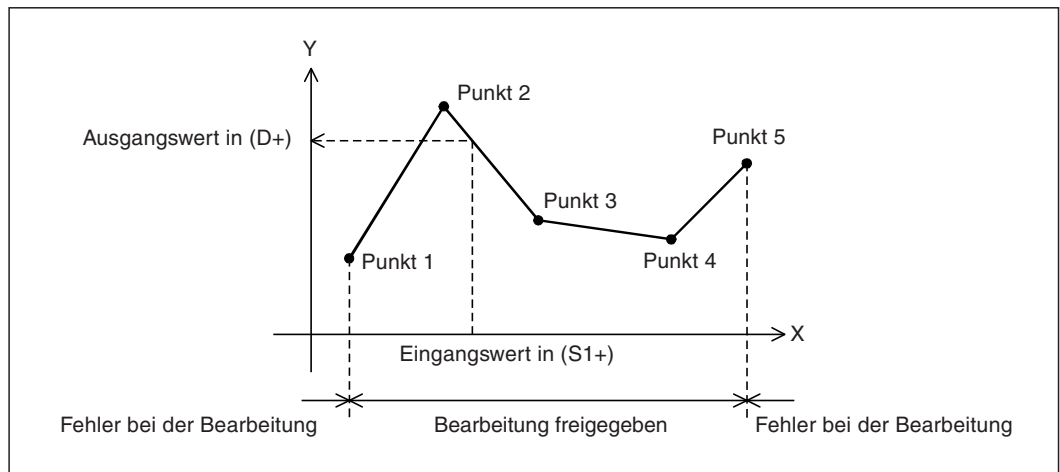


Abb. 7-274: Wenn der Eingangswert außerhalb des durch die Tabelle vorgegebenen Kurvenverlaufs liegt, tritt ein Fehler auf.

- Die Wandlungscharakteristik ist in einer Tabelle hinterlegt, deren erste Adresse in (S2+) angegeben wird.

Bedeutung		Operand
Anzahl der Punkte		(S2+)
Punkt 1	X-Koordinate	(S2+)+1
	Y-Koordinate	(S2+)+2
Punkt 2	X-Koordinate	(S2+)+3
	Y-Koordinate	(S2+)+4
Punkt 3	X-Koordinate	(S2+)+5
	Y-Koordinate	(S2+)+6
:	:	:
Punkt n	X-Koordinate	(S2+)+(2n-1)
	Y-Koordinate	(S2+)+2n

Tab. 7-275:

Belegung der Tabelle mit den Kurvenpunkten bei der 16-Bit-Verarbeitung

- Falls der berechnete Ausgangswert Nachkommastellen besitzt, wird der Wert gerundet. Maßgebend für die Rundung ist der Wert der ersten Nachkommastelle.

Beschreibung für 32-Bit-Verarbeitung (DSCL, DSCLP)

- Der Eingangswert in $((S1+)+1)$ und $((S1+)+0)$ wird anhand einer vorgegebenen Wandlungscharakteristik skaliert und in $((D+)+1)$ und $((D+)+0)$ gespeichert.

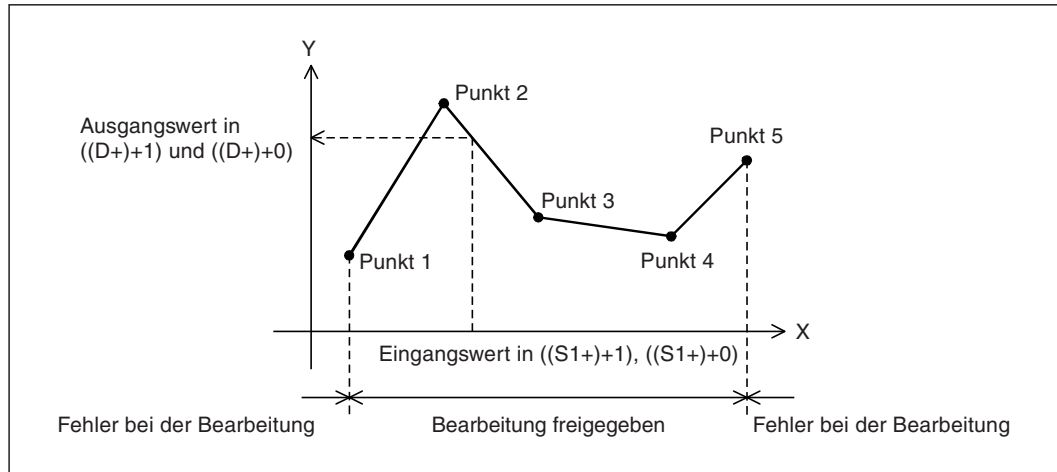


Abb. 7-276: Wenn der Eingangswert außerhalb des durch die Tabelle vorgegebenen Kurvenverlaufs liegt, tritt ein Fehler auf.

- Die Wandlungscharakteristik ist in einer Tabelle hinterlegt, deren erste Adresse in $(S2+)$ angegeben wird.

Bedeutung		Operanden
Anzahl der Punkte		$(S2+)+1, (S2+)$
Punkt 1	X-Koordinate	$(S2+)+3, (S2+)+2$
	Y-Koordinate	$(S2+)+5, (S2+)+4$
Punkt 2	X-Koordinate	$(S2+)+7, (S2+)+6$
	Y-Koordinate	$(S2+)+9, (S2+)+8$
Punkt 3	X-Koordinate	$(S2+)+11, (S2+)+10$
	Y-Koordinate	$(S2+)+13, (S2+)+12$
:	:	:
Punkt n	X-Koordinate	$(S2+)+(4n-1), (S2+)+(4n-2)$
	Y-Koordinate	$(S2+)+(4n+1), (S2+)+(4n)$

Tab. 7-70:

Belegung der Tabelle mit den Kurvenpunkten bei der 32-Bit-Verarbeitung

- Falls der berechnete Ausgangswert Nachkommastellen besitzt, wird der Wert gerundet. Maßgebend für die Rundung ist der Wert der ersten Nachkommastelle.

HINWEIS

Mit einer SCL2-Anweisung (Abschnitt 7.19.7) können ebenfalls Werte skaliert werden. Sie unterscheidet sich von der SCL-Anweisung durch eine andere Struktur der Tabelle mit den Kurvenpunkten.

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- In der Tabelle mit den Kurvenpunkten wurden die Werte nicht in aufsteigender Reihenfolge der X-Koordinaten eingetragen.
- Der Wert in $(S1+)$ bzw. $((S1+)+1)$ und $((S1+)+0)$ liegt außerhalb der Werte in der Tabelle.
- Der Abstand zweier aufeinanderfolgender Punkte ist größer als 65535.

Beispiel zur Vorgabe der Kurvenpunkte

Die folgende Abbildung zeigt als Beispiel für die 16-Bit-Verarbeitung eine Kurve, in der die Punkte mit ihren X- und Y-Koordinaten eingetragen sind.

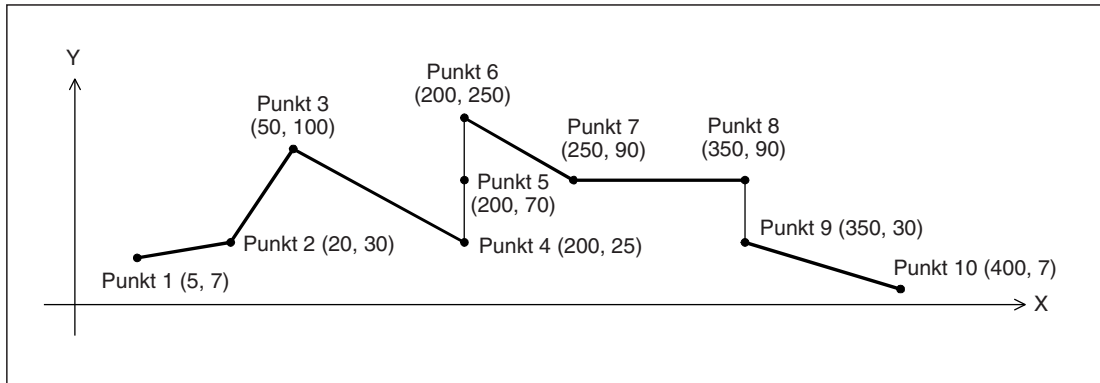


Abb. 7-277: Kurvenverlauf für das Beispiel

Dadurch, dass für die Punkte 4, 5 und 6 dieselbe X-Koordinate (200) angegeben ist, wird ein Zwischenwert ausgegeben. Ist bei drei oder mehr Punkten die X-Koordinate gleich, wird die Y-Koordinate des zweiten Punktes ausgegeben. In diesem Beispiel wird der Ausgangswert (Zwischenwert) durch die Y-Koordinate von Punkt 5 bestimmt.

Haben zwei Punkte dieselbe X-Koordinate, wie in diesem Beispiel Punkt 8 und Punkt 9, wird der Ausgabewert vom zweiten Punkt in der Reihenfolge bestimmt (in diesem Beispiel von Punkt 9).

Bedeutung	Operand	Operandenadresse	Inhalt	
Anzahl der Punkte	(S2+)	D0	10	
Punkt 1	X-Koordinate	(S2+)+1	D1	5
	Y-Koordinate	(S2+)+2	D2	7
Punkt 2	X-Koordinate	(S2+)+3	D3	20
	Y-Koordinate	(S2+)+4	D4	30
Punkt 3	X-Koordinate	(S2+)+5	D5	50
	Y-Koordinate	(S2+)+6	D6	100
Punkt 4	X-Koordinate	(S2+)+7	D7	200
	Y-Koordinate	(S2+)+8	D8	25
Punkt 5	X-Koordinate	(S2+)+9	D9	200
	Y-Koordinate	(S2+)+10	D10	70
Punkt 6	X-Koordinate	(S2+)+11	D11	200
	Y-Koordinate	(S2+)+12	D12	250
Punkt 7	X-Koordinate	(S2+)+13	D13	250
	Y-Koordinate	(S2+)+14	D14	90
Punkt 8	X-Koordinate	(S2+)+15	D15	350
	Y-Koordinate	(S2+)+16	D16	90
Punkt 9	X-Koordinate	(S2+)+17	D17	350
	Y-Koordinate	(S2+)+18	D18	30
Punkt 10	X-Koordinate	(S2+)+19	D19	400
	Y-Koordinate	(S2+)+20	D20	7

Tab. 7-71: Belegung der Koordinatentabelle für dieses Beispiel (Die Spalte „Operandenadresse“ gibt die Operanden für den Fall an, dass für (S2+) D0 angegeben wurde.)



Beispiel ▾

Im folgenden Beispielprogramm wird der Inhalt von D0 mit Hilfe einer Tabelle skaliert, die ab R0 gespeichert ist. Der Ausgangswert wird in D10 gespeichert.

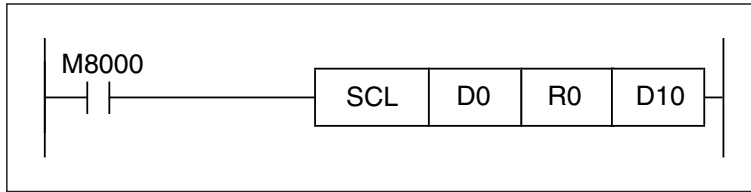


Abb. 7-278:
Programmbeispiel zur SCL-Anweisung

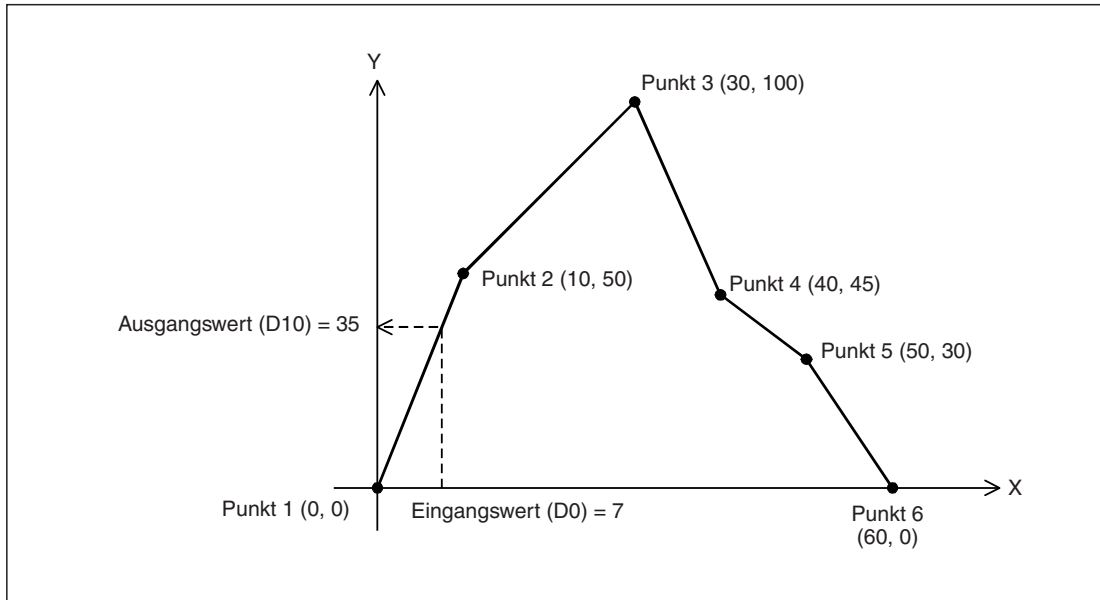


Abb. 7-279: Kurve zur Skalierung des Eingangswerts bei diesem Beispiel

Bedeutung		Operand	Operandenadresse	Inhalt
Anzahl der Punkte		(S2+)	R0	6
Punkt 1	X-Koordinate	(S2+)+1	R1	0
	Y-Koordinate	(S2+)+2	R2	0
Punkt 2	X-Koordinate	(S2+)+3	R3	10
	Y-Koordinate	(S2+)+4	R4	50
Punkt 3	X-Koordinate	(S2+)+5	R5	30
	Y-Koordinate	(S2+)+6	R6	100
Punkt 4	X-Koordinate	(S2+)+7	R7	40
	Y-Koordinate	(S2+)+8	R8	45
Punkt 5	X-Koordinate	(S2+)+9	R9	50
	Y-Koordinate	(S2+)+10	R10	30
Punkt 6	X-Koordinate	(S2+)+11	R11	60
	Y-Koordinate	(S2+)+12	R12	0

Tab. 7-72:Koordinatentabelle für dieses Beispiel



7.19.5 Zahl im ASCII-Code in Binärwert wandeln (DABIN)

			DABIN		FNC 260				
			Wandlung von ASCII-Code in Binär						
			CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
Operanden	S+	D+	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	T, C, D, R	KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□), V, Z	●		16 Bit	32 Bit	DABIN DABINP	5	
					●	●	DDABIN DDABINP	9	

Funktion

Wandlung dezimaler ASCII-Daten in 16-/32-Bit-Binär-Daten

Beschreibung für 16-Bit-Verarbeitung (DABIN, DABINP)

- Die DABIN-Anweisung konvertiert die in ((S+)+0) bis ((S+)+2) angegebenen Dezimalzahl im ASCII-Code in das Format BIN-16-Bit und speichert sie in (D+).
- Jede gespeicherte Stelle der Zahl im ASCII-Code kann einen Wert zwischen „30H“ und „39H“ annehmen.
- Der in ((S+)+0) bis ((S+)+2) angegebene Wert im ASCII-Code kann im Bereich zwischen -32768 und 32767 liegen.
- Wenn als Vorzeichen der ASCII-Code „20H“ (Leerzeichen) im niederwertigen Byte von ((S+)+0) gespeichert ist, wird der Binärwert positiv. Beim ASCII-Code „2DH“ (Minuszeichen) als Leerzeichen wird ein negativer Binärwert gespeichert.
- Enthält eine Stelle der zu wandelnden Zahl den Wert „20H“ (Leerzeichen) oder „00H“ (NUL), wird dieser Wert automatisch als „30H“ („0“) interpretiert.

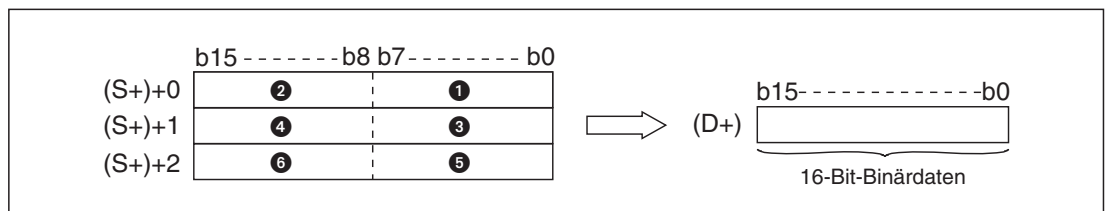


Abb. 7-280: Eine DABIN-Anweisung wandelt eine Zahl mit bis zu 5 Stellen in Binärdaten.

- 1 ASCII-Code des Vorzeichens
- 2 ASCII-Code der Zehntausender-Stelle
- 3 ASCII-Code der Tausender-Stelle
- 4 ASCII-Code der Hunderter-Stelle
- 5 ASCII-Code der Zehner-Stelle
- 6 ASCII-Code der Einer-Stelle

Beispiel ▾

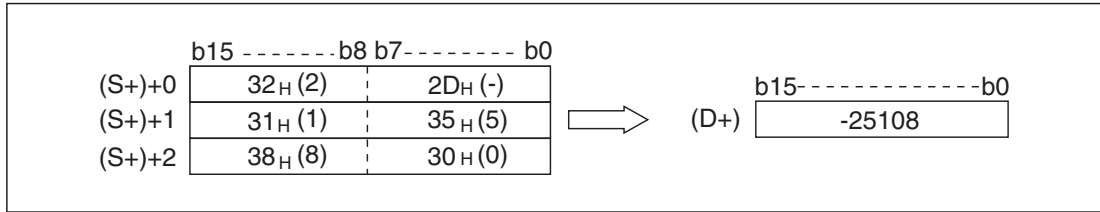


Abb. 7-281: Die Zahl „-25108“ im ASCII-Code wird mit einer DABIN-Anweisung gewandelt.



Beschreibung für 32-Bit-Verarbeitung (DDABIN, DDABINP)

- Die DDABIN-Anweisung konvertiert die in ((S+)+0) bis ((S+)+5) angegebenen Dezimalzahl im ASCII-Code in das Format BIN-32-Bit und speichert sie in ((D+)+1) und ((D+)+0).
- Jede gespeicherte Stelle der Zahl im ASCII-Code kann einen Wert zwischen „30H“ und „39H“ annehmen.
- Der in ((S+)+0) bis ((S+)+5) angegebene Wert im ASCII-Code kann im Bereich zwischen -2147483648 und 2147483647 liegen. Der Inhalt des höherwertigen Bytes von ((S+)+5) wird ignoriert.
- Wenn als Vorzeichen der ASCII-Code „20H“ (Leerzeichen) im niederwertigen Byte von ((S+)+0) gespeichert ist, wird der Binärwert positiv. Beim ASCII-Code „2DH“ (Minuszeichen) als Leerzeichen wird ein negativer Binärwert gespeichert.
- Enthält eine Stelle der zu wandelnden Zahl den Wert „20H“ (Leerzeichen) oder „00H“ (NUL), wird dieser Wert automatisch als „30H“ („0“) interpretiert.

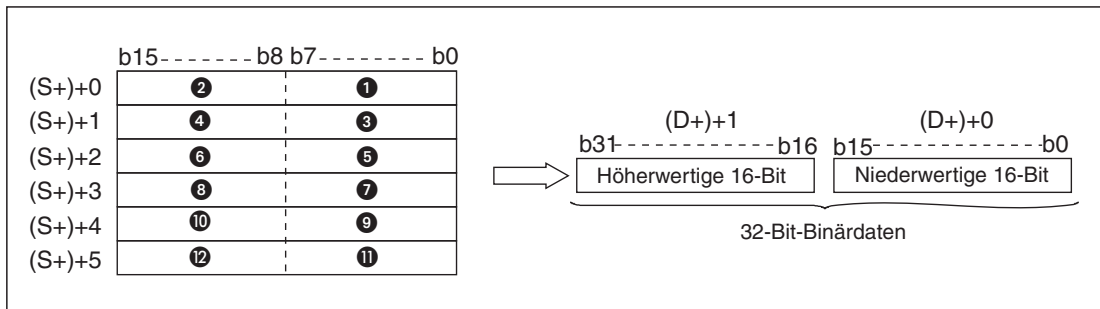


Abb. 7-282: Eine DDABIN-Anweisung wandelt eine Zahl mit bis zu 10 Stellen in Binärdaten.

- 1 ASCII-Code des Vorzeichens
- 2 ASCII-Code der Milliarden-Stelle
- 3 ASCII-Code der Hundertmillionen-Stelle
- 4 ASCII-Code der Zehnmillionen-Stelle
- 5 ASCII-Code der Millionen-Stelle
- 6 ASCII-Code der Hunderttausender-Stelle
- 7 ASCII-Code der Zehntausender-Stelle
- 8 ASCII-Code der Tausender-Stelle
- 9 ASCII-Code der Hunderter-Stelle
- 10 ASCII-Code der Zehner-Stelle
- 11 ASCII-Code der Einer-Stelle
- 12 Der Inhalt dieses Bytes wird ignoriert.

Beispiel ▾

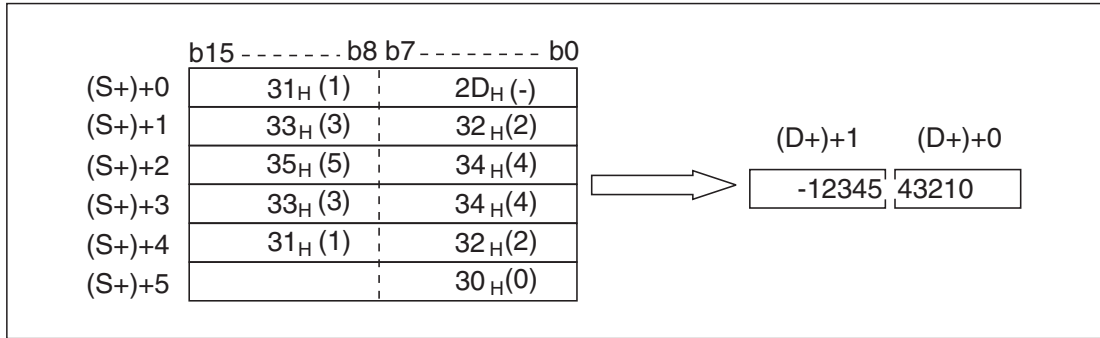


Abb. 7-283: Die Zahl „-1234543210“ im ASCII-Code wird mit einer DDABIN-Anweisung gewandelt.



Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Das Vorzeichen in den unteren 8 Bits des Operanden ((S+)+0) enthält einen anderen Wert als „20H“ oder „2DH“.
- Der ASCII-Code in ((S+)+0) bis ((S+)+2) bzw. ((S+)+0) bis ((S+)+5) entspricht anderen Werten als „30H“ bis „39H“, „20H“ oder „00H“.
- Die Zahl ab (S+) liegt außerhalb der folgenden Bereiche:
 - DABIN-, DABINP-Anweisung: -32768 bis 32767
 - DDABIN-, DDABINP-Anweisung: -2147483648 bis 2147483647
- Durch die Angabe von (S+) wird der zulässige Bereich für diesen Operandentyp überschritten. (Bei der 16-Bit-Ausführung werden nach (S+) noch zwei und bei der 32-Bit-Ausführung noch 5 weitere Operanden belegt.)

Beispiel ▾

Beim Einschalten von X0 wandelt das folgende Programm die fünfstellige Dezimalzahl, die im ASCII-Code ab D20 gespeichert ist, in einen Binärwert und speichert ihn in D0.

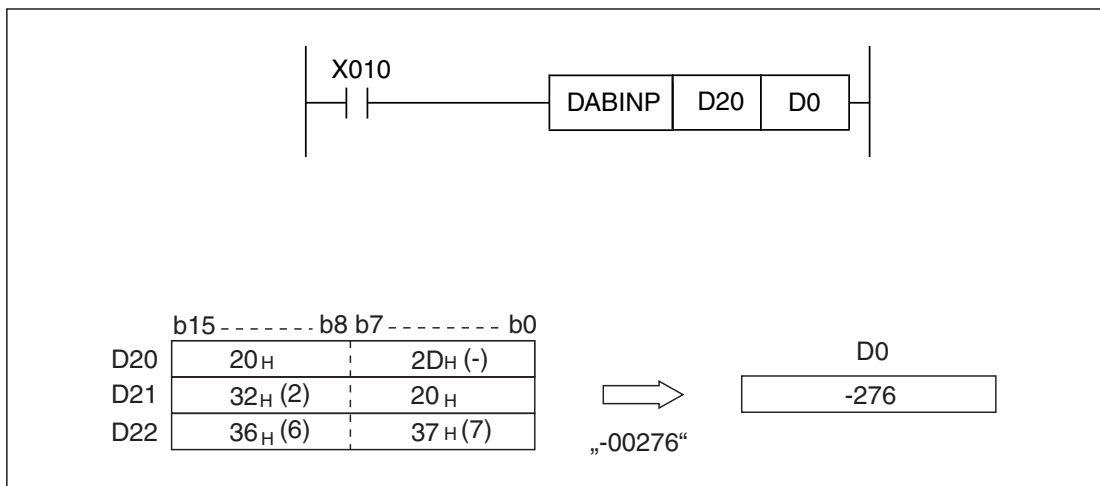


Abb. 7-284: Der Wert „- 276“ wird als „-00276“ interpretiert und als „-276“ gespeichert.



7.19.6 Binärwert in ASCII-Code wandeln (BINDA)

			BINDA		FNC 261				
			Wandlung eines Binärwertes in ASCII-Code						
			CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
Operanden	S+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□), V, Z	T, C, D, R		●	16 Bit	32 Bit	BINDA BINDAP	5	
				●	●	DBINDA DBINDAP	9		

Funktion

Wandlung von 16-/32-Bit-Binärdaten in Dezimalzahlen im ASCII-Code

Beschreibung für 16-Bit-Verarbeitung (BINDA, BINDAP)

- Die BINDA-Anweisung konvertiert die in (S+) angegebene 16-Bit-Binärzahl in eine Dezimalzahl im ASCII-Code und speichert sie ab (D+).
- Die in (S+) angegebene 16-Bit-Binärzahl kann in dem Bereich zwischen -32768 und 32767 liegen.
- Ist die 16-Bit-Binärzahl positiv, wird als Vorzeichen der ASCII-Code „20H“ (Leerzeichen) im niederwertigen Byte von ((D+)+0) gespeichert. Bei einer negativen Binärzahl, wird hier der ASCII-Code „2DH“ (Minuszeichen) als Vorzeichen gespeichert.
- Enthält die Binärzahlen vorangestellte Nullen, werden diese in der gewandelten Zahl durch Leerzeichen („20H“) ersetzt. Zum Beispiel werden im Wert „00325“ die Nullen der Zehntausender- und Tausender-Stellen durch „20H“ ersetzt.
- Die Speicherung der Daten in ((D+)+3) hängt vom Zustand des Sondermerkers M8091 ab.
 - Ist M8091 nicht gesetzt, wird in ((D+)+3) der ASCII-Code „00H“ (NUL) eingetragen.
 - Ist M8091 gesetzt, wird der Inhalt von ((D+)+3) nicht verändert.

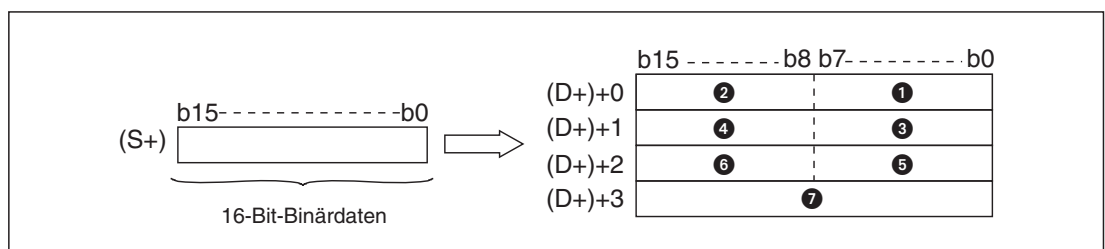


Abb. 7-285: Eine BINDA-Anweisung wandelt Binärdaten in eine Zahl mit bis zu 5 Stellen.

- ① ASCII-Code des Vorzeichens
- ② ASCII-Code der Zehntausender-Stelle
- ③ ASCII-Code der Tausender-Stelle
- ④ ASCII-Code der Hunderter-Stelle
- ⑤ ASCII-Code der Zehner-Stelle
- ⑥ ASCII-Code der Einer-Stelle
- ⑦ „00H“ (M8091 = 0) oder unveränderter Inhalt (M8091 = 1)

Beispiel ▾

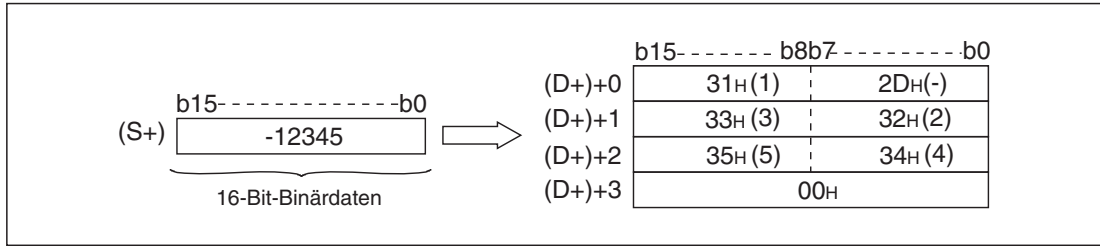


Abb. 7-286: Die Zahl „-12345“ wird mit einer BINDA-Anweisung gewandelt. Der Sondermerker M8091 ist in diesem Beispiel nicht gesetzt.



Beschreibung für 32-Bit-Verarbeitung (DBINDA, DBINDAP)

- Die BINDA-Anweisung konvertiert die in ((S+)+1) und ((S+)+0) angegebene 32-Bit-Binärzahl in eine Dezimalzahl im ASCII-Code und speichert sie ab (D+).
- Die in ((S+)+1) und ((S+)+0) angegebene 32-Bit-Binärzahl kann in dem Bereich zwischen -2147483648 und 2147483647 liegen.
- Ist die 16-Bit-Binärzahl positiv, wird als Vorzeichen der ASCII-Code „20H“ (Leerzeichen) im niederwertigen Byte von ((D+)+0) gespeichert. Bei einer negativen Binärzahl, wird hier der ASCII-Code „2DH“ (Minuszeichen) als Vorzeichen gespeichert.
- Enthält die Binärzahlen vorangestellte Nullen, werden diese in der gewandelten Zahl durch Leerzeichen („20H“) ersetzt. Zum Beispiel werden im Wert „0012034560“ die Nullen Milliarden- und Hundertmillionen-Stellen durch „20H“ ersetzt.
- Die Speicherung der Daten in ((D+)+5) hängt vom Zustand des Sondermerkers M8091 ab.
 - Ist M8091 nicht gesetzt, wird der ASCII-Code „00H“ (NUL) in das höherwertige Byte von ((D+)+5) eingetragen.
 - Ist M8091 gesetzt, wird der ursprüngliche Inhalt des höherwertigen Bytes von ((D+)+5) nicht verändert.

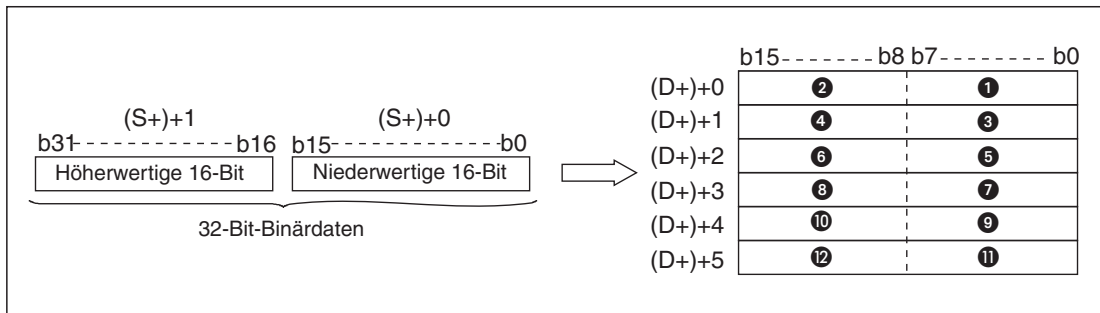


Abb. 7-287: Eine DBINDA-Anweisung wandelt Binärdaten in eine Zahl mit bis zu 10 Stellen.

- ① ASCII-Code des Vorzeichens
- ② ASCII-Code der Milliarden-Stelle
- ③ ASCII-Code der Hundertmillionen-Stelle
- ④ ASCII-Code der Zehnmillionen-Stelle
- ⑤ ASCII-Code der Millionen-Stelle
- ⑥ ASCII-Code der Hunderttausender-Stelle
- ⑦ ASCII-Code der Zehntausender-Stelle
- ⑧ ASCII-Code der Tausender-Stelle
- ⑨ ASCII-Code der Hunderter-Stelle

- ⑩ ASCII-Code der Zehner-Stelle
- ⑪ ASCII-Code der Einer-Stelle
- ⑫ „00H“ (M8091 = 0) oder unveränderter Inhalt (M8091 = 1)

Beispiel ▾

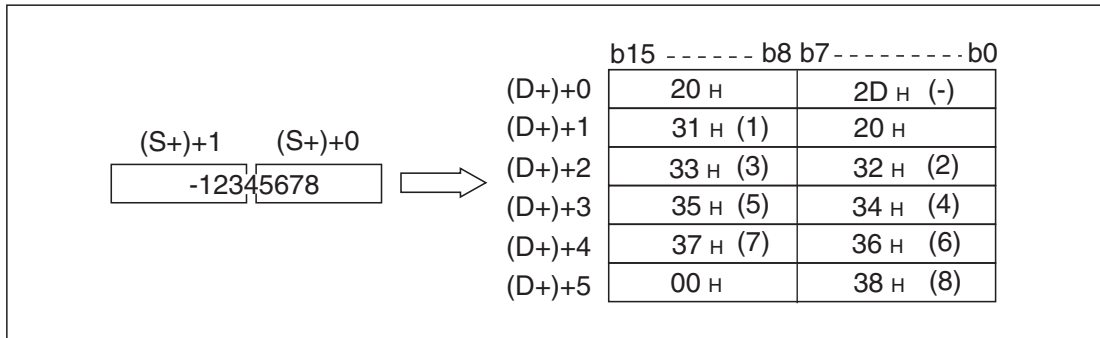


Abb. 7-288: Die Zahl „-12345678“ wird mit einer DBINDA-Anweisung in den ASCII-Code gewandelt. Der Sondermerker M8091 ist in diesem Beispiel nicht gesetzt.



Fehlerquellen

Im folgenden Fall tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- Durch die Angabe von (D+) wird der zulässige Bereich für diesen Operandentyp überschritten. (Bei der 16-Bit-Ausführung werden nach (D+) noch zwei bzw. drei und bei der 32-Bit-Ausführung noch 5 weitere Operanden belegt.)

Beispiel ▾

Beim Einschalten von X0 wandelt das folgende Programm den in D1000 gespeicherten Binärwert in eine ASCII-codierte Zahl und speichert diesen Wert ab D0.

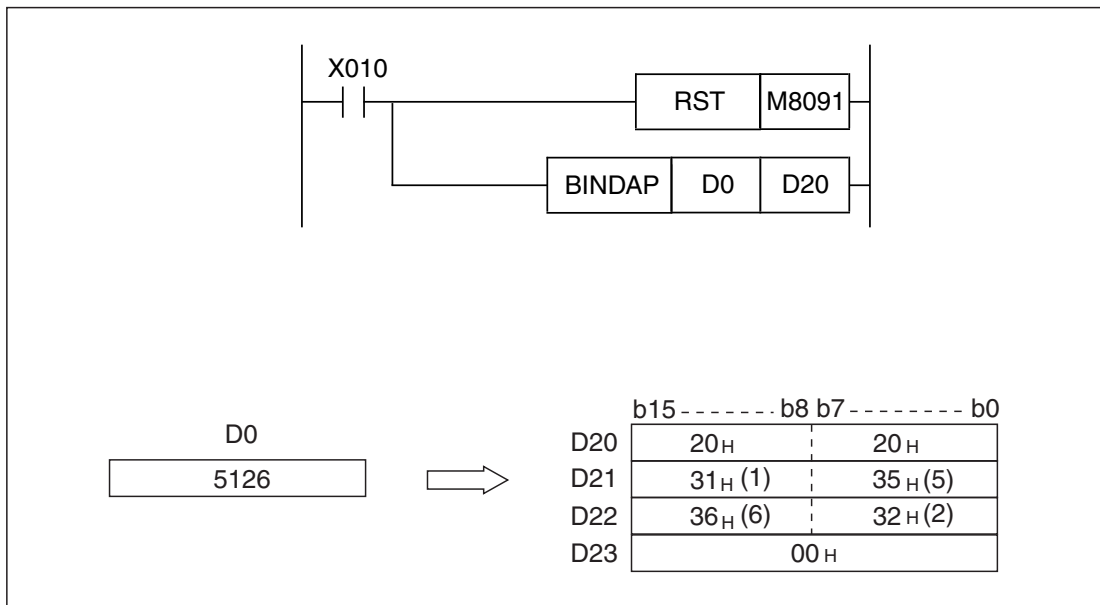


Abb. 7-289: Vor der Wandlung wird M8091 zurückgesetzt, damit in D23 der Wert „00H“ eingetragen wird.



7.19.7 Werte skalieren (SCL2)

				SCL2		FNC 269				
				Skalierung						
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
										●
Operanden	S1+	S2+	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	KnX, KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□), K, H	D, R	KnY, KnM, KnS, T, C, D, R, Sondermodule (U□/G□)	●	16 Bit	32 Bit	SCL2 SCL2P	7		
					●	●	DSCL2 DSCL2P	13		

Funktion

Werte mit Hilfe einer Tabelle skalieren, die nach Kurvenpunkten sortiert ist

Beschreibung für 16-Bit-Verarbeitung (SCL2, SCL2P)

- Der Eingangswert in (S1+) wird anhand einer vorgegebenen Wandlungscharakteristik skaliert und in (D+) gespeichert.

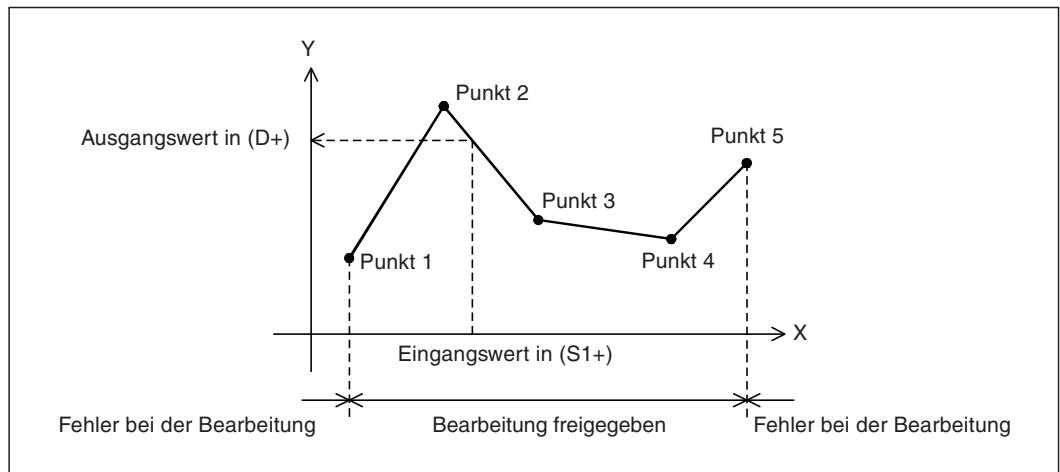


Abb. 7-290: Wenn der Eingangswert außerhalb des durch die Tabelle vorgegebenen Kurvenverlaufs liegt, tritt ein Fehler auf.

- Die Wandlungscharakteristik ist in einer Tabelle hinterlegt, deren erste Adresse in (S2+) angegeben wird.

Bedeutung		Operand
Anzahl der Punkte		(S2+)
X-Koordinaten	Punkt 1	(S2+)+1
	Punkt 2	(S2+)+2
	:	:
	Punkt n	(S2+)+n
Y-Koordinaten	Punkt 1	(S2+)+(n+1)
	Punkt 2	(S2+)+(n+2)
	:	:
	Punkt n	(S2+)+(2n)

Tab. 7-73:

Belegung der Tabelle mit den Kurvenpunkten bei der 16-Bit-Verarbeitung

- Falls der berechnete Ausgangswert Nachkommastellen besitzt, wird der Wert gerundet. Maßgebend für die Rundung ist der Wert der ersten Nachkommastelle.

Beschreibung für 32-Bit-Verarbeitung (DSCL2, DSCL2P)

- Der Eingangswert in ((S1+)+1) und ((S1+)+0) wird anhand einer vorgegebenen Wandlungscharakteristik skaliert und in ((D+)+1) und ((D+)+0) gespeichert.

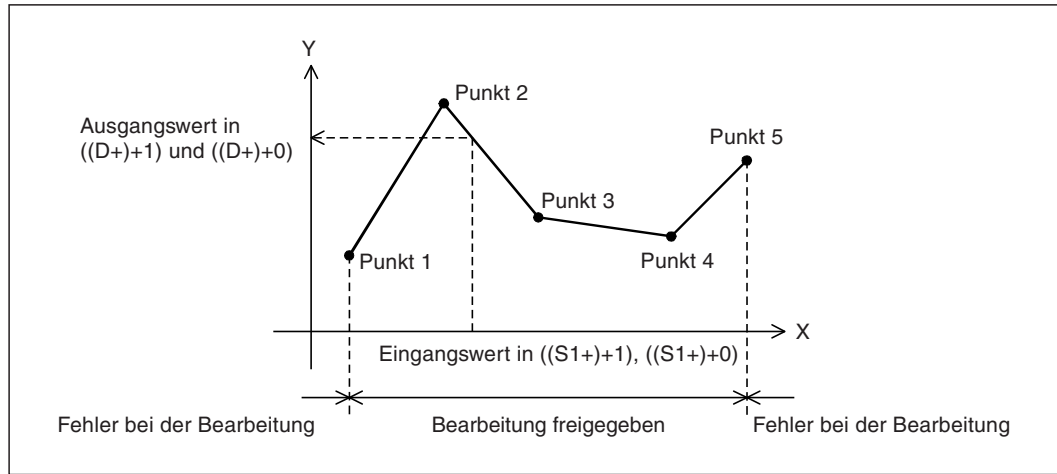


Abb. 7-291: Wenn der Eingangswert außerhalb des durch die Tabelle vorgegebenen Kurvenverlaufs liegt, tritt ein Fehler auf.

- Die Wandlungscharakteristik ist in einer Tabelle hinterlegt, deren erste Adresse in (S2+) angegeben wird.

Bedeutung		Operand
Anzahl der Punkte		(S2+)+1, (S2+)
X-Koordinaten	Punkt 1	(S2+)+3, (S2+)+2
	Punkt 2	(S2+)+5, (S2+)+4
	:	:
	Punkt n	(S2+)+(2n+1), (S2+)+(2n)
Y-Koordinaten	Punkt 1	(S2+)+(2n+3), (S2+)+(2n+2)
	Punkt 2	(S2+)+(2n+5), (S2+)+(2n+4)
	:	:
	Punkt n	(S2+)+(4n+1), (S2+)+(4n)

Tab. 7-74:

Belegung der Tabelle mit den Kurvenpunkten bei der 32-Bit-Verarbeitung

- Falls der berechnete Ausgangswert Nachkommastellen besitzt, wird der Wert gerundet. Maßgebend für die Rundung ist der Wert der ersten Nachkommastelle.

HINWEIS

Mit einer SCL-Anweisung (Abschnitt 7.19.4) können ebenfalls Werte skaliert werden. Sie unterscheidet sich von der SCL2-Anweisung durch eine andere Struktur der Tabelle mit den Kurvenpunkten.

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6706“ eingetragen:

- In der Tabelle mit den Kurvenpunkten wurden die Werte nicht in aufsteigender Reihenfolge der X-Koordinaten eingetragen.
- Der Wert in (S1+) bzw. ((S1+)+1) und ((S1+)+0) liegt außerhalb der Werte in der Tabelle.
- Der Abstand zweier aufeinanderfolgender Punkte ist größer als 65535.

Beispiel zur Vorgabe der Kurvenpunkte

Die folgende Abbildung zeigt als Beispiel für die 16-Bit-Verarbeitung eine Kurve, in der die Punkte mit ihren X- und Y-Koordinaten eingetragen sind.

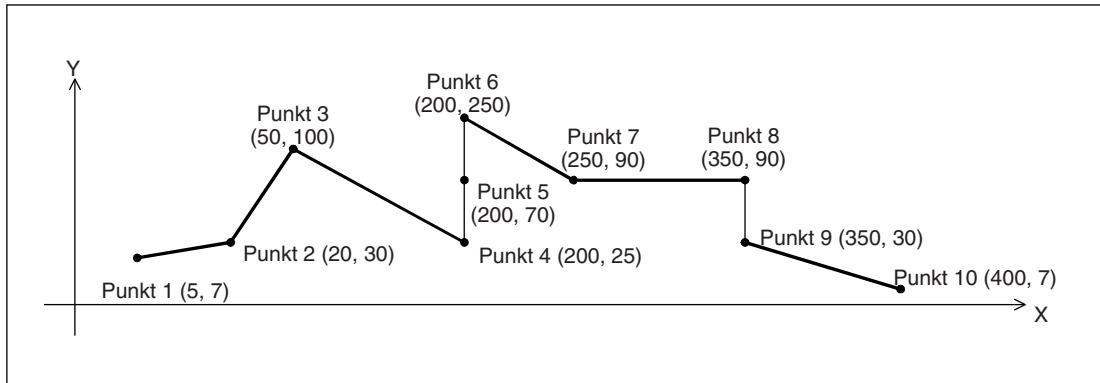


Abb. 7-292: Kurvenverlauf für das Beispiel

Dadurch, dass für die Punkte 4, 5 und 6 dieselbe X-Koordinate (200) angegeben ist, wird ein Zwischenwert ausgegeben. Ist bei drei oder mehr Punkten die X-Koordinate gleich, wird die Y-Koordinate des zweiten Punktes ausgegeben. In diesem Beispiel wird der Ausgangswert (Zwischenwert) durch die Y-Koordinate von Punkt 5 bestimmt.

Haben zwei Punkte dieselbe X-Koordinate, wie in diesem Beispiel Punkt 8 und Punkt 9, wird der Ausgabewert vom zweiten Punkt in der Reihenfolge bestimmt (in diesem Beispiel von Punkt 9).

Bedeutung	Operand	Operandenadresse	Inhalt	
Anzahl der Punkte	(S2+)	D0	10	
X-Koordinaten	Punkt 1	(S2+)+1	D1	5
	Punkt 2	(S2+)+2	D2	20
	Punkt 3	(S2+)+3	D3	50
	Punkt 4	(S2+)+4	D4	200
	Punkt 5	(S2+)+5	D5	200
	Punkt 6	(S2+)+6	D6	200
	Punkt 7	(S2+)+7	D7	250
	Punkt 8	(S2+)+8	D8	350
	Punkt 9	(S2+)+9	D9	350
	Punkt 10	(S2+)+10	D10	400
Y-Koordinaten	Punkt 1	(S2+)+11	D11	7
	Punkt 2	(S2+)+12	D12	30
	Punkt 3	(S2+)+13	D13	100
	Punkt 4	(S2+)+14	D14	25
	Punkt 5	(S2+)+15	D15	70
	Punkt 6	(S2+)+16	D16	250
	Punkt 7	(S2+)+17	D17	90
	Punkt 8	(S2+)+18	D18	90
	Punkt 9	(S2+)+19	D19	30
	Punkt 10	(S2+)+20	D20	7

Tab. 7-75: Belegung der Koordinatentabelle für dieses Beispiel (Die Spalte „Operandenadresse“ gibt die Operanden für den Fall an, dass für (S2+) D0 angegeben wurde.)



Beispiel ▾

Im folgenden Beispielprogramm wird der Inhalt von D0 mit Hilfe einer Tabelle skaliert, die ab R0 gespeichert ist. Der Ausgangswert wird in D10 gespeichert.

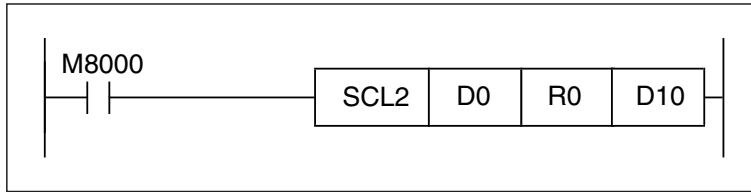


Abb. 7-293:
Programmbeispiel zur SCL-Anweisung

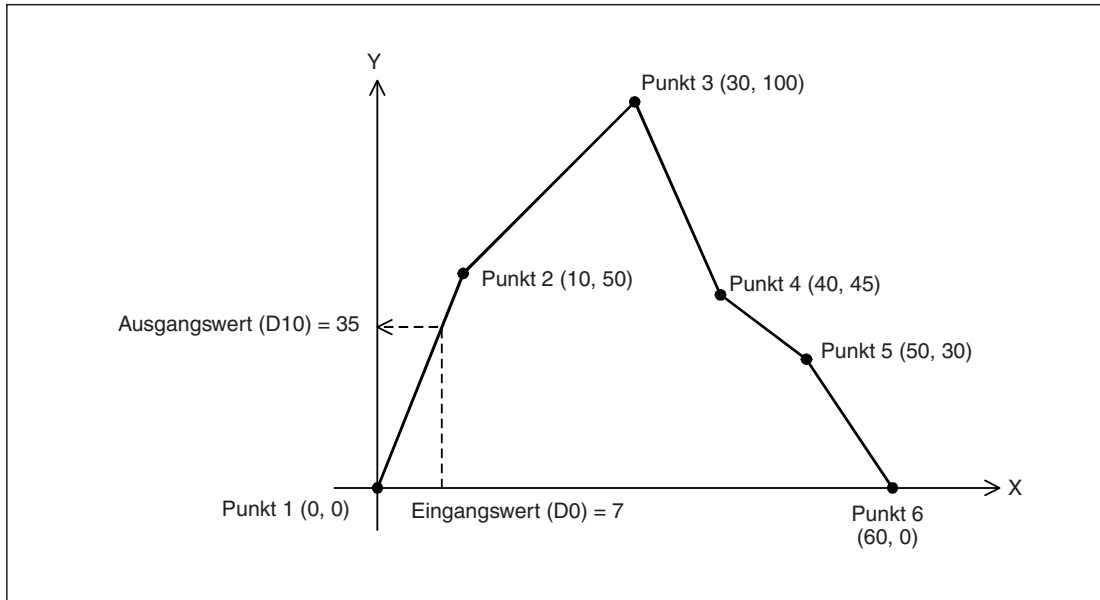


Abb. 7-294: Kurve zur Skalierung des Eingangswerts bei diesem Beispiel

Bedeutung	Operand	Operandenadresse	Inhalt	
Anzahl der Punkte	(S2+)	D0	6	
X-Koordinaten	Punkt 1	(S2+)+1	D1	0
	Punkt 2	(S2+)+2	D2	10
	Punkt 3	(S2+)+3	D3	30
	Punkt 4	(S2+)+4	D4	40
	Punkt 5	(S2+)+5	D5	50
	Punkt 6	(S2+)+6	D6	60
Y-Koordinaten	Punkt 1	(S2+)+11	D11	0
	Punkt 2	(S2+)+12	D12	50
	Punkt 3	(S2+)+13	D13	100
	Punkt 4	(S2+)+14	D14	45
	Punkt 5	(S2+)+15	D15	30
	Punkt 6	(S2+)+16	D16	0

Tab. 7-76:Koordinatentabelle für dieses Beispiel



7.20 Kommunikation mit Frequenzumrichtern

Die in diesem Abschnitt beschriebenen Anweisungen ermöglichen den Datenaustausch mit Frequenzumrichtern der A700-, D700-, E700-, F700-, A500-, E500-, F500-, S500- und V500-Serie von Mitsubishi Electric.

Symbol	FNC	Bedeutung	Abschnitt
IVCK	270	Status des Frequenzumrichters prüfen	7.20.1
IVDR	271	Frequenzumrichter steuern	7.20.2
IVRD	272	Parameter des Frequenzumrichters lesen	7.20.3
IVWR	273	Parameter in Frequenzumrichter schreiben	7.20.4
IVBWR	274	Parameter blockweise in Frequenzumrichter schreiben	7.20.5
IVMC	275	Kommando/Sollfrequenz in Frequenzumrichter schreiben und Status/Istfrequenz (Drehzahl) aus Frequenzumrichter lesen	7.20.6

Tab. 7-77: Übersicht der Anweisungen zur Kommunikation mit Frequenzumrichtern

HINWEISE

Eine IVCK-, IVDR-, IVRD-, IVWR-, IVBWR- oder IVMC-Anweisung darf nicht gleichzeitig mit einer der folgenden Anweisungen für dieselbe Schnittstelle ausgeführt werden:

- RS (FNC80)
- RS2 (FNC87)
- ADPRW (FNC276)
- FLCRT (FNC300) bis FLSTRD (FNC305)

Für dieselbe Schnittstelle können mehrere Anweisungen zur Kommunikation mit Frequenzumrichtern (FNC270 bis FNC275) gleichzeitig ausgeführt werden.

Die Anwendung dieser Anweisungen ist im Kommunikationshandbuch zu den Steuerungen der MELSEC FX-Familie (Art.-Nr. 137315) beschrieben.

Ob ein SPS-Grundgerät der FX3G/FX3GC/FX3GE/FX3S/FX3U/FX3UC-Serie mit einem Frequenzumrichter einer bestimmten Serie kommunizieren kann, hängt von der Version des SPS-Grundgeräts ab.

SPS	Kommunikation mit Frequenzumrichtern der Serien		
	A500, E500, F500, S500, V500	A700, F700	D700, E700
FX3G	ab Version 1.10		
FX3GC	ab Version 1.40		
FX3GE	ab Version 1.00		
FX3S	ab Version 1.00		
FX3U	ab Version 2.20		ab Version 2.32
FX3UC	ab Version 1.00	ab Version 2.20	ab Version 2.32

Tab. 7-78: Kompatibilität zwischen FX-SPS und Frequenzumrichter

Bei der Kommunikation mit Frequenzumrichtern können die folgenden Sondermerker und -register ausgewertet werden:

Sondermerker		Bedeutung
Schnittstelle Ch. 1	Schnittstelle Ch. 2	
M8029		Ausführung der Anweisung beendet
M8063	M8438	Fehler bei der seriellen Kommunikation
M8151	M8156	Datenaustausch mit Frequenzumrichter aktiv ^①
M8152	M8157	Fehler beim Datenaustausch mit Frequenzumrichter ^①
M8153	M8158	Gespeicherter Fehler beim Datenaustausch mit Frequenzumrichter ^①
M8154	M8159	Fehler bei der Ausführung einer IVBWR-Anweisung ^①

Tab. 7-79: Sondermerker in Verbindung mit der Kommunikation mit Frequenzumrichtern

^① Diese Merker werden zurückgesetzt, wenn die Betriebsart der SPS von STOP nach RUN wechselt.

Sonderregister		Bedeutung
Schnittstelle Ch. 1	Schnittstelle Ch. 2	
D8063	D8438	Fehlercode bei Störung der seriellen Kommunikation
D8150	D8155	Wartezeit für eine Reaktion des Frequenzumrichters
D8151	D8156	Schrittnummer bei Datenaustausch mit Frequenzumrichter ^②
D8152	D8157	Fehlercode bei Störung des Datenaustausches mit Frequenzumrichtern ^①
D8153	D8158	Schrittnummer, bei der bei Kommunikation mit Frequenzumrichtern der Fehler aufgetreten ist ^②
D8154	D8159	Parameternummer bei Fehler bei der Ausführung einer IVBWR-Anweisung ^②

Tab. 7-80: Sonderregister in Verbindung mit der Kommunikation mit Frequenzumrichtern

^① Diese Register werden gelöscht, wenn die Betriebsart der SPS von STOP nach RUN wechselt.

^② Nach dem Start der SPS wird in diese Register der Wert „-1“ eingetragen.

7.20.1 Status eines Frequenzumrichters prüfen (IVCK)

					IVCK		FNC 270				
					Status eines Frequenzumrichters prüfen						
Operanden	S1+	S2+	D+	n	CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
									●	●	●
	S1+	S2+	D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	D, R ^① , U□/G□ ^② , K, H		KnY, KnM, KnS, D, R ^① , U□/G□ ^② , K, H	K, H		16 Bit	32 Bit	IVCK		9	
						●					

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktion

Status eines Frequenzumrichters auslesen (Monitorfunktion)

Beschreibung

- Mit einer IVCK-Anweisung kann der Status eines Frequenzumrichters geprüft werden.
- Die Funktion einer IVCK-Anweisung entspricht einer EXTR K10-Anweisung bei der FX2N- oder FX2NC-Serie (Abschnitt 7.13.1).
- In (S1+) wird die Stationsnummer des Frequenzumrichters (0 bis 31) angegeben.
- (S2+) enthält den Anweisungscode für den Frequenzumrichter (siehe folgende Tabelle)

Anweisungscode	Bedeutung der vom Frequenzumrichter gesendeten Daten	Frequenzumrichter-Serien, die den Anweisungscode unterstützen								
		F700	A700	E700	D700	V500	F500	A500	E500	S500
H7B	Betriebsart	●	●	●	●	●	●	●	●	●
H6F	Ausgangsfrequenz	●	●	●	●	●	●	●	●	●
H70	Ausgangsstrom	●	●	●	●	●	●	●	●	●
H71	Ausgangsspannung	●	●	●	●	●	●	●	●	
H72	Sonderüberwachung	●	●	●	●	●	●	●		
H73	Auswahlnummer zur Sonderüberwachung	●	●	●	●	●	●	●		
H74	Alarm Definition	●	●	●	●	●	●	●	●	●
H75	Alarm Definition	●	●	●	●	●	●	●	●	●
H76	Alarm Definition	●	●	●	●	●	●	●	●	
H77	Alarm Definition	●	●	●	●	●	●	●	●	
H79	Frequenzumrichter-Status (erweitert)	●	●	●	●					
H7A	Frequenzumrichter-Status	●	●	●	●	●	●	●	●	●
H6E	Eingestellte Ausgangsfrequenz (EEPROM)	●	●	●	●	●	●	●	●	●
H6D	Eingestellte Ausgangsfrequenz (RAM)	●	●	●	●	●	●	●	●	●
H7F	Erweiterte Einstellung für Link-Parameter	Diese Codes können nicht in (S2+) der IVCK-Anweisung angegeben werden. Sie werden automatisch ausgeführt, wenn für eine IVRD-Anweisung ein Code zum Lesen des 2. Parametersatzes angegeben wird.								
H6C	Umschaltung 2. Parametersatz									

Tab. 7-81: Anweisungscode in (S2+) für die IVCK-Anweisung

- (D+) gibt den Operanden an, in dem die vom Frequenzumrichter gesendeten Daten abgelegt werden.
- In (n) wird die Nummer der verwendeten Schnittstelle* angegeben (1: Ch. 1, 2: Ch. 2).

* Die Schnittstelle Ch. 2 steht bei einem FX3G-Grundgerät mit 14 oder 24 E/A und den Grundgeräten der FX3S-Serie nicht zur Verfügung.

Bei einem Grundgerät der FX3GE-Serie kann für n nur die Schnittstelle Ch. 2 angegeben werden, weil Ch. 1 von der integrierten Ethernet-Schnittstelle belegt wird.

7.20.2 Frequenzumrichter steuern (IVDR)

					IVDR		FNC 271				
					Frequenzumrichter steuern						
					CPU	FX1S	FX1N	FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
									●	●	●
Operanden	S1+	S2+	S3+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	D, R ^① , U□/G□ ^② , K, H		KnX, KnY, KnM, KnS, D, R ^① , U□/G□ ^②		K, H		16 Bit	32 Bit	IVDR		9
						●					

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktion

Steuerung eines Frequenzumrichters

Beschreibung

- Mit einer IVDR-Anweisung kann ein Frequenzumrichter gesteuert werden.
- Die Funktion einer IVDR-Anweisung entspricht einer EXTR K11-Anweisung bei der FX2N- oder FX2NC-Serie (Abschnitt 7.13.1).
- In (S1+) wird die Stationsnummer des Frequenzumrichters (0 bis 31) angegeben.
- (S2+) enthält den Anweisungscode für den Frequenzumrichter (siehe Tabelle auf der folgenden Seite)

Anweisungscode	Bedeutung der an den Frequenzumrichter gesendeten Daten	Frequenzumrichter-Serien, die den Anweisungscode unterstützen								
		F700	A700	E700	D700	V500	F500	A500	E500	S500
HFB	Betriebsart	●	●	●	●	●	●	●	●	●
HF3	Auswahlnummer zur Sonderüberwachung	●	●	●	●	●	●	●		
HF9	Betriebssignal (erweitert)	●	●	●	●					
HFA	Betriebssignal	●	●	●	●	●	●	●	●	●
HEE	Ausgangsfrequenz (Eintrag ins EEPROM)	●	●	●	●	● ^③	●	●	●	●
HED	Ausgangsfrequenz (Eintrag ins RAM)	●	●	●	●	● ^③	●	●	●	●
HFD ^①	Frequenzumrichter zurücksetzen ^②	●	●	●	●	●	●	●	●	●
HF4	Alarmliste löschen	●	●	●	●		●	●	●	●
HFC	Alle Parameter löschen	●	●	●	●	●	●	●	●	●
HFC	Benutzerdefiniertes Löschen	●	●	●	●		●	●		
HFF	Erweiterte Einstellung für Link-Parameter	●	●	●	●	●	●	●	●	●

Tab. 7-82: Anweisungscode in (S2+) für die IVDR-Anweisung

- ① Der Anweisungscode HFD (Frequenzumrichter zurücksetzen) verlangt keine Reaktion vom Frequenzumrichter. Das bedeutet, dass kein Fehler auftritt, wenn dieser Anweisungscode an eine Stationsnummer gesendet wird, unter der kein Frequenzumrichter erreichbar ist.
Es dauert ca. 2,2 s, bis ein Frequenzumrichter zurückgesetzt ist.
- ② Geben Sie beim Zurücksetzen des Frequenzumrichters als Operanden (S3+) der IVDR-Anweisung H9696 an. H9966 darf nicht angegeben werden.
- ③ Bitte übertragen Sie unmittelbar, bevor mit einer IVDR-Anweisung die Ausgangsfrequenz eingestellt wird, den Wert „0“ mit dem Anweisungscode HFF (Erweiterte Einstellung für Link-Parameter) an den Frequenzumrichter. Wird der Wert „0“ nicht übertragen, wird eventuell die Ausgangsfrequenz nicht korrekt eingestellt.
- (S3+) enthält die Daten, die an den Frequenzumrichter übertragen werden.
 - In (n) wird die Nummer der verwendeten Schnittstelle* angegeben (1: Ch. 1, 2: Ch. 2).
- * Die Schnittstelle Ch. 2 steht bei einem FX3G-Grundgerät mit 14 oder 24 E/A und den Grundgeräten der FX3S-Serie nicht zur Verfügung.
Bei einem Grundgerät der FX3GE-Serie kann für n nur die Schnittstelle Ch. 2 angegeben werden, weil Ch. 1 von der integrierten Ethernet-Schnittstelle belegt wird.

7.20.3 Parameter des Frequenzumrichters lesen (IVRD)

					IVRD		FNC 272				
					Parameter aus Frequenzumrichter lesen						
					CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
									●	●	●
Operanden	S1+	S2+	D+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	D, R ^① , U□/G□ ^② , K, H		D, R ^① , U□/G□ ^②		K, H		16 Bit	32 Bit	IVRD		9
							●				

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktion

Auslesen der Parameter eines Frequenzumrichters

Beschreibung

- Mit einer IVRD-Anweisung können im Frequenzumrichter gespeicherte Parameter gelesen werden.
- Die Funktion einer IVRD-Anweisung entspricht einer EXTR K12-Anweisung bei der FX2N- oder FX2NC-Serie (Abschnitt 7.13.1).
- In (S1+) wird die Stationsnummer des Frequenzumrichters (0 bis 31) angegeben.
- (S2+) enthält die Nummer des Parameters, dessen Einstellung gelesen werden soll.
- In (D+) wird der Operand angegeben, in dem die gelesenen Parameter abgelegt werden.
- In (n) wird die Nummer der verwendeten Schnittstelle* angegeben (1: Ch. 1, 2: Ch. 2).

* Die Schnittstelle Ch. 2 steht bei einem FX3G-Grundgerät mit 14 oder 24 E/A und den Grundgeräten der FX3S-Serie nicht zur Verfügung.

Bei einem Grundgerät der FX3GE-Serie kann für n nur die Schnittstelle Ch. 2 angegeben werden, weil Ch. 1 von der integrierten Ethernet-Schnittstelle belegt wird.

7.20.4 Parameter in Frequenzumrichter schreiben (IVWR)

					IVWR		FNC 273				
					Parameter in Frequenzumrichter schreiben						
					CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
									●		●
Operanden	S1+	S2+	S3+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	D, R ^① , U□/G□ ^② , K, H				K, H		16 Bit	32 Bit	IVWR		9
							●				

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktion

Wert eines Parameters im Frequenzumrichter verändern

Beschreibung

- Mit einer IVWR-Anweisung können Parameter im Frequenzumrichter verändert werden.
- Die Funktion einer IVWR-Anweisung entspricht einer EXTR K13-Anweisung bei der FX2N- oder FX2NC-Serie (Abschnitt 7.13.1).
- In (S1+) wird die Stationsnummer des Frequenzumrichters (0 bis 31) angegeben.
- (S2+) enthält die Nummer des Parameters, der geändert werden soll.
- (S3+) enthält die Daten, die an den Frequenzumrichter übertragen werden.
- In (n) wird die Nummer der verwendeten Schnittstelle angegeben (1: Ch. 1, 2: Ch. 2*).

HINWEISE

Die Schnittstelle Ch. 2 steht bei einem FX3G-Grundgerät mit 14 oder 24 E/A und den Grundgeräten der FX3S-Serie nicht zur Verfügung.

Bei einem Grundgerät der FX3GE-Serie kann für n nur die Schnittstelle Ch. 2 angegeben werden, weil Ch.1 von der integrierten Ethernet-Schnittstelle belegt wird.

Hinweise zur Passwortfunktion bei den Frequenzumrichtern der D700-Serie

- Verhalten beim Auftreten eines Kommunikationsfehler

Tritt bei der Ausführung einer Anweisung zur Kommunikation mit einem Frequenzumrichter ein Fehler auf, wiederholt das FX-Grundgerät den Kommunikationsversuch bis zu zwei Mal. (Mit dem ersten Kommunikationsversuch und den beiden Wiederholungen werden also maximal drei Kommunikationsversuche ausgeführt.)

Wird in einem Frequenzumrichter der D700-Serie durch den Parameter Pr. 297 die Anzahl der fehlerhaften Passworteingaben* angezeigt, und tritt ein Passwort-Reset-Fehler auf, kann es daher vorkommen, dass die angezeigte Anzahl nicht mit der tatsächlichen Anzahl der fehlerhaften Passworteingaben übereinstimmt.

Wiederholen Sie beim Schreiben von Daten in Pr. 297 im Ablaufprogramm nicht automatisch die Anweisung zur Kommunikation mit einem Frequenzumrichter.

Fälle, in denen bei der Ausführung einer Anweisung zur Kommunikation mit einem Frequenzumrichter ein Passwort-Reset-Fehler auftritt und die tatsächliche Anzahl der fehlerhaften Passworteingaben:

- Schreiben eines fehlerhaften Passworts in Pr. 297 nach Eingabe eines falschen Passworts. Wird die Anweisung zum Schreiben einmal ausgeführt, werden drei fehlerhafte Passworteingaben registriert.
- Wenn das Passwort wegen elektromagnetischer Störungen etc. nicht korrekt in Pr. 297 eingetragen werden kann, werden bis zu drei fehlerhafte Passworteingaben registriert.

* Wenn durch Pr. 297 die Anzeige der Anzahl der fehlerhaften Passworteingaben aktiviert ist und fünf Mal ein fehlerhaftes Passwort eingegeben wird, kann die dadurch verursachte Sperre des Lesens und Schreibens auch durch die Eingabe des korrekten Passworts nicht aufgehoben werden. Dies kann nur durch das Löschen aller Parameter im Frequenzumrichter der D700-Serie geschehen.

- Festlegung eines Passworts

Um durch eine Anweisung zur Kommunikation mit einem Frequenzumrichter ein Passwort in einen Frequenzumrichter der D700-Serie einzutragen, schreiben Sie das Passwort in Pr. 297, lesen dann Pr. 297 aus und prüfen anschließend, ob das Passwort fehlerfrei eingetragen wurde.*

Falls das Passwort wegen elektromagnetischer Störungen etc. nicht korrekt in Pr. 297 übertragen werden konnte, wiederholt das FX-Grundgerät automatisch das Schreiben und es kann vorkommen, dass das eingetragene Passwort durch die Wiederholungen gesperrt wird.

* Wird Pr. 297 ausgelesen und als Inhalt ein Wert zwischen 0 und 4 ermittelt, wurde das Passwort fehlerfrei eingetragen.

7.20.5 Parameter blockweise in Frequenzumrichter schreiben (IVBWR)

					IVBWR		FNC 274				
					Parameter in Frequenzumrichter schreiben						
					CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
											●
Operanden	S1+	S2+	S3+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	D, R, U□/G□, K, H				K, H		16 Bit	32 Bit	IVBWR	9	
						●					

Funktion

Werte von mehreren Parameter im Frequenzumrichter verändern

Beschreibung

- Mit einer IVBWR-Anweisung können mehrere Parameter gleichzeitig im Frequenzumrichter verändert werden.
- In (S1+) wird die Stationsnummer des Frequenzumrichters (0 bis 31) angegeben.
- (S2+) enthält die Anzahl der Parameter, die geändert werden sollen.
- (S3+) enthält die erste Adresse eines Operandenbereichs, der die Nummern der zu ändernden Parameter und die Werte dieser Parameter enthält.

Operand	Bedeutung	
(S3+)+0	1. Parameter	Parameternummer
(S3+)+1		Sollwert
(S3+)+2	2. Parameter	Parameternummer
(S3+)+3		Sollwert
:	:	:
(S3+)+(2m-4)*	(m-1)-ter Parameter	Parameternummer
(S3+)+(2m-3)*		Sollwert
(S3+)+(2m-2)*	m-ter Parameter	Parameternummer
(S3+)+(2m-1)*		Sollwert

Tab. 7-83: Belegung der Operanden ab (S3+)

* „m“ ist die Anzahl der Parameter, die in (S2+) angegeben wird.

- In (n) wird die Nummer der verwendeten Schnittstelle angegeben (1: Ch. 1, 2: Ch. 2).

HINWEIS

Mit einer IVBWR-Anweisung können nur Parameter in Frequenzumrichtern der A700-, D700-, E700- und F700-Serie geändert werden.

7.20.6 Mehrfachanweisung für Frequenzumrichter (IVMC)

						IVMC		FNC 275				
						Mehrfachanweisung für Frequenzumrichter						
						CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
										●	●	●
Operanden	S1+	S2+	S3+	D+	n	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	D, R ^① , U□/G□ ^② , K, H		D, R ^① , U□/G□ ^②	KnY, KnM, KnS, D, R ^① , U□/G□ ^②	K, H			16 Bit	32 Bit	IVMC		11
								●				

① Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

② Nur bei FX3U und FX3UC

Funktion

Eine IVMV-Anweisung schreibt ein Kommando und die Sollfrequenz in einen Frequenzumrichter und liest gleichzeitig den Status und die Istfrequenz (Drehzahl) aus dem Frequenzumrichter.

Beschreibung

- Mit einer IVMC-Anweisung können ein Kommando und die Soll-Ausgangsfrequenz zum Frequenzumrichter übermittelt werden. Gleichzeitig kann der Status und die Ist-Ausgangsfrequenz aus dem Frequenzumrichter gelesen werden.
- In (S1+) wird die Stationsnummer des Frequenzumrichters (0 bis 31) angegeben.
- Der Inhalt von (S2+) gibt den Typ der Daten an, die in den Frequenzumrichter geschrieben und aus dem Frequenzumrichter gelesen werden sollen.
- (S3+) enthält die erste Adresse eines Operandenbereichs, in dem die Daten gespeichert sind, die zum Frequenzumrichter übertragen werden. (Der auf (S3+) folgende Operand wird ebenfalls belegt, siehe folgende Tabelle.)
- In (D+) wird der Operand angegeben, in dem die gelesenen Daten abgelegt werden. (Der auf (D+) folgende Operand wird ebenfalls belegt, siehe folgende Tabelle.)
- In (n) wird die Nummer der verwendeten Schnittstelle* angegeben (1: Ch. 1, 2: Ch. 2).

* Die Schnittstelle Ch. 2 steht bei einem FX3G-Grundgerät mit 14 oder 24 E/A und den Grundgeräten der FX3S-Serie nicht zur Verfügung.

Bei einem Grundgerät der FX3GE-Serie kann für n nur die Schnittstelle Ch. 2 angegeben werden, weil Ch. 1 von der integrierten Ethernet-Schnittstelle belegt wird.

Inhalt von (S2+) (Typ der Send- und Empfangsdaten)	Sendedaten (Daten, die in den Frequenzumrichter geschrieben werden)		Empfangsdaten (Daten, die aus dem Frequenzumrichter gelesen werden)	
	Daten 1 [(S3+)]	Daten 2 [(S3+)+1]	Daten 1 [(D+)]	Daten 2 [(D+)+1]
H0000	Betriebssignal (erweitert)	Ausgangsfrequenz (RAM)	Frequenzumrichter- Status (erweitert)	Ausgangsfrequenz (Drehzahl)
H0001				Sonderüberwachung
H0010		Ausgangsfrequenz (RAM, EEPROM)		Ausgangsfrequenz (Drehzahl)
H0011				Sonderüberwachung

Tab. 7-84: *Sende- und Empfangsdaten in Abhängigkeit vom Inhalt von (S2+)*

HINWEISE

Eine IVMC-Anweisung kann für die Frequenzumrichter der folgenden Serien verwendet werden:

- FR-E700 (ab Februar 2009)
- FR-D700 (alle Geräte)

Bitte prüfen Sie in der von Ihnen verwendeten Version der Programmier-Software, ob die IVMC-Anweisung unterstützt wird.

Für (S3+) und (D+) werden jeweils zwei aufeinanderfolgende Operanden belegt. Diese Operanden dürfen nicht von Anweisungen oder Anwendungen verwendet werden, die nicht im Zusammenhang mit der Steuerung des Frequenzumrichters stehen.

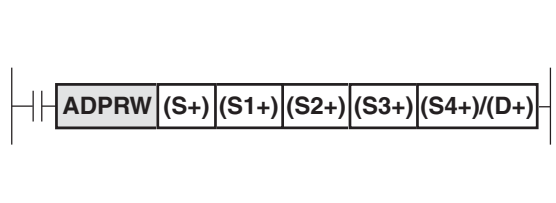
Wird, z.B. durch Indexvergabe, für (D+) ein Operand angegeben, der den zulässigen Bereich überschreitet, werden die aus dem Frequenzumrichter gelesenen Daten nicht ab (D+) gespeichert. Die in (S3+) und (S3+)+1 gespeicherten Daten können aber eventuell in den Frequenzumrichter geschrieben werden.

Falls in (S2+) ein undefinierter Wert angegeben wird, werden eventuell unzulässige Daten an den Frequenzumrichter übertragen und aus dem Frequenzumrichter gelesen. Die Inhalte von (D+) und (D+)+1 werden dadurch aktualisiert.

Mit einer IVMC-Anweisung wird der Frequenzumrichter-Status zum Zeitpunkt der Kommunikation mit dem Frequenzumrichter gelesen und in (D+) gespeichert. Dadurch kann der durch die IVMC-Anweisung eingestellte Status erst mit der nächsten oder einer späteren Lese-Anweisung (IVMC oder IVCK) erfasst werden.

7.21 Kommunikation über MODBUS

7.21.1 MODBUS Lesen/Schreiben (ADPRW)

						ADPRW		FNC 276				
						Aus Pufferspeicher von Sondermodulen lesen						CPU
Operanden	S+	S1+	S2+	S3+	S4+/D+	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	D ^① , R ^② , K, H				X, Y, M ^① , S, D ^① , R ^②		16 Bit	32 Bit	ADPRW	11		

① Sondermerker und -register können nicht angegeben werden.

② Nur bei FX3G/FX3GC/FX3GE/FX3U/FX3UC

Funktion

Die ADPRW-Anweisung ermöglicht den Datenaustausch des MODBUS-Masters mit Slave-Stationen (lesen und schreiben).

Beschreibung

- In (S+) wird die Adresse der Slave-Station angegeben (K0 bis K32). Bei Angabe der Adresse K0 wird das Kommando an alle Slaves gesendet (Broadcast).
- (S1+) enthält den Funktionscode.
- (S2+), (S3+) und (S4+)/(D+) enthalten die Parameter der Funktion.

Die folgende Tabelle zeigt zu jedem Funktionscode die erforderlichen Parameter.

Funktionscode	Parameter			
	(S1+)	(S2+)	(S3+)	(S4+)/D
1H Read coils (Status eines Ausgangs oder mehrerer Ausgänge lesen)	MODBUS-Adresse: 0000H bis FFFFH	Anzahl der zu lesenden Operanden: 1 bis 2000	Ziel der Daten in der SPS (Anfangsadresse)	
			Verfügbare Operanden	D, R, M, Y, S
			Blocklänge	$[(S3+)+15] \div 16^{①}$
2H Read discrete inputs (Status eines Eingangs oder mehrerer Eingänge lesen)	MODBUS-Adresse: 0000H bis FFFFH	Anzahl der zu lesenden Operanden: 1 bis 2000	Ziel der Daten in der SPS (Anfangsadresse)	
			Verfügbare Operanden	D, R, M, Y, S
			Blocklänge	$[(S3+)+15] \div 16^{①}$
3H Read holding registers (Status von Ausgangs-Registern wortweise lesen)	MODBUS-Adresse: 0000H bis FFFFH	Anzahl der zu lesenden Register: 1 bis 125	Ziel der Daten in der SPS (Anfangsadresse)	
			Verfügbare Operanden	D, R
			Blocklänge	(S3+)
4H Read input registers (Status von Eingangs-Registern wortweise lesen)	MODBUS-Adresse: 0000H bis FFFFH	Anzahl der zu lesenden Register: 1 bis 125	Ziel der Daten in der SPS (Anfangsadresse)	
			Verfügbare Operanden	D, R
			Blocklänge	(S3+)

Tab. 7-85: Funktionscodes und Parameter der ADPRW-Anweisung (1)

Funktionscode	Parameter				
	(S1+)	(S2+)	(S3+)	(S4+)/D	
5H Write Single Coil (Ein- oder Ausschalten eines einzelnen Ausgangs)	MODBUS-Adresse: 0000H bis FFFFH	0 (fest eingestellt)	Datenquelle in der SPS (Anfangsadresse)		
			Verfügbare Operanden	D, R, K, H, M, X, Y, S (D, R, M, X, Y und S können per Indexvergabe adressiert werden.)	
			Blocklänge	1 Operand	
6H Write Single Register (Schreiben eines Wertes in ein Ausgangs-Register)	MODBUS-Adresse: 0000H bis FFFFH	0 (fest eingestellt)	Datenquelle in der SPS (Anfangsadresse)		
			Verfügbare Operanden	D, R, K, H (D und R können per Indexvergabe adressiert werden.)	
			Blocklänge	1 Operand	
7H Read exception status ^② (Auslesen des Fehlerstatus)	0 (fest eingestellt)	0 (fest eingestellt)	Ziel der Daten in der SPS (Anfangsadresse)		
			Verfügbare Operanden	D, R	
			Blocklänge	1 Operand	
8H Diagnose ^②	Unterfunktionscode: 0H Return query data (Anforderung zurückschicken)	Daten zur Unterfunktion (Daten, die zurückgeschickt werden sollen (Schleifentest)): 0 bis 65535	Daten des Schleifentest (Der Slave sendet die Daten in (S3+) unverändert zurück.)		
			Verfügbare Operanden	D, R	
			Blocklänge	1 Operand	
	Unterfunktionscode: 1H Restart communication (Wiederaufnahme der Kommunikation)	Daten zur Unterfunktion 0000H: Kommunikationsereignisse nicht löschen FF00H: Kommunikationsereignisse löschen	Der Slave sendet die Daten in (S3+) unverändert zurück.		
			Verfügbare Operanden	D, R	
			Blocklänge	1 Operand	
	Unterfunktionscode: 2H Return diagnostic register (Diagnoseregister auslesen)	0 (fest eingestellt)	Ziel der Daten in der SPS (Anfangsadresse)		
			Verfügbare Operanden	D, R	
			Blocklänge	1 Operand	
	Unterfunktionscode: 3H Change ASCII input delimiter (Zweites Byte der Endekennung (LF (0AH)) im ASCII-Modus in festgelegte Daten wandeln)	Daten zur Unterfunktion Endekennung: 00H bis FFH	Der Slave sendet die Daten in (S3+) unverändert zurück.		
			Verfügbare Operanden	D, R	
			Blocklänge	1 Operand	
	Unterfunktionscode: 4H Force listen only mode (Slave in den Offline-Modus schalten)	0 (fest eingestellt)	0 (fest eingestellt)		
			Verfügbare Operanden	D, R	
Blocklänge			0		
Unterfunktionscode: AH Clear counters and diagnostic register (Zähler und Diagnoseregister löschen)	0 (fest eingestellt)	Der Slave sendet die Daten in (S3+) unverändert zurück.			
		Verfügbare Operanden	D, R		
		Blocklänge	1 Operand		
Unterfunktionscode: BH Return bus message count (Anzahl der über den Bus gesendeten Nachrichten an den Master übertragen)	0 (fest eingestellt)	Ziel der Daten in der SPS (Anfangsadresse)			
		Verfügbare Operanden	D, R		
		Blocklänge	1 Operand		

Tab. 7-85: Funktionscodes und Parameter der ADPRW-Anweisung (2)

Funktionscode	Parameter			
(S1+)	(S2+)	(S3+)	(S4+)/D	
8H Diagnose ^②	Unterfunktionscode: CH Return bus communication error count (Anzahl der Fehlermeldungen, die bei der Kommunikation aufgetreten sind, an den Master übertragen)	0 (fest eingestellt)	Ziel der Daten in der SPS (Anfangsadresse)	
			Verfügbare Operanden	D, R
			Blocklänge	1 Operand
	Unterfunktionscode: DH Return bus exception error count (Anzahl der Ausnahmefehler an den Master übertragen)	0 (fest eingestellt)	Ziel der Daten in der SPS (Anfangsadresse)	
			Verfügbare Operanden	D, R
			Blocklänge	1 Operand
	Unterfunktionscode: EH Return slave message count (Anzahl der vom Slave gesendeten Nachrichten (einschließlich der im Broadcast-Verfahren empfangenen Anforderungen) an den Master übertragen)	0 (fest eingestellt)	Ziel der Daten in der SPS (Anfangsadresse)	
			Verfügbare Operanden	D, R
Blocklänge			1 Operand	
Unterfunktionscode: FH Return slave no response count (Anzahl der empfangenen Broadcast-Mitteilungen an den Master übertragen)	0 (fest eingestellt)	Ziel der Daten in der SPS (Anfangsadresse)		
		Verfügbare Operanden	D, R	
		Blocklänge	1 Operand	
Unterfunktionscode: 10H Return slave NAK count (Angabe, wie oft vom Slave die Meldung „NAK“ gesendet wurde, an den Master übertragen)	0 (fest eingestellt)	Ziel der Daten in der SPS (Anfangsadresse)		
		Verfügbare Operanden	D, R	
		Blocklänge	1 Operand	
Unterfunktionscode: 11H Return slave busy count (Angabe, wie oft die Meldung „Slave ist beschäftigt“ gesendet wurde, an den Master übertragen)	0 (fest eingestellt)	Ziel der Daten in der SPS (Anfangsadresse)		
		Verfügbare Operanden	D, R	
		Blocklänge	1 Operand	
Unterfunktionscode: 12H Return bus character overrun count (Anzahl der Anforderungsnachrichten, die die zulässige Größe überschritten haben, an den Master übertragen)	0 (fest eingestellt)	Ziel der Daten in der SPS (Anfangsadresse)		
		Verfügbare Operanden	D, R	
		Blocklänge	1 Operand	
BH Get communications event counter ^② (Anzahl der Anforderungsnachrichten, die fehlerfrei ausgeführt wurden, an den Master übertragen)	0 (fest eingestellt)	0 (fest eingestellt)	Ziel der Daten in der SPS (Anfangsadresse) (D+): Status der Programmanweisung (D+)+1: Zähler für Kommunikationsereignisse	
			Verfügbare Operanden	D, R
			Blocklänge	2 Operanden

Tab. 7-85: Funktionscodes und Parameter der ADPRW-Anweisung (3)

Funktionscode	Parameter			
	(S1+)	(S2+)	(S3+)	(S4+)/D
CH Get communications event counter ^② (Anzahl der Anforderungsnachrichten, die fehlerfrei ausgeführt wurden, an den Master übertragen)	0 (fest eingestellt)	0 (fest eingestellt)	Ziel der Daten in der SPS (Anfangsadresse) (D+): Status der Programmanweisung (D+)+1: Zähler für Kommunikationsereignisse (D+)+2: Anzahl der über den Bus gesendeten Nachrichten (D+)+3: Länge des Speichers für Kommunikationsereignisse (D+)+4 bis (D+)+35: Bis zu 64 Byte Kommunikationsereignisse	
			Verfügbare Operanden	D, R
			Blocklänge	4 bis 36 Operanden
FH Write multiple coils (Mehrere Ausgänge ein- oder ausschalten)	MODBUS-Adresse: 0000H bis FFFFH	Anzahl der zu steuernden Ausgänge: 1 bis 1968	Datenquelle in der SPS (Anfangsadresse)	
			Verfügbare Operanden	D, R, K, H, M, X, Y, S (D, R, M, X, Y und S können per Indexvergabe adressiert werden.)
10H Write multiple registers (In mehrere Ausgangs-Register schreiben)	MODBUS-Adresse: 0000H bis FFFFH	Anzahl der zu beschreibenden Register: 1 bis 123	Datenquelle in der SPS (Anfangsadresse)	
			Verfügbare Operanden	D, R, K, H (D und R können per Indexvergabe adressiert werden.)
11H Report slave ID ^② (Informationen über die Slave-Station lesen)	0 (fest eingestellt)	0 (fest eingestellt)	Ziel der Daten in der SPS (Anfangsadresse) (D+): Typ des Slave (D+)+1: Betriebsart (RUN/STOP)	
			Verfügbare Operanden	D, R
			Blocklänge	2 Operanden
16H Mask write register ^② (Ausgangs-Register Bitweise UND-/ODER verknüpfen)	MODBUS-Adresse: 0000H bis FFFFH	UND-Maske: 0000H bis FFFFH	ODER-Maske: 0000H bis FFFFH	
			Verfügbare Operanden	D, R, K, H (D und R können per Indexvergabe adressiert werden.)
17H Read/write multiple registers ^② (Lesen und Schreiben aus bzw. in mehrere Ausgangs-Register)	MODBUS-Adresse (S2+): Startadresse der zu beschreibenden Ausgangs-Register (0000H bis FFFFH) (S2+)+1: Startadresse der zu lesenden Ausgangs-Register (0000H bis FFFFH)	(S3+): Anzahl der zu beschreibenden Register (1 bis 121) (S3+)+1: Anzahl der zu lesenden Register (1 bis 125)	Ziel der Daten in der SPS (Anfangsadresse) (S4+): Zu schreibende Daten 1 (S4+)+1: Zu schreibende Daten 2 (S4+)+(Anzahl der zu beschreibenden Register (S3+))-1: Zu schreibende Daten (S3+) (S4+)+(S3+): Gelesene Daten 1 (S4+)+(S3+)+1: Gelesene Daten 2 (S4+)+(S3+)+(Anzahl der zu lesenden Register ((S3+)+1))-1: Gelesene Daten ((S3+)+1)	
			Verfügbare Operanden	D, R
			Blocklänge	Anzahl der zu beschreibenden Register (S3+)+ Anzahl der zu lesenden Register ((S3+)+1)

Tab. 7-85: Funktionscodes und Parameter der ADPRW-Anweisung (4)

- ① Diese Formel gilt für die Operanden D und R.
- ② Dieser Funktionscode kann nur bei einem FX3U- oder FX3UC-Grundgerät angewendet werden.

7.22 Datenaustausch mit Sondermodulen

Symbol	FNC	Bedeutung	Abschnitt
RBFM	278	Aus Pufferspeicher von Sondermodulen lesen	7.22.1
WBFM	279	In Pufferspeicher von Sondermodulen schreiben	7.22.2

Tab. 7-86: Übersicht der Anweisungen zur Kommunikation mit Sondermodulen

Mit den hier beschriebenen Anweisungen RBFM und WBFM können, wie auch mit den FROM- und TO-Anweisungen, Daten zwischen dem SPS-Grundgerät und angeschlossenen Sondermodulen ausgetauscht werden. Im Gegensatz zur FROM/TO-Anweisung werden bei einer RBFM- oder WBFM-Anweisung die Daten in mehreren Programmzyklen übertragen.

HINWEIS

Die Adressierung von Sondermodulen und der Aufbau der Pufferspeicher und die Anweisungen FROM und TO sind im Abschnitt 7.3 beschrieben.

7.22.1 Aus Pufferspeicher von Sondermodulen lesen (RBFM)

						RBFM		FNC 278				
						Aus Pufferspeicher von Sondermodulen lesen						
						CPU	FX1s	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3s	FX3U FX3UC
												●
Operanden	n1	n2	D+	n3	n4	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	D, R, K, H		D*, R	D, R, K, H				16 Bit	32 Bit	RBFM		11
							●					

* Für (D+) können keine Sonderregister angegeben werden.

Funktion

Auslesen von Daten aus den Pufferspeichern der angeschlossenen Sondermodule. Die Übertragung der Daten kann auf mehrere Programmzyklen verteilt werden.

Beschreibung

- In (n1) wird die Adresse des Sondermoduls angegeben.
- (n2) gibt die erste Pufferspeicheradresse an, aus der gelesen werden soll.
- (n3) gibt die Anzahl der Datenworte an, die gelesen werden soll.
- (n4) gibt an, wie viele Datenworte in einem Programmzyklus übertragen werden sollen.
- Die erste Adresse des Operandenbereichs, in dem die gelesenen Daten im Grundgerät gespeichert werden sollen, wird in (D+) angegeben.
- Zulässige Bereiche:
 - n1 = 0 bis 7
 - n2 = 0 bis 32766
 - n3 = 1 bis 32767
 - n4 = 1 bis 32767

- Die Anzahl der zur Ausführung der RBFM- Anweisung benötigten Programmzyklen kann durch Division der in (n3) angegebenen gesamten Anzahl der Datenwörter durch die in (n4) angegebene Anzahl der Daten pro Zyklus berechnet werden:

$$\text{Anzahl Programmzyklen} = [n3]/[n4]$$

Falls bei der Division ein Rest entsteht, werden die verbleibenden Daten in einem zusätzlichen Programmzyklus übertragen.

- Die folgenden Sondermerker sollten bei der Ausführung einer RBFM-Anweisung im Programm überwacht und zur Verriegelung und Fehlermeldung verwendet werden:
 - M8029: Dieser Merker wird gesetzt, wenn die Ausführung der RBFM-Anweisung fehlerfrei abgeschlossen wurde.
 - M8328: Dieser Merker zeigt an, das eine RBFM- oder WBFM-Anweisung in einem anderen Programmschritt ausgeführt wird und dabei auf dasselbe Sondermodul zugegriffen wird. Die Ausführung weiterer Anweisungen mit derselben Pufferspeicheradresse wird solange ausgesetzt, bis die Bearbeitung dieser Anweisung abgeschlossen ist.
 - M8329: Wenn dieser Merker gesetzt wird, ist bei der Ausführung der RBFM-Anweisung ein Fehler aufgetreten

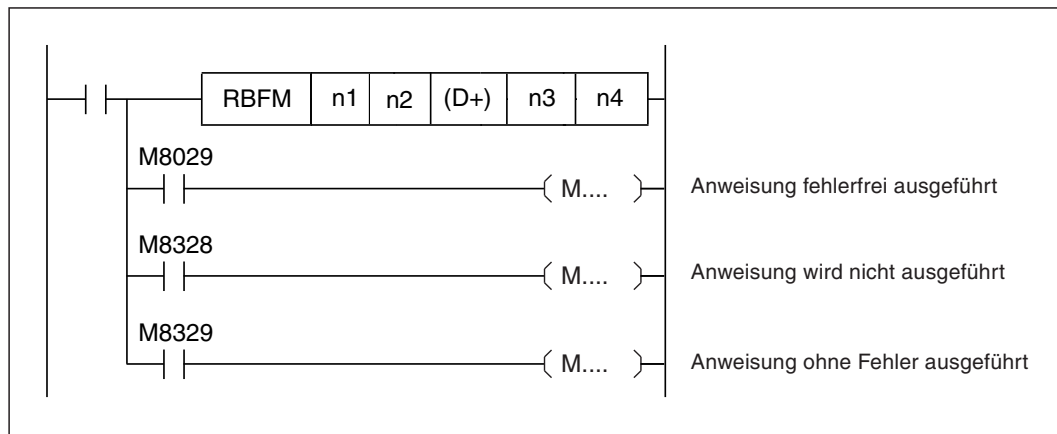


Abb. 7-295: Der Zustand der Merker M8029, M8328 und M8329 muss unmittelbar nach einer RBFM-Anweisung geprüft werden.

Fehlerquellen

Im folgenden Fall tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6708“ eingetragen:

- Das Sondermodul mit der in (n1) angegebenen Adresse existiert nicht.

7.22.2 In Pufferspeicher von Sondermodulen schreiben (WBFM)

					WBFM		FNC 279				
					In Pufferspeicher von Sondermodulen schreiben						
CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC					
						●					
Operanden	n1	n2	S+	n3	n4	Puls-Anweisung (P)		Verarbeitung		Programmschritte	
	D, R, K, H		D*, R	D, R, K, H				16 Bit	32 Bit	WBFM	
							●				

* Für (S+) können keine Sonderregister angegeben werden.

Funktion

Übertragen von Daten in dem Pufferspeicher von angeschlossenen Sondermodule. Die Übertragung der Daten kann auf mehrere Programmzyklen verteilt werden.

Beschreibung

- In (n1) wird die Adresse des Sondermoduls angegeben.
- (n2) gibt die erste Pufferspeicheradresse an, in die gelesen werden soll.
- (n3) gibt die Anzahl der Datenworte an, die übertragen werden soll.
- (n4) gibt an, wie viele Datenworte in einem Programmzyklus übertragen werden sollen.
- Die erste Adresse des Operandenbereichs, in dem die Daten im Grundgerät gespeichert sind, wird in (S+) angegeben.
- Zulässige Bereiche:
 n1 = 0 bis 7
 n2 = 0 bis 32766
 n3 = 1 bis 32767
 n4 = 1 bis 32767
- Die Anzahl der zur Ausführung der WBFM- Anweisung benötigten Programmzyklen kann durch Division der in (n3) angegebenen gesamten Anzahl der Datenwörter durch die in (n4) angegebene Anzahl der Daten pro Zyklus berechnet werden:

$$\text{Anzahl Programmzyklen} = \lceil n3 / n4 \rceil$$
 Falls bei der Division ein Rest entsteht, werden die verbleibenden Daten in einem zusätzlichen Programmzyklus übertragen.
- Die folgenden Sondermerker sollten bei der Ausführung einer WBFM-Anweisung im Programm überwacht und zur Verriegelung und Fehlermeldung verwendet werden:
 - M8029: Dieser Merker wird gesetzt, wenn die Ausführung der WBFM-Anweisung fehlerfrei abgeschlossen wurde.
 - M8328: Dieser Merker zeigt an, das eine RBFM- oder WBFM-Anweisung in einem anderen Programmschritt ausgeführt wird und dabei auf dasselbe Sondermodul zugegriffen wird. Die Ausführung weiterer Anweisungen mit derselben Pufferspeicheradresse wird solange ausgesetzt, bis die Bearbeitung dieser Anweisung abgeschlossen ist.
 - M8329: Wenn dieser Merker gesetzt wird, ist bei der Ausführung der WBFM-Anweisung ein Fehler aufgetreten.

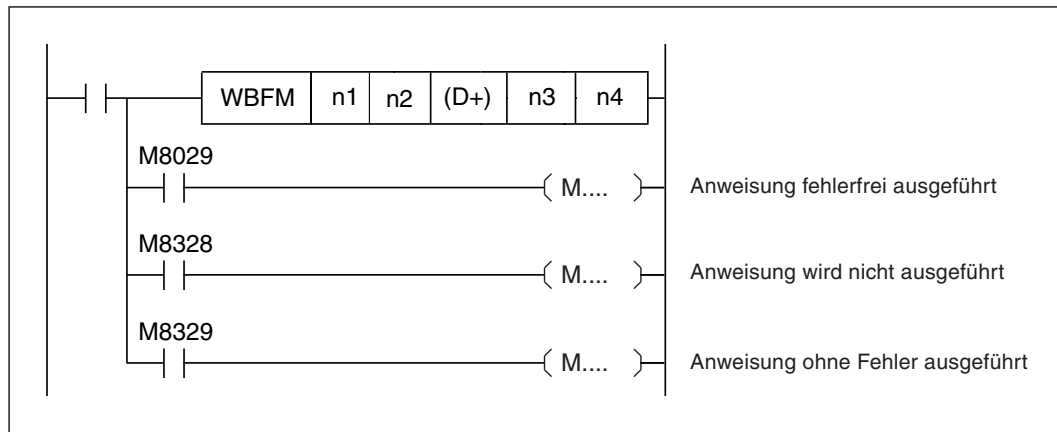


Abb. 7-297: Der Zustand der Merker M8029, M8328 und M8329 muss unmittelbar nach einer WBFM-Anweisung geprüft werden.

Fehlerquellen

Im folgenden Fall tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird der Fehlercode „6708“ eingetragen:

- Das Sondermodul mit der in (n1) angegebenen Adresse existiert nicht.

7.22.3 Hinweise zur Anwendung der RBFM- und WBFM-Anweisung

Auslösung des Watchdog-Timers

Auch wenn die Daten, die zwischen SPS-Grundgerät und Sondermodulen übertragen werden, aufgeteilt und in mehreren Programmzyklen übertragen werden, kann ein Watchdog-Timer-Fehler auftreten. In diesem Fall können die Einstellung des Watchdog-Timers oder die Parameter der RBFM- oder WBFM-Anweisung geändert werden.

- Änderung der Einstellung des Watchdog-Timers

Durch den Eintrag eines neuen Wertes in das Sonderregister D8000 wird die Überwachungzeit geändert. Diese Zeit ist auf 200 ms voreingestellt. Um die neue Zeit sofort zu aktivieren, sollte zusätzlich eine WDT-Anweisung ausgeführt werden. Ohne diese Anweisung wird die neue Einstellung des Watchdog-Timers erst im nächsten Programmzyklus gültig.

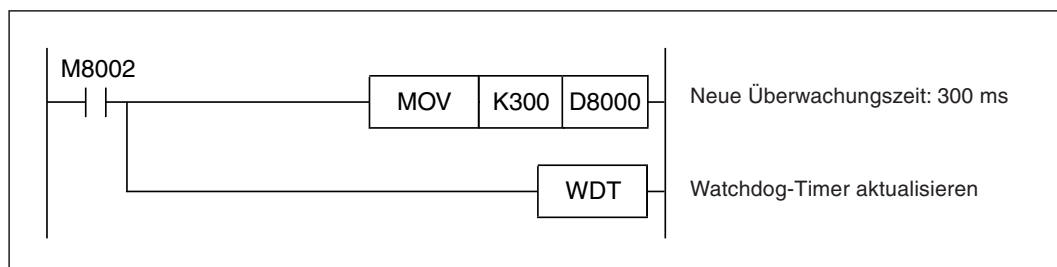


Abb. 7-296: Beispiel zur Änderung der Einstellung des Watchdog-Timers: Im ersten Programmzyklus nach dem Anlauf der SPS wird M8002 für einen Zyklus gesetzt und die Einstellung des Watchdog-Timers verändert.

- Änderung der Anzahl der pro Programmzyklus übertragenen Daten

(n4) gibt an, wie viele Datenworte in einem Programmzyklus übertragen werden sollen. Reduzieren Sie diesen Wert, damit der Watchdog-Timer nicht anspricht.

Unterbrechung einer RBFM- oder WBFM-Anweisung

Die Ausführung einer RBFM- oder WBFM-Anweisung darf nicht unterbrochen werden. Wird die Ausführung abgebrochen, werden nicht alle Daten übertragen. Die bis zur Unterbrechung erfassten Daten werden jedoch im Grundgerät bzw. im Pufferspeicher des Sondermoduls eingetragen.

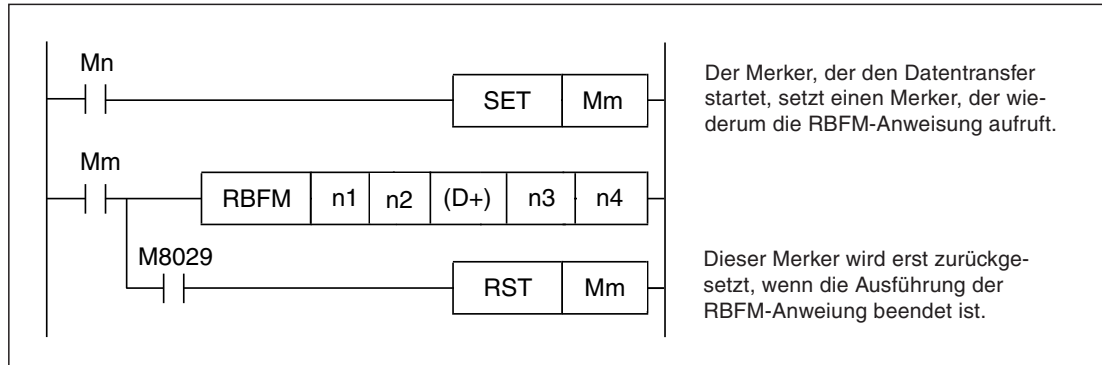


Abb. 7-298: In diesem Programmbeispiel wird durch einen zweiten Merker (Mm) verhindert, dass die RBFM-Anweisung nach dem Zurücksetzen des Startkommandos (Mn) nicht vollständig ausgeführt wird.

Verwendung von Index-Registern

Falls Index-Register verwendet werden, ist der Inhalt der Register beim Aufruf einer RBFM- oder WBFM-Anweisung maßgebend. Die Ausführung dieser Anweisungen wird nicht beeinflusst, wenn sich der Inhalt der Index-Register nach dem Start einer RBFM- oder WBFM-Anweisung ändert.

Speicherbereiche im Grundgerät für Daten vom und zum Sondermodul

Während der Ausführung einer RBFM-Anweisung werden Daten aus dem Pufferspeicher eines Sondermoduls in den Operandenbereich von ((D+)+0) bis ((D+)+(n3)) eingetragen. Die Inhalte dieser Operanden können erst ausgewertet werden, nachdem die Ausführung der RBFM-Anweisung abgeschlossen wurde. Dies wird durch M8029 angezeigt.

Eine WBFM-Anweisung liest Daten aus den Operandenbereich von ((S+)+0) bis ((S+)+(n3)) und überträgt sie in den Pufferspeicher eines Sondermoduls. Während der Ausführung der WBFM-Anweisung darf der Inhalt dieses Operandenbereichs nicht verändert werden, weil sonst eventuell die falschen Daten übertragen werden.

Gleichzeitiger Zugriff auf denselben Pufferspeicherbereich

Während der Ausführung einer RBFM-Anweisung sollten nicht mit einer WBFM-Anweisung Daten in denselben Pufferspeicherbereich des Sondermoduls übertragen werden. Die RBFM-Anweisung erfasst dadurch eventuell nicht die gewünschten Daten.

7.22.4 Programmbeispiel zur RBFM- und WBFM-Anweisung

In diesem Programmbeispiel werden Daten in den Pufferspeicher des Sondermoduls mit der Adresse 2 übertragen und aus dem Pufferspeicher dieses Moduls gelesen.

- Beim Einschalten des Eingangs X0 werden die Inhalte der Datenregister D100 bis D179 (80 Adressen) in die Pufferspeicheradressen 1001 bis 1080 übertragen. In jedem Programmzyklus werden 16 Pufferspeicheradressen beschrieben.
- Beim Einschalten von X1 werden die Inhalte der Pufferspeicheradressen 2001 bis 2080 (80 Adressen) in die Datenregister D200 bis D279 übertragen. In jedem Programmzyklus werden 16 Pufferspeicheradressen gelesen.

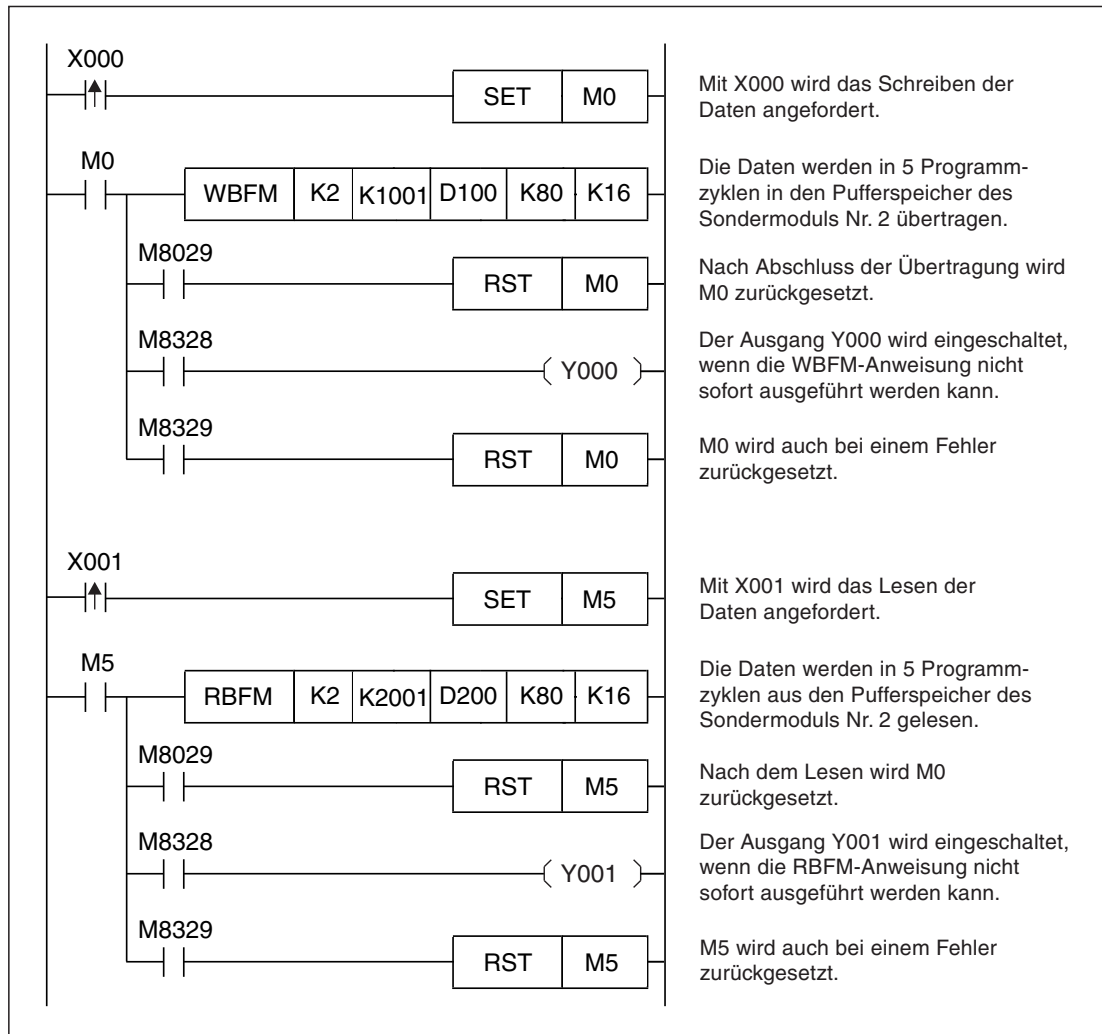


Abb. 7-299: Programmbeispiel zum Schreiben und Lesen von jeweils 80 Datenworten in und aus dem Pufferspeicher des Sondermoduls mit der Adresse 2.



7.23 Anweisung für High-Speed-Counter

						DHSCT		FNC 280				
						Istwert eines High-Speed-Counters mit Daten in Tabellen vergleichen						
						CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
												●
Operanden	S1+	n1	S2+	D+	n2	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	D, R	K, H	C*	Y, M, S	K, H			16 Bit	32 Bit	DHSC		21
									●			

* Für (S2+) kann nur ein High-Speed-Counter (C235 bis C255) angegeben werden.

Funktion

Abhängig vom Vergleich des Istwerts eines High-Speed-Counters mit bis zu 128 Sollwerten werden bis zu 16 Bit-Operanden gesetzt oder zurückgesetzt.

Beschreibung

- Mit (S1+) wird die erste Adresse des Operandenbereichs angegeben, in dem die Vergleichswerte und das zugehörige Bit-Muster für die Ausgabe abgelegt sind. Für jeden Vergleichswert werden drei 16-Bit-Operanden benötigt.
- (n1) gibt die Anzahl der Vergleichswerte an. Die Größe des mit (S1+) angegebenen Operandenbereichs kann mit der Formel „(n1) x 3“ berechnet werden. Für (n1) können Werte von 1 bis 128 angegeben werden.
- Der High-Speed-Counter, dessen Istwert verglichen werden soll, wird mit (S2+) angegeben.
- In (D+) wird der erste Operand des Bereichs angegeben, der durch den Vergleich beeinflusst werden soll.
- (n2) gibt die Anzahl der Bit-Operanden an, die ab (D+) durch den Vergleich gesetzt oder zurückgesetzt werden. (n2) kann Werte zwischen 1 und 16 annehmen.
- Bei der Ausführung der DHSCT-Anweisung wird der Istwert des in (S2+) angegebenen Zählers nacheinander mit den ab (S1+) eingetragenen Sollwerten verglichen. Bei einer Übereinstimmung zwischen Soll- und Istwert wird ein Bitmuster, das ebenfalls in dem mit (S1+) angegebenen Operandenbereich gespeichert ist, an die durch (D+) und (n2) definierten Operanden ausgegeben.

Tabelleneintrag	Vergleichswert	Bitmuster	Beeinflusste Operanden
0	(S1+)+1, (S1+)	(S1+)+2	(D+) bis [(D+)+(n2-1)]
1	(S1+)+4, (S1+)+3	(S1+)+5	
2	(S1+)+7, (S1+)+6	(S1+)+8	
:		:	
m-2	(S1+)+(3m-5), (S1+)+(3m-6)	(S1+)+(3m-4)	
m-1	(S1+)+(3m-2), (S1+)+(3m-3)	(S1+)+(3m-1)	

Tab. 7-87: Zusammenhang zwischen den Operanden der DHSCT-Anweisung

Beispiel ▾

Für (D+) wurde der Ausgang Y0, für (n2) der Wert 10 und als Bitmuster „A716H“ angegeben.

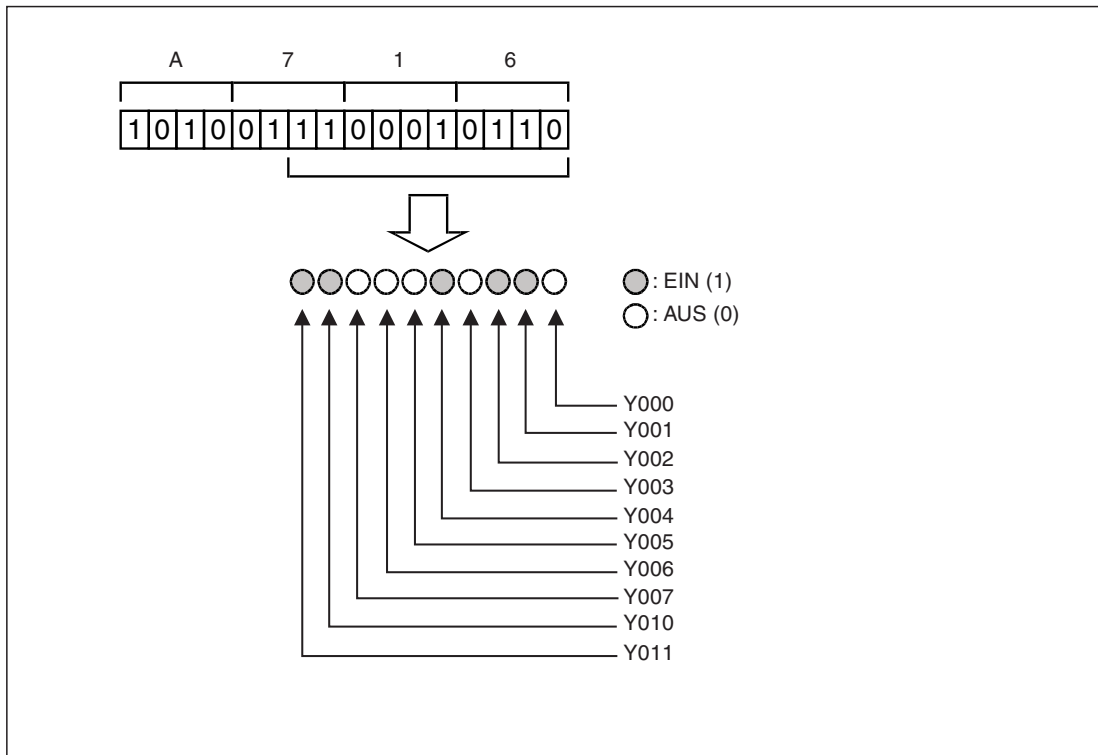


Abb. 7-300: Wenn der Istwert dem angegebenen Sollwert entspricht, werden die Ausgänge Y0 bis Y11 so geschaltet, wie es durch das Bitmuster vorgegeben

△

- Wird für (D+) ein Ausgang (Y) angegeben, erfolgt die Aktualisierung des Ausgangszustands unmittelbar bei der Ausführung der DHSCT-Anweisung. (Normalerweise werden Ausgänge erst am Ende eines Programmzyklus aktualisiert.) Die niederwertigste Adresse des Ausgangs muss in diesem Fall „0“ sein (z. B. Y000, Y010, Y020 usw.).
- Es wird nur jeweils ein Vergleichswert mit dem Istwert des Zählers verglichen. Der Vergleich mit dem nächste Tabelleneintrag erfolgt erst, wenn der Zähleristwert mit dem aktuellen Vergleichswert übereinstimmt.

Falls z. B. der Istwert eines aufwärtszählenden Zähler verglichen werden soll, muss die DHSCT-Anweisung gestartet werden, solange der Istwert kleiner als der Vergleichswert im ersten Tabelleneintrag ist. Das Beispiel auf der Seite 7-307 verdeutlicht diese Zusammenhänge.

- Nach einer Übereinstimmung zwischen Vergleichs- und Istwert wird das Sonderregister D8138, das als Tabellenzeiger dient, um den Wert „1“ erhöht. Danach wird der Istwert mit dem nächsten Wert verglichen. Dies wiederholt sich so lange, bis der Inhalt von D8138 den mit (n2) vorgegebenen Wert erreicht. In diesem Fall wird der Merker M8138 gesetzt, um anzuzeigen, dass alle Vergleiche ausgeführt wurden. Der Inhalt von D8138 wird anschließend auf „0“ gesetzt und der Soll-/Istwertvergleich beginnt wieder beim ersten Vergleichswert.
- Wird die Eingangsbedingung der DHSCT-Anweisung ausgeschaltet, werden keine Vergleiche mehr ausgeführt und der Tabellenzeiger D8138 auf „0“ gesetzt.

HINWEISE

In einem Programm kann nur eine DHSCT-Anweisung ausgeführt werden. Falls eine weitere DHSCT-Anweisung gestartet wird, tritt ein Fehler mit dem Code 6765 auf und die Anweisung wird nicht ausgeführt.

Im ersten Programmzyklus, in dem eine DHSCT-Anweisung ausgeführt wird, werden interne Verknüpfungen ausgeführt. Dadurch werden die Ausgangszustände erst ab dem zweiten Programmzyklus beeinflusst.

Bis zu 32 Anweisungen für High-Speed-Counter (DHSCT, DHSCS, DHSCR und DHSZ) können in einem Programm gleichzeitig aktiv sein. Beim Start der 33. Anweisung tritt ein Fehler mit dem Code 6705 auf und die Anweisung wird nicht ausgeführt.

Wird der Zähler in (S2+) durch ein Index-Register angegeben, werden alle High-Speed-Counter als Software-Zähler behandelt.

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird ein Fehlercode eingetragen:

- Für (S2+) wurde kein High-Speed-Counter (C235 bis C255) angegeben. (Fehlercode „6706“)
- Durch die Angabe von (n1) wird der zulässige Bereich für den in (S1+) angegebenen Operandentyp überschritten. (Fehlercode „6706“)
- Durch die Angabe von (n2) wird der zulässige Bereich für den in (D+) angegebenen Operandentyp überschritten. (Fehlercode „6706“)
- Ein Programm enthält mehrere DHSCT-Anweisungen. (Fehlercode „6765“)
- Ein Programm enthält mehr als 32 Anweisungen für High-Speed-Counter, wie z. B. DHSCT, DHSCS, DHSCR oder DHSZ. (Fehlercode „6706“)

Beispiel ▾

Mit dem folgenden Programm werden abhängig vom Istwert des Zählers C235 die Ausgänge Y010 bis Y012 geschaltet. Der Istwert wird mit 5 Werten verglichen, die mit den entsprechenden Bitmustern für die Ausgänge in der SPS ab dem Datenregister D200 gespeichert sind.

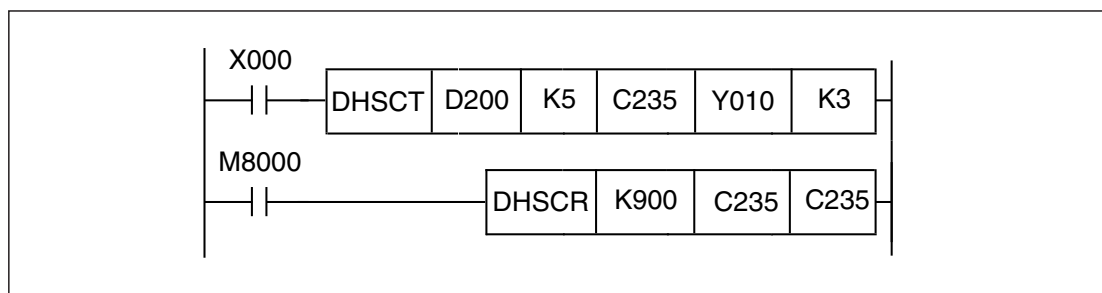


Abb. 7-301: Wenn C235 den Wert „900“ erreicht, wird dieser Zähler zurückgesetzt.

Tabelleneintrag	Vergleichswert		Bitmuster		Tabellenzeiger D8138
	Operand	Inhalt	Operand	Inhalt	
0	D201, D200	321	D202	0001H	0
1	D204, D203	432	D205	0007H	1
2	D207, D206	543	D208	0002H	2
3	D210, D209	764	D211	0000H	3
4	D213, D212	800	D214	0003H	4

Tab. 7-88: Vergleichswerte und Bitmuster für dieses Beispiel

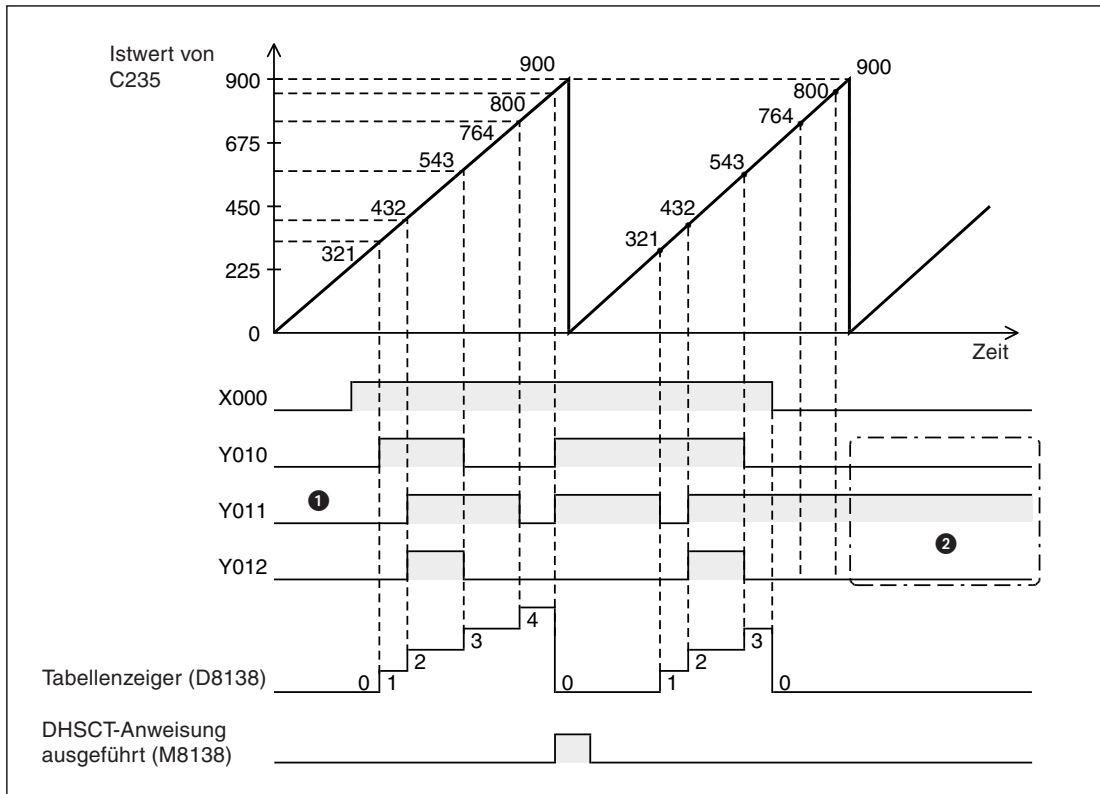


Abb. 7-302: Signalverläufe für das Programmbeispiel

- ① Solange die DHSC-Anweisung nicht ausgeführt wird, werden die Ausgänge nicht angesteuert.
- ② Weil diese DHSC-Anweisung nicht mehr ausgeführt wird, werden die Zustände der Ausgänge Y10, Y11 und Y12 durch diese Anweisung nicht mehr verändert.



7.24 Anweisungen für erweiterte File-Register

Symbol	FNC	Bedeutung	Abschnitt
LOADR	290	Daten aus erweiterten File-Registern lesen	7.24.1
SAVER	291	Daten in erweiterte File-Register schreiben	7.24.2
INITR	292	Erweiterte Register und erweiterte File-Register initialisieren	7.24.3
LOGR	293	Werte von Operanden in erweiterte Register oder erweiterte File-Register speichern	7.24.4
RWER	294	Daten aus erweiterte Register in erweiterte File-Register übertragen	7.24.5
INITER	295	Erweiterte File-Register initialisieren	7.24.6

Tab. 7-89:Übersicht der Anweisungen für erweiterte File-Register

7.24.1 Daten aus erweiterten File-Registern lesen (LOADR)

		LOADR	FNC 290					
		Daten aus erweiterte File-Register lesen						
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
						●		●
Operanden	S+	n	Puls-Anweisung (P)		Verarbeitung		Programmschritte	
	R	D, K, H	●		16 Bit	32 Bit	LOADR LOADRP	5

Funktion

Datentransfer aus erweiterten File-Registern (ER) einer Speicherkassette (Flash-EEPROM bei FX3U/FX3UC, EEPROM bei FX3G) in erweiterte Register (R) im RAM der SPS.

Beschreibung

- In (S+) wird die erste Adresse der erweiterten Register angegeben, ab der die Daten gespeichert werden sollen. Die Daten werden aus den erweiterten File-Registern ab derselben Adresse gelesen.
- (n) gibt die Anzahl der Adressen an, deren Inhalte übertragen werden sollen. Für (n) können Werte von 0 bis 32767 (1 bis 24000 bei FX3G) angegeben werden. Wird bei einer FX3U/FX3UC der Wert „0“ vorgegeben, werden 32767 Adressen transferiert.

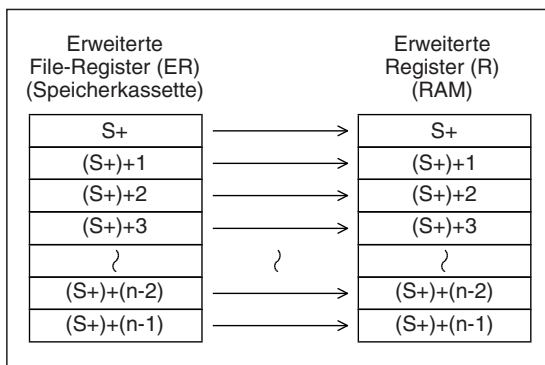


Abb. 7-303:

Eine LOADR-Anweisung transferiert Daten aus erweiterten File-Registern in erweiterte Register

- Im Gegensatz zu den Anweisungen SAVER, INITR und LOGR müssen bei einer LOADR-Anweisung keine Sektoren angegeben werden.

HINWEIS

Wird eine LOADR-Anweisung von einer FX3G-SPS ausgeführt, in der keine Speicherkassette installiert ist, werden die Daten aus den erweiterten File-Registern (ER) gelesen, die im internen EEPROM des Grundgeräts abgelegt sind. Auch in diesem Fall erfolgt das Lesen ab der in (S+) angegebenen Adresse.

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird ein Fehlercode eingetragen:

- Die Adresse der zu übertragenden Register überschreitet den Wert „32767“. (Fehlercode „6706“) In diesem Fall werden die Inhalte der Register bis R32767 transferiert.
- Es ist keine Speicherkassette installiert. (Fehlercode „6771“) Dieser Fehler wird nur bei einer FX3U oder FX3UC gemeldet.

Beispiel ▾

Mit dem folgenden Programm werden aus einer Speicherkassette die Inhalte der erweiterten File-Register ER1 bis ER4000 in die erweiterten Register R1 bis R4001 im RAM der SPS übertragen.

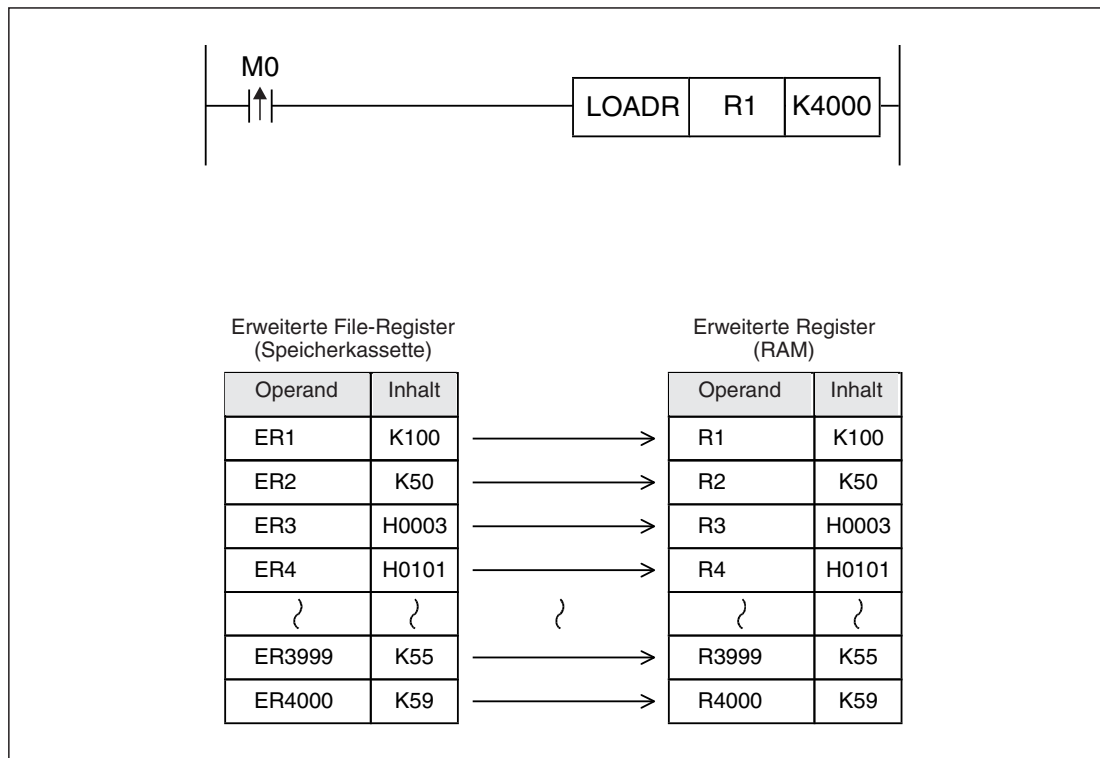
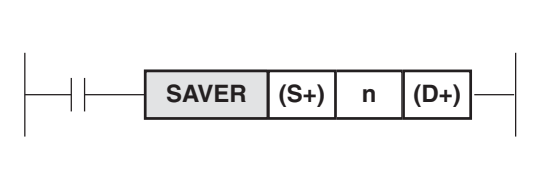


Abb. 7-304: Wenn M0 gesetzt wird, werden die Inhalte der erweiterten File-Register ER1 bis ER4001 in die erweiterten Register übertragen.



7.24.2 Daten in erweiterte File-Register schreiben (SAVER)

				SAVER		FNC 291					
				Datentransfer in erweiterte File-Register							
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
										●	
Operanden	S+	n	D+	Puls-Anweisung (P)	Verarbeitung		Programmschritte				
	R	K, H	D			16 Bit	32 Bit	SAVER		7	
						●					

Funktion

Datentransfer aus erweiterte Register (R) im RAM der SPS in erweiterte File-Register (ER) in einer Speicherkassette (Flash-EEPROM). Es werden die Inhalte von 2048 Registern (1 Sektor) übertragen.

Beschreibung

- In (S+) wird die erste Adresse des Registerbereichs (Sektors) angegeben, der übertragen werden soll.

Angabe für (S+)	Sektor	Übertragener Bereich
R0	0	R0 bis R2047
R2048	1	R2048 bis R4095
R4096	2	R4096 bis R6143
R6144	3	R6144 bis R8191
R8192	4	R8192 bis R10239
R10240	5	R10240 bis R12287
R12288	6	R12288 bis R14335
R14336	7	R14336 bis R16383
R16384	8	R16384 bis R18431
R18432	9	R18432 bis R20479
R20480	10	R20480 bis R22527
R22528	11	R22528 bis R24575
R24576	12	R24576 bis R26623
R26624	13	R26624 bis R28671
R28672	14	R28672 bis R30719
R30720	15	R30720 bis R32767

Tab. 7-90: Durch die Angabe einer Anfangsadresse wird der Sektor festgelegt.

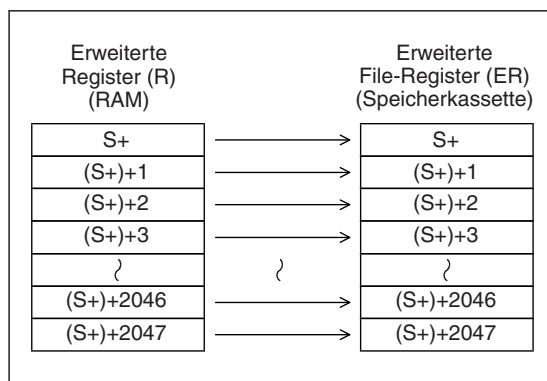


Abb. 7-305:

Eine SAVER-Anweisung transferiert die Inhalte von 2048 erweiterten Registern in erweiterte File-Register

- (n) gibt an, wie viele Registerinhalte in einem Programmzyklus übertragen werden sollen. Für (n) können Werte von 0 bis 2048 angegeben werden. Wird der Wert „0“ vorgegeben, werden in einem Zyklus 2048 Adressen transferiert.

Die Anzahl der zur Ausführung der SAVER-Anweisung benötigten Programmzyklen kann durch Division der 2048 übertragenen Register durch die in (n) angegebene Anzahl der Daten pro Zyklus berechnet werden:

$$\text{Anzahl Programmzyklen} = 2048/[n]$$

Falls bei der Division ein Rest entsteht, werden die verbleibenden Daten in einem zusätzlichen Programmzyklus übertragen.

- In (D+) wird die Anzahl der bereits übertragenen Register eingetragen.

Hinweise zur Anwendung einer SAVER-Anweisung

- Alternativ kann auch eine RWER-Anweisung zur Übertragung der Daten verwendet werden (siehe Abschnitt 7.24.5). Bei der RWER-Anweisung besteht keine Beschränkung auf Sektoren und es kann eine beliebige Anzahl Register übertragen werden.
- Das Schreiben von 2048 Registern in eine Flash-EPROM-Speicherkassette dauert ca. 50 bis 60 ms. Falls durch die Vorgabe für (n) eine große Anzahl Daten in einem Programmzyklus übertragen wird (z. B. bei Vorgabe der Werte „0“ oder „2048“), kann der Watchdog-Timer ansprechen. Wählen Sie in diesem Fall für (n) einen Wert zwischen „1“ und „1024“ und übertragen Sie dadurch die Daten in mehreren Zyklen.
- Der Sondermerker M8029 wird gesetzt, wenn die Ausführung der SAVER-Anweisung fehlerfrei abgeschlossen wurde. Dieser Merker wird aber auch durch andere Anweisungen gesteuert. Deshalb muss der Zustand des Merkers M8029 unmittelbar nach einer SAVER-Anweisung geprüft werden.
- Eine SAVER-Anweisung kann nur ausgeführt werden, wenn vorher die in der SAVER-Anweisung angegebenen Register mit einer INITER- oder INITR-Anweisung initialisiert wurden. Ohne diese Initialisierung tritt beim Start der SAVER-Anweisung ein Fehler mit dem Code 6770 auf.
- Die Ausführung einer SAVER-Anweisung darf nicht unterbrochen werden. Wird die Ausführung abgebrochen, werden eventuell fehlerhafte Daten in die erweiterten File-Register eingetragen.

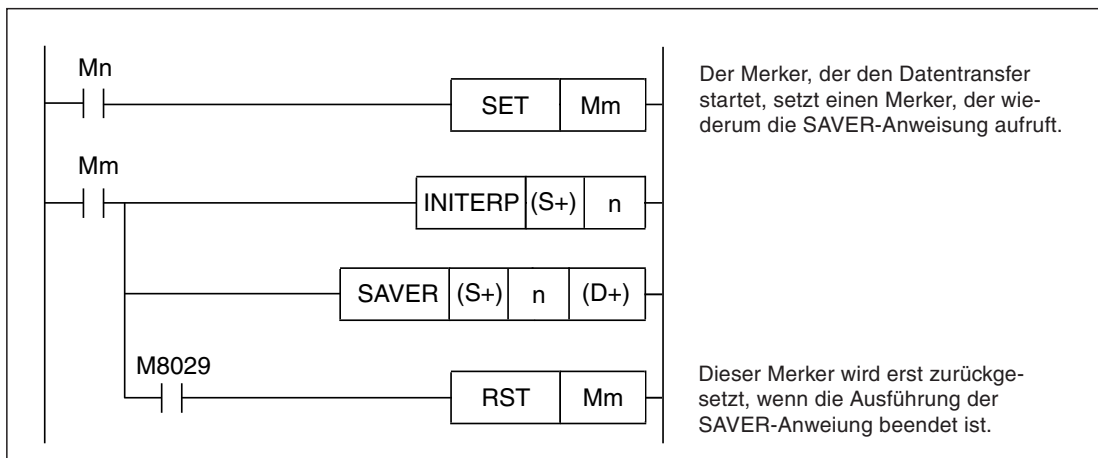


Abb. 7-306: In diesem Programmbeispiel wird durch einen zweiten Merker (Mm) verhindert, dass die SAVER-Anweisung nach dem Zurücksetzen des Startkommandos (Mn) nicht vollständig ausgeführt wird.

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird ein Fehlercode eingetragen:

- In (S+) wurden andere Operanden angegeben als die Anfangsadressen der Sektoren. (Fehlercode „6706“)
- Es ist keine Speicherkassette installiert. (Fehlercode „6771“)
- Der Schreibschutz der Speicherkassette ist aktiviert. (Fehlercode „6770“)
- Die Daten konnten nicht übertragen werden, weil z. B. die erweiterten File-Register nicht initialisiert worden sind. (Fehlercode „6770“)

Wenn dieser Fehler auftritt, gehen die Daten in den erweiterten Registern (R) verloren. Um dies zu vermeiden, sollten die Inhalte der erweiterten Registern vor der Ausführung einer SAVER-Anweisung mit Hilfe der Programmier-Software GX Developer, GX IEC Developer oder GX Works2 gesichert werden.

Beispiel ▾

Das folgende Programm wird verwendet, um die geänderten Inhalte der erweiterten Register R10 bis R19 (Sektor 0) zur Datensicherung in die erweiterten File-Register zu übertragen. Wenn der Eingang X0 eingeschaltet wird, werden in jedem Programmzyklus 128 Register transferiert.

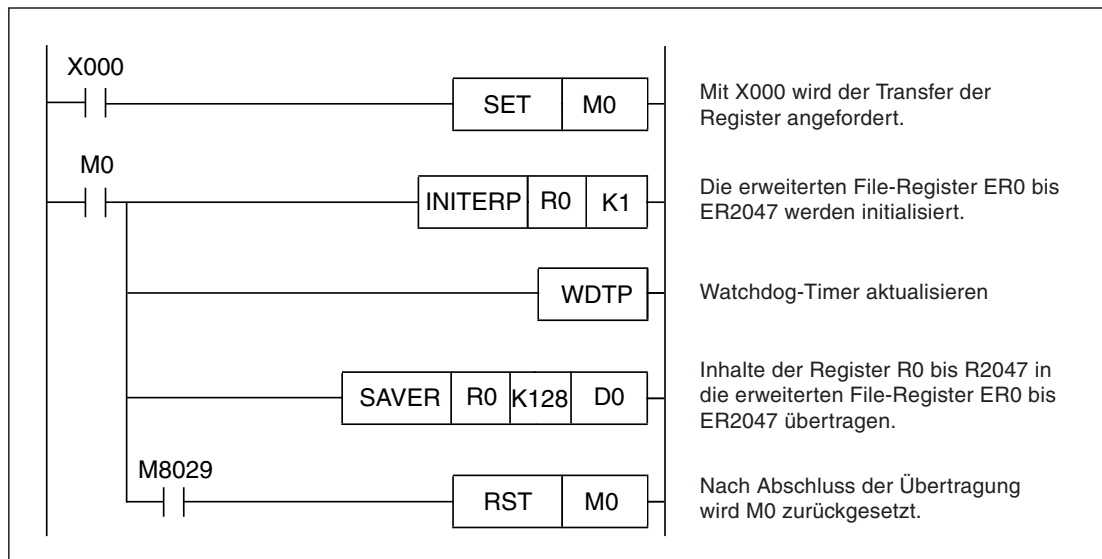


Abb. 7-307: Vor der Ausführung einer SAVER-Anweisung müssen die erweiterten File-Register mit einer INITER-Anweisung initialisiert werden.

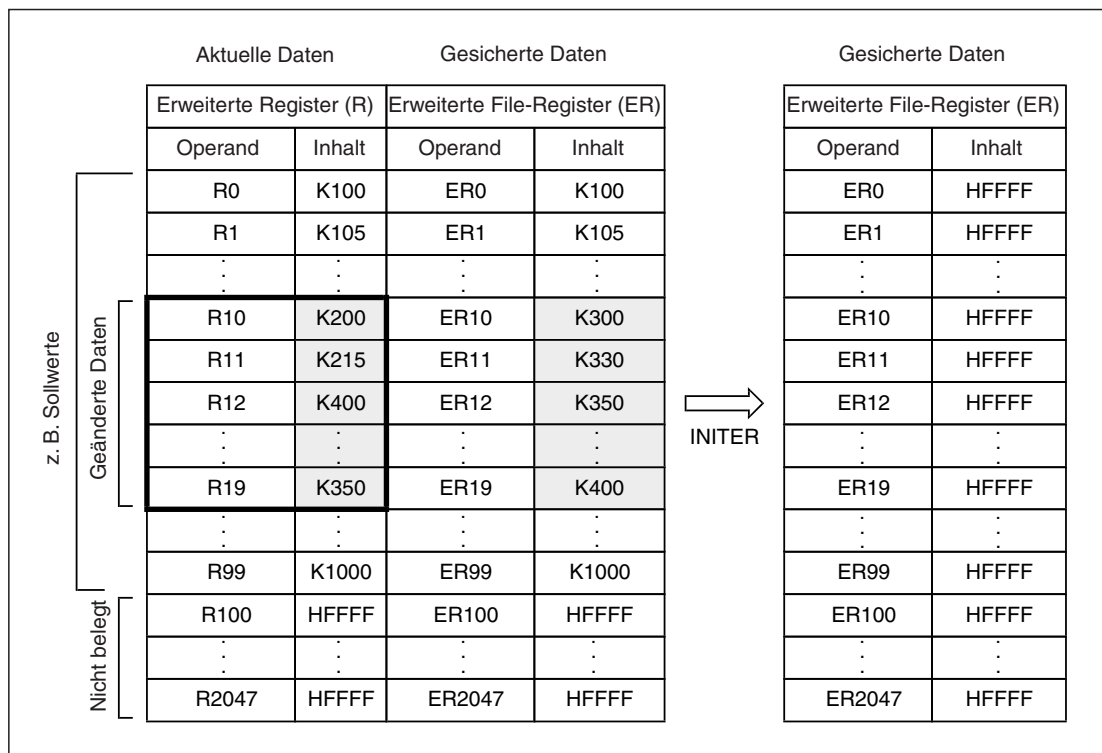


Abb. 7-308: Durch die INTR-Anweisung wird in alle erweiterten File-Register des Sektors 0 der Wert FFFFH eingetragen.

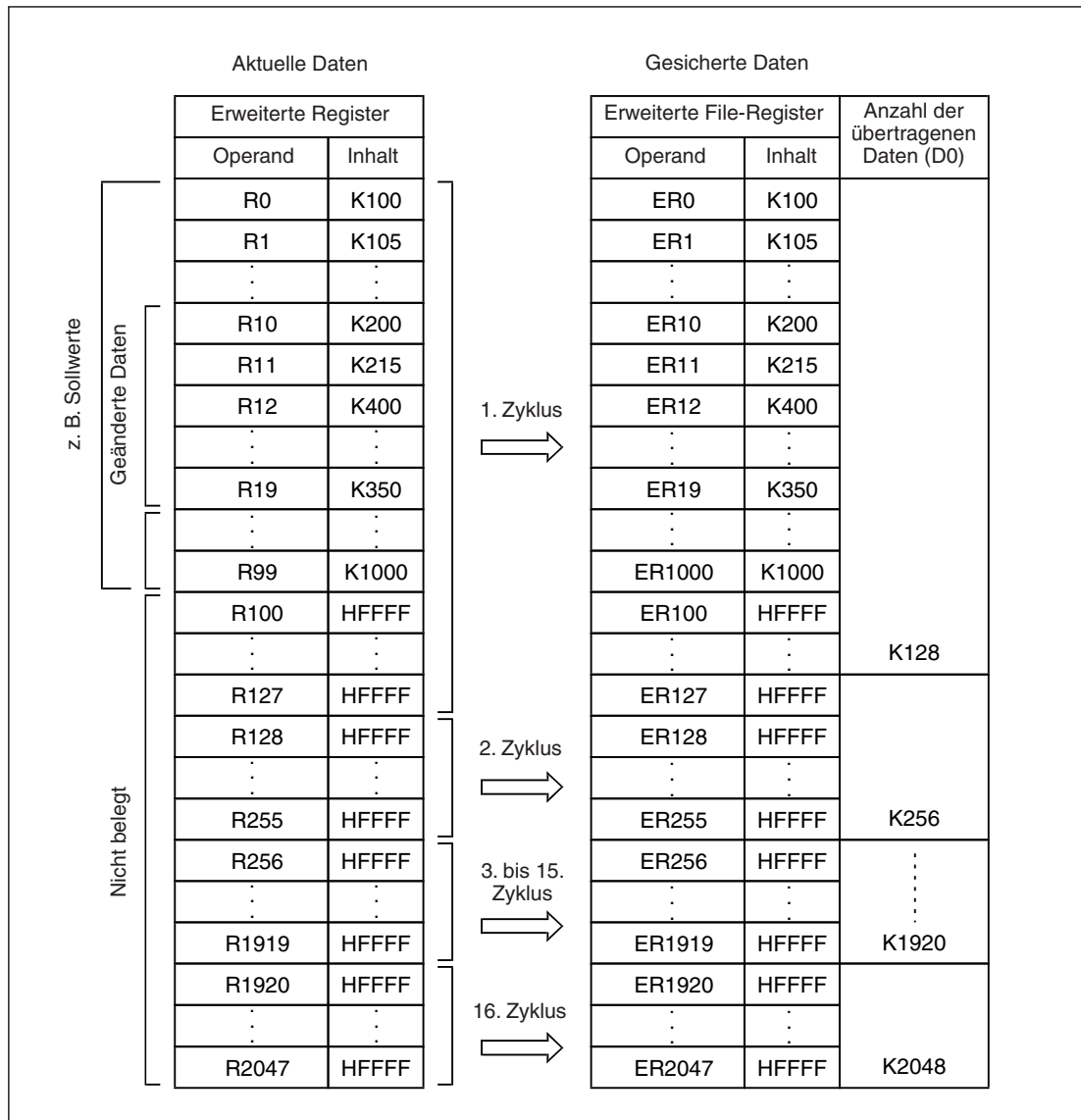


Abb. 7-309: Mit der SAVER-Anweisung werden anschließend die Inhalte der erweiterten Register in die Speicherkassette übertragen.



7.24.3 Erweiterte Register und erweiterte File-Register initialisieren (INITR)

			INITR		FNC 292				
			Erweiterte Register/File-Register initialisieren						
			CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
									●
Operanden	S+	n	Puls-Anweisung (P)	Verarbeitung		Programmschritte			
	R	D, K, H	●	16 Bit	32 Bit	INITR	INITRP	5	

Funktion

Initialisierung von erweiterten Registern (R) und erweiterten File-Registern (ER). Dabei wird in die Register der Wert „FFFFH“ eingetragen.

Beschreibung

- Die INITR-Anweisung trägt in erweiterte Register, die sich im RAM des SPS-Grundgeräts befinden, und in erweiterte File-Register, die in einer Speicherkassette (Flash-EPROM) abgelegt sind, zur Initialisierung den Wert „FFFFH“ (entspricht dezimal „-1“) ein. Eine Initialisierung der erweiterten Register muss vorgenommen werden, bevor diese Register, z. B. mit einer LOGR-Anweisung, beschrieben werden.
- Falls in der SPS keine Speicherkassette installiert ist, werden mit der INITR-Anweisung nur die erweiterten Register im RAM der CPU initialisiert.
- Die Initialisierung erfolgt sektorweise. Pro Sektor werden 2048 Register initialisiert. Mit einer INITR-Anweisung können mehrere Sektoren initialisiert werden.
- In (S+) wird die erste Adresse des ersten Registerbereichs (Sektors) angegeben, der initialisiert werden soll.

Angabe für (S+)	Sektor	Initialisierter Bereich	
		Erweiterte Register	Erweiterte File-Register
R0	0	R0 bis R2047	ER0 bis ER2047
R2048	1	R2048 bis R4095	ER2048 bis ER4095
R4096	2	R4096 bis R6143	ER4096 bis ER6143
R6144	3	R6144 bis R8191	ER6144 bis ER8191
R8192	4	R8192 bis R10239	ER8192 bis ER10239
R10240	5	R10240 bis R12287	ER10240 bis ER12287
R12288	6	R12288 bis R14335	ER12288 bis ER14335
R14336	7	R14336 bis R16383	ER14336 bis ER16383
R16384	8	R16384 bis R18431	ER16384 bis ER18431
R18432	9	R18432 bis R20479	ER18432 bis ER20479
R20480	10	R20480 bis R22527	ER20480 bis ER22527
R22528	11	R22528 bis R24575	ER22528 bis ER24575
R24576	12	R24576 bis R26623	ER24576 bis ER26623
R26624	13	R26624 bis R28671	ER26624 bis ER28671
R28672	14	R28672 bis R30719	ER28672 bis ER30719
R30720	15	R30720 bis R32767	ER30720 bis ER32767

Tab. 7-91: Durch die Angabe einer Anfangsadresse wird der erste Sektor festgelegt.

- In (n) wird die Anzahl der zu initialisierenden Sektoren angegeben.

Operand	Erweiterte Register		Erweiterte File-Register	
	Inhalt vor Ausführung der INITR-Anweisung	Inhalt nach Ausführung der INITR-Anweisung	Inhalt vor Ausführung der INITR-Anweisung	Inhalt nach Ausführung der INITR-Anweisung
(S+)	0010H	FFFFH	1234H	FFFFH
(S+)+1	0020H	FFFFH	5678H	FFFFH
(S+)+2	0011H	FFFFH	90ABH	FFFFH
:	:	:	:	:
(S+)+(2048 x n)-1	ABCDH	FFFFH	CDEFH	FFFFH

Tab. 7-92: Durch eine INITR-Anweisung werden erweiterte Register und erweiterte File-Register mit dem Wert FFFFH beschrieben.

Hinweise zur Anwendung der INITR-Anweisung

- Mit einer INITR-Anweisung werden gleichzeitig die erweiterten Registern (R) und die erweiterten File-Registern (ER) initialisiert. Verwenden Sie eine INITER-Anweisung (Abschnitt 7.24.6), wenn nur erweiterte File-Register in einer Speicherkassette initialisiert werden sollen.
 - Die Initialisierung eines Sektors (2048 Register) einer Speicherkassette dauert ca. 18 ms. Dadurch kann ein Watchdog-Timer-Fehler auftreten. (Falls keine Speicherkassette installiert ist und nur erweiterte Register im RAM initialisiert werden, wird für die Initialisierung eines Sektors max. 1 ms benötigt.) In diesem Fall können die Einstellung des Watchdog-Timers oder die Programmsequenz mit der INITR-Anweisung geändert werden.
- Änderung der Einstellung des Watchdog-Timers
 Durch den Eintrag eines neuen Wertes in das Sonderregister D8000 wird die Überwachungszeit geändert. Diese Zeit ist auf 200 ms voreingestellt. Um die neue Zeit sofort zu aktivieren, sollte zusätzlich eine WDT-Anweisung ausgeführt werden. Ohne diese Anweisung wird die neue Einstellung des Watchdog-Timers erst im nächsten Programmzyklus gültig.

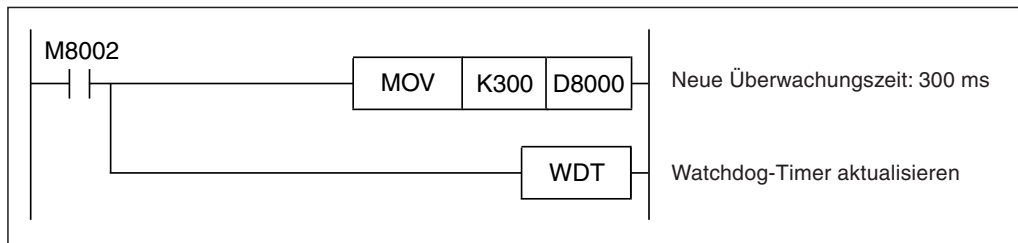


Abb. 7-310: Beispiel zur Änderung der Einstellung des Watchdog-Timers: Im ersten Programmzyklus nach dem Anlauf der SPS wird M8002 für einen Zyklus gesetzt und die Einstellung des Watchdog-Timers verändert.

- Aktualisierung des Watchdog-Timers vor und nach Ausführung der INITR-Anweisung

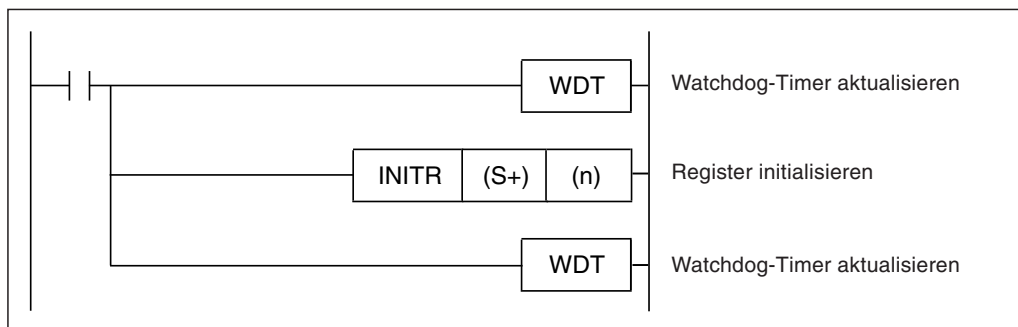


Abb. 7-311: Durch die Aktualisierung des Watchdog-Timers wird verhindert, dass bei der Initialisierung ein Fehler auftritt.

- In eine Speicherkassette können ca. 10.000 mal Daten eingetragen werden. Jede Ausführung einer INITR-Anweisung zählt als ein Schreibvorgang. Achten Sie darauf, dass die zulässige Zahl der Schreibvorgänge nicht überschritten wird. Eine INITR-Anweisung sollte daher nicht in jedem Programmzyklus, sondern flankengesteuert ausgeführt werden.

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird ein Fehlercode eingetragen:

- In (S+) wurden andere Operanden angegeben als die Anfangsadressen der Sektoren. (Fehlercode „6706“)
- Die Adresse der zu initialisierenden Register überschreitet den Wert „32767“. (Fehlercode „6706“) In diesem Fall werden die Register bis R32767 bzw. bis ER32767 transferiert.
- Der Schreibschutz der Speicherkassette ist aktiviert. (Fehlercode „6770“)

Beispiel ▾

Mit dem folgenden Beispielprogramm werden die erweiterten Register R0 bis R2048 initialisiert (Sektor 0). Falls eine Speicherkassette installiert ist, werden die erweiterten File-Register ER0 bis ER2048 ebenfalls initialisiert.

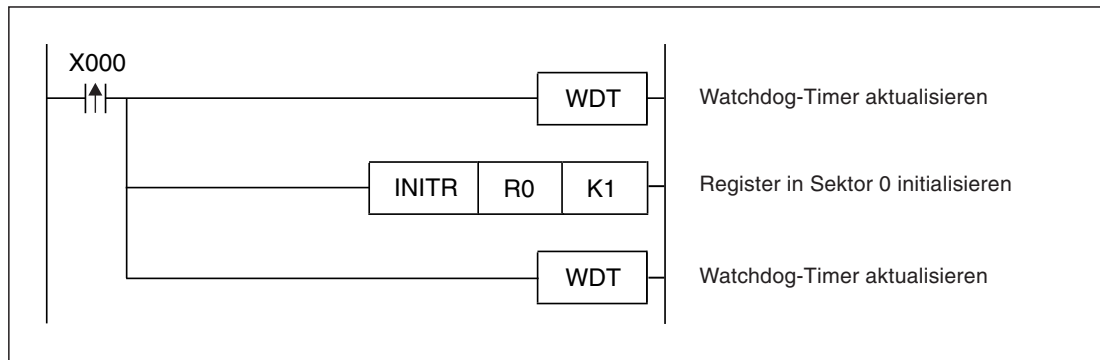


Abb. 7-312: Beim Einschalten des Eingangs X0 werden die erweiterten Register R0 bis R2048 und die erweiterten File-Register ER0 bis ER2048 initialisiert.

△

7.24.4 Operandenwerte in erweiterte Register/File-Register speichern (LOGR)

						LOGR		FNC 293				
						Operandenwerte in erweiterte Register oder erweiterte File-Register speichern						
						CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
												●
Operanden	S+	n1	D1+	n2	D2+	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	T*, C, D	D	R	K, H	D	●	16 Bit	32 Bit	LOGR LOGRP	11		

* Die Zähler C200 bis C255 können nicht angegeben werden.

Funktion

Speicherung der Inhalte von Wort-Operanden in erweiterte Register (R) und erweiterte File-Register (ER).

Beschreibung

- Mit einer LOGR-Anweisung kann eine Datenaufzeichnung realisiert werden. Dabei werden die Inhalte von bis zu 8000 Wort-Operanden aus einem zusammenhängenden Bereich gespeichert.
- Falls in der SPS keine Speicherkassette installiert ist, werden die Daten nur in die erweiterten Register im RAM der CPU übertragen.
- (S+) gibt die erste Adresse des aufzuzeichnenden Operandenbereichs an
- (n1) legt die Anzahl der Operanden fest, deren Inhalt aufgezeichnet werden soll. Es können zwischen 1 und 8000 Operanden erfasst werden.
- In (D1+) wird die erste Adresse des ersten Registerbereichs (Sektors) angegeben, in dem die Daten gespeichert werden sollen.

Angabe für (D1+)	Sektor	Speicherbereich	
		Erweiterte Register	Erweiterte File-Register
R0	0	R0 bis R2047	ER0 bis ER2047
R2048	1	R2048 bis R4095	ER2048 bis ER4095
R4096	2	R4096 bis R6143	ER4096 bis ER6143
R6144	3	R6144 bis R8191	ER6144 bis ER8191
R8192	4	R8192 bis R10239	ER8192 bis ER10239
R10240	5	R10240 bis R12287	ER10240 bis ER12287
R12288	6	R12288 bis R14335	ER12288 bis ER14335
R14336	7	R14336 bis R16383	ER14336 bis ER16383
R16384	8	R16384 bis R18431	ER16384 bis ER18431
R18432	9	R18432 bis R20479	ER18432 bis ER20479
R20480	10	R20480 bis R22527	ER20480 bis ER22527
R22528	11	R22528 bis R24575	ER22528 bis ER24575
R24576	12	R24576 bis R26623	ER24576 bis ER26623
R26624	13	R26624 bis R28671	ER26624 bis ER28671
R28672	14	R28672 bis R30719	ER28672 bis ER30719
R30720	15	R30720 bis R32767	ER30720 bis ER32767

Tab. 7-93: Durch die Angabe einer Anfangsadresse wird der erste Sektor festgelegt.

- (n2) gibt die Anzahl der Sektoren an, die mit den Daten beschrieben werden sollen. (n2) kann Werte zwischen 1 und 16 annehmen. Es werden solange Daten gespeichert, bis alle angegebenen Register gefüllt sind.
- (D2) enthält die Anzahl der bereits erfassten Daten.

Die folgende Abbildung zeigt die Struktur der Daten bei der Aufzeichnung.

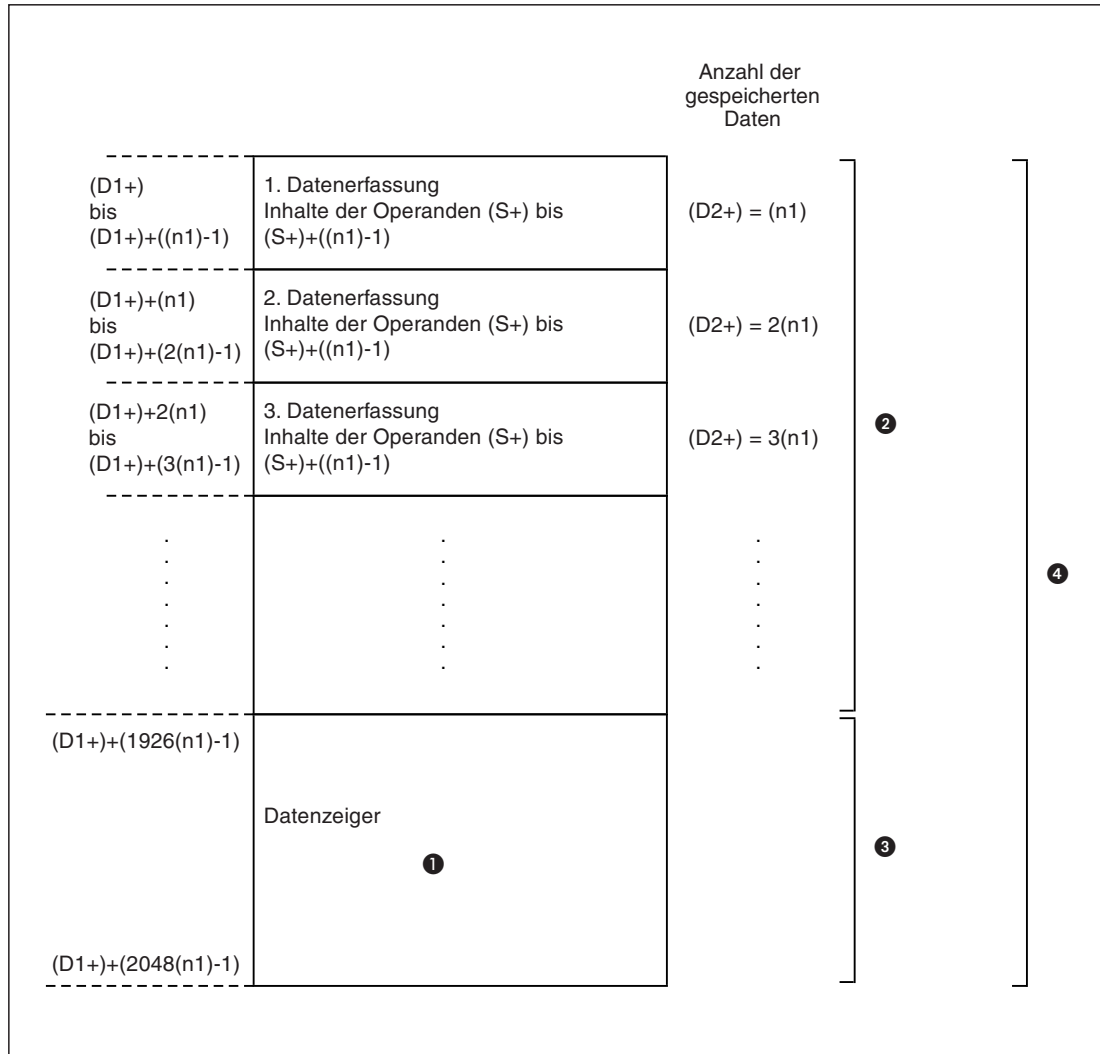


Abb. 7-313: Belegung der erweiterten Register und erweiterten File-Register durch eine LOGR-Anweisung.

- 1 Für jedes Wort, in dem erfasste Daten eingetragen wurden, wird in diesem Bereich ein Bit zurückgesetzt. (Bei der Initialisierung wurden alle Bits eines Registers gesetzt.) Begonnen wird bei Bit 0 des Operanden (D1+)+(1926n1)-1). Wenn alle 16 Bits dieses Operanden zurückgesetzt sind, wird dieser Vorgang beim Bit 0 des nächsten Operanden fortgesetzt.
- 2 In diesem Bereich werden die erfassten Daten eingetragen. Die Größe dieses Bereichs beträgt 1926 x (n2) Register.
- 3 Dieser Bereich dient zur Steuerung der Aufzeichnung und enthält den Datenzeiger. Dieser Bereich beansprucht 122 x (n2) Register.
- 4 Der gesamte Bereich für die Datenaufzeichnung ist 2048 x (n2) Register groß.

Hinweise zur Anwendung der LOGR-Anweisung

- Durch eine LOGR-Anweisung werden zyklisch Daten aufgezeichnet. Sollen Daten nur einmalig zu einem bestimmten Zeitpunkt erfasst werden, verwenden Sie bitte eine LOGRP-Anweisung.
- Vor der Ausführung einer LOGR-Anweisung müssen die erweiterten Register und erweiterten File-Register durch eine INITR-Anweisung initialisiert werden (Abschnitt 7.24.3). Ohne diese Initialisierung tritt bei der Ausführung der LOGR-Anweisung ein Fehler mit dem Code 6770 auf.

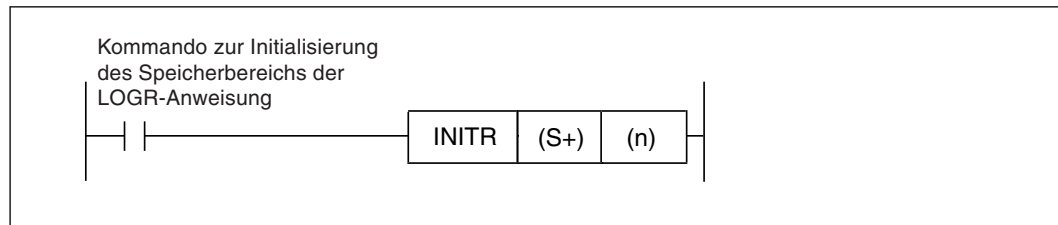


Abb. 7-314: Vor der Datenaufzeichnung müssen die Register initialisiert werden.

Für (S+) der INITR-Anweisung muss derselbe Operand angegeben werden wie für (D1+) der LOGR-Anweisung.

Der Parameter (n) der INITR-Anweisung und der Parameter (n2) der LOGR-Anweisung müssen identisch sein.

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird ein Fehlercode eingetragen:

- In (S+) wurden andere Operanden angegeben als die Anfangsadressen der Sektoren. (Fehlercode „6706“)
- Es konnten nicht alle erfassten Daten gespeichert werden, weil der zur Verfügung stehende Speicherplatz nicht ausreicht. (Fehlercode „6706“)
- Der Schreibschutz der Speicherkassette ist aktiviert. (Fehlercode „6770“)
- Die Daten konnten nicht gespeichert werden, weil z. B. die erweiterten File-Register nicht initialisiert worden sind. (Fehlercode „6770“)

Wenn dieser Fehler auftritt, gehen die Daten in den erweiterten Registern (R) verloren. Um dies zu vermeiden, sollten die Inhalte der erweiterten Registern mit Hilfe der Programmier-Software gesichert werden.

Beispiel ▾

Wenn der Eingang X1 eingeschaltet ist, werden mit dem folgenden Beispielprogramm die Inhalte der Datenregister D0 und D1 erfasst und in die erweiterten Register R2048 bis R6143 gespeichert.

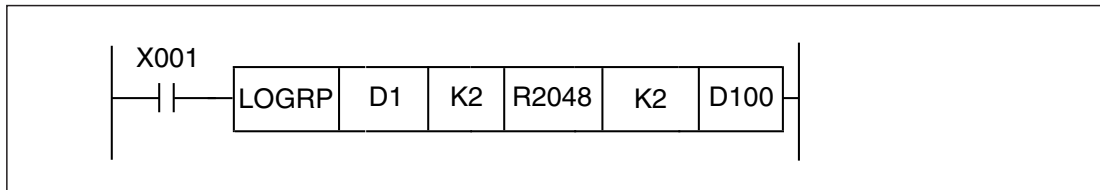


Abb. 7-316: Bei diesem Beispielprogramm enthält D100 die Anzahl der gespeicherten Daten.

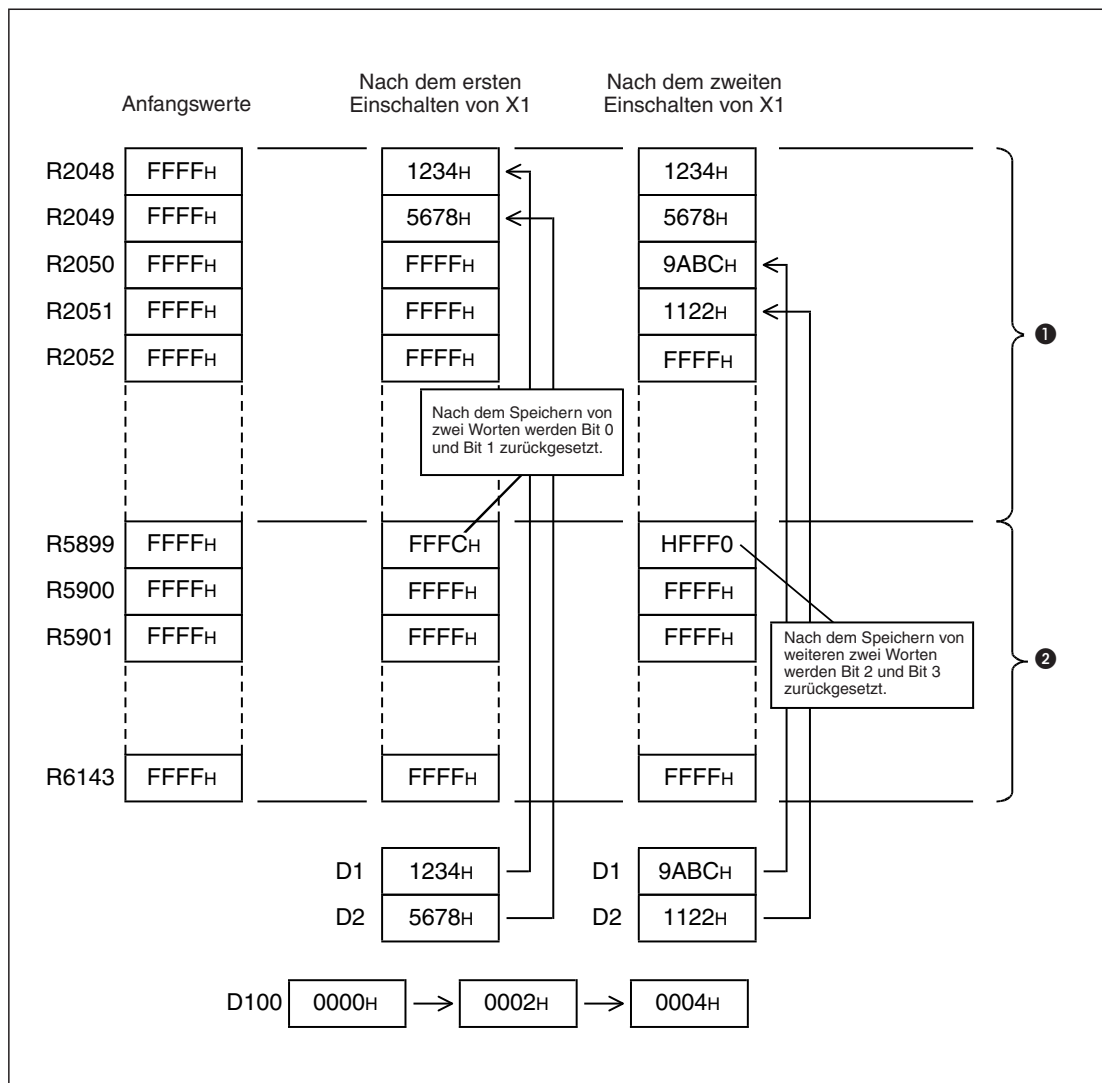


Abb. 7-315: Beispiel für die Inhalte der Register, wenn der Eingang X1 zweimal geschaltet wird.

- ① Dieser Bereich, der 3852 Register (1926 x 2) umfasst, steht zur Speicherung der Inhalte von D0 und D1 zur Verfügung.
- ② Dieser Bereich mit 244 Registern dient zur Steuerung der Aufzeichnung und enthält den Datenzeiger.



7.24.5 Daten aus erweiterte Register in erweiterte File-Register übertragen (RWER)

			RWER		FNC 294				
			Erweiterte Register in erweiterte File-Register übertragen						
			CPU	FX1s	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3s	FX3U FX3UC
							●		●
Operanden	S+	n	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	R	D, K, H	●		16 Bit	32 Bit	RWER RWERP		5

Funktion

Datentransfer aus erweiterte Register (R) im RAM der SPS in erweiterte File-Registern (ER) in einer Speicherkassette (Flash-EEPROM bei FX3U/FX3UC, EEPROM bei FX3G). Mit einer RWER-Anweisung können zwischen 1 und 32767 Register übertragen werden. Im Gegensatz zur SAVER-Anweisung (Abschnitt 7.24.2) besteht keine Bindung an Sektionen.

Beschreibung

- In (S+) wird die erste Adresse des Registerbereichs angegeben, dessen Inhalt übertragen werden soll.
- (n) gibt an, wie viele Registerinhalte übertragen werden sollen. Für (n) können Werte von 0 bis 32767 (1 bis 24000 bei FX3G) angegeben werden. Wird bei einer FX3U/FX3UC der Wert „0“ vorgegeben, werden die Inhalte von 32768 Adressen transferiert.

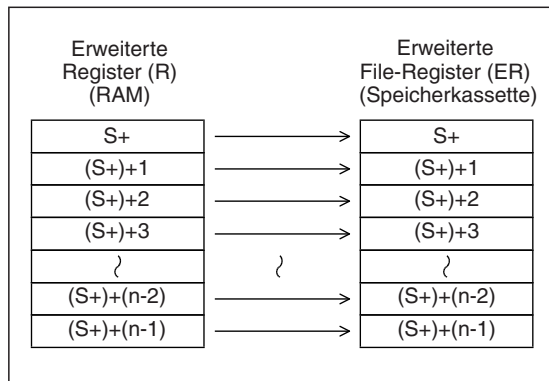


Abb. 7-317:

Eine RWER-Anweisung transferiert die Inhalte von (n) erweiterten Registern in erweiterte File-Register

Hinweise zur Anwendung einer RWER-Anweisung

- Es ist nicht notwendig, die Register vor der Ausführung einer RWER-Anweisung mit einer INITR- oder INITER-Anweisung zu initialisieren.
- Die Ausführung einer RWER-Anweisung darf nicht unterbrochen werden. Wird die Ausführung z. B. durch einen Ausfall der Versorgungsspannung abgebrochen, werden eventuell fehlerhafte Daten in die erweiterten File-Register eingetragen. Sichern Sie die Inhalte der erweiterten Register regelmäßig mit Hilfe der Programmier-Software GX Developer oder GX IEC Developer.
- Das Schreiben von Daten eines 1 Sektors in eine Flash-EEPROM-Speicherkassette dauert ca. 47 ms. Dadurch kann der Watchdog-Timer ansprechen. Die folgende Abbildung zeigt ein Beispielprogramm, bei dem die Einstellung des Watchdog-Timers vor der Ausführung einer RWER-Anweisung verändert und nach Abschluss der Datenübertragung wieder auf den ursprünglichen Wert eingestellt wird.

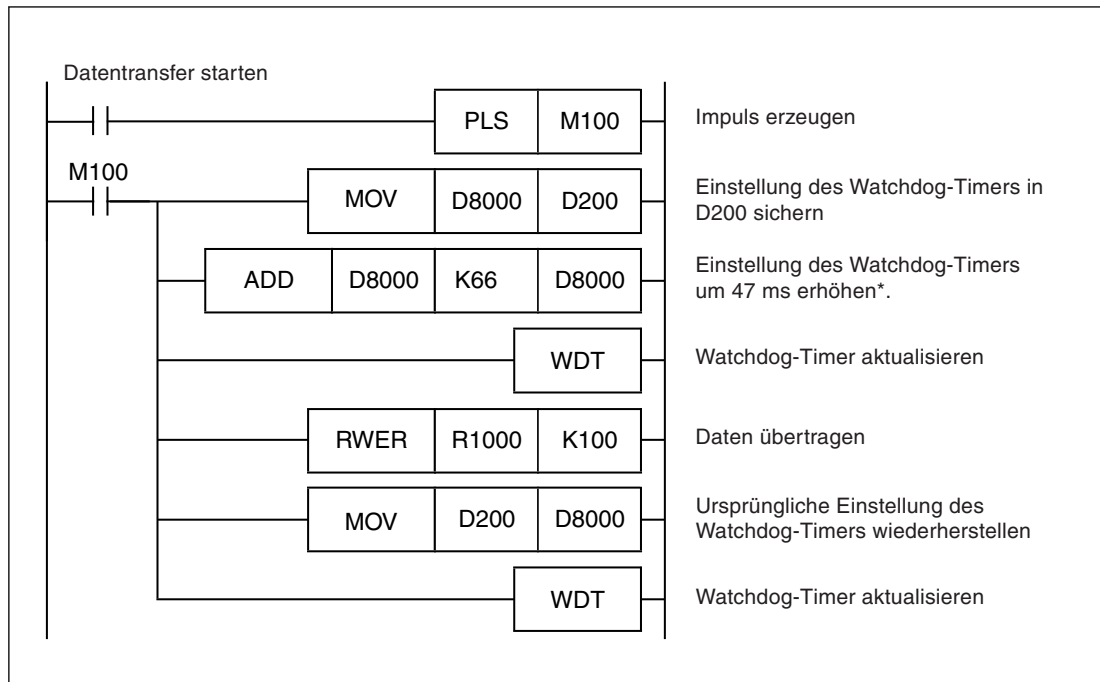


Abb. 7-318: Programmsequenz zur Anpassung des Watchdog-Timers

* Für jeden Sektor, der mit Daten beschrieben wird, sollte die Einstellung des Watchdog-Timers um 47 ms erhöht werden. Die Grenzen der Sektoren sind im Abschnitt 7.24.3 dargestellt.

- In eine Speicherkassette können ca. 10.000 mal und in das interne EEPROM einer FX3G können bis zu 20.000 mal Daten eingetragen werden. Jede Ausführung einer RWER-Anweisung zählt als ein Schreibvorgang. Achten Sie darauf, dass die zulässige Zahl der Schreibvorgänge nicht überschritten wird. Eine RWER-Anweisung sollte daher nicht in jedem Programmzyklus, sondern flankengesteuert ausgeführt werden.

HINWEIS

Wird eine RWER-Anweisung von einer FX3G-SPS ausgeführt, in der keine Speicherkassette installiert ist, werden die Daten in erweiterte File-Register (ER) im internen EEPROM des Grundgeräts übertragen. Auch in diesem Fall erfolgt die Übertragung ab der in (S+) angegebenen Adresse.

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird ein Fehlercode eingetragen:

- Die letzte Adresse der zu übertragenden Register überschreitet bei einer FX3U oder FX3UC den Wert „32767“. (Fehlercode „6706“) In diesem Fall werden Daten bis zur letzten Adresse (R32767) gelesen und transferiert.
- Die letzte Adresse der zu übertragenden Register überschreitet bei einer FX3G den Wert „23999“. (Fehlercode „6706“) In diesem Fall werden Daten bis zur letzten Adresse (R23999) gelesen und transferiert.
- Es ist keine Speicherkassette installiert. (Fehlercode „6771“) Dieser Fehler wird nur bei einer FX3U oder FX3UC gemeldet.
- Der Schreibschutz der Speicherkassette ist aktiviert. (Fehlercode „6770“)

Beispiel ▾

Wenn der Eingang X0 eingeschaltet ist, werden mit dem folgenden Beispielprogramm die Inhalte der erweiterten Register R10 bis R19 zur Datensicherung in die erweiterten File-Register zu übertragen.

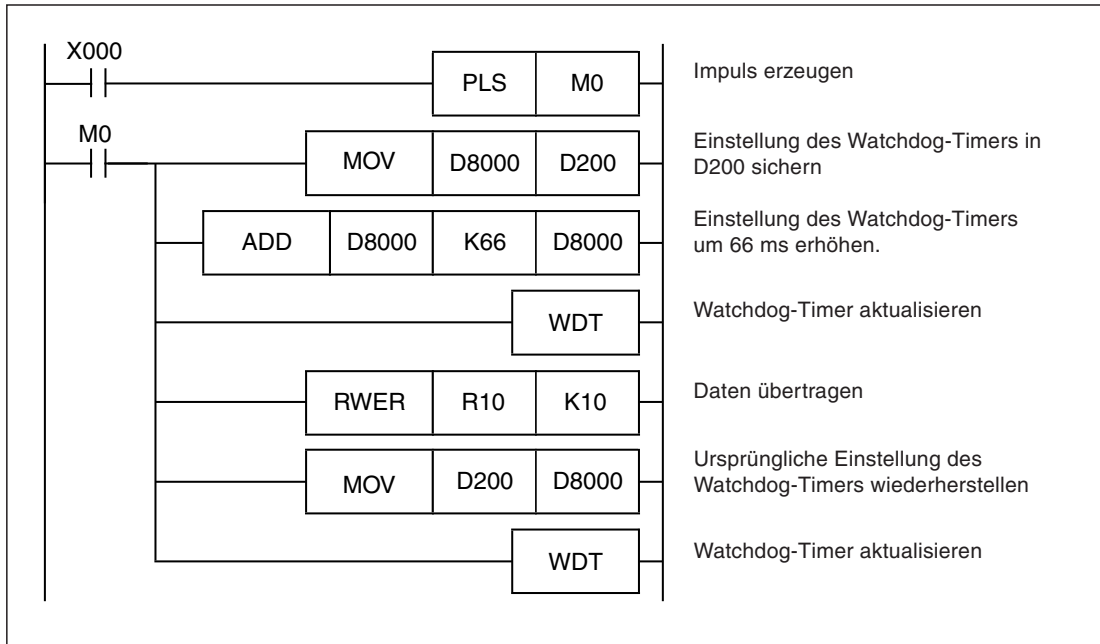


Abb. 7-319: Mit diesem Beispielprogramm werden die Inhalte der 10 erweiterten Register R10 bis R19 in eine Speicherkassette gesichert. Die Einstellung des Watchdog-Timers wird während des Datentransfers erhöht.

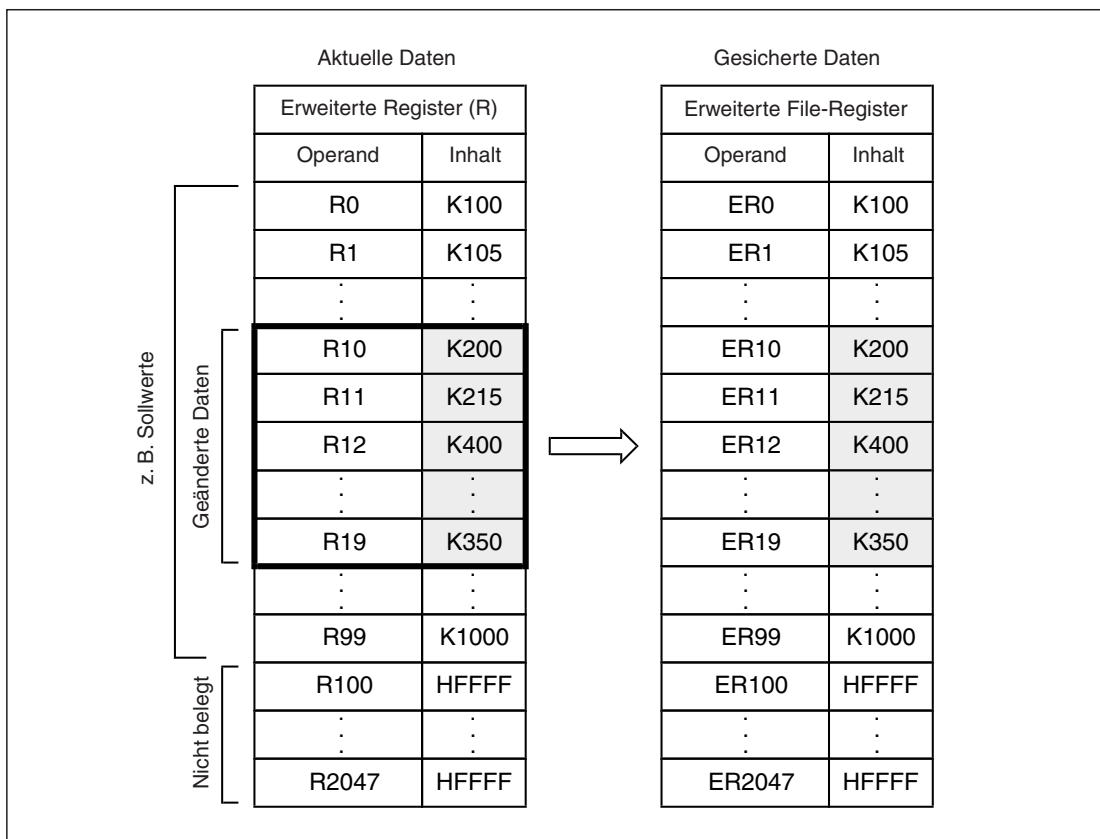


Abb. 7-320: Die Daten werden beim Einschalten von X001 transferiert.



7.24.6 Erweiterte File-Register initialisieren (INITER)

			INITER		FNC 295				
			Erweiterte File-Register initialisieren						
			CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
Operanden	S+	n	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	R	K, H	●		16 Bit	32 Bit	INITER INITERP	5	

Funktion

Initialisierung von erweiterten File-Registern (ER). Dabei wird in die Register der Wert „FFFFH“ eingetragen.

Beschreibung

- Die INITER-Anweisung trägt in erweiterte File-Register, die in einer Speicherkassette (Flash-EPROM) abgelegt sind, zur Initialisierung den Wert „FFFFH“ (entspricht dezimal „-1“) ein. Eine Initialisierung der erweiterten File-Register muss vorgenommen werden, bevor diese Register, z. B. mit einer SAVER-Anweisung, beschrieben werden.
- Die Initialisierung erfolgt sektorweise. Pro Sektor werden 2048 Register initialisiert. Mit einer INITER-Anweisung können mehrere Sektoren initialisiert werden.
- In (S+) wird die erste Adresse des ersten Registerbereichs (Sektors) angegeben, der initialisiert werden soll. Bitte beachten Sie, dass als Adresse „R“ und nicht „ER“ angegeben werden muss.

Angabe für (S+)	Sektor	Initialisierter Bereich
R0	0	ER0 bis ER2047
R2048	1	ER2048 bis ER4095
R4096	2	ER4096 bis ER6143
R6144	3	ER6144 bis ER8191
R8192	4	ER8192 bis ER10239
R10240	5	ER10240 bis ER12287
R12288	6	ER12288 bis ER14335
R14336	7	ER14336 bis ER16383
R16384	8	ER16384 bis ER18431
R18432	9	ER18432 bis ER20479
R20480	10	ER20480 bis ER22527
R22528	11	ER22528 bis ER24575
R24576	12	ER24576 bis ER26623
R26624	13	ER26624 bis ER28671
R28672	14	ER28672 bis ER30719
R30720	15	ER30720 bis ER32767

Tab. 7-94: Durch die Angabe einer Anfangsadresse wird der erste Sektor festgelegt.

- In (n) wird die Anzahl der zu initialisierenden Sektoren angegeben.

Operand	Inhalt vor Ausführung der INITER-Anweisung	Inhalt nach Ausführung der INITER-Anweisung
(S+)	1234H	FFFFH
(S+)+1	5678H	FFFFH
(S+)+2	90ABH	FFFFH
:	:	:
(S+)+(2048 x n)-1	CDEFH	FFFFH

Tab. 7-95: Durch eine INITER-Anweisung werden erweiterte File-Register mit dem Wert FFFFH beschrieben.

Hinweise zur Anwendung der INITER-Anweisung

- Mit einer INITER-Anweisung werden nur erweiterte File-Register in einer Speicherkassette initialisiert. Verwenden Sie eine INITER-Anweisung (Abschnitt 7.24.3), wenn gleichzeitig erweiterte Register (R) und erweiterte File-Registern (ER) initialisiert werden sollen.
- Die Initialisierung eines Sektors (2048 Register) einer Speicherkassette dauert ca. 25 ms. Dadurch kann ein Watchdog-Timer-Fehler auftreten. In diesem Fall können die Einstellung des Watchdog-Timers oder die Programmsequenz mit der INITER-Anweisung geändert werden.
 - Änderung der Einstellung des Watchdog-Timers
 Durch den Eintrag eines neuen Wertes in das Sonderregister D8000 wird die Überwachungszeit geändert. Diese Zeit ist auf 200 ms voreingestellt. Um die neue Zeit sofort zu aktivieren, sollte zusätzlich eine WDT-Anweisung ausgeführt werden. Ohne diese Anweisung wird die neue Einstellung des Watchdog-Timers erst im nächsten Programmzyklus gültig.

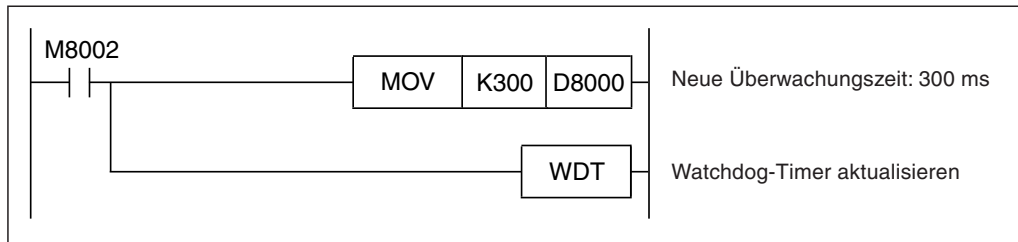


Abb. 7-321: Beispiel zur Änderung der Einstellung des Watchdog-Timers: Im ersten Programmzyklus nach dem Anlauf der SPS wird M8002 für einen Zyklus gesetzt und die Einstellung des Watchdog-Timers verändert.

- Aktualisierung des Watchdog-Timers vor und nach Ausführung der INITER-Anweisung

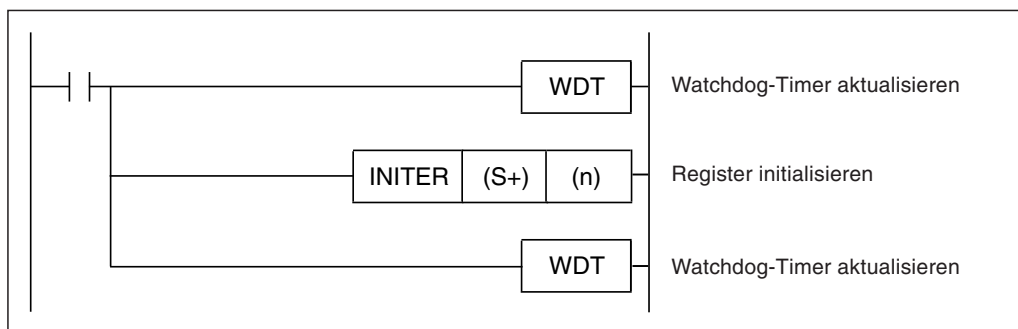


Abb. 7-322: Durch die Aktualisierung des Watchdog-Timers wird verhindert, dass bei der Initialisierung ein Fehler auftritt.

- In eine Speicherkassette können ca. 10.000 mal Daten eingetragen werden. Jede Ausführung einer INITER-Anweisung zählt als ein Schreibvorgang. Achten Sie darauf, dass die zulässige Zahl der Schreibvorgänge nicht überschritten wird. Eine INITER-Anweisung sollte daher nicht in jedem Programmzyklus, sondern flankengesteuert ausgeführt werden.

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, der Sondermerker M8067 wird gesetzt, und in das Sonderregister D8067 wird ein Fehlercode eingetragen:

- In (S+) wurden andere Operanden angegeben als die Anfangsadressen der Sektoren. (Fehlercode „6706“)
- Die Adresse der zu initialisierenden Register überschreitet den Wert „32767“. (Fehlercode „6706“) In diesem Fall werden die Register bis ER32767 initialisiert.
- Der Schreibschutz der Speicherkassette ist aktiviert. (Fehlercode „6770“)
- Es ist keine Speicherkassette installiert. (Fehlercode „6771“)

Beispiel ▾

Mit dem folgenden Beispielprogramm werden die erweiterten File-Register ER0 bis ER2048 initialisiert (Sektor 0).

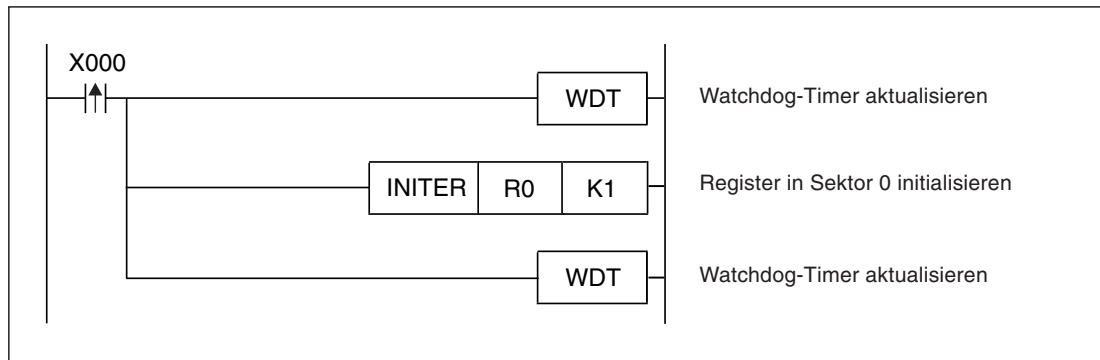


Abb. 7-323: Beim Einschalten des Eingangs X0 werden die erweiterten File-Register ER0 bis ER2048 initialisiert.



7.25 Anweisungen für Adaptermodul FX3U-CF-ADP

Die in diesem Abschnitt beschriebenen Anweisungen können angewendet werden, wenn an einem FX3U- oder FX3UC-Grundgerät ein CF-Speicherkartenadaptermodul FX3U-CF-ADP angeschlossen ist.

Symbol	FNC	Bedeutung	Abschnitt
FLCRT	300	Datei erzeugen/prüfen	7.25.1
FLDEL	301	Datei löschen/CF-Speicherkarte formatieren	7.25.2
FLWR	302	Daten in CF-Speicherkarte schreiben	7.25.3
FLRD	303	Daten aus CF-Speicherkarte lesen	7.25.4
FLCMD	304	Anweisung für FX3U-CF-ADP	7.25.5
FLSTRD	305	Status des FX3U-CF-ADP lesen	7.25.6

Tab. 7-96: Übersicht der Anweisungen für das Adaptermodul FX3U-CF-ADP

HINWEIS

Eine der Anweisungen FLCRT (FNC300) bis FLSTRD (FNC305) darf nicht gleichzeitig mit einer der folgenden Anweisungen für dieselbe Schnittstelle ausgeführt werden:

- RS (FNC80)
- RS2 (FNC87)
- IVCK (FNC270) bis IVMC (FNC275)
- ADPRW (FNC276)

7.25.1 Datei erzeugen/prüfen (FLCRT)

					FLCRT		FNC 300					
					Datei in CF-Speicherkarte erzeugen/prüfen							
					CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	
												●
Operanden	S1+	S2+	S3+	n	Puls-Anweisung (P)		Verarbeitung		Programmschritte			
	D*, R, K, H	D*, R, Zeichenfolge	D*, R	K, H			16 Bit	32 Bit	FLCRT		9	

* Sonderregister können nicht angegeben werden.

Funktion

Eine FLCRT-Anweisung erzeugt eine Datei in einer CF-Speicherkarte, die in einem Adaptermodul FX3U-CF-ADP installiert ist. Wenn eine FLCRT-Anweisung nach der Erzeugung einer neuen Datei ausgeführt wird, prüft sie den Bezug des Datei-Namens zur Datei-Nummer.

Beschreibung

- In (S1+) wird die Dateinummer angegeben („0“ bis „63“).
 - Angabe der Datei-Nummer „0“
Wird in (S1+) als Datei-Nummer „0“ angegeben, erzeugt die FLCRT-Anweisung eine FIFO-Datei. Falls durch die SPS mehr als eine FIFO-Datei erzeugt wurde, bezieht sich ein FIFO-Speichervorgang (First in, first out) auf die einzelnen Dateien. Dabei bleibt die neueste Datei erhalten und ältere Dateien werden gelöscht. So wird sichergestellt, dass der durch die FIFO-Dateien und den anderen Dateien belegte Speicherplatz die festgelegte Kapazität nicht überschreitet.
 - Angabe einer Datei-Nummer aus dem Bereich 1 bis 63
Wird in (S1+) eine Datei-Nummer von „1“ bis „63“ angegeben, erzeugt die FLCRT-Anweisung eine Datei mit dieser Nummer und dem in (S2+) angegebenen Datei-Namen. Im Ablaufprogramm wird zur Angabe dieser Datei die Datei-Nummer verwendet. Jede Datei in der CF-Speicherkarte erhält eine Datei-Nummer und kann anhand einer Tabelle mit den Datei-Nummern zugeordnet werden.

Falls bereits eine Datei mit dem angegebenen Namen existiert und in der Tabelle der Dateinummern eingetragen ist, wird die Ausführung der FLCRT-Anweisung abgebrochen und keine neue Datei erzeugt.

Existiert bereits eine Datei mit dem angegebenen Namen, die aber nicht in der Tabelle der Dateinummern eingetragen ist, trägt die FLCRT-Anweisung diese Datei in die Tabelle ein.
- Der Name der Datei wird in (S2+) als Zeichenfolge angegeben (siehe folgende Tabelle).
- (S3+) enthält die Parameter zur Erzeugung einer Datei (siehe folgende Tabelle).
- Mit n wird die Nummer des Kommunikationskanal angegeben, den das Adaptermodul FX3U-CF-ADP belegt (siehe folgende Tabelle).

Operand		Beschreibung	Datentyp
(S1+)		<p>Datei-Nummer</p> <p>Diese Datei-Nummer wird mit dem Datei-Namen in Zusammenhang gebracht.</p> <p>Die FLCRT-Anweisung erzeugt eine Datei und ordnet die Datei-Nummer gleichzeitig dem Datei-Namen zu. Danach wird zur Angabe der Datei im Programm die Datei-Nummer verwendet.</p> <p>Zulässiger Wertebereich: 0 bis 63 (Bei Angabe von „0“ wird eine FIFO-Datei erzeugt.)</p>	BIN-16-Bit
(S2+)		<p>Datei-Name</p> <ul style="list-style-type: none"> • Wenn in (S1+) „0“ angegeben ist (FIFO-Datei) Der Datei-Name wird nicht verwendet (Die Angabe in (S2+) wird ignoriert). Verwenden Sie einen nicht verwendeten Operanden vom Typ D oder R. • Wenn in (S1+) ein Wert von „1“ bis „63“ angegeben ist Der Datei-Name wird mit bis zu 8 Zeichen angegeben, auf denen „Null“ oder „Null + Null“ folgt. Es können Zeichen mit halber Breite sowie Symbole mit halber Breite verwendet werden, die auch unter MS-DOS zulässig sind. <p>Symbole mit halber Breite: !, #, \$, %, &, ' (,), +, -, @, ^, _, ;, ~</p> <p>Die Dateierweiterung ist fest auf „CSV“ eingestellt.</p>	Zeichenfolge
Parameter zur Erzeugung einer Datei	(S3+)	<p>Zeitstempel</p> <p>Hier wird eingestellt, ob die Datei mit einem Zeitstempel versehen werden soll.</p> <ul style="list-style-type: none"> • 0: Kein Zeitstempel (NULL) • 1: yyyy/mm/dd hh:mm:ss • 2: yy/mm/dd hh:mm:ss • 3: dd/mm/yyyy hh:mm:ss • 4: dd/mm/yy hh:mm:ss • 5: mm/dd/yyyy hh:mm:ss • 6: mm/dd/yy hh:mm:ss • 7: hh:mm:ss 	BIN-16-Bit
	(S3+)+1	<p>Datentyp</p> <p>Hier wird der Typ der zu speichernden Daten festgelegt.</p> <ul style="list-style-type: none"> • 0: Keine Angabe eines Datentyps (gemischte Datentypen) • 1: Bit • 2: Dezimal (16-bit) • 3: Dezimal (32-bit) • 4: Hexadezimal (16-bit) • 5: Hexadezimal (32-bit) • 6: REAL (Gleitkommazahlen) mit Exponent • 7: Zeichenfolge 	BIN-16-Bit
	(S3+)+2	<p>Maximale Anzahl der Zeilen</p> <p>Zulässiger Wertebereich: 1 bis 32767</p> <p>Verwenden Sie diese Einstellung, um die Größe der Datei an die Dateigröße anzupassen, die mit der verwendeten Applikations-Software verarbeitet werden kann.</p>	BIN-16-Bit
	(S3+)+3	<ul style="list-style-type: none"> • Wenn in (S1+) „0“ angegeben ist (FIFO-Datei) Es wird eingestellt (in %), welcher Anteil der Gesamtspeicherkapazität der CF-Speicherkarte verwendet werden soll. Zulässiger Wertebereich: 10 bis 90 (%) • Wenn in (S1+) ein Wert von „1“ bis „63“ angegeben ist Es wird eingestellt, wie vorgegangen werden soll, wenn die eingestellte maximale Anzahl der Zeilen erreicht wird. <p>0: Die Ausführung wird angehalten. (In die Datei werden keine weiteren Daten gespeichert.)</p> <p>1: Weitere Daten werden ab dem Anfang der Datei gespeichert (Ringpuffer).</p>	BIN-16-Bit
n		<p>Kanalnummer des FX3U-CF-ADP</p> <ul style="list-style-type: none"> • 1: Ch. 1 • 2: Ch. 2 	BIN-16-Bit

Tab. 7-97: Beschreibung der Operanden einer FLCRT-Anweisung

Hinweise zur Anwendung der FLCRT-Anweisung

- Wenn in (S1+) als Datei-Nummer „0“ angegeben ist.
 - Das Adaptermodul FX3U-CF-ADP kann bis zu 1000 Dateien erzeugen. (Maßgebend für die Anzahl der Dateien ist die Kapazität der CF-Speicherkarte).
 - Die Dateien erhalten die Bezeichnungen „FILE0000.CSV“ bis „FILE0999.CSV“.
- Wenn in (S1+) eine Datei-Nummer von „1“ bis „63“ angegeben ist.
 - Der Anwender kann bis zu 63 Dateien erzeugen. (Maßgebend für die Anzahl der Dateien ist die Kapazität der CF-Speicherkarte).
 - Werden für dieselbe Datei-Nummer verschiedene Datei-Namen oder für den selben Datei-Namen verschiedene Datei-Nummern angegeben, wird die Ausführung der FLCRT-Anweisung abgebrochen.

7.25.2 Datei löschen/CF-Speicherkarte formatieren (FLDEL)

				FLDEL		FNC 301				
				Datei(en) in CF-Speicherkarte löschen CF-Speicherkarte formatieren						
	CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC	●		
Operanden	S1+	S2+	n		Puls-Anweisung (P)		Verarbeitung		Programmschritte	
	D*, R, K, H		K, H				16 Bit	32 Bit	FLDEL	7

* Sonderregister können nicht angegeben werden.

Funktion

Eine FLDEL-Anweisung löscht eine oder mehrere in einer CF-Speicherkarte gespeicherten Dateien oder formatiert die CF-Speicherkarte.

Beschreibung

- In (S1+) wird die Nummer der zu löschenden Datei bzw. die Funktion angegeben, die von der FLDEL-Anweisung ausgeführt werden soll (siehe folgende Tabelle).
- (S2+) enthält die Parameter zum Löschen der Datei bzw. zum Formatieren der CF-Speicherkarte (siehe folgende Tabelle).
- Mit n wird die Nummer des Kommunikationskanal angegeben, den das Adaptermodul FX3U-CF-ADP belegt (siehe folgende Tabelle).

Operand	Beschreibung	Datentyp
(S1+)	Datei-Nummer/Funktionsauswahl <ul style="list-style-type: none"> • -1 (FFFFH): Die FLDEL-Anweisung löscht alle Dateien mit den Datei-Nummern 0 bis 63. • 0 bis 63: Die FLDEL-Anweisung löscht die Datei mit der entsprechenden Datei-Nummer. • 512 (200H): Die FLDEL-Anweisung formatiert die CF-Speicherkarte. 	BIN-16-Bit
(S2+)	Parameter zum Löschen der Datei/Format der CF-Speicherkarte <ul style="list-style-type: none"> • Wenn in (S1+) „-1“ oder ein Wert zwischen 0 und 63 angegeben ist: Parameter zum Löschen der Datei(en) <ul style="list-style-type: none"> 0: Die FLDEL-Anweisung löscht die angegebene Datei bzw. die angegebenen Dateien. 1: Die FLDEL-Anweisung löscht den Zusammenhang zwischen Datei-Nummer und Datei-Namen (Die Datei wird nicht gelöscht.) Falls aber (S1+) den Wert „0“ enthält, löscht die FLDEL-Anweisung unabhängig von der Einstellung in (S1+) die Datei. • Wenn in (S1+) der Wert von „512 (200H)“ angegeben ist: <ul style="list-style-type: none"> Format der CF-Speicherkarte 256 (100H): Die FLDEL-Anweisung formatiert die CF-Speicherkarte im FAT16-Format. 	BIN-16-Bit
n	Kanalnummer des FX3U-CF-ADP <ul style="list-style-type: none"> • 1: Ch. 1 • 2: Ch. 2 	BIN-16-Bit

Tab. 7-98: Operanden der FLDEL-Anweisung

HINWEIS

Wenn in (S1+) einer der Werte „0“ (FIFO-Datei) oder „-1“ (alle Dateien) angegeben ist, kann der Löschvorgang, abhängig von der Anzahl der Dateien, bis zu einer Minute dauern.

7.25.3 Daten in CF-Speicherkarte schreiben (FLWR)

						FLWR	FNC 302					
						Datei in CF-Speicherkarte schreiben						
						CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
Operanden	S1+	S2+	S3+	D+	n	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	D*, R, K, H	X, Y, M, S, T, C, D, R	D*, R	K, H			16 Bit	32 Bit	FLWR	11		

* Sonderregister können nicht angegeben werden.

Funktion

Eine FLWR-Anweisung schreibt Daten in eine Datei in einer CF-Speicherkarte, die in einem Adaptermodul FX3U-CF-ADP installiert ist.

Beschreibung

- (S1+) enthält die Nummer der Datei.
- In (S2+) wird der Anweisung die erste Adresse des Bereichs übergeben, in dem in der SPS die zu schreibenden Daten gespeichert sind.
- (S3+) enthält die Parameter zum Schreiben in die Datei (siehe folgende Tabelle).
- Der Inhalt von (D+) gibt die Position des Datenzeigers nach dem Schreiben der Daten an.
- Mit n wird die Nummer des Kommunikationskanal angegeben, den das Adaptermodul FX3U-CF-ADP belegt (siehe folgende Tabelle).
- Eine FLWR-Anweisung schreibt die Daten, die durch (S2+) angegeben sind, in eine durch ihre Dateinummer angegebene Datei in der CF-Speicherkarte oder in den Puffer des FX3U-CF-ADP. Die FLWR-Anweisung kann Daten ab der in (S3+)+1 angegebenen Position überschreiben oder die Daten anfügen. Falls in den Puffer des FX3U-CF-ADP geschrieben wird, können Daten nur angefügt werden.

Nach dem Schreiben der Daten wird der Datenzeiger auf die folgenden Werte gestellt:

- Wenn die Daten an bestehende Daten angefügt werden:
 - Zeilenposition nach dem Schreiben: Zuletzt beschriebene Zeile + 1
 - Spaltenposition nach dem Schreiben: 1
- Wenn bestehende Daten überschrieben werden:
 - Zeilenposition nach dem Schreiben:
 - Zuletzt beschriebene Zeile, wenn Daten nicht in die letzte Spalte der Zeile eingetragen wurden; Zuletzt beschriebene Zeile + 1, wenn die Daten in die letzte Spalte der Zeile eingetragen wurden.
 - Spaltenposition nach dem Schreiben:
 - Erste Spalte nach der zuletzt beschriebenen Spalte, wenn Daten nicht in die letzte Spalte der Zeile eingetragen wurden; „1“, wenn die Daten in die letzte Spalte einer Zeile eingetragen wurden.

Beim Anfügen und beim Überschreiben von Daten wird die Anzahl Zeilen beschrieben, die bei der Erzeugung der Datei als maximale Anzahl der Zeilen angegeben wurde. Falls Daten in die letzte Spalte eingetragen wurden, hängt nach dem Schreiben die Zeilenposition vom Typ der Datei und den Einstellungen ab.

- Wenn bei einer normalen Datei das Schreiben beim Erreichen der maximalen Zeilenposition gestoppt wird:
 Zeilenposition nach dem Schreiben = Max. Anzahl der Zeilen + 1 („-32768“ wenn als max. Anzahl der Zeilen „32767“ angegeben wurde.
- Wenn bei einer normalen Datei das Schreiben beim Erreichen der maximalen Zeilenposition in der ersten Zeile fortgesetzt wird:
 Zeilenposition nach dem Schreiben = 1

Die Spaltenposition nach dem Schreiben ist in beiden Fällen „1“.

Operand		Beschreibung	Datentyp
(S1+)		Datei-Nummer Zulässiger Wertebereich: 0 bis 63	BIN-16-Bit
(S2+)		Erste Adresse des Operandenbereichs, in dem die Daten gespeichert sind, die in die Datei in der CF-Speicherkarte geschrieben werden sollen.	—
Parameter zum Schreiben der Daten	(S3+)	Datentyp Hier wird der Typ der zu schreibenden Daten festgelegt. <ul style="list-style-type: none"> • 0: Keine Angabe eines Datentyps (gemischte Datentypen) • 1: Bit • 2: Dezimal (16-bit) • 3: Dezimal (32-bit) • 4: Hexadezimal (16-bit) • 5: Hexadezimal (32-bit) • 6: REAL (Gleitkommazahlen) mit Exponent (32-bit) • 7: Zeichenfolge (maximal 512 Zeichen halber oder ganzer Breite) • 8: Datei-Name (Zeichenfolge aus max. 32 Zeichen halber oder ganzer Breite, die Dateierweiterung sowie Datum und Uhrzeit werden automatisch hinzugefügt.) 	BIN-16-Bit
	(S3+)+1	Angabe, in welche Zeile geschrieben werden soll oder ob die Daten an bestehende Daten angefügt werden sollen Zulässige Wertebereiche <ul style="list-style-type: none"> • Zeilenposition: „1“ bis zur eingestellten maximalen Anzahl der Zeilen in der Datei • Zum Anfügen der Daten: -1 	BIN-16-Bit
	(S3+)+2	Angabe, in welche Spalte geschrieben werden soll Zulässige Wertebereiche <ul style="list-style-type: none"> • Spaltenposition: 1 bis 254 • Zum Anfügen der Daten: -1 	BIN-16-Bit
	(S3+)+3	Anzahl der zu schreibenden Daten Zulässiger Wertebereich: 1 bis 254	BIN-16-Bit
Position des Datenzeigers nach dem Schreiben	(D+)	Zeilenposition: „1“ bis zur eingestellten maximalen Anzahl der Zeilen in der Datei	BIN-16-Bit
	(D+)+1	Spaltenposition: 1 bis 254	BIN-16-Bit
n		Kanalnummer des FX3U-CF-ADP <ul style="list-style-type: none"> • 1: Ch. 1 • 2: Ch. 2 	BIN-16-Bit

Tab. 7-99: Operanden der FLWR-Anweisung

Hinweise zur Anwendung der FLWR-Anweisung

- Die Ausführung einer FLWR-Anweisung wird abgebrochen, wenn keine CF-Speicherkarte installiert ist.
- Bitte beachten Sie die maximale Anzahl der Schreibvorgänge, die für eine CF-Speicherkarte ausgeführt werden dürfen und dass bei jeder Ausführung einer FLWR-Anweisung ein Schreibvorgang stattfindet. Werden zum Beispiel einmal pro Minute Daten in eine CF-Speicherkarte übertragen, werden 100.000 Schreibvorgänge in ca. 2 Monaten ausgeführt.
- Auch wenn Daten in den internen Speicher des Adaptermoduls FX3U-CF-ADP geschrieben werden, werden bei einem Datenüberlauf Daten in die CF-Speicherkarte eingetragen.

Wenn als Ziel der Daten der interne Speicher des FX3U-CF-ADP angegeben wurde, trägt die FLWR-Anweisung die Daten in die CF-Speicherkarte ein, wenn der interne Speicher voll ist. Die Daten im internen Speicher des Adaptermoduls FX3U-CF-ADP werden bei einem Spannungsausfall (auch bei einem kurzzeitigen Ausfall) gelöscht.

- Falls in (S3+) als Datentyp „8“ (Dateiname) angegeben ist, kann der Anwender dadurch nur die Überschrift angeben, bevor weitere Daten geschrieben werden. Die Dateierweiterung sowie Datum und Uhrzeit werden automatisch hinzugefügt.
- Zur Ausführung einer FLWR-Anweisung werden eventuell mehrere Programmzyklen benötigt. Falls Datenkonsistenz gefordert ist, sollten geeignete Maßnahmen ergriffen werden (wie beispielsweise die Sicherung der Daten in anderen Operanden).
- Falls als Datentyp nicht „Bit“ eingestellt ist und in (S2+) ein Bit-Operand angegeben wird, muss dessen Adresse entweder „0“ oder ein Vielfaches von 16 sein.

Wird in (S2+) ein Wort-Operand angegeben und ist als Datentyp „Bit“ eingestellt, entnimmt die FLWR-Anweisung die zu schreibenden Daten ab dem niederwertigsten Bit des entsprechenden Operanden.

- Falls in (S3+) als Datentyp „7“ (Zeichenfolge) oder „8“ (Dateiname) angegeben ist, kennzeichnet der Code 00H das Ende der Zeichenfolge und muss an die Zeichenfolge angefügt werden.

7.25.4 Daten aus CF-Speicherkarte lesen (FLRD)

						FLRD		FNC 303				
						Datei aus CF-Speicherkarte lesen						
						CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
Operanden	S1+	S2+	D1+	D2+	n	Puls-Anweisung (P)		Verarbeitung		Programmschritte		
	D*, R, K, H	D*, R	Y, M, S, D, R	D*, R	K, H			16 Bit	32 Bit	FLRD	11	

* Sonderregister können nicht angegeben werden.

Funktion

Eine FLRD-Anweisung liest Daten aus einer Datei in einer CF-Speicherkarte, die in einem Adaptermodul FX3U-CF-ADP installiert ist.

Beschreibung

- In (S1+) wird die Nummer der Datei angegeben, aus der gelesen werden soll.
- (S2+) enthält die Parameter zum Lesen aus der Datei (siehe folgende Tabelle).
- In (D1+) wird der Anweisung die erste Adresse des Bereichs übergeben, in dem in der SPS die aus der Datei gelesenen Daten gespeichert werden sollen.
- (D2+) enthält die Anzahl der Daten in der angegebenen Zeile.
- Mit n wird die Nummer des Kommunikationskanal angegeben, den das Adaptermodul FX3U-CF-ADP belegt (siehe folgende Tabelle).
- Eine FLRD-Anweisung liest Daten, die durch ihre Zeilen- und Spaltenposition in der Datei und ihre Anzahl ((S2+), (S2+)+1, usw.) spezifiziert sind, aus einer durch ihre Dateinummer in (S1+) bestimmten Datei und speichert die Daten in dem durch (D1+) angegebenen Operandenbereich.

Operand		Beschreibung	Datentyp
(S1+)		Datei-Nummer Zulässiger Wertebereich: 0 bis 63	BIN-16-Bit
Parameter zum Schreiben der Daten	(S2+)	Datentyp Hier wird der Typ der zu schreibenden Daten festgelegt. <ul style="list-style-type: none"> • 0: Keine Angabe eines Datentyps (gemischte Datentypen) • 1: Bit • 2: Dezimal (16-bit) • 3: Dezimal (32-bit) • 4: Hexadezimal (16-bit) • 5: Hexadezimal (32-bit) • 6: REAL (Gleitkommazahlen) mit Exponent (32-bit) • 7: Zeichenfolge (maximal 512 Zeichen halber oder ganzer Breite) 	BIN-16-Bit
	(S2+)+1	Angabe, ab welcher Zeile Daten gelesen werden sollen Zulässiger Wertebereich: „1“ bis zur eingestellten maximalen Anzahl der Zeilen in der Datei	BIN-16-Bit
	(S2+)+2	Angabe, ab welcher Spalte Daten gelesen werden sollen Zulässiger Wertebereich: 1 bis 254	BIN-16-Bit
	(S2+)+3	Anzahl der zu lesenden Daten Zulässiger Wertebereich: 1 bis 254	BIN-16-Bit
(D1+)		Erste Adresse des Operandenbereichs, in dem die aus der CF-Speicherkarte gelesenen Daten gespeichert werden sollen.	BIN-16-Bit
(D2+)		Anzahl der in der angegebenen Zeile enthaltenen Daten Wertebereich: 1 bis 254 0: Keine Daten	BIN-16-Bit
n		Kanalnummer des FX3U-CF-ADP <ul style="list-style-type: none"> • 1: Ch. 1 • 2: Ch. 2 	BIN-16-Bit

Tab. 7-100: Operanden der FLRD-Anweisung

Hinweise zur Anwendung der FLRD-Anweisung

- Die Ausführung einer FLRD-Anweisung wird abgebrochen, wenn keine CF-Speicherkarte installiert ist.
- Zur Ausführung einer FLRD-Anweisung werden eventuell mehrere Programmzyklen benötigt. Falls Datenkonsistenz gefordert ist, sollten die gelesenen Daten erst verwendet werden, nachdem geprüft wurde, dass die Ausführung der FLRD-Anweisung abgeschlossen ist.
- Falls in (S2+) als Datentyp nicht „Bit“ eingestellt ist und in (D1+) ein Bit-Operand angegeben wird, muss dessen Adresse entweder „0“ oder ein Vielfaches von 16 sein.

Wird in (D1+) ein Wort-Operand angegeben und ist als Datentyp „Bit“ eingestellt, speichert die FLRD-Anweisung die gelesenen Daten ab dem niederwertigsten Bit des entsprechenden Wort-Operanden.

- Wird als Datentyp ein anderer Typ als „Zeichenfolge“ eingestellt und reicht die Anzahl der Operanden nicht aus, um die gelesenen Daten zu speichern, liest die FLRD-Anweisung keine Daten aus der CF-Speicherkarte, und es tritt ein Fehler auf.

Wird als Datentyp „Zeichenfolge“ eingestellt, ist die Länge der Zeichenfolge nicht bekannt. Die SPS speichert so viele Daten wie möglich. Falls der Lesevorgang nach dem Erreichen des letzten Operanden noch nicht abgeschlossen ist, tritt ein Fehler auf.

7.25.5 Anweisung für FX3U-CF-ADP (FLCMD)

		FLCMD		FNC 304				
		Anweisung für FX3U-CF-ADP						
		CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC
								●
Operanden	S+	n	Puls-Anweisung (P)		Verarbeitung		Programmschritte	
	D*, R, K, H	K, H			16 Bit	32 Bit	FLCMD	5
					●			

* Sonderregister können nicht angegeben werden.

Funktion

Mit einer FLCMD-Anweisung kann der Betrieb des Adaptermoduls FX3U-CF-ADP beeinflusst werden.

Beschreibung

- In (S+) wird das Kommando für das FX3U-CF-ADP eingetragen.
 - Wird in (S+) der Wert „-1“ eingetragen, überträgt die FLCMD-Anweisung alle Daten aus dem internen Speicher des FX3U-CF-ADP in die CF-Speicherkarte.
 - Enthält (S+) einen Wert aus dem Bereich von 0 bis 63 (Datei-Nummer), überträgt die FLCMD-Anweisung die im internen Speicher des FX3U-CF-ADP gespeicherten Daten der entsprechenden Datei in die CF-Speicherkarte.
 - Beim Wert „256“ (100H) in (S+) versetzt die FLCMD-Anweisung die CF-Speicherkarte in den Status „installiert“. In diesem Zustand kann die SPS auf die Speicherkarte zugreifen.
 - Beim Wert „512“ (200H) in (S+) versetzt die FLCMD-Anweisung die CF-Speicherkarte in den Status „nicht installiert“. In diesem Zustand kann die SPS nicht auf die Speicherkarte zugreifen.
 - Enthält (S+) den Wert „1280“ (500H), löscht die FLCMD-Anweisung die im FX3U-CF-ADP gespeicherten Fehlercodes.
- Mit n wird die Nummer des Kommunikationskanal angegeben, den das Adaptermodul FX3U-CF-ADP belegt (siehe folgende Tabelle).

Operand	Beschreibung	Datentyp
(S+)	Funktionsauswahl <ul style="list-style-type: none"> • -1 (FFFFH): Alle Daten im internen Speicher des FX3U-CF-ADP in die CF-Speicherkarte übertragen. • 0 bis 63: Die im internen Speicher des FX3U-CF-ADP abgelegten Daten der Datei mit der entsprechenden Datei-Nummer in die CF-Speicherkarte übertragen. • 256 (100H): CF-Speicherkarte in den Status „installiert“ versetzen • 512 (200H): CF-Speicherkarte in den Status „nicht installiert“ versetzen • 1280 (500H): Im FX3U-CF-ADP gespeicherte Fehlercodes löschen 	BIN-16-Bit
n	Kanalnummer des FX3U-CF-ADP <ul style="list-style-type: none"> • 1: Ch. 1 • 2: Ch. 2 	BIN-16-Bit

Tab. 7-101: Operanden der FLCMD-Anweisung

7.25.6 Status des FX3U-CF-ADP lesen (FLSTRD)

				FLSTRD	FNC 305											
				Status des FX3U-CF-ADP lesen												
				CPU	FX1S	FX1N	FX2N FX2NC	FX3G FX3GC FX3GE	FX3S	FX3U FX3UC						
																●
Operanden	S+	D+	n	Puls-Anweisung (P)		Verarbeitung		Programmschritte								
	D*, R, K, H	D*, R	K, H			16 Bit	32 Bit	FLSTRD	7							
						●										

* Sonderregister können nicht angegeben werden.

Funktion

Eine FLSTRD-Anweisung liest den Status (einschließlich Informationen über Dateien und Fehler) aus dem FX3U-CF-ADP.

Beschreibung

- In (S+) wird angegeben, welche Informationen aus dem FX3U-CF-ADP gelesen werden sollen.
- In (D+) wird der Anweisung die erste Adresse des Bereichs übergeben, in dem die aus dem FX3U-CF-ADP gelesenen Informationen gespeichert werden sollen.
- Die Anzahl der nach (D+) belegten Operanden und deren Inhalt hängen vom Inhalt von (S+) ab.
 - Enthält (S+) einen Wert aus dem Bereich von 0 bis 63 (Datei-Nummer), liest die FLSTRD-Anweisung die letzte Zeilenposition und die letzte Spaltenposition der entsprechenden Datei und speichert sie in (D+) und (D+)+1.

Operand	Beschreibung
(D+)	Letzte Zeilenposition Wertebereich: „1“ bis zur eingestellten maximalen Anzahl der Zeilen in der Datei
(D+)+1	Letzte Spaltenposition

- Beim Wert „256“ (100H) in (S+) liest die FLSTRD-Anweisung die Nummern der in der CF-Speicherkarte gespeicherten Dateien. In (D+)+1 bis (D+)+3 wird für jede existierende Datei-Nummer ein Bit besetzt. Die Zuordnung der Bits zu den Datei-Nummern zeigt die folgende Tabelle. Die Ziffern 0 bis 64 stehen für die Datei-Nummer.

Operand	Bits															
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
(D+)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
(D+)+1	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
(D+)+2	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
(D+)+3	63	63	61	60	59	58	57	56	55	54	53	52	51	50	49	48

- Wenn (S+) den Wert „512“ (200H) enthält, liest die FLSTRD-Anweisung die Informationen über die Speicherkapazität der CF-Speicherkarte und speichert sie in den folgenden Operanden:

Operand	Beschreibung
(D+)+1, (D+)	Gesamte Speicherkapazität der CF-Speicherkarte in der Einheit „kByte“ Bei einer geringeren Speicherkapazität als 1 kB wird der Wert „1“ gespeichert.
(D+)+3, (D+)+2	Belegte Speicherkapazität der CF-Speicherkarte in der Einheit „kByte“ Bei einer geringeren Datengröße als 1 kB wird der Wert „1“ gespeichert.
(D+)+5, (D+)+4	Freie Speicherkapazität der CF-Speicherkarte in der Einheit „kByte“ Ist die freie Kapazität geringer als 1 kB, wird der Wert „1“ gespeichert.

- Beim Wert „768“ (300H) in (S+) liest die FLSTRD-Anweisung die Version des Speicherkartenadaptermoduls FX3U-CF-ADP und speichert sie in (D+).

Operand	Beschreibung
(D+)	Version des CF-Speicherkartenadaptermoduls FX3U-CF-ADP Zum Beispiel wird bei der Version 1.00 der Wert „100“ gespeichert.

- Wenn (S+) den Wert „1024“ (400H) enthält, liest die FLSTRD-Anweisung Fehlerinformationen aus dem FX3U-CF-ADP und setzt in (D+) das entsprechende Bit.

Operand	Beschreibung
(D+)	Fehlerinformationen <ul style="list-style-type: none"> • Bit 0: Keine CF-Speicherkarte installiert • Bit 1: Die CF-Speicherkarte ist voll • Bit 2: Im FX3U-CF-ADP ist ein Fehler aufgetreten • Bit 3: Hardware-Fehler des FX3U-CF-ADP • Bit 4: Fehler der CF-Speicherkarte Die Bits 5 bis 15 werden nicht verwendet.

- Beim Wert „1280“ (500H) in (S+) liest die FLSTRD-Anweisung die Fehlercodes und detaillierte Informationen zu Fehlern, die im Adaptermodul FX3U-CF-ADP aufgetreten sind. Es werden die Fehlercodes und Informationen der letzten fünf Fehler gespeichert.

Operand	Beschreibung	
(D+)	Fehler 1	Fehlercode
(D+)+1		Detaillierte Informationen
(D+)+2	Fehler 2	Fehlercode
(D+)+3		Detaillierte Informationen
(D+)+4	Fehler 3	Fehlercode
(D+)+5		Detaillierte Informationen
(D+)+6	Fehler 4	Fehlercode
(D+)+7		Detaillierte Informationen
(D+)+8	Fehler 5	Fehlercode
(D+)+9		Detaillierte Informationen

- Mit n wird die Nummer des Kommunikationskanal angegeben, den das Adaptermodul FX3U-CF-ADP belegt (siehe folgende Tabelle).

Operand	Beschreibung	Datentyp
(S+)	Festlegung, welche Informationen aus dem Adaptermodul FX3U-CF-ADP gelesen werden sollen. <ul style="list-style-type: none"> • 0 bis 63: Letzte Zeilen- und Spaltenposition der entsprechenden Datei • 256 (100H): Datei-Nummern in der CF-Speicherkarte • 512 (200H): Speicherkapazität der CF-Speicherkarte • 768 (300H): Version des FX3U-CF-ADP • 1024 (400H): Fehlerinformationen des FX3U-CF-ADP • 1280 (500H): Fehlercodes 	BIN-16-Bit
(D+)	Erste Adresse des Operandenbereichs, in dem die aus dem FX3U-CF-ADP gelesenen Informationen gespeichert werden sollen.	BIN-16-Bit
n	Kanalnummer des FX3U-CF-ADP <ul style="list-style-type: none"> • 1: Ch. 1 • 2: Ch. 2 	BIN-16-Bit

Tab. 7-102: Operanden der FLSTRD-Anweisung

8 Sonderfunktionen

Die Steuerungen der FX-Familie stellen einige Sonderfunktionen zur Verfügung, mit denen Sie die Einsatzmöglichkeiten der Steuerung erweitern können. Diese Sonderfunktionen sind deshalb in einem eigenen Kapitel zusammengefasst, weil sie nicht unmittelbar durch eine bestimmte Anweisung ausgeführt werden.

In diesem Kapitel werden folgende Sonderfunktionen beschrieben:

- Datenerhalt im STOP-Modus
- Betrieb mit konstanter Programmzykluszeit
- Pulse-Catch-Funktion
- Eingangsfiler einstellen
- Integrierte Potentiometer bei FX1S und FX1N
- Integrierte Uhr
- File-Register
- RUN/STOP-Umschaltung
- 24 V DC-Grundgeräte
- Anzeigemodul FX1N-5DM für FX1S und FX1N

8.1 Datenerhalt im STOP-Modus

Im „normalen“ Betriebsablauf setzen Steuerungen der FX-Familie alle Ausgangssignalzustände auf „0“, sobald die Steuerung vom RUN- in den STOP-Modus geschaltet wird. Für einige Anwendungen ist es jedoch sinnvoll, auch im STOP-Modus die zuletzt vorhandenen Ausgangssignalzustände aufrecht zu erhalten. Dies können Sie erreichen, wenn Sie den Sondermarker M8033 im SPS-Programm setzen. Die Istwerte von Timern und Countern bleiben dann ebenfalls erhalten.

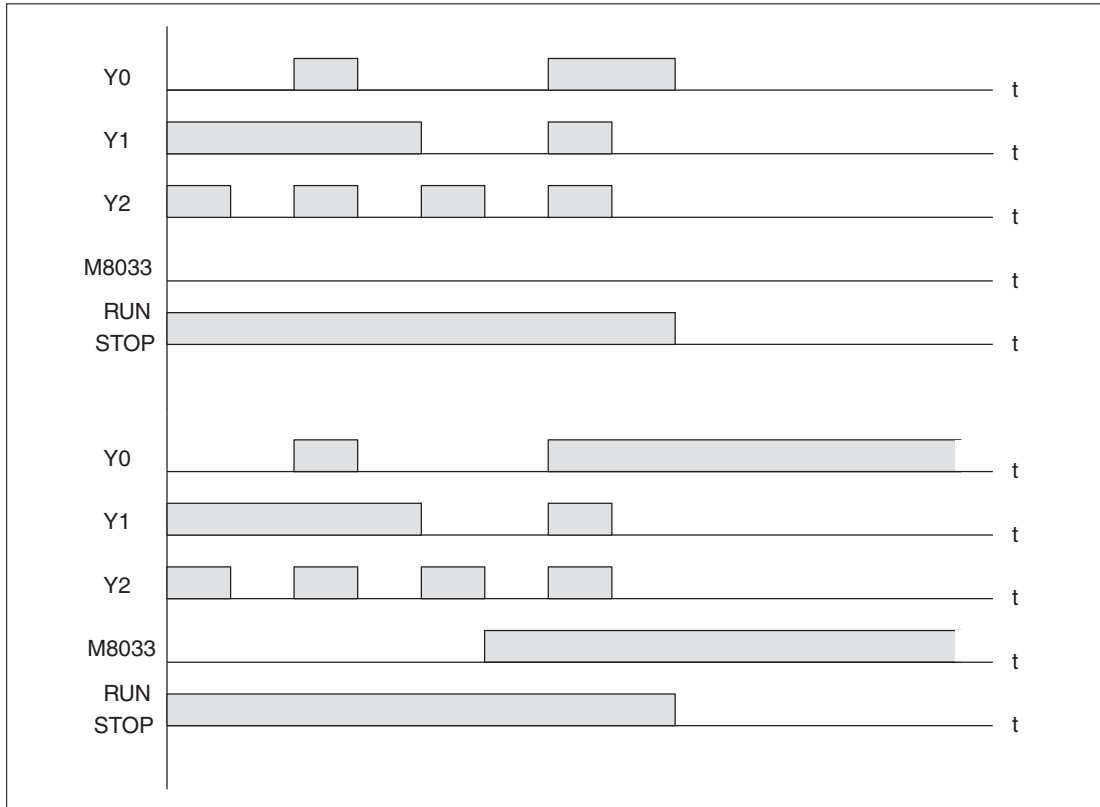


Abb. 8-1: Beispiel für Datenerhalt im STOP-Modus

Das folgende Beispiel zeigt die notwendige Programmierung:

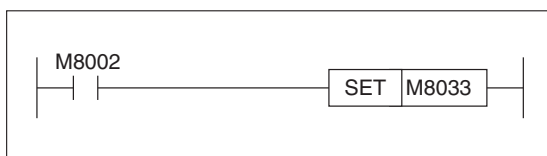


Abb. 8-2: Programmierbeispiel

8.2 Betrieb mit konstanter Programmzykluszeit

Die Steuerungen der FX-Familie können mit einer konstanten, vom SPS-Programm unabhängigen, Programmzykluszeit betrieben werden. Dies ist z. B. beim Einsatz der RAMP-Applikationsanweisung notwendig.

Um eine konstante Programmzykluszeit zu erreichen, muss der Sondermerker M8039 im SPS-Programm gesetzt werden. Die Programmzykluszeit kann in Einheiten von 1 ms festgelegt werden. Der Wert der gewählten Programmzykluszeit wird in das Datenregister D8039 eingetragen. Wählen Sie die Programmzykluszeit größer als die durchschnittliche Programmzykluszeit. Der Wert der durchschnittlichen Programmzykluszeit wird von der SPS automatisch im Datenregister D8010 abgespeichert.

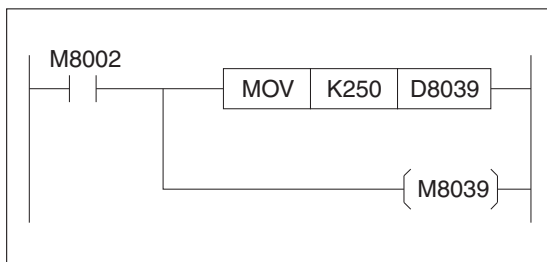


Abb. 8-3:

Festlegen einer konstanten Programmzykluszeit auf den Wert 250 ms im Datenregister D8039

8.3 Pulse-Catch-Funktion

Mit der Pulse-Catch-Funktion können sehr kurzzeitige Eingangssignalimpulse, wie z. B. Lichtschrankensignale, von der Steuerung verarbeitet werden.

Die Pulse-Catch-Funktion kann nur für Signale benutzt werden, die über die Eingänge X0 bis X5 (X0 bis X7 bei FX3U und FX3UC) in die Steuerung eingespeist werden. Es kann in jedem Programmzyklus jeweils nur 1 Impuls verarbeitet werden.

Die minimale Impulslängen, die noch von der Steuerung verarbeitet werden können, betragen z. B. bei der FX3-Serie:

SPS	Minimale Impulslänge an Eingang							
	X0	X1	X2	X3	X4	X5	X6	X7
FX3G FX3GG FX3GE	10 µs		50 µs	10 µs		50 µs	—	—
FX3S	10 µs		50 µs				—	—
FX3U FX3UC	5 µs						50 µs	

Tab. 8-1: Minimal erfassbare Impulslängen

Bei jedem Impuls an einem der Eingänge wird automatisch von der Steuerung ein Sondermerker gesetzt. Dieser Sondermerker kann dann im Programm weiterverarbeitet werden. Damit die Steuerung einen neuen Impuls an einem Eingang erkennen kann, muss der zugehörige Sondermerker vorher im Programm zurückgesetzt werden.

Eingang	X0	X1	X2	X3	X4	X5	X6*	X7*
Sondermerker	M8170	M8171	M8172	M8173	M8174	M8175	M8176*	M8177*

Tab. 8-2: Eingänge und zugehörige Sondermerker

* Nur bei FX3U und FX3UC

Damit ein Impuls erfasst werden kann, muss – außer bei der FX1S-, FX1N-, FX3G-, FX3GC-, FX3G- und FX3S-Serie – vorher eine EI-Anweisung ausgeführt werden.

Die folgende Abbildung zeigt das Prinzip der Pulse-Catch-Funktion am Beispiel eines Signals am Eingang X0.

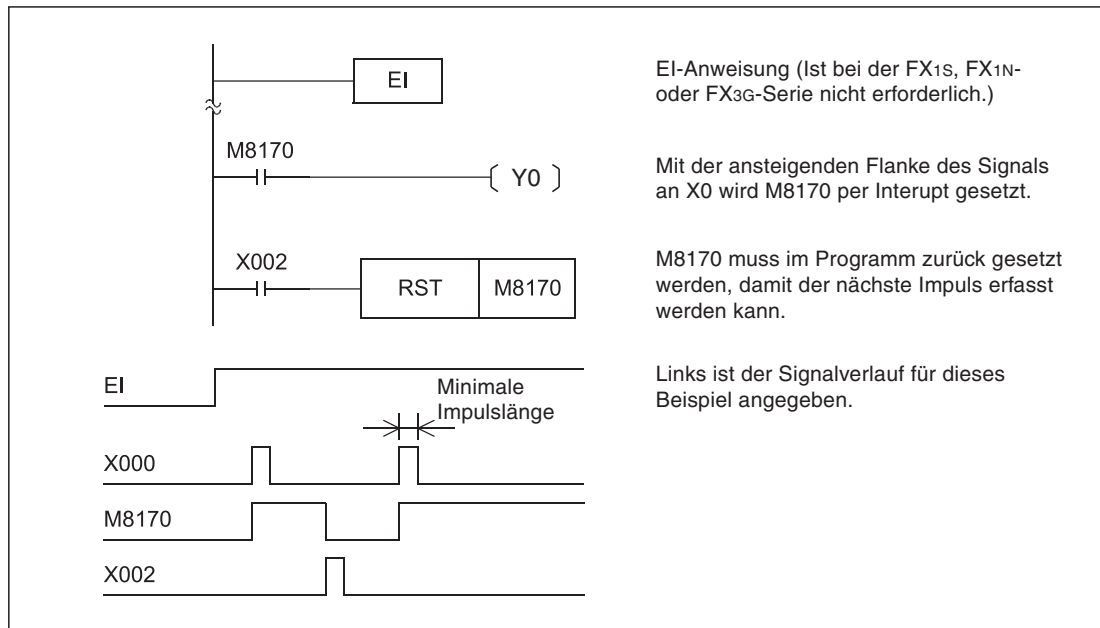


Abb. 8-4: Beispielprogramm und Signalverlauf für die Pulse-Catch-Funktion

HINWEISE

Die Pulse-Catch-Funktion ist keine High-Speed-Funktion. Es kann in jedem Programmzyklus jeweils nur 1 Eingangsimpuls verarbeitet werden.

Beachten Sie, dass die Eingänge X0 bis X5 bzw. X7 nicht gleichzeitig als Interrupt-Eingänge für die Pulse-Catch-Funktion und als Zählengänge für High-Speed-Counter eingesetzt werden können. Eine Doppelbelegung von Eingängen ist nicht zulässig.

Bei Steuerungen der FX2N-, FX2NC-, FX3U- und FX3UC-Serie müssen zur Nutzung der Pulse-Catch-Funktion Interrupts durch eine EI-Anweisung freigegeben werden.

Kontinuierliche Impulse mit hoher Frequenz sollten per Eingangs-Interrupt oder High-Speed-Counter erfasst werden.

Für die Pulse-Catch-Funktion ist die Einstellung der Filterzeiten für die Eingänge nicht erforderlich.

Der Zustand der Sondermerker M8050 bis M8055 (Interrupt-Programme sperren/freigeben) hat keinen Einfluss auf die Pulse-Catch-Funktion.

Beispiel ▾

Einsatz der Pulse-Catch-Funktion zum Zählen von Impulsen an einer Lichtschranke über den Eingang X3

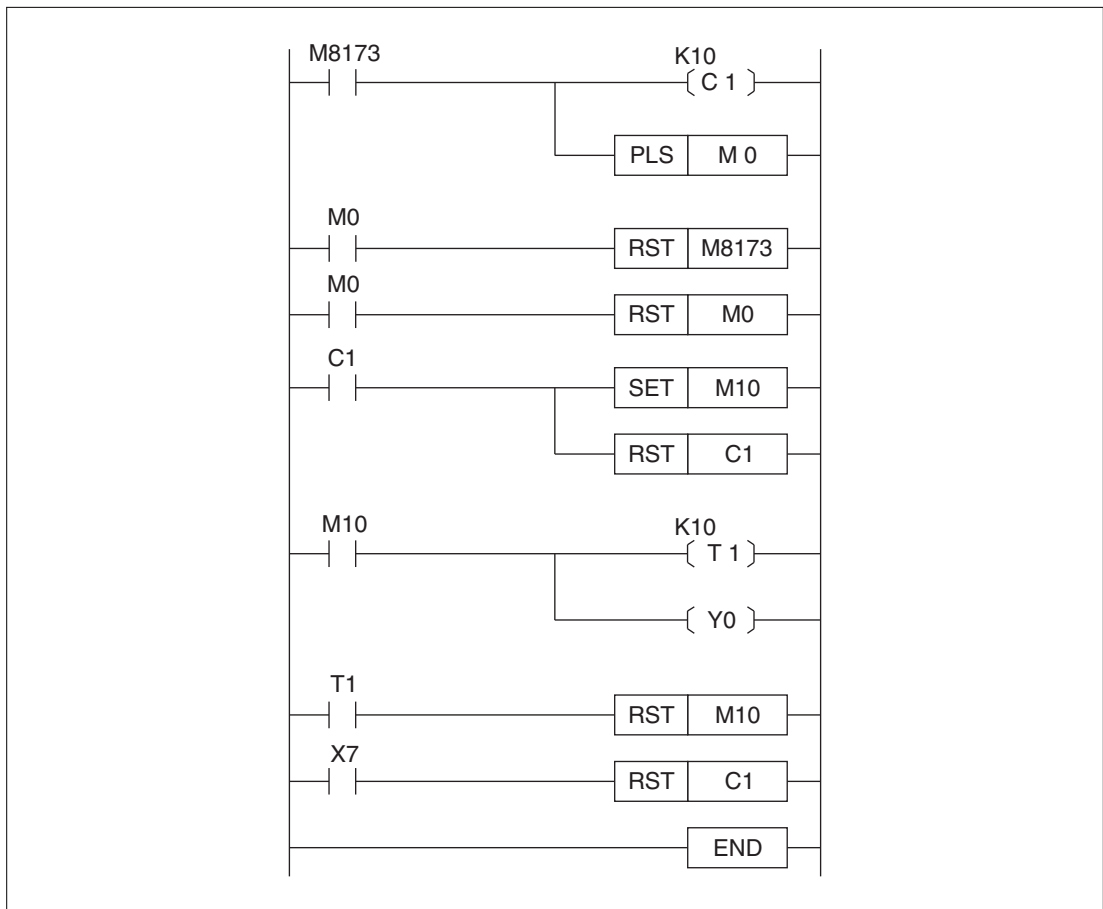


Abb. 8-5: Programmierbeispiel für den Einsatz der Pulse-Catch-Funktion und das Zählen von Impulsen einer Lichtschranke über den Eingang X3

Nach 10 Impulsen wird der Ausgang Y0 für 1s gesetzt. Der Zähler C1 wird über den Eingang X7 zurückgesetzt.

△

8.4 Impulsweiten- und Periodendauermessung

Mit der im Abschnitt 8.3 beschriebenen Pulse-Catch-Funktion kann die FX-SPS auf kurze Eingangsimpulse reagieren. Bei Steuerungen der FX3G-, FX3GC- und FX3GE-Serie kann zusätzlich auch die Länge eines Eingangsimpulses oder die Periodendauer von Impulsen (und damit auch die Frequenz) erfasst werden.

Bei der Impulsweiten- und Periodendauermessung wird der Wert von 1/6 μ s-Ringzählern bei der ansteigenden und der abfallenden Flanke eines Eingangssignals in Sonderregister gespeichert. Aus der Differenz dieser Werte ergibt sich die Impulsweite. Die Differenz der Zählerwerte bei der vorherigen und der aktuellen ansteigende Flanke der Eingangsimpulse entspricht der Periodendauer. Die Werte für die Impulsweite und die Periodendauer werden von der FX3G durch „60“ geteilt und stehen danach in der Einheit „10 μ s“ zur Verfügung.

Die minimale Impulsweite, die gemessen werden kann, ist 10 μ s. Maximal kann ein Impuls 100 s lang sein. Die minimale erfassbare Periodendauer ist 20 μ s.

Die Impulsweiten- und Periodendauermessung wird aktiviert, indem der Sondermerker M8075 im Programm als Eingangsbedingung verwendet wird. In der folgenden OUT-Anweisung wird ein Sondermerker verwendet, der einem der Eingänge X0, X1, X3 oder X4 zugeordnet ist. Die Messung wird dann für die Impulse an diesem Eingang ausgeführt. Die auf den folgenden Seiten gezeigten Beispiele verdeutlichen diese Zusammenhänge.

Impuls- eingang	Start der Impuls- weiten-/Perioden- dauermessung	Periodendauer messen*	Ringzählerwert für steigende Flanke* [1/6 μ s]	Ringzählerwert für fallende Flanke* [1/6 μ s]	Impulsweite/ Periodendauer [10 μ s]
X0	M8076	M8080	D8075, D8074	D8077, D8076	D8079, D8078
X1	M8077	M8081	D8081, D8080	D8083, D8082	D8085, D8084
X3	M8078	M8082	D8087, D8086	D8089, D8088	D8091, D8090
X4	M8079	M8083	D8093, D8092	D8095, D8094	D8097, D8096

Tab. 8-3: Sondermerker und -register für die Impulsweiten- und Periodendauermessung

* Diese Merker und Register werden bei einem Stopp der SPS zurückgesetzt bzw. gelöscht.

HINWEISE

Die Impulsweite oder die Periodendauer eines Signals kann auf diese Weise nur von einer FX3G, FX3GC oder FX3GE gemessen werden. Die oben aufgeführten Sondermerker und -register haben in den übrigen FX-Grundgeräten andere Funktionen.

Die Impulsweiten-/Periodendauermessung und Eingangs-Interrupts können für den selben Eingang gleichzeitig verwendet werden.

Werden die Impulsweiten-/Periodendauermessung und eine der Anweisungen SPD, DSZR oder ZRN für den selben Eingang ausgeführt, tritt ein Fehler auf.

Ein Eingang, der für die Impulsweiten-/Periodendauermessung verwendet wird, kann nicht für die Pulse-Catch-Funktion verwendet werden.

Wird derselbe Eingang für die Impulsweiten-/Periodendauermessung und für einen High-Speed-Counter verwendet, tritt ein Fehler auf.

Wird die Impulsweiten-/Periodendauermessung zusammen mit High-Speed-Countern verwendet, wird die Gesamtfrequenz der High-Speed-Counter beeinflusst.

Bei der Impulsweiten-/Periodendauermessung darf die Gesamtfrequenz der Impulse an den vier Eingängen 50 kHz nicht überschreiten.

Beispiel ▾

Impulsweitenmessung

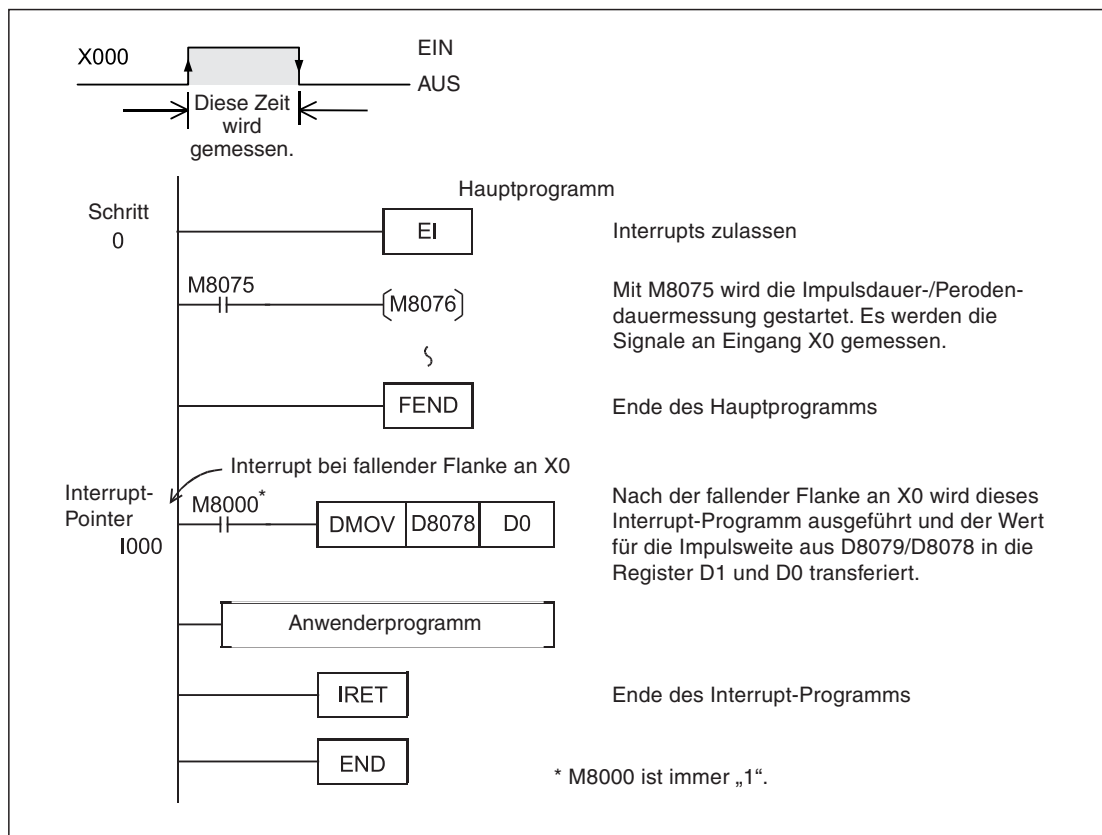


Abb. 8-6: Beispielprogramm zur Messung der Impulsweite eines Signals am Eingang X0



Beispiel ▾ Periodendauermessung

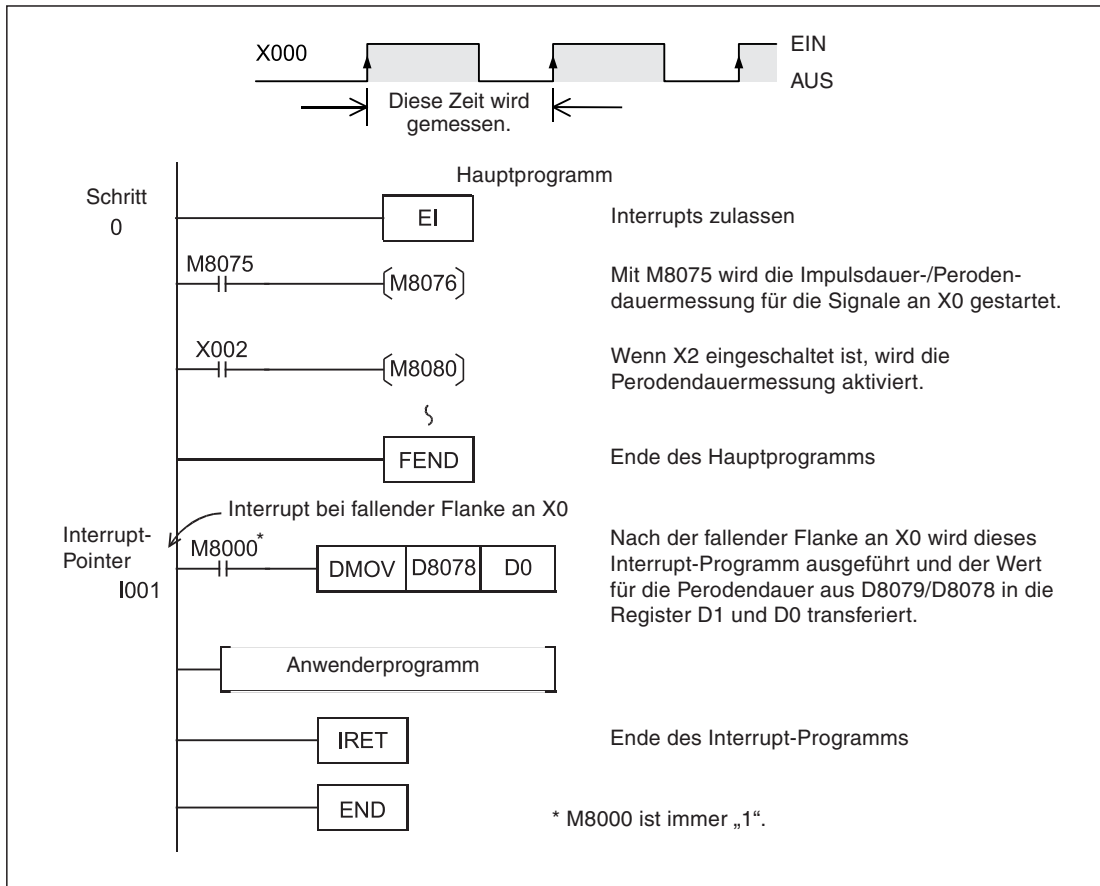


Abb. 8-4: Beispielprogramm zur Messung der Periodendauer eines Signals am Eingang X0

Die erste ansteigende Flanke des Eingangssignals nach dem Übergang der SPS in den RUN-Modus oder nach der Aktivierung der Periodendauermessung (im oben abgebildeten Beispiel durch Setzen von M8080) wird nicht zur Messung der Periodendauer verwendet und die entsprechenden Sonderregister (z.B. D8079/D8078) werden nicht aktualisiert. Die Periodendauermessung beginnt in diesen Fällen bei der nächsten steigenden Flanke.

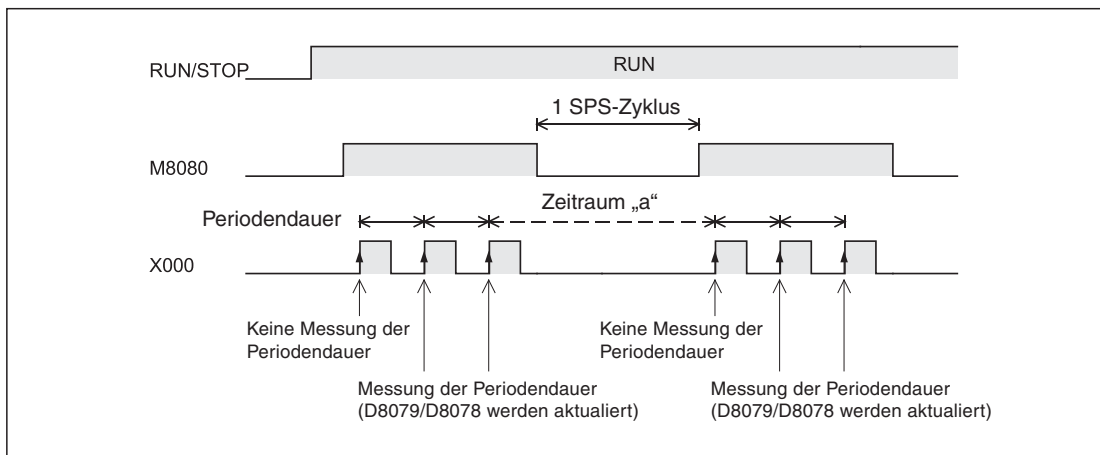


Abb. 8-7: Zeitlicher Verlauf der Signale für das oben abgebildete Beispiel

Falls die Messung der Periodendauer unterbrochen werden soll, muss der Sondermerker zur Aktivierung der Periodendauerermessung (z.B. M8080) für mindestens einen SPS-Zyklus

zurückgesetzt werden. Wird dieser Sondermerker nur für eine kürzere Zeit zurückgesetzt, wird der in der Abbildung oben dargestellte Zeitraum „a“ als Periodendauer gespeichert.



Beispiel ▾

Messung der zeitlichen Abstands zweier Signale

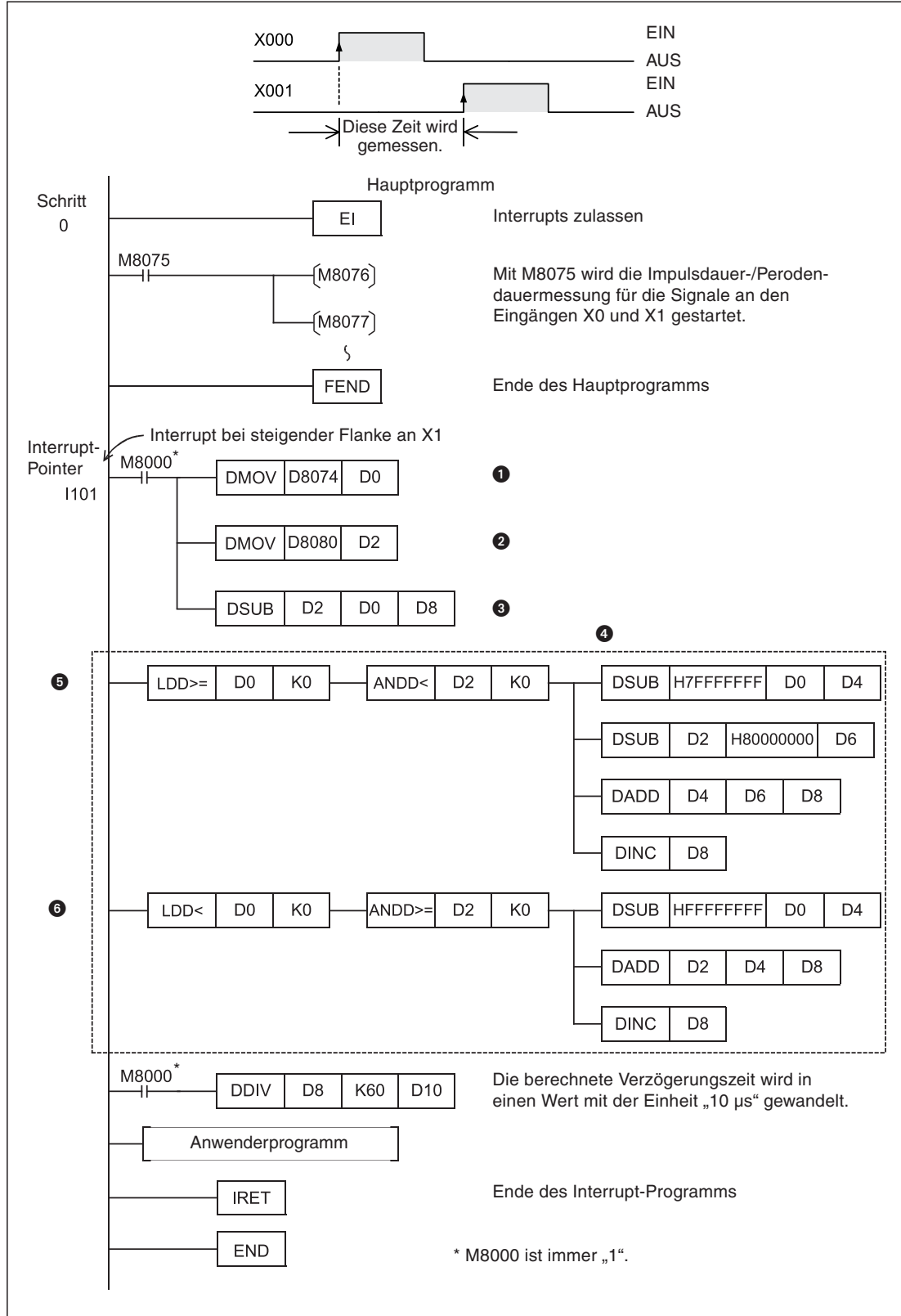


Abb. 8-8: Beispiel zur Messung der Verzögerung beim Einschalten zweier Eingänge

- ① Nach der steigenden Flanke an X1 wird dieses Interrupt-Programm ausgeführt und der Ringzählerwert bei steigender Flanke an X0 von D8075/D8074 in die Register D1 und D0 transferiert.
- ② Der Ringzählerwert bei steigender Flanke an X1 wird aus den Sonderregistern D8081 und D8080 in die Register D3 und D2 übertragen.
- ③ Der „Ringzählerwert bei steigender Flanke an X0“ wird vom „Ringzählerwert bei steigender Flanke an X1“ abgezogen und das Ergebnis in die Register D9 und D8 eingetragen.
- ④ Der Wert der Ringzähler umfasst 32 Bit, einschließlich des höchstwertigen Bits. Weil eine DSUB-Anweisung das höchstwertige Bit als Vorzeichenbit interpretiert, ist das Ergebnis der Subtraktion bei bestimmten Ringzählerwerten nicht korrekt. Um das richtige Ergebnis zu erhalten, muss die folgende, mit einer Strichlinie eingerahmte, Programmsequenz eingefügt werden.
- ⑤ Wenn der „Ringzählerwert bei steigender Flanke an X0“ zwischen 0H und 7FFFFFFFH und der „Ringzählerwert bei steigender Flanke an X1“ zwischen 80000000H und FFFFFFFFH liegt, wird die Verzögerungszeit neu berechnet und das Ergebnis in die Register D9 und D8 eingetragen.
- ⑥ Wenn der „Ringzählerwert bei steigender Flanke an X0“ zwischen 80000000H und FFFFFFFFH und der „Ringzählerwert bei steigender Flanke an X1“ zwischen 0H und 7FFFFFFFH liegt, wird die Verzögerungszeit erneut berechnet und das Ergebnis in die Register D9 und D8 eingetragen.

8.5 EingangsfILTER einstellen

8.5.1 FX1S-, FX1N-, FX3G-, FX3GC-, FX3GE-, FX3S-, FX3U- und FX3UC-Serie

Die Verzögerungszeit der 24-V-Eingänge wird in der SPS in den Eingangsfiltern festgelegt. Die Verzögerungszeit lässt sich in 1-ms-Schritten von 0 bis 60 ms festlegen (0 bis 15 ms bei FX3G/FX3GC/FX3GE/FX3S). Der Standardwert beträgt 10 ms.

Die Filterzeit der Eingänge X0 bis X17* wird im Sonderregister D8020 festgelegt.

* X0 bis X7 bei einem FX3G-, FX3GC-, FX3GE-Grundgerät und den Grundgeräten der FX3U- und der FX3UC-Serie mit 16 Ein- und Ausgängen.

Bei jedem Einschalten der Spannungsversorgung des SPS-Grundgeräts wird in das Sonderregister D8020 der Standardwert von 10 ms eingetragen.

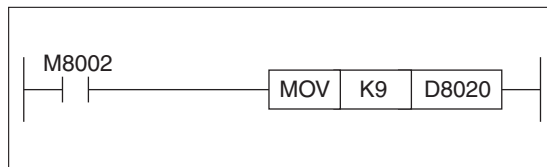


Abb. 8-9:

Einstellung der Verzögerungszeit der Eingangssignalverarbeitung für die Eingänge X0 bis X7 auf den Wert 9 ms im Datenregister D8020

HINWEISE

Die Verzögerungszeit lässt sich nur blockweise für die Eingänge X0 bis X7 bzw. X10 bis X17 einstellen.

Wenn der Wert „0“ eingetragen wird, beträgt die Verzögerungszeit bei der FX1S- und FX1N-Serie für X0 und X1 10 μ s und für die restlichen Eingänge 50 μ s.

- Bei der FX3G-, FX3GC- und FX3GE-Serie beträgt die Verzögerungszeit bei der Vorgabe des Werts „0“ in D8020 10 μ s für X0, X1, X3 und X4 sowie 50 μ s für X2, X5, X6 und X7.
- Bei der FX3S-Serie beträgt die Verzögerungszeit bei Vorgabe des Werts „0“ in D8020 10 μ s für X0 und X1, 50 μ s für X2 bis X7 sowie 200 μ s für X10 bis X17.
- Bei der FX3U- und FX3UC-Serie beträgt die Verzögerungszeit bei der Vorgabe des Werts „0“ in D8020 5 μ s für X0 bis X5, 50 μ s für X6 und X7 sowie 200 μ s für X10 bis X17 (Bei den FX3U-/FX3UC-Grundgeräten mit 16 E/A sind nur die Eingänge X0 bis X7 einstellbar).

Wenn Sie sehr kurze Verzögerungszeiten einstellen ($\leq 5 \mu$ s), müssen Sie gewährleisten, dass die Eingangssignale nicht durch Störungen beeinflusst werden. Hierdurch kann es zu Programmablauffehlern kommen. Verwenden Sie kurze Leitungen (max. 5 m) und installieren Sie am Eingang einen Widerstand (1,5 k Ω , 1/2 W), damit von dem am Eingang angeschlossenen Gerät ein Mindestlaststrom von 20 mA geliefert wird (siehe Hardware-Beschreibung der verwendeten SPS).

Die Einstellung der EingangsfILTERzeit ist für die folgenden Funktionen nicht erforderlich:

- Verwendung eines Eingangs zur Auslösung eines Interrupts
- Verwendung eines Eingangs im Zusammenhang mit High-Speed-Countern
- Erfassung von Impulsen mit der Pulse-Catch-Funktion (siehe Abschnitt 8.3)

8.5.2 FX2N -, FX2NC-, FX3U- und FX3UC-Serie

Bei der FX2N- und FX2NC-Serie werden die EingangsfILTER über die Anweisung REFF eingestellt (siehe Abschnitt 6.7.2).

Bei den Grundgeräten der FX3U- und FX3UC-Serie können die EingangsfILTER auch durch die Anweisung REFF eingestellt werden (siehe Abschnitt 6.7.2).

8.6 Integrierte Potentiometer

Bei den Steuerungen der bei FX1S-, FX1N-, FX3G-, FX3GE- und FX3S-Serie kann der Inhalt zweier Datenregister durch analoge Potentiometer im Bereich von 0 bis 255 verändert werden (siehe auch Abschnitt 3.8.5).

FX1S FX1N FX3G FX3GE FX3S	Potentiometer 1 (VR1)	D8030
	Potentiometer 2 (VR2)	D8031

Tab. 8-5:
Datenregister der Potentiometer

Beispiel ▾ 100-ms-Timer mit variabler Zeit

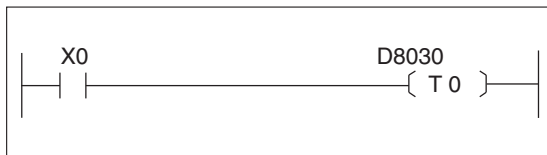


Abb. 8-10:
Timer mit Sollwert, der durch das Potentiometer VR1 verändert werden kann

Wenn D8030 den Inhalt „200“ hat, läuft beim Einschalten von X0 eine Zeit von 20 s ab.



Beispiel ▾ Wird eine kürzere oder eine kleiner abgestufte Zeit benötigt, kann ein 10-ms-Timer eingesetzt werden.

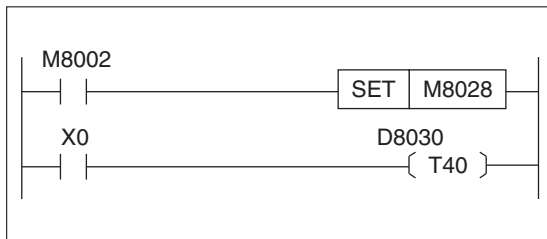


Abb. 8-11:
10-ms-Timer mit variabler Zeit (FX1S)



8.7 Integrierte Uhr der SPS

In den Steuerungen der FX-Familie ist eine Uhr eingebaut. Diese Uhr ist batteriegepuffert und läuft auch weiter, wenn die Versorgungsspannung der Steuerung abgeschaltet ist.

Funktionen

Die batteriegepufferte Uhr liefert Informationen zum Datum und zur Uhrzeit.

Schaltjahre werden im Zeitraum von 1980 bis 2079 berücksichtigt.

Sonderregister	Zeit	Einstellung
D8013	Sekunden	0–59
D8014	Minuten	0–59
D8015	Stunden	0–23
D8016	Tag	1–31
D8017	Monat	1–12
D8018	Jahr	0–99 (1980–2079)
D8019	Wochentag	0–6 (Sonntag–Samstag)

Tab. 8-6:

Register für die interne Uhr der FX-Familie. Hier können Werte zum Stellen der Uhr eingetragen oder die aktuellen Uhrzeit und das Datum entnommen werden. D8018 kann auf eine vierstellige Anzeige umgeschaltet werden (siehe unten).

Genauigkeit

Genauigkeitsschwankungen liegen beim Einsatz des Geräts in einem Bereich von 25°C bei ± 45 s innerhalb eines Monats.

Sondermerker für den Betrieb der internen Uhr

Sondermerker	Bedeutung	Beschreibung
M8015	Uhr anhalten Zeit einstellen	Ist M8015 eingeschaltet („1“), stoppt die Uhr. Mit der fallenden Flanke von M8015 (Wechsel von „1“ nach „0“) werden die Daten aus D8013 bis D8019 in die Uhr übernommen und die Uhr wieder gestartet.
M8016	Datenerhalt	Ist M8016 eingeschaltet („1“), werden die Daten in den entsprechenden Sonderregistern „eingefroren“. Die Uhr läuft aber weiter.
M8017	Auf-/Abrunden der Minuten	Mit der Steigenden Flanke von M8017 (Wechsel von „0“ nach „1“) wird die Uhr auf die nächste Minute vor- oder zurückgestellt. (Bei einer Sekundenangabe von 0 bis 29 wird diese auf 0 gesetzt. Bei einer Sekundenangabe von 30 bis 59 wird diese ebenfalls auf 0 gesetzt, aber die Minutenangabe um 1 erhöht.)
M8018	Uhr aktiviert	M8018 wird automatisch gesetzt („1“), um anzuzeigen, dass die Uhrfunktion aktiviert ist.
M8019	Fehlerhafte Daten	M8019 wird eingeschaltet, wenn die eingegebenen Daten außerhalb des zulässigen Bereichs liegen.

Tab. 8-7: Bedeutung der Sondermerker für die interne Uhr

HINWEIS

Bei den Steuerungen der MELSEC FX-Familie können spezielle Anweisungen zum Stellen und Auswerten der internen Uhr verwendet werden. Nähere Informationen enthält der Abschnitt 7.10.

Angabe der Jahreszahl in den vierstelligen Modus einschalten

Mit der folgenden Programmsequenz kann die Anzeige der Jahreszahl in D8018 in den vierstelligen Modus umgeschaltet werden.

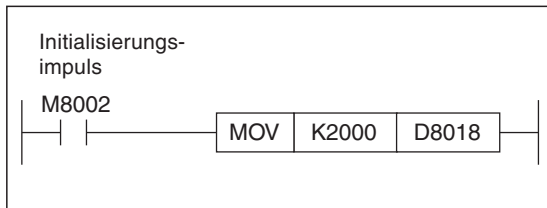


Abb. 8-12:

Mit diesem Programm steht ab dem zweiten Programmzyklus in D8018 die Jahreszahl als vierstelliger Wert zur Verfügung.

HINWEISE

Die SPS zeigt normalerweise die Jahreszahl als zweistelligen Wert an. Wird in der Betriebsart RUN der feste Wert 2000 mit einer MOV-Anweisung in das Sonderregister D8018 (Angabe der Jahreszahl) transferiert, wird die SPS dadurch in den vierstelligen Modus umgeschaltet.

Lassen Sie die oben abgebildete Programmsequenz jedenmal ausführen, wenn die SPS in die Betriebsart RUN geschaltet wird. Mit ihr wird nur der Modus bei der Anzeige der Jahreszahl umgeschaltet, die Uhrzeit und das Datum werden dadurch nicht beeinflusst.

Im vierstelligen Modus entsprechen die Werte „80“ bis „99“ den Jahren 1980 bis 1999 und die Werte „00“ bis „79“ den Jahren 2000 bis 2079.
Beispiele: 80 = 1980, 99 = 1999, 00 = 2000, 79 = 2079.

Beim Anschluss eines Handprogrammiergeräts FX-10DU-E/20DU-E/25DU-E muss für die Jahreszahl der zweistellige Modus verwendet werden. Im vierstelligen Modus wird die Jahreszahl vom Handprogrammiergerät nicht korrekt angezeigt. Beachten Sie bitte auch, dass die Uhr in den zweistelligen Modus für die Jahreszahl geschaltet wird, falls sie durch ein Handprogrammiergerät FX-10DU-E/20DU-E/25DU-E eingestellt wird, während die SPS im vierstelligen Modus ist.

8.8 RUN-/STOP-Umschaltung

Die Steuerungen der MELSEC FX-Familie können durch Eingangssignale in die Betriebsart RUN oder STOP geschaltet werden. Bei einer kurzzeitigen Betätigung des RUN-Tasters geht die SPS in die Betriebsart RUN und bei einer kurzzeitigen Betätigung des STOP-Tasters in die Betriebsart STOP.

Für die Betriebsartumschaltung stehen die drei Sondermerker M8035 bis M8037 zur Verfügung:

Sondermerker	RUN	STOP
M8035	1	0
M8036	1	0
M8037	0	1

Tab. 8-8:
Bedeutung der Sondermerker

Beschaltung und Programmbeispiel

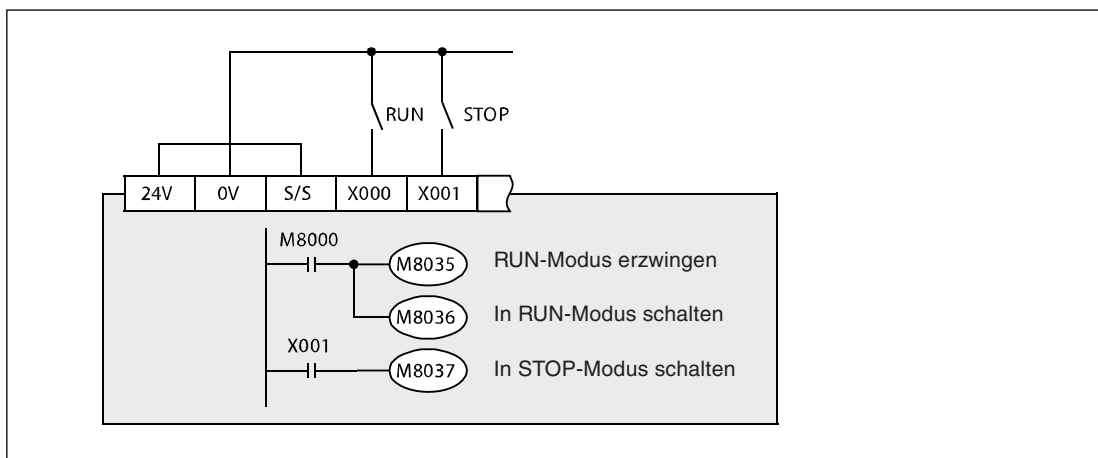


Abb. 8-13: Programmbeispiel und Beispiel zur Verdrahtung der Eingänge bei einem FX3U-Grundgerät mit minusschaltenden Eingängen

Vorbereitung und Einstellung in den SPS-Parametern

- ① Stellen Sie den RUN/STOP-Schalter der SPS in die Stellung „STOP“.
- ② Wählen Sie den für den RUN-Taster verwendeten Eingang (im Beispiel oben X0) in den SPS-Parametern auf der Registerkarte „SPS-System (1)“ als „RUN-Modus Kontakt“. Bitte beachten Sie, dass die zur Verfügung stehenden Eingänge vom Typ des SPS-Grundgeräts abhängen.
- ③ Als Eingang für den STOP-Taster kann ein beliebiger Eingang in der SPS verwendet werden. (Im Beispiel oben wurde X1 gewählt.)
- ④ Übertragen Sie das Programm und die Parameter in die SPS.
- ⑤ Damit die geänderten Parameter gültig werden, muss die Versorgungsspannung der SPS aus- und wieder eingeschaltet werden.

HINWEISE

Werden der RUN- und der STOP-Taster gleichzeitig betätigt, hat der STOP-Taster Vorrang.

Befindet sich der in der SPS integrierte RUN/STOP-Schalter in der Stellung RUN, kann die SPS in den RUN-Modus gebracht werden. Wird aber der STOP-Taster betätigt, geht die SPS in die Betriebsart STOP. (Befindet sich der RUN/STOP-Schalter in der Stellung RUN, hat der STOP-Taster Vorrang.)

RUN/STOP-Umschaltung durch ein Programmierwerkzeug

- Verwendung der Programmier-Software für PC
In der Programmier-Software ist eine Funktion integriert, mit der die Betriebsart der SPS vom PC aus umgeschaltet werden kann.
- Verwendung eines anderen Programmierwerkzeugs
Wenn M8035 (RUN-Modus erzwingen) und M8036 (In RUN-Modus schalten) den Zustand „1“ haben, geht die SPS in die Betriebsart RUN. Wenn M8037 (In STOP-Modus schalten) in den Zustand „1“ gebracht wird, wechselt die SPS in die Betriebsart STOP.
- Auch wenn sich der in der SPS integrierte RUN/STOP-Schalter in der Stellung RUN befindet, hat das STOP-Signal vom Programmierwerkzeug oder M8037 (In STOP-Modus schalten) Vorrang.

8.9 FX2N-Grundgeräte mit 24 V DC-Versorgung

Um einen einwandfreien Betrieb dieser Geräte zu gewährleisten, muss folgende Programmzeile an den Anfang des SPS-Programms eingefügt werden.

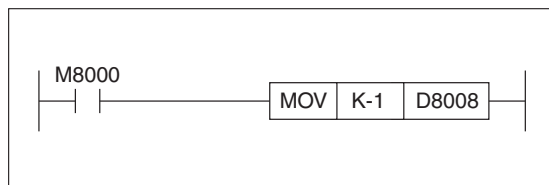


Abb. 8-14:
Programmierbeispiel

Bei AC-gespeisten Grundgeräten der FX2N-Serie kann die Erfassung der Spannungsausfallzeit durch Eintrag des gewünschten Erfassungszeitraums in das Sonderregister D8008 vorgegeben werden.

Bei DC-gespeisten Modulen muss dieser Erkennungszeitraum auf 5 ms festgelegt werden. Dies wird durch den Eintrag des Wertes -1 in D8008 erreicht.

Ohne diesen Eintrag kann es bei Spannungsausfällen der DC-Versorgung zu fehlerhaften Erfassungsdaten kommen.

8.10 Anzeigemodul FX1N-5DM für FX1S und FX1N

Das Anzeigemodul FX1N-5DM kann zur Beobachtung der Zustände von Operanden und zur Eingabe von Daten bei den Steuerungen der FX1S- und FX1N-Serie eingesetzt werden.

8.10.1 Funktionen

Funktionen zur Bedienung

Die folgenden Funktionen können nur durch die Tasten am Anzeigemodul FX1N-5DM angewählt und ausgeführt werden. Nähere Hinweise zur Verwendung der Funktionstasten finden Sie in der Bedienungsanleitung des Anzeigemoduls.

Funktion		Beschreibung
Uhr	Anzeige	Anzeige der in der FX1S/FX1N integrierten Uhr
	Stellen	Einstellen des Datums und der Uhrzeit
Beobachten von Operanden	Bit-Operanden	Anzeige des Zustandes von Eingängen (X), Ausgängen (Y), Merkern (M) und Sondermerkern (S)
	Word-Operanden (16 Bit)	Istwerte von Timern (T), Zählern (C) und Registern (D) anzeigen Die Sollwerte der Timer und Zähler können eingestellt werden.
	Word-Operanden (32 Bit)	Anzeige der Istwerte von 32-Bit-Zählern und -Registern (D) Einstellung der Sollwerte der 32-Bit-Zähler
Beobachten des Pufferspeichers (nur bei FX1N)		Anzeige des Pufferspeicherinhaltes von Sondermodulen
Fehleranzeige		Anzeige von Fehlercodes und der Schrittnummer des Fehlers
Forcen von Operanden		Zwangswaises Ein- und Ausschalten von Bit-Operanden (X, M, S)
Rücksetzen von Timern und Zählern		Löschen des Istwertes von Timern und Zählern
Verändern von Daten	Istwerte	Änderung der Istwerte von Timern, Zählern und Register
	Sollwerte	Änderung der Sollwerte von Timern und Zählern

Tab. 8-9: Funktionen, die durch die Funktionstasten angewählt werden können

Funktionen zur Steuerung des Anzeigemoduls

Das Anzeigemodul kann durch das SPS-Programm gesteuert werden.

Funktion	Beschreibung
Schutz vor unbefugten Zugriffen	Wahlweise können alle Funktionen, nur Beobachtungsfunktionen oder nur die Anzeige von Datum und Uhrzeit freigegeben werden.
Festlegung der angezeigten Operanden	Der Anwender kann die Art und die Adresse der Operanden festlegen, die angezeigt werden.
Ein- und Ausschalten der Fehleranzeige	Wahlweise kann die Anzeige von Fehlercode ein- oder ausgeschaltet werden.
Automatisches Ausschalten der Beleuchtung der Anzeige	Einstellung der Zeit, nach der sich die Hintergrundbeleuchtung der Anzeige automatisch ausschaltet
Erfassung der Funktionstasten	Der Zustand (EIN oder AUS) der vier Funktionstasten des Anzeigemoduls kann ausgewertet werden

Tab. 8-10: Funktionen, die durch das SPS-Programm ausgeführt werden können

HINWEIS

Wenn das SPS-Programm durch ein Passwort gegen unbefugtes Lesen oder Schreiben geschützt ist, wird mit dem Anzeigemodul nur das Datum und die Uhrzeit dargestellt. Wenn in diesem Fall eine andere Funktion angewählt wird, blinkt die Anzeige für ca. 5 Sekunden.

8.10.2 Operanden zur Steuerung des Anzeigemoduls

Die Sonderregister D8158 und D8159 dienen zur Steuerung des Anzeigemoduls. In diesen Registern wird die Anfangsadresse eines Register- bzw. eines Merkerbereiches als Konstante eingetragen. Diese beiden Bereiche (5 Register und 15 Merker) enthalten die Einstellungen zur Steuerung des Anzeigemoduls.

Sonderregister		Operand zur Steuerung	Funktion	Beschreibung
Adresse	Inhalt			
D8158	Kn	Dn	Festlegung der angezeigten Operanden	Art der Operanden
		Dn + 1		Adresse des Operanden
		Dn + 2	Automatisches Ausschalten der Beleuchtung der Anzeige	Zeit bis zum Ausschalten der Beleuchtung [Minuten]
		Dn + 3	Schutz vor unbefugten „Zugriffen“	Schutzfunktion
		Dn + 4	Reserviert	
D8159	Kn	Mn	Festlegung der angezeigten Operanden	Freigabe zur Änderung von Operanden
		Mn + 1		Rückmeldung: Änderung abgeschlossen
		Mn + 2	Automatisches Ausschalten der Beleuchtung der Anzeige	Automatisches Ausschalten sperren, Beleuchtung einschalten
		Mn + 3	Einschalten der Fehleranzeige	
		Mn + 4	Abfrage der Funktionstasten des Anzeigemoduls. Solange eine Taste betätigt wird, ist der entsprechende Merker gesetzt.	Status der „ESC“-Taste
		Mn + 5		Status der „-“-Taste
		Mn + 6		Status der „+“-Taste
		Mn + 7		Status der „OK“-Taste
		Mn + 8	Reserviert	
		Mn + 9		
		Mn + 10		
		Mn + 11		
		Mn + 12		
		Mn + 13		
		Mn + 14		

Tab. 8-11: Register- und Merkerbereich zur Steuerung der Anzeige

Wenn in D8158 oder D8159 ein negativer Wert oder ein Wert, der außerhalb des Register- oder Merkerbereiches liegt, eingetragen wird, werden die Steuerfunktionen nicht ausgeführt. Nach dem Einschalten der Steuerung ist der Inhalt der Register D8158 und D8159 „-1“, die Steuerfunktionen sind abgeschaltet.

8.10.3 Festlegung der angezeigten Operanden

Durch einen Eintrag in das erste Register des in D8158 angegebenen Registerbereiches können die Operanden festgelegt werden, die mit dem Anzeigemodul dargestellt werden.

Wert in Dn	Ausgewählter Operand
1	Eingänge (X)
2	Ausgänge (Y)
3	Merker (M)
4	Schrittmerker (S)
5	Timer (T)
6	Zähler (C), Soll- und Istwerte von 16-Bit-Zählern und Sollwerte von 32-Bit-Zählern
7	16-Bit-Register (D)
8	32-Bit-Register (D)
9	Datum und Uhrzeit
10	Zähler (C), Soll- und Istwerte von 16-Bit-Zählern und Istwerte von 32-Bit-Zählern

Tab. 8-12:

Auswahl der Operanden durch Eintrag eines Wertes in Dn

HINWEISE

Wenn in dem Register ein Wert eingetragen wird, der außerhalb des Bereiches 1 bis 10 liegt, wird kein Operand ausgewählt und alle Bedienerfunktionen sind erlaubt.

Wenn bei der Festlegung von Timer- und Zähleradressen ein Operand angegeben wird, der nicht im Programm verwendet wird, wird der Operand angezeigt, dessen Adresse dem gewünschten Operanden am nächsten liegt und der im Programm verwendet wird.

Liegt die angegebene Adresse außerhalb des zulässigen Bereiches, wird der Operand mit der höchsten Adresse angezeigt, der im Programm verwendet wird.

Wenn für den gewählten Timer oder Zähler keine OUT-Anweisung im Programm vorhanden ist, wird „----“ angezeigt.

Mit der „+“- und der „-“-Taste am Anzeigemodul können aufeinanderfolgende Operanden ausgewählt werden.

Wenn die Operanden nicht aufeinanderfolgend sind und trotzdem die Auswahl mit der „+“- und der „-“-Taste erfolgen soll, muss das SPS-Programm erweitert werden. Die Operandenadressen, die angezeigt werden sollen, werden dabei in einem Index-Register abgelegt. Eine Betätigung der „+“- und der „-“-Taste wird vom Programm erfasst, der Inhalt des Registers wird erhöht oder erniedrigt und dadurch die Operandenadresse verändert.

Der Inhalt der Registerbereiches wird auf „0“ gesetzt, wenn die Spannungsversorgung der Steuerung ausgeschaltet wird und die Register nicht im gepufferten Bereich liegen. Dadurch wird kein Operand mehr ausgewählt und alle Bedienerfunktionen sind erlaubt.

Wählen Sie aus diesem Grund für die Steuerfunktionen einen Registerbereich, der auch bei ausgeschalteter Spannung seine Daten nicht verliert.

8.10.4 Schutz vor unbefugten „Zugriffen“

In das vierte Register ($Dn + 3$) des in D8158 angegebenen Registerbereiches wird ein Wert eingetragen, mit dem die Funktionen eingeschränkt werden können.

Wert in $Dn + 3$	Beschreibung
0	Kein Schutz, alle Funktionen sind erlaubt
1	Nur das Datum und die Uhrzeit wird angezeigt, das Datum und die Uhrzeit können nicht verändert werden.
2	Nur die Beobachtung von Operanden ist möglich, die Werte der Operanden können nicht verändert werden.
Alle anderen Werte	Kein Schutz, alle Funktionen sind erlaubt

Tab. 8-13: Schutzfunktionen

8.10.5 Änderung des Zustandes oder der Werte von Operanden

Diese Funktion ermöglicht dem Bediener die Änderung der angezeigten Operanden mit Hilfe der Funktionstasten.

Sonderregister		Operand zur Steuerung	Beschreibung
Adresse	Inhalt		
D8159	Kn	Mn	Freigabe zur Änderung von Operanden
		$Mn + 1$	Rückmeldung: Änderung abgeschlossen

Tab. 8-14: Merker zur Änderung von Operanden

Um den Wert eines Operanden zu verändern, muss der Merker Mn gesetzt sein. Bei zurückgesetztem Merker Mn ist die Funktion gesperrt. Aus diesem Grund wird die Anwendung der SET-Anweisung empfohlen.

Bei gesetztem Merker Mn können Ausgänge (Y), Merker (M) und Schrittmerker (S) gesetzt oder rückgesetzt werden und die Soll- und Istwerte der Wortoperanden D, T und C verändert werden.

Um Bit-Operanden zu ändern, wird der blinkende Cursor unter den Operanden plaziert. Durch Betätigung der „OK“-Taste wird der Operanden gesetzt oder zurückgesetzt. Mit der „ESC“-Taste wird der Editier-Modus verlassen. Der Merker ($Mn + 1$) wird ausgeschaltet und Mn wird zurückgesetzt.

Nach Betätigung der „OK“-Taste wird bei Wortoperanden der aktuelle Wert blinkend dargestellt. Mit der „+“- und der „-“-Taste kann der Wert verändert werden. Nach erneuter Betätigung der „OK“-Taste wird der Wert übernommen.

Wenn die „ESC“-Taste vor der „OK“-Taste betätigt wird, wird die Änderung abgebrochen. Nach der Übernahme des Wertes wird mit der „ESC“-Taste der Editier-Modus verlassen. Der Merker ($Mn + 1$) wird ausgeschaltet und Mn wird zurückgesetzt.



Wenn zur Auswahl der Operanden die Betätigung der „+“- und der „-“-Taste vom SPS-Programm erfasst wird, muss eine Verriegelung für den Fall vorgesehen werden, dass mit den beiden Tasten der Zustand oder der Wert von Operanden geändert wird.

8.10.6 Automatisches Ausschalten der Beleuchtung der Anzeige

Die Beleuchtung der Anzeige schaltet sich nach einer bestimmten Zeit automatisch aus. Diese Zeit wird in einem Register vorgegeben. Zusätzlich kann die Beleuchtung ständig ein- oder ausgeschaltet werden.

Sonderregister		Operand zur Steuerung	Funktion	Beschreibung
Adresse	Inhalt			
D8158	Kn	Dn + 2	Zeit bis zum Ausschalten der Beleuchtung [Minuten]	0 (Voreinstellung): 10 Minuten 1 bis 240: 1 bis 240 Minuten >240: 240 Minuten Negativer Wert: Ständig AUS
D8159	Kn	Mn + 2	Automatisches Ausschalten sperren, Beleuchtung einschalten	0: Automatisches Ausschalten freigegeben 1: Automatisches Ausschalten gesperrt, Licht ist immer eingeschaltet

Tab. 8-15: Register und Merker zur Steuerung der Beleuchtung

HINWEIS

Nachdem die Beleuchtung verloschen ist, wird sie wieder eingeschaltet, sobald eine Taste des Anzeigemoduls betätigt wird. Die Anzeige zeigt dasselbe wie vor dem Ausschalten der Beleuchtung. Diese erste Tastenbetätigung dient nur zum Einschalten der Beleuchtung und hat sonst keine Funktion.

8.10.7 Freigeben und Sperren der Fehleranzeige

Durch Setzen oder Rücksetzen des Merkers (Mn + 3) kann ausgewählt werden, welche Fehlermeldungen angezeigt werden.

Sonderregister		Operand zur Steuerung	Funktion	Beschreibung
Adresse	Inhalt			
D8159	Kn	Mn + 3	Einschalten der Fehleranzeige	0: Anzeige von PLC-Hardware-, Parameter-, Programmsyntax- und Programmierfehlern 1: Zusätzliche Anzeige von Konfigurations-, Parallel-Link-Fehlern und Fehlern beim Betrieb

Tab. 8-16: Register- und Merkerbereich zur Steuerung der Anzeige

Bei Betätigung einer beliebigen Taste des Anzeigemoduls oder wenn der Fehler nicht mehr ansteht, verschwindet die Fehleranzeige.

Treten mehrere Fehler gleichzeitig auf, hat die Anzeige der Fehlercodes Vorrang, die unbedingt (auch wenn (Mn + 3) = 0 ist) angezeigt werden. Zusätzlich hat der Fehler mit dem niedrigeren Fehlercode höhere Priorität.

9 Sondermerker, Sonderregister

9.1 Sondermerker (M8000 bis M8511)

Durch Einsatz von Sondermerkern können Sie bestimmte SPS-Betriebszustände in einem SPS-Programm abfragen bzw. ein- oder ausschalten.

Die Sondermerker lassen sich in zwei Gruppen einteilen:

- ❶ Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontakthanweisung (z. B. LD- oder LDI-Anweisung) abgefragt werden kann.
- ❷ Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

HINWEIS

Sondermerker, die Hardware- oder Programmablauf-Fehler anzeigen, sind im folgenden Kapitel 10 beschrieben.

9.1.1 SPS-Status (M8000–M8009)

Sondermerker Operanden-adresse	① Signal-zustand abfragen	② Signal-zustand festlegen	CPU	Bedeutung
M8000	●	—	FX1S FX1N FX2N FX2NC FX3G FX3GC FX3GE FX3S FX3U FX3UC	SPS-Status anzeigen: RUN (Ist immer „1“ im RUN-Modus)
M8001	●	—		SPS-Status anzeigen: RUN (Ist immer „0“ im RUN-Modus)
M8002	●	—		Initialisierungs-impuls
M8003	●	—		Initialisierungs-impuls
M8004	●	—		SPS-Fehler*
M8005	●	—	FX2N FX2NC FX3G FX3GC FX3GE FX3U FX3UC	Der Merker wird gesetzt, wenn die Batteriespannung den in D8006 angegebenen Wert unterschreitet.
M8006	●	—		Speichert den Fehler „niedrige Batteriespannung“
M8007	●	—	FX2N FX2NC FX3U FX3UC	Wird bei kurzzeitigem Spannungsausfall gesetzt
M8008	●	—		Meldet einen Spannungsausfall
M8009	●	—	FX2N FX2NC FX3G FX3GC FX3GE FX3U FX3UC	Meldet den Ausfall der 24-V-DC Servicespannung

RUN-Modus

M8004

M8000

M8001

M8002

M8003

Tab. 9-1: Sondermerker für SPS-Status

* M8004 wird bei einer FX1S, FX1N, FX2N oder FX2NC gesetzt, wenn M8060, M8061, M8064, M8065, M8066 oder M8067 gesetzt ist.
Bei einer FX3G/FX3GC/FX3GE/FX3S/FX3U oder FX3UC wird M8004 gesetzt, wenn M8060, M8061, M8063, M8064, M8065, M8066 oder M8067 gesetzt ist.

- ① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontaktanweisung (z. B. LD- oder LDI-Anweisung) abgefragt werden kann.
- ② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

HINWEIS

Die Beschreibung der Sondermerker M8005 bis M8009 ist den entsprechenden Bedienungsanleitungen (Hardware-Beschreibungen) zu den CPUs zu entnehmen.

9.1.2 Zeittakte und Uhr (M8011–M8019)

Sondermerker Operanden- adresse	① Signal- zustand abfragen	② Signal- zustand festlegen	CPU	Bedeutung
M8010	—	—	—	Reserviert
M8011	●	—	FX1S FX1N FX2N FX2NC FX3G FX3GC FX3GE FX3S FX3U FX3UC	Taktgeber: 10 ms Es wird ein Takt von 10 ms erzeugt.
M8012	●	—		Taktgeber: 100 ms Es wird ein Takt von 100 ms erzeugt.
M8013	●	—		Taktgeber: 1 s Es wird ein Takt von 1 s erzeugt.
M8014	●	—		Taktgeber: 1 min Es wird ein Takt von 1 min erzeugt.
M8015	●	●		Zeiteinstellung* Ist der Merker gesetzt, stoppt die Uhr. Wird der Merker zurückgesetzt, werden die Daten aus D8013 bis D8019 für die Uhr übernommen und die Uhr läuft weiter.
M8016	●	—		Registerdaten* Ist der Merker gesetzt, werden die Inhalte von D8013 bis D8019 eingefroren, die Uhr läuft aber weiter.
M8017	●	●		Minuten runden* Wenn M8017 gesetzt ist, wird die Minutenanzeige auf den vollen Minutenbetrag gerundet.
M8018	●	—		Uhr ist installiert (Ist immer „1“.)*
M8019	●	—		Einstellfehler* Die Einstellung der Uhrzeitdaten erfolgte außerhalb des erlaubten Bereiches.

Tab. 9-2: Sondermerker für Zeittakte und integrierter Uhr

* Bei einer FX2NC muss eine Speicherkarte mit integrierter Uhr installiert werden.

- ① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontaktanweisung (z. B. LD- oder LDI-Anweisung) abgefragt werden kann
- ② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können

HINWEIS

Die Funktion der Merker M8015 bis M8019 ist im Abschnitt 8.7 beschrieben.

9.1.3 Flags (M8020–M8029)

Sondermerker Operanden- adresse	① Signal- zustand abfragen	② Signal- zustand festlegen	CPU	Bedeutung
M8020	●	—	FX1S FX1N FX2N	Das Zero Flag wird gesetzt, wenn das Ergebnis einer Addition oder Subtraktion gleich Null ist.
M8021	●	—	FX2NC FX3G FX3GC	Das Borrow Flag wird gesetzt, wenn das Ergebnis einer Subtraktion (FNC21) kleiner als der kleinste zulässige negative Wert ist.
M8022	●	—	FX3GE FX3S FX3U FX3UC	Das Carry Flag wird gesetzt bei einem Zahlenwertübertrag bei einer Addition oder bei einem Datenübertrag während der Ausführung einer Verschiebeanweisung.
M8023	—	—	—	Reserviert
M8024 ^①	●	●	FX1S FX1N FX2N FX2NC FX3G FX3GC FX3GE FX3S FX3U FX3UC	Ausführungsrichtung der BMOV-Anweisung: M8024 = 0: Transfer von (S+) nach (D+) M8024 = 1: Transfer von (D+) nach (S+)
M8025 ^②	—	●	FX2N FX2NC FX3U FX3UC	Bei gesetztem Merker werden alle HSC-Anweisungen (FNC 53–55) verarbeitet, sobald der externe HSC-Rücksetzeingang aktiviert wird.
M8026 ^②	—	—		RAMP-Anweisung (FNC 67) wird gehalten.
M8027 ^②	—	—		Daten-String für 16 Elemente in der PR-Anweisung (FNC 77).
M8028	—	—	FX1S	Die Verwendung der 10-ms-Timer T32 bis T62 wird ermöglicht.
	—	—	FX2N (ab V3.0) FX2NC (ab V3.0) FX3G FX3GC FX3GE FX3U FX3UC	M8028 = 0: FROM/TO-Anweisungen werden bei einem Interrupt nicht unterbrochen M8028 = 1: FROM/TO-Anweisungen werden bei einem Interrupt unterbrochen und nach der Ausführung des Interrupt-Programms fortgesetzt.
M8029	●	—	FX1S FX1N FX2N FX2NC FX3G FX3GC FX3GE FX3S FX3U FX3UC	Anweisung vollständig abgearbeitet

Tab. 9-3: Sondermerker zur Anzeige und zur Einstellung von besonderen Zuständen

- ① Bei einer FX1S, FX1N, FX2N oder FX2NC wird dieser Merker bei einem STOP der SPS nicht zurückgesetzt. Bei einer FX3G, FX3GC, FX3GE, FX3S, FX3U oder FX3UC wird dieser Merker bei einem STOP der SPS zurückgesetzt.
- ② Bei einer FX2N oder FX2NC wird dieser Merker bei einem STOP der SPS nicht zurückgesetzt. Bei einer FX3U oder FX3UC wird dieser Merker bei einem STOP der SPS zurückgesetzt.
- ① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontaktanweisung (z. B. LD- oder LDI-Anweisung) abgefragt werden kann.
- ② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

9.1.4 SPS-Modus (M8030–M8039)

Sondermerker Operanden- adresse	① Signal- zustand abfragen	② Signal- zustand festlegen	CPU	Bedeutung
M8030	●	—	FX2N, FX2NC FX3G, FX3GC FX3GE, FX3U, FX3UC	BATT. LED ausschalten Wenn M8030 „1“ ist, leuchtet die BATT. LED bei zu niedriger Batteriespannung nicht.
M8031	●	●	FX1S FX1N FX2N FX2NC FX3G FX3GC FX3GE FX3S FX3U FX3UC	Alle Operanden zurücksetzen, deren Datenwerte nicht im gelatchten Speicher abgespeichert werden*.
M8032	●	●		Alle Operanden zurücksetzen, deren Datenwerte im gelatchten Speicher gespeichert werden*.
M8033	●	●		Datenwerterhalt im STOP-Modus Der Inhalt des Image-Registers und des Datenspeichers bleibt erhalten, wenn die SPS vom RUN- in den STOP-Modus geschaltet wird.
M8034	●	●		Verhindern der Ausgabe: Alle Ausgänge werden ausgeschaltet, das Programm wird aber weiter abgearbeitet
M8035	●	●		RUN/STOP-Modus zwangsweise setzen
M8036	●	●		Erzwungener RUN-Modus durch Setzen des Sondermerkers im SPS-Programm
M8037	●	●		Erzwungener STOP-Modus durch Setzen des Sondermerkers im SPS-Programm
M8038	—	●		FX1S, FX1N FX2N (ab V2.0), FX2NC, FX3G, FX3GC, FX3GE, FX3S, FX3U, FX3UC
M8039	●	●	FX1S, FX1N FX2N, FX2NC, FX3G, FX3GC, FX3GE, FX3S, FX3U, FX3UC	SPS-Programm mit konstanter Programmzykluszeit. Wird M8039 gesetzt, arbeitet die SPS das Programm mit einer konstanten Programmzykluszeit ab, dessen Wert im Datenregister D8039 gespeichert ist.

Tab. 9-4: Sondermerker für SPS-Modus

* Es werden die Zustände der Operanden Y, M, S, T und C und die Istwerte von Timern und Countern sowie Datenregister (D) und erweiterte Register (R, nur bei FX3G, FX3GC, FX3GE, FX3U und FX3UC) gelöscht. Bei einer FX3G, FX3GC, FX3GE, FX3S, FX3U oder FX3UC werden auch Sonderregister gelöscht.
File-Register (D) im Programmspeicher und erweiterte File-Register in einer Speicherkassette (ER, nur bei FX3G, FX3U und FX3UC) werden nicht gelöscht.

- ① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontaktanweisung (z. B. LD- oder LDI-Anweisung) abgefragt werden kann.
- ② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

9.1.5 STL-Status (M8040–M8049)

Sondermerker Operanden- adresse	① Signal- zustand abfragen	② Signal- zustand festlegen	CPU	Bedeutung
M8040	●	●	FX1S FX1N FX2N FX2NC FX3G FX3GC FX3GE FX3S FX3U FX3UC	Weiterschaltung nicht möglich Wird M8040 gesetzt, ist eine Weiterschaltung nicht möglich.
M8041 ^①	●	●		Beginn einer Weiterschaltung Die Weiterschaltung ist während des Automatikbetriebes möglich.
M8042	●	●		Startimpuls Bei entsprechender Eingangsbedingung wird ein Startimpuls gegeben.
M8043 ^①	●	●		Rückkehr zur Ausgangsposition ausgeführt M8043 wird gesetzt, wenn die Ausgangsposition erreicht ist.
M8044 ^①	●	●		Bedingung zur Rückkehr in die Ausgangsposition M8044 wird gesetzt, wenn die Ausgangsposition erkannt wurde.
M8045	●	●		Rücksetzen aller Ausgänge ist nicht möglich Wird M8045 gesetzt, ist es nicht möglich, alle Ausgänge zurückzusetzen.
M8046 ^②	●	—		STL-Status ist gesetzt M8046 wird gesetzt, wenn einer der Schrittstatusoperanden S0 bis S899 oder S1000 bis S4095 ^③ und M8047 gesetzt sind.
M8047 ^②	●	●		STL-Status anzeigen Wird M8047 gesetzt, wird in den Datenregistern D8040 bis D8047 der Schrittstatus der ersten 8 Schrittstatusoperanden angezeigt.
M8048 ^②	●	—	FX2N FX2NC FX3G FX3GC FX3GE FX3U FX3UC	Fehlermerker anzeigen Der Merker ist gesetzt, wenn die Überwachung der Fehlermerker (M8049) aktiviert ist und ein Fehlermerker gesetzt wird.
M8049 ^①	—	●	FX3G FX3GC FX3U FX3UC	Anzeige der Fehlermerker ermöglichen Bei gesetztem Merker kann über D8049 die Anzeige der Fehlermerker aktiviert werden.

Tab. 9-5: Sondermerker für STL-Status

① Diese Merker werden bei einem Stopp der SPS zurückgesetzt.

② Wird bei der Bearbeitung der END-Anweisung ausgeführt.

③ S1000 bis S4095 stehen nur bei einer FX3G, FX3GC, FX3GE, FX3U oder FX3UC zur Verfügung.

① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontaktanweisung (z. B. LD- oder LDI-Anweisung) abgefragt werden kann.

② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

9.1.6 Interrupt-Programm (M8050–M8059)

Sondermerker Operanden- adresse	① Signal- zustand abfragen	② Signal- zustand festlegen	CPU	Bedeutung
M8050*	●	●	FX1S	Interrupt-Programm I00** wird nicht ausgeführt.
M8051*	●	●	FX1N FX2N	Interrupt-Programm I10** wird nicht ausgeführt.
M8052*	●	●	FX2NC FX3G	Interrupt-Programm I20** wird nicht ausgeführt.
M8053*	●	●	FX3GC	Interrupt-Programm I30** wird nicht ausgeführt.
M8054*	●	●	FX3GE	Interrupt-Programm I40** wird nicht ausgeführt.
M8055*	●	●	FX3S FX3U FX3UC	Interrupt-Programm I50** wird nicht ausgeführt.
M8056*	●	●	FX2N FX2NC	Interrupt-Programm I6** wird nicht ausgeführt.
M8057*	●	●	FX3G FX3GC FX3GE	Interrupt-Programm I7** wird nicht ausgeführt.
M8058*	●	●	FX3S FX3U FX3UC	Interrupt-Programm I8** wird nicht ausgeführt.
M8059*	●	●	FX2N FX2NC FX3U FX3UC	Schaltet die Interrupts I010 bis I060 aus.

Tab. 9-6: Sondermerker für Interrupt-Programm

* Diese Merker werden bei einem Stopp der SPS zurückgesetzt.

- ① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontaktanweisung (z. B. LD- oder LDI-Anweisung) abgefragt werden kann.
- ② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

HINWEIS

Wird eine EI-Anweisung (FNC 04) im Anwenderprogramm ausgeführt, werden alle Interrupts solange freigegeben, bis einer der Sondermerker M8050 bis M8058 gesetzt wird. In diesem Fall wird für jeden gesetzten Sondermerker der zugehörige Interrupt gesperrt, das heißt, er kann nicht aktiviert werden.

9.1.7 Sondermerker für Fehlermeldungen (M8060–M8069)

Sondermerker Operanden- adresse	① Signal- zustand abfragen	② Signal- zustand festlegen	CPU	Bedeutung
M8060	●	—	FX2N, FX2NC FX3G, FX3GC, FX3GE FX3U, FX3UC	E/A-Konfigurationsfehler
M8061	●	—	FX1S, FX1N, FX2N, FX2NC FX3G, FX3GC FX3GE, FX3S, FX3U, FX3UC	SPS-Hardwarefehler
M8062	●	—	FX1S, FX1N FX2N, FX2NC, FX3S ^④ , FX3U ^④ , FX3UC ^④	Kommunikation zwischen SPS und Program- miergerät gestört
			FX3G ^⑤ , FX3GC ^⑤ , FX3GE ^⑤	Fehler bei serieller Kommunikation (Ch. 0)
M8063 ^①	●	—	FX1S, FX1N FX2N, FX2NC FX3G, FX3GC FX3GE, FX3S, FX3U, FX3UC	Fehler bei serieller Kommunikation (Ch. 1)
M8064	●	—		Parameterfehler
M8065	●	—		Programmsyntax-Fehler
M8066	●	—		Programmierfehler
M8067 ^②	●	—		Ausführungsfehler
M8068	—	●		Ausführungsfehler (gespeichert)
M8069 ^③	—	●	FX2N, FX2NC FX3G, FX3GC FX3GE, FX3U, FX3UC	E/A-Bus-Fehler

Tab. 9-7: Sondermerker zur Fehleranzeige

- ① Bei Steuerungen der FX1S-, FX1N-, FX2N- und FX2NC-Serie wird dieser Merker zurückgesetzt, wenn die SPS von STOP in RUN geschaltet wird. Bei einer SPS der FX3G-, FX3GC-, FX3GE-, FX3S-, FX3U- oder FX3UC-Serie wird M8063 beim Einschalten der Versorgungsspannung zurückgesetzt.
Ein Fehler bei der seriellen Kommunikation über Kanal 2 wird bei einer FX3G, FX3GC, FX3GE, FX3U oder FX3UC durch M8438 angezeigt.
- ② Dieser Merker wird zurückgesetzt, wenn die Steuerung von STOP in RUN geschaltet wird.
- ③ Nach dem Setzen von M8069 wird eine Kontrolle des E/A-Busses ausgeführt. Wenn hierbei ein Fehler erkannt wird, wird der Fehlercode 6130 in das Sonderregister D8069 geschrieben und der Sondermerker M8061 gesetzt.
- ④ Wird nur gesetzt, wenn in einem FX3S-, FX3U- oder FX3UC-Grundgerät ein Fehler beim Zugriff auf den Speicher aufgetreten ist (Fehlercode 6230).
- ⑤ Dieser Merker wird beim Einschalten der Versorgungsspannung der SPS zurückgesetzt.
- ① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontaktanweisung (z.B. LD- oder LDI-Anweisung) abgefragt werden kann.
- ② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

9.1.8 Pulse-Catch-Funktion (M8170–M8177)

Wird ein Impuls-Signal an einem der Eingänge X0 bis X5 (X0 bis X7 bei FX3U und FX3UC) erkannt, wird einer der unten aufgeführten Merker gesetzt. Nach dem Zurücksetzen wird der Sondermerker automatisch wieder mit dem nächsten Impuls-Signal gesetzt. Auf diese Weise können schnelle Eingangsimpulse erkannt und gespeichert werden.

Bei Steuerungen der FX2N-, FX2NC-, FX3U- und FX3UC-Serie müssen zur Nutzung der Pulse-Catch-Funktion Interrupts durch eine EI-Anweisung freigegeben werden. Bei Steuerungen der FX1S-, FX1N-, FX3G-, FX3GC-, FX3GE- und FX3S-Serie ist dies nicht erforderlich.

Die Sondermerker M8170 bis M8177 werden auch beim Übergang der SPS vom STOP- in den RUN-Modus zurückgesetzt.

Sondermerker Operanden- adresse	① Signal- zustand abfragen	② Signal- zustand festlegen	CPU	Bedeutung
M8170	●	—	FX1S	Impuls-Catch X0
M8171	●	—	FX1N	Impuls-Catch X1
M8172	●	—	FX2N	Impuls-Catch X2
M8173	●	—	FX2NC	Impuls-Catch X2
M8174	●	—	FX3G	Impuls-Catch X3
M8175	●	—	FX3GC	Impuls-Catch X3
M8176	●	—	FX3GE	Impuls-Catch X4
M8177	●	—	FX3S	Impuls-Catch X4
M8175	●	—	FX3U	Impuls-Catch X5
M8176	●	—	FX3UC	Impuls-Catch X5
M8176	●	—	FX3U	Impuls-Catch X6
M8177	●	—	FX3UC	Impuls-Catch X7

Tab. 9-8: Sondermerker für Pulse-Catch-Funktion

- ① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontaktanweisung (z. B. LD- oder LDI-Anweisung) abgefragt werden kann.
- ② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

9.1.9 Link- und Sonderfunktionen (M8070–M8199)

Sondermerker Operanden- adresse	① Signal- zustand abfragen	② Signal- zustand festlegen	CPU	Bedeutung
M8070	●	●	FX1S FX1N FX2N FX2NC FX3G FX3GC FX3GE FX3S FX3U FX3UC	Der Merker wird gesetzt, wenn es sich bei der SPS um eine Master-Station in einer parallelen Link-Verbindung handelt.
M8071	●	●		Der Merker wird gesetzt, wenn es sich bei der SPS um eine Slave-Station in einer parallelen Link-Verbindung handelt.
M8072	●	—		Kennung für eine parallele Link-Verbindung der SPS
M8073	●	—		Kennung, dass M8070 bzw. M8071 in einer parallelen Link-Verbindung falsch gesetzt sind.
M8074	Reserviert			
M8075	—	—	FX2N, FX2NC FX3U, FX3UC	Bereitschaft für Sampling Trace anfordern
	●	—	FX3G (ab V1.10) FX3GC, FX3GE	Sondermerker für Impulsweiten-/Periodendauerermessung
M8076	—	—	FX2N, FX2NC FX3U, FX3UC	Start des Sampling Trace anfordern
	●	●	FX3G (ab V1.10) FX3GC, FX3GE	Impulsweiten-/Periodendauerermessung an Eingang X000 starten
M8077	●	—	FX2N, FX2NC FX3U, FX3UC	Wird während des Sampling Trace gesetzt
	●	●	FX3G (ab V1.10) FX3GC, FX3GE	Impulsweiten-/Periodendauerermessung an Eingang X001 starten
M8078	●	—	FX2N, FX2NC FX3U, FX3UC	Wird gesetzt, wenn das Sampling Trace abgeschlossen ist.
	●	●	FX3G (ab V1.10) FX3GC, FX3GE	Impulsweiten-/Periodendauerermessung an Eingang X003 starten
M8079	—	—	FX2N, FX2NC FX3U, FX3UC	Sondermerker für Sampling Trace
	●	●	FX3G (ab V1.10) FX3GC, FX3GE	Impulsweiten-/Periodendauerermessung an Eingang X004 starten
M8080	●	●	FX3G (ab V1.10) FX3GC, FX3GE	Periodendauer des Signals an X000 messen
M8081	●	●		Periodendauer des Signals an X001 messen
M8082	●	●		Periodendauer des Signals an X003 messen
M8083	●	●		Periodendauer des Signals an X004 messen
M8084 bis M8089	—	—	—	Reserviert
M8090	●	—	FX3U, FX3UC (ab V2.20)	Bei einer BKCMPI-Anweisung (FNC194 bis FNC199) ergeben alle Vergleiche das Ergebnis „Wahr“.
M8091	●	●		Inhalt des letzten Zeichens bei einer COMRD- oder BINDA-Anweisung
M8092 bis M8098	—	—	—	Reserviert
M8099	●	●	FX2N, FX2NC FX3U, FX3UC	Ringzähler aktivieren*

Tab. 9-9: Sondermerker für Link- und Sonderfunktionen (1)

* Bei einer FX2N oder FX2NC wird der Ringzähler (Sonderregister D8099) am Ende des Programmzyklus aktiviert, in dem M8099 gesetzt wurde. Bei einer FX3U oder FX3UC wird der Ringzähler unmittelbar nach dem Setzen von M8099 aktiviert.

Sondermerker Operanden- adresse	① Signal- zustand abfragen	② Signal- zustand festlegen	CPU	Bedeutung
M8100 bis M8103	—	—	—	Reserviert
M8104	●	—	FX2N/FX2NC	Speichererweiterung ist installiert (ab Version 3)
M8105	●	—	FX3G, FX3GC FX3GE, FX3S FX3U, FX3UC	Es wird in die Speicherkassette geschrieben.
M8106	—	—	—	Reserviert
M8107	●	—	FX3U, FX3UC	Eintrag von Operandenkommentar wird geprüft.
M8108	—	—	—	Reserviert
M8109	●	—	FX2N, FX2NC FX3G, FX3GC FX3GE, FX3U, FX3UC	Fehler bei der Aktualisierung der Ausgänge
M8110	—	—	—	Reserviert
M8111	—	—	—	Reserviert
M8112	●	●	FX1S/FX1N	Erweiterungsadapter FX1N-4EX-BD: Eingang BX0
				Erweiterungsadapter FX1N-2AD-BD: Eingangsmodus für Kanal 1 ändern
				Erweiterungsadapter FX1N-1DA-BD: Ausgangsmodus ändern
M8113	●	●		Erweiterungsadapter FX1N-4EX-BD: Eingang BX1
				Erweiterungsadapter FX1N-2AD-BD: Eingangsmodus für Kanal 2 ändern
				Erweiterungsadapter FX1N-4EX-BD: Eingang BX2
M8114	●	●		Erweiterungsadapter FX1N-4EX-BD: Eingang BX3
M8115	●	●		Erweiterungsadapter FX1N-4EX-BD: Eingang BX3
M8116	●	●		Erweiterungsadapter FX1N-2EYT-BD: Ausgang BY0
M8117	●	●	Erweiterungsadapter FX1N-2EYT-BD: Ausgang BY1	
M8118	—	—	—	Reserviert
M8119				
M8120				
M8121*	●	—	FX1S, FX1N FX2N, FX2NC FX3G, FX3GC FX3GE, FX3S, FX3U, FX3UC	RS-Datenübertragung verzögert
M8122*	●	●		RS-Datenübertragung-Flag
M8123*	●	●		RS-Datenempfang beendet
M8124	●	—		RS-Carrier-Flag-Erkennung
M8125	—	—	—	Reserviert
M8126	●	—	FX1S, FX1N FX2N, FX2NC FX3G, FX3GC FX3GE, FX3S, FX3U, FX3UC	Computer-Link: Global-Funktion
M8127	●	—	FX2N, FX2NC FX3G, FX3GC FX3GE, FX3S, FX3U, FX3UC	Computer-Link: Handshake-Signal der On-Demand-Funktion

Tab. 9-10: Sondermerker für Link- und Sonderfunktionen (2)

* Diese Merker werden zurückgesetzt, wenn die Steuerung von RUN in STOP geschaltet oder die RS-Anweisung nicht bearbeitet wird.

Sondermerker Operanden- adresse	① Signal- zustand abfragen	② Signal- zustand festlegen	CPU	Bedeutung
M8128	●	●	FX1S, FX1N FX2N, FX2NC FX3G, FX3GC FX3GE, FX3S, FX3U, FX3UC	Computer-Link: Fehler bei der On-Demand-Funktion
M8129	●	●		Computer-Link: Sendedatenformat der On-Demand-Funktion (Wort oder Byte) „1“: Übertragung in Byte-Einheiten (8 Bits) „0“: Übertragung in Worteinheiten (16 Bits) RS-Anweisung: Zeitüberlauf
M8130	●	●	FX2N, FX2NC FX3U, FX3UC	Auswahl der Vergleichstabellen, die mit der HSZ-Anweisung verwendet werden
M8131	●	—		Kennung zur Beendigung der Verarbeitung des HSZ-Vergleichs
M8132	●	●		Festlegung der Bedeutung der PLSY-Anweisung mit den HSZ-Vergleichstabellen
M8133	●	—		Kennung für das Verarbeitungsende für den HSZ-Vergleich (bei Verwendung der PLSY-Anweisung)
M8134 bis M8137	—	—	—	Reserviert
M8138	●	—	FX3U, FX3UC	Bearbeitung der DHSCT-Anweisung beendet
M8139	●	—		Eine Anweisung für High-Speed-Counter (DHSCS, DHSCR, DHSZ, DHSCT) wird ausgeführt.
M8140	●	—	FX1S, FX1N	Impulsausgang der ZRN-Anweisung löschen
M8141 bis M8144	—	—	—	Reserviert
M8145	—	●	FX1S FX1N FX3G FX3GC FX3GE FX3S	Impulsausgabe an Y0 sofort stoppen
M8146	—	●		Impulsausgabe an Y1 sofort stoppen
M8147	●	—		Überwachung der Impulsausgabe an Y0 (Busy/Ready)
M8148	●	—		Überwachung der Impulsausgabe an Y1 (Busy/Ready)
M8149 und M8150	—	—	—	Reserviert
M8151*	●	—	FX3G (ab V 1.10), FX3GC, FX3GE, FX3S, FX3U, FX3UC	Datenaustausch mit Frequenzumrichter über Kanal 1
M8152*	●	—		Fehler bei der Kommunikation mit Frequenzumrichter (Kanal 1)
M8153*	●	—		Fehler bei der Kommunikation mit Frequenzumrichter über Kanal 1 (Gespeicherter Fehler)
M8154*	●	—	FX3U, FX3UC	Fehler bei Ausführung einer IVBWR-Anweisung (Kanal 1)
		—	FX2N/FX2NC (ab V 3.00)	Wartezeit für Kommunikation (EXTR-Anweisung)
M8155	●	—	FX2N/FX2NC (ab V 3.00)	Kommunikation aktiv (EXTR-Anweisung)
M8156*	●	—	FX3G (ab V 1.10), FX3GC, FX3GE, FX3U, FX3UC	Datenaustausch mit Frequenzumrichter über Kanal 2
		—	FX2N/FX2NC (ab V 3.00)	Kommunikations- oder Parameterfehler (EXTR-Anweisung)

Tab. 9-11: Sondermerker für Link- und Sonderfunktionen (3)

* Diese Merker werden zurückgesetzt, wenn die Steuerung von STOP in RUN geschaltet wird.

Sondermerker Operanden- adresse	① Signal- zustand abfragen	② Signal- zustand festlegen	CPU	Bedeutung
M8157	●	—	FX3G (ab V 1.10), FX3GC, FX3GE, FX3U, FX3UC	Fehler bei der Kommunikation mit Frequenzum- richter (Kanal 2)
		—	FX2N/FX2NC (ab V 3.00)	Gespeicherter Kommunikationsfehler (EXTR-Anweisung)
M8158	●	—	FX3G (ab V 1.10), FX3GC, FX3GE, FX3U, FX3UC	Fehler bei der Kommunikation mit Frequenzum- richter über Kanal 2 (Gespeicherter Fehler)
M8159	●	—	FX3U, FX3UC	Fehler bei Ausführung einer IVBWR-Anweisung (Kanal 2)
M8160*	●	●	FX2N, FX2NC FX3U, FX3UC	XCH-Anweisung als Byte-Tauschfunktion
M8161*	●	●	FX1S, FX1N FX2N, FX2NC FX3G, FX3GC	8-Bit Modus (RS, ASC, ASCI, HEX, CCD, CRC) Eine CRC-Anweisung kann nur von einer FX3U oder FX3UC ausgeführt werden.
M8162	●	●	FX3GE, FX3S, FX3U, FX3UC	High-Speed-Modus bei Parallel-Link (32-Bit für jede Datenrichtung)
M8163	—	—	—	Reserviert
M8164*	●	●	FX2N/FX2NC (ab V 2.00)	Bei gesetztem Merker wird der Wert in D8164 als Anzahl der mit der FROM/TO-Anweisung zu übertragenden Operanden interpretiert.
M8165*	●	●	FX3U, FX3UC (ab V 2.20)	Bei SORT2-Anweisung in absteigender Rei- henfolge sortieren.
M8166	—	—	—	Reserviert
M8167*	●	●	FX2N, FX2NC FX3U, FX3UC	Hexadezimal-Format für die HKY-Anweisung
M8168*	●	●	FX2N, FX2NC FX3G, FX3GC FX3GE, FX3S, FX3U, FX3UC	Hexadezimal-Format für die SMOV-Anweisung
M8169	—	—	—	Reserviert
M8170 bis M8175	●	●	FX1S, FX1N FX2N, FX2NC FX3G, FX3GC FX3GE, FX3S, FX3U, FX3UC	Impuls-Catch-Merker für die Eingänge X0 bis X5 (siehe Abschnitt 9.1.8)
M8176	●	●	FX3U, FX3UC	Impuls-Catch-Flags für die Eingänge X6 und X7 (siehe Abschnitt 9.1.8)
M8177				
M8178	●	●	FX3G, FX3GC FX3GE, FX3U FX3UC	Kanalwahl bei Parallel-Link (M8178 = 0: Kanal 1, M8178 = 1: Kanal 2)
M8179	●	●		Kanalwahl bei n:n-Netzwerk (M8179 = 0: Kanal 1, M8179 = 1: Kanal 2)
M8180 bis M8182	—	—	—	Reserviert

Tab. 9-12: Sondermerker für Link- und Sonderfunktionen (4)

* Diese Merker werden zurückgesetzt, wenn die Steuerung von RUN in STOP geschaltet wird.

Sondermerker Operanden- adresse	① Signal- zustand abfragen	② Signal- zustand festlegen	CPU	Bedeutung
M8183 M504 bei FX1S	●	—	FX1S FX1N FX2N (ab V2.00) FX3G FX3GC FX3GE FX3S FX3U FX3UC	Kommunikationsfehler in der Master-Station eines n:n-Netzwerks
M8184 M505 bei FX1S	●	—		Wird gesetzt, wenn in der 1. Slave-Station eines n:n-Netzwerks ein Kommunikationsfehler aufgetreten ist.
M8185 M506 bei FX1S	●	—		Wird gesetzt, wenn in der 2. Slave-Station eines n:n-Netzwerks ein Kommunikationsfehler aufgetreten ist.
M8186 M507 bei FX1S	●	—		Wird gesetzt, wenn in der 3. Slave-Station eines n:n-Netzwerks ein Kommunikationsfehler aufgetreten ist.
M8187 M508 bei FX1S	●	—		Wird gesetzt, wenn in der 4. Slave-Station eines n:n-Netzwerks ein Kommunikationsfehler aufgetreten ist.
M8188 M509 bei FX1S	●	—		Wird gesetzt, wenn in der 5. Slave-Station eines n:n-Netzwerks ein Kommunikationsfehler aufgetreten ist.
M8189 M510 bei FX1S	●	—		Wird gesetzt, wenn in der 6. Slave-Station eines n:n-Netzwerks ein Kommunikationsfehler aufgetreten ist.
M8190 M511 bei FX1S	●	—		Wird gesetzt, wenn in der 7. Slave-Station eines n:n-Netzwerks ein Kommunikationsfehler aufgetreten ist.
M8191 M503 bei FX1S	●	—		Wird gesetzt, wenn in einem n:n-Netzwerk mit einer anderen Station kommuniziert wird.
M8192 bis M8197	—	—		—
M8198*	●	●	FX3U, FX3UC	C251, C252, C254: Jede Flanke der Phasen A und B zählen
M8199*	●	●		C253, C255: Jede Flanke der Phasen A und B zählen

Tab. 9-13: Sondermerker für Link- und Sonderfunktionen (5)

* Diese Merker werden zurückgesetzt, wenn die Steuerung von RUN in STOP geschaltet wird.

- ① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontaktanweisung (z. B. LD- oder LDI-Anweisung) abgefragt werden kann.
- ② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

HINWEISE

Die Hinweise zu den Sondermerkern M8060 bis M8069 in Kapitel 11 (Programmfehler) sind ebenfalls zu beachten.

Bei den Steuerungen der FX1S-Serie sind die Merker M504 bis M511 für ein n:n-Netzwerk reserviert. Diese Merker können nicht für andere Funktionen verwendet werden.

9.1.10 Auf-/Abwärts-Counter (M8200–M8254)

Sondermerker Operanden- adresse	① Signal- zustand abfragen	② Signal zustand festlegen	CPU	Bedeutung
M8200	—	●	FX1S FX1N FX2N FX2NC FX3G FX3GC FX3GE FX3S FX3U FX3UC	Wird einer dieser Sondermerker gesetzt, wird der zugehörige Counter als abwärtszählender Counter definiert. (M8200 = C200, M8201 = C201 usw.). Die Counter sind aufwärtszählend, solange der zugehörige Sondermerker nicht gesetzt ist.
M8201	—	●		
M8203	—	●		
M8204	—	●		
M8205	—	●		
M8206	—	●		
M8207	—	●		
M8208	—	●		
M8209	—	●		
M8210	—	●		
M8211	—	●		
M8212	—	●		
M8213	—	●		
M8214	—	●		
M8215	—	●		
M8216	—	●		
M8217	—	●		
M8218	—	●		
M8219	—	●		
M8220	—	●		
M8221	—	●		
M8222	—	●		
M8223	—	●		
M8224	—	●		
M8225	—	●		
M8226	—	●		
M8227	—	●		
M8228	—	●		
M8229	—	●		
M8230	—	●		
M8231	—	●		
M8232	—	●		
M8233	—	●		
M8234	—	●		

Tab. 9-14: Sondermerker für Auf-/Abwärts-Counter

- ① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontaktanweisung (z. B. LD- oder LDI-Anweisung) abgefragt werden kann.
- ② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

1-Phasen-Counter mit einem Zählengang (M8235–M8245)

Sondermerker Operanden-adresse	① Signal-zustand abfragen	② Signal-zustand festlegen	CPU	Bedeutung
M8235	●	●	FX1S FX1N FX2N FX2NC FX3G FX3GC FX3GE FX3S FX3U FX3UC	Ist M8235 gesetzt, zählt C235 abwärts.
M8236	●	●		Ist M8236 gesetzt, zählt C236 abwärts.
M8237	●	●		Ist M8237 gesetzt, zählt C237 abwärts.
M8238	●	●		Ist M8238 gesetzt, zählt C238 abwärts.
M8239	●	●		Ist M8239 gesetzt, zählt C239 abwärts.
M8240	●	●		Ist M8240 gesetzt, zählt C240 abwärts.
M8241	●	●		Ist M8241 gesetzt, zählt C241 abwärts.
M8242	●	●		Ist M8242 gesetzt, zählt C242 abwärts.
M8243	●	●		Ist M8243 gesetzt, zählt C243 abwärts.
M8244	●	●		Ist M8244 gesetzt, zählt C244 abwärts.
M8245	●	●	Ist M8245 gesetzt, zählt C245 abwärts.	

Tab. 9-15: Sondermerker für 1-Phasen-Counter mit einem Zählengang**1-Phasen-Counter mit zwei Zählengängen (M8246 – M8250)**

Sondermerker Operanden-adresse	① Signal-zustand abfragen	② Signal-zustand festlegen	CPU	Bedeutung
M8246	●	—	FX1S FX1N FX2N FX2NC FX3G FX3GC FX3GE FX3S FX3U FX3UC	M8246 wird gesetzt, wenn C246 abwärts zählt. M8246 wird zurückgesetzt, wenn C246 aufwärts zählt.
M8247	●	—		M8247 wird gesetzt, wenn C247 abwärts zählt. M8247 wird zurückgesetzt, wenn C247 aufwärts zählt.
M8248	●	—		M8248 wird gesetzt, wenn C248 abwärts zählt. M8248 wird zurückgesetzt, wenn C248 aufwärts zählt.
M8249	●	—		M8249 wird gesetzt, wenn C249 abwärts zählt. M8249 wird zurückgesetzt, wenn C249 aufwärts zählt.
M8250	●	—		M8250 wird gesetzt, wenn C250 abwärts zählt. M8250 wird zurückgesetzt, wenn C250 aufwärts zählt.

Tab. 9-16: Sondermerker für 2-Phasen-Counter mit zwei Zählengängen

- ① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontaktanweisung (z. B. LD- oder LDI-Anweisung) abgefragt werden kann.
- ② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

A/B-Phasen-Counter mit zwei Zähleringängen (M8251–M8255)

Sondermerker Operanden- adresse	① Signal- zustand abfragen	② Signal- zustand festlegen	CPU	Bedeutung
M8251	●	—	FX1S FX1N FX2N FX2NC FX3G FX3GC FX3GE FX3S FX3U FX3UC	M8251 wird gesetzt, wenn C251 abwärts zählt. M8251 wird zurückgesetzt, wenn C251 aufwärts zählt.
M8252	●	—		M8252 wird gesetzt, wenn C252 abwärts zählt. M8252 wird zurückgesetzt, wenn C252 aufwärts zählt.
M8253	●	—		M8253 wird gesetzt, wenn C253 abwärts zählt. M8253 wird zurückgesetzt, wenn C253 aufwärts zählt.
M8254	●	—		M8254 wird gesetzt, wenn C254 abwärts zählt. M8254 wird zurückgesetzt, wenn C254 aufwärts zählt.
M8255	●	—		M8255 wird gesetzt, wenn C255 abwärts zählt. M8255 wird zurückgesetzt, wenn C255 aufwärts zählt.

Tab. 9-17: Sondermerker für A/B-Phasen-Counter mit zwei Zähleringängen

- ① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontakthanweisung (z. B. LD- oder LDI-Anweisung) abgefragt werden kann.
- ② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

9.1.11 Analoge Adaptermodule und Erweiterungsadapter (M8260–M8299)

Sondermerker Operanden- adresse	① Signal- zustand abfragen	② Signal- zustand festlegen	CPU	Bedeutung
M8260 bis M8269	●	—	FX3U, FX3UC (ab V2.00)	Sondermerker für das 1. analoge Adaptermodul
	●	—	FX3G (ab V1.10) FX3GE, FX3S	Sondermerker für den 1. analogen Erweiterungsadapter ^①
M8270 bis M8279	●	—	FX3U, FX3UC (ab V2.00)	Sondermerker für das 2. analoge Adaptermodul
	●	—	FX3G (ab V1.10)	Sondermerker für den 2. analogen Erweiterungsadapter ^②
M8280 bis M8289	●	—	FX3U, FX3UC (ab V2.00)	Sondermerker für das 3. analoge Adaptermodul
	●	—	FX3G, FX3GC, FX3GE, FX3S	Sondermerker für das 1. analoge Adaptermodul
M8290 bis M8299	●	—	FX3U, FX3UC (ab V2.00)	Sondermerker für das 4. analoge Adaptermodul
	●	—	FX3G ^③ , FX3GC, FX3GE ^④	Sondermerker für das 2. analoge Adaptermodul

Tab. 9-18: Sondermerker für angeschlossene Analogmodule

- ① Installiert im Erweiterungssteckplatz der Grundgeräte FX3G-14M□/□ oder FX3G-24M□/□, auf dem linken Erweiterungssteckplatz (Position 1) der Grundgeräte FX3G-40□/□ oder FX3G-60M□/□ oder in einem FX3GE- oder FX3S-Grundgerät.
- ② Installiert auf dem rechten Erweiterungssteckplatz (Position 2) der Grundgeräte FX3G-40□/□ oder FX3G-60M□/□.
- ③ Ein zweites analoges Adaptermodul kann nur bei den Grundgeräten FX3G-40□/□ und FX3G-60M□/□ installiert werden.
- ④ Bei den Grundgeräten der FX3GE-Serie gelten die integrierten analogen Eingänge und der integrierte analoge Ausgang als 1. analoges Adaptermodul. Wird ein analoges Adaptermodul angeschlossen, gilt dieses Adaptermodul als 2. analoges Adaptermodul.
- ① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontaktanweisung (z. B. LD- oder LDI-Anweisung) abgefragt werden kann.
- ② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

HINWEIS

Die Funktion der Sondermerker M8260 bis M8299 ist in den Handbüchern zu den Analogmodulen beschrieben.

9.1.12 Flags (M8300–M8311)

Sondermerker Operanden- adresse	① Signal- zustand abfragen	② Signal- zustand festlegen	CPU	Bedeutung
M8300 bis M8303	—	—	—	Reserviert
M8304	●	—	FX3U (ab V2.30) FX3UC (ab V2.30) FX3G, FX3GC, FX3GE, FX3S	Null Dieser Merker wird gesetzt, wenn das Ergebnis einer Multiplikation oder Division gleich Null ist.
M8305	●	—	—	Reserviert
M8306	●	—	FX3U (ab V2.30) FX3UC (ab V2.30) FX3G, FX3GC, FX3GE, FX3S	Übertrag Dieser Merker wird gesetzt, wenn durch das Ergebnis einer Division ein Zahlen- wertübertrag entsteht.
M8307 bis M8311	—	—	—	Reserviert

Tab. 9-19: Sondermerker zur Anzeige bei arithmetischen Operationen

- ① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontaktanweisung (z. B. LD- oder LDI-Anweisung) abgefragt werden kann.
- ② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

9.1.13 Sondermerker zur Fehlerdiagnose (M8312–M8329)

Sondermerker Operanden-adresse	① Signal-zustand abfragen	② Signal-zustand festlegen	CPU	Bedeutung
M8312	●	—	FX3G FX3GC FX3GE FX3S	Datenverlust bei der integrierten Uhr Der Zustand von M8312 bleibt auch bei einem Ausfall der Versorgungsspannung erhalten. M8312 wird beim Stellen der Uhr automatisch zurückgesetzt.
M8313 bis M8315	—	—	—	Reserviert
M8316	●	—	FX3U FX3UC	Es wurde eine nicht existierende E/A-Adresse angesprochen. M8316 wird gesetzt, wenn die betreffende E/A-Adresse direkt in z. B. einer LD-, AND-, OR- oder OUT-Anweisung verwendet oder wenn der nicht existierende Ein- oder Ausgang indirekt über Index-Register angesprochen wird.
M8317	—	—	—	Reserviert
M8318	●	—	FX3U FX3UC (ab V2.20)	Fehler bei der Initialisierung eines Pufferspeichers M8318 wird gesetzt, wenn beim Zugriff auf den Pufferspeicher eines Sondermoduls mit einer FROM- oder TO-Anweisung ein Fehler aufgetreten ist. In diesem Fall enthält D8318 die Nummer des Sondermoduls und D8319 die Pufferspeicheradresse.
M8319 bis M8321	—	—	—	Reserviert
M8322 bis M8324	●	—	FX3UC-32MT-LT2	Sondermerker zur Angabe des SPS-Typs und dem Zustand der Konfiguration von CC-Link/LT
M8325 bis M8327	—	—	—	Reserviert
M8328	●	—	FX3U FX3UC (ab V2.30)	Eine Anweisung kann nicht ausgeführt werden.
M8329	●	—	FX3G FX3GC FX3GE FX3S FX3U FX3UC	Bei der Ausführung einer Anweisung ist ein Fehler aufgetreten.

Tab. 9-20: Sondermerker zur Fehlerdiagnose bei einer FX3G/FX3GC/FX3GE/FX3S/FX3U oder FX3UC

- ① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontaktanweisung (z. B. LD- oder LDI-Anweisung) abgefragt werden kann.
- ② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

9.1.14 Sondermerker für Impulsausgabe und Positionierung (M8330–M8379)

Sondermerker Operanden- adresse	① Signal- zustand abfragen	② Signal- zustand festlegen	CPU	Bedeutung	
M8330	●	—	FX3U, FX3UC (ab V2.20)	DUTY-Anweisung	Ausgangssignal 1
M8331	●	—			Ausgangssignal 2
M8332	●	—			Ausgangssignal 3
M8333	●	—			Ausgangssignal 4
M8334	●	—			Ausgangssignal 5
M8335	—	—	—	Reserviert	
M8336*	●	●	FX3U, FX3UC (ab V1.30)	DVIT-Anweisung	Freigabe der Interrupts
M8337	—	—	—	Reserviert	
M8338	●	●	FX3G, FX3GC, FX3GE, FX3S FX3U, FX3UC (ab V2.20)	PLSV-Anweisung	Beschleunigung/ Verzögerung
M8339	—	—	—	Reserviert	
M8340	●	—	FX3G, FX3GC, FX3GE, FX3S FX3U, FX3UC	Ausgang Y000	Impulsüberwachung (0: Bereit, 1: Aktiv)
M8341*	●	●			Rücksetzen des Ausgangs freigeben
M8342*	●	●			Drehrichtung für Referenz- punktfahrt
M8343	●	●			Begrenzung bei Vorwärts- bewegung
M8344	●	●			Begrenzung bei Rückwärts- bewegung
M8345*	●	●			Signalform des Referenz- punktannäherungsschalters (Öffner/Schließer)
M8346*	●	●			Signalform des Nullpunkt- signals (Öffner/Schließer)
M8347*	●	●	FX3U, FX3UC	Signalform des Interrupt- Signals (Öffner/Schließer)	
M8348	●	—	FX3G, FX3U, FX3UC	Positionieranweisung aktiv	
M8349*	●	●	—	—	Impulsausgabe stoppen
M8350	●	—	FX3G, FX3GC, FX3GE, FX3S FX3U, FX3UC	Ausgang Y001	Impulsüberwachung (0: Bereit, 1: Aktiv)
M8351*	●	●			Rücksetzen des Ausgangs freigeben
M8352*	●	●			Drehrichtung für Referenz- punktfahrt
M8353	●	●			Begrenzung bei Vorwärts- bewegung
M8354	●	●			Begrenzung bei Rückwärts- bewegung
M8355*	●	●			Signalform des Referenz- punktannäherungsschalters (Öffner/Schließer)

Tab. 9-21: Sondermerker für Signalausgabe und Positionierung (1)

* Diese Merker werden bei einem STOP der SPS zurückgesetzt.

- ① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontaktanweisung (z. B. LD- oder LDI-Anweisung) abgefragt werden kann.
- ② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

Sondermerker Operanden- adresse	① Signal- zustand abfragen	② Signal- zustand festlegen	CPU	Bedeutung		
M8356*	●	●	FX3G, FX3GC, FX3GE, FX3S FX3U, FX3UC	Ausgang Y001	Signalform des Nullpunkt- signals (Öffner/Schließer)	
M8357*	●	●	FX3U, FX3UC		Signalform des Interrupt- Signals (Öffner/Schließer)	
M8358	●	—	FX3G, FX3GC, FX3GE, FX3S		Positionieranweisung aktiv	
M8359*	●	●	FX3U, FX3UC		Impulsausgabe stoppen	
M8360	●	—	FX3G, FX3GC, FX3GE, FX3U, FX3UC	Ausgang Y002	Impulsüberwachung (0: Bereit, 1: Aktiv)	
M8361*	●	●			Rücksetzen des Ausgangs freigeben	
M8362*	●	●			Drehrichtung für Referenz- punktfahrt	
M8363	●	●			Begrenzung bei Vorwärts- bewegung	
M8364	●	●			Begrenzung bei Rückwärts- bewegung	
M8365*	●	●			Signalform des Referenz- punktannäherungsschalters (Öffner/Schließer)	
M8366*	●	●			Signalform des Nullpunkt- signals (Öffner/Schließer)	
M8367*	●	●			FX3U, FX3UC	Signalform des Interrupt- Signals (Öffner/Schließer)
M8368	●	—			FX3G, FX3GC, FX3GE, FX3U, FX3UC	Positionieranweisung aktiv
M8369*	●	●			FX3U, FX3UC	Impulsausgabe stoppen
M8370	●	—	FX3U	Ausgang Y003 (Dieser Ausgang steht nur zur Ver- fügung, wenn zwei Adapter FX3U-2HSY-ADP angeschlossen sind.)	Impulsüberwachung (0: Bereit, 1: Aktiv)	
M8371*	●	●			Rücksetzen des Ausgangs freigeben	
M8372*	●	●			Drehrichtung für Referenz- punktfahrt	
M8373	●	●			Begrenzung bei Vorwärts- bewegung	
M8374	●	●			Begrenzung bei Rückwärts- bewegung	
M8375*	●	●			Signalform des Referenz- punktannäherungsschalters (Öffner/Schließer)	
M8376*	●	●			Signalform des Nullpunkt- signals (Öffner/Schließer)	
M8377*	●	●			Signalform des Interrupt- Signals (Öffner/Schließer)	
M8378	●	—			Positionieranweisung aktiv	
M8379*	●	●			Impulsausgabe stoppen	

Tab. 9-22: Sondermerker für Signalausgabe und Positionierung (2)

* Diese Merker werden bei einem STOP der SPS zurückgesetzt.

- ① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontaktanweisung (z. B. LD- oder LDI-Anweisung) abgefragt werden kann.
- ② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

9.1.15 Sondermerker für Kommunikation (M8370–M8379)

Sondermerker Operanden- adresse	① Signal- zustand abfragen	② Signal- zustand festlegen	CPU	Bedeutung	
M8370	—	—	—	Reserviert	
M8371*	●	—	FX3G, FX3GC, FX3GE	RS2-Anweisung (Kanal 0)	Warten auf Senden
M8372*	●	●			Senden anfordern
M8373*	●	●			Empfang beendet
M8374 bis M8378	—	—	—	Reserviert	
M8379	●	●	FX3G, FX3GC, FX3GE	RS2-Anweisung (Kanal 0)	Überwachungszeit abgelaufen

Tab. 9-23: Sondermerker für die RS2-Anweisung bei einer FX3G, FX3GC oder FX3GE

* Diese Merker werden bei einem STOP der SPS zurückgesetzt.

- ① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontaktanweisung (z. B. LD- oder LDI-Anweisung) abgefragt werden kann.
- ② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

9.1.16 Sondermerker für High-Speed-Counter (M8380–M8392)

Sondermerker Operanden- adresse	① Signal- zustand abfragen	② Signal- zustand festlegen	CPU	Bedeutung	
M8380*	●	—	FX3U, FX3UC	Status von C235, C241, C244, C246, C247, C249, C251, C252 und C254	
M8381*	●	—		Status von C236	
M8382*	●	—		Status von C237, C242 und C245	
M8383*	●	—		Status von C238, C248, C248 (OP), C250, C253 und C255	
M8384*	●	—		Status von C239 und C243	
M8385*	●	—		Status von C240	
M8386*	●	—		Status von C244 (OP)	
M8387*	●	—		Status von C245 (OP)	
M8388	●	—	FX3G, FX3GC, FX3GE, FX3S FX3U, FX3UC	Funktion der High-Speed-Counter ändern	
M8389	●	●	FX3U, FX3UC	Signalform des externen Rücksetzeingangs (Öffner/Schließer)	
M8390	●	●		Funktionsumschaltung für C244	
M8391	●	●		Funktionsumschaltung für C245	
M8392	●	●	FX3G, FX3GC, FX3GE, FX3S FX3U, FX3UC	Funktionsumschaltung für C248 und C253	

Tab. 9-24: Sondermerker für High-Speed-Counter

* Diese Merker werden bei einem STOP der SPS zurückgesetzt.

- ① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontaktanweisung (z. B. LD- oder LDI-Anweisung) abgefragt werden kann.
- ② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

9.1.17 Sondermerker für Interrupt-Programme und High-Speed-Counter (M8393–M8397)

Sondermerker Operanden-adresse	① Signal-zustand abfragen	② Signal-zustand festlegen	CPU	Bedeutung
M8393	●	—	FX3U, FX3UC	Einstellung der Verzögerungszeit
M8394	●	—		Operand zum Aufruf eines Interrupt-Programms durch eine DHCMOV-Anweisung
M8395	●	—	FX3G, FX3GC, FX3GE	Funktionsumschaltung für C254
M8396	—	—	—	Reserviert
M8397				

Tab. 9-25: Sondermerker für Interrupt-Programme und High-Speed-Counter

- ① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontaktanweisung (z. B. LD- oder LDI-Anweisung) abgefragt werden kann.
- ② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

9.1.18 Sondermerker für Ringzähler (M8398)

Sondermerker Operanden-adresse	① Signal-zustand abfragen	② Signal-zustand festlegen	CPU	Bedeutung
M8398	●	●	FX3G, FX3GC, FX3GE, FX3S, FX3U, FX3UC	Aktivierung des Ringzählers (32 Bit, 1 ms) Der Ringzähler (D8399 und D8398) wird nach dem Setzen von M8398 aktiviert.
M8399	—	—	—	Reserviert

Tab. 9-26: Sondermerker für Ringzähler

- ① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontaktanweisung (z. B. LD- oder LDI-Anweisung) abgefragt werden kann.
- ② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

9.1.19 Sondermerker für Kommunikation (M8400–M8459)

HINWEIS

Bitte beachten Sie, dass die einzelnen Merker aus dem Bereich M8400 bis M8459, abhängig von der verwendeten Kommunikationsmethode, unterschiedliche Bedeutungen haben. Eine ausführliche Beschreibung dieser Merker finden Sie im Kommunikationshandbuch der Steuerungen der MELSEC FX-Familie oder in den Bedienungsanleitungen der verwendeten Module.

RS2-Anweisungen und RS485-Netzwerk

Sondermerker Operanden- adresse	① Signal- zustand abfragen	② Signal- zustand festlegen	CPU	Bedeutung	
M8400	—	—	—	Reserviert	
M8401*	●	—	FX3G, FX3GC, FX3S, FX3U, FX3UC	RS2-Anweisung (Kanal 1)	Warten auf Senden
M8402*	●	●			Senden anfordern
M8403*	●	●			Empfang beendet
M8404	●	—			Trägersignal erkannt
M8405	●	—	FX3G, FX3GC, FX3S, FX3U (ab V2.30), FX3UC (ab V2.30)		Betriebsbereitschaft (DSR-Signal)
M8406–M8408	—	—	—	Reserviert	
M8409	●	●	FX3G, FX3GC, FX3S, FX3U, FX3UC	RS2-Anweisung (Kanal 1)	Überwachungszeit abgelaufen
M8410–M8420	—	—	—	Reserviert	
M8421	●	—	FX3G, FX3GC, FX3GE, FX3U, FX3UC	RS2-Anweisung Kanal 2	Warten auf Senden
M8422*	●	●			Senden anfordern
M8423*	●	●			Empfang beendet
M8424	●	—			Trägersignal erkannt
M8425	●	—	FX3G, FX3GC, FX3GE, FX3U (ab V2.30), FX3UC (ab V2.30)		Betriebsbereitschaft (DSR-Signal)
M8426	●	—	FX3G, FX3GC, FX3GE, FX3U, FX3UC	RS485-Netz- werk (Kanal 2)	Globaler Modus
M8427	●	—			Senden von Daten auf Anforderung
M8428	●	●			Fehler beim Senden von Daten auf Anforderung
M8429	●	●		RS2-Anweisung (Kanal 2)	Überwachungszeit abgelau- fen (RS2)
				RS485-Netz- werk (Kanal 2)	Umschaltung zwischen „Byte“ und „Wort“ beim Senden von Daten auf Anforderung

Tab. 9-27: Sondermerker für RS2-Anweisungen und RS485-Netzwerk

* Diese Merker werden zurückgesetzt, wenn die SPS gestoppt oder die RS2-Anweisung nicht mehr ausgeführt wird.

- ① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontaktanweisung (z. B. LD- oder LDI-Anweisung) abgefragt werden kann.
- ② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

MODBUS-Kommunikation

Sondermerker Operanden-adresse	① Signal-zustand abfragen	② Signal-zustand festlegen	CPU	Bedeutung
MODBUS-Kommunikation über Kanal 1 des SPS-Grundgeräts				
M8401	●	—	FX3G ^① , FX3GC, FX3S, FX3U ^② FX3UC ^②	MODBUS-Anforderung wird bearbeitet
M8402	●	—		MODBUS-Kommunikationsfehler
M8403	●	—		MODBUS-Kommunikationsfehler (gespeichert)
M8404	●	—		Offline-Modus
M8405–M8407	—	—	—	Reserviert
M8408	●	—	FX3G ^① , FX3GC, FX3S, FX3U ^② FX3UC ^②	Wiederholversuch
M8409	●	—		Antwortüberwachungszeit überschritten
M8410	—	—	—	Reserviert
MODBUS-Kommunikation (gemeinsam für Kommunikation über Kanal 1 oder Kanal 2)				
M8411	●	●	FX3G ^① , FX3GC, FX3GE, FX3S, FX3U ^② FX3UC ^②	MODBUS konfigurieren
MODBUS-Kommunikation über Kanal 2 des SPS-Grundgeräts				
M8421	●	—	FX3G ^① , FX3GC, FX3GE, FX3S, FX3U ^② FX3UC ^②	MODBUS-Anforderung wird bearbeitet
M8422	●	—		MODBUS-Kommunikationsfehler
M8423	●	—		MODBUS-Kommunikationsfehler (gespeichert)
M8424	●	—		Offline-Modus
M8425–M8427	—	—	—	Reserviert
M8428	●	—	FX3G ^① , FX3GC, FX3GE, FX3S, FX3U ^② FX3UC ^②	Wiederholversuch
M8429	●	—		Antwortüberwachungszeit überschritten
M8430	—	—	—	Reserviert

Tab. 9-28: Sondermerker für MODBUS-Kommunikation

① Ab Version 1.30

② Ab Version 2.40

- ① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontaktanweisung (z. B. LD- oder LDI-Anweisung) abgefragt werden kann.
- ② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

CF-Speicherkartenadaptermodul FX3U-CF-ADP

Sondermerker Operanden- adresse	① Signal- zustand abfragen	② Signal- zustand festlegen	CPU	Bedeutung
Kommunikation mit dem FX3U-CF-ADP über Kanal 1 des SPS-Grundgeräts				
M8400–M8401	—	—	—	Reserviert
M8402	●	—	FX3U ^① FX3UC ^①	Anweisung für FX3U-CF-ADP wird ausgeführt
M8403	—	—	—	Reserviert
M8404	●	—	FX3U ^① FX3UC ^①	FX3U-CF-ADP ist bereit
M8405	●	—	—	CF-Speicherkarte ist installiert
M8406–M8409	—	—	—	Reserviert
M8410	●	●	FX3U ^① FX3UC ^①	Kommunikation mit FX3U-CF-ADP stoppen, um Statusinformationen zu lesen
M8411–M8417	—	—	—	Reserviert
M8418 ^②	●	●	FX3U ^① FX3UC ^①	Fehler bei Ausführung einer Anweisung für das FX3U-CF-ADP
M8419	—	—	—	Reserviert
Kommunikation mit dem FX3U-CF-ADP über Kanal 2 des SPS-Grundgeräts				
M8420–M8421	—	—	—	Reserviert
M8422	●	—	FX3U ^① FX3UC ^①	Anweisung für FX3U-CF-ADP wird ausgeführt
M8423	—	—	—	Reserviert
M8424	●	—	FX3U ^① FX3UC ^①	FX3U-CF-ADP ist bereit
M8425	●	—	—	CF-Speicherkarte ist installiert
M8426–M8429	—	—	—	Reserviert
M8430	●	●	FX3U ^① FX3UC ^①	Kommunikation mit FX3U-CF-ADP stoppen, um Statusinformationen zu lesen
M8431–M8437	—	—	—	Reserviert
M8438 ^②	●	●	FX3U ^① FX3UC ^①	Fehler bei Ausführung einer Anweisung für das FX3U-CF-ADP
M8439	—	—	—	Reserviert

Tab. 9-29: Sondermerker für das CF-Speicherkartenadaptermodul FX3U-CF-ADP

① Ab Version 2.61

② Der Sondermerker wird zurückgesetzt, wenn die SPS von der Betriebsart STOP in die Betriebsart RUN geschaltet wird.

- ① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontaktanweisung (z. B. LD- oder LDI-Anweisung) abgefragt werden kann.
- ② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

Ethernet-Modul FX3U-ENET-ADP

Sondermerker Operanden- adresse	① Signal- zustand abfragen	② Signal- zustand festlegen	CPU	Bedeutung
Kommunikation mit dem FX3U-ENET-ADP über Kanal 1 des SPS-Grundgeräts				
M8400–M8403	—	—	—	Reserviert
M8404	●	—	FX3G ^① FX3GC ^① FX3GE, FX3S, FX3U ^② FX3UC ^②	FX3U-ENET-ADP ist bereit
M8405	—	—	—	Reserviert
M8406	●	—	FX3G ^① FX3GC ^① FX3GE, FX3S, FX3U ^② FX3UC ^②	Einstellung der Zeit wird ausgeführt (Dieser Merker wird nur gesetzt, wenn die SNTP-Funktion in den Parametern freigegeben wurde.)
M8407–M8410	—	—	—	Reserviert
M8411	●	●	FX3G ^① FX3GC ^① FX3GE, FX3S, FX3U ^② FX3UC ^②	Einstellung der Zeit ausführen (Nur gültig, wenn die SNTP-Funktion in den Parametern freigegeben wurde.)
M8412–M8415	—	—	—	Reserviert
Kommunikation mit dem FX3U-ENET-ADP über Kanal 2 des SPS-Grundgeräts				
M8420–M8423	—	—	—	Reserviert
M8424	●	—	FX3G ^① FX3GC ^① FX3S, FX3U ^② FX3UC ^②	FX3U-ENET-ADP ist bereit
M8425	—	—	—	Reserviert
M8426	●	—	FX3G ^① FX3GC ^① FX3S, FX3U ^② FX3UC ^②	Einstellung der Zeit wird ausgeführt (Dieser Merker wird nur gesetzt, wenn die SNTP-Funktion in den Parametern freigegeben wurde.)
M8427–M8430	—	—	—	Reserviert
M8431	●	●	FX3G ^① FX3GC ^① FX3S, FX3U ^② FX3UC ^②	Einstellung der Zeit ausführen (Nur gültig, wenn die SNTP-Funktion in den Parametern freigegeben wurde.)
M8432–M8435	—	—	—	Reserviert
Kommunikation mit dem FX3U-ENET-ADP (gemeinsam für Kommunikation über Kanal 1 oder Kanal 2)				
M8490, M8491	—	—	—	Reserviert
M8492	●	●	FX3G ^③ FX3GC ^③ FX3GE FX3S	Schreibanforderung in Speicher für IP-Adresse
M8493	●	—		Schreiben in Speicher für IP-Adresse beendet
M8494	●	—		Fehler beim Schreiben in IP-Adressenspeicher
M8495	●	●		Löschen des IP-Adressenspeichers anfordern
M8496	●	—		Löschen des IP-Adressenspeichers beendet
M8497	●	—		Fehler beim Löschen des IP-Adressenspeichers
M8498	●	—		Änderung der IP-Adresse freigegeben

Tab. 9-30: Sondermerker für das Ethernet-Adaptermodul FX3U-ENET-ADP

① Ab Version 2.00

② Ab Version 3.10

③ Ab Version 2.10

- ① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontaktanweisung (z. B. LD- oder LDI-Anweisung) abgefragt werden kann.
- ② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

Kommunikationsfehler

Sondermerker Operanden-adresse	① Signal-zustand abfragen	② Signal-zustand festlegen	CPU	Bedeutung
M8430–M8437	—	—	—	Reserviert
M8438	●	●	FX3G, FX3GC, FX3GE, FX3U, FX3UC	Fehler bei serieller Kommunikation über Kanal 2 Dieser Merker wird beim Einschalten der Versorgungsspannung der SPS zurückgesetzt.
M8439–M8448	—	—	—	Reserviert
M8449	●	—	FX3G, FX3GC, FX3GE, FX3U, FX3UC*	Sondermodulfehler
M8450–M8459	—	—	—	Reserviert

Tab. 9-31: Sondermerker für Kommunikationsfehler

* Ab Version 2.20

9.1.20**Sondermerker für Positionierung (M8460–M8467)**

Sondermerker Operanden-adresse	① Signal-zustand abfragen	② Signal-zustand festlegen	CPU	Bedeutung	
M8460	●	●	FX3U, FX3UC (ab V2.20)	Ausgang Y000	
M8461	●	●		Interrupt ^① für DVIT-Anweisung	Ausgang Y001
M8462	●	●			Ausgang Y002
M8463	●	●	FX3U ^② FX3UC (ab V2.20)	Ausgang Y003	
M8464	●	●	FX3G, FX3GC, FX3GE, FX3S, FX3U, FX3UC (ab V2.20)	Ausgang Y000 (D8464)	
M8465	●	●		Rücksetzen eines Operanden durch DSZR- oder ZRN-Anweisung ^③	Ausgang Y001 (D8465)
M8466	●	●	FX3G, FX3GE, FX3U, FX3UC (ab V2.20)	Ausgang Y002 (D8466)	
M8467	●	●	FX3U ^②	Ausgang Y003 (D8467)	

Tab. 9-32: Sondermerker für die Positionierung

- ① Wird in D8336 als Interrupt-Quelle für einen der Ausgänge Y0 bis Y3 der Wert „8“ eingetragen, wird ein Interrupt ausgelöst, wenn der entsprechende Sondermerker gesetzt wird.
- ② Der Ausgang Y003 steht nur zur Verfügung, wenn zwei Adaptermodule FX3U-2HSY-ADP angeschlossen sind.
- ③ Ist einer der Sondermerker M8464 bis M8467 gesetzt, wird am Referenzpunkt nicht der entsprechende Ausgang zurückgesetzt, sondern ein Operand, der in den Sonderregistern D8464 bis D8467 angegeben ist.
- ① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontaktanweisung (z. B. LD- oder LDI-Anweisung) abgefragt werden kann.
- ② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

9.1.21 Sondermerker zur Fehlerdiagnose (2) (M8468–M8511)

Sondermerker Operanden- adresse	① Signal- zustand abfragen	② Signal- zustand festlegen	CPU	Bedeutung
M8468–M8486	—	—	—	Reserviert
M8487	●	—	FX3S	Fehler bei Kommunikation über USB-Schnittstelle
M8488	—	—	—	Reserviert
M8489	●	—	FX3G ^① FX3GC ^① FX3GE, FX3S FX3U ^② FX3UC ^②	Sondermodulparameterfehler
M8490–M8511	—	—	—	Reserviert

Tab. 9-33: Sondermerker zur Fehlerdiagnose (2)

① Ab Version 2.00

② Ab Version 3.10

9.2 Sonderregister (D8000 bis D8511)

In den Sonderregistern sind Datenwerte über SPS-Betriebszustände gespeichert. Die Datenwerte können vom SPS-Programm gelesen bzw. auch verändert werden.

Die Sonderregister lassen sich in zwei Gruppen einteilen:

- ① Sonderregister, deren Datenwerte von einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte von einem SPS-Programm gelesen und verändert werden können.

9.2.1 SPS-Status (D8000–D8009)

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung
D8000	—	●	FX1S FX1N FX2N FX2NC FX3G FX3GC FX3GE FX3S FX3U FX3UC	Watch-Dog-Timer-Einstellung in Einheiten von 1 ms. Der Wert muss größer als die maximale Zykluszeit (D8012) sein. Standardwert: 200 ms
D8001	●	—		Versionsnummer FX1S/FX3S: 22V _{VV} FX1N/FX3G/FX3GC/FX3GE: 26V _{VV} FX2N/FX2NC/FX3U/FX3UC: 24V _{VV} FX3S: 28V _{VV} (z. B. FX1N Version 1.00 → 26100)
D8002	●	—		Speicherkapazität: 0002 → 2k-Schritte (nur bei FX1S) 0004 → 4k-Schritte (FX2N/FX2NC/FX3S*) 0008 → 8k- oder mehr Schritte (nicht bei FX1S) Bei mehr als 16k-Schritten wird in D8002 der Wert „8“ und in D8102 der Wert „16“, „32“ oder „64“ eingetragen.
D8003	●	—		Speichertyp: 00H → RAM (optional) 01H → EPROM (optional) 02H → EEPROM (optional) 0AH → EEPROM (optional, schreibgeschützt) 10H → Integrierter Speicher
D8004	●	—		Fehlermerkeradresse Enthält D8004 z. B. den Wert 8060, ist der Fehlermerker M8060 gesetzt.
D8005	●	—		Batteriespannung; Zum Beispiel bedeutet der Wert 36 eine Spannung von 3,6 V.
D8006	●	—		Wert der Batteriespannung, bei dem der Fehler „niedrige Batteriespannung“ gemeldet wird. Standardeinstellungen: FX2N/FX2NC: 3,0 V („30“) FX3G/FX3GC/FX3GE/FX3U/FX3UC: 2,7 V („27“)
D8007	●	—	Anzahl kurzzeitiger Spannungsausfälle seit dem letzten Einschalten der Versorgungsspannung	

Tab. 9-34: Sonderregister für SPS-Status (1)

* Bei einer FX3S wird auch der Wert „4“ angezeigt, wenn in den Parametern 16k-Schritte eingestellt wurden.

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung
D8008	—	●	FX2N FX2NC FX3U FX3UC	Vorgabe der Verzögerungszeit, die zwischen einem Spannungsausfall und dem Herunterfahren der CPU vergehen soll Standardeinstellungen: FX2N/FX3U: 10 ms (AC-Spannungsversorgung) FX2NC/FX3UC: 5 ms (DC-Spannungsversorgung) FX2N (DC-Spannungsversorgung): Bitte beachten Sie die Hinweise im Abschnitt 8.9.
D8009	●	—	FX2N, FX2NC, FX3G, FX3U, FX3UC	Speicherung der niedrigsten Geräteadresse, die von einem 24-V-DC-Spannungsausfall betroffen ist.

Tab. 9-35: Sonderregister für SPS-Status (2)

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

9.2.2 Zykluszeitmessung und Uhrzeit/Datum (D8010–D8019)

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung
D8010	●	—	FX1S FX1N FX2N FX2NC FX3G FX3GC FX3GE FX3S FX3U FX3UC	Aktuelle Programmzykluszeit in Einheiten von 0,1 ms ^①
D8011	●	—		Minimale Programmzykluszeit in Einheiten von 0,1 ms ^①
D8012	●	—		Maximale Programmzykluszeit in Einheiten von 0,1 ms ^①
D8013	—	●	FX1S FX1N FX2N FX2NC ^② FX3G FX3GC FX3GE FX3S FX3U FX3UC	Integrierte Uhr: Sekunden (0–59)
D8014	—	●		Integrierte Uhr: Minuten (0–59)
D8015	—	●		Integrierte Uhr: Stunden (0–23)
D8016	—	●		Integrierte Uhr: Datum (Tag, 1–31)
D8017	—	●		Integrierte Uhr: Datum (Monat, 1–12)
D8018	—	●		Integrierte Uhr: Datum (Jahr, 0–99)
D8019	—	●		Integrierte Uhr: Wochentag (Sonntag = 0, Samstag = 6)

Tab. 9-36: Sonderregister für Zykluszeitmessung und die integrierte Uhr der SPS

- ① Der angezeigte Wert beinhaltet auch die Wartezeit beim Betrieb mit konstanter Zykluszeit (M8039 ist in diesem Fall gesetzt).
- ② Bei einer FX2NC muss eine Speicherkarte mit integrierter Uhr installiert werden.

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

9.2.3 Eingangsfilter und Indexregister (D8020–D8029)

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung
D8020	—	●	FX1S, FX1N, FX3G, FX3GC FX3GE, FX3U*, FX3UC*	Eingangsfilter für Eingänge X0 bis X7; Voreinstellung: 10 ms (siehe auch Abschnitt 8.5)
			FX2N, FX2NC FX3S, FX3U, FX3UC	Eingangsfilter für Eingänge X0 bis X17; Voreinstellung: 10 ms (siehe auch Abschnitt 8.5)
D8021–D8027	—	—	—	Reserviert
D8028	●	—	FX1S, FX1N, FX2N, FX2NC, FX3G, FX3GC FX3GE, FX3S, FX3U, FX3UC	Aktueller Wert im Index-Register Z0
D8029	●	—		Aktueller Wert im Index-Register V0

Tab. 9-37: Sonderregister für Eingangsfilter und Indexregister

* Bei FX3U- und FX3UC-Grundgeräten mit 16 E/A

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

9.2.4 Analoge Potentiometer und konstante Zykluszeit (D8030–D8039)

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung
D8030	●	—	FX1S, FX1N, FX3G, FX3GE, FX3S	Gelesener Wert vom Potentiometer VR 1 (0 bis 255)
D8031	●	—		Gelesener Wert vom Potentiometer VR 2 (0 bis 255)
D8032–D8038	—	—	—	Reserviert
D8039	—	●	FX1S, FX1N, FX2N, FX2NC, FX3G, FX3GC FX3GE, FX3S, FX3U, FX3UC	Einstellung der konstante Programmzykluszeit in Einheiten von 1 ms Beim Einschalten der SPS wird der Wert 0 [ms] eingetragen.

Tab. 9-38: Sonderregister für analoge Potentiometer und konstante Zykluszeit

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

9.2.5 STL-Status (D8040–D8059)

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung
D8040*	●	—	FX1S, FX1N, FX2N, FX2NC, FX3G, FX3GC FX3GE, FX3S, FX3U, FX3UC	Nummer des 1. aktivierten Schrittstatus
D8041*	●	—		Nummer des 2. aktivierten Schrittstatus
D8042*	●	—		Nummer des 3. aktivierten Schrittstatus
D8043*	●	—		Nummer des 4. aktivierten Schrittstatus
D8044*	●	—		Nummer des 5. aktivierten Schrittstatus
D8045*	●	—		Nummer des 6. aktivierten Schrittstatus
D8046*	●	—		Nummer des 7. aktivierten Schrittstatus
D8047*	●	—		Nummer des 8. aktivierten Schrittstatus
D8048	—	—	—	Reserviert
D8049	●	—	FX2N, FX2NC, FX3G, FX3GC, FX3GE, FX3U, FX3UC	Letzter Fehlermerker Das Register speichert den letzten aktiven Fehlermerker aus dem Bereich S900 bis S999.
D8050 bis D8059	—	—	—	Reserviert

Tab. 9-39: Sonderregister für STL-Status

* Die niedrigste Nummer der aktiven Schritte (S0 bis S899, bei FX3G/FX3GC/FX3GE/FX3U/FX3UC auch S1000 bis S4095) wird in D8040 gespeichert. Die zweitniedrigste Nummer wird in D8041 gespeichert. Die Nummern der aktiven Schritte werden dann nacheinander in die folgenden Register bis D8047 eingetragen (max. 8 Schritte).

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

9.2.6 Fehlerdiagnose (D8060–D8069)

Die Sonderregister zur Fehlerdiagnose werden im Abschnitt 10.1.2 beschrieben.

9.2.7 Register für Link- und Sonderfunktionen (D8070–D8100)

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung
D8070	●	—	FX1S, FX1N, FX2N, FX2NC, FX3G, FX3GC FX3GE, FX3S, FX3U, FX3UC	Überwachungszeit bei Parallel-Link (500 ms)
D8071–D8073	—	—	—	Reserviert
D8074 bis D8098	—	—	FX2N, FX2NC, FX3U, FX3UC	Beim Sampling Trace werden diese Register von der SPS verwendet
	●	—	FX3G (ab V1.10), FX3GC, FX3GE	Diese Register werden für die Impulsweiten- und Periodendauermessung verwendet (siehe Abschnitt 8.4). Bei einem Stop der SPS werden die Register gelöscht.
D8099	—	●	FX2N, FX2NC, FX3U, FX3UC	Ring-Zähler, einstellbar von 0 bis 32767 in 0,1 ms-Schritten Bei einer FX2N oder FX2NC wird der Ring- zähler am Ende des Programmzyklus akti- viert, in dem M8099 gesetzt wurde. Bei einer FX3U oder FX3UC wird der Ring- zähler unmittelbar nach dem Setzen von M8099 aktiviert.
D8100	—	—	—	Reserviert

Tab. 9-40: Sonderregister für Link- und Sonderfunktionen

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

9.2.8 Sonstige Register (D8101–D8119)

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung
D8101	●	—	FX3G, FX3GC, FX3GE, FX3S, FX3U, FX3UC	Versionsnummer FX3G/FX3GC/FX3GE: 26V _{VV} FX3S: 28V _{VV} FX3U/FX3UC: 16V _{VV} (z. B. FX3U Version 1.00 → 16100)
D8102	●	—	FX1S FX1N FX2N FX2NC FX3G FX3GC FX3GE FX3S FX3U FX3UC	Speicherkapazität: 0002 → 2k-Schritte (nur bei FX1S) 0004 → 4k-Schritte (FX2N, FX2NC, FX3S*) 0008 → 8k-Schritte (FX1N, FX2N, FX2NC) 0016 → 16k-Schritte (FX2N, FX2NC, FX3U*, FX3UC*) 0032 → 32k-Schritte (FX3G, FX3GC, FX3GE) 0064 → 64k-Schritte (FX3U, FX3UC) * mit FX3U-FLROM-16
D8103	—	—	—	Reserviert
D8104	●	—	FX2N/FX2NC (ab V3.00)	Identifikations-Code für Speichererweiterung
D8105	●	—		Versionsnummer der Speichererweiterung (z. B. Version 1.00 → 100)
D8106	—	—	—	Reserviert
D8107	●	—	FX3U, FX3UC	Anzahl der gespeicherten Operandenkommentare
D8108	●	—	FX3G, FX3GC, FX3GE, FX3U, FX3UC	Anzahl der angeschlossenen Sondermodule
D8109	●	—	FX2N, FX2NC, FX3G, FX3GC, FX3GE, FX3U, FX3UC	Operandenadresse, an der ein Ausgangsaktualisierungsfehler aufgetreten ist.
D8110	—	—	—	Reserviert
D8111				
D8112	●	—	FX1S FX1N	Erweiterungsadapter FX1N-2AD-BD: Digitaler Eingangswert Kanal 1
D8113	●	—		Erweiterungsadapter FX1N-2AD-BD: Digitaler Eingangswert Kanal 2
D8114	●	●		Erweiterungsadapter FX1N-1DA-BD: Digitaler Ausgangswert
D8115–D8119	—	—	—	Reserviert

Tab. 9-41: Sonstige Sonderregister

* Bei einer FX3S wird auch der Wert „4“ angezeigt, wenn in den Parametern 16k-Schritte eingestellt wurden.

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

9.2.9 Register für Kommunikation (D8120–D8129)

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung
D8120 ^①	—	●	FX1S, FX1N, FX2N, FX2NC, FX3G, FX3GC FX3GE, FX3S, FX3U, FX3UC	Kommunikations-Format
D8121 ^①	—	●		Nummer der lokalen Station (RS485-Netzwerk)
D8122 ^②	●	—		RS-Anweisung, Menge der zu übertragenden Rest-Daten
D8123 ^②	●	—		RS-Anweisung, Menge der empfangenen Daten
D8124	—	●		RS-Anweisung, Telegramm-Header (STX(02H))
D8125	—	●		232ADP, Telegramm-Ende (ETX(03H))
D8126	—	—		—
D8127	—	●	FX1S, FX1N, FX2N, FX2NC, FX3G, FX3GC FX3GE, FX3S, FX3U, FX3UC	RS485-Netzwerk, Kopfadresse der angeforderten Station
D8128	—	●		RS485-Netzwerk, Datenlänge der angeforderten Daten
D8129 ^①	—	●		RS485, Überwachungszeiteinstellung

Tab. 9-42: Sonderregister für Kommunikationsadapter

① Der Inhalt dieser Register bleibt bei einem Ausfall der Versorgungsspannung erhalten.

② Dieses Register wird bei einem Stop der SPS gelöscht.

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

9.2.10 Ausführungsregister für HSZ-, PLSY- und Positionieranweisungen (D8130–D8149)

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung
D8130	●	—	FX2N, FX2NC, FX3U, FX3UC	Aktueller Vergleichszyklus der HSZ-Anweisung
D8131	●	—		Aktueller Vergleichszyklus der HSZ-Anweisung bei aktivierter PLSY-Anweisung
D8132, D8133	●	—		Ausgabefrequenz für die PLSY-Anweisung bei Verwendung der HSZ-Anweisung.
D8134, D8135	●	—		Kopie der Werte für die Vergleichsoperation bei Verwendung der HSZ-Anweisung in Verbindung mit der PLSY-Anweisung
D8136, D8137	●	—	FX1S, FX1N, FX2N, FX2NC, FX3G, FX3GC FX3GE, FX3S, FX3U, FX3UC	Anzahl der mittels PLSY- und PLSR-Anweisung an Y0 und Y1 ausgegebenen Impulse
D8138	●	—	FX3U, FX3UC	Tabellenzähler der DHSCT-Anweisung
D8139	●	—		Anzahl der ausgeführten DHSZ-, DHSCS-, DHSCR- und DHSCT-Anweisungen
D8140, D8141	●	—	FX1S, FX1N, FX2N, FX2NC, FX3GC, FX3GE, FX3S, FX3U, FX3UC	Anzahl der mittels PLSY- und PLSR-Anweisung an Y0 ausgegebenen Impulse
D8142, D8143	●	—		Anzahl der mittels PLSY- und PLSR-Anweisung an Y1 ausgegebenen Impulse
D8144	—	—	—	Reserviert
D8145	—	●	FX1S, FX1N	Drehzahl-Offset für ZRN-, DRVI- und DRVA-Anweisung; Voreinstellung: 0
D8146, D8147	—	●		Maximale Frequenz der Ausgangsimpulse [Hz] für ZRN-, DRVI- und DRVA-Anweisung; Voreinstellung: 100 000
D8148	—	●		Beschleunigungs-/Verzögerungszeit [ms] für ZRN-, DRVI- und DRVA-Anweisung; Voreinstellung: 100
D8149	—	—	—	Reserviert

Tab. 9-43: Sonderregister für HSZ-, PLSY- und Positionieranweisungen

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

9.2.11 Sonderregister für Kommunikation mit Frequenzumrichtern

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung	
D8150	—	●	FX3G (ab V1.10), FX3GC, FX3GE, FX3S, FX3U, FX3UC	Kommunikation über Kanal 1	Wartezeit auf Antwort vom Frequenzumrichter
D8151	●	—			Schrittnummer der Anwei- sung bei der Kommunika- tion mit Frequenzumrichter (Voreinstellung: -1)
D8152	●	—			Fehlercode
D8153	●	—			Schrittnummer der Anwei- sung, bei der während der Kommunikation ein Fehler aufgetreten ist (Gespeicherter Wert, Voreinstellung: -1)
D8154	●	—	FX3U, FX3UC	Kommunikation über Kanal 1	Nummer des Parameters, bei dem während der Aus- führung einer IVBWR- Anweisung ein Fehler auf- getreten ist. (Voreinstellung: -1)
	●	—	FX2N/FX2NC (ab V3.0)		Wartezeit auf Antwort bei einer EXTR- Anweisung
D8155	—	●	FX3G (ab V1.10), FX3GC, FX3GE, FX3U, FX3UC	Kommunikation über Kanal 2	Wartezeit auf Antwort vom Frequenzumrichter
	●	—	FX2N/FX2NC (ab V3.0)		Schrittnummer der Anweisung bei der Kom- munikation mittels einer EXTR-Anweisung
D8156	●	—	FX3G (ab V1.10), FX3GC, FX3GE, FX3U, FX3UC	Kommunikation über Kanal 2	Schrittnummer der Anwei- sung bei der Kommunika- tion mit Frequenzumrichter (Voreinstellung: -1)
	●	—	FX2N/FX2NC (ab V3.0)		Fehlercode bei einer EXTR-Anweisung
D8157	●	—	FX3G (ab V1.10), FX3U, FX3UC	Kommunikation über Kanal 2	Fehlercode
	●	—	FX2N/FX2NC (ab V3.0)		Gespeicherter Fehlercode bei einer EXTR-Anweisung (Voreinstellung: -1)
D8158	●	—	FX3G (ab V1.10), FX3GC, FX3GE, FX3U, FX3UC	Kommunikation über Kanal 2	Schrittnummer der Anwei- sung, bei der während der Kommunikation ein Fehler aufgetreten ist (Gespeicherter Wert, Vor- einstellung: -1)
D8159	●	—	FX3U, FX3UC		Nummer des Parameters, bei dem während der Aus- führung einer IVBWR- Anweisung ein Fehler auf- getreten ist. (Voreinstellung: -1)

Tab. 9-44: Sonderregister für die Kommunikation mit Frequenzumrichtern

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

9.2.12 Verschiedene Funktionen (D8158–D8169)

Sonderregister Operanden-adresse	① Daten-werte lesen	② Daten-werte verändern	CPU	Bedeutung
D8158	—	●	FX1S FX1N	Steuerregister für Anzeigemodul FX1N-5DM, Voreinstellung: -1
D8159	—	●		Steuerregister für Anzeigemodul FX1N-5DM, Voreinstellung: -1
D8160–D8163	—	—	—	Reserviert
D8164	—	●	FX2N (ab V2.00), FX2NC	Anzahl der bei einer FROM/TO-Anweisung übertragenen Operanden
D8165	—	—	—	Reserviert
D8166	●	—	FX3U, FX3UC (ab V3.00)	Sondermodulfehler
D8167, D8168	—	—	—	Reserviert
D8169	●	—	FX3G, FX3GC, FX3GE, FX3S, FX3U, FX3UC (ab V2.00)	<p>Art der Zugriffsbeschränkung durch das zweite Passwort</p> <p><input type="checkbox"/><input type="checkbox"/>00H: Keine Beschränkung, es ist kein 2. Passwort eingerichtet</p> <p><input type="checkbox"/><input type="checkbox"/>10H: Schreibschutz für Programm</p> <p><input type="checkbox"/><input type="checkbox"/>11H: Lese- und Schreibschutz für Programm</p> <p><input type="checkbox"/><input type="checkbox"/>12H: Alle Online-Operationen sind gesperrt.</p> <p><input type="checkbox"/><input type="checkbox"/>20H: Das Passwort wurde aufgehoben, es bestehen keine Einschränkungen</p> <p>Die als „<input type="checkbox"/><input type="checkbox"/>“ dargestellten Stellen werden vom System verwendet.</p>

Tab. 9-45: Register für diverse Funktionen

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

9.2.13 Sonderregister für ein n:n-Netzwerk (D8173–D8180, D8201–D8259)

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung	
D8170–D8172	—	—	—	Reserviert	
D8173	●	—	FX1S, FX1N, FX2N (ab V2.00), FX2NC, FX3G, FX3GC, FX3GE, FX3S, FX3U, FX3UC	Stationsnummer	
D8174	●	—		Gesamte Anzahl der Slave-Stationen	
D8175	●	—		Aktualisierter Bereich	
D8176	—	●		Einstellungen	Stationsnummer
D8177	—	●			Anzahl der Slave-Stationen
D8178	—	●			Aktualisierter Bereich
D8179	—	●			Wiederholungsversuche
D8180	—	●		Überwachungszeit	
D8181	—	—	—	Reserviert	
D8182–D8199	●	—	FX1S bis FX3UC	Indexregister (siehe folgenden Abschnitt)	
D8200	—	—	—	Reserviert	
D8201 (D201 bei FX1S)	●	—	FX1S, FX1N, FX2N (ab V2.00), FX2NC, FX3G, FX3GC, FX3GE, FX3S, FX3U, FX3UC	Aktuelle Abtastzeit	
D8202 (D202 bei FX1S)	●	—		Maximale Abtastzeit	
D8203 (D203 bei FX1S)	●	—		Anzahl der Kommunikationsfehler bei der Master-Station	
D8204 (D204 bei FX1S)	●	—		Anzahl der Kommunikationsfehler bei der Slave-Station 1	
D8205 (D205 bei FX1S)	●	—		Anzahl der Kommunikationsfehler bei der Slave-Station 2	
D8206 (D206 bei FX1S)	●	—		Anzahl der Kommunikationsfehler bei der Slave-Station 3	
D8207 (D207 bei FX1S)	●	—		Anzahl der Kommunikationsfehler bei der Slave-Station 4	
D8208 (D208 bei FX1S)	●	—		Anzahl der Kommunikationsfehler bei der Slave-Station 5	
D8209 (D209 bei FX1S)	●	—		Anzahl der Kommunikationsfehler bei der Slave-Station 6	
D8210 (D210 bei FX1S)	●	—		Anzahl der Kommunikationsfehler bei der Slave-Station 7	
D8211 (D211 bei FX1S)	●	—		Fehlercode des Kommunikationsfehler bei der Master-Station	
D8212 (D212 bei FX1S)	●	—		Fehlercode des Kommunikationsfehler bei der Slave-Station 1	
D8213 (D213 bei FX1S)	●	—		Fehlercode des Kommunikationsfehler bei der Slave-Station 2	
D8214 (D214 bei FX1S)	●	—		Fehlercode des Kommunikationsfehler bei der Slave-Station 3	
D8215 (D215 bei FX1S)	●	—		Fehlercode des Kommunikationsfehler bei der Slave-Station 4	
D8216 (D216 bei FX1S)	●	—		Fehlercode des Kommunikationsfehler bei der Slave-Station 5	
D8217 (D217 bei FX1S)	●	—		Fehlercode des Kommunikationsfehler bei der Slave-Station 6	
D8218 (D218 bei FX1S)	●	—		Fehlercode des Kommunikationsfehler bei der Slave-Station 7	
D8219–D8259	—	—	—	Reserviert	

Tab. 9-46: Sonderregister für ein n:n-Netzwerk

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

9.2.14 Index-Register (D8182–D8199)

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung
D8182	●	—	FX1S, FX1N, FX2N, FX2NC, FX3G, FX3GC, FX3GE, FX3S, FX3U, FX3UC	Index-Register Z1
D8183	●	—		Index-Register V1
D8184	●	—		Index-Register Z2
D8185	●	—		Index-Register V2
D8186	●	—		Index-Register Z3
D8187	●	—		Index-Register V3
D8188	●	—		Index-Register Z4
D8189	●	—		Index-Register V4
D8190	●	—		Index-Register Z5
D8191	●	—		Index-Register V5
D8192	●	—		Index-Register Z6
D8193	●	—		Index-Register V6
D8194	●	—		Index-Register Z7
D8195	●	—		Index-Register V7
D8196–D8199	—	—		—

Tab. 9-24: Index-Register

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

9.2.15 Sonderregister für analoge Adaptermodule und Erweiterungsadapter (D8260–D8299)

Sonderregister Operandenadresse	① Datenwerte lesen	② Datenwerte verändern	CPU	Bedeutung
D8260 bis D8269	—	●	FX3U, FX3UC (ab V2.00)	Sonderregister für das 1. analoge Adaptermodul
	—	●	FX3G (ab V1.10) FX3GE, FX3S	Sonderregister für den 1. analogen Erweiterungsadapter ^①
D8270 bis D8279	—	●	FX3U, FX3UC (ab V2.00)	Sonderregister für das 2. analoge Adaptermodul
	—	●	FX3G (ab V1.10)	Sonderregister für den 2. analogen Erweiterungsadapter ^②
D8280 bis D8289	—	●	FX3U, FX3UC (ab V2.00)	Sonderregister für das 3. analoge Adaptermodul
	—	●	FX3G, FX3GE, FX3S	Sonderregister für das 1. analoge Adaptermodul
D8290 bis D8299	—	●	FX3U, FX3UC (ab V2.00)	Sonderregister für das 4. analoge Adaptermodul
	—	●	FX3G ^③ , FX3GC FX3GE ^④	Sonderregister für das 2. analoge Adaptermodul

Tab. 9-47: Sonderregister für angeschlossene Analogmodule

- ① Installiert im Erweiterungssteckplatz der Grundgeräte FX3G-14M□/□ oder FX3G-24M□/□, auf dem linken Erweiterungssteckplatz (Position 1) der Grundgeräte FX3G-40□/□ oder FX3G-60M□/□ oder in einem FX3GE- oder FX3S-Grundgerät.
- ② Installiert auf dem rechten Erweiterungssteckplatz (Position 2) der Grundgeräte FX3G-40□/□ oder FX3G-60M□/□.
- ③ Ein zweites analoges Adaptermodul kann nur bei den Grundgeräten FX3G-40□/□ und FX3G-60M□/□ installiert werden.
- ④ Bei den Grundgeräten der FX3GE-Serie gelten die integrierten analogen Eingänge und der integrierte analoge Ausgang als 1. analoges Adaptermodul. Wird ein analoges Adaptermodul angeschlossen, gilt dieses Adaptermodul als 2. analoges Adaptermodul.
- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

HINWEIS

Die Funktion der Sonderregister D8260 bis D8299 ist in den Handbüchern zu den Analogmodulen beschrieben.

9.2.16 Sonderregister für Anzeigemodule (D8300–D8309)

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung
D8300	—	●	FX3G (ab V1.10) FX3GE FX3U FX3UC	Steuerregister für Anzeigemodul, Voreinstellung: „-1“
D8301	—	●		Steuermerker für Anzeigemodul Voreinstellung: „-1“
D8302	●	—		Spracheinstellung*: „0“: Japanisch Andere Werte als „0“: Englisch
D8303	●	—		Kontrasteinstellung der LC-Anzeige Voreinstellung: „0“
D8304–D8309	—	—	—	Reserviert

Tab. 9-48: Sonderregister für Anzeigemodule FX3G-5DM und FX3U-7DM

* Diese Einstellung bleibt auch beim Ausschalten der Versorgungsspannung erhalten.

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

9.2.17 Sonderregister für die RND-Anweisung (D8310 und D8311)

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung
D8310	●	—	FX3U FX3UC	Daten zur Erzeugung einer Zufallszahl (niederwertiges Wort)
D8311	●	—		Daten zur Erzeugung einer Zufallszahl (höherwertiges Wort)

Tab. 9-49: Sonderregister zur Erzeugung einer Zufallszahl mit einer RND-Anweisung

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

9.2.18 Sonderregister zur Fehlerdiagnose (D8312–D8328)

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung	
D8312	—	●	FX3U, FX3UC	Gespeicherte Nummer des Schritts, bei dem ein Fehler aufgetreten ist (Anzeige durch M8068, 32 Bit)	Niederwertiges Wort
D8313	—	●			Höherwertiges Wort
D8314*	●	—		Nummer des Schritts, bei dem ein Fehler aufgetreten ist (Anzeige durch M8065 bis M8067, 32 Bit)	Niederwertiges Wort
D8315*	●	—			Höherwertiges Wort
D8316	●	—		Nummer des Schritts, in dem eine ungültige E/A-Adresse verwendet wird (direkt oder indirekt über Index-Register adressiert) (Anzeige durch M8316, 32 Bit)	Niederwertiges Wort
D8317	●	—			Höherwertiges Wort
D8318	●	—	FX3U, FX3UC (ab V2.20)	Fehler bei Initialisierung eines Pufferspeichers (Anzeige durch M8318)	Nr. des Sondermoduls
D8319	●	—			Pufferspeicher- adresse
D8320–D8328	—	—	—	Reserviert	

Tab. 9-50: Sonderregister, die Programmschrittnummern oder Informationen zur Fehlerdiagnose enthalten

* Der Inhalt von D8114 und D8115 wird gelöscht, wenn die SPS in die Betriebsart RUN geschaltet wird.

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

9.2.19 Sonderregister für Impulsausgabe und Positionierung

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung	
D8329	—	—	—	Reserviert	
D8330	●	—	FX3U, FX3UC (ab V2.20)	DUTY-Anweisung: Zykluszähler für Ausgang 1	
D8331	●	—		DUTY-Anweisung: Zykluszähler für Ausgang 2	
D8332	●	—		DUTY-Anweisung: Zykluszähler für Ausgang 3	
D8333	●	—		DUTY-Anweisung: Zykluszähler für Ausgang 4	
D8334	●	—		DUTY-Anweisung: Zykluszähler für Ausgang 5	
D8336	●	—	FX3U, FX3UC (ab V1.30)	Interrupteingang für DVIT-Anweisung	
D8337–D8339	—	—	—	Reserviert	

Tab. 9-51: Sonderregister für Impulsausgabe und Positionierung (1)

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung		
D8340	—	●	FX3G, FX3GC, FX3GE, FX3S, FX3U, FX3UC	Y000	Aktueller Wert	Bits 15–0
D8341	—	●			Bits 31–16	
D8342	—	●			Minimale Geschwindigkeit, Voreinstellung: 0	
D8343	—	●			Maximale Geschwindigkeit, Voreinstellung: 100000	Bits 15–0
D8344	—	●			Bits 31–16	
D8345	—	●			Kriechgeschwindigkeit, Voreinstellung: 1000	
D8346	—	●			Geschwindigkeit für Refer- enzpunktfahrt, Voreinstellung: 50000	Bits 15–0
D8347	—	●			Bits 31–16	
D8348	—	●			Beschleunigungszeit, Voreinstellung: 100	
D8349	—	●			Verzögerungszeit, Voreinstellung: 100	
D8350	—	●			FX3G, FX3GC, FX3GE, FX3S, FX3U, FX3UC	Y001
D8351	—	●	Bits 31–16			
D8352	—	●	Minimale Geschwindigkeit, Voreinstellung: 0			
D8353	—	●	Maximale Geschwindigkeit, Voreinstellung: 100000	Bits 15–0		
D8354	—	●	Bits 31–16			
D8355	—	●	Kriechgeschwindigkeit, Voreinstellung: 1000			
D8356	—	●	Geschwindigkeit für Refer- enzpunktfahrt, Voreinstellung: 50000	Bits 15–0		
D8357	—	●	Bits 31–16			
D8358	—	●	Beschleunigungszeit, Voreinstellung: 100			
D8359	—	●	Verzögerungszeit, Voreinstellung: 100			
D8360	—	●	FX3G, FX3U, FX3UC	Y002		
D8361	—	●			Bits 31–16	
D8362	—	●			Minimale Geschwindigkeit, Voreinstellung: 0	
D8363	—	●			Maximale Geschwindigkeit, Voreinstellung: 100000	Bits 15–0
D8364	—	●			Bits 31–16	
D8365	—	●			Kriechgeschwindigkeit, Voreinstellung: 1000	
D8366	—	●			Geschwindigkeit für Refer- enzpunktfahrt, Voreinstellung: 50000	Bits 15–0
D8367	—	●			Bits 31–16	
D8368	—	●			Beschleunigungszeit, Voreinstellung: 100	
D8369	—	●			Verzögerungszeit, Voreinstellung: 100	
D8370	—	●			FX3U	Y003*
D8371	—	●	Bits 31–16			
D8372	—	●	Minimale Geschwindigkeit, Voreinstellung: 0			
D8373	—	●	Maximale Geschwindigkeit, Voreinstellung: 100000	Bits 15–0		
D8374	—	●	Bits 31–16			
D8375	—	●	Kriechgeschwindigkeit, Voreinstellung: 1000			
D8376	—	●	Geschwindigkeit für Refer- enzpunktfahrt, Voreinstellung: 50000	Bits 15–0		
D8377	—	●	Bits 31–16			
D8378	—	●	Beschleunigungszeit; Voreinstellung: 100			
D8379	—	●	Verzögerungszeit; Voreinstellung: 100			
D83800–D8392	—	—	—	—		

Tab. 9-52: Sonderregister für Positionierung

* Der Ausgang Y003 steht nur zur Verfügung, wenn zwei Adaptermodule FX3U-2HSY-ADP angeschlossen sind.

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

9.2.20 Sonderregister für Kommunikation (D8370–D8392)

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung	
D8370	—	●	FX3G FX3GC FX3GE	RS2-Anweisung (Kanal 0)	Einstellung des Übertragungs- formats
D8371	—	—	—	Reserviert	
D8372*	●	—	FX3G FX3GC FX3GE	RS2-Anweisung (Kanal 0)	Anzahl der noch zu übertragen- den Daten
D8373*	●	—			Anzahl der empfangenen Daten
D8374	—	—	—	Reserviert	
D8375	●	—	FX3G FX3GC FX3GE	RS2-Anweisung (Kanal 0)	Anzeige der Kommunikations- Parameter
D8376–D8378	—	—	—	Reserviert	
D8379	—	●	FX3G FX3GC FX3GE	RS2-Anweisung (Kanal 0)	Überwachungszeit
D8380	—	●			Header 1 und 2, Voreinstellung: STX
D8381	—	●			Header 3 und 4
D8382	—	●			Endekennung 1 und 2 Voreinstellung: ETX
D8383	—	●			Endekennung 3 und 4
D8384	●	—			Empfangene Prüfsumme
D8385	●	—			Berechnete Prüfsumme für emp- fangene Daten
D8386	●	—			Gesendete Prüfsumme
D8387	—	—	—	Reserviert	
D8388					
D8389	●	—	FX3G FX3GC FX3GE	RS2-Anweisung (Kanal 0)	Anzeige der Betriebsart
D8390–D8392	—	—	—	Reserviert	

Tab. 9-53: Sonderregister für eine RS2-Anweisung

* Die Inhalte dieser Sonderregister werden bei einem Stopp der SPS gelöscht.

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

9.2.21 Sonderregister für Interrupt-Programme (D8393–D8397)

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung
D8393	—	●	FX3U, FX3UC	Verzögerungszeit
D8394	—	—	—	Reserviert
D8395				
D8396	—	●	FX3UC	Konfiguration des integrierten CC-Link/LT (nur bei FX3UC-32MT-LT2)
D8397	—	—	—	Reserviert

Tab. 9-54: Sonderregister für Interrupt-Programme

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

9.2.22 Ringzähler (D8398 und D8399)

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung
D8398	—	●	FX3G, FX3GC, FX3GE, FX3S, FX3U, FX3UC	32-Bit-Ringzähler, Zählbereich: 0 bis 2147483647
D8399	—	●		Der Ringzähler wird nach dem Setzen von M8398 aktiviert.

Tab. 9-55: Die Register D8398 und D8399 speichern den Zählwert des Ringzählers

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

9.2.23 Sonderregister für Kommunikation (D8400–D8439)

HINWEIS

Bitte beachten Sie, dass die einzelnen Register aus dem Bereich D8400 bis D8459, abhängig von der verwendeten Kommunikationsmethode, unterschiedliche Bedeutungen haben. Eine ausführliche Beschreibung dieser Merker finden Sie im Kommunikationshandbuch der Steuerungen der MELSEC FX-Familie oder in den Bedienungsanleitungen der verwendeten Module.

RS2-Anweisungen und RS485-Netzwerk

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung	
D8400	—	●	FX3G, FX3GC, FX3S, FX3U, FX3UC	RS2-Anweisung (Kanal 1)	Einstellung des Übertragungsformats
D8401	—	—	—	Reserviert	
D8402*	●	—	FX3G, FX3GC, FX3S, FX3U, FX3UC	RS2-Anweisung (Kanal 1)	Anzahl der noch zu übertragenden Daten
D8403*	●	—			Anzahl der empfangenen Daten
D8404	—	—	—	Reserviert	
D8405	●	—	FX3G, FX3GC, FX3S, FX3U, FX3UC	RS2-Anweisung (Kanal 1)	Anzeige der Kommunikations-Parameter
D8406–D8408	—	—	—	Reserviert	
D8409	—	●	FX3G, FX3GC, FX3S, FX3U, FX3UC	RS2-Anweisung (Kanal 1)	Überwachungszeit
D8410	—	●			Header 1 und 2, Voreinstellung: STX
D8411	—	●			Header 3 und 4
D8412	—	●			Endekennung 1 und 2 Voreinstellung: ETX
D8413	—	●			Endekennung 3 und 4
D8414	●	—			Empfangene Prüfsumme
D8415	●	—			Berechnete Prüfsumme für empfangene Daten
D8416	●	—			Gesendete Prüfsumme
D8417	—	—	—	Reserviert	
D8418					
D8419	●	—	FX3G, FX3GC, FX3S, FX3U, FX3UC	RS2-Anweisung (Kanal 1)	Anzeige der Betriebsart

Tab. 9-56: Sonderregister zur Kommunikation mit einer RS2-Anweisung (Kanal 1)

* Die Inhalte dieser Sonderregister werden bei einem Stopp der SPS gelöscht.

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung	
D8420	—	●	FX3G, FX3GC, FX3GE, FX3U, FX3UC	RS2-Anweisung (Kanal 2)	Einstellung des Übertragungs- formats
D8421	—	●		RS485- Netzwerk	Einstellung der Stationsnummer
D8422*	●	—		RS2-Anweisung (Kanal 2)	Anzahl der noch zu übertragen- den Daten
D8423*	●	—			Anzahl der empfangenen Daten
D8424	—	—	—	Reserviert	
D8425	●	—	FX3G, FX3GC, FX3GE, FX3U, FX3UC	RS2-Anweisung (Kanal 2)	Anzeige der Kommunikations- Parameter
D8426	—	—	—	Reserviert	
D8427	—	●	FX3G, FX3GC, FX3GE, FX3U, FX3UC	RS485- Netzwerk	Erstes Registers beim Senden auf Anforderung
D8428	—	●			Anzahl der Register beim Senden auf Anforderung
D8429	—	●	FX3G, FX3GC, FX3GE, FX3U, FX3UC	RS2-Anweisung (Kanal 2)	Überwachungszeit
D8430	—	●			Header 1 und 2, Voreinstellung: STX
D8431	—	●			Header 3 und 4
D8432	—	●			Endekennung 1 und 2 Voreinstellung: ETX
D8433	—	●			Endekennung 3 und 4
D8434	●	—			Empfangene Prüfsumme
D8435	●	—			Berechnete Prüfsumme für emp- fangene Daten
D8436	●	—			Gesendete Prüfsumme
D8437	—	—	—	Reserviert	
D8438	●	—	FX3G, FX3GC, FX3GE, FX3U, FX3UC	Fehlercode für Fehler bei serieller Kommunikation über Kanal 2	
D8439	●	—	FX3G, FX3GC, FX3GE, FX3U, FX3UC	RS2-Anweisung (Kanal 2)	Anzeige der Betriebsart

Tab. 9-57: Sonderregister zur Kommunikation mit einer RS2-Anweisung (Kanal 2) oder über ein RS485-Netzwerk

* Die Inhalte dieser Sonderregister werden bei einem Stopp der SPS gelöscht.

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

MODBUS-Kommunikation

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung
MODBUS-Kommunikation über Kanal 1 des SPS-Grundgeräts				
D8400	—	●	FX3G ^① , FX3GC, FX3S, FX3U ^② , FX3UC ^②	Kommunikationsformat
D8401	—	●		Protokoll
D8402	—	●		Fehlercode für die Kommunikation
D8403	—	●		Zusätzliche Informationen zum Fehler
D8404	—	●		Schrittnummer, bei der ein Kommunikationsfehler aufgetreten ist
D8405	●	—		Anzeige des Kommunikationsformats
D8406	● ^③	● ^③	FX3U ^② , FX3UC ^②	Endekennung im ASCII-Modus
D8407	●	—	FX3G ^① , FX3GC, FX3S, FX3U ^② , FX3UC ^②	Nummer des ausgeführten Schritts
D8408	●	—		Aktuelle Zahl der Wiederholversuche
D8409	—	●		Antwortüberwachungszeit
D8410	—	●		Broadcast-Verzögerungszeit
D8411	—	●		Wiederholungsintervall
D8412	—	●		Anzahl der Wiederholversuche
D8413	—	—		—
D8414	—	●	FX3G ^① , FX3GC, FX3S, FX3U ^② , FX3UC ^②	Slave-Stationsnummer
D8415	—	●	FX3U ^② , FX3UC ^②	Einstellungen zum Speichern des Kommunikationsstatus
D8416	—	●		Startadresse in der SPS zum Speichern des Kommunikationsstatus
D8417, D8418	—	—	—	Reserviert
D8419	●	—	FX3G ^① , FX3GC, FX3S, FX3U ^② , FX3UC ^②	Anzeige der Betriebsart

Tab. 9-58: Sonderregister für die MODBUS-Kommunikation (1)

① Ab Version 1.30

② Ab Version 2.40

③ In einer Slave-Station: Nur lesen des Datenwerts; In der Master-Station: Lesen und verändern des Datenwerts

① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.

② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung
MODBUS-Kommunikation über Kanal 2 des SPS-Grundgeräts				
D8420	—	●	FX3G ^① , FX3GC, FX3S, FX3U ^② FX3UC ^②	Kommunikationsformat
D8421	—	●		Protokoll
D8422	—	●		Fehlercode für die Kommunikation
D8423	—	●		Zusätzliche Informationen zum Fehler
D8424	—	●		Schrittnummer, bei der ein Kommunikationsfehler aufgetreten ist
D8425	●	—		Anzeige des Kommunikationsformats
D8426	● ^③	● ^③	FX3U ^② FX3UC ^②	Endekennung im ASCII-Modus
D8427	●	—	FX3G ^① , FX3GC, FX3S, FX3U ^② FX3UC ^②	Nummer des ausgeführten Schritts
D8428	●	—		Aktuelle Zahl der Wiederholversuche
D8429	—	●		Antwortüberwachungszeit
D8430	—	●		Broadcast-Verzögerungszeit
D8431	—	●		Wiederholungsintervall
D8432	—	●		Anzahl der Wiederholversuche
D8433	—	—	—	Reserviert
D8434	—	●	FX3G ^① , FX3GC, FX3S, FX3U ^② FX3UC ^②	Slave-Stationsnummer
D8435	—	●	FX3U ^② FX3UC ^②	Einstellungen zum Speichern des Kommunikationsstatus
D8436	—	●		Startadresse in der SPS zum Speichern des Kommunikationsstatus
D8437, D8438	—	—	—	Reserviert
D8439	●	—	FX3G ^① , FX3GC, FX3S, FX3U ^② FX3UC ^②	Anzeige der Betriebsart

Tab. 9-59: Sonderregister für die MODBUS-Kommunikation (2)

① Ab Version 1.30

② Ab Version 2.40

③ In einer Slave-Station: Nur lesen des Datenwerts; In der Master-Station: Lesen und verändern des Datenwerts

① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.

② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung	
MODBUS-Kommunikation (gemeinsam für Kommunikation über Kanal 1 oder Kanal 2)					
D8470	—	●	FX3U ^① FX3UC ^①	1. Zuordnung der MODBUS-Operanden	Operandencode, Anzahl der Operanden
D8471					Startadresse in der SPS
D8472	—	●	FX3U ^① FX3UC ^①	2. Zuordnung der MODBUS-Operanden	Operandencode, Anzahl der Operanden
D8473					Startadresse in der SPS
D8474	—	●	FX3U ^① FX3UC ^①	3. Zuordnung der MODBUS-Operanden	Operandencode, Anzahl der Operanden
D8475					Startadresse in der SPS
D8476	—	●	FX3U ^① FX3UC ^①	4. Zuordnung der MODBUS-Operanden	Operandencode, Anzahl der Operanden
D8477					Startadresse in der SPS
D8478	—	●	FX3U ^① FX3UC ^①	5. Zuordnung der MODBUS-Operanden	Operandencode, Anzahl der Operanden
D8479					Startadresse in der SPS
D8480	—	●	FX3U ^① FX3UC ^①	6. Zuordnung der MODBUS-Operanden	Operandencode, Anzahl der Operanden
D8481					Startadresse in der SPS
D8482	—	●	FX3U ^① FX3UC ^①	7. Zuordnung der MODBUS-Operanden	Operandencode, Anzahl der Operanden
D8483					Startadresse in der SPS
D8484	—	●	FX3U ^① FX3UC ^①	8. Zuordnung der MODBUS-Operanden	Operandencode, Anzahl der Operanden
D8485					Startadresse in der SPS

Tab. 9-60: Sonderregister für die MODBUS-Kommunikation (3)

① Ab Version 2.40

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

CF-Speicherkartenadaptermodul FX3U-CF-ADP

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung
Kommunikation mit dem FX3U-CF-ADP über Kanal 1 des SPS-Grundgeräts				
D8400–D8401	—	—	—	Reserviert
D8402	●	—	FX3U ^① FX3UC ^①	Schrittnummer der für das FX3U-CF-ADP ausgeführten Anweisung ^②
D8403				
D8404, D8405	—	—	—	Reserviert
D8406	●	—	FX3U ^① FX3UC ^①	Status des FX3U-CF-ADP
D8407	—	—	—	Reserviert
D8408	●	●	FX3U ^① FX3UC ^①	Version des FX3U-CF-ADP
D8409–D8413	—	—	—	Reserviert
D8414	●	—	FX3U ^① FX3UC ^①	Schrittnummer der Anweisung, bei deren Ausführung ein Fehler aufgetreten ist (M8418 ist gesetzt)
D8415				
D8416	—	—	—	Reserviert
D8417	●	—	FX3U ^① FX3UC ^①	Detaillierter Code des Fehlers, der bei Ausführung einer Anweisung für das FX3U-CF-ADP aufgetreten ist. ^{②③}
D8418	●	—		Code des Fehlers, der bei Ausführung einer Anweisung für das FX3U-CF-ADP aufgetreten ist. ^{②③}
D8419	●	—		Anzeige der Betriebsart
Kommunikation mit dem FX3U-CF-ADP über Kanal 2 des SPS-Grundgeräts				
D8420–D8421	—	—	—	Reserviert
D8422	●	—	FX3U ^① FX3UC ^①	Schrittnummer der für das FX3U-CF-ADP ausgeführten Anweisung ^②
D8423				
D8424, D8425	—	—	—	Reserviert
D8426	●	—	FX3U ^① FX3UC ^①	Status des FX3U-CF-ADP
D8427	—	—	—	Reserviert
D8428	●	●	FX3U ^① FX3UC ^①	Version des FX3U-CF-ADP
D8429–D8423	—	—	—	Reserviert
D8424	●	—	FX3U ^① FX3UC ^①	Schrittnummer der Anweisung, bei deren Ausführung ein Fehler aufgetreten ist (M8418 ist gesetzt)
D8425				
D8426	—	—	—	Reserviert
D8427	●	—	FX3U ^① FX3UC ^①	Detaillierter Code des Fehlers, der bei Ausführung einer Anweisung für das FX3U-CF-ADP aufgetreten ist. ^{②③}
D8428	●	—		Code des Fehlers, der bei Ausführung einer Anweisung für das FX3U-CF-ADP aufgetreten ist. ^{②③}
D8429	●	—		Anzeige der Betriebsart

Tab. 9-61: Sonderregister für das CF-Speicherkartenadaptermodul FX3U-CF-ADP

① Ab Version 2.61

② Wird gelöscht, wenn die SPS von der Betriebsart STOP in die Betriebsart RUN geschaltet wird.

③ Eine ausführliche Beschreibung der Fehlercodes enthält die Bedienungsanleitung zum FX3U-CF-ADP.

① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.

② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

Ethernet-Modul FX3U-ENET-ADP

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung	
Kommunikation mit dem FX3U-ENET-ADP über Kanal 1 des SPS-Grundgeräts					
D8400	●	—	FX3G ^① FX3GC ^① FX3GE FX3S FX3U ^② FX3UC	IP-Adresse	Niederwertiges Wort
D8401					Höherwertiges Wort
D8402	●	—		Subnet-Mask	Niederwertiges Wort
D8403					Höherwertiges Wort
D8404	●	—		IP-Adresse des Stan- dard-Routers	Niederwertiges Wort
D8405					Höherwertiges Wort
D8406	●	—		Zustand des Moduls	
D8407	●	—		Verbindungsstatus der Ethernet-Schnittstelle	
D8408	●	—		Version des FX3U-ENET-ADP	
D8409	—	●		Kommunikationsüberwachungszeit	
D8410	—	●		Zwangweise Aufhebung einer Verbindung	
D8411	●	—		Ergebnis beim Einstellen der Zeit	
D8412–D8414	●	—		Host-MAC-Adresse	
D8415	—	—		—	Reserviert
D8416	●	—	FX3G ^① FX3GC ^①	Modulkennung des FX3U-ENET-ADP	
D8417	●	—	FX3GE FX3S FX3U ^② FX3UC ^②	Fehlercode des FX3U-ENET-ADP	
D8418	—	—	—	Reserviert	
D8419	●	—	FX3G ^① FX3GC ^① FX3GE FX3S FX3U ^② FX3UC ^②	Betriebsart	

Tab. 9-62: Sonderregister für das Ethernet-Adaptermodul FX3U-ENET-ADP (1)

① Ab Version 2.00

② Ab Version 3.10

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung	
Kommunikation mit dem FX3U-ENET-ADP über Kanal 2 des SPS-Grundgeräts					
D8420	●	—	FX3G ^① FX3GC ^① FX3GE FX3S FX3U ^② FX3UC ^②	IP-Adresse	Niederwertiges Wort
D8421					Höherwertiges Wort
D8422	●	—		Subnet-Maske	Niederwertiges Wort
D8423					Höherwertiges Wort
D8424	●	—		IP-Adresse des Stan- dard-Routers	Niederwertiges Wort
D8425					Höherwertiges Wort
D8426	●	—		Zustand des Moduls	
D8427	●	—		Verbindungsstatus der Ethernet-Schnittstelle	
D8428	●	—		Version des FX3U-ENET-ADP	
D8429	—	●		Kommunikationsüberwachungszeit	
D8430	—	●		Zwangweise Aufhebung einer Verbindung	
D8431	●	—		Ergebnis beim Einstellen der Zeit	
D8432–D8434	●	—		Host-MAC-Adresse	
D8435	—	—		—	Reserviert
D8436	●	—	FX3G ^① FX3GC ^①	Modulkennung des FX3U-ENET-ADP	
D8437	●	—	FX3GE FX3S FX3U ^② FX3UC ^②	Fehlercode des FX3U-ENET-ADP	
D8438	—	—	—	Reserviert	
D8439	●	—	FX3G ^① FX3GC ^① FX3GE FX3S FX3U ^② FX3UC ^②	Betriebsart	
Kommunikation mit dem FX3U-ENET-ADP (gemeinsam für Kommunikation über Kanal 1 oder Kanal 2)					
D8490, D8491	—	—	—	Reserviert	
M8492	—	●	FX3G ^③ FX3GC ^③ FX3GE FX3S	Eingestellte IP-Adresse	Niederwertiges Wort
M8493					Höherwertiges Wort
M8494	—	●		Eingestellte Subnet-Maske	Niederwertiges Wort
M8495					Höherwertiges Wort
M8496	—	●		Eingestellte IP-Adres- se des Standard- Routers	Niederwertiges Wort
M8497					Höherwertiges Wort
M8498	●	—		Fehlercode für Schreiben in IP-Adressenspeicher	
M8499	●	—		Fehlercode für Löschen des IP-Adressenspeichers	

Tab. 9-63: Sonderregister für das Ethernet-Adaptermodul FX3U-ENET-ADP (2)

① Ab Version 2.00

② Ab Version 3.10

③ Ab Version 2.10

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

9.2.24 Fehlererkennung bei Sondermodulen (D8440–D8459)

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung
D8440–D8448	—	—	—	Reserviert
D8449	—	●	FX3G FX3GC FX3GE FX3U FX3UC*	Fehlercode bei Sondermodulfehler (M8449 ist in diesem Fall gesetzt.)
D8450–D8459	—	—	—	Reserviert

Tab. 9-64: Sonderregister zur Fehlererkennung bei Sondermodulen

* ab Version 2.20

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

9.2.25 Sonderregister für Positionierung (D8460–D8467)

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung	
D8460–D8463	—	—	—	Reserviert	
D8464	—	●	FX3G, FX3GC, FX3GE, FX3S, FX3U, FX3UC ^①	Ausgang Y000 (M8464)	
D8465	—	●		Angabe eines Operanden, der durch eine DSZR- oder ZRN-Anweisung zurückgesetzt wird ^②	Ausgang Y001 (M8465)
D8466	—	●		FX3G, FX3GE, FX3U, FX3UC ^①	Ausgang Y002 (M8466)
D8467	—	●		FX3U ^③	Ausgang Y003 (M8467)

Tab. 9-65: Sonderregister für Positionierung

① ab Version 2.20

② Ist einer der Sondermerker M8464 bis M8467 gesetzt, wird am Referenzpunkt nicht der entsprechende Ausgang zurückgesetzt, sondern ein Operand, der in den Sonderregistern D8464 bis D8467 angegeben ist.

③ Der Ausgang Y003 steht nur zur Verfügung, wenn zwei Adaptermodule FX3U-2HSY-ADP angeschlossen sind.

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

9.2.26 Sonderregister zur Fehlerdiagnose (2) (D8468–D8511)

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung
D8468–D8486	—	—	—	Reserviert
D8487	●	—	FX3S	Fehler bei Kommunikation über USB-Schnittstelle
D8488	—	—	—	Reserviert
D8489	●	—	FX3G ① FX3GC ① FX3GE, FX3S FX3U ② FX3UC ②	Fehlercodes des Sondermodulparameterfehlers
D8490–D8511	—	—	—	Reserviert

Tab. 9-66: Sonderregister zur Fehlerdiagnose (2)

① Ab Version 2.00

② Ab Version 3.10

10 Programmfehler

10.1 Fehlererkennung

10.1.1 Sondermerker (M8060–M8069)

Sondermerker Operanden- adresse	① Signal- zustand abfragen	② Signal- zustand festlegen	CPU	Bedeutung	(„PROG-E“/ „ERROR“)- LED	SPS- Modus	
M8060	●	—	FX2N, FX2NC, FX3G, FX3GC, FX3GE, FX3U, FX3UC	E/A-Konfigurations- fehler	Aus	RUN	
M8061	●	—	FX1S, FX1N, FX2N, FX2NC FX3G, FX3GC, FX3GE, FX3S, FX3U, FX3UC	SPS-Hardwarefehler	Ein	STOP	
M8062	●	—	FX1S, FX1N, FX2N, FX2NC, FX3S ^① , FX3U ^① , FX3UC ^①	Kommunikation zwi- schen SPS und Pro- grammiergerät gestört	Aus	RUN	
	●	—	FX3G, FX3GC, FX3GE	Fehler bei serieller Kommunikation (Ch. 0)			
M8063 ^②	●	—	FX1S, FX1N, FX2N, FX2NC FX3G, FX3GC, FX3GE, FX3S, FX3U, FX3UC	Fehler bei serieller Kommunikation (Ch. 1)	Aus	RUN	
M8064	●	—		Parameterfehler	Blinkt	STOP	
M8065	●	—		Programmsyntax- fehler	Blinkt	STOP	
M8066	●	—		Programmierfehler	Blinkt	STOP	
M8067 ^③	●	—		Ausführungsfehler	Aus	RUN	
M8068	—	●		Ausführungsfehler (gespeichert)	Aus	RUN	
M8069 ^④	—	●		FX2N, FX2NC, FX3G, FX3GC, FX3GE, FX3U, FX3UC	E/A-Bus-Fehler	—	—

Tab. 10-1: Sondermerker für Fehlererkennung

- ① Wird nur gesetzt, wenn in einem FX3S-, FX3U- oder FX3UC-Grundgerät ein Fehler beim Zugriff auf den Speicher aufgetreten ist (Fehlercode 6230).
 - ② Bei Steuerungen der FX1S-, FX1N-, FX2N- und FX2NC-Serie wird dieser Merker zurückgesetzt, wenn die SPS von STOP in RUN geschaltet wird. Bei einer SPS der FX3G-, FX3GC-, FX3GE-, FX3S-, FX3U- oder FX3UC-Serie wird M8063 beim Einschalten der Versorgungsspannung zurückgesetzt. Ein Fehler bei der seriellen Kommunikation über Kanal 2 wird bei einer FX3G, FX3GC, FX3GE, FX3U oder FX3UC durch M8438 angezeigt.
 - ③ Dieser Merker wird zurückgesetzt, wenn die Steuerung von STOP in RUN geschaltet wird.
 - ④ Nach dem Setzen von M8069 wird eine Kontrolle des E/A-Bus ausgeführt. Wenn hierbei ein Fehler erkannt wird, wird der Fehlercode 6130 in das Sonderregister D8069 geschrieben und der Sondermerker M8061 gesetzt.
- ① Sondermerker, bei denen nur der Signalzustand in einem SPS-Programm mit einer Kontaktauweisung (z.B. LD- oder LDI-Anweisung) abgefragt werden kann.
 - ② Sondermerker, die direkt mit einer Anweisung in einem SPS-Programm gesetzt bzw. zurückgesetzt werden können.

10.1.2 Sonderregister (D8060–D8069, D8449)

Sonderregister Operanden- adresse	① Daten- werte lesen	② Daten- werte verändern	CPU	Bedeutung
D8060	●	—	FX2N, FX2NC, FX3G, FX3GC, FX3GE, FX3U, FX3UC	E/A-Adresse des fehlerhaften Grund- oder Erweiterungsgerätes Angabe als vierstellige Zahl: 1. Ziffer: 0 = Ausgang, 1 = Eingang 2. bis 4. Ziffer: Angabe des ersten Operanden des fehlerhaftem E/A-Moduls
D8061	●	—	FX2N, FX2NC FX3G, FX3GC, FX3GE, FX3S, FX3U, FX3UC	Fehlercode des SPS-Hardwarefehlers (siehe Fehlercodetabelle)
D8062	●	—	FX1S, FX1N, FX2N, FX2NC FX3G, FX3GC, FX3GE, FX3S, FX3U, FX3UC	Fehlercode für Kommunikationsfehler zwischen SPS und Programmiergerät (siehe Fehlercode-tabelle)
	●	—	FX3G, FX3GC, FX3GE	Fehlercode für Fehler bei serieller Kommunika-tion (Ch. 0)
D8063 ^①	●	—	FX1S, FX1N, FX2N, FX2NC FX3G, FX3GC, FX3GE, FX3S, FX3U, FX3UC	Fehlercode für Fehler bei serieller Kommunika-tion (Ch. 1)
D8064	●	—		Fehlercode des Parameterfehlers
D8065	●	—		Fehlercode des Programmsyntaxfehlers
D8066	●	—		Fehlercode des Programmierfehlers
D8067 ^②	●	—		Fehlercode des Ausführungsfehlers
D8068 ^②	—	●		Schrittnummer des Ausführungsfehlers Bei mehr als 32k-Schritten wird die Schritt-nummer in D8313 und D8312 gespeichert.
D8069 ^②	●	—		Schrittnummer der Fehler M8065 bis M8067 Bei mehr als 32k-Schritten wird die Schritt-nummer in D8315 und D8314 gespeichert.
D8449	●	—		FX3G, FX3GC, FX3GE, FX3U, FX3UC ^③

Tab. 10-2: Sonderregister für Fehlererkennung

① Bei Steuerungen der FX1S-, FX1N-, FX2N- und FX2NC-Serie wird dieses Register gelöscht, wenn die SPS von STOP in RUN geschaltet wird. Bei einer SPS der FX3G-, FX3GC-, FX3GE-, FX3S-, FX3U- oder FX3UC-Serie wird D8063 beim Einschalten der Versorgungsspannung gelöscht.

② Diese Register werden gelöscht, wenn die SPS in die Betriebsart RUN gebracht wird.

③ Ab Version 2.20

- ① Sonderregister, deren Datenwerte in einem SPS-Programm nur gelesen werden können.
- ② Sonderregister, deren Datenwerte in einem SPS-Programm gelesen und verändert werden können.

10.2 Fehlercodes

10.2.1 Fehlercodes (3801–3840)

Fehler	Sonderregister	Fehlercode	Bedeutung	Fehler beheben
Fehler bei serieller Kommunikation	D8438	0000	Kein Fehler	Überprüfen Sie die Spannungsversorgung und Verdrahtung. Nähere Hinweise finden Sie im FX-Kommunikationshandbuch.
		3801	Paritäts-/Überlauf-/Rahmenfehler	
		3802	Zeichenfehler	
		3803	Prüfsummenfehler	
		3804	Formatfehler	
		3805	Ungültiger Befehl	
		3806	Überwachungszeit abgelaufen	
		3807	Fehler bei der Initialisierung des Modems	
		3808	Parameterfehler bei einem n:n-Netzwerk	
		3809	n:n-Netzwerk: fehlerhafte Einstellungen	
		3812	Zeichenfehler bei Parallel-Link	
		3813	Prüfsummenfehler bei Parallel-Link	
		3814	Formatfehler bei Parallel-Link	
		3820	Fehler beim Datenaustausch mit einem Frequenzumrichter	
		3821	MODBUS: Kommunikationsfehler	
		3830	Fehler beim Zugriff auf den Speicher	Bei Verwendung einer Speicherkassette prüfen Sie bitte, ob die Speicherkassette korrekt installiert ist. Tritt der Fehler wieder auf oder wird keine Speicherkassette verwendet, ist wahrscheinlich die SPS defekt. Wenden Sie sich an ihren Mitsubishi Vertriebspartner.
		3840	Fehler beim Anschluss eines Adaptermoduls	Prüfen Sie den Anschluss der Adaptermodule.

Tab. 10-3: Fehlercodes bei Störungen der seriellen Kommunikation

10.2.2 Fehlercodes (6101–6409)

Fehler	Sonderregister	Fehlercode	Bedeutung	Fehler beheben
SPS-Hardwarefehler	D8061	0000	Kein Fehler	Prüfen Sie die Verbindung zwischen Programmiergerät und Steuerung.
		6101	RAM-Fehler	
		6102	Schaltkreis fehlerhaft	
		6103	E/A-Fehler (M8069 = „1“)	
		6104	Fehler in der 24-V-DC-Versorgungsspannung (M8069 = „1“)	Die Programmzykluszeit ist größer als der in D8000 angegebene Wert.
		6105	Watch-Dog-Timer-Fehler	Prüfen Sie die Versorgungsspannung (24 V DC) der Erweiterungsmodule. Diese muss innerhalb von 10 s nach Einschalten des Grundgeräts anliegen.
		6106	Fehler bei der Erzeugung der E/A-Tabelle (CPU-Fehler) Nach dem Einschalten der SPS war die Spannungsversorgung eines Erweiterungsmoduls mit integrierter Spannungsversorgung nicht eingeschaltet. Oder die E/A-Zuordnung in einem CC-Link-Netzwerk war nicht möglich.	
		6107	Fehler bei der Systemkonfiguration	Prüfen Sie die Anzahl der angeschlossenen Sondermodule
Kommunikationsfehler zwischen SPS und Programmiergerät (nur bei FX2N und FX2NC)	D8062	0000	Kein Fehler	Beheben Sie die Fehlerursache und wiederholen Sie die Übertragung.
		6201	Paritäts-/Überlauf-/Rahmenfehler	
		6202	Kommunikationszeichen fehlerhaft	
		6203	Prüfsummen-Fehler bei Datenübertragung	
		6204	Datenformat fehlerhaft	
		6205	Anweisungsfehler	
Fehler bei serieller Kommunikation		6230	Fehler beim Zugriff auf den Speicher	Bei Verwendung einer Speicherkassette prüfen Sie bitte, ob die Speicherkassette korrekt installiert ist. Tritt der Fehler wieder auf oder wird keine Speicherkassette verwendet, ist wahrscheinlich die SPS defekt. Wenden Sie sich an ihren Mitsubishi Vertriebspartner.

Tab. 10-4: Fehlercodes (6101 bis 6230)

Fehler	Sonderregister	Fehlercode	Bedeutung	Fehler beheben
Fehler bei der seriellen Kommunikation	D8063	0000	Kein Fehler	Überprüfen Sie die Spannungsversorgung und die Verdrahtung. Nähere Hinweise finden Sie im FX-Kommunikationshandbuch.
		6301	Paritäts-/Überlauf-/Rahmenfehler	
		6302	Zeichenfehler	
		6303	Prüfsummen-Fehler	
		6304	Formatfehler	
		6305	Falscher Befehl (Computer-Link) Bei Stationsnummer FF war der empfangene Befehl nicht GW (global).	
		6306	Watch-Dog-Timer-Fehler	
		6307	Fehler bei der Initialisierung des Modems	
		6308	Parameterfehler bei einem n:n-Netzwerk	
		6312	Zeichenfehler im Parallel-Link	
		6313	Prüfsummen-Fehler im Parallel-Link	
		6314	Datenformatfehler im Parallel-Link	
		6320	Fehler beim Datenaustausch mit einem Frequenzumrichter	
		6321	MODBUS: Kommunikationsfehler	
Parameterfehler	D8064	6330	Fehler beim Zugriff auf den Speicher	Bei Verwendung einer Speicherkassette prüfen Sie bitte, ob die Speicherkassette korrekt installiert ist. Tritt der Fehler wieder auf oder wird keine Speicherkassette verwendet, ist wahrscheinlich die SPS defekt. Wenden Sie sich an ihren Mitsubishi Vertriebspartner.
		6340	Fehler beim Anschluss eines Adaptermoduls	Prüfen Sie den Anschluss der Adaptermodule.
		0000	Kein Fehler	<ul style="list-style-type: none"> • Stoppen Sie die SPS und korrigieren Sie die fehlerhaften Parameter. • Wenn eine Speicherkassette installiert ist, vergewissern Sie sich, dass keine der folgenden Funktionen bei einer SPS verwendet wird, die diese Funktionen nicht unterstützt: <ul style="list-style-type: none"> – SPS permanent sperren (FX3U/FX3UC ab V 2.61) – Schreibschutz des Programms für Blockpasswörter (FX3U/FX3UC ab V 3.00) – FX3U-FLROM-1M (FX3U/FX3UC ab V 3.00)
Parameterfehler	D8064	6401	Programm-Prüfsummen-Fehler	<ul style="list-style-type: none"> • Stoppen Sie die SPS und korrigieren Sie die fehlerhaften Parameter. • Korrigieren Sie die Sondermodulparameter, und schalten Sie dann die Versorgungsspannung der SPS aus und wieder ein.
		6402	Einstellung der Speicherkapazität fehlerhaft	
		6403	Einstellung für Latch-Operanden fehlerhaft	
Parameterfehler	D8064	6404	Einstellung für Kommentarbereich fehlerhaft	<ul style="list-style-type: none"> • Prüfen Sie den Fehlercode (Inhalt von D8489), führen Sie bei den Sondermodulen und Adaptermodulen eine Fehlerdiagnose aus, und korrigieren Sie Einstellung der Sondermodulparameter. • Korrigieren Sie die Sondermodulparameter, und schalten Sie dann die Versorgungsspannung der SPS aus und wieder ein.
		6405	Einstellung für File-Register fehlerhaft	
		6406	Fehler beim Eintrag der Vorgabewerte in Pufferspeicher von Sondermodulen oder Prüfsummenfehler bei einer Positionieranweisung	
Parameterfehler	D8064	6407	Andere falsche Parameter	<ul style="list-style-type: none"> • Prüfen Sie den Fehlercode (Inhalt von D8489), führen Sie bei den Sondermodulen und Adaptermodulen eine Fehlerdiagnose aus, und korrigieren Sie Einstellung der Sondermodulparameter. • Korrigieren Sie die Sondermodulparameter, und schalten Sie dann die Versorgungsspannung der SPS aus und wieder ein.
		6409	Andere falsche Parameter	
		6420	Prüfsummenfehler bei Sondermodulparametern	
Parameterfehler	D8064	6421	Fehlerhafte Einstellung der Sondermodulparameter	<ul style="list-style-type: none"> • Prüfen Sie den Fehlercode (Inhalt von D8489), führen Sie bei den Sondermodulen und Adaptermodulen eine Fehlerdiagnose aus, und korrigieren Sie Einstellung der Sondermodulparameter. • Korrigieren Sie die Sondermodulparameter, und schalten Sie dann die Versorgungsspannung der SPS aus und wieder ein.
		6421	Fehlerhafte Einstellung der Sondermodulparameter	

Tab. 10-5: Fehlercodes (6301 bis 6421)

10.2.3 Fehlercodes (6501–6511)

Fehler	Sonderregister	Fehlercode	Bedeutung	Fehler beheben
Programmsyntaxfehler	D8065	0000	Kein Fehler	Anweisungen werden schon während der Programmierung geprüft. Sollte ein Programmsyntaxfehler auftreten, korrigieren Sie bitte die entsprechende Anweisung.
		6501	Anweisung, Operandensymbol oder Operandenadresse falsch programmiert.	
		6502	Keine OUT-T-Anweisung oder OUT-C-Anweisung vor der Programmierung eines entsprechenden Sollwertes.	
		6503	<ul style="list-style-type: none"> Einer OUT-T-Anweisung oder OUT-C-Anweisung folgt keine Sollwertangabe. Die Anzahl der Operanden für eine Applikationsanweisung ist nicht ausreichend. 	
		6504	<ul style="list-style-type: none"> Dieselbe Pointer-Markierung ist öfter verwendet worden. Dieselbe Eingangsbedingung für ein Interrupt-Programm oder einen High-Speed-Counter wurde öfter verwendet. 	
		6505	Unzulässige Operandenadresse	
		6506	Ungültige Anweisung	
		6507	Ungültige Pointervergabe (P)	
		6508	Ungültige Interrupt-Pointer-Vergabe (I)	
		6509	Andere Fehler	
		6510	Fehlerhafte Nummer der MC-Verschachtelungsebene (N)	
6511	Die Interrupt- und High-Speed-Counter-Eingangsadressen überlappen.			

Tab. 10-6: Fehlercodes (6501 bis 6510)

10.2.4 Fehlercodes (6601–6609)

Fehler	Sonderregister	Fehlercode	Bedeutung	Fehler beheben
Programmierfehler	D8066	0000	Kein Fehler	<p>Ein Programmfehler tritt auf, wenn eine fehlerhafte Kombination von Anweisungen oder eine falsche Beziehung zwischen paarweise zusammengehörigen Anweisungen auftritt.</p> <p>Die erkannten Fehler müssen Sie im Programmier-Modus korrigieren.</p>
		6601	Die LD- oder LDI-Anweisung wurde 9 Mal oder häufiger nacheinander programmiert.	
		6602	<ul style="list-style-type: none"> Keine LD- oder LDI-Anweisung. Die LD-, LDI-, AND-, ANI-Anweisung wurde unzulässig eingesetzt. Folgende Anweisungen sind nicht korrekt verknüpft: STL, RET, MCR, P, I, EI, DI, IRET, FOR, NEXT, FEND, SRET, END. Die MPP-Anweisung fehlt. 	
		6603	Die MPS-Anweisung wurde 12 Mal oder häufiger nacheinander programmiert.	
		6604	Die MPS-, MRD- und MPP-Anweisungen wurden unzulässig eingesetzt.	
		6605	<ul style="list-style-type: none"> Die STL-Anweisung wurde 9 Mal oder häufiger nacheinander programmiert. Die MC-, MCR- oder Interrupt-Anweisung wurde innerhalb eines Schrittes programmiert. Die RET-Anweisung wurde außerhalb des Schrittstatus programmiert oder fehlt im Programm. 	
		6606	<ul style="list-style-type: none"> Pointer P oder Interrupt-Pointer I fehlen. Die IRET-/ SRET-Anweisung fehlt. Die IRET-/ SRET- und Interrupt-Anweisungen wurden im Hauptprogramm programmiert. STL/RET/MC oder MCR wurden in einem Unterprogramm oder einer Interrupt-Routine programmiert. 	
		6607	<ul style="list-style-type: none"> Unzulässige FOR-NEXT-Anweisungen, 6 oder mehr Ebenen. Die folgenden Anweisungen wurden in einer FOR-NEXT-Schleife programmiert: STL, RET, MC, MCR, IRET, FEND, SRET, END. 	
		6608	<ul style="list-style-type: none"> Unzulässige MC-, MCR-Anweisungen MCR N0 fehlt. Eine IRET-/ SRET oder Interrupt-Anweisung wurde zwischen den MC- und MCR-Blöcken programmiert. 	
		6609	Andere Fehler	

Tab. 10-7: Fehlercodes (6601 bis 6609)

10.2.5 Fehlercodes (6610–6632)

Fehler	Sonderregister	Fehlercode	Bedeutung	Fehler beheben
Programmfehler	D8066	6610	Die LD-/LDI-Anweisung wurde mehr als 8 Mal in Folge programmiert.	Diese Fehler treten bei einer falschen Kombination von Anweisungen auf. Beheben Sie die Fehler im Programmiermodus.
		6611	Die Anzahl der LD-/LDI-Anweisungen ist kleiner als die der ANB-/ORB-Anweisungen.	
		6612	Die Anzahl der LD-/LDI-Anweisungen ist größer als die der ANB-/ORB-Anweisungen.	
		6613	Die MPS-Anweisung wurde mehr als 12 Mal in Folge programmiert.	
		6614	Die MPS-Anweisung fehlt.	
		6615	Die MPP-Anweisung fehlt.	
		6616	Inkorrekte Verwendung der MPS-, MRD- und MPP-Anweisung. Unter Umständen fehlt die Spulenangabe.	
		6617	Eine der folgenden Anweisungen ist nicht mit der aktiven Bus-Linie verbunden: STL, RET, MCR, Pointer (P), Interrupt (I), EI, DI, SRET, IRET, FOR, NEXT, FEND und END.	
		6618	Die STL-, RET-, MC- oder MCR-Anweisungen wurden innerhalb einer Unter- oder Interrupt-Routine programmiert.	
		6619	Eine ungültige Anweisung wurde innerhalb einer FOR/NEXT-Schleife programmiert: STL, RET, MC, MCR, I, IRET, SRET	
		6620	Die Verschachtelungsgröße für FOR/NEXT-Schleifen wurde überschritten.	
		6621	Ungleiche Anzahl von FOR- und NEXT-Anweisungen.	
		6622	Die NEXT-Anweisung wurde nicht gefunden.	
		6623	Die MC-Anweisung wurde nicht gefunden.	
		6624	Die MCR-Anweisung wurde nicht gefunden.	
		6625	Eine STL-Verzweigungsanweisung treibt mehr als 8 Parallelzweige.	
		6626	Eine ungültige Anweisung wurde innerhalb eines STL-, RET-Blocks programmiert: MC, MCP, I, IRET, SRET.	
		6627	Die RET-Anweisung wurde nicht gefunden.	
		6628	Inkorrekte Programmierung einer I-, IRET- oder SRET-Anweisung im Hauptprogramm	
		6629	Das Pointer (P)- oder Interrupt (I)-Label wurde nicht gefunden.	
6630	Die SRET- oder IRET- Anweisung wurde nicht gefunden.			
6631	Die SRET-Anweisung wurde an einer ungültigen Stelle programmiert.			
6632	Die IRET-Anweisung wurde an einer ungültigen Stelle programmiert.			

Tab. 10-6: Fehlercodes (6610 bis 6632)

10.2.6 Fehlercodes (6701–6710)

Fehler	Sonderregister	Fehlercode	Bedeutung	Fehler beheben
Ausführungsfehler	D8067	0000	Kein Fehler	<p>Diese Fehler treten während der Ausführung einer Anweisung auf. Im Fehlerfall müssen Sie die SPS stoppen und die Fehler im Programmier-Modus beseitigen.</p> <p>Ein Ausführungsfehler kann selbst dann auftreten, wenn kein Syntax- oder Programmfehler gemeldet wird. (So ist D500Z z. B. eine gültige Darstellung. Hat Z jedoch einen Wert von 100, wird versucht auf das Datenregister D600 zuzugreifen. In diesem Fall tritt ein Fehler auf, da D600 nicht existiert.)</p> <p>Dieser Fehler tritt bei der Ausführung der Operation auf.</p> <ul style="list-style-type: none"> • Prüfen Sie das Programm und den Inhalt der in den Anweisungen verwendeten Operanden. • Prüfen Sie, ob die Pufferspeicherbereiche, auf die zugegriffen werden soll, im Sondermodul existieren. • Prüfen Sie, ob die Sondermodule korrekt angeschlossen sind. <p>Diese Fehler treten während der Ausführung einer Anweisung auf. Im Fehlerfall müssen Sie die SPS stoppen und die Fehler im Programmier-Modus beseitigen.</p> <p>Ein Ausführungsfehler kann selbst dann auftreten, wenn kein Syntax- oder Programmfehler gemeldet wird. (So ist D500Z z. B. eine gültige Darstellung. Hat Z jedoch einen Wert von 100, wird versucht auf das Datenregister D600 zuzugreifen. In diesem Fall tritt ein Fehler auf, da D600 nicht existiert.)</p> <p>Zum Beispiel wurde bei einer Transferanweisung derselbe Operand als Ziel und als Quelle angegeben.</p>
		6701	<ul style="list-style-type: none"> • Für die CJ-Anweisung wurde kein Sprungziel angegeben. • Eine Pointer-Markierung wird in einem Block programmiert, der erst nach der END-Anweisung abgearbeitet wird. • Eine unabhängige Sprungmarke wurde innerhalb einer FOR-NEXT-Schleife oder in einer Unterroutine festgelegt. 	
		6702	6 oder mehr CALL-Anweisungen	
		6703	3 oder mehr Interrupt-Ebenen	
		6704	6 oder mehr FOR-NEXT-Ebenen	
		6705	In einer Applikationsanweisung wurde ein falscher Operand eingesetzt.	
		6706	Der Operandenbereich oder Datenbereich, der in einer Applikationsanweisung programmiert wurde, liegt außerhalb des zulässigen Bereichs.	
		6707	Der Zugriff ist auf ein File-Register erfolgt, dass sich außerhalb des zulässigen Adressenbereiches befindet.	
		6708	Fehler in Verbindung mit einer FROM/TO-Anweisung	
		6709	Andere Fehler (z. B. fehlende IRET-Anweisung, unzulässige Beziehung zwischen FOR-NEXT usw.)	
6710	Fehlerhafte Angabe von Operanden einer Anweisung			

Tab. 10-7: Fehlercodes (6701 bis 6710)

10.2.7 Fehlercodes (6730–6773)

Fehler	Sonderregister	Fehlercode	Bedeutung	Fehler beheben
PID-Ausführungsfehler	D8067	6730	Abtastzyklus T_S ($T_S < 0$ oder > 32767)	Der angegebene Parameterwert liegt außerhalb des zulässigen Bereiches. Der PID-Befehl muss zurückgesetzt werden, bevor die Ausführung fortgesetzt wird.
		6732	Filterkoeffizient α ($\alpha < 0$ oder ≥ 101)	
		6733	Proportionalkonstante K_P ($K_P < 0$ oder > 32767)	
		6734	Integrationskonstante T_I ($T_I < 0$ oder 32767)	
		6735	Differentiationverstärkung K_D ($K_D < 0$ oder ≥ 101)	
		6736	Differentiationskonstante T_D ($T_D < 0$ oder > 32767)	
		6740	Die Abtastzeit T_S ist kleiner als die Programm-Zykluszeit	Abtastzeit wird gleich der Zykluszeit gesetzt.
		6742	Der Istwert von Δ ist zu groß	Die betroffenen Daten werden auf den nächsten Grenzwert zurückgesetzt. Bei allen Fehlercodes (außer 6745) ist das entweder -32768 oder +32767. Die Ausführung wird fortgesetzt aber der PID-Befehl sollte zurückgesetzt werden.
		6743	Die berechnete Abweichung ε ist zu groß	
		6744	Das Integrationsergebnis ist zu groß	
		6745	Differentiationswert ist zu groß oder Differenzwert übersteigt den zul. Bereich	
		6746	Differentiationsergebnis ist zu groß	
		6747	Das gesamte PID-Ergebnis ist zu groß	Korrigieren Sie die Einstellungen
		6748	Der obere Grenzwert für die vom PID-Regler ausgegebenen Werte ist kleiner als der untere Grenzwert.	
		6749	Die Einstellungen für den Istwert- oder den Ausgangswertalarm sind nicht korrekt. (z. B. sind negative Werte eingestellt.)	Die Schwankungen können vom PID-Regler nicht ausgeglichen werden. Die Ausführung des PID-Befehls wird beendet. Der PID-Befehle muss zurückgesetzt werden.
		6750	Sollwert - Istwert < 150 oder der Regelkreis ist instabil (Die Differenz zwischen Soll- und Istwert schwankt zu stark und zu schnell)	
		6751	Der Soll-Wert ist erheblich zu groß.	
6752	Große Schwankungen bei der Auto-Tuning-Funktion			
6753	Unzulässiger Ausgangswert beim Auto-Tuning (Oberer Grenzwert $<$ unterer Grenzwert)	Korrigieren Sie die Einstellungen		
6754	Unzulässige Istwert-Hysterese beim Auto-Tuning (SHPV < 0)			
6755	Beim Auto-Tuning wurden die verwendeten Operanden überschrieben.	Stellen Sie sicher, dass die von der PID-Regelung verwendeten Operanden im Programm nicht überschrieben werden.		

Tab. 10-10: Fehlercodes (6730 bis 6755)

HINWEIS

| Weitere Details zu den Fehlercodes der PID-Anweisung enthält Abschnitt 7.3.8.

Fehler	Sonderregister	Fehlercode	Bedeutung	Fehler beheben
PID-Ausführungsfehler	D8067	6756	Fehlerhaftes Ergebnis beim Auto-Tuning wegen zu großer Messzeit	Die Zeit für das Auto-Tuning ist länger als erforderlich. Vergrößern Sie die Spanne zwischen dem oberen und dem unteren Grenzwert des Ausgangswertes, verkleinern Sie den Filterkoeffizienten α oder verringern Sie die Schwelle für den Istwert (SHPV).
		6757	Beim Auto-Tuning wird der zulässige Bereich für die Proportionalkonstante K_P überschritten. (K_P darf zwischen 0 und 32767 liegen.)	Im Vergleich mit dem Ist-Wert ist die Änderung der Proportionalkonstanten K_P gering. Multiplizieren Sie den Istwert mit dem Faktor 10, damit sich dieser Wert während des Auto-Tuning stärker ändert.
		6758	Beim Auto-Tuning wird der zulässige Bereich für die Integrationskonstante T_I überschritten. (T_I darf zwischen 0 und 32767 liegen.)	Die Zeit für das Auto-Tuning ist länger als erforderlich. Vergrößern Sie die Spanne zwischen dem oberen und dem unteren Grenzwert des Ausgangswertes, verkleinern Sie den Filterkoeffizienten α oder verringern Sie die Schwelle für den Istwert (SHPV).
		6759	Beim Auto-Tuning wird der zulässige Bereich für die Differentiationskonstante T_D überschritten. (T_D darf zwischen 0 und 32767 liegen.)	
Ausführungsfehler	D8067	6760	Prüfsummenfehler beim Lesen der Absolut-Position von einem Servo-Verstärker	Prüfen Sie den Anschluss des Servo-Verstärkers und die Parameter. Prüfen Sie auch die ABS-Anweisung.
		6762	Die Schnittstelle, die für die Kommunikation mit einem Frequenzumrichter angegeben wurde, wird bereits für einen anderen Datenaustausch verwendet.	Prüfen und korrigieren Sie die Einstellungen.
		6763	Der Eingang, der für eine DSZR-, DVIT- oder ZRN-Anweisung angegeben wurde, wird bereits von einer anderen Anweisung verwendet.	Prüfen Sie, dass der Eingang einer DSZR-, DVIT- oder ZRN-Anweisung nicht für Interrupts, High-Speed-Counter, die Pulse-Catch-Funktion oder eine SPD-Anweisung verwendet wird.
			Der Operand für den Interrupt einer DVIT-Anweisung überschreitet den zulässigen Bereich.	Prüfen Sie den Inhalt von D8336.
6764	Ein Ausgang zur Ausgabe von Impulsen wird bereits in einer Positionier- oder Impulsausgabeanweisung verwendet.	Prüfen und korrigieren Sie die Einstellungen.		
Ausführungsfehler	D8067	6765	Es sind zu viele Applikationsanweisungen programmiert.	Verringern Sie die Zahl der Applikationsanweisungen im Programm
		6770	Fehler beim Schreiben in eine Speicherkassette (Flash-EPROM)	Tauschen Sie die Speicherkassette.
		6771	Es ist keine Flash-EPROM-Speicherkassette installiert.	Prüfen Sie, ob eine Speicherkassette korrekt installiert ist.
		6772	Der Schreibschutz der Flash-EPROM-Speicherkassette ist aktiviert.	Schalten Sie den Schreibschutz an der Kassette aus, bevor Sie Daten in die Speicherkassette transferieren.
		6773	Während der Übertragung des Programms in der Betriebsart RUN konnte nicht auf die Flash-EPROM-Speicherkassette zugegriffen werden.	Während Änderungen in der Betriebsart RUN sollten keine Daten in die Speicherkassette übertragen oder aus diesem Speicher gelesen werden.

Tab. 10-11: Fehlercodes (6756 bis 6764)

HINWEIS | Weitere Details zu den Fehlercodes der PID-Anweisung enthält Abschnitt 7.3.8.

10.2.8 Fehlercodes (8702–8730)

Fehler	Sonderregister	Fehlercode	Bedeutung	Fehler beheben
USB-Kommunikation	D8487	8702	Zeichenfehler	Prüfen Sie die Leitungsverbindung zwischen dem Programmierwerkzeug und der SPS. Diese Fehler können auftreten, wenn das USB-Kabel während der Monitorfunktion entfernt und wieder angeschlossen wird.
		8703	Prüfsummenfehler	
		8704	Fehlerhaftes Datenformat	
		8705	Fehlerhaftes Kommando	Bei Verwendung einer Speicherkassette prüfen Sie bitte, ob die Speicherkassette korrekt installiert ist. Tritt der Fehler wieder auf oder wird keine Speicherkassette verwendet, ist wahrscheinlich die SPS defekt. Wenden Sie sich an ihren Mitsubishi Vertriebspartner.
		8730	Fehler beim Zugriff auf den Speicher	

Tab. 10-12: Fehlercodes 8702 bis 8730 (Fehler bei USB-Kommunikation)

10.2.9 Fehlercodes bei Fehlern in Sondermodulen

Fehler	Sonderregister	Fehlercode	Bedeutung	Fehler beheben
Fehler bei einem Sondermodul	D8449	0000	Kein Fehler	—
		□020*	Datensummenfehler	Prüfen Sie, ob die Sondermodule korrekt angeschlossen sind.
		□021*	Datenfehler	
		□022*	Fehler beim Zugriff auf das System	
		□025*	Fehler bei der Summe der Zugriffe in einer anderen Station über CC-Link	
		□026*	Nachrichtenfehler in einer anderen Station über CC-Link	
		□030*	Fehler beim Zugriff auf den Speicher	Bei Verwendung einer Speicherkassette prüfen Sie bitte, ob die Speicherkassette korrekt installiert ist. Tritt der Fehler wieder auf oder wird keine Speicherkassette verwendet, ist wahrscheinlich die SPS defekt. Wenden Sie sich an ihren Mitsubishi Vertriebspartner.
		□080*	Fehler bei der Ausführung einer FROM- oder TO-Anweisung.	<ul style="list-style-type: none"> • Prüfen Sie das Programm und den Inhalt der in den Anweisungen verwendeten Operanden. • Prüfen Sie, ob die Pufferspeicherbereiche, auf die zugegriffen werden soll, im Sondermodul existieren. • Prüfen Sie, ob die Sondermodule korrekt angeschlossen sind.
		□090*	Fehler beim Zugriff auf periphere Geräte	Prüfen Sie die Verbindung zwischen dem Programmierwerkzeug und der SPS. Prüfen Sie, ob die Sondermodule korrekt angeschlossen sind.

Tab. 10-13: Fehlercodes in D8449 bei Fehlern in Sondermodule

* Der Platzhalter „□“ steht für die Nummer des Sondermoduls. Diese kann zwischen 0 und 7 liegen (siehe Abschnitt 7.3)

10.2.10 Fehlercodes bei Fehlern in Sondermodulparameter

Fehler	Sonderregister	Fehlercode	Bedeutung	Fehler beheben
Fehler in Sondermodulparameter	D8489	□□01*	Überwachungszeit bei Einstellung von Sondermodulparametern überschritten	Schalten Sie die Versorgungsspannung der SPS aus. Prüfen Sie dann die Spannungsversorgung und den Anschluss der Sonder- und Adaptermodule.
		□□02*	Fehlerhafte Einstellung der Sondermodulparameter	Die Sondermodulparameter sind nicht korrekt eingestellt. <ul style="list-style-type: none"> Führen Sie bei den Sondermodulen/Adaptermodulen eine Fehlerdiagnose aus, und stellen Sie die Parameter korrekt ein. Korrigieren Sie die Sondermodulparameter, und schalten Sie dann die Versorgungsspannung der SPS aus und wieder ein.
		□□03*	Sondermodul für die eingestellten Sondermodulparameter existiert nicht.	Es sind Sondermodulparameter eingestellt, aber das entsprechende Sonder- oder Adaptermodul ist nicht angeschlossen. Prüfen Sie, ob Sondermodule/Adaptermodule angeschlossen sind.
		□□04*	Eingestellte Sondermodulparameter werden nicht unterstützt.	Prüfen Sie die Sondermodulparameter und vergewissern Sie sich, dass die angeschlossenen Sondermodule/Adaptermodule die eingestellten Parameter unterstützen.

Tab. 10-14: Fehler bei Sondermodulparametern

* Der Platzhalter „□□“ steht für die Nummer des Sonder- oder Adaptermoduls. Diese kann zwischen 00 (Sondermodul 0) und 70 (Sondermodul 7) liegen (siehe Abschnitt 7.3). Falls der Platzhalter „□□“ den Wert „81“ oder „82“ annimmt, deutet dies auf den 1. bzw. 2. Kommunikationskanal (Adaptermodul) hin.

Falls ein Fehler in mehreren Sonder- oder Adaptermodulen aufgetreten ist, gibt „□□“ von allen Sonder- oder Adaptermodulen, in denen der Fehler aufgetreten ist, das Modul mit der niedrigsten Nummer an.

10.2.11 Anzeige von Fehlern in Sondermodule (FX3U, FX3UC ab V 3.00)

Fehler	Sonderregister	Gesetztes Bit	Bedeutung	Fehler beheben
Sondermodulfehler	D8166	Bit 0	Fehler beim Zugriff auf Sondermodul 0	Diese Fehler treten auf, wenn eine Operation oder die END-Anweisung ausgeführt wird. <ul style="list-style-type: none"> Prüfen Sie das Programm und den Inhalt der in den Anweisungen verwendeten Operanden. Prüfen Sie, ob die Pufferspeicherbereiche, auf die zugegriffen werden soll, im Sondermodul existieren. Prüfen Sie, ob die Sondermodule korrekt angeschlossen sind.
		Bit 1	Fehler beim Zugriff auf Sondermodul 1	
		Bit 2	Fehler beim Zugriff auf Sondermodul 2	
		Bit 3	Fehler beim Zugriff auf Sondermodul 3	
		Bit 4	Fehler beim Zugriff auf Sondermodul 4	
		Bit 5	Fehler beim Zugriff auf Sondermodul 5	
		Bit 6	Fehler beim Zugriff auf Sondermodul 6	
		Bit 7	Fehler beim Zugriff auf Sondermodul 7	
Bit 8 bis Bit 15	—	—		

Tab. 10-15: Sondermodulfehler

A Technische Daten

A.1 Übersicht der Grundbefehle

Anweisung	Kontaktplansymbol	Bedeutung	Operanden	Programmschritte ^①	Referenz
LD		LADE; Beginn einer Verknüpfung mit Abfrage auf Signalzustand „1“	X, Y, M, S, T, C, D□.b ^②	1	Abschnitt 4.2
LDI		LADE NICHT; Beginn einer Verknüpfung mit Abfrage auf Signalzustand „0“			
OUT		AUSGABE; Ausgabe, Zuweisung eines Verknüpfungsergebnisses	Y, M, S, T, C, D□.b ^②	Y, M: 1 S, Sondermerker: 2 T: 3, C: 3 C(32 Bit): 5	Abschnitt 4.3
AND		UND; UND-Verknüpfung mit Abfrage auf Signalzustand „1“	X, Y, M, S, T, C, D□.b ^②	1	Abschnitt 4.4
ANI		UND Nicht; UND-Verknüpfung mit Abfrage auf Signalzustand „0“			
OR		ODER; ODER-Verknüpfung mit Abfrage auf Signalzustand „1“	X, Y, M, S, T, C, D□.b ^②	1	Abschnitt 4.5
ORI		ODER Nicht; ODER-Verknüpfung mit Abfrage auf Signalzustand „0“			
LDP		LADE; (gepulst) Beginn einer Verknüpfung mit Abfrage der ansteigenden Flanke	X, Y, M, S, T, C, D□.b ^②	2	Abschnitt 4.6
LDF		LADE; (gepulst) Beginn einer Verknüpfung mit Abfrage der abfallenden Flanke			
ANDP		UND; (gepulst) UND-Verknüpfung mit Abfrage der ansteigenden Flanke	X, Y, M, S, T, C, D□.b ^②	2	Abschnitt 4.7

Tab. A-1: Grundbefehlsübersicht (Teil 1)

① Die Anzahl der Programmschritte bei einer FX3G, FX3GC, FX3GE, FX3S, FX3U oder FX3UC ist im Abschnitt 4.1.2 angegeben.







② Nur bei FX3U und FX3UC

Anweisung	Kontaktplansymbol	Bedeutung	Operanden	Programmschritte ^①	Referenz
ANDF		UND; (gepulst) UND-Verknüpfung mit Abfrage der abfallenden Flanke	X, Y, M, S, T, C, D□.b ^②	2	Abschnitt 4.7
ORP		ODER; ODER-Verknüpfung mit Abfrage der ansteigenden Flanke	X, Y, M, S, T, C, D□.b ^②	2	Abschnitt 4.8
ORF		ODER; ODER-Verknüpfung mit Abfrage der abfallenden Flanke			
ANB		UND-Block; Koppelbefehl: Reihenschaltung von Parallelverknüpfungen	—	1	Abschnitt 4.9
ORB		ODER-Block; Koppelbefehl: Parallelschaltung von Reihenverknüpfungen	—	1	Abschnitt 4.10
MPS		Push Down Stack; Abspeichern eines Verknüpfungsergebnisses	—	1	Abschnitt 4.11
MRD		Read Down Stack; Lesen eines Verknüpfungsergebnisses	—	1	
MPP		Pop Up Stack; Lesen und Löschen des Verknüpfungsspeichers	—	1	
MC		Master Control; Setzen einer Kontrollbedingung	Y, M, keine Sondermerker	3	Abschnitt 4.12
MCR		Master Control Reset; Rücksetzen einer Kontrollbedingung	N	2	
SET		Setzen; Operanden setzen	Y, M, S, D□.b ^②	Y, M: 1 S, Sondermerker: 2	Abschnitt 4.13
RST		Rücksetzen; Operanden rücksetzen	Y, M, S, D, V, Z, T, C, D□.b ^②	Y, M: 1 D, V, Z, Sondermerker: 3 T, C: 2	

Tab. A-2: Grundbefehlsübersicht (Teil 2)

① Die Anzahl der Programmschritte bei einer FX3G, FX3GC, FX3GE, FX3S, FX3U oder FX3UC ist im Abschnitt 4.1.2 angegeben.

② Nur bei FX3U und FX3UC

Anweisung	Kontaktplansymbol	Bedeutung	Operanden	Programmschritte ^①	Referenz
PLS		Impulserzeugung; Erzeugen eines einmaligen Impulses bei ansteigender Flanke	Y, M	2	Abschnitt 4.14
PLF		Impulserzeugung; Erzeugen eines einmaligen Impulses bei abfallender Flanke			
INV		Inversion; Umkehrung von Verarbeitungsergebnissen	—	1	Abschnitt 4.15
MEP		Impulserzeugung; Erzeugen eines einmaligen Impulses, wenn das Verknüpfungsergebnis „1“ wird	—	1	Abschnitt 4.16
MEF		Impulserzeugung; Erzeugen eines einmaligen Impulses, wenn das Verknüpfungsergebnis „1“ wird	—	1	
NOP	—	Leerzeile; Leerzeile ohne Funktion	—	1	Abschnitt 4.17
END		Ende; SPS-Programmende	—	1	Abschnitt 4.18

Tab. A-3: Grundbefehlsübersicht (Teil 3)

① Die Anzahl der Programmschritte bei einer FX3G, FX3GC, FX3GE, FX3S, FX3U oder FX3UC ist im Abschnitt 4.1.2 angegeben.

A.2 Allgemeine Systemdaten MELSEC FX1S

Merkmal	Technische Daten
Programmbearbeitung	Zyklische Abarbeitung des gespeicherten Programms
Ein-/Ausgangsbearbeitung	Prozessabbildverarbeitung Direktverarbeitende Anweisung vorhanden Eingangsfiter von 0 bis 15 ms einstellbar
Programmiersprache	Anweisungsliste und Kontaktplan nach DIN 19239 Schrittanweisungen
Ausführungszeiten der Anweisungen	Grundbefehl: 0,55 bis 0,7 μ s Applikationsanweisungen: siehe Anhang B
Programmkapazität	2k-Schritte, EEPROM-Baustein
Anzahl der Anweisungen	Grundbefehlssatz: 29 Schrittsteueranweisungen: 2 Applikationsanweisungen: 85

Tab. A-4: Allgemeine Systemdaten MELSEC FX1S

A.3 Operanden MELSEC FX1S

Merkmal		Technische Daten			
Ein-/Ausgänge	FX1S-10MR-ES/UL FX1S-10MR-DS FX1S-10MT-DSS	X0–X5	6 Eingänge		
		Y0–Y3	4 Ausgänge		
	FX1S-14MR-ES/UL FX1S-14MR-DS FX1S-14MT-DSS	X0–X7	8 Eingänge		
		Y0–Y5	6 Ausgänge		
	FX1S-20MR-ES/UL FX1S-20MR-DS FX1S-20MT-DSS	X0–X13	12 Eingänge		
		Y0–Y7	8 Ausgänge		
	FX1S-30MR-ES FX1S-30MR-DS FX1S-30MT-DSS	X0–X17	16 Eingänge		
		Y0–Y15	14 Ausgänge		
Merker	Merker	M0–M383	384 Adressen		
	Latch-Merker	M384–M511	Istwert im EEPROM gespeichert	128 Adressen	
	Sondermerker	M8000–M8254	256 Adressen		
Schrittstatus	Initialisierung	S0–S9	10 Adressen		
	Allgemein	S10–S127	118 Adressen		
Timer	100 ms	0,1–3 276,7s	T0–T62	63 Adressen	
	10 ms	0,01–327,67s	T32–T62	31 Adressen, wenn M8028 aktiviert	
	1 ms	0,001–32,767s	T63	1 Adresse	
Counter	Aufwärtszählend	16 Bit +1 bis +32 767	Allgemein	C0–C15	16 Adressen
			Istwert im EEPROM gespeichert	C16–C31	16 Adressen

Tab. A-5: Operanden MELSEC FX1S (1)

Merkmal		Technische Daten			
High-Speed-Counter	1-Phasen-Counter ohne Start und Reset, Auf-/Abwärtszählend	32 Bit	Istwert im EEPROM gespeichert	C235–C240	6 Zähler
	1-Phasen-Counter mit Start und Reset, Auf-/Abwärtszählend	32 Bit		C241–C245	5 Zähler
	2-Phasen-Counter, Auf-/Abwärtszählend	32 Bit		C246–C250	5 Zähler
	A/B-Phasen-Counter	32 Bit		C251–C255	5 Zähler
Register	Datenregister	16 Bit	Allgemein	D0–D127	128 Adressen
			Istwert im EEPROM gespeichert	D128–D255	128 Adressen
	File-Register	Festlegung durch Parameter in 3 Blöcken zu 500 Programmschritten		D1000–D2499	1500 Adressen
	Sonderregister	16 Bit	D8000 – D8255	256 Adressen	
	Extern veränderbare Register	16 Bit, Wertebereich von 0 bis 255 durch VR1 und VR2 einstellbar		D8030, D8031	2 Adressen
	Index-Register	16 Bit	V, Z	16 Adressen	
Pointer	Pointer Sprunganweisung			P0–P63	64 Adressen
	Interrupt-Pointer	Interrupt-Eingänge: X0 bis X3		I00*–I130*	6 Adressen
Nesting	Programmverzweigung, Hauptkontakt			N0–N7	8 Adressen
Konstanten	Dezimal	16 Bit		-32 768 bis +32 767	
		32 Bit		-2 147 483 648 bis +2 147 438 647	
	Hexdezimal	16 Bit		0 bis FFFF _H	
		32 Bit		0 bis FFFFFFFF _H	

Tab. A-6: Operanden MELSEC FX1S (2)

A.4 Allgemeine Systemdaten MELSEC FX1N

Merkmal	Technische Daten
Programmabarbeitung	Zyklische Abarbeitung des gespeicherten Programms
Ein-/Ausgangsbearbeitung	Prozeßabbildverarbeitung Direkverarbeitende Anweisungen vorhanden Eingangsfiler von 0 bis 15 ms einstellbar
Programmiersprache	Anweisungsliste und Kontaktplan nach DIN 19239 Schrittanweisungen
Ausführungszeiten der Anweisungen	Grundbefehl: 0,55 bis 0,7 µs Applikationsanweisungen: siehe Anhang B
Programmkapazität	8k-Schritte, EEPROM-Baustein
Anzahl der Anweisungen	Grundbefehlssatz: 29 Schrittsteueranweisungen: 2 Applikationsanweisungen: 89

Tab. A-7: Allgemeine Systemdaten MELSEC FX1N

A.5 Operanden MELSEC FX1N

Merkmal	Technische Daten				
Ein-/Ausgänge	FX1N-□□□-MR-DS FX1N-□□□-MR-ES/UL FX1N-□□□-MT-ESS/UL FX1N-□□□-MT-DSS	Die maximale E/A-Hardwarekonfiguration beträgt 128 Ein-/Ausgangsadressen in Summe. Mit der Software können maximal 128 Ein- und 128 Ausgänge adressiert werden.			
Merker	Merker	M0–M383		384 Adressen	
	Latch-Merker	M384–M1535	Istwert im EEPROM gespeichert	1152 Adressen	
	Sondermerker	M8000–M8255		256 Adressen	
Schrittstatus	Initialisierung	S0–S9		10 Adressen	
	Allgemein	S10–S999		990 Adressen	
Timer	100 ms	0,1–3276,7s	T0–T199	200 Adressen	
	10 ms	0,01–327,67s	T200–T245	46 Adressen	
	1 ms	0,001–32,767	T246–T249	4 Adressen	
	100 ms, remanent	0–25,5s	T250–T255	6 Adressen	
Counter	Aufwärtszählend	16 Bit +1 bis +32 767	Allgemein	C0–C15	16 Adressen
			Istwert im EEPROM gespeichert	C16–199	184 Adressen
	Auf-/Abwärtszählend	32 Bit	Allgemein	C200–C219	16 Adressen
			Istwert im EEPROM gespeichert	C220–C234	15 Adressen

Tab. A-8: Operanden MELSEC FX1N (1)

Merkmal		Technische Daten			
High-Speed-Counter	1-Phasen-Counter ohne Start und Reset, Auf-/Abwärtszählend	32 Bit	Istwert im EEPROM gespeichert	C235–C240	6 Adressen
	1-Phasen-Counter mit Start und Reset, Auf-/Abwärtszählend	32 Bit		C241–C245	5 Adressen
	2-Phasen-Counter, Auf-/Abwärtszählend	32 Bit		C246–C250	5 Adressen
	A/B-Phasen-Counter	32 Bit		C251–C255	5 Adressen
Register	Datenregister	16 Bit	Allgemein	D0–D127 D1000–D7999	7128 Adressen
			Istwert im EEPROM gespeichert	D128–D255	128 Adressen
	File-Register	16 Bit	Festlegung durch Parameter in 14 Blöcken zu 500 Programmschritten	D1000–D7999	7000 Adressen
	Sonderregister	16 Bit		D8000–D8255	256 Adressen
	Extern veränderbare Register	16 Bit, Werte von 0 bis 255 durch VR1 und VR2 einstellbar		D8030, D8031	2 Adressen
	Index-Register	16 Bit		V, Z	16 Adressen
Pointer	Pointer Sprung-Anweisung			P0 – P127	128 Adressen
	Interrupt-Pointer	Interrupt-Eingänge: X0 – X3		I00*–I130*	6 Adressen
Nesting	Programmverzweigung, Hauptkontakt			N0–N7	8 Adressen
Konstanten	Dezimal	16 Bit		-32 768 bis +32 767	
		32 Bit		-2 147 483 648 bis +2 147 438 647	
	Hexdezimal	16 Bit		0 bis FFFF _H	
		32 Bit		0 bis FFFFFFFF _H	

Tab. A-9: Operanden MELSEC FX1N (2)

A.6 Applikationsanweisungen MELSEC FX1S/FX1N

Einteilung	Anweisung	FNC	Bedeutung	Abschnitt
Programmablaufanweisungen	CJ	00	Sprung innerhalb eines Programms	6.2.1
	CALL	01	Aufruf eines Unterprogramms	6.2.2
	SRET	02	Ende eines Unterprogramms	6.2.3
	IRET	03	Interrupt-Programm abschließen	6.2.4
	EI	04	Interrupt-Programm aktivieren	
	DI	05	Interrupt-Programm deaktivieren	
	FEND	06	Ende eines Programmbereichs	6.2.5
	WDT	07	Watch-Dog-Timer auffrischen	6.2.6
	FOR	08	Anfang einer Programmwiederholung	6.2.7
	NEXT	09	Ende einer Programmwiederholung	
Vergleichs- und Transferanweisungen	CMP	10	Numerische Daten vergleichen	6.3.1
	ZCP	11	Numerische Datenbereiche vergleichen	6.3.2
	MOV	12	Datentransfer	6.3.3
	BMOV	15	Block-Datentransfer	6.3.6
	BCD	18	BCD-Konvertierung	6.3.9
	BIN	19	Binär-Konvertierung	6.3.10
Arithmetische Anweisungen	ADD	20	Addition numerischer Daten	6.4.1
	SUB	21	Subtraktion numerischer Daten	6.4.2
	MUL	22	Multiplikation numerischer Daten	6.4.3
	DIV	23	Division numerischer Daten	6.4.4
	INC	24	Inkrementieren	6.4.5
	DEC	25	Dekrementieren	6.4.6
	WAND	26	Logische UND-Verknüpfung	6.4.7
	WOR	27	Logische ODER-Verknüpfung	6.4.8
	WXOR	28	Logische Exklusiv-ODER-Verknüpfung	6.4.9
Verschiebeanweisung	SFTR	34	Binäre Daten bitweise verschieben, rechts	6.5.5
	SFTL	35	Binäre Daten bitweise verschieben, links	
	SFWR	38	Schreiben in einen FIFO-Speicher	6.5.8
	SFRD	39	Schreiben aus einem FIFO-Speicher	6.5.9
Datenoperationen	ZRST	40	Operandenbereiche zurücksetzen	6.6.1
	DECO	41	Daten decodieren	6.6.2
	ENCO	42	Daten codieren	6.6.3

Tab. A-10: Übersicht der Applikationsanweisungen FX1S/FX1N (1)

Einteilung	Anweisung	FNC	Bedeutung	Abschnitt
High-Speed-Anweisungen	REF	50	Ein- und Ausgänge auffrischen	6.7.1
	MTR	52	Einlesen einer Matrix	6.7.3
	DHSCS	53	Setzen durch High-Speed-Counter	6.7.4
	DHSCR	54	Rücksetzen durch High-Speed-Counter	
	SPD	56	Geschwindigkeitserkennung	6.7.6
	PLSY	57	Impulsausgabe einer definierten Anzahl von Impulsen	6.7.7
	PWM	58	Impulsausgabe mit Impulsweitenmodulation	6.7.8
	PLSR	59	Ausgabe einer bestimmten Anzahl von Impulsen	6.7.9
Anwendungsbezogene Anweisungen	IST	60	Schrittstatus initialisieren	6.8.1
	ABSD	62	Absoluter Counter-Vergleich	6.8.3
	INCD	63	Inkrementaler Counter-Vergleich	6.8.4
	ALT	66	Flip-Flop-Funktion	6.8.6
	RAMP	67	Rampenfunktion	6.8.7

Tab. A-11: Übersicht der Applikationsanweisungen FX1S/FX1N (2)

HINWEIS

| Die Applikationsanweisungen FNC 70 bis FNC 305 werden im Kapitel 7 beschrieben.

A.7 Allgemeine Systemdaten MELSEC FX2N/FX2NC

Merkmal	Technische Daten
Programmabarbeitung	Zyklische Abarbeitung des gespeicherten Programms
Ein-/Ausgangsbearbeitung	Prozeßabbildverarbeitung Direkverarbeitende Anweisungen vorhanden Eingangsfiler von 0 bis 15 ms einstellbar
Programmiersprache	SPS-Befehlsvorrat nach DIN 19239
Ausführungszeiten der Anweisungen	Grundbefehl: 0,08 µs Applikationsanweisung: siehe Anhang B
Programmkapazität	8k-Schritte: interner RAM-Speicher 16k-Schritte: RAM-, EEPROM-Kassette (optional)
Anzahl der Anweisungen	Grundbefehlssatz: 20 STL-Anweisung: 2 Applikationsanweisungen: 125

Tab. A-12: Allgemeine Systemdaten MELSEC FX2N und FX2NC

A.8 Operanden MELSEC FX2N/FX2NC

Merkmal	Technische Daten				
Ein-/Ausgänge	FX2N-□□□-MR-DS FX2N-□□□-MR-ES/UL FX2N-□□□-MT-ESS/UL FX2N-□□□-MT-DSS	Die maximale E/A-Hardwarekonfiguration beträgt 255 Ein-/Ausgangsadressen in Summe. Mit der Software können maximal 255 Ein- und 255 Ausgänge adressiert werden.			
Merker	Merker	M0–M3071	3072 Adressen		
	Latch-Merker	M500–M3071	2572 Adressen (anteilig)		
	Sondermerker	M8000–M8255	256 Adressen		
Schrittstatus	Initialisierung	S0–S9	10 Adressen (anteilig)		
	Allgemein	S0–S999	1000 Adressen		
	Latch-Merker	S500–S999	500 Adressen (anteilig)		
	Fehlermerker	S900–S999	100 Adressen		
Timer	100 ms	0,1–3 276,7 s	T0–T199	200 Adressen	
	10 ms	0,01–327,67 s	T200–T245	46 Adressen	
	1 ms (remanent)	0,001–32.767 s	T246–T249	4 Adressen	
	100 ms (remanent)	0,1–3276,7 s	T250–T255	6 Adressen	
Counter	Aufwärtszählend 16 Bit	+1 bis +32 767	Allgemein	C0–C199	200 Adressen
			Istwert im EEPROM gespeichert	C100–C199	100 Adressen (anteilig)
	Aufwärtszählend 32 Bit	+1 bis +2147483647	Allgemein	C200–C234	35 Adressen
			Istwert im EEPROM gespeichert	C219–C234	15 Adressen (anteilig)

Tab. A-13: Operanden MELSEC FX2N und FX2NC (1)

Merkmal		Technische Daten			
High-Speed-Counter	1-Phasen-Counter	-2147483648 bis +2147483647	Istwert im EEPROM gespeichert. Zählfrequenz aller Counter ≤ 20kHz	C235–C240	6 Adressen
	1-Phasen-Counter mit Start- und Reset-Eingang			C241–C245	5 Adressen
	2-Phasen-Counter			C246–C250	5 Adressen
	A/B-Phasen-Counter			C251–C255	5 Adressen
Register	Datenregister	16 Bit	Allgemein	D0–D7999	8000 Adressen
			Latch	D200–D7999	7800 Adressen (anteilig)
	File-Register	16 Bit	Festlegung durch Parameter in 14 Blöcken zu 500 Programmschritten	D1000–D7999	7000 Adressen
	Sonderregister	16 Bit		D8000–D8255	256 Adressen
	Index-Register	16 Bit		V0–V7, Z0–Z7	16 Adressen
Pointer	Pointer Sprung- anweisung			P0–P63	128 Adressen
	Interrupt-Pointer □ =1 (ansteigende Flanke) □ =0 (abfallende Flanke) **= Zeit in ms	Interrupt-Eingänge: X0 – X3		I00□–I50□	6 Adressen
		Interrupt-Timer		I6**–I8**	3 Adressen
	Interrupt-Counter			I010–I060	6 Adressen
Nesting	Programmverzweigung, Hauptkontakt			N0–N7	8 Adressen
Konstanten	Dezimal	16 Bit		-32 768 bis +32 767	
		32 Bit		-2 147 483 648 bis +2 147 438 647	
	Hexdezimal	16 Bit		0 bis FFFF _H	
		32 Bit		0 bis FFFFFFFF _H	

Tab. A-14: Operanden MELSEC FX2N und FX2NC (2)

A.9 Applikationsanweisungen MELSEC FX2N/FX2NC

Einteilung	Anweisung	FNC	Bedeutung	Abschnitt
Programmablaufanweisungen	CJ	00	Sprung innerhalb eines Programms	6.2.1
	CALL	01	Aufruf eines Unterprogramms	6.2.2
	SRET	02	Ende eines Unterprogramms	6.2.3
	IRET	03	Interrupt-Programm abschließen	6.2.4
	EI	04	Interrupt-Programm aktivieren	
	DI	05	Interrupt-Programm deaktivieren	
	FEND	06	Ende eines Programmbereichs	6.2.5
	WDT	07	Watch-Dog-Timer auffrischen	6.2.6
	FOR	08	Anfang einer Programmwiederholung	6.2.7
	NEXT	09	Ende einer Programmwiederholung	
Vergleichs- und Transferanweisungen	CMP	10	Numerische Daten vergleichen	6.3.1
	ZCP	11	Numerische Datenbereiche vergleichen	6.3.2
	MOV	12	Datentransfer	6.3.3
	SMOV	13	Shift-Transfer	6.3.4
	CML	14	Kopieren und invertieren	6.3.5
	BMOV	15	Block-Transfer	6.3.6
	FMOV	16	Transfer von gleichen Daten	6.3.7
	XCH	17	Austausch von Daten	6.3.8
	BCD	18	BCD-Konvertierung	6.3.9
	BIN	19	Binär-Konvertierung	6.3.10
Arithmetische Anweisungen	ADD	20	Addition numerischer Daten	6.4.1
	SUB	21	Subtraktion numerischer Daten	6.4.2
	MUL	22	Multiplikation numerischer Daten	6.4.3
	DIV	23	Division numerischer Daten	6.4.4
	INC	24	Inkrementieren	6.4.5
	DEC	25	Dekrementieren	6.4.6
	AND	26	Logische UND-Verknüpfung	6.4.7
	OR	27	Logische ODER-Verknüpfung	6.4.8
	XOR	28	Logische Exklusiv-ODER-Verknüpfung	6.4.9
	NEG	29	Negation von Daten	6.4.10
Verschiebeanweisungen	ROR	30	Rotation nach rechts	6.5.1
	ROL	31	Rotation nach links	6.5.2
	RCR	32	Rotieren von Bits nach rechts	6.5.3
	RCL	33	Rotieren von Bits nach links	6.5.4
	SFTR	34	Binäre Daten bitweise verschieben, rechts	6.5.5
	SFTL	35	Binäre Daten bitweise verschieben, links	
	WSFR	36	Daten wortweise nach rechts verschieben	6.5.6
	WSFL	37	Daten wortweise nach links verschieben	6.5.7
	SFWR	38	Schreiben in einen FIFO-Speicher	6.5.8
	SFRD	39	Lesen aus einem FIFO-Speicher	6.5.9

Tab. A-15: Gesamtübersicht der Applikationsanweisungen FX2N und FX2NC (1)

Einteilung	Anweisung	FNC	Bedeutung	Abschnitt
Datenoperationen	ZRST	40	Operandenbereiche zurücksetzen	6.6.1
	DECO	41	Daten decodieren	6.6.2
	ENCO	42	Daten codieren	6.6.3
	SUM	43	Ermittlung gesetzter Bits	6.6.4
	BON	44	Überprüfen eines Bits	6.6.5
	MEAN	45	Ermittlung von Durchschnittswerten	6.6.6
	ANS	46	Starten eines Zeitintervalls	6.6.7
	ANR	47	Rücksetzen von Anzeige-Bits	6.6.8
	SQR	48	Ermittlung der Quadratwurzel	6.6.9
	FLT	49	Umwandlung des Zahlenformats	6.6.10
High-Speed-Anweisungen	REF	50	Ein- und Ausgänge auffrischen	6.7.1
	REFF	51	Einstellen der Eingangsfiler	6.7.2
	MTR	52	Einlesen einer Matrix (MTR)	6.7.3
	DHSCS	53	Setzen durch High-Speed-Counter	6.7.4
	DHSCR	54	Rücksetzen durch High-Speed-Counter	6.7.4
	DHSZ	55	Bereichsvergleich	6.7.5
	SPD	56	Geschwindigkeitserkennung	6.7.6
	PLSY	57	Impulsausgabe einer definierten Anzahl von Impulsen	6.7.7
	PWM	58	Impulsausgabe mit Impulsweitenmodulation	6.7.8
	PLSR	59	Ausgabe einer bestimmten Anzahl von Impulsen	6.7.9
Anwendungsbezogene Anweisungen	IST	60	Schrittstatus initialisieren	6.8.1
	SER	61	Suchanweisung	6.8.2
	ABSD	62	Absoluter Counter-Vergleich	6.8.3
	INCD	63	Inkrementaler Counter-Vergleich	6.8.4
	TTMR	64	Teaching-Timer	6.8.5
	STMR	65	Sonder-Timer	6.8.6
	ALT	66	Flip-Flop-Funktion	6.8.7
	RAMP	67	Rampenfunktion	6.8.8
	ROTC	68	Rundtisch-Positionierung	6.8.9
	SORT	69	Sortieranweisung	6.8.10

Tab. A-16: Gesamtübersicht der Applikationsanweisungen FX2N und FX2NC (2)

HINWEIS

| Die Applikationsanweisungen FNC 70 bis FNC 305 werden im Kapitel 7 beschrieben.

A.10 Allgemeine Systemdaten MELSEC FX3G/3GC/3GE

Merkmal	Technische Daten
Programmabarbeitung	Zyklische Abarbeitung des gespeicherten Programms
Ein-/Ausgangsbearbeitung	Prozeßabbildverarbeitung Direkverarbeitende Anweisungen vorhanden Eingangsfiler von 0 bis 15 ms einstellbar
Programmiersprache	SPS-Befehlsvorrat nach DIN 19239
Ausführungszeiten der Anweisungen	Grundbefehl: 0,21 µs pro Anweisung im Standard-Modus ^① ; 0,42 µs pro Anweisung im erweiterten Modus ^① Applikationsanweisung: siehe Anhang B
Programmkapazität	32k-Schritte: interner EEPROM-Speicher 32k-Schritte: EEPROM-Kassette (optional)
Anzahl der Anweisungen	Grundbefehlssatz: 29 STL-Anweisungen: 2 Applikationsanweisungen: 124 (122 bei FX3GC)

Tab. A-17: Allgemeine Systemdaten MELSEC FX3G/FX3GC/FX3GE

^① Der Standard-Modus ist ausgewählt, wenn die Programmspeicherkapazität durch Parameter auf maximal 16000 Programmschritte eingestellt ist.

Der erweiterte Modus ist ausgewählt, wenn die Programmspeicherkapazität durch Parameter auf 16001 oder mehr Programmschritte eingestellt ist.

A.11 Operanden MELSEC FX3G/FX3GC/FX3GE

Merkmal	Technische Daten				
Ein-/Ausgänge	Es können maximal 128 Ein- und 128 Ausgänge im Grundgerät und in Erweiterungsmodulen adressiert werden (X000 bis X177 und Y000 bis Y177). Die Summe der Eingänge und der Ausgänge im Grund- und Erweiterungsgeräten darf jedoch 128 nicht überschreiten. Zusätzlich können 128 E/As in einem CC-Link-Netzwerk angesprochen werden. Die Summe der Eingänge und der Ausgänge im Grund- und Erweiterungsgeräten und der Ein- und Ausgänge in einem Netzwerk darf 256 nicht überschreiten.				
Merker	Merker	M0–M383	384 Adressen		
	Latch-Merker ^①	M384–M1535	1152 Adressen		
	Merker ^②	M1536–M7679	6144 Adressen		
	Sondermerker	M8000–M8511	512 Adressen		
Schrittstatus	Initialisierung ^①	S0–S9	10 Adressen		
	Latch-Merker ^①	S10–S999	990 Adressen		
	Merker ^②	S1000–S4095	3096 Adressen		
Timer ^③	100 ms	0,1–3276,7 s	T0–T199	200 Adressen	
	10 ms	0,01–327,67 s	T200–T245	46 Adressen	
	1 ms (remanent) ^①	0,001–32,767 s	T246–T249	4 Adressen	
	100 ms (remanent) ^①	0,1–3276,7 s	T250–T255	6 Adressen	
	1 ms	0,001–32,767 s	T256–T319	64 Adressen	
Counter	Aufwärtszählend 16 Bit	Zählbereich: +1 bis +32 767	Allgemein	C0–C15	16 Adressen
			Istwert im EEPROM gespeichert	C16–C199	184 Adressen
	Auf- und abwärtszählend 32 Bit	Zählbereich: -2147483648 bis +2147483647	Allgemein	C200–C219	20 Adressen
			Istwert im EEPROM gespeichert	C220–C234	15 Adressen

Tab. A-18: Operanden MELSEC FX3G/FX3GC/FX3GE (1)

Merkmal		Technische Daten			
High-Speed-Counter	1-Phasen-Counter mit einem Zählengang	Zählbereich: -2147483648 bis +2147483647	Istwert im EEPROM gespeichert.	C235–C245	11 Adressen
	1-Phasen-Counter mit zwei Zählengängen			C246–C250	5 Adressen
	2-Phasen-Counter			C251–C255	5 Adressen
Register (Jeweils 2 Register können zu einem 32-Bit-Register zusammengefasst werden.)	Datenregister	16 Bit	Allgemein	D0–D127	128 Adressen
			Latch ^①	D128–D1099	972 Adressen
			Allgemein ^④	D1100–D7999	972 Adressen
	File-Register ^①	16 Bit	Festlegung durch Parameter in Blöcken zu je 500 Adressen	D1000–D7999	max. 7000 Adressen (anteilig)
	Sonderregister	16 Bit		D8000–D8511	512 Adressen
Indexregister	16 Bit		V0–V7, Z0–Z7	16 Adressen	
Erweiterte Register ^④		16 Bit		R0 bis R23999	24000 Adressen
Erweiterte File-Register ^⑤		16 Bit		ER0 bis ER23999	24000 Adressen
Pointer	Pointer für Sprunganweisungen			P0–P2047	2048 Adressen
	Interrupt-Pointer □ =1 (ansteigende Flanke) □ =0 (abfallende Flanke) **= Zeit in ms	Interrupt-Eingänge: X0–X5		I00□–I50□	6 Adressen
		Interrupt-Timer		I6**–I8**	3 Adressen
Nesting	Programmverzweigung, Hauptkontakt			N0–N7	8 Adressen
Konstanten	Dezimal	16 Bit		-32 768 bis +32 767	
		32 Bit		-2 147 483 648 bis +2 147 438 647	
	Hexdezimal	16 Bit		0 bis FFFF _H	
		32 Bit		0 bis FFFFFFFF _H	
	Gleitkommazahl ^⑥	32 Bit		-1,0 x 2 ¹²⁸ bis -1,0 x 2 ⁻¹²⁶ 0 1,0 x 2 ⁻¹²⁶ bis -1,0 x 2 ⁺¹²⁸	

Tab. A-19: Operanden MELSEC FX3G/FX3GC/FX3GE (2)

- ① Die Operandenzustände bzw. -inhalte werden im EEPROM gespeichert.
- ② Wenn die optionale Batterie installiert ist, kann diesen Merkern in den SPS-Parametern die Funktion von Latch-Merkern zugewiesen werden. Sie werden dann durch die Batterie gepuffert.
- ③ Alle Timer arbeiten als Einschaltverzögerung.
- ④ Wenn die optionale Batterie installiert ist, kann diesen Registern in den SPS-Parametern die Funktion von Latch-Registern zugewiesen werden. Sie werden dann durch die Batterie gepuffert.
- ⑤ Die Operandeninhalte werden im EEPROM des Grundgeräts oder – bei installierter Speicherkassette – im EEPROM der Speicherkassette gespeichert.
- ⑥ Bei FX3G-Grundgeräten ab der Version 1.10.

A.12 Applikationsanweisungen MELSEC FX3G/3GC/3GE

Einteilung	Anweisung	FNC	Bedeutung	Abschnitt
Programmablaufanweisungen	CJ	00	Sprung innerhalb eines Programms	6.2.1
	CALL	01	Aufruf eines Unterprogramms	6.2.2
	SRET	02	Ende eines Unterprogramms	6.2.3
	IRET	03	Interrupt-Programm abschließen	6.2.4
	EI	04	Interrupt-Programm aktivieren	
	DI	05	Interrupt-Programm deaktivieren	
	FEND	06	Ende eines Programmbereichs	6.2.5
	WDT	07	Watch-Dog-Timer auffrischen	6.2.6
	FOR	08	Anfang einer Programmwiederholung	6.2.7
	NEXT	09	Ende einer Programmwiederholung	
Vergleichs- und Transferanweisungen	CMP	10	Numerische Daten vergleichen	6.3.1
	ZCP	11	Numerische Datenbereiche vergleichen	6.3.2
	MOV	12	Datentransfer	6.3.3
	SMOV	13	Shift-Transfer	6.3.4
	CML	14	Kopieren und invertieren	6.3.5
	BMOV	15	Block-Transfer	6.3.6
	FMOV	16	Transfer von gleichen Daten	6.3.7
	BCD	18	BCD-Konvertierung	6.3.9
	BIN	19	Binär-Konvertierung	6.3.10
Arithmetische Anweisungen	ADD	20	Addition numerischer Daten	6.4.1
	SUB	21	Subtraktion numerischer Daten	6.4.2
	MUL	22	Multiplikation numerischer Daten	6.4.3
	DIV	23	Division numerischer Daten	6.4.4
	INC	24	Inkrementieren	6.4.5
	DEC	25	Dekrementieren	6.4.6
	AND	26	Logische UND-Verknüpfung	6.4.7
	OR	27	Logische ODER-Verknüpfung	6.4.8
	XOR	28	Logische Exklusiv-ODER-Verknüpfung	6.4.9
Verschiebeanweisungen	ROR	30	Rotation nach rechts	6.5.1
	ROL	31	Rotation nach links	6.5.2
	SFTR	34	Binäre Daten bitweise verschieben, rechts	6.5.5
	SFTL	35	Binäre Daten bitweise verschieben, links	
	WSFR	36	Daten wortweise nach rechts verschieben	6.5.6
	WSFL	37	Daten wortweise nach links verschieben	6.5.7
	SFWR	38	Schreiben in einen FIFO-Speicher	6.5.8
	SFRD	39	Lesen aus einem FIFO-Speicher	6.5.9

Tab. A-20: Übersicht der Applikationsanweisungen für MELSEC FX3G/FX3GC/FX3GE (1)

Einteilung	Anweisung	FNC	Bedeutung	Abschnitt
Daten- operationen	ZRST	40	Operandenbereiche zurücksetzen	6.6.1
	DECO	41	Daten decodieren	6.6.2
	ENCO	42	Daten codieren	6.6.3
	SUM	43	Ermittlung gesetzter Bits	6.6.4
	BON	44	Überprüfen eines Bits	6.6.5
	MEAN	45	Ermittlung von Durchschnittswerten	6.6.6
	ANS	46	Starten eines Zeitintervalls	6.6.7
	ANR	47	Rücksetzen von Anzeige-Bits	6.6.8
	FLT	49	Umwandlung des Zahlenformats	6.6.10
High-Speed- Anweisungen	REF	50	Ein- und Ausgänge auffrischen	6.7.1
	MTR	52	Einlesen einer Matrix (MTR)	6.7.3
	DHSCS	53	Setzen durch High-Speed-Counter	6.7.4
	DHSCR	54	Rücksetzen durch High-Speed-Counter	6.7.4
	DHSZ	55	Bereichsvergleich	6.7.5
	SPD	56	Geschwindigkeitserkennung	6.7.6
	PLSY	57	Impulsausgabe einer definierten Anzahl von Impulsen	6.7.7
	PWM	58	Impulsausgabe mit Impulsweitenmodulation	6.7.8
	PLSR	59	Ausgabe einer bestimmten Anzahl von Impulsen	6.7.9
Anwendungs- bezogene Anweisungen	IST	60	Schrittstatus initialisieren	6.8.1
	SER	61	Suchanweisung	6.8.2
	ABSD	62	Absoluter Counter-Vergleich	6.8.3
	INCD	63	Inkrementaler Counter-Vergleich	6.8.4
	ALT	66	Flip-Flop-Funktion	6.8.7
	RAMP	67	Rampenfunktion	6.8.8

Tab. A-21: Übersicht der Applikationsanweisungen für MELSEC FX3G/FX3GC/FX3GE (2)

HINWEIS

Die Applikationsanweisungen FNC 70 bis FNC 305 werden im Kapitel 7 beschrieben.

A.13 Allgemeine Systemdaten MELSEC FX3S

Merkmal	Technische Daten
Programmabarbeitung	Zyklische Abarbeitung des gespeicherten Programms
Ein-/Ausgangsbearbeitung	Prozeßabbildverarbeitung Direktverarbeitende Anweisungen vorhanden Eingangsfiter von 0 bis 15 ms einstellbar
Programmiersprache	SPS-Befehlsvorrat nach DIN 19239
Ausführungszeiten der Anweisungen	Grundbefehl: 0,21 µs pro Anweisung Applikationsanweisung: siehe Anhang B
Programmkapazität	16k-Schritte: interner EEPROM-Speicher 32k-Schritte: EEPROM-Kassette (optional)
Anzahl der Anweisungen	Grundbefehlssatz: 29 STL-Anweisungen: 2 Applikationsanweisungen: 116

Tab. A-22: Allgemeine Systemdaten MELSEC FX3S

A.14 Operanden MELSEC FX3S

Merkmal	Technische Daten				
Ein-/Ausgänge	Abhängig vom Typ des Grundgeräts maximal 16 Eingänge (X000 bis X017) und maximal 14 Ausgänge (Y000 bis Y015). Nicht erweiterbar.				
Merker	Merker	M0–M383	384 Adressen		
	Latch-Merker ^①	M384–M511	128 Adressen		
	Merker	M512–M1535	1024 Adressen		
	Sondermerker	M8000–M8511	512 Adressen		
Schrittstatus	Initialisierung ^①	S0–S9	10 Adressen		
	Latch-Merker ^①	S10–S127	118 Adressen		
	Merker	S128–S255	128 Adressen		
Timer ^②	100 ms	0,1–3276,7 s	T0–T31	32 Adressen	
	100 ms/10 ms	0,1–3276,7 s / 0,01–327,67 s	T32–T62	31 Adressen Wenn M8038 gesetzt ist, arbeiten T32 bis T62 als 10-ms-Timer.	
	1 ms	0,001–32,767 s	T63–T127	65 Adressen	
	1 ms (remanent) ^①	0,001–32,767 s	T128–T131	4 Adressen	
	100 ms (remanent) ^①	0,1–3276,7 s	T132–T137	6 Adressen	
Counter	Aufwärtszählend 16 Bit	Zählbereich: +1 bis +32 767	Allgemein	C0–C15	16 Adressen
			Istwert im EEPROM gespeichert	C16–C31	16 Adressen
	Auf- und abwärtszählend 32 Bit	Zählbereich: -2147483648 bis +2147483647	Allgemein	C200–C234	35 Adressen
High-Speed-Counter	1-Phasen-Counter mit einem Zähleingang ^①	Zählbereich: -2147483648 bis +2147483647	Istwert im EEPROM gespeichert.	C235–C245	11 Adressen
	1-Phasen-Counter mit zwei Zähleingängen ^①			C246–C250	5 Adressen
	2-Phasen-Counter ^①			C251–C255	5 Adressen

Tab. A-23: Operanden MELSEC FX3S (1)

Merkmal		Technische Daten			
Register (Jeweils 2 Register können zu einem 32-Bit-Register zusammengefasst werden.)	Datenregister	16 Bit	Allgemein	D0–D127	128 Adressen
			Latch ^①	D128–D255	128 Adressen
			Allgemein	D256–D2999	2744 Adressen
	File-Register ^①	16 Bit	Festlegung durch Parameter in Blöcken zu je 500 Adressen	D1000–D2999	max. 2000 Adressen (anteilig)
	Sonderregister	16 Bit		D8000–D8511	512 Adressen
Indexregister	16 Bit		V0–V7, Z0–Z7	16 Adressen	
Pointer	Pointer für Sprunganweisungen			P0–P255	2048 Adressen
	Interrupt-Pointer □ =1 (ansteigende Flanke) □ =0 (abfallende Flanke) ** = Zeit in ms	Interrupt-Eingänge: X0 – X5		I00□–I50□	6 Adressen
	Interrupt-Timer			I6**–I8**	3 Adressen
Nesting	Programmverzweigung, Hauptkontakt			N0–N7	8 Adressen
Konstanten	Dezimal	16 Bit		-32 768 bis +32 767	
		32 Bit		-2 147 483 648 bis +2 147 438 647	
	Hexdezimal	16 Bit		0 bis FFFF _H	
		32 Bit		0 bis FFFFFFFF _H	
	Gleitkommazahl	32 Bit		-1,0 x 2 ¹²⁸ bis -1,0 x 2 ⁻¹²⁶ 0 1,0 x 2 ⁻¹²⁶ bis -1,0 x 2 ⁺¹²⁸	

Tab. A-24: Operanden MELSEC FX3S (2)

- ① Die Operandenzustände bzw. -inhalte werden im EEPROM gespeichert.
 ② Alle Timer arbeiten als Einschaltverzögerung.

A.15 Applikationsanweisungen MELSEC FX3S

Einteilung	Anweisung	FNC	Bedeutung	Abschnitt
Programmablaufanweisungen	CJ	00	Sprung innerhalb eines Programms	6.2.1
	CALL	01	Aufruf eines Unterprogramms	6.2.2
	SRET	02	Ende eines Unterprogramms	6.2.3
	IRET	03	Interrupt-Programm abschließen	6.2.4
	EI	04	Interrupt-Programm aktivieren	
	DI	05	Interrupt-Programm deaktivieren	
	FEND	06	Ende eines Programmbereichs	6.2.5
	WDT	07	Watch-Dog-Timer auffrischen	6.2.6
	FOR	08	Anfang einer Programmwiederholung	6.2.7
	NEXT	09	Ende einer Programmwiederholung	
Vergleichs- und Transferanweisungen	CMP	10	Numerische Daten vergleichen	6.3.1
	ZCP	11	Numerische Datenbereiche vergleichen	6.3.2
	MOV	12	Datentransfer	6.3.3
	SMOV	13	Shift-Transfer	6.3.4
	CML	14	Kopieren und invertieren	6.3.5
	BMOV	15	Block-Transfer	6.3.6
	FMOV	16	Transfer von gleichen Daten	6.3.7
	BCD	18	BCD-Konvertierung	6.3.9
	BIN	19	Binär-Konvertierung	6.3.10
Arithmetische Anweisungen	ADD	20	Addition numerischer Daten	6.4.1
	SUB	21	Subtraktion numerischer Daten	6.4.2
	MUL	22	Multiplikation numerischer Daten	6.4.3
	DIV	23	Division numerischer Daten	6.4.4
	INC	24	Inkrementieren	6.4.5
	DEC	25	Dekrementieren	6.4.6
	AND	26	Logische UND-Verknüpfung	6.4.7
	OR	27	Logische ODER-Verknüpfung	6.4.8
	XOR	28	Logische Exklusiv-ODER-Verknüpfung	6.4.9
Verschiebeanweisungen	ROR	30	Rotation nach rechts	6.5.1
	ROL	31	Rotation nach links	6.5.2
	SFTR	34	Binäre Daten bitweise verschieben, rechts	6.5.5
	SFTL	35	Binäre Daten bitweise verschieben, links	
	WSFR	36	Daten wortweise nach rechts verschieben	6.5.6
	WSFL	37	Daten wortweise nach links verschieben	6.5.7
	SFWR	38	Schreiben in einen FIFO-Speicher	6.5.8
	SFRD	39	Lesen aus einem FIFO-Speicher	6.5.9
	Datenoperationen	ZRST	40	Operandenbereiche zurücksetzen
DECO		41	Daten decodieren	6.6.2
ENCO		42	Daten codieren	6.6.3
SUM		43	Ermittlung gesetzter Bits	6.6.4
BON		44	Überprüfen eines Bits	6.6.5
MEAN		45	Ermittlung von Durchschnittswerten	6.6.6

Tab. A-25: Übersicht der Applikationsanweisungen für MELSEC FX3S (1)

Einteilung	Anweisung	FNC	Bedeutung	Abschnitt
High-Speed-Anweisungen	REF	50	Ein- und Ausgänge auffrischen	6.7.1
	MTR	52	Einlesen einer Matrix (MTR)	6.7.3
	DHSCS	53	Setzen durch High-Speed-Counter	6.7.4
	DHSCR	54	Rücksetzen durch High-Speed-Counter	6.7.4
	DHSZ	55	Bereichsvergleich	6.7.5
	SPD	56	Geschwindigkeitserkennung	6.7.6
	PLSY	57	Impulsausgabe einer definierten Anzahl von Impulsen	6.7.7
	PWM	58	Impulsausgabe mit Impulsweitenmodulation	6.7.8
	PLSR	59	Ausgabe einer bestimmten Anzahl von Impulsen	6.7.9
Anwendungsbezogene Anweisungen	IST	60	Schrittstatus initialisieren	6.8.1
	SER	61	Suchanweisung	6.8.2
	ABSD	62	Absoluter Counter-Vergleich	6.8.3
	INCD	63	Inkrementaler Counter-Vergleich	6.8.4
	ALT	66	Flip-Flop-Funktion	6.8.7
	RAMP	67	Rampenfunktion	6.8.8

Tab. A-26: Übersicht der Applikationsanweisungen für MELSEC FX3S (2)

HINWEIS

| Die Applikationsanweisungen FNC 70 bis FNC 305 werden im Kapitel 7 beschrieben.

A.16 Allgemeine Systemdaten MELSEC FX3U/FX3UC

Merkmal	Technische Daten
Programmabarbeitung	Zyklische Abarbeitung des gespeicherten Programms
Ein-/Ausgangsbearbeitung	Prozeßabbildverarbeitung Direkverarbeitende Anweisungen vorhanden Eingangsfiler von 0 bis 15 ms einstellbar
Programmiersprache	SPS-Befehlsvorrat nach DIN 19239
Ausführungszeiten der Anweisungen	Grundbefehl: 0,065 µs Applikationsanweisung: siehe Anhang B
Programmkapazität	64k-Schritte: interner RAM-Speicher 64k-Schritte: RAM-, EEPROM-Kassette (optional)
Anzahl der Anweisungen	Grundbefehlssatz: 27 (bis Version 2.30), 29 (ab Version 2.30) STL-Anweisung: 2 Applikationsanweisung: 218, 181 (bis Version 2.20), 209 (ab Version 2.20)

Tab. A-27: Allgemeine Systemdaten MELSEC FX3U/FX3UC

A.17 Operanden MELSEC FX3U/FX3UC

Merkmal	Technische Daten				
Ein-/Ausgänge	Es können maximal 248 Ein- und 248 Ausgänge im Grundgerät und in Erweiterungsmodulen adressiert werden (X000 bis X367 und Y000 bis Y367). Die Summe der Eingänge und der Ausgänge im Grund- und Erweiterungsgeräten darf jedoch 256 nicht überschreiten. Zusätzlich können 224 E/As in einem CC-Link-Netzwerk oder 248 E/As in einem ASI-Netzwerk angesprochen werden. Die Summe der Eingänge und der Ausgänge im Grund- und Erweiterungsgeräten und der Ein- und Ausgänge in einem Netzwerk darf 384 nicht überschreiten.				
Merker	Merker	M0–M7679	7680 Adressen		
	Latch-Merker	M500–M7679	7180 Adressen (anteilig)		
	Sondermerker	M8000–M8511	512 Adressen		
Schrittstatus	Initialisierung	S0–S9	10 Adressen (anteilig)		
	Allgemein	S10–S499	490 Adressen		
	Latch-Merker (variabel)	S500–S899	400 Adressen (anteilig)		
	Fehlermerker	S900–S999	100 Adressen		
	Latch-Merker (fest)	S1000–S4095	3096 Adressen		
Timer	100 ms	0,1–3276,7 s	T0–T191	192 Adressen	
	10 ms	0,01–327,67 s	T200–T245	46 Adressen	
	1 ms (remanent)	0,001–32,767 s	T246–T249	4 Adressen	
	100 ms (remanent)	0,1–3276,7 s	T250–T255	6 Adressen	
	1 ms	0,001–32,767 s	T256 – T511	256 Adressen	
Counter	Aufwärtszählend 16 Bit	Zählbereich: +1 bis +32 767	Allgemein	C0–C199	200 Adressen
			Istwert im EEPROM gespeichert	C100–C199	100 Adressen (anteilig)
	Auf- und Awärtszählend 32 Bit	Zählbereich: -2147483648 bis +2147483647	Allgemein	C200–C234	35 Adressen
			Istwert im EEPROM gespeichert	C219–C234	15 Adressen (anteilig)

Tab. A-28: Operanden MELSEC FX3U/FX3UC (1)

Merkmal		Technische Daten			
High-Speed-Counter	1-Phasen-Counter	Zählbereich: -2147483648 – +2147483647	Istwert im EEPROM gespeichert.	C235–C240	6 Adressen
	1-Phasen-Counter mit Start- und Reset-Eingang			C241–C245	5 Adressen
	2-Phasen-Counter			C246–C250	5 Adressen
	A/B-Phasen-Counter			C251–C255	5 Adressen
Register (Jeweils 2 Register können zu einem 32-Bit- Register zusammen- gefasst wer- den.)	Datenregister	16 Bit	Allgemein	D0–D7999	8000 Adressen
			Latch	D200–D7999	7800 Adressen (anteilig)
	File-Register	16 Bit	Festlegung durch Parame- ter in Blöcken zu je 500 Adressen	D512–D7999	7488 Adressen
	Sonderregister	16 Bit		D8000–D8511	512 Adressen
	Index-Register	16 Bit		V0–V7, Z0–Z7	16 Adressen
Erweiterte Register		16 Bit	Inhalte bleiben bei Spannungs- ausfall erhalten	R0 bis R32767	32768 Adressen
Erweiterte File-Register		16 Bit	Nur bei instal- lierter Speicher- kassette nutzbar	ER0 bis ER32767	32768 Adressen
Pointer	Pointer Sprung- anweisung			P0–P4095	4096 Adressen
	Interrupt-Pointer □ = 1 (ansteigende Flanke) □ = 0 (abfallende Flanke) ** = Zeit in ms	Interrupt-Eingänge: X0–X5		I00□–I50□	6 Adressen
		Interrupt-Timer		I6**–I8**	3 Adressen
	Interrupt-Counter		I010–I060	6 Adressen	
Nesting	Programmverzweigung, Hauptkontakt			N0–N7	8 Adressen
Konstanten	Dezimal	16 Bit		-32 768 bis +32 767	
		32 Bit		-2 147 483 648 bis +2 147 438 647	
	Hexdezimal	16 Bit		0 bis FFFF _H	
		32 Bit		0 bis FFFFFFFF _H	
	Gleitkommazahl	32 Bit		-1,0 x 2 ¹²⁸ bis -1,0 x 2 ⁻¹²⁶	
				0 1,0 x 2 ⁻¹²⁶ bis -1,0 x 2 ⁺¹²⁸	
	Zeichenfolgen	Zeichenfolgen werden im Programm durch Anführungsstriche gekenn- zeichnet (z. B. "MITSUBISHI") Bis zu 32 Zeichen, von denen jedes ein Byte belegt, können angegeben werden.			

Tab. A-29: Operanden MELSEC FX3U/FX3UC (2)

A.18 Applikationsanweisungen MELSEC FX3U/FX3UC

Einteilung	Anweisung	FNC	Bedeutung	Abschnitt
Programmablaufanweisungen	CJ	00	Sprung innerhalb eines Programms	6.2.1
	CALL	01	Aufruf eines Unterprogramms	6.2.2
	SRET	02	Ende eines Unterprogramms	6.2.3
	IRET	03	Interrupt-Programm abschließen	6.2.4
	EI	04	Interrupt-Programm aktivieren	
	DI	05	Interrupt-Programm deaktivieren	
	FEND	06	Ende eines Programmbereichs	6.2.5
	WDT	07	Watch-Dog-Timer auffrischen	6.2.6
	FOR	08	Anfang einer Programmwiederholung	6.2.7
	NEXT	09	Ende einer Programmwiederholung	
Vergleichs- und Transferanweisungen	CMP	10	Numerische Daten vergleichen	6.3.1
	ZCP	11	Numerische Datenbereiche vergleichen	6.3.2
	MOV	12	Datentransfer	6.3.3
	SMOV	13	Shift-Transfer	6.3.4
	CML	14	Kopieren und invertieren	6.3.5
	BMOV	15	Block-Transfer	6.3.6
	FMOV	16	Transfer von gleichen Daten	6.3.7
	XCH	17	Austausch von Daten	6.3.8
	BCD	18	BCD-Konvertierung	6.3.9
	BIN	19	Binär-Konvertierung	6.3.10
Arithmetische Anweisungen	ADD	20	Addition numerischer Daten	6.4.1
	SUB	21	Subtraktion numerischer Daten	6.4.2
	MUL	22	Multiplikation numerischer Daten	6.4.3
	DIV	23	Division numerischer Daten	6.4.4
	INC	24	Inkrementieren	6.4.5
	DEC	25	Dekrementieren	6.4.6
	AND	26	Logische UND-Verknüpfung	6.4.7
	OR	27	Logische ODER-Verknüpfung	6.4.8
	XOR	28	Logische Exklusiv-ODER-Verknüpfung	6.4.9
	NEG	29	Negation von Daten	6.4.10
Verschiebeanweisungen	ROR	30	Rotation nach rechts	6.5.1
	ROL	31	Rotation nach links	6.5.2
	RCR	32	Rotieren von Bits nach rechts	6.5.3
	RCL	33	Rotieren von Bits nach links	6.5.4
	SFTR	34	Binäre Daten bitweise verschieben, rechts	6.5.5
	SFTL	35	Binäre Daten bitweise verschieben, links	6.5.5
	WSFR	36	Daten wortweise nach rechts verschieben	6.5.6
	WSFL	37	Daten wortweise nach links verschieben	6.5.7
	SFWR	38	Schreiben in einen FIFO-Speicher	6.5.8
	SFRD	39	Lesen aus einem FIFO-Speicher	6.5.9

Tab. A-30: Gesamtübersicht der Applikationsanweisungen für FX3U und FX3UC (1)

Einteilung	Anweisung	FNC	Bedeutung	Abschnitt
Daten- operationen	ZRST	40	Operandenbereiche zurücksetzen	6.6.1
	DECO	41	Daten decodieren	6.6.2
	ENCO	42	Daten codieren	6.6.3
	SUM	43	Ermittlung gesetzter Bits	6.6.4
	BON	44	Überprüfen eines Bits	6.6.5
	MEAN	45	Ermittlung von Durchschnittswerten	6.6.6
	ANS	46	Starten eines Zeitintervalls	6.6.7
	ANR	47	Rücksetzen von Anzeige-Bits	6.6.8
	SQR	48	Ermittlung der Quadratwurzel	6.6.9
	FLT	49	Umwandlung des Zahlenformats	6.6.10
High-Speed- Anweisungen	REF	50	Ein- und Ausgänge auffrischen	6.7.1
	REFF	51	Einstellen der Eingangfilter	6.7.2
	MTR	52	Einlesen einer Matrix (MTR)	6.7.3
	DHSCS	53	Setzen durch High-Speed-Counter	6.7.4
	DHSCR	54	Rücksetzen durch High-Speed-Counter	6.7.4
	DHSZ	55	Bereichsvergleich	6.7.5
	SPD	56	Geschwindigkeitserkennung	6.7.6
	PLSY	57	Impulsausgabe einer definierten Anzahl von Impulsen	6.7.7
	PWM	58	Impulsausgabe mit Impulsweitenmodulation	6.7.8
	PLSR	59	Ausgabe einer bestimmten Anzahl von Impulsen	6.7.9
Anwendungs- bezogene Anweisungen	IST	60	Schrittstatus initialisieren	6.8.1
	SER	61	Suchanweisung	6.8.2
	ABSD	62	Absoluter Counter-Vergleich	6.8.3
	INCD	63	Inkrementaler Counter-Vergleich	6.8.4
	TTMR	64	Teaching-Timer	6.8.5
	STMR	65	Sonder-Timer	6.8.6
	ALT	66	Flip-Flop-Funktion	6.8.7
	RAMP	67	Rampenfunktion	6.8.8
	ROTC	68	Rundtisch-Positionierung	6.8.9
SORT	69	Sortieranweisung	6.8.10	

Tab. A-31: Gesamtübersicht der Applikationsanweisungen für FX3U und FX3UC (2)

HINWEIS

Die Applikationsanweisungen FNC 70 bis FNC 305 werden im Kapitel 7 beschrieben.

B Ausführungszeiten der Anweisungen

B.1 Ausführungszeiten der FX1S-/FX1N-Serie

HINWEISE

„EIN-Schaltzeit“ ist die Zeit, die bei eingeschalteter Eingangsbedingung zur Ausführung der Anweisung benötigt wird. „AUS-Schaltzeit“ ist die Zeit, die die Anweisung benötigt, wenn die Eingangsbedingung nicht erfüllt ist.

Eine Beschreibung der Fußnoten befindet sich am Ende des Abschnitts.

B.1.1 Grundbefehle und Schrittstatus-Anweisungen

Anweisung	Bedeutung	Operanden	Programmschritte	Ausführungszeiten [μ s]	
				EIN-Schaltzeit	AUS-Schaltzeit
LD	Beginn einer Verknüpfung, Abfrage auf „1“-Signal	X, Y, M, S, T, C, Sondermerker	1	0,7	
LDI	Beginn einer Verknüpfung, Abfrage auf „0“-Signal				
AND	UND-Verknüpfung, Abfrage auf „1“-Signal		1	0,65	
ANI	UND-Verknüpfung, Abfrage auf „0“-Signal				
OR	ODER-Verknüpfung, Abfrage auf „1“-Signal				
ORI	ODER-Verknüpfung, Abfrage auf „0“-Signal				
LDP	Lade (gepulst); Beginn einer Verknüpfung mit Abfrage der ansteigenden Flanke	X, Y, M, S, T, C	1	11,7	
LDF	Lade (gepulst); Beginn einer Verknüpfung mit Abfrage der abfallenden Flanke				
ANDP	UND (gepulst); UND-Verknüpfung mit Abfrage der ansteigenden Flanke				
ANDF	UND (gepulst); UND-Verknüpfung mit Abfrage der abfallenden Flanke				
ORP	ODER (gepulst); ODER-Verknüpfung mit Abfrage der ansteigenden Flanke				
ORF	ODER (gepulst); ODER-Verknüpfung mit Abfrage der abfallenden Flanke				
ANB	UND-Block, Reihenschaltung von Parallelverknüpfungen	—	1	0,55	
ORB	ODER-Block, Parallelschaltung von Reihenverknüpfungen				

Tab. B-1: Ausführungszeiten für Grundbefehle und Schrittstatus-Anweisungen bei FX1S/FX1N (1)

Anweisung	Bedeutung	Operanden	Programm- schritte	Ausführungszeiten [μ s]	
				EIN- Schaltzeit	AUS- Schaltzeit
MPS	Abspeichern eines Verknüpfungsergebnisses	—	1	0,5	
MRD	Lesen eines Verknüpfungsergebnisses			0,55	
MPP	Lesen und Löschen des Verknüpfungsspeichers			0,5	
INV	Inversion, Umkehrung von Verarbeitungsergebnissen	—	1	0,08	
MC	Setzen einer Kontrollbedingung	N - Y, M	3	8,6	8,0
MCR	Rücksetzen einer Kontrollbedingung	N (Nesting)	2	4,1	
NOP	Leerzeile	—	1	0,45	
END	Programmende		1	450	
STL	Schrittstatus ausführen	S	1	15,8 + 8,2n ^①	
RET	Schrittstatus beenden	—	1	4,8	
OUT	Ausgabe, Zuweisung einer Verknüpfung	Y, M	1	0,7	
		S	2	4,4	
		Sondermerker	2	2,8	
		T-K	3	11,2	10,2
		T-D	3	12,2	11,2
		C-K (16 Bit)	3	8,1	6,9
		C-D (16 Bit)	3	9,5	8,0
		C-K (32 Bit)	5	8,1	6,8
C-D (32 Bit)	5	9,5	8,0		
SET	Setzen, Operanden setzen	Y, M	1	0,85	
		S	2	4,2	2,4
		S bei Einsatz in STL-Stufe ^①		18,6 + 6,8n	2,4
		Sondermerker	2	2,8	
RST	Rücksetzen, Operanden zurücksetzen	Y, M	1	0,85	
		S	2	3,8	2,4
		Sondermerker	2	2,8	
		T, C	2	8,7	7,3
		D, V, Z, Sonderregister	3	3,8	1,1
PLS	Impulserzeugung bei ansteigender Flanke	Y, M	2	10,8	
PLF	Impulserzeugung bei abfallender Flanke	Y, M	2	10,8	
P	Pointer adressieren	0 bis 63	1	0,45	
I	Interrupt-Pointer adressieren	□□□	1	0,45	

Tab. B-2: Ausführungszeiten für Grundbefehle und Schrittstatus-Anweisungen bei FX1S/FX1N (2)

B.1.2 Programmverzweigungsanweisungen

Anweisung		Ausführungszeit [μ s]		
Symbol	FNC-Anweisung	EIN-Schaltzeit	AUS-Schaltzeit	Sonderfunktion der FNC
CJ	FNC 00 /S+	7,1	6,4	—
CALL	FNC 01 /S+	9,3	3,2	—
SRET	FNC 02	8,3	—	—
IRET	FNC 03 ②	8,1	—	—
EI	FNC 04 ②	6,0	—	—
DI	FNC 05 ②	5,3	—	—
FEND	FNC 06 ②	450	—	—
WDT	FNC 07	3,7	2,7	—
FOR	FNC 08 /S+ ②	7,5	—	—
NEXT	FNC 09 ②	4,6	—	—

Tab. B-3: Ausführungszeiten für Programmverzweigungsanweisungen bei FX1S/FX1N

B.1.3 Vergleichs- und Transferanweisungen

Anweisung		Ausführungszeit [μ s]			
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit	Sonderfunktion der FNC
CMP	FNC 10 /S1+ /S2+ /D+	16	40	2,5	—
		32	41	4,5	—
ZCP	FNC 11 /S1+ /S2+ /S+ /D+	16	45	2,5	—
		32	47	4,5	—
MOV	FNC 12 /S+ /D+	16	19	2,5	—
		32	22	3,0	—
BMOV	FNC 15 /S+ /D+ /n ④	16	78 + 22n	2,5	—
BCD	FNC 18 /S+ /D+	16	30	2,5	—
		32	38,6	3,0	—
BIN	FNC 19 /S+ /D+	16	30	2,5	—
		32	35,5	3,0	—

Tab. B-4: Ausführungszeiten für Vergleichs- und Transferanweisungen bei FX1S/FX1N

B.1.4 Arithmetische Anweisungen

Anweisung		Ausführungszeit [μ s]			
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit	Sonderfunktion der FNC
ADD	FNC 20 /S1+ /S2+ /D+ Z, Cy, Br	16	37,5	2,5	—
		32	40,2	4,5	224
SUB	FNC 21 /S1+ /S2+ /D+ Z, Cy, Br	16	37,5	2,5	—
		32	40,5	4,5	232
MUL	FNC 22 /S1+ /S2+ /D+	16	38,2	2,5	—
		32	50,3	4,5	162
DIV	FNC 23 /S1+ /S2+ /D+	16	39,2	2,5	—
		32	63,5	4,5	197
INC	FNC 24 /D+ ③	16	14,5	2,5	—
		32	16,7	4,5	—
DEC	FNC 25 /D+ ③	16	14,5	2,5	—
		32	16,7	4,5	—
WAND	FNC 26 /S1+ /S2+ /D+	16	35,7	2,5	—
		32	37,3	4,5	—
WOR	FNC 27 /S1+ /S2+ /D+	16	35,7	2,5	—
		32	37,3	4,5	—
WXOR	FNC 28 /S1+ /S2+ /D+	16	35,7	2,5	—
		32	37,3	4,5	—

Tab. B-5: Ausführungszeiten für arithmetische Anweisungen bei FX1S/FX1N

Br (Borrow): M8021
 Cy (Carry): M8022
 F (Anweisung vollständig abgearbeitet): M8029

B.1.5 Rotations- und Shift-Anweisungen

Anweisung		Ausführungszeit [μ s]			
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit	Sonderfunktion der FNC
SFTR	FNC 34 /S+ /D+ /n1 /n2 ③ ⑥	16	55 + 1,25n	2,5	—
SFTL	FNC 35 /S+ /D+ /n1 /n2 ③ ⑥	16	56,1 + 1,25n	2,5	—
SFWR	FNC 38 /S+ /D+ /n ③ ⑦	16	41,6	2,5	—
SFRD	FNC 39 /S+ /D+ /n ③ ⑦	16	52,3	2,5	—

Tab. B-6: Ausführungszeiten für Rotations- und Shift-Anweisungen bei FX1S/FX1N

B.1.6 Datenoperationen

Anweisung		Ausführungszeit [μ s]			
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit	Sonderfunktion der FNC
ZRST	FNC 40 /D1+ /D2+ ⑧	16 (D)	32,4 + 0,5n	2,5	—
		16 (S)	37,8 + 0,9n		
		16 (C)			
		16 (T)			
		16 (M)	51,8 + 0,8n		
		16 (Y)			
DECO	FNC 41 /S+ /D+ /n	16	65,6	2,5	—
ENCO	FNC 42 /S+ /D+ /n	16	46,7	2,5	—

Tab. B-7: Ausführungszeiten für Datenoperationen bei FX1S/FX1N

B.1.7 High-Speed-Anweisungen

Anweisung		Ausführungszeit [μ s]			
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit	Sonderfunktion der FNC
REF	FNC 50 /D /n ⑩	16	19,5 + 4,3n	2,5	—
MTR	FNC 52 /S /D1 /D2 /n	16	22,6	9,8	—
HSCS	FNC 53 /S1+ /S2+ /D+ ⑫	32	46,8	4,5	—
HSCR	FNC 54 /S1+ /S2+ /D+ ⑫	32	46,8	4,5	—
SPD	FNC 56 /S1+ /S2+ /D+	②	39,5	43,8	—
		16	82,6	22,8	—
PLSY	FNC 57 /S1+ /S2+ /D+	32	100,6	34,9	—
		16	38,7	42,6	—
PLSR	FNC 59 /S1+ /S2+ /S3+ /D+	16	91,6	27,8	—
		32	113,7	41,6	

Tab. B-8: Ausführungszeiten für High-Speed-Anweisungen bei FX1S/FX1N

B.1.8 Anwendungsbezogene Anweisungen

Anweisung		Ausführungszeit [μ s]			
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit	Sonderfunktion der FNC
IST	FNC 60 /S /D1 /D2	16	81,7	2,5	—
ABSD	FNC 62 /S1+ /S2+ /D+ /n ¹⁴	16	56,5 + 6,3n	2,5	—
		32	62,7 + 11n	2,5	
INCD	FNC 63 /S1+ /S2+ /D+ /n	16	60,5	52,7	—
ALT	FNC 66 /D+	16	21,8	2,5	—
RAMP	FNC 67 /S1+ /S2+ /D+ /n	16	52,5	44,8	—

Tab. B-9: Ausführungszeiten für anwendungsbezogene Anweisungen bei FX1S/FX1N

B.1.9 Spezielle Anweisungen

Anweisung		Ausführungszeit [μ s]			
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit	Sonderfunktion der FNC
DSW	FNC 72 /S+ /D1+ /D2+ /n	16	95,0	92,6	—
SEGL	FNC 74 /S+ /D+ /n	16 1Set	84,5	40,7	—
FROM (nur FX1N)	FNC 78 /n1 /n2 /D+ /n3 ¹⁶	16	87 + 483n	2,5	—
		32	102 + 973n	4,5	
TO (nur FX1N)	FNC 79 /n1 /n2 /S+ /n3 ¹⁶	16	85 + 542n	2,5	—
		32	98 + 1121n	4,5	
RS	FNC 80	16	56,3	9,2	—
PRUN	FNC 81 /S+ /D+ ¹⁷	16	46,7 + 1,0n	2,5	—
		32	47,7 + 1,0n	3,0	—
ASCI	FNC 82	16	52,8 + 5,8n	2,5	—
HEX	FNC 83	16	54 + 8,9n	2,5	—
CCD	FNC 84	16	54,3 + 4,5n	2,5	—
VRRD	FNC 85 /S+ /D+	16	142,7	8,9	—
VRSC	FNC 86 /S+ /D+	16	142,7	8,9	—
PID	FNC 88	16	65,5	8,5	—

Tab. B-10: Ausführungszeiten für spezielle Anweisungen bei FX1S/FX1N

B.1.10 Positionier-Anweisungen

Anweisung		Bit	Ausführungszeiten [μ s]	
Symbol	FNC-Anweisung		EIN-Schaltzeit	AUS-Schaltzeit
ABS	FNC 155 /S+ /D1+ /D2+	32	86,7	85,7
ZRN	FNC 156 /S1+ /S2+ /S3+ /D	16	107,8	27,8
		32	130,5	40,8
PLSV	FNC 157 /S+ /D1 /D2+	16	79,6	22,7
		32	97,8	33,5
DRVI	FNC 158 /S1+ /S2+ /D1 /D2+	16	87,7	26,8
		32	110,6	40,7
DRVA	FNC 158 /S1+ /S2+ /D1 /D2+	16	89,6	26,8
		32	112,7	40,7

Tab. B-13: Ausführungszeiten für Positionier-Anweisungen bei FX1S/FX1N

B.1.11 Anweisungen für die integrierte Uhr

Anweisung		Bit	Ausführungszeit [μ s]		
Symbol	FNC-Anweisung		EIN-Schaltzeit	AUS-Schaltzeit	Sonderfunktion der FNC
TCMP	FNC 160 /S1+ /S2+ /S3+ /S+ /D+	16	52,6	2,5	—
TZCP	FNC 161 /S1+ /S2+ /S+ /D+	16	64,7	2,5	—
TADD	FNC 162 /S1+ /S2+ /D+	16	42,9	2,5	—
TSUB	FNC 163 /S1+ /S2+ /D+	16	42,9	2,5	—
TRD	FNC 166 /D+	16	29,7	2,5	—
TWR	FNC 167 /S+	16	633,5	2,5	—
HOUR	FNC 169 /S1+ /D1+ /D2+	16	39,7	38,7	
		32	41,9	40,6	

Tab. B-11: Ausführungszeiten für Anweisungen für die integrierte Uhr einer FX1S/FX1N-SPS

B.1.12 Datenaustausch mit Analogeingangsmodulen

Anweisung		Bit	Ausführungszeit [μ s]		
Symbol	FNC-Anweisung		EIN-Schaltzeit	AUS-Schaltzeit	Sonderfunktion der FNC
RD3A (nur FX1N)	FNC 176 /M1+ /M2+ /D+	16	1248,3	7,5	—
WR3A (nur FX1N)	FNC 177 /M1+ /M2+ /D+	16	1263,7	7,5	—

Tab. B-12: Ausführungszeiten für Anweisungen der FX1N zur Kommunikation mit Analogmodulen

B.1.13 Vergleichsanweisungen

Anweisung		Ausführungszeit [μ s]			
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit	Sonderfunktion der FNC
LD□	FNC 224 – 230 /S1+ /S2+	16	27,6		—
		32	28,2		
AND□	FNC 232 – 238 /S1+ /S2+	16	27,6		—
		32	28,2		
OR□	FNC 240 – 246 /S1+ /S2+	16	27,6		—
		32	28,2		

Tab. B-14: Ausführungszeiten für Vergleichsanweisungen bei FX1S/FX1N

- ① „n“ gibt die Anzahl der statischen STL-Anweisungen (Anzahl der parallelen/zusammenführenden Anweisungen) an.
- ② Die gekennzeichneten Anweisungen benötigen keine Kontakte.
- ③ Wird die statische Anweisung eingesetzt und nicht die Pulsanweisung, verändert sich der Wert der Zieladresse zyklisch.
- ④ „n“ gibt die Anzahl der zu verschiebenden Register an ($n \leq 512$).
- ⑥ „n“ gibt die Anzahl der zu verarbeitenden Bit-Operanden an.
- ⑦ „n“ gibt die Anzahl der zu verarbeitenden Operanden an ($2 \leq n \leq 512$).
- ⑧ „n“ gibt den zurückzusetzenden Operandenbereich an.
Der Operandentyp ist in den Klammern angegeben.
- ⑩ „n“ gibt den zu aktualisierenden Operandenbereich an ($8 \leq n \leq 128$ in 8-ter-Schritten).
- ⑫ Diese Anweisung kann maximal 6 mal zur gleichen Zeit aktiv sein.
- ⑭ „n“ gibt die Anzahl der Ausgangsadressen an ($n \leq 64$).
- ⑯ „n“ gibt die Anzahl der in das/aus dem Sondermodul zu schreibenden oder lesenden Datenworte an.
- ⑰ „n“ gibt die Anzahl der zu schreibenden oder lesenden Single-Byte-Datenworte (8 Bit) bei Parallelbetrieb zweier FX-Steuerungen an.

B.2 Ausführungszeiten der FX2N-/FX2NC-Serie

HINWEIS | Eine Beschreibung der Fußnoten befindet sich am Ende des Abschnitts.

B.2.1 Grundbefehle und Schrittstatus-Anweisungen

Anweisung	Bedeutung	Operanden	Programmschritte	Ausführungszeiten [µs]	
				EIN-Schaltzeit	AUS-Schaltzeit
LD	Beginn einer Verknüpfung, Abfrage auf „1“-Signal	X, Y, M, S, T, C, Sondermerker	1	0,08	
LDI	Beginn einer Verknüpfung, Abfrage auf „0“-Signal				
AND	UND-Verknüpfung, Abfrage auf „1“-Signal				
ANI	UND-Verknüpfung, Abfrage auf „0“-Signal				
OR	ODER-Verknüpfung, Abfrage auf „1“-Signal				
ORI	ODER-Verknüpfung, Abfrage auf „0“-Signal				
LDP	Lade (gepulst); Beginn einer Verknüpfung mit Abfrage der ansteigenden Flanke	X, Y, M, S, T, C	1	43,2	
LDF	Lade (gepulst); Beginn einer Verknüpfung mit Abfrage der abfallenden Flanke				
ANDP	UND (gepulst); UND-Verknüpfung mit Abfrage der ansteigenden Flanke				
ANDF	UND (gepulst); UND-Verknüpfung mit Abfrage der abfallenden Flanke				
ORP	ODER (gepulst); ODER-Verknüpfung mit Abfrage der ansteigenden Flanke				
ORF	ODER (gepulst); ODER-Verknüpfung mit Abfrage der abfallenden Flanke				
ANB	UND-Block, Reihenschaltung von Parallelverknüpfungen	—	1	0,08	
ORB	ODER-Block, Parallelschaltung von Reihenverknüpfungen				
MPS	Abspeichern eines Verknüpfungsergebnisses				
MRD	Lesen eines Verknüpfungsergebnisses				
MPP	Lesen und Löschen des Verknüpfungsspeichers				

Tab. B-15: Ausführungszeiten für Grundbefehle und Schrittstatus-Anweisungen bei FX2N und FX2NC (1)

Anweisung	Bedeutung	Operanden	Schritte	Ausführungszeiten [μs]	
				EIN-schaltzeit	AUS-schaltzeit
INV	Inversion, Umkehrung von Verarbeitungsergebnissen	—	1	0,08	
MC	Setzen einer Kontrollbedingung	N - Y, M	3	24,8	27,5
MCR	Rücksetzen einer Kontrollbedingung	N (Nesting)	2	20,8	
NOP	Leerzeile	—	1	0,08	
END	Programmende		1	508	
STL	Schrittstatus ausführen	S	1	27,3 + 12,6n ^①	
RET	Schrittstatus beenden	—	1	21,6	
OUT	Ausgabe, Zuweisung einer Verknüpfung	Y, M	1	0,08	
		S	2	24,4	24,3
		Sondermerker	2	0,16	0,16
		T-K	3	42,3	37,4
		T-D	3	42,2	37,2
		C-K (16 Bit)	3	25,5	24,9
		C-D (16 Bit)	3	25,3	25,0
		C-K (32 Bit)	5	25,3	24,9
		C-D (32 Bit)	5	25,2	24,9
SET	Setzen, Operanden setzen	Y, M	1	0,08	
		S	2	23,7	17,2
		S bei Einsatz in STL-Stufe ^①		27,3 + 12,6n	
		Sondermerker	2	0,16	0,16
RST	Rücksetzen, Operanden zurücksetzen	Y, M	1	0,16	
		S	2	23,1	17,3
		Sondermerker	2	0,16	0,16
		T, C	2	27	25
		D, V, Z, Sonderregister	3	21,9	17,1
PLS	Impulserzeugung bei ansteigender Flanke	Y, M	2	0,32	0,32
PLF	Impulserzeugung bei abfallender Flanke	Y, M	2	0,32	
P	Pointer adressieren	0 bis 63	1	0,08	
I	Interrupt-Pointer adressieren	I□□□	1	0,08	

Tab. B-16: Ausführungszeiten für Grundbefehle und Schrittstatus-Anweisungen bei FX2N und FX2NC (2)

B.2.2 Programmverzweigungsanweisungen

Anweisung		Ausführungszeit [μ s]		
Symbol	FNC-Anweisung	EIN-Schaltzeit	AUS-Schaltzeit	Sonderfunktion der FNC
CJ	FNC 00 /S+	29	6,4	—
CALL	FNC 01 /S+	32,2	6,4	—
SRET	FNC 02 ②	21,2	21,2	—
IRET	FNC 03 ②	18,8	18,1	—
EI	FNC 04 ②	55,8	55,8	—
DI	FNC 05 ②	18,5	18,5	—
FEND	FNC 06 ②	508		—
WDT	FNC 07	26,3	6,4	—
FOR	FNC 08 /S+ ②	27,6	27,6	—
NEXT	FNC 09 ②	5,2	5,2	—

Tab. B-17: Ausführungszeiten für Programmverzweigungsanweisungen bei FX2N und FX2NC

B.2.3 Vergleichs- und Transferanweisungen

Anweisung		Ausführungszeit [μ s]			
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit	Sonderfunktion der FNC
CMP	FNC 10 /S1+ /S2+ /D+	16	87,6	6,4	—
		32	91,9	6,4	—
ZCP	FNC 11 /S1+ /S2+ /S+ /D+	16	103,2	6,4	—
		32	108,9	6,4	—
MOV	FNC 12 /S+ /D+	16	1,52	1,52	—
		32	1,84	1,84	—
SMOV	FNC 13 /S+ /m1 /m2 /D+ /n	16	155,2	6,4	—
CML	FNC 14 /S+ /D+	16	51,4	6,4	—
		32	55,9	6,4	—
BMOV	FNC 15 /S+ /D+ /n ④	16	97 + 1,7n	6,4	—
FMOV	FNC 16 /S+ /D+ /n ④	16	69,1 + 2,8n	6,4	—
		32	73,2 + 5,2n	6,4	—
XCH	FNC 17 /D1+ /D2+ ③	16	57,2	6,4	—
		32	64	6,4	—
BCD	FNC 18 /S+ /D+	16	37,9	6,4	—
		32	57,6	6,4	—
BIN	FNC 19 /S+ /D+	16	32,4	6,4	—
		32	44,5	6,4	—

Tab. B-18: Ausführungszeiten für Vergleichs- und Transferanweisungen bei FX2N und FX2NC

B.2.4 Arithmetische Anweisungen

Anweisung		Ausführungszeit [μ s]			
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit	Sonderfunktion der FNC
ADD	FNC 20 /S1+ /S2+ /D+ Z, Cy, Br	16	27,6	6,4	—
		32	28,9	6,4	224
SUB	FNC 21 /S1+ /S2+ /D+ Z, Cy, Br	16	27,6	6,4	—
		32	28,9	6,4	232
MUL	FNC 22 /S1+ /S2+ /D+	16	25,2	6,4	—
		32	31,4	6,4	162
DIV	FNC 23 /S1+ /S2+ /D+	16	32	6,4	—
		32	36,4	6,4	197
INC	FNC 24 /D+ ③	16	18,8	6,4	—
		32	20,2	6,4	—
DEC	FNC 25 /D+ ③	16	18,9	6,4	—
		32	20	6,4	—
WAND	FNC 26 /S1+ /S2+ /D+	16	23,4	6,4	—
		32	24,8	6,4	—
WOR	FNC 27 /S1+ /S2+ /D+	16	23,5	6,4	—
		32	24,7	6,4	—
WXOR	FNC 28 /S1+ /S2+ /D+	16	23,5	6,4	—
		32	25,0	6,4	—
NEG	FNC 29 /D+ ③	16	35,3	6,4	—
		32	38,4	6,4	—

Tab. B-19: Ausführungszeiten für arithmetische Anweisungen bei FX2N und FX2NC

Br (Borrow): M8021
 Cy (Carry): M8022
 F (Anweisung vollständig abgearbeitet): M8029

B.2.5 Rotations- und Shift-Anweisungen

Anweisung		Ausführungszeit [μ s]			
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit	Sonderfunktion der FNC
ROR	FNC 30 /D+ /n Cy 3 5	16	61,7	6,4	—
		32	65,3	6,4	—
ROL	FNC 31 /D+ /n Cy 3 5	16	61,2	6,4	—
		32	65,2	6,4	—
RCR	FNC 32 /D+ /n Cy 3 5	16	66,3 + 2,2n	6,4	—
		32	69,7 + 2,6n	6,4	—
RCL	FNC 33 /D+ /n Cy 3 5	16	65,8 + 2,2n	6,4	—
		32	69,5 + 2,6n	6,4	—
SFTR	FNC 34 /S+ /D+ /n1 /n2 3 6	16	107 + 53,8n	6,4	—
SFTL	FNC 35 /S+ /D+ /n1 /n2 3 6	16	105 + 53,8n	6,4	—
WSFR	FNC 36 /S+ /D+ /n1 /n2 3 4	16	126 + 11,7n	6,4	—
WSFL	FNC 37 /S+ /D+ /n1 /n2 3 4	16	125 + 11,8n	6,4	—
SFWR	FNC 38 /S+ /D+ /n 3 7	16	83,9	6,4	—
SFRD	FNC 39 /S+ /D+ /n 3 7	16	80,2	6,4	—

Tab. B-20: Ausführungszeiten für Rotations- und Shift-Anweisungen bei FX2N und FX2NC

B.2.6 Datenoperationen

Anweisung		Ausführungszeit [μ s]			
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit	Sonderfunktion der FNC
ZRST	FNC 40 /D1+ /D2+ ⑧	16 (D)	77 + 1,7n	6,4	—
		16 (S)	83 + 11,1n		
		16 (C)			
		16 (T)			
		16 (M)	89,2 + 9,4n		
		16 (Y)			
DECO	FNC 41 /S+ /D+ /n	16	76	6,4	—
ENCO	FNC 42 /S+ /D+ /n	16	81,8	6,4	—
SUM	FNC 43 /S+ /D+	16	72,8	6,4	—
		32	94,6	6,4	—
BON	FNC 44 /S+ /D+ /n	16	78,2	6,4	—
		32	82,3	6,4	—
MEAN	FNC 45 /S+ /D+ /n ⑨	16	83,8 + 3,4n	6,4	—
		32	90,9 + 6,7n	6,4	—
ANS	FNC 46 /S+ /m /D+	16	100,8	6,4	—
ANR	FNC 47 ③	16	37,7	6,4	—
SQR	FNC 48	16	150,2	6,4	—
		32	154,8	6,4	344
FLT	FNC 49	16	66,8	6,4	—
		32	66,8	6,4	—

Tab. B-21: Ausführungszeiten für Datenoperationen bei FX2N und FX2NC

B.2.7 High-Speed-Anweisungen

Anweisung		Ausführungszeit [μ s]			
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit	Sonderfunktion der FNC
REF	FNC 50 /D /n ⑩	16	99,6 + 0,6n	6,4	—
REFF	FNC 51 /n ⑪	16	65,3 + 1,7n	6,4	—
MTR	FNC 52 /S /D1 /D2 /n	16	39,1	23,6	—
HSCS	FNC 53 /S1+ /S2+ /D+ ⑫	32	87,8	6,4	—
HSCR	FNC 54 /S1+ /S2+ /D+ ⑫	32	88,6	6,4	—
HSZ	FNC 55 /S1+ /S2+ /S+ /D+ ⑫	32	100,6	6,4	—
SPD	FNC 56 /S1+ /S2+ /D+	②	80,2	80,2	—
PLSY	FNC 57 /S1+ /S2+ /D+	16	85	73,3	—
		32	86,6	75,8	—
PWM	FNC 58 /S1+ /S2+ /D+	16	70,4	73,3	—
PLSR	FNC 59 /S1+ /S2+ /S3+ /D+	16	122,6	87,5	—
		32	125,6	90,5	

Tab. B-22: Ausführungszeiten für High-Speed-Anweisungen bei FX2N und FX2NC

B.2.8 Anwendungsbezogene Anweisungen

Anweisung		Ausführungszeit [μ s]			
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit	Sonderfunktion der FNC
IST	FNC 60 /S /D1 /D2	16	114,3	6,4	—
SER	FNC 61 $\text{\textcircled{13}}$	16	129,2 + 8,6n	22,9	—
		32	147 + 9n	29	
ABSD	FNC 62 /S1+ /S2+ /D+ /n $\text{\textcircled{14}}$	16	91,8 + 20,2n	6,4	—
		32	97,5 + 21,5n	6,4	
INCD	FNC 63 /S1+ /S2+ /D+ /n	16	110,5	19,5	—
TTMR	FNC 64 /D+ /n	16	54,9	44,9	—
STMR	FNC 65 /S+ /m /D+	16	84,4	84,4	—
ALT	FNC 66 /D+	16	50,1	6,4	—
RAMP	FNC 67 /S1+ /S2+ /D+ /n	16	98,1	81,6	—
ROTC	FNC 68 /S+ /m1 /m2 /D+	16	118,4	107,2	—
SORT	FNC 69 $\text{\textcircled{15}}$	16	50,5	19,5	—

Tab. B-23: Ausführungszeiten für anwendungsbezogene Anweisungen bei FX2N/FX2NC

B.2.9 Spezielle Anweisungen

Anweisung		Ausführungszeit [μ s]			
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit	Sonderfunktion der FNC
TKY	FNC 70 /S+ /D1+ /D2+	16	97,2	22,2	—
		32	98,7	22,2	—
HKY	FNC 71 /S+ /D1+ /D2+ /D3+	16	92,2	27,4	—
		32	65,0	6,4	—
DSW	FNC 72 /S+ /D1+ /D2+ /n	16	92,2	27,4	—
SEGD	FNC 73 /S1+ /D+	16	65	6,4	—
SEGL	FNC 74 /S+ /D+ /n	16 1Set	105,9	26,5	—
ARWS	FNC 75 /S1+ /D+ /D2+ /n	16	134,4	22,1	—
ASC	FNC 76 /S+ /D+	16	49,5	6,4	—
PR	FNC 77 /S+ /D+	16	während des Drückens: 114,8 bei beendetem Druckvorgang: 88	88,5	—
FROM	FNC 78 /n1 /n2 /D+ /n3 $\textcircled{16}$	16	97 + 487n	6,4	—
		32	99 + 962n	6,4	
TO	FNC 79 /n1 /n2 /S+ /n3 $\textcircled{16}$	16	94 + 557n	6,4	—
		32	96 + 1099n	6,4	
RS	FNC 80	16	117,6	18	—
		32			
PRUN	FNC 81 /S+ /D+ $\textcircled{17}$	16	65,6 + 17n	6,4	—
		32	67 + 17,7n	6,4	—
ASCI	FNC 82	16	88,2 + 10,8n	6,4	—
HEX	FNC 83	16	89,7 + 20n	6,4	—
CCD	FNC 84	16	90,5 + 4,8n	6,4	—
VRRD	FNC 85 /S+ /D+	16	209,7	27,3	—
VRSC	FNC 86 /S+ /D+	16	202,4	27,3	—
PID	FNC 88	16	155	89	—

Tab. B-24: Ausführungszeiten für spezielle Anweisungen bei FX2N und FX2NC (1)

Anweisung		Ausführungszeit [μ s]			
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit	Sonderfunktion der FNC
ECMP	FNC 110 /S1+ /S2+ /D+	32	104,4	6,4	—
EZCP	FNC 111 /S1+ /S2+ /S3+ /D+	32	124,5	6,4	—
EBCD	FNC 118 /S+ /D+	32	106,9	6,4	—
EBIN	FNC 119 /S+ /D+	32	81,3	6,4	—
EADD	FNC 120 /S1+ /S2+ /D+	32	117,4	6,4	—
ESUB	FNC 121 /S1+ /S2+ /D+	32	117,4	6,4	—
EMUL	FNC 122 /S1+ /S2+ /D+	32	96,4	6,4	—
EDIV	FNC 123 /S1+ /S2+ /D+	32	100,4	6,4	—
ESQR	FNC 127 /S+ /D+	32	152,1	6,4	—
INT	FNC 129 /S+ /D+	16	67,5	6,4	—
		32	70,4	6,4	
SIN	FNC 130 /S+ /D+	32	199,5	6,4	—
COS	FNC 131 /S+ /D+	32	262,5	6,4	—
TAN	FNC 132 /S+ /D+	32	425,3	6,4	—
SWAP	FNC 147 /S+	16	36,1	6,4	—
		32	41,2	6,4	
TCMP	FNC 160 /S1+ /S2+ /S3+ /S+ /D+	16	134,2	6,4	—
TZCP	FNC 161 /S1+ /S2+ /S+ /D+	16	140,2	6,4	—
TADD	FNC 162 /S1+ /S2+ /D+	16	118,8	6,4	—
TSUB	FNC 163 /S1+ /S2+ /D+	16	109,4	6,4	—
TRD	FNC 166 /D+	16	46,2	6,4	—
TWR	FNC 167 /S+	16	112	6,4	—
GRY	FNC 170 /S+	16	102,5	6,4	—
		32	107,1	6,4	
GBIN	FNC 171 /D+	16	103,4	6,4	—
		32	107,5	6,4	
RD3A	FNC 176 /M1+ /M2+ /D+	16	1248,3	7,5	
WR3A	FNC 177 /M1+ /M2+ /D+	16	1263,7	7,5	
LD □	FNC 224 – 230 /S1+ /S2+	16	1,52		—
		32	1,84		
AND □	FNC 232 – 238 /S1+ /S2+	16	1,52		—
		32	1,84		
OR □	FNC 240 – 246 /S1+ /S2+	16	1,52		—
		32	1,84		

Tab. B-25: Ausführungszeiten für spezielle Anweisungen bei FX2N und FX2NC (2)

- ① „n“ gibt die Anzahl der statischen STL-Anweisungen (Anzahl der parallelen/zusammenführenden Anweisungen) an.
- ② Die gekennzeichneten Anweisungen benötigen keine Kontakte.
- ③ Wird die statische Anweisung eingesetzt und nicht die Pulsanweisung, verändert sich der Wert der Zieladresse zyklisch.
- ④ „n“ gibt die Anzahl der zu verschiebenden Register an ($n \leq 512$).
- ⑤ „n“ gibt die Anzahl der zu verarbeitenden Bit-Operanden an ($n \leq 16$ im 16-Bit-Modus, $n \leq 32$ im 32-Bit-Modus).
- ⑥ „n“ gibt die Anzahl der zu verarbeitenden Bit-Operanden an.
- ⑦ „n“ gibt die Anzahl der zu verarbeitenden Operanden an ($2 \leq n \leq 512$).
- ⑧ „n“ gibt den zurückzusetzenden Operandenbereich an. Der Operandentyp ist in den Klammern angegeben.
- ⑨ „n“ gibt die Operandenadressen an, die mit der MEAN-Anweisung verarbeitet werden ($1 \leq n \leq 64$).
- ⑩ „n“ gibt den zu aktualisierenden Operandenbereich an ($8 \leq n \leq 128$ in 8-ter-Schritten).
- ⑪ „n“ gibt die Zeitkonstante des Eingangsfilters an ($0 \leq n \leq 60$ ms).
- ⑫ Diese Anweisung kann maximal 6 mal zur gleichen Zeit aktiv sein.
- ⑬ „n“ gibt die Anzahl der Stapel-Elemente an ($n \leq 256$ für 16-Bit-Verarbeitung, $n \leq 128$ für 32-Bit-Verarbeitung).
- ⑭ „n“ gibt die Anzahl der Ausgangsadressen an ($n \leq 64$).
- ⑮ „n“ gibt die Anzahl der Elemente der Datentabelle an ($1 \leq m1 \leq 32$). Zur vollständigen SORT-Verarbeitung wird die SORT-Anweisung $m1$ mal ausgeführt.
- ⑯ „n“ gibt die Anzahl der in das/aus dem Sondermodul zu schreibenden oder lesenden Datenworte an.
- ⑰ „n“ gibt die Anzahl der zu schreibenden oder lesenden Single-Byte-Datenworte (8 Bit) bei Parallelbetrieb zweier FX-Steuerungen an.

B.3 Ausführungszeiten der FX3G-/FX3GC-/FX3GE-Serie

HINWEISE

Eine Beschreibung der Fußnoten (①, ②, etc.) befindet sich am Ende des Abschnitts.

Werden Applikationsanweisungen als Puls-Anweisungen programmiert (Anhängen eines „P“, z. B. MOVP), gelten andere Ausführungszeiten (siehe Abschnitt B.6).

B.3.1 Grundbefehle und Schrittstatus-Anweisungen

Anweisung	Bedeutung	Operanden	Programm-schritte	Ausführungszeiten [µs]			
				Standard-Modus		Erweiterter Modus	
				EIN-Schaltzeit	AUS-Schaltzeit	EIN-Schaltzeit	AUS-Schaltzeit
LD	Beginn einer Verknüpfung, Abfrage auf „1“-Signal	X, Y, M, S, T, C, Sondermerker	1, 2 oder 3 (siehe Abschnitt 4.1.2)	0,21 / 1,34 ^①		0,42 / 1,68 ^①	
LDI	Beginn einer Verknüpfung, Abfrage auf „0“-Signal						
AND	UND-Verknüpfung, Abfrage auf „1“-Signal			0,2 / 1,16 ^② / 1,2 ^③		0,41 / 1,48 ^② / 1,52 ^③	
ANI	UND-Verknüpfung, Abfrage auf „0“-Signal						
OR	ODER-Verknüpfung, Abfrage auf „1“-Signal			0,21 / 1,28 ^② / 1,32 ^③		0,42 / 1,6 ^①	
ORI	ODER-Verknüpfung, Abfrage auf „0“-Signal						
LDP	Lade (gepulst); Beginn einer Verknüpfung mit Abfrage der ansteigenden Flanke	X, Y, M, S, T, C	2 oder 3 (siehe Abschnitt 4.1.2)	2,32 / 2,4 ^③	2,28 / 2,36 ^③	2,72 / 2,8 ^③	2,68 / 2,76 ^③
LDF	Lade (gepulst); Beginn einer Verknüpfung mit Abfrage der abfallenden Flanke			2,36 / 2,44 ^③	2,32 / 2,4 ^③	2,76 / 2,84 ^③	2,72 / 2,8 ^③
ANDP	UND (gepulst); UND-Verknüpfung mit Abfrage der ansteigenden Flanke			2,36 / 2,48 ^③	2,32 / 2,44 ^③	2,72 / 2,8 ^③	2,68 / 2,76 ^③
ANDF	UND (gepulst); UND-Verknüpfung mit Abfrage der abfallenden Flanke			2,4 / 2,48 ^③	2,36 / 2,44 ^③	2,76 / 2,8 ^③	2,72 / 2,76 ^③
ORP	ODER (gepulst); ODER-Verknüpfung mit Abfrage der ansteigenden Flanke			2,36 / 2,48 ^③	2,32 / 2,44 ^③	2,72 / 2,8 ^③	2,68 / 2,76 ^③
ORF	ODER (gepulst); ODER-Verknüpfung mit Abfrage der abfallenden Flanke			2,4 / 2,48 ^③	2,36 / 2,44 ^③	2,76 / 2,8 ^③	2,72 / 2,76 ^③
ANB	UND-Block, Reihenschaltung von Parallelverknüpfungen			—	1	0,2	
ORB	ODER-Block, Parallelschaltung von Reihenverknüpfungen	0,2				0,41	

Tab. B-26: Ausführungszeiten für Grundbefehle und Schrittstatus-Anweisungen bei einer Steuerung der FX3G-, FX3GC- oder FX3GE-Serie (1)

- ① Beim Zugriff auf die Operanden M1536 – M7679, M8256 – M8511 oder S1024 – S4095 (Ausführung der Anweisung in 2 oder 3 Schritten)
- ② Beim Zugriff auf die Operanden M1536 – M3583, M8256 – M8511 oder S1024 – S4095 (Ausführung der Anweisung in 2 Schritten)
- ③ Beim Zugriff auf die Merker M3584 bis M7679 (Ausführung der Anweisung in 3 Schritten)

Anweisung	Bedeutung	Operanden	Programmschritte	Ausführungszeiten [μs]			
				Standard-Modus		Erweiterter Modus	
				EIN-Schaltzeit	AUS-Schaltzeit	EIN-Schaltzeit	AUS-Schaltzeit
MPS	Verknüpfungsergebnis speichern	—	1	0,21		0,42	
MRD	Verknüpfungsergebnis lesen			0,2		0,41	
MPP	Lesen und Löschen des Verknüpfungsspeichers			0,17		0,38	
INV	Inversion, Verarbeitungsergebnisumkehr			0,17		0,38	
MEP	Impuls erzeugen, wenn das Verknüpfungsergebnis „1“ wird	—	1	1,52		1,73	
MEF	Impuls erzeugen, wenn das Verknüpfungsergebnis „0“ wird	—	1	1,6		1,81	
MC	Setzen einer Kontrollbedingung	N - Y, M	3 oder 4 (Abschnitt 4.1.2)	1,63	1,76	2,01	2,13
MCR	Rücksetzen einer Kontrollbedingung	N (Nesting)	2	1,3		1,51	
NOP	Leerzeile	—	1	0,21		0,42	
END	Programmende		1	$292 + (2,5 \times X) + (3,75 \times Y)$ X: Anzahl der Eingänge Y: Anzahl der Ausgänge Falls eine FEND- und eine END-Anweisung zusammen verwendet werden, ist nur die Ausführungszeit der END-Anweisung relevant.			
STL	Schrittstatus ausführen	S	1	1,8 + 0,49n ①		2,18 + 0,49n ①	
RET	Schrittstatus beenden	—	1	1,05		1,05	
OUT	Ausgabe, Zuweisung eines Verknüpfungsergebnisses	Y	1	0,21		0,42	
		M	1, 2 oder 3 (Abschnitt 4.1.2)	0,21 / 1,2 ^①		0,42 / 1,52 ^② / 1,56 ^③	
		S	2	1,25	1,25	1,65	1,65
		T-K	3	2,41	2,16	3,19	2,95
		T-K (T246 – T319)		2,69	2,04	3,47	2,82
		T-D	3	2,77	2,54	3,55	3,32
		T-D (T246 – T319)		3,06	2,41	3,83	3,19
		C-K (16 Bit)	3	1,9	1,57	2,68	2,35
		C-K (32 Bit)	5	2,25	1,72	3,02	2,89
		C-K (C235 – C255)		1,6	1,06	2,66	1,45
		C-D (16 Bit)		3	2,26	1,93	3,04
		C-D (32 Bit)	5	2,49	2,35	3,25	3,1
		C-D (C235 – C255)		2,21	1,06	3,00	1,45

Tab. B-27: Ausführungszeiten für Grundbefehle und Schrittstatus-Anweisungen bei einer Steuerung der FX3G-, FX3GC- oder FX3GE-Serie (2)

- ① Beim Zugriff auf die Operanden M1536 – M7679 oder M8000 – M8511 (Ausführung der Anweisung in 2 oder 3 Schritten)
- ② Beim Zugriff auf die Operanden M1536 – M3583 oder M8000 – M8511 (Ausführung der Anweisung in 2 Schritten)
- ③ Beim Zugriff auf die Merker M3584 bis M7679 (Ausführung der Anweisung in 3 Schritten)

Anweisung	Bedeutung	Operanden	Programmschritte	Ausführungszeiten [µs]			
				Standard-Modus		Erweiterter Modus	
				EIN-Schaltzeit	AUS-Schaltzeit	EIN-Schaltzeit	AUS-Schaltzeit
SET	Setzen, Operanden setzen	Y	1	0,23		0,44	
		M	1, 2 oder 3 (Abschnitt 4.1.2)	0,23 / 0,72 ^①		0,44 / 0,88 ^①	
		S	2	1,23	0,59	1,62	0,78
		S bei Einsatz in STL-Stufe ①		1,6 + 0,51n		1,66 + 0,55Wn	
RST	Rücksetzen, Operanden rücksetzen	Y	1	0,23		0,44	
		M	1, 2 oder 3 (siehe Abschnitt 4.1.2)	0,23 / 0,72 ^①		0,44 / 0,88 ^①	
		S	2	1,17	0,59	1,55	0,78
		T	2	1,68	1,21	2,08	1,6
		C (16 Bit)	2	1,68	1,18	2,08	1,19
		C (32 Bit)	2	1,82	1,19	2,31	1,58
		D, V, Z, Sonderregister	3	1,08	0,54	1,66	0,74
PLS	Impulserzeugung bei ansteigender Flanke	Y, M	2 oder 3 (siehe Abschnitt 4.1.2)	1,52 / 1,56 ^②		1,88	
PLF	Impulserzeugung bei abfallender Flanke						

Tab. B-29: Ausführungszeiten für Grundbefehle und Schrittstatus-Anweisungen bei einer Steuerung der FX3G-, FX3GC- oder FX3GE-Serie (3)

- ① Beim Zugriff auf M1536–M7679 oder M8000–M8511 (Ausführung der Anweisung in 2 oder 3 Schritten)
- ② Beim Zugriff auf die Merker M3584 bis M7679 (Ausführung der Anweisung in 3 Schritten)

B.3.2 Programmverzweigungsanweisungen

Anweisung		Ausführungszeit [µs]			
Symbol	FNC-Anweisung	Standard-Modus		Erweiterter Modus	
		EIN-Schaltzeit	AUS-Schaltzeit	EIN-Schaltzeit	AUS-Schaltzeit
CJ	FNC 00 /S+	2,59	0,61	2,93	0,8
CALL	FNC 01 /S+	6,44	0,61	7,38	0,8
SRET	FNC 02 ②		—		—
IRET	FNC 03 ②	6,34	—	6,9	—
EI	FNC 04 ②	2,52	—	2,71	—
DI	FNC 05 ②	1,94	—	2,12	—
FEND	FNC 06 ②	$292 + (2,5 \times X) + (3,75 \times Y)$ X: Anzahl der Eingänge, Y: Anzahl der Ausgänge Falls eine FEND- und eine END-Anweisung zusammen verwendet werden, ist nur die Ausführungszeit der END-Anweisung relevant.			
WDT	FNC 07	1,94	0,61	2,16	0,8
FOR	FNC 08 /S+ ②	4,98	—	5,76	—
NEXT	FNC 09 ②		—		—

Tab. B-28: Ausführungszeiten für Programmverzweigungsanweisungen bei einer FX3G/FX3GC/FX3GE

B.3.3 Vergleichs- und Transferanweisungen

Anweisung		Ausführungszeit [μ s]				
Symbol	FNC-Anweisung	Bit	Standard-Modus		Erweiterter Modus	
			EIN-Schaltzeit	AUS-Schaltzeit	EIN-Schaltzeit	AUS-Schaltzeit
CMP	FNC 10 /S1+ /S2+ /D+	16	11,18	0,61	12,91	0,8
		32	11,54		13,66	
ZCP	FNC 11 /S1+ /S2+ /S+ /D+	16	12,51	0,61	14,62	0,8
		32	13,15		16,02	
MOV	FNC 12 /S+ /D+	16	0,52	0,38	1,29	0,61
		32	0,61	0,41	1,56	0,65
SMOV	FNC 13 /S+ /m1 /m2 /D+ /n	16	16,19	0,61	18,68	0,8
CML	FNC 14 /S+ /D+	16	6,68	0,61	8,03	0,8
		32	7,15		8,5	
BMOV	FNC 15 /S+ /D+ /n ④	16	15,08 + 5,58n	0,61	17,54 + 5,58n	0,8
FMOV	FNC 16 /S+ /D+ /n ④	16	10,20 + 0,77n	0,61	11,90 + 0,77n	0,8
		32	10,87 + 0,79n		12,94 + 0,79n	
BCD	FNC 18 /S+ /D+	16	2,93	0,41	3,9	0,61
		32	4,6	0,43	5,8	0,65
BIN	FNC 19 /S+ /D+	16	2,93	0,41	4,01	0,61
		32	4,61	0,43	5,82	0,65

Tab. B-30: Ausführungszeiten für Vergleichs- und Transferanweisungen bei einer FX3G/FX3GC/FX3GE

HINWEIS

Weitere Angaben zu den Ausführungszeiten der MOV-Anweisung enthält der Abschnitt B.7.2.

B.3.4 Arithmetische Anweisungen

Anweisung		Ausführungszeit [μ s]				
Symbol	FNC-Anweisung	Bit	Standard-Modus		Erweiterter Modus	
			EIN-Schaltzeit	AUS-Schaltzeit	EIN-Schaltzeit	AUS-Schaltzeit
ADD	FNC 20 /S1+ /S2+ /D+ Z, Cy, Br	16	1,79	0,41	3,14	0,61
		32	2,13	0,45	3,87	0,65
SUB	FNC 21 /S1+ /S2+ /D+ Z, Cy, Br	16	1,79	0,41	3,14	0,61
		32	2,13	0,45	3,87	0,65
MUL	FNC 22 /S1+ /S2+ /D+	16	1,29	0,41	2,78	0,61
		32	3,21	0,45	4,88	0,65
DIV	FNC 23 /S1+ /S2+ /D+	16	1,93	0,41	3,48	0,61
		32	3,7	0,45	5,3	0,65
INC	FNC 24 /D+ ③	16	0,78	0,41	1,53	0,61
		32	1,01		1,96	0,65
DEC	FNC 25 /D+ ③	16	0,78	0,41	1,53	0,61
		32	1,01		1,96	0,65
WAND	FNC 26 /S1+ /S2+ /D+	16	1,24	0,41	2,61	0,61
		32	1,59	0,45	3,34	0,65
WOR	FNC 27 /S1+ /S2+ /D+	16	1,24	0,41	2,61	0,61
		32	1,59	0,45	3,34	0,65
WXOR	FNC 28 /S1+ /S2+ /D+	16	1,24	0,41	2,61	0,61
		32	1,59	0,45	3,34	0,65

Tab. B-31: Ausführungszeiten für arithmetische Anweisungen bei einer FX3G/FX3GC/FX3GE

Z (Zero): M8020
 Br (Borrow): M8021
 Cy (Carry): M8022
 F (Anweisung vollständig abgearbeitet): M8029

B.3.5 Rotations- und Shift-Anweisungen

Anweisung		Ausführungszeit [μ s]				
Symbol	FNC-Anweisung	Bit	Standard-Modus		Erweiterter Modus	
			EIN-Schaltzeit	AUS-Schaltzeit	EIN-Schaltzeit	AUS-Schaltzeit
ROR	FNC 30 /D+ /n Cy ③ ⑤	16	11,96	0,61	13,11	0,8
		32	16,65		18,18	
ROL	FNC 31 /D+ /n Cy ③ ⑤	16	11,96	0,61	13,11	0,8
		32	16,65		18,18	
SFTR	FNC 34 /S+ /D+ /n1 /n2 ③ ⑥	16	14,51 + 0,46n	0,61	16,61 + 0,46n	0,8
SFTL	FNC 35 /S+ /D+ /n1 /n2 ③ ⑥	16	14,7 + 0,46n	0,61	16,82 + 0,46n	0,8
WSFR	FNC 36 /S+ /D+ /n1 /n2 ③ ④	16	11,74 + 1,44n	0,61	13,91 + 1,44n	0,8
WSFL	FNC 37 /S+ /D+ /n1 /n2 ③ ④	16	12,02 + 1,48n	0,61	14,11 + 1,48n	0,8
SFWR	FNC 38 /S+ /D+ /n ③ ⑦	16	7,46	0,61	9,19	0,8
SFRD	FNC 39 /S+ /D+ /n ③ ⑦	16	9,27	0,61	10,99	0,8

Tab. B-32: Ausführungszeiten für Rotations- und Shift-Anweisungen bei einer FX3G/FX3GC/FX3GE

B.3.6 Datenoperationen

Anweisung		Ausführungszeit [μ s]				
Symbol	FNC-Anweisung	Bit	Standard-Modus		Erweiterter Modus	
			EIN-Schaltzeit	AUS-Schaltzeit	EIN-Schaltzeit	AUS-Schaltzeit
ZRST	FNC 40 /D1+ /D2+ ⑧	16 (D)	7,96 + 0,22n	0,61	10,26 + 0,22n	0,8
		16 (T)	8,77 + 0,46n		10,68 + 0,46n	
		16 (M)	14,76 + 0,37n		16,16 + 0,37n	
DECO	FNC 41 /S+ /D+ /n	16	8,5	0,61	10,31	0,8
ENCO	FNC 42 /S+ /D+ /n	16	9,29	0,61	11,02	0,8
SUM	FNC 43 /S+ /D+	16	6,95	0,61	8,3	0,8
		32	7,43		8,76	
BON	FNC 44 /S+ /D+ /n	16	10,4	0,61	12,13	0,8
		32	10,77		12,83	
MEAN	FNC 45 /S+ /D+ /n ⑨	16	13,83 + 3,33n	0,61	16,28 + 3,32n	0,8
		32	14,69 + 3,5n		17,6 + 3,49n	
ANS	FNC 46 /S+ /m /D+	16	9,55	9,09	11,27	10,81
ANR	FNC 47 ③	16	11,08	0,61	11,12	0,8
FLT	FNC 49	16	6,96	0,61	8,28	0,8
		32	7,56		8,88	

Tab. B-33: Ausführungszeiten für Datenoperationen bei einer FX3G/FX3GC/FX3GE

B.3.7 High-Speed-Anweisungen

Anweisung		Ausführungszeit [μ s]					
Symbol	FNC-Anweisung	Bit	Standard-Modus		Erweiterter Modus		
			EIN-Schaltzeit	AUS-Schaltzeit	EIN-Schaltzeit	AUS-Schaltzeit	
REF	FNC 50 /D /n ⑩	16	9 + 1,43n	0,61	10,27 + 1,42n	0,8	
MTR	FNC 52 /S /D1 /D2 /n	16	5,07	1,65	5,84	1,85	
HSCS	FNC 53 /S1+ /S2+ /D+ ⑪	32	10,8	0,61	12,8	0,8	
HSCR	FNC 54 /S1+ /S2+ /D+ ⑪	32	10,79	0,61	12,71	0,8	
HSZ	FNC 55 /S1+ /S2+ /S+ /D+ ⑪	32	12,2	0,61	14,86	0,8	
SPD	FNC 56 /S1+ /S2+ /D+	16	15,51	3,08	17,25	3,27	
		32	15,86	3,12	18,01	3,31	
PLSY	FNC 57 /S1+ /S2+ /D+	16	38,21	4,63	40,1	5,5	
		32	38,61	4,78	41,33	5,36	
PWM	FNC 58 /S1+ /S2+ /D+	16	10,09	8,11	12,08	8,5	
PLSR	FNC 59 /S1+ /S2+ /S3+ /D+	16	133,41	4,61	134,7	5,3	
		32		4,66	136,7	5,33	

Tab. B-34: Ausführungszeiten für High-Speed-Anweisungen bei einer FX3G/FX3GC/FX3GE

B.3.8 Anwendungsbezogene Anweisungen

Anweisung		Ausführungszeit [μ s]					
Symbol	FNC-Anweisung	Bit	Standard-Modus		Erweiterter Modus		
			EIN-Schaltzeit	AUS-Schaltzeit	EIN-Schaltzeit	AUS-Schaltzeit	
IST	FNC 60 /S /D1 /D2	16	23,33	0,61	25,3	0,8	
SER	FNC 61 ⑫	16	21,33 + 3,97n	2,4	23,99 + 3,96n	2,6	
		32	22,14 + 4,29n	2,3	25,7 + 4,29n	2,5	
ABSD	FNC 62 /S1+ /S2+ /D+ /n ⑬	16	14,72 + 2,25n	0,61	17,04 + 2,25n	0,8	
		32	15,22 + 3,44n		17,96 + 3,44n		
INCD	FNC 63 /S1+ /S2+ /D+ /n	16	18,68	3,1	20,98	3,29	
ALT	FNC 66 /D+	16	5,82	0,61	6,62	0,8	
RAMP	FNC 67 /S1+ /S2+ /D+ /n	16	12,33	9,89	15,21	12,37	

Tab. B-35: Ausführungszeiten für anwendungsbezogene Anweisungen bei einer FX3G/FX3GC/FX3GE

B.3.9 Spezielle Anweisungen

Anweisung		Ausführungszeit [μ s]				
Symbol	FNC-Anweisung	Bit	Standard-Modus		Erweiterter Modus	
			EIN-Schaltzeit	AUS-Schaltzeit	EIN-Schaltzeit	AUS-Schaltzeit
DSW	FNC 72 /S+ /D1+ /D2+ /n	16	24,74	18	27,08	20,3
SEGL	FNC 74 /S+ /D+ /n	16 1Set	17,76	6,27	19,34	6,41
FROM ^①	FNC 78 /n1 /n2 /D+ /n3 ¹⁴	16	115,6 + 454,4n	0,61	117,29 + 454,74n	0,8
		32	119,14 + 904,86n		121,98 + 905,03n	
FROM ^②		16	114,98 + 968,02n	0,61	117,81 + 968,19n	0,8
		32	119,62 + 1931,4n		123,62 + 1931,4n	
FROM ^③		16	52,58 + 131,43n	0,61	55,58 + 131,43n	0,8
		32	11,67 + 242,33n		14,67 + 242,34n	
TO ^①	FNC 79 /n1 /n2 /S+ /n3 ¹⁴	16	74,84 + 515,17n	0,61	76,67 + 515,34n	0,8
		32	78,31 + 1025,7n		82,39 + 1025,7n	
TO ^②		16	77,62 + 1025,4n	0,61	81,62 + 1025,4n	0,8
		32	78,93 + 2047,1n		82,93 + 2047,1n	
TO ^③		16	129,9 + 134,1n	0,61	132,91 + 134,1n	0,8
		32	13,67 + 350,33n		18,58 + 350,43n	
RS	FNC 80	16	17,21	2,52	19,99	2,91
PRUN	FNC 81 /S+ /D+ ¹⁵	16	11,41 + 1,67n	0,61	12,92 + 1,6n	0,8
		32	11,65 + 1,63n		12,87 + 1,65n	
ASCI	FNC 82	16	13,34 + 1,52n	0,61	14,07 + 1,52n	0,8
HEX	FNC 83	16	11,89 + 2,66n	0,61	13,66 + 2,66n	0,8
CCD	FNC 84	16	11,98 + 1,16n	0,61	13,7 + 1,16n	0,8
VRRD	FNC 85 /S+ /D+	16	102,46	0,61	103,88	0,8
VRSC	FNC 86 /S+ /D+	16	102,46	0,61	103,88	0,8
RS2	FNC87	16	23,48	2,68	26,52	2,92
PID	FNC 88	16	29,1	15,41	31,5	18,005

Tab. B-36: Ausführungszeiten für spezielle Anweisungen bei einer FX3G/FX3GC/FX3GE (1)

- ① Diese Zeiten gelten, wenn mit dieser Anweisung auf die Pufferspeicheradressen 0 bis 31 eines Sondermoduls der FX2N-Serie zugegriffen wird.
- ② Diese Zeiten gelten, wenn mit dieser Anweisung auf den Pufferspeicher eines Sondermoduls der FX2N-Serie ab der Adresse 32 zugegriffen wird.
- ③ Diese Zeiten gelten, wenn mit dieser Anweisung auf den Pufferspeicher eines Sondermoduls der FX3U- oder FX3UC-Serie zugegriffen wird.

Anweisung		Ausführungszeit [μ s]					
Symbol	FNC-Anweisung	Bit	Standard-Modus		Erweiterter Modus		
			EIN-Schaltzeit	AUS-Schaltzeit	EIN-Schaltzeit	AUS-Schaltzeit	
ECMP	FNC 110 /S1+ /S2+ /D+	32	12,46	0,61	14,68	0,8	
EMOV	FNC 112 /S+ /D+	32	6,36	0,61	7,68	0,8	
EADD	FNC 120 /S1+ /S2+ /D+	32	11,96	0,61	13,96	0,8	
ESUB	FNC 121 /S1+ /S2+ /D+	32	12,06	0,61	13,98	0,8	
EMUL	FNC 122 /S1+ /S2+ /D+	32	12,76	0,61	14,68	0,8	
EDIV	FNC 123 /S1+ /S2+ /D+	32	20,86	0,61	22,68	0,8	
ESQR	FNC 127 /S+ /D+	32	10,66	0,61	11,88	0,8	
INT	FNC 129 /S+ /D+	16	12,06	0,61	13,28	0,8	
		32					
DSZR	FNC 150 /S1+ /S2+ /D1+ /D2+	16	166,48	4,56	168,92	4,96	
TBL	FNC 152 /D+ /n	—	①	0,61	②	0,8	
ABS	FNC 155 /S+ /D1+ /D2+	32	16,68	3,68	18,52	3,88	
ZRN	FNC 156 /S1+ /S2+ /S3+ /D+	16	140,48	4,68	142,92	5,08	
		32	141,48		143,92		
PLSV	FNC 157 /S+ /D1+ /D2+	16	160,48	4,64	161,92	5,04	
		32			162,92	5	
DRVI	FNC 158 /S1+ /S2+ /D1+ /D2+	16	249,48	4,78	250,92	5,12	
		32			252,92		
DRVA	FNC 159 /S1+ /S2+ /D1+ /D2+	16	249,46	4,74	252,92	5,16	
		32					
TCMP	FNC 160 /S1+ /S2+ /S3+ /S+ /D+	16	15,07	0,61	17,57	0,8	
TZCP	FNC 161 /S1+ /S2+ /S3+ /D+	16	17,84	0,61	20,33	0,8	
TADD	FNC 162 /S1+ /S2+ /D+	16	11,98	0,61	13,91	0,8	
TSUB	FNC 163 /S1+ /S2+ /D+	16	11,98	0,61	13,94	0,8	
TRD	FNC 166 /D+	16	6,76	0,61	7,52	0,8	
TWR	FNC 167 /S+	16	337,1	0,61	339,6	0,8	
HOOR	FNC 169 /S+ /D1+ /D2+	16	10,86	10,44	12,59	12,17	
		32	11,57		13,69	13,01	
GRY	FNC 170 /S+	16	11,46	0,61	12,62	0,8	
		32	11,81		13,35		
GBIN	FNC 171 /D+	16	11,46	0,61	12,62	0,8	
		32	11,83		13,37		
RD3A	FNC 176 /M1+ /M2+ /D+	16	1939 (FX2N-2AD)	0,61	1941 (FX2N-2AD)	0,8	
WR3A	FNC 177 /M1+ /M2+ /D+	16	3086 (FX2N-2DA)	0,61	3089 (FX2N-2DA)	0,8	

Tab. B-37: Ausführungszeiten für spezielle Anweisungen bei einer FX3G/FX3GC/FX3GE (2)

- ① Die Ausführungszeit hängt davon ab, mit welcher Positionieranweisung eine TBL-Anweisung kombiniert wird:
DPLSV-Anweisung: 160,48 μ s, DDRVI-Anweisung: 249,48 μ s, DRRVA-Anweisung: 249,46 μ s.
- ② Die Ausführungszeit hängt davon ab, mit welcher Positionieranweisung eine TBL-Anweisung kombiniert wird:
DPLSV-Anweisung: 162,92 μ s, DDRVI-Anweisung: 252,92 μ s, DRRVA-Anweisung: 252,92 μ s.

Anweisung		Ausführungszeit [μ s]				
Symbol	FNC-Anweisung	Bit	Standard-Modus		Erweiterter Modus	
			EIN-Schaltzeit	AUS-Schaltzeit	EIN-Schaltzeit	AUS-Schaltzeit
LD=	FNC 224 /S1+ /S2+	16	1,56	1,52	2,28	2,24
		32	1,8	1,76	2,72	2,68
LD>	FNC 225 /S1+ /S2+	16	1,92	1,92	2,68	2,68
		32	2,36	2,32	3,24	3,24
LD<	FNC 226 /S1+ /S2+	16	1,96	1,88	2,72	2,64
		32	2,4	2,28	3,28	3,2
LD<>	FNC 228 /S1+ /S2+	16	1,52	1,56	2,24	2,28
		32	1,76	1,8	2,68	2,72
LD<=	FNC 229 /S1+ /S2+	16	1,84	1,96	2,6	2,68
		32	2,28	2,4	3,16	3,24
LD>=	FNC 230 /S1+ /S2+	16	1,84	1,92	2,6	2,72
		32	2,28	2,36	3,16	3,28
AND=	FNC 232 /S1+ /S2+	16	1,52	1,46	2,28	2,24
		32	1,76	1,7	2,96	2,64
AND>	FNC 233 /S1+ /S2+	16	1,88	1,84	2,6	2,6
		32	2,24	2,28	3,2	3,2
AND<	FNC 234 /S1+ /S2+	16	1,92	1,84	2,68	2,56
		32	2,36	2,18	3,28	3,16
AND<>	FNC 236 /S1+ /S2+	16	1,48	1,5	2,24	2,24
		32	1,68	1,7	2,64	2,64
AND<=	FNC 237 /S1+ /S2+	16	1,8	1,88	2,56	2,68
		32	2,2	2,26	3,16	3,24
AND>=	FNC 238 /S1+ /S2+	16	1,8	1,92	2,52	2,64
		32	2,16	2,32	3,12	3,24
OR=	FNC 240 /S1+ /S2+	16	1,54	1,52	2,28	2,24
		32	1,76	1,72	2,68	2,64
OR>	FNC 241 /S1+ /S2+	16	1,92	1,88	2,64	2,64
		32	2,32	2,32	3,24	3,24
OR<	FNC 242 /S1+ /S2+	16	1,96	1,84	2,68	2,6
		32	2,36	2,28	3,28	3,2
OR<>	FNC 244 /S1+ /S2+	16	1,52	1,56	2,24	2,28
		32	1,72	1,76	2,64	2,68
OR<=	FNC 245 /S1+ /S2+	16	1,84	1,92	2,56	2,64
		32	2,2	2,32	3,16	3,24
OR>=	FNC 246 /S1+ /S2+	16	1,84	1,96	2,56	2,68
		32	2,2	2,36	3,16	3,28
IVCK	FNC 270 /S1+ /S2+ /D+ /n	16	10,06	4,46	10,88	4,88
IVDR	FNC 271 /S1+ /S2+ /S3+ /n	16	10,06	4,46	10,88	4,88
IVRD	FNC 272 /S1+ /S2+ /D+ /n	16	10,06	4,46	10,88	4,88
IVWR	FNC 273 /S1+ /S2+ /S3+ /n	16	10,06	4,46	10,88	4,88
IVMC	FNC 275 /S1+ /S2+ /S3+ /D+ /n	16	44,2	4,46	45,4	4,88
ADPRW	FNC 276 /S1+ /S2+ /S3+ /S4+ /D+	16	68,15	8,83	71,48	9,77
LOADR	FNC 290 /S+ /n	16	107,23 + 30,25n	0,61	108,59 + 30,33n	0,8

Tab. B-38: Ausführungszeiten für spezielle Anweisungen bei einer FX3G/FX3GC/FX3GE (3)

HINWEIS

Weitere Angaben zu den Ausführungszeiten der LD□-, AND□- und OR□-Anweisungen finden Sie im Abschnitt B.7.2.

Anweisung		Ausführungszeit [μ s]				
Symbol	FNC-Anweisung	Bit	Standard-Modus		Erweiterter Modus	
			EIN-Schaltzeit	AUS-Schaltzeit	EIN-Schaltzeit	AUS-Schaltzeit
RWER	FNC 294 /S+ /n	16	4692.4 + 128.57n n: Anzahl Sektoren	0,61	4700.4 + 128.57n n: Anzahl Sektoren	0,8

Tab. B-39: Ausführungszeiten für spezielle Anweisungen bei einer FX3G/FX3GC/FX3GE (3)

- ① „n“ gibt die Anzahl der statischen STL-Anweisungen (Anzahl der parallelen/zusammenführenden Anweisungen) an.
- ② Die gekennzeichneten Anweisungen benötigen keine Kontakte.
- ③ Wird die statische Anweisung eingesetzt und nicht die Pulsanweisung, verändert sich der Wert der Zieladresse zyklisch.
- ④ „n“ gibt die Anzahl der zu verschiebenden Register an ($n \leq 512$).
- ⑤ „n“ gibt die Anzahl der zu verarbeitenden Bit-Operanden an ($n \leq 16$ im 16-Bit-Modus, $n \leq 32$ im 32-Bit-Modus).
- ⑥ „n“ gibt die Anzahl der zu verarbeitenden Bit-Operanden an.
- ⑦ „n“ gibt die Anzahl der zu verarbeitenden Operanden an ($2 \leq n \leq 512$).
- ⑧ „n“ gibt den zurückzusetzenden Operandenbereich an. Der Operandentyp ist in den Klammern angegeben.
- ⑨ „n“ gibt die Operandenadressen an, die mit der MEAN-Anweisung verarbeitet werden ($1 \leq n \leq 64$).
- ⑩ „n“ gibt den zu aktualisierenden Operandenbereich an ($8 \leq n \leq 128$ in 8-ter-Schritten).
- ⑪ Diese Anweisung kann max. 6 mal zur gleichen Zeit aktiv sein.
- ⑫ „n“ gibt die Anzahl der Stapel-Elemente an ($n \leq 128$).
- ⑬ „n“ gibt die Anzahl der Ausgangsadressen an ($n \leq 64$).
- ⑭ „n“ gibt die Anzahl der in das/aus dem Sondermodul zu schreibenden oder lesenden Datenworte an.
- ⑮ „n“ gibt die Anzahl der zu schreibenden oder lesenden Single-Byte-Datenworte (8 Bit) bei Parallelbetrieb zweier FX-Steuerungen an.

B.4 Ausführungszeiten der FX3S-Serie

HINWEISE

| Eine Beschreibung der Fußnoten (①, ②, etc.) befindet sich am Ende des Abschnitts.

| Werden Applikationsanweisungen als Puls-Anweisungen programmiert (Anhängen eines „P“, z. B. MOVP), gelten andere Ausführungszeiten (siehe Abschnitt B.6).

B.4.1 Grundbefehle und Schrittstatus-Anweisungen

Anweisung	Bedeutung	Operanden	Programmschritte	Ausführungszeiten [µs]	
				EIN-Schaltzeit	AUS-Schaltzeit
LD	Beginn einer Verknüpfung, Abfrage auf „1“-Signal	X, Y, M, S, T, C, Sondermerker	1 oder 2 (siehe Abschnitt 4.1.2)	0,21 / 1,34 ^①	
LDI	Beginn einer Verknüpfung, Abfrage auf „0“-Signal				
AND	UND-Verknüpfung, Abfrage auf „1“-Signal			0,2 / 1,16 ^①	
ANI	UND-Verknüpfung, Abfrage auf „0“-Signal				
OR	ODER-Verknüpfung, Abfrage auf „1“-Signal			0,21 / 1,28 ^①	
ORI	ODER-Verknüpfung, Abfrage auf „0“-Signal				
LDP	Lade (gepulst); Beginn einer Verknüpfung mit Abfrage der ansteigenden Flanke	X, Y, M, S, T, C	2	2,32	2,28
LDF	Lade (gepulst); Beginn einer Verknüpfung mit Abfrage der abfallenden Flanke			2,59	2,32
ANDP	UND (gepulst); UND-Verknüpfung mit Abfrage der ansteigenden Flanke			2,36	2,32
ANDF	UND (gepulst); UND-Verknüpfung mit Abfrage der abfallenden Flanke			2,4	2,36
ORP	ODER (gepulst); ODER-Verknüpfung mit Abfrage der ansteigenden Flanke			2,36	2,36
ORF	ODER (gepulst); ODER-Verknüpfung mit Abfrage der abfallenden Flanke			2,4	2,36
ANB	UND-Block, Reihenschaltung von Parallelverknüpfungen			—	1
ORB	ODER-Block, Parallelschaltung von Reihenverknüpfungen	0,2			

Tab. B-40: Ausführungszeiten für Grundbefehle und Schrittstatus-Anweisungen bei einer Steuerung der FX3S-Serie (1)

① Bei Ausführung der Anweisung in 2 Schritten.

Anweisung	Bedeutung	Operanden	Programm- schritte	Ausführungszeiten [μ s]	
				EIN- Schaltzeit	AUS- Schaltzeit
MPS	Abspeichern eines Verknüpfungsergebnisses	—	1	0,21	
MRD	Lesen eines Verknüpfungsergebnisses			0,2	
MPP	Lesen und Löschen des Verknüpfungsspeichers			0,17	
INV	Inversion, Umkehrung von Verarbeitungsergebnissen			0,17	
MEP	Impuls erzeugen, wenn das Verknüpfungsergebnis „1“ wird	—	1	1,52	
MEF	Impuls erzeugen, wenn das Verknüpfungsergebnis „0“ wird	—	1	1,6	
MC	Setzen einer Kontrollbedingung	N - Y, M	3 oder 4 (Abschnitt 4.1.2)	1,63	1,76
MCR	Rücksetzen einer Kontrollbedingung	N (Nesting)	2	1,3	
NOP	Leerzeile	—	1	0,21	
END	Programmende		1	209 Falls eine FEND- und eine END-Anweisung zusammen verwendet werden, ist nur die Ausführungszeit der END-Anweisung relevant.	
STL	Schrittstatus ausführen	S	1	3,36 + 1,19n ①	
RET	Schrittstatus beenden	—	1	2,19	
OUT	Ausgabe, Zuweisung eines Verknüpfungsergebnisses	Y	1	0,21	
		M	1 oder 2 (siehe Abschnitt 4.1.2)	0,21 / 1,2 ^①	
		S	2	1,25	
		T-K	3	2,41	2,16
		T-D	3	2,77	2,54
		C-K (16 Bit)	3	1,9	1,57
		C-K (32 Bit)	5	2,25	
		C-K (C235 – C255)		1,6	1,06
		C-D (16 Bit)	3	2,26	1,93
		C-D (32 Bit)	5	2,49	
C-D (C235 – C255)	2,21	1,06			
SET	Setzen, Operanden setzen	Y	1	0,23	
		M	1 oder 2 (siehe Abschnitt 4.1.2)	0,23 / 0,72 ^①	
		S	2	1,23	0,59
		S bei Einsatz in STL-Stufe ①		1,33 + 0,51n (1,23 wenn n = 0)	

Tab. B-41: Ausführungszeiten für Grundbefehle und Schrittstatus-Anweisungen bei einer Steuerung der FX3S-Serie (2)

^① Bei Ausführung der Anweisung in 2 Schritten.

Anweisung	Bedeutung	Operanden	Programmschritte	Ausführungszeiten [µs]	
				EIN-Schaltzeit	AUS-Schaltzeit
RST	Rücksetzen, Operanden rücksetzen	Y	1	0,23	
		M	1, 2 oder 3 (siehe Abschnitt 4.1.2)	0,23 / 1,22 ^①	
		S	2	1,17	0,59
		T	2	1,68	1,21
		C (16 Bit)	2	1,77	1,18
		C (32 Bit)	2	1,82	1,19
		D, V, Z, Sonderregister	3	1,08	0,54
PLS	Impulserzeugung bei ansteigender Flanke	Y, M	2 oder 3 (siehe Abschnitt 4.1.2)	1,52	
PLF	Impulserzeugung bei abfallender Flanke				

Tab. B-43: Ausführungszeiten für Grundbefehle und Schrittstatus-Anweisungen bei einer Steuerung der FX3S-Serie (3)

① Bei Ausführung der Anweisung in 2 Schritten.

B.4.2 Programmverzweigungsanweisungen

Anweisung			Ausführungszeit [µs]	
Symbol	FNC-Anweisung		EIN-Schaltzeit	AUS-Schaltzeit
CJ	FNC 00 /S+		2,59	0,61
CALL	FNC 01 /S+		6,44	0,61
SRET	FNC 02 ②			—
IRET	FNC 03 ②		6,34	—
EI	FNC 04 ②		2,52	—
DI	FNC 05 ②		1,94	—
FEND	FNC 06 ②		209 Falls eine FEND- und eine END-Anweisung zusammen verwendet werden, ist nur die Ausführungszeit der END-Anweisung relevant.	
WDT	FNC 07		1,94	0,61
FOR	FNC 08 /S+ ②		4,98	—
NEXT	FNC 09 ②			—

Tab. B-42: Ausführungszeiten für Programmverzweigungsanweisungen bei einer FX3S

B.4.3 Vergleichs- und Transferanweisungen

Anweisung		Ausführungszeit [μ s]		
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit
CMP	FNC 10 /S1+ /S2+ /D+	16	11,18	0,61
		32	11,54	
ZCP	FNC 11 /S1+ /S2+ /S+ /D+	16	12,51	0,61
		32	13,15	
MOV	FNC 12 /S+ /D+	16	0,52	0,38
		32	0,61	0,41
SMOV	FNC 13 /S+ /m1 /m2 /D+ /n	16	16,19	0,61
CML	FNC 14 /S+ /D+	16	6,68	0,61
		32	7,15	
BMOV	FNC 15 /S+ /D+ /n ④	16	15,95 + 3,96n	0,61
FMOV	FNC 16 /S+ /D+ /n ④	16	10,20 + 0,77n	0,61
		32	10,87 + 0,79n	
BCD	FNC 18 /S+ /D+	16	2,93	0,41
		32	4,6	0,43
BIN	FNC 19 /S+ /D+	16	2,93	0,41
		32	4,61	0,43

Tab. B-44: Ausführungszeiten für Vergleichs- und Transferanweisungen bei einer FX3S

B.4.4 Arithmetische Anweisungen

Anweisung		Ausführungszeit [μ s]		
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit
ADD	FNC 20 /S1+ /S2+ /D+ Z, Cy, Br	16	1,79	0,41
		32	2,13	0,45
SUB	FNC 21 /S1+ /S2+ /D+ Z, Cy, Br	16	1,79	0,41
		32	2,13	0,45
MUL	FNC 22 /S1+ /S2+ /D+	16	1,29	0,41
		32	3,21	0,45
DIV	FNC 23 /S1+ /S2+ /D+	16	1,93	0,41
		32	3,7	0,45
INC	FNC 24 /D+ ③	16	0,78	0,41
		32	1,01	
DEC	FNC 25 /D+ ③	16	0,78	0,41
		32	1,01	
WAND	FNC 26 /S1+ /S2+ /D+	16	1,24	0,41
		32	1,59	0,45
WOR	FNC 27 /S1+ /S2+ /D+	16	1,24	0,41
		32	1,59	0,45
WXOR	FNC 28 /S1+ /S2+ /D+	16	1,24	0,41
		32	1,59	0,45

Tab. B-45: Ausführungszeiten für arithmetische Anweisungen bei einer FX3S

Z (Zero): M8020
 Br (Borrow): M8021
 Cy (Carry): M8022
 F (Anweisung vollständig abgearbeitet): M8029

B.4.5 Rotations- und Shift-Anweisungen

Anweisung		Ausführungszeit [μ s]		
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit
ROR	FNC 30 /D+ /n Cy ③ ⑤	16	7,37	0,61
		32	8,94	
ROL	FNC 31 /D+ /n Cy ③ ⑤	16	7,37	0,61
		32	8,94	
SFTR	FNC 34 /S+ /D+ /n1 /n2 ③ ⑥	16	14,51 + 0,46n	0,61
SFTL	FNC 35 /S+ /D+ /n1 /n2 ③ ⑥	16	14,7 + 0,46n	0,61
WSFR	FNC 36 /S+ /D+ /n1 /n2 ③ ④	16	11,74 + 1,44n	0,61
WSFL	FNC 37 /S+ /D+ /n1 /n2 ③ ④	16	12,02 + 1,48n	0,61
SFWR	FNC 38 /S+ /D+ /n ③ ⑦	16	7,46	0,61
SFRD	FNC 39 /S+ /D+ /n ③ ⑦	16	9,27	0,61

Tab. B-46: Ausführungszeiten für Rotations- und Shift-Anweisungen bei einer FX3S

B.4.6 Datenoperationen

Anweisung		Ausführungszeit [μ s]		
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit
ZRST	FNC 40 /D1+ /D2+ ⑧	16 (D)	7,96 + 0,22n	0,61
		16 (T)	8,77 + 0,46n	
		16 (M)	14,76 + 0,37n	
DECO	FNC 41 /S+ /D+ /n	16	8,5	0,61
ENCO	FNC 42 /S+ /D+ /n	16	9,29	0,61
SUM	FNC 43 /S+ /D+	16	6,95	0,61
		32	7,43	
BON	FNC 44 /S+ /D+ /n	16	10,4	0,61
		32	10,77	
MEAN	FNC 45 /S+ /D+ /n ⑨	16	13,83 + 3,33n	0,61
		32	14,69 + 3,5n	
FLT	FNC 49	16	6,96	0,61
		32	7,56	

Tab. B-47: Ausführungszeiten für Datenoperationen bei einer FX3S

B.4.7 High-Speed-Anweisungen

Anweisung		Ausführungszeit [μ s]		
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit
REF	FNC 50 /D /n ⑩	16	X: 8,75 + 0,45n Y: 8,85 + 0,90n	0,61
MTR	FNC 52 /S /D1 /D2 /n	16	14,72	1,65
HSCS	FNC 53 /S1+ /S2+ /D+ ⑪	32	10,8	0,61
HSCR	FNC 54 /S1+ /S2+ /D+ ⑪	32	10,79	0,61
HSZ	FNC 55 /S1+ /S2+ /S+ /D+ ⑪	32	12,2	0,61
SPD	FNC 56 /S1+ /S2+ /D+	16	15,51	3,08
		32	15,86	3,12
PLSY	FNC 57 /S1+ /S2+ /D+	16	38,21	6,23
		32	38,61	
PWM	FNC 58 /S1+ /S2+ /D+	16	21,79	8,11
PLSR	FNC 59 /S1+ /S2+ /S3+ /D+	16	156,25	6,19
		32	157,02	

Tab. B-48: Ausführungszeiten für High-Speed-Anweisungen bei einer FX3s

B.4.8 Anwendungsbezogene Anweisungen

Anweisung		Ausführungszeit [μ s]		
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit
IST	FNC 60 /S /D1 /D2	16	23,33	0,61
SER	FNC 61 ⑫	16	21,33 + 3,97n	2,4
		32	22,14 + 4,29n	2,3
ABSD	FNC 62 /S1+ /S2+ /D+ /n ⑬	16	14,72 + 2,25n	0,61
		32	15,22 + 3,44n	
INCD	FNC 63 /S1+ /S2+ /D+ /n	16	23,69	3,04
ALT	FNC 66 /D+	16	5,82	0,61
RAMP	FNC 67 /S1+ /S2+ /D+ /n	16	12,33	9,89

Tab. B-49: Ausführungszeiten für anwendungsbezogene Anweisungen bei einer FX3s

B.4.9 Spezielle Anweisungen

Anweisung		Ausführungszeit [μ s]		
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit
DSW	FNC 72 /S+ /D1+ /D2+ /n	16	24,74	18
SEGL	FNC 74 /S+ /D+ /n	16	17,76	6,27
RS	FNC 80	16	17,21	2,52
PRUN	FNC 81 /S+ /D+ ¹⁴	16	11,41 + 1,67n	0,61
		32	11,65 + 1,63n	
ASCI	FNC 82	16	12,34 + 1,52n	0,61
HEX	FNC 83	16	11,89 + 2,66n	0,61
CCD	FNC 84	16	11,98 + 1,16n	0,61
VRRD	FNC 85 /S+ /D+	16	102,46	0,61
VRSC	FNC 86 /S+ /D+	16	102,46	0,61
RS2	FNC 87	16	23,48	2,68
PID	FNC 88	16	29,1	15,41
ECMP	FNC 110 /S1+ /S2+ /D+	32	12,46	0,61
EMOV	FNC 112 /S+ /D+	32	6,36	0,61
EADD	FNC 120 /S1+ /S2+ /D+	32	11,96	0,61
ESUB	FNC 121 /S1+ /S2+ /D+	32	12,06	0,61
EMUL	FNC 122 /S1+ /S2+ /D+	32	12,76	0,61
EDIV	FNC 123 /S1+ /S2+ /D+	32	20,86	0,61
ESQR	FNC 127 /S+ /D+	32	10,66	0,61
INT	FNC 129 /S+ /D+	16	12,06	0,61
		32		
DSZR	FNC 150 /S1+ /S2+ /D1+ /D2+	16	166,48	6,19
ABS	FNC 155 /S+ /D1+ /D2+	32	16,68	3,68
ZRN	FNC 156 /S1+ /S2+ /S3+ /D+	16	151,38	6,17
		32	151,63	
PLSV	FNC 157 /S+ /D1+ /D2+	16	160,48	6,1
		32		
DRVI	FNC 158 /S1+ /S2+ /D1+ /D2+	16	275,38	6,04
		32	275,94	
DRVA	FNC 159 /S1+ /S2+ /D1+ /D2+	16	276,08	6,23
		32	276,65	
TCMP	FNC 160 /S1+ /S2+ /S3+ /S+ /D+	16	15,07	0,61
TZCP	FNC 161 /S1+ /S2+ /S3+ /D+	16	17,84	0,61
TADD	FNC 162 /S1+ /S2+ /D+	16	11,98	0,61
TSUB	FNC 163 /S1+ /S2+ /D+	16	11,98	0,61
TRD	FNC 166 /D+	16	6,76	0,61
TWR	FNC 167 /S+	16	337,1	0,61
HOUR	FNC 169 /S+ /D1+ /D2+	16	10,86	10,44
		32	11,57	
GRY	FNC 170 /S+	16	11,46	0,61
		32	11,81	
GBIN	FNC 171 /D+	16	11,46	0,61
		32	11,83	

Tab. B-50: Ausführungszeiten für spezielle Anweisungen bei einer FX3S (1)

Anweisung		Ausführungszeit [μ s]		
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit
LD=	FNC 224 /S1+ /S2+	16	1,56	1,52
		32	1,8	1,76
LD>	FNC 225 /S1+ /S2+	16	1,92	1,92
		32	2,36	2,32
LD<	FNC 226 /S1+ /S2+	16	1,96	1,88
		32	2,4	2,28
LD<>	FNC 228 /S1+ /S2+	16	1,52	1,56
		32	1,76	1,8
LD<=	FNC 229 /S1+ /S2+	16	1,84	1,96
		32	2,28	2,4
LD>=	FNC 230 /S1+ /S2+	16	1,84	1,92
		32	2,28	2,36
AND=	FNC 232 /S1+ /S2+	16	1,52	1,46
		32	1,76	1,7
AND>	FNC 233 /S1+ /S2+	16	1,88	1,84
		32	2,24	2,28
AND<	FNC 234 /S1+ /S2+	16	1,92	1,84
		32	2,36	2,18
AND<>	FNC 236 /S1+ /S2+	16	1,48	1,5
		32	1,68	1,7
AND<=	FNC 237 /S1+ /S2+	16	1,8	1,88
		32	2,2	2,26
AND>=	FNC 238 /S1+ /S2+	16	1,8	1,92
		32	2,16	2,32
OR=	FNC 240 /S1+ /S2+	16	1,54	1,52
		32	1,76	1,72
OR>	FNC 241 /S1+ /S2+	16	1,92	1,88
		32	2,32	2,32
OR<	FNC 242 /S1+ /S2+	16	1,96	1,84
		32	2,36	2,28
OR<>	FNC 244 /S1+ /S2+	16	1,52	1,56
		32	1,72	1,76
OR<=	FNC 245 /S1+ /S2+	16	1,84	1,92
		32	2,2	2,32
OR>=	FNC 246 /S1+ /S2+	16	1,84	1,96
		32	2,2	2,36
IVCK	FNC 270 /S1+ /S2+ /D+ /n	16	32,6	4,46
IVDR	FNC 271 /S1+ /S2+ /S3+ /n	16	31,6	4,46
IVRD	FNC 272 /S1+ /S2+ /D+ /n	16	35,3	4,46
IVWR	FNC 273 /S1+ /S2+ /S3+ /n	16	34,6	4,46
IVMC	FNC 275 /S1+ /S2+ /S3+ /D+ /n	16	44,2	4,46
ADPRW	FNC 276 /S1+ /S2+ /S3+ /S4+(/D+)	16	67,19	9,83

Tab. B-51: Ausführungszeiten für spezielle Anweisungen bei einer FX3S (2)

- ① „n“ gibt die Anzahl der statischen STL-Anweisungen (Anzahl der parallelen/zusammenführenden Anweisungen) an.
- ② Die gekennzeichneten Anweisungen benötigen keine Kontakte.
- ③ Wird die statische Anweisung eingesetzt und nicht die Pulsanweisung, verändert sich der Wert der Zieladresse zyklisch.
- ④ „n“ gibt die Anzahl der zu verschiebenden Register an ($n \leq 512$).
- ⑤ „n“ gibt die Anzahl der zu verarbeitenden Bit-Operanden an ($n \leq 16$ im 16-Bit-Modus, $n \leq 32$ im 32-Bit-Modus).
- ⑥ „n“ gibt die Anzahl der zu verarbeitenden Bit-Operanden an.
- ⑦ „n“ gibt die Anzahl der zu verarbeitenden Operanden an ($2 \leq n \leq 512$).
- ⑧ „n“ gibt den zurückzusetzenden Operandenbereich an. Der Operandentyp ist in den Klammern angegeben.
- ⑨ „n“ gibt die Operandenadressen an, die mit der MEAN-Anweisung verarbeitet werden ($1 \leq n \leq 64$).
- ⑩ „n“ gibt den zu aktualisierenden Operandenbereich an ($8 \leq n \leq 128$ in 8-ter-Schritten).
- ⑪ „n“ gibt die Anzahl der Stapel-Elemente an ($n \leq 128$).
- ⑫ „n“ gibt die Anzahl der Ausgangsadressen an ($n \leq 64$).
- ⑬ „n“ gibt die Anzahl der in das/aus dem Sondermodul zu schreibenden oder lesenden Datenworte an.
- ⑭ „n“ gibt die Anzahl der zu schreibenden oder lesenden Single-Byte-Datenworte (8 Bit) bei Parallelbetrieb zweier FX-Steuerungen an.

B.5 Ausführungszeiten der FX3U-/FX3UC-Serie

HINWEISE

Eine Beschreibung der Fußnoten (❶, ❷, etc.) befindet sich am Ende des Abschnitts.

Werden Applikationsanweisungen als Puls-Anweisungen programmiert (Anhängen eines „P“, z. B. MOVP), gelten andere Ausführungszeiten (siehe Abschnitt B.6).

B.5.1 Grundbefehle und Schrittstatus-Anweisungen

Anweisung	Bedeutung	Operanden	Programmschritte	Ausführungszeiten [μs]	
				EIN-Schaltzeit	AUS-Schaltzeit
LD	Beginn einer Verknüpfung, Abfrage auf „1“-Signal	X, Y, M, S, T, C, Sondermerker, D□.b	1, 2 oder 3 (siehe Abschnitt 4.1.2)	0,065 (0,129 μs beim Zugriff auf die Operanden M1536 bis M3583, M8256 bis M8511 und S1024 bis S4095) (0,193 μs beim Zugriff auf die Merker M3584 bis M7679) Angaben zu den Ausführungszeiten dieser Anweisungen bei Index-Vergabe finden Sie im Abschnitt B.7.1.	
LDI	Beginn einer Verknüpfung, Abfrage auf „0“-Signal				
AND	UND-Verknüpfung, Abfrage auf „1“-Signal				
ANI	UND-Verknüpfung, Abfrage auf „0“-Signal				
OR	ODER-Verknüpfung, Abfrage auf „1“-Signal				
ORI	ODER-Verknüpfung, Abfrage auf „0“-Signal				
LDP	Lade (gepulst); Beginn einer Verknüpfung mit Abfrage der steigenden Flanke	X, Y, M, S, T, C, D□.b	2 oder 3 (Abschnitt 4.1.2)	7,8	
LDF	Lade (gepulst); Beginn einer Verknüpfung mit Abfrage der fallenden Flanke				
ANDP	UND (gepulst); UND-Verknüpfung mit Abfrage der ansteigenden Flanke				
ANDF	UND (gepulst); UND-Verknüpfung mit Abfrage der abfallenden Flanke				
ORP	ODER (gepulst); ODER-Verknüpfung mit Abfrage der ansteigenden Flanke				
ORF	ODER (gepulst); ODER-Verknüpfung mit Abfrage der abfallenden Flanke	—	1	0,065	
ANB	UND-Block, Reihenschaltung von Parallelverknüpfungen				
ORB	ODER-Block, Parallelschaltung von Reihenverknüpfungen				
MPS	Speichern eines Verknüpfungsergebnisses				
MRD	Lesen eines Verknüpfungsergebnisses				
MPP	Lesen und Löschen des Verknüpfungsspeichers				
INV	Inversion, Umkehrung von Verknüpfungsergebnissen	7,4			

Tab. B-52: Ausführungszeiten für Grundbefehle und Schrittstatus-Anweisungen bei einer Steuerung der FX3U- oder FX3UC-Serie (1)

Anweisung	Bedeutung	Operanden	Programmschritte	Ausführungszeiten [μ s]	
				EIN-Schaltzeit	AUS-Schaltzeit
MEP	Impuls erzeugen, wenn das Verknüpfungsergebnis „1“ wird	—	1	3,4	
MEF	Impuls erzeugen, wenn das Verknüpfungsergebnis „0“ wird	—	1	3,4	
MC	Setzen einer Kontrollbedingung	N - Y, M	3 oder 4 (siehe Abschnitt 4.1.2)	4,3	4,7
MCR	Rücksetzen einer Kontrollbedingung	N (Nesting)	2	3,9	
NOP	Leerzeile		1	0,065	
END	Programmende	—	1	113,9 + (2,13 x X) + (3,25 x Y) X: Anzahl der Eingänge Y: Anzahl der Ausgänge Falls eine FEND- und eine END-Anweisung zusammen verwendet werden, ist nur die Ausführungszeit der END-Anweisung relevant.	
STL	Schrittstatus ausführen	S	1	5,1 + 1,6n ①	
RET	Schrittstatus beenden	—	1	2,9	
OUT	Ausgabe, Zuweisung eines Verknüpfungsergebnisses	Y	1	0,065	
		M	1, 2 oder 3 (siehe Abschnitt 4.1.2)	0,065 (0,129 μ s bei den Operanden M1536 bis M3583 und M8256 bis M8511) (0,193 μ s bei den Merkern M3584 bis M7679)	
		S	2	4,8	4,8
		T-K	3	0,71	
		T-K (T192 – T199, T246 – T511)		11,6	8,2
		T-D	3	0,71	
		T-D (T192 – T199, T246 – T511)		11,6	8,2
		C-K (16 Bit)	5	0,71	
		C-K (32 Bit)		6,1	
		C-K (C235 – C255)		9,5	
		C-D (16 Bit)	5	0,71	
		C-D (32 Bit)		6,1	
		C-D (C235 – C255)		9,5	

Tab. B-53: Ausführungszeiten für Grundbefehle und Schrittstatus-Anweisungen bei einer Steuerung der FX3U- oder FX3UC-Serie (2)

Anweisung	Bedeutung	Operanden	Programm- schritte	Ausführungszeiten [μ s]	
				EIN- Schaltzeit	AUS- Schaltzeit
SET	Setzen, Operanden setzen	Y	1	0,065	
		M	1, 2 oder 3 (siehe Abschnitt 4.1.2)	0,065 (0,129 μ s bei den Operanden M1536 bis M3583 und M8256 bis M8511) (0,193 μ s bei den Merkern M3584 bis M7679)	
		S	2	4,7	0,13
		S bei Einsatz in STL-Stufe ①		6,6 + 0,9n	
RST	Rücksetzen, Operanden rücksetzen	Y	1	0,065	
		M	1, 2 oder 3 (siehe Abschnitt 4.1.2)	0,065 (0,129 μ s bei den Operanden M1536 bis M3583 und M8000 bis M8511) (0,193 μ s bei den Merkern M3584 bis M7679)	
		S	2	4,6	0,13
		T	2	0,45	0,45
		C (16 Bit)	2	0,45	0,45
		C (32 Bit)	2	5,8	4,8
		D, V, Z, Sonderregister	3	5,4	0,195
PLS	Impulserzeugung bei ansteigender Flanke	Y, M	2 oder 3 (siehe Abschnitt 4.1.2)	0,257	
PLF	Impulserzeugung bei abfallender Flanke			(0,321 μ s bei M3584 bis M7679)	

Tab. B-54: Ausführungszeiten für Grundbefehle und Schrittstatus-Anweisungen bei einer Steuerung der FX3U- oder FX3UC-Serie (3)

B.5.2 Programmverzweigungsanweisungen

Anweisung			Ausführungszeit [μ s]	
Symbol	FNC-Anweisung		EIN-Schaltzeit	AUS-Schaltzeit
CJ	FNC 00 /S+		8,0	0,195
CALL	FNC 01 /S+		13,5	0,195
SRET	FNC 02	②		—
IRET	FNC 03	②	4,4	—
EI	FNC 04	②	3,8	—
DI	FNC 05	②	3,7	—
FEND	FNC 06	②	$113,9 + (2,13 \times X) + (3,25 \times Y)$ X: Anzahl der Eingänge Y: Anzahl der Ausgänge Falls eine FEND- und eine END-Anweisung zusammen verwendet werden, ist nur die Ausführungszeit der END-Anweisung relevant.	
WDT	FNC 07		5,4	0,065
FOR	FNC 08 /S+	②	11,6	—
NEXT	FNC 09	②		—

Tab. B-56: Ausführungszeiten für Programmverzweigungsanweisungen bei einer FX3U oder FX3UC

B.5.3 Vergleichs- und Transferanweisungen

Anweisung			Ausführungszeit [μ s]		
Symbol	FNC-Anweisung		Bit	EIN-Schaltzeit	AUS-Schaltzeit
CMP	FNC 10 /S1+ /S2+ /D+		16	15,5	0,455
			32	16,0	0,845
ZCP	FNC 11 /S1+ /S2+ /S+ /D+		16	18,9	0,585
			32	19,7	1,105
MOV	FNC 12 /S+ /D+		16	0,64	0,32
			32	1,48	1,48
SMOV	FNC 13 /S+ /m1 /m2 /D+ /n		16	22,9	0,715
CML	FNC 14 /S+ /D+		16	10,6	0,325
			32	10,2	0,585
BMOV	FNC 15 /S+ /D+ /n	④	16	$13,9 + 0,44n$	0,455
FMOV	FNC 16 /S+ /D+ /n	④	16	$14,2 + 0,19n$	0,455
			32	$14,0 + 0,38n$	0,845
XCH	FNC 17 /D1+ /D2+	③	16	10,7	0,325
			32	11,4	0,585
BCD	FNC 18 /S+ /D+		16	7,94	0,325
			32	12,49	0,585
BIN	FNC 19 /S+ /D+		16	4,38	0,325
			32	5,32	0,585

Tab. B-55: Ausführungszeiten für Vergleichs- und Transferanweisungen bei einer FX3U oder FX3UC

B.5.4 Arithmetische Anweisungen

Anweisung		Ausführungszeit [μ s]		
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit
ADD	FNC 20 /S1+ /S2+ /D+ Z, Cy, Br	16	4,77	0,455
		32	5,72	0,845
SUB	FNC 21 /S1+ /S2+ /D+ Z, Cy, Br	16	4,82	0,455
		32	5,78	0,845
MUL	FNC 22 /S1+ /S2+ /D+	16	4,6	0,455
		32	5,7	0,845
DIV	FNC 23 /S1+ /S2+ /D+	16	6,3	0,455
		32	7,67	0,845
INC	FNC 24 /D+ ③	16	6,2	0,195
		32	6,4	0,325
DEC	FNC 25 /D+ ③	16	6,2	0,195
		32	6,4	0,325
WAND	FNC 26 /S1+ /S2+ /D+	16	3,57	0,455
		32	4,55	0,845
WOR	FNC 27 /S1+ /S2+ /D+	16	3,57	0,455
		32	4,55	0,845
WXOR	FNC 28 /S1+ /S2+ /D+	16	3,57	0,455
		32	4,55	0,845
NEG	FNC 29 /D+ ③	16	7,6	0,195
		32	8,0	0,325

Tab. B-57: Ausführungszeiten für arithmetische Anweisungen bei einer FX3U oder FX3UC

Z (Zero):	M8020
Br (Borrow):	M8021
Cy (Carry):	M8022
F (Anweisung vollständig abgearbeitet):	M8029

B.5.5 Rotations- und Shift-Anweisungen

Anweisung		Ausführungszeit [μ s]		
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit
ROR	FNC 30 /D+ /n Cy ③ ⑤	16	10,5	0,325
		32	11,5	0,585
ROL	FNC 31 /D+ /n Cy ③ ⑤	16	10,5	0,325
		32	11,5	0,585
RCR	FNC 32 /D+ /n Cy ③ ⑤	16	10,9	0,325
		32	11,8	0,585
RCL	FNC 33 /D+ /n Cy ③ ⑤	16	10,9	0,325
		32	11,8	0,585
SFTR	FNC 34 /S+ /D+ /n1 /n2 ③ ⑥	16	23,2 + 0,08n	0,585
SFTL	FNC 35 /S+ /D+ /n1 /n2 ③ ⑥	16	23,2 + 0,08n	0,585
WSFR	FNC 36 /S+ /D+ /n1 /n2 ③ ④	16	7,5 + 0,44n	0,585
WSFL	FNC 37 /S+ /D+ /n1 /n2 ③ ④	16	7,5 + 0,44n	0,585
SFWR	FNC 38 /S+ /D+ /n ③ ⑦	16	8,1	0,455
SFRD	FNC 39 /S+ /D+ /n ③ ⑦	16	7,7	0,455

Tab. B-59: Ausführungszeiten für Rotations- und Shift-Anweisungen bei einer FX3U oder FX3UC

B.5.6 Datenoperationen

Anweisung		Ausführungszeit [μ s]		
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit
ZRST	FNC 40 /D1+ /D2+ ⑧	16 (D)	11,1 + 0,19n	0,325
		16 (T)	17,1 + 0,23n	
		16 (M)	20,7 + 0,02n	
DECO	FNC 41 /S+ /D+ /n	16	13,5	0,455
ENCO	FNC 42 /S+ /D+ /n	16	18,0	0,455
SUM	FNC 43 /S+ /D+	16	12,7	0,325
		32	16,9	0,585
BON	FNC 44 /S+ /D+ /n	16	14,4	0,455
		32	15,1	0,845
MEAN	FNC 45 /S+ /D+ /n ⑨	16	11,8 + 0,41n	0,455
		32	17,8 + 2,13n	0,845
ANS	FNC 46 /S+ /m /D+	16	20,4	19,7
ANR	FNC 47 ③	16	7,0	0,065
SQR	FNC 48	16	9,7	0,325
		32	12,1	0,585
FLT	FNC 49	16	9,8	0,325
		32	9,5	0,585

Tab. B-58: Ausführungszeiten für Datenoperationen bei einer FX3U/FX3UC

B.5.7 High-Speed-Anweisungen

Anweisung		Ausführungszeit [μ s]		
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit
REF	FNC 50 /D /n ⑩	16	4,5 + 1,39n	0,325
REFF	FNC 51 /n ⑪	16	14,4 + 0,24n	0,195
MTR	FNC 52 /S /D1 /D2 /n	16	5,9	5,5
HSCS	FNC 53 /S1+ /S2+ /D+ ⑫	32	20,0	0,845
HSCR	FNC 54 /S1+ /S2+ /D+ ⑫	32	20,0	0,845
HSZ	FNC 55 /S1+ /S2+ /S+ /D+ ⑫	32	22,0	1,105
SPD	FNC 56 /S1+ /S2+ /D+	16	16	12,6
		32		
PLSY	FNC 57 /S1+ /S2+ /D+	16	20,0	6,9
		32	13,6	
PWM	FNC 58 /S1+ /S2+ /D+	16	10,6	6,2
PLSR	FNC 59 /S1+ /S2+ /S3+ /D+	16	11,2	7,0
		32		

Tab. B-61: Ausführungszeiten für High-Speed-Anweisungen bei einer FX3U oder FX3UC

B.5.8 Anwendungsbezogene Anweisungen

Anweisung		Ausführungszeit [μ s]		
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit
IST	FNC 60 /S /D1 /D2	16	28,5	0,455
SER	FNC 61 ⑬	16	16,4 + 1,4n	0,585
		32	18,5 + 2,13n	1,105
ABSD	FNC 62 /S1+ /S2+ /D+ /n ⑭	16	19 + 0,85n	0,585
		32	20,0 + 1,23n	1,105
INCD	FNC 63 /S1+ /S2+ /D+ /n	16	23,7	6,5
TTMR	FNC 64 /D+ /n	16	10,4	9,2
STMR	FNC 65 /S+ /m /D+	16	19,0	21,0
ALT	FNC 66 /D+	16	11,6	0,2
RAMP	FNC 67 /S1+ /S2+ /D+ /n	16	15,0	7,5
ROTC	FNC 68 /S+ /m1 /m2 /D+	16	25,8	24,8
SORT	FNC 69 ⑮	16	18,4	6,6

Tab. B-60: Ausführungszeiten für anwendungsbezogene Anweisungen bei einer FX3U oder FX3UC

B.5.9 Spezielle Anweisungen

Anweisung		Ausführungszeit [μ s]		
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit
TKY	FNC 70 /S+ /D1+ /D2+	16	21,5	5,2
		32	21,8	
HKY	FNC 71 /S+ /D1+ /D2+ /D3+	16	32,0	5,7
		32	32,3	
DSW	FNC 72 /S+ /D1+ /D2+ /n	16	26,8	22,1
SEGD	FNC 73 /S1+ /D+	16	10,8	0,325
SEGL	FNC 74 /S+ /D+ /n	16 1Set	22,3	7,5
ARWS	FNC 75 /S1+ /D+ /D2+ /n	16	28,8	5,2
ASC	FNC 76 /S+ /D+	16	19,8	0,715
PR	FNC 77 /S+ /D+	16	24,0	13,6
FROM ^①		16	141 + 419n	0,585
		32	119 + 841n	1,105
FROM ^②	FNC 78 /n1 /n2 /D+ /n3 ¹⁶	16	107 + 903n	0,585
		32	119 + 1791n	1,105
FROM ^③		16	27,9 + 108n	0,585
		32	17,6 + 187,4n	1,105
TO ^①		16	87 + 483n	0,585
		32	73 + 967n	1,105
TO ^②	FNC 79 /n1 /n2 /S+ /n3 ¹⁶	16	73 + 967n	0,585
		32	67 + 1923n	1,105
TO ^③		16	96,7 + 119,2n	0,585
		32	17,3 + 297,7n	1,105
RS	FNC 80	16	15,6	5,7
		32		
PRUN	FNC 81 /S+ /D+ ¹⁷	16	17,1 + 1,67n	0,325
		32	18,2 + 2,9n	0,585
ASCI	FNC 82	16	13,5 + 1,45n	0,455
HEX	FNC 83	16	13,6 + 1,89n	0,455
CCD	FNC 84	16	13,6 + 1,63n	0,455
VRRD	FNC 85 /S+ /D+	16	129,50	0,325
VRSC	FNC 86 /S+ /D+	16	129,94	0,325
RS2	FNC 87	16	18,1	5,3
PID	FNC 88	16	20,0	8,9

Tab. B-62: Ausführungszeiten für spezielle Anweisungen bei einer FX3U oder FX3UC (1)

- ① Diese Zeiten gelten, wenn mit dieser Anweisung auf die Pufferspeicheradressen 0 bis 31 eines Sondermoduls der FX0N-, FX2N- oder FX2NC-Serie zugegriffen wird.
- ② Diese Zeiten gelten, wenn mit dieser Anweisung auf den Pufferspeicher eines Sondermoduls der FX0N-, FX2N- oder FX2NC-Serie ab der Adresse 32 zugegriffen wird.
- ③ Diese Zeiten gelten, wenn mit dieser Anweisung auf den Pufferspeicher eines Sondermoduls der FX3U-Serie zugegriffen wird.

Anweisung		Ausführungszeit [μ s]		
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit
ZPUSH	FNC 102 /D+	16	16,0	0,195
ZPOP	FNC 103 /D+	16	16,0	0,195
ECMP	FNC 110 /S1+ /S2+ /D+	32	18,2	0,845
EZCP	FNC 111 /S1+ /S2+ /S3+ /D+	32	21,6	1,105
EMOV	FNC 112 /S+ /D+	32	10,0	0,585
ESTR	FNC 116 /S1+ /S2+ /D+	32	27 + 1,7n + 1,2m n: Anzahl der Zeichen m: Anzahl der Nachkommastellen	0,845
EVAL	FNC 117 /S+ /D+		26 + 3,8n	0,585
EBCD	FNC 118 /S+ /D+	32	10,0	0,585
EBIN	FNC 119 /S+ /D+	32	11,9	0,585
EADD	FNC 120 /S1+ /S2+ /D+	32	14,2	0,845
ESUB	FNC 121 /S1+ /S2+ /D+	32	14,2	0,845
EMUL	FNC 122 /S1+ /S2+ /D+	32	14,1	0,845
EDIV	FNC 123 /S1+ /S2+ /D+	32	17,7	0,845
EXP	FNC 124 /S+ /D+	32	11,9	0,585
LOGE	FNC 125 /S+ /D+	32	24,0	0,585
LOG10	FNC 126 /S+ /D+	32	24,3	0,585
ESQR	FNC 127 /S+ /D+	32	10,6	0,585
INT	FNC 129 /S+ /D+	16	13,2	0,325
		32	13,0	0,585
SIN	FNC 130 /S+ /D+	32	12,0	0,585
COS	FNC 131 /S+ /D+	32	23,2	0,585
TAN	FNC 132 /S+ /D+	32	12,0	0,585
ASIN	FNC 133 /S+ /D+	32	13,5	0,585
ACOS	FNC 134 /S+ /D+	32	13,5	0,585
ATAN	FNC 135 /S+ /D+	32	12,0	0,585
RAD	FNC 136 /S+ /D+	32	14,9	0,585
DEG	FNC 137 /S+ /D+	32	14,9	0,585
WSUM	FNC 140 /S+ /D+ /n	16	11,7 + 0,38n	0,455
		32	14,1 + 1,94n	0,845
WTOB	FNC 141 /S+ /D+ /n	16	12,6 + 1,43n	0,455
BTOW	FNC 142 /S+ /D+ /n	16	12,6 + 0,92n	0,455
UNI	FNC 142 /S+ /D+ /n	16	11,6 + 0,4n	0,455
DIS	FNC 143 /S+ /D+ /n	16	10,6 + 0,2n	0,455
SWAP	FNC 147 /S+	16	7,7	0,195
		32	8,0	0,325
SORT2	FNC 149 /S+ /n1 /n2 /D+ /m	16	13,2	6,5
		32	15,2	7,7

Tab. B-63: Ausführungszeiten für spezielle Anweisungen bei einer FX3U oder FX3UC (2)

Anweisung		Ausführungszeit [μ s]		
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit
DSZR	FNC 150 /S1+ /S2+ /D1+ /D2+	16	170,0	7,0
DVIT	FNC 151 /S1+ /S2+ /D1+ /D2+	16	178,0	7,1
		32		
TBL	FNC 152 /D+ /n	—	①	7,1
ABS	FNC 155 /S+ /D1+ /D2+	32	25,4	22,2
ZRN	FNC 156 /S1+ /S2+ /S3+ /D+	16	58,0	7,1
		32	62,0	
PLSV	FNC 157 /S+ /D1+ /D2+	16	144,0	7,1
		32		
DRVI	FNC 158 /S1+ /S2+ /D1+ /D2+	16	178,0	7,1
		32		
DRVA	FNC 159 /S1+ /S2+ /D1+ /D2+	16	178,0	7,1
		32		
TCMP	FNC 160 /S1+ /S2+ /S3+ /S+ /D+	16	21,3	0,715
TZCP	FNC 161 /S1+ /S2+ /S3+ /D+	16	22,6	0,585
TADD	FNC 162 /S1+ /S2+ /D+	16	13,4	0,455
TSUB	FNC 163 /S1+ /S2+ /D+	16	13,4	0,455
HTOS	FNC 164 /S+ /D+	16	10,8	0,325
		32	11,0	0,585
STOH	FNC 165 /S+ /D+	16	11,4	0,325
		32	11,6	0,585
TRD	FNC 166 /D+	16	10,0	0,195
TWR	FNC 167 /S+	16	344,4	0,195
HOOR	FNC 169 /S+ /D1+ /D2+	16	15,5	15,2
		32	16,1	15,9
GRY	FNC 170 /S+	16	10,2	0,325
		32	10,7	0,585
GBIN	FNC 171 /D+	16	15,4	0,325
		32	16,0	0,585
RD3A	FNC 176 /M1+ /M2+ /D+	16	1404 (FX0N-3A) 1828 (FX2N-2AD)	0,455
WR3A	FNC 177 /M1+ /M2+ /D+	16	1466 (FX0N-3A) 2919 (FX2N-2DA)	0,455
COMRD	FNC 182 /S+ /D+	16	33,7	0,325
RND	FNC 183 /D+	16	8,5	0,195
DUTY	FNC 186 /n1 /n2 /D+	16	6,0	6,0
CRC	FNC 188 /S+ /D+ /n	16	12,6 + 0,82n	0,455
DHCMOV	FNC 189 /S+ /D+ /n	32	14,8	0,845
BK+	FNC 192 /S1+ /S2+ /D+ /n	16	13,1 + 0,66n	0,585
		32	13,9 + 1,23n	1,105
BK-	FNC 193 /S1+ /S2+ /D+ /n	16	13,1 + 0,66n	0,585
		32	13,9 + 1,23n	1,105
BKCOMP □	FNC 194 – 199 /S1+ /S2+ /D+ /n	16	19,6 + 1,88n	0,585
		32	20,3 + 2,26n	1,105

Tab. B-64: Ausführungszeiten für spezielle Anweisungen bei einer FX3U oder FX3UC (3)

① Die Ausführungszeit hängt davon ab, mit welcher Positionieranweisung eine TBL-Anweisung kombiniert wird:
 DDVIT-Anweisung: 178,0 μ s, DPLSV-Anweisung: 144,0 μ s, DDRVI-Anweisung: 178,0 μ s, DRVA-Anweisung:
 178,0 μ s.

Anweisung		Ausführungszeit [μ s]		
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit
STR	FNC 200 /S1+ /S2+ /D+	16	34,6	0,455
		32	47,0	0,845
VAL	FNC 201 /S+ /D1+ /D2+	16	20,7	0,455
		32	29,2	0,845
\$+	FNC 202 /S1+ /S2+ /D+	16	24,8 + 1,5m m: Anzahl der Zeichen	0,455
LEN	FNC 203 /S+ /D+	16	12 + 0,44m m: Anzahl der Zeichen	0,325
RIGHT	FNC 204 /S+ /D+ /n	16	18,1 + 1,06n + 0,47m n: Anzahl der Zeichen m: Anzahl der extrahierten Zeichen	0,455
LEFT	FNC 205 /S+ /D+ /n	16	18,1 + 1,06n + 0,47m n: Anzahl der Zeichen m: Anzahl der extrahierten Zeichen	0,455
MIDR	FNC 206 /S1+ /D+ /S2+	16	25 + 0,59n + 0,68m n: Position des Zeichens m: Anzahl der Zeichen	0,455
MIDW	FNC 207 /S1+ /D+ /S2+	16	25,8 + 0,3m + 0,44n n: Position des Zeichens m: Anzahl der gespeicherten Zeichen	0,455
INSTR	FNC 208 /S1+ /S2+ /D+ /n	16	20,6 + 2,98n m: Anzahl der gesuchten Zeichen	0,585
\$MOV	FNC 209 /S+ /D+	16	16 + 1,52n	0,325
FDEL	FNC 210 /S+ /D+ /n	16	43 + 0,95m ¹⁸⁾	—
FINS	FNC 211 /S+ /D+ /n	16	63 + 0,98m ¹⁹⁾	—
POP	FNC 212 /S+ /D+ /n	16	7,8	0,455
SFR	FNC 213 /D+ /n	16	9,3	0,325
SFL	FNC 214 /D+ /n	16	9,3	0,325
LD □	FNC 224 – 230 /S1+ /S2+	16	1,22	
		32	1,48	
AND □	FNC 232 – 238 /S1+ /S2+	16	1,22	
		32	1,48	
OR □	FNC 240 – 246 /S1+ /S2+	16	1,22	
		32	1,48	
LIMIT	FNC 256 /S1+ /S2+ /S3+ /D+	16	8,1	0,585
		32	8,6	1,105
BAND	FNC 257 /S1+ /S2+ /S3+ /D+	16	8,1	0,585
		32	8,6	1,105
ZONE	FNC 258 /S1+ /S2+ /S3+ /D+	16	7,9	0,585
		32	8,5	1,105
SCL	FNC 259 /S1+ /S2+ /D+	16	15,9	0,455
		32	16,8	0,845
DABIN	FNC 260 /S+ /D+	16	13,7	0,325
		32	19,5	0,585
BINDA	FNC 261 /S+ /D+	16	16,7	0,325
		32	23,1	0,585

Tab. B-65: Ausführungszeiten für spezielle Anweisungen bei einer FX3U oder FX3UC (4)

Anweisung		Ausführungszeit [μ s]		
Symbol	FNC-Anweisung	Bit	EIN-Schaltzeit	AUS-Schaltzeit
SCL2	FNC 269 /S1+ /S2+ /D+	16	$2,79 + 5,21n$ n: Anzahl der Koordinatenpunkte	0,455
		32	$29,06 + 7,94n$ n: Anzahl der Koordinatenpunkte	0,845
IVCK	FNC 270 /S1+ /S2+ /D+ /n	16	14,1	6,5
IVDR	FNC 271 /S1+ /S2+ /S3+ /n	16	14,1	6,5
IVRD	FNC 272 /S1+ /S2+ /D+ /n	16	16,2	6,5
IVWR	FNC 273 /S1+ /S2+ /S3+ /n	16	16,2	6,5
IVBWR	FNC 274 /S1+ /S2+ /S3+ /n	16	20,0	6,5
RBFM ^①	FNC 278 /n1 /n2 /D3+ /n3 /n4	16	$50 + 900n$ n: Anzahl der in einem Programmzyklus übertragenen Daten	0,715
RBFM ^②		16	$244 + 103n$ n: Anzahl der in einem Programmzyklus übertragenen Daten	
WBFM ^①	FNC 279 /n1 /n2 /D3+ /n3 /n4	16	$24 + 966n$ n: Anzahl der in einem Programmzyklus übertragenen Daten	0,715
WBFM ^②		16	$292 + 116n$ n: Anzahl der in einem Programmzyklus übertragenen Daten	
DHSCT	FNC 280 /S1+ /n1 /S2+ /D+ /n2	32	30,0	1,365
LOADR	FNC 290 /S+ /n	16	$13,2 + 0,44n$	0,325
SAVER	FNC 291 /S+ /n /D+	16	$166n$	6,4
INITR	FNC 292 /S+ /n	16	$17600n$	0,325
LOGR	FNC 293 /S+ /n1 /D1+ /n2 /D2+	16	$244 + 17,9(n2)$	0,715
RWER	FNC 294 /S+ /n	16	$46700n$ n: Anzahl der Sektoren	0,325
INITER	FNC 295 /S+ /n	16	$17300n$	0,325

Tab. B-66: Ausführungszeiten für spezielle Anweisungen bei einer FX3U oder FX3UC (5)

- ① Diese Zeiten gelten, wenn mit dieser Anweisung auf den Pufferspeicher eines Sondermoduls der FX0N-, FX2N- oder FX2NC-Serie zugegriffen wird.
- ② Diese Zeiten gelten, wenn mit dieser Anweisung auf den Pufferspeicher eines Sondermoduls der FX3U-Serie zugegriffen wird.

- ① „n“ gibt die Anzahl der statischen STL-Anweisungen (Anzahl der parallelen/zusammenführenden Anweisungen) an.
- ② Die gekennzeichneten Anweisungen benötigen keine Kontakte.
- ③ Wird die statische Anweisung eingesetzt und nicht die Pulsanweisung, verändert sich der Wert der Zieladresse zyklisch.
- ④ „n“ gibt die Anzahl der zu verschiebenden Register an ($n \leq 512$).
- ⑤ „n“ gibt die Anzahl der zu verarbeitenden Bit-Operanden an ($n \leq 16$ im 16-Bit-Modus, $n \leq 32$ im 32-Bit-Modus).
- ⑥ „n“ gibt die Anzahl der zu verarbeitenden Bit-Operanden an.
- ⑦ „n“ gibt die Anzahl der zu verarbeitenden Operanden an ($2 \leq n \leq 512$).
- ⑧ „n“ gibt den zurückzusetzenden Operandenbereich an. Der Operandentyp ist in den Klammern angegeben.
- ⑨ „n“ gibt die Operandenadressen an, die mit der MEAN-Anweisung verarbeitet werden ($1 \leq n \leq 64$).
- ⑩ „n“ gibt den zu aktualisierenden Operandenbereich an ($8 \leq n \leq 128$ in 8-ter-Schritten).
- ⑪ „n“ gibt die Zeitkonstante des Eingangsfilters an ($0 \leq n \leq 60$ ms).
- ⑫ Diese Anweisung kann max. 6 mal zur gleichen Zeit aktiv sein.
- ⑬ „n“ gibt die Anzahl der Stapel-Elemente an ($n \leq 256$ für 16-Bit-Verarbeitung, $n \leq 128$ für 32-Bit-Verarbeitung).
- ⑭ „n“ gibt die Anzahl der Ausgangsadressen an ($n \leq 64$).
- ⑮ „n“ gibt die Anzahl der Elemente der Datentabelle an ($1 \leq m1 \leq 32$). Zur vollständigen SORT-Verarbeitung wird die SORT-Anweisung $m1$ mal ausgeführt.
- ⑯ „n“ gibt die Anzahl der in das/aus dem Sondermodul zu schreibenden oder lesenden Datenworte an.
- ⑰ „n“ gibt die Anzahl der zu schreibenden oder lesenden Single-Byte-Datenworte (8 Bit) bei Parallelbetrieb zweier FX-Steuerungen an.
- ⑱ $m = (\text{Anzahl der Tabellenzeilen}) - (\text{Position der gelöschten Daten})$
Die Anzahl der Tabellenzeilen entspricht dem Inhalt von (D+) und die Position der gelöschten Daten wird bei dieser Anweisung mit (n) angegeben.
- ⑲ $m = (\text{Anzahl der Tabellenzeilen}) - (\text{Position der eingefügten Daten})$
Die Anzahl der Tabellenzeilen entspricht dem Inhalt von (D+) und die Position der eingefügten Daten wird bei dieser Anweisung mit (n) angegeben.

B.6 Verarbeitungszeiten bei Puls-Ausführung

Werden Applikationsanweisungen als Puls-Anweisungen programmiert (Anhängen eines „P“, z. B. MOV_P), gelten andere Ausführungszeiten, als die, die in den vorhergehenden Abschnitten angegeben sind. Pulsanweisungen werden nur nach einer steigenden Flanke der Eingangsverknüpfung ausgeführt, benötigen aber auch Zeit, wenn sie nicht ausgeführt werden.

B.6.1 Verarbeitungszeiten für Grundgeräte der FX3G/FX3GC/FX3GE-Serie

Anweisung		Verarbeitungszeit [μ s]			
Symbol	FNC	Standard-Modus		Erweiterter Modus	
		Bei steigender Flanke der Eingangsverknüpfung	Wenn die Anweisung nicht ausgeführt wird	Bei steigender Flanke der Eingangsverknüpfung	Wenn die Anweisung nicht ausgeführt wird
MOV _P	12	4,98	1,44	5,43	1,81
BCDP	18	5,03		5,42	
BINP	19	5,03		5,21	
ADDP	20	6,37		6,68	
SUBP	21	6,47		6,78	
MULP	22	7,07		7,24	
DIVP	23	7,03		7,44	
INCP	24	3,48		3,49	
DECP	25	3,48		3,49	
WANDP	26	6,42		6,81	
WORP	27	6,42		6,81	
WXORP	28	6,42		6,81	
Alle anderen Applikationsanweisungen		Die in den Tabellen im Abschnitt B.3 in der Spalte „EIN-Schaltzeit“ angegebenen Zeiten verlängern sich bei gepulster Ausführung um jeweils 0,68 μ s.	Die in den Tabellen im Abschnitt B.3 in der Spalte „AUS-Schaltzeit“ angegebenen Zeiten verlängern sich bei gepulster Ausführung um jeweils 0,6 μ s.	Die in den Tabellen im Abschnitt B.3 in der Spalte „EIN-Schaltzeit“ angegebenen Zeiten verlängern sich bei gepulster Ausführung um jeweils 0,7 μ s.	Die in den Tabellen im Abschnitt B.3 in der Spalte „AUS-Schaltzeit“ angegebenen Zeiten verlängern sich bei gepulster Ausführung um jeweils 0,6 μ s.

Tab. B-67: Verarbeitungszeiten der Applikationsanweisungen bei Puls-Ausführung (FX3G/FX3GC/FX3GE)

B.6.2 Verarbeitungszeiten für Grundgeräte der FX3S-Serie

Anweisung		Verarbeitungszeit [μ s]	
Symbol	FNC	Bei steigender Flanke der Eingangsverknüpfung	Wenn die Anweisung nicht ausgeführt wird
MOVP	12	4,98	1,44
BCDP	18	7,18	
BINP	19	7,77	
ADDP	20	6,37	
SUBP	21	6,47	
MULP	22	7,07	
DIVP	23	7,03	
INCP	24	3,48	
DECP	25	3,48	
WANDP	26	6,42	
WORP	27	6,42	
WXORP	28	6,42	
Alle anderen Applikationsanweisungen		Die in den Tabellen im Abschnitt B.4 in der Spalte „EIN-Schaltzeit“ angegebenen Zeiten verlängern sich bei gepulster Ausführung um jeweils 0,68 μ s.	

Tab. B-68: Verarbeitungszeiten der Applikationsanweisungen bei Puls-Ausführung (FX3S)

B.6.3 Verarbeitungszeiten für Grundgeräte der FX3U/FX3UC-Serie

Falls Applikationsanweisungen als Puls-Anweisungen programmiert werden, verlängern sich die Ausführungszeiten:

- **MOVP-Anweisung**

Bei einer steigenden Flanke der Eingangsverknüpfung wird zur Ausführung die Zeit benötigt, die in der Tabelle im Abschnitt B.5 in der Spalte „EIN-Schaltzeit“ angegeben ist.

Wird die Anweisung nicht ausgeführt, ist die Zeit maßgebend, die in der Tabelle im Abschnitt B.5 in der Spalte „AUS-Schaltzeit“ aufgeführt ist.

- **Alle anderen Applikationsanweisungen**

Die in den Tabellen im Abschnitt B.5 in den Spalten „EIN-Schaltzeit“ und „AUS-Schaltzeit“ angegebenen Zeiten verlängern sich bei gepulster Ausführung (Anhängen eines „P“) um jeweils 0,45 μ s.

B.7 Abhängigkeit von Operanden und Index-Vergabe

In den folgenden Beispielen für Grundbefehle, der MOV-Anweisung und Vergleichsanweisungen hängt die Ausführungszeit von den in der Anweisung verwendeten Operanden ab. Die Ausführungszeit wird auch durch die Art der Zuweisung der Operanden beeinflusst (direkt oder durch Index-Vergabe).

B.7.1 Ausführungszeiten der Grundbefehle

- FX3G/FX3GC/FX3GE/FX3S

Bei einem FX3G-, FX3GC-, FX3GE- oder FX3S-Grundgerät können Bit-Operanden nicht durch Index-Vergabe zugewiesen werden. Die Angabe einzelner Bits von Wort-Operanden ist ebenfalls nicht möglich.

- FX3U/FX3UC

Angegebener Operand	Bedingung	Ausführungszeit der Anweisung
Bit-Operand	Ohne Index-Vergabe	0,065 µs
	Mit Index-Vergabe	11,9 µs
Wort-Operand	Angabe eines einzelnen Bits (D□.b)	8,8 µs

Tab. B-69: Ausführungszeiten der Grundbefehle (LD, LDI, AND, ANI, OR, ORI) bei einer FX3U/FX3UC in Abhängigkeit von Operanden und Index-Vergabe

B.7.2 Ausführungszeiten der Applikationsanweisungen

MOV-Anweisung

- FX3G/FX3GC/FX3GE

Eingangsbedingung der Anweisung	S (Datenquelle)	D (Ziel der Daten)							
		Standard-Modus				Erweiterter Modus			
		Ohne Index-Vergabe		Mit Index-Vergabe		Ohne Index-Vergabe		Mit Index-Vergabe	
		KnY, KnM, KnS*	T, C, D, R	KnY, KnM, KnS*	T, C, D, R	KnY, KnM, KnS*	T, C, D, R	KnY, KnM, KnS*	T, C, D, R
EIN	KnX, KnY, KnM, KnS	0,64 µs	0,72 µs	14,68 µs	8,28 µs	1,56 µs	1,8 µs	16,42 µs	9,92 µs
	T, C, D, R	0,88 µs	0,84 µs	13,08 µs	6,68 µs	1,84 µs	1,96 µs	14,82 µs	8,32 µs
	K, H	0,48 µs	0,52 µs	12,28 µs	5,78 µs	1,24 µs	1,4 µs	13,22 µs	7,22 µs
AUS	KnX, KnY, KnM, KnS	0,36 µs		0,58 µs		0,52 µs		0,82 µs	
	T, C, D, R								
	K, H								

Tab. B-70: Ausführungszeiten der MOV-Anweisung bei einer FX3G/FX3GC/FX3GE in Abhängigkeit von Operanden und Index-Vergabe

* Bei der Angabe der Blocklänge als „K4“ und einer Operandenadresse, die entweder „0“ oder ein Vielfaches von 8 ist (z. B. K4M0, K4M8 etc.).

● FX3S

Eingangsbedingung der Anweisung	S (Datenquelle)	D (Ziel der Daten)			
		Ohne Index-Vergabe		Mit Index-Vergabe	
		KnY, KnM, KnS*	T, C, D	KnY, KnM, KnS*	T, C, D
EIN	KnX, KnY, KnM, KnS*	0,64 µs	0,72 µs	14,68 µs	8,28 µs
	T, C, D	0,88 µs	0,64 µs	13,08 µs	6,68 µs
	K, H	0,48 µs	0,52 µs	12,28 µs	5,78 µs
AUS	KnX, KnY, KnM, KnS*	0,36 µs		0,58 µs	
	T, C, D				
	K, H				

Tab. B-71: Ausführungszeiten der MOV-Anweisung bei einer FX3S in Abhängigkeit von Operanden und Index-Vergabe

* Bei der Angabe der Blocklänge als „K4“ und einer Operandenadresse, die entweder „0“ oder ein Vielfaches von 8 ist (z. B. K4M0, K4M8 etc.).

● FX3U/FX3UC

Eingangsbedingung der Anweisung	S (Datenquelle)	D (Ziel der Daten)							
		Ohne Index-Vergabe				Mit Index-Vergabe			
		KnY, KnM, KnS*	T, C, D	R	U□\G□	KnY, KnM, KnS*	T, C, D	R	U□\G□
EIN	KnX, KnY, KnM, KnS	15,5 µs	12,1 µs	13,8 µs	212,9 µs	18,6 µs	16,6 µs	16,9 µs	214,7 µs
	T, C, D	12,1 µs	0,64 µs	10,4 µs	209,7 µs	16,6 µs	14,5 µs	14,8 µs	212,8 µs
	R	13,8 µs	10,4 µs	12,1 µs	211,3 µs	16,9 µs	14,8 µs	15,1 µs	213,1 µs
	U□\G□	131,8 µs	128,5 µs	130,2 µs	377,9 µs	133,1 µs	129,9 µs	131,5 µs	379,4 µs
	K, H	12,5 µs	0,64 µs	10,8 µs	210,1 µs	15,4 µs	13,3 µs	13,6 µs	211,6 µs
AUS	KnX, KnY, KnM, KnS		0,325 µs						
	T, C, D	0,325 µs	0,32 µs		0,325 µs				0,325 µs
	R								
	U□\G□		0,325 µs						
	K, H		0,32 µs						

Tab. B-72: Ausführungszeiten der MOV-Anweisung bei einer FX3U oder FX3UC in Abhängigkeit von Operanden und Index-Vergabe

DMOV-Anweisung

- FX3G/FX3GC/FX3GE

Eingangsbedingung der Anweisung	S (Datenquelle)	D (Ziel der Daten)							
		Standard-Modus				Erweiterter Modus			
		Ohne Index-Vergabe		Mit Index-Vergabe		Ohne Index-Vergabe		Mit Index-Vergabe	
		KnY, KnM, KnS*	T, C, D, R	KnY, KnM, KnS*	T, C, D, R	KnY, KnM, KnS*	T, C, D, R	KnY, KnM, KnS*	T, C, D, R
EIN	KnX, KnY, KnM, KnS	0,68 µs	0,88 µs	19,48 µs	8,68 µs	1,6 µs	2,12 µs	21,32 µs	10,32 µs
	T, C, D, R	1,24 µs	1,04 µs	18,18 µs	7,28 µs	2,2 µs	2,4 µs	19,92 µs	9,02 µs
	K, H	0,56 µs	0,56 µs	17,18 µs	6,28 µs	1,48 µs	1,88 µs	19,12 µs	8,22 µs
AUS	KnX, KnY, KnM, KnS	0,4 µs		0,58 µs		0,62 µs		0,82 µs	
	T, C, D, R								
	K, H								

Tab. B-73: Ausführungszeiten der DMOV-Anweisung bei einer FX3G/FX3GC/FX3GE in Abhängigkeit von Operanden und Index-Vergabe

* Bei der Angabe der Blocklänge als „K8“ und einer Operandenadresse, die entweder „0“ oder ein Vielfaches von 8 ist (z. B. K8M0, K8M8 etc.).

- FX3S

Eingangsbedingung der Anweisung	S (Datenquelle)	D (Ziel der Daten)			
		Ohne Index-Vergabe		Mit Index-Vergabe	
		KnY, KnM, KnS*	T, C, D	KnY, KnM, KnS*	T, C, D
EIN	KnX, KnY, KnM, KnS*	0,68 µs	0,88 µs	19,48 µs	8,68 µs
	T, C, D	1,24 µs	1,04 µs	18,18 µs	7,28 µs
	K, H	0,56 µs	0,56 µs	17,18 µs	6,28 µs
AUS	KnX, KnY, KnM, KnS*	0,4 µs		0,58 µs	
	T, C, D				
	K, H				

Tab. B-74: Ausführungszeiten der DMOV-Anweisung bei einer FX3S in Abhängigkeit von Operanden und Index-Vergabe

* Bei der Angabe der Blocklänge als „K8“ und einer Operandenadresse, die entweder „0“ oder ein Vielfaches von 8 ist (z. B. K8M0, K8M8 etc.).

● FX3U/FX3UC

Ein-gangs-bedin-gung der Anwei-sung	S (Daten-quelle)	D (Ziel der Daten)							
		Ohne Index-Vergabe				Mit Index-Vergabe			
		KnY, KnM, KnS*	T, C, D	R	U□\G□	KnY, KnM, KnS*	T, C, D	R	U□\G□
EIN	KnX, KnY, KnM, KnS	15,5 µs	12,4 µs	13,8 µs	311,1 µs	18,7 µs	17,0 µs	17,1 µs	313,0 µs
	T, C, D	12,4 µs	1,48 µs	10,7 µs	308,9 µs	17,1 µs	15,3 µs	15,4 µs	310,6 µs
	R	13,8 µs	10,7 µs	12,2 µs	211,3 µs	16,9 µs	14,8 µs	15,1 µs	213,1 µs
	U□\G□	199,8 µs	195,9 µs	197,4 µs	542,6 µs	201,2 µs	197,3 µs	198,8 µs	544,0 µs
	K, H	13,0 µs	1,48 µs	11,4 µs	308,0 µs	16,2 µs	14,4 µs	14,5 µs	309,7 µs
AUS	KnX, KnY, KnM, KnS	0,585 µs	0,585 µs	0,585 µs	0,585 µs	0,585 µs	0,585 µs	0,585 µs	0,585 µs
	T, C, D		1,48 µs						
	R		0,585 µs						
	U□\G□		0,585 µs						
	K, H		1,48 µs						

Tab. B-75: Ausführungszeiten der DMOV-Anweisung bei einer FX3U oder FX3UC in Abhängigkeit von Operanden und Index-Vergabe

16-Bit-Vergleichsanweisungen (LD□, AND□ und OR□)

- FX3G/FX3GC/FX3GE

S (Datenquelle)	D (Ziel der Daten)							
	Standard-Modus				Erweiterter Modus			
	Ohne Index-Vergabe		Mit Index-Vergabe		Ohne Index-Vergabe		Mit Index-Vergabe	
	KnY, KnM, KnS	T, C, D, R	KnY, KnM, KnS	T, C, D, R	KnY, KnM, KnS	T, C, D, R	KnY, KnM, KnS	T, C, D, R
KnX, KnY, KnM, KnS*	2,12 µs	2,2 µs	11,28 µs	9,18 µs	3,0 µs	3,12 µs	13,12 µs	10,7 µs
T, C, D, R	2,24 µs	2,36 µs	9,78 µs	7,58 µs	3,16 µs	3,24 µs	11,52 µs	9,32 µs
K, H	1,4 µs	1,52 µs	8,88 µs	6,78 µs	2,12 µs	2,24 µs	10,32 µs	8,12 µs

Tab. B-78: Ausführungszeiten der 16-Bit-Vergleichsanweisungen bei einer FX3G/FX3GC/FX3GE in Abhängigkeit von Operanden und Index-Vergabe

* Bei der Angabe der Blocklänge als „K4“ und einer Operandenadresse, die entweder „0“ oder ein Vielfaches von 8 ist (z. B. K4M0, K4M8 etc.).

- FX3S

S (Datenquelle)	D (Ziel der Daten)			
	Ohne Index-Vergabe		Mit Index-Vergabe	
	KnY, KnM, KnS*	T, C, D	KnY, KnM, KnS	T, C, D
KnX, KnY, KnM, KnS*	2,12 µs	2,2 µs	11,28 µs	9,18 µs
T, C, D	2,24 µs	2,36 µs	9,78 µs	7,58 µs
K, H	1,4 µs	1,52 µs	8,88 µs	6,78 µs

Tab. B-77: Ausführungszeiten der 16-Bit-Vergleichsanweisungen bei einer FX3S in Abhängigkeit von Operanden und Index-Vergabe

* Bei der Angabe der Blocklänge als „K4“ und einer Operandenadresse, die entweder „0“ oder ein Vielfaches von 8 ist (z. B. K4M0, K4M8 etc.).

- FX3U/FX3UC

S (Datenquelle)	D (Ziel der Daten)							
	Ohne Index-Vergabe				Mit Index-Vergabe			
	KnY, KnM, KnS	T, C, D	R	U□\G□	KnY, KnM, KnS	T, C, D	R	U□\G□
KnX, KnY, KnM, KnS	16,2 µs	13,0 µs	14,7 µs	133,1 µs	19,4 µs	17,4 µs	17,6 µs	134,7 µs
T, C, D	13,0 µs	1,22 µs	11,3 µs	129,9 µs	17,4 µs	15,4 µs	15,7 µs	132,8 µs
R	14,7 µs	11,3 µs	12,9 µs	131,4 µs	17,6 µs	15,7 µs	16,0 µs	133,1 µs
U□\G□	133,0 µs	129,3 µs	131,4 µs	298,2 µs	134,3 µs	131,0 µs	138,9 µs	299,4 µs
K, H	13,4 µs	1,22 µs	11,7 µs	130,2 µs	16,3 µs	14,2 µs	14,5 µs	131,6 µs

Tab. B-76: Ausführungszeiten der 16-Bit-Vergleichsanweisungen bei einer FX3U oder FX3UC in Abhängigkeit von Operanden und Index-Vergabe

32-Bit-Vergleichsanweisungen (DLD□, DAND□ und DOR□)

- FX3G/FX3GC/FX3GE

S (Datenquelle)	D (Ziel der Daten)							
	Standard-Modus				Erweiterter Modus			
	Ohne Index-Vergabe		Mit Index-Vergabe		Ohne Index-Vergabe		Mit Index-Vergabe	
	KnY, KnM, KnS	T, C, D, R	KnY, KnM, KnS	T, C, D, R	KnY, KnM, KnS	T, C, D, R	KnY, KnM, KnS	T, C, D, R
KnX, KnY, KnM, KnS*	2,52 µs	2,76 µs	11,68 µs	9,68 µs	3,48 µs	3,68 µs	13,32 µs	11,32 µs
T, C, D, R	2,8 µs	3,0 µs	10,72 µs	8,28 µs	3,72 µs	3,92 µs	11,92 µs	9,92 µs
K, H	1,4 µs	1,76 µs	9,28 µs	7,28 µs	2,36 µs	2,68 µs	11,12 µs	9,12 µs

Tab. B-79: Ausführungszeiten der 32-Bit-Vergleichsanweisungen bei einer FX3G/FX3GC/FX3GE in Abhängigkeit von Operanden und Index-Vergabe

* Bei der Angabe der Blocklänge als „K8“ und einer Operandenadresse, die entweder „0“ oder ein Vielfaches von 8 ist (z. B. K8M0, K8M8 etc.).

- FX3S

S (Datenquelle)	D (Ziel der Daten)			
	Ohne Index-Vergabe		Mit Index-Vergabe	
	KnY, KnM, KnS*	T, C, D	KnY, KnM, KnS	T, C, D
KnX, KnY, KnM, KnS*	2,52 µs	2,76 µs	11,68 µs	9,68 µs
T, C, D	2,8 µs	3 µs	10,72 µs	8,28 µs
K, H	1,4 µs	1,76 µs	9,28 µs	7,28 µs

Tab. B-80: Ausführungszeiten der 32-Bit-Vergleichsanweisungen bei einer FX3S in Abhängigkeit von Operanden und Index-Vergabe

* Bei der Angabe der Blocklänge als „K8“ und einer Operandenadresse, die entweder „0“ oder ein Vielfaches von 8 ist (z. B. K8M0, K8M8 etc.).

- FX3U/FX3UC

S (Datenquelle)	D (Ziel der Daten)							
	Ohne Index-Vergabe				Mit Index-Vergabe			
	KnY, KnM, KnS*	T, C, D	R	U□\G□	KnY, KnM, KnS*	T, C, D	R	U□\G□
KnX, KnY, KnM, KnS	16,4 µs	13,2 µs	14,7 µs	201,1 µs	19,6 µs	17,9 µs	18,0 µs	202,7 µs
T, C, D	13,2 µs	1,48 µs	11,6 µs	197,1 µs	17,9 µs	16,1 µs	16,2 µs	200,3 µs
R	14,7 µs	11,6 µs	13,0 µs	198,6 µs	18,0 µs	16,2 µs	16,3 µs	200,4 µs
U□\G□	201,0 µs	197,1 µs	198,6 µs	432,4 µs	202,4 µs	198,5 µs	200,0 µs	433,8 µs
K, H	13,9 µs	1,48 µs	12,3 µs	197,8 µs	17,1 µs	15,3 µs	15,4 µs	199,4 µs

Tab. B-81: Ausführungszeiten der 32-Bit-Vergleichsanweisungen bei einer FX3U oder FX3UC in Abhängigkeit von Operanden und Index-Vergabe

B.8 Ausführungszeiten für Pointer (P, I)

B.8.1 Grundgeräte der FX3G-, FX3GC- und FX3GE-Serie

Anweisung	Verarbeitungszeit [μ s]			
	Standard-Modus		Erweiterter Modus	
	EIN-Schaltzeit	AUS-Schaltzeit	EIN-Schaltzeit	AUS-Schaltzeit
P***	0,21		0,42	
I***	0,21		0,42	

Tab. B-82: Ausführungszeiten für Pointer bei einer FX3G/FX3GC/FX3GE

B.8.2 Grundgeräte der FX3S-Serie

Anweisung	Verarbeitungszeit [μ s]	
	EIN-Schaltzeit	AUS-Schaltzeit
P***	0,21	
I***	0,21	

Tab. B-83: Ausführungszeiten für Pointer bei einer FX3S

B.8.3 Grundgeräte der FX3U- und FX3UC-Serie

Anweisung	Verarbeitungszeit [μ s]	
	EIN-Schaltzeit	AUS-Schaltzeit
P0 bis P255	0,065	
P256 bis P4095	0,129	
I***	0,065	

Tab. B-84: Ausführungszeiten für Pointer bei einer FX3U oder FX3UC

C ASCII-Code

Bits 3 bis 0		Bits 6 bis 4							
		0	1	2	3	4	5	6	7
		000	001	010	011	100	101	110	111
0	0000	NUL	DLE	SP	0	@	P	'	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	!!	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	\$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(8	H	X	h	x
9	1001	HT	EM)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[k	{
C	1100	FF	FS	,	<	L	\	l	
D	1101	CR	GS	-	=	M]	m	}
E	1110	SO	RS	.	>	N	↑	n	~
F	1111	SI	VS	/	?	O	←	o	DEL

Tab. C-1:ASCII-Code

Beispiele für Verschlüsselungen im ASCII-Code:

00110100 = 34H: „4“

01000111 = 47H: „G“

00001101 = 0DH: CR (**C**arriage **R**eturn = Wagenrücklauf)

Index

Ziffern und Zeichen

\$+	7-222
\$MOV	7-240
1-Phasen-Counter	3-19
16-Bit-Counter	
adressieren	3-13
programmieren	3-13
rücksetzen	4-29
16-Bit-Register	3-26
2-Phasen-Counter	3-20
32-Bit-Anweisungen	6-8
32-Bit-Counter	3-17
32-Bit-Register	3-26
7-Segment-Anzeige	7-13

A

ABSD	6-120
ADD	6-45
ADPRW	7-295
ALT	6-127
ANB	4-20
AND	4-10
AND (UND-verknüpfter Vergleich)	7-255
ANDF	4-16
ANDP	4-16
ANI	4-10
ANR	6-81
ANS	6-80
ARWS	7-17
ASC	7-19
ASCI	7-37

ASCII-Code

Siehe auch Zeichenfolgen

aus Binärdaten erzeugen	7-276
aus hexadezimalen Wert erzeugen	7-37
in Binär-Daten wandeln	7-273
in hexadezimalen Wert wandeln	7-39
Übersicht	C-1

Addition	6-45
----------	------

Adressierung

16-Bit-Counter	3-13
32-Bit-Counter	3-15
Ein-/Ausgänge	3-2
Interrupt-Pointer	3-42
Merker	3-5
Nesting	3-46
Pointer	3-42
Register	3-28
Schrittstatus	3-23

Allgemeine Systemdaten

MELSEC FX1N	A-6
MELSEC FX1S	A-4
MELSEC FX2N	A-10
MELSEC FX2NC	A-10
MELSEC FX3G/FX3GC/FX3GE	A-14
MELSEC FX3S	A-18
MELSEC FX3U	A-22
MELSEC FX3UC	A-22

Anweisungen

32-Bit	6-8
Anwendungsbezogene	6-111

Anweisungsliste	2-6
-----------------	-----

Anwendungsbereich	1-1
-------------------	-----

Anzeige-Bits zurücksetzen	6-81
---------------------------	------

Applikationsanweisungen

Ausführung durch statisches Signal . . .	6-6
Übersicht	6-9
Übersicht FX1S/FX1N	A-8
Übersicht FX2N/FX2NC	A-12
Übersicht FX3G/FX3GC/FX3GE . . .	A-16
Übersicht FX3S	A-20
Übersicht FX3U/FX3UC	A-24
Arcuscosinusberechnung	7-96
Arcussinusberechnung	7-94
Arcustangensberechnung	7-98
Arithmetische Anweisungen	
mit Festkommazahlen	6-45
mit Gleitkommazahlen	7-58
Aufwärtszähler	4-46
Ausführungsregister für HSZ- und PLSY-Anweisungen	9-38
Ausführungszeiten	
Übersicht FX1S/FX1N	B-1
Übersicht FX2N/FX2NC	B-9
Übersicht FX3G/FX3GC/FX3GE . . .	B-19
Übersicht FX3S	B-31
Übersicht FX3U und FX3UC	B-40
Ausgangsoffset von 16- und 32-Bit-Binärdaten	7-266
Ausgänge	3-2
Adressierung	3-2
Doppelbelegung	4-9
auffrischen	6-85
programmieren	3-4
Ausschalten beliebiger Interrupts	3-46
Auto-Tuning-Funktion	7-49

B

BCD	6-41
BCD-Format	3-41
BIN	6-43
BINDA	7-276
BK+	7-200
BK-	7-203
BMOV	6-37
BON	6-78
BTOW	7-110
Befehl (Anweisung für SPS)	2-4
Begrenzung des Ausgangswertebereichs . .	7-260
Beschaltung der SPS	2-8
Bit	6-2
Bit-Operanden	
Definition	6-2
zusammenfassen	6-3
Bits überprüfen	6-78
Bitweise verschieben	6-64
Block-Transfer	6-37
Borrow Flag	6-47
Byte	6-2

C

CALL	6-16
CCD	7-41
CF-Speicherkartenadaptermodul	
Siehe FX3U-CF-ADP	
CJ	6-12
CML	6-36
CMP	6-28
COMRD	7-186
CRC	7-192
Carry Flag	6-47

Counter	
1-Phasen	3-19
16-Bit	3-13
High-Speed	3-17
absoluter Vergleich	6-120
rücksetzen	4-29
Counter (Sondermerker)	9-15

D

D8008 (Spannungsausfall)	8-17
D8020 (EingangsfILTER)	8-12
D8067 (PID-Anweisung)	7-51
D8120 (RS-Anweisung)	7-31
DABIN	7-273
DABS	7-134
DACOS	7-96
DADD	6-45
DAND	7-255
DASIN	7-94
DATAN	7-98
DBCD	6-41
DBIN	6-43
DCMP	6-28
DCOS	7-92
DDEC	6-54
DDEG	7-102
DDIV	6-51
DEADD	7-78
DEBCD	7-76
DEC	6-54
DECMP	7-59
DECO	6-73
DEDIV	7-81
DEMUL	7-80
DENEG	7-89
DESQR	7-88
DESTR	7-63
DESUB	7-79
DEVAL	7-71

DEXP	7-82
DEZCP	7-60
DGBIN	7-162
DGRY	7-161
DHCMOV	7-195
DHSCR	6-92
DHSCS	6-92
DHSCT	7-305
DHSZ	6-94
DI	6-18
DINC	6-53
DINT	7-90
DIS	7-114
DIV	6-51
DLOG10	7-86
DLOGE	7-84
DMOV	7-62
DMUL	6-49
DOR	6-56
DPLSR	6-108
DPLSY	6-104
DRAD	7-100
DRVA	7-141
DRVI	7-139
DSIN	7-91
DSPD	6-99
DSUB	6-47
DSW	7-11
DSWAP	7-116
DSZR	7-131
DTAN	7-93
DUTY	7-190
DWAND	6-55
DXOR	6-57
DZCP	6-30

Daten	
codieren	6-75
decodieren	6-73
nach links rotieren	6-61
nach rechts rotieren	6-60
verschieben	6-64
Daten (Wort) nach links verschieben	6-67
Daten (Wort) nach rechts verschieben	6-66
Datenausgabe	7-21
Datenblöcke	
Inhalte addieren	7-200
Inhalte subtrahieren	7-203
Inhalte vergleichen	7-206
Datenlisten	
Elemente einfügen	7-244
Elemente löschen	7-242
Datenoperationen	6-71
Datenregister	3-26
Datenrotation	6-63
Datenstruktur	6-5
Datentransfer	
gleicher Wert in mehrere Zieloperanden	6-38
von einer Quelle in ein Ziel	6-32
Datenverarbeitungsanweisungen II	7-105
Datenwort	6-2
Datenübertragung	
seriell (RS-Anweisung)	7-30
seriell (RS2-Anweisung)	7-45
Dekrementieren	6-54
Dezimalkonstante	3-25
Dezimalzahlensystem	3-36
Division	6-51
Doppelregister	3-26

E

EI-Anweisung	6-18
ENCO-Anweisung	6-75
EXTR	7-165
Echtzeituhr-Anweisungen	7-143
Eingang	
Abfragebeispiel	4-38
als Operand	3-2
Eingangsfiler	
einstellen	8-12
einstellen mit REFF-Anweisung	6-87
Eingangssignale	
Verarbeitung	3-3
kurzzeitige Impulse erfassen	8-4
Eingänge	3-2
Adressierung	3-2
auffrischen	6-85
programmieren	3-4
Erweiterte File-Register	
Inhalte in erweiterte Register transferieren	7-309
initialisieren (INITER)	7-326
initialisieren (INITR)	7-316
Erweiterte Register	
Inhalte in erweiterte File-Register transferieren	7-323
Inhalte sektionsweise in erweiterte File-Register transferieren	7-311
initialisieren	7-316
Exklusiv-ODER-Verknüpfung	6-57

F

FDEL	7-242
FEND-Anweisung	6-22
FIFO-Speicher	
Lesen aus Speicher	6-69
Schreiben in Speicher	6-68
Zuletzt eingetragene Daten lesen	7-246
FINS	7-244
FLCMD	7-339
FLCRT	7-330
FLDEL	7-333
FLRD	7-337
FLSTRD	7-340
FLT-Anweisung	6-83
FLWR	7-334
FMOV-Anweisung	6-38
FNC-Nummer	6-1
FOR-Anweisung	6-25
FROM	7-25
FX3U-CF-ADP	
Anweisungen	7-329
Sondermerker	9-27
Sonderregister	9-54
FX3U-ENET-ADP	
Sondermerker	9-28
Sonderregister	9-55
Fehlercodes	10-4
Fehlerdiagnose	
mit Sondermerker	10-1
mit Sonderregister	10-2
File-Register	
Anwendung	3-33
Flag	
Bedeutung	6-8
Sondermerker	9-4
Sonderregister	9-33
Flip-Flop-Funktion	6-127
Flussdiagramm (Ablaufsteuerung)	5-4

Frequenzumrichter

Mehrfachanweisung	7-293
Parameter blockweise schreiben	7-292
Parameter lesen	7-289
Parameter schreiben	7-290
Status prüfen	7-285
steuern	7-287

Funktionsplan	2-6
-------------------------	-----

G

GBIN	7-162
GRY	7-161
Geschwindigkeitserkennung	6-99
Gleitkommazahlen	
Darstellung	3-37
Konstante	3-25
Zahlensystem	3-38
als Exponent zur Basis e	7-82
Übersicht der Anweisungen	7-58
Gray-Code-Anweisungen	7-161
Grundbefehle	
Übersicht	A-1
Grundbefehlssatz	4-1
Ausführungszeiten	B-1
Übersicht	4-2

H

HEX	7-39
HKY	7-8
HOUR	7-159
HTOS	7-152
Hauptkontakt programmieren	4-25
Hexadezimalkonstante	3-25
Hexadezimalsystem	3-41
High-Speed-Anweisungen	6-84
High-Speed-Counter	3-17
Zähleingänge	3-17
setzen und rücksetzen	6-92

I

INC-Anweisung	6-53
INCD	6-122
INITER	7-326
INTR	7-316
INSTR	7-238
INT	7-90
INV	4-32
IRET	6-18
IST	6-112
IVBWR	7-292
IVCK	7-285
IVDR	7-287
IVMC	7-293
IVRD	7-289
IVWR	7-290
Impuls	
ausgeben	6-104
erzeugen (MEP, MEF)	4-117
erzeugen (PLS, PLF)	4-30
erzeugen (PWM)	6-106
Impulssignal	6-6
Impulsweitenmessung	8-7
Impulsweitenmodulation	6-106
Index-Register	
Einsatz	6-7
Inhalte sichern	7-54
Inhalte wiederherstellen	7-57
Übersicht	9-42
Inkrementaler Counter-Vergleich	6-122
Inkrementieren	6-53
Interrupt-Pointer	
Ausführungszeiten	B-61
Beschreibung	3-42
Interrupt-Programm	
Einsatz	6-18
High-Speed-Counter lesen	7-195
Invertierfunktion	6-36

J

Jahreszahl der internen Uhr	
auf vierstellige Darstellung	
umschalten	7-156

K

Kommunikation	
mit Frequenzumrichtern	7-283
mit Sondermodulen	7-299
Kommunikationsadapter (Sonderregister)	9-37
Kontaktplan	2-7
Kontrollbedingung setzen und rücksetzen	4-25
Konvertierung	
ASCII	7-19
BCD	6-41
Binär	6-43
Kopierfunktion	6-36

L

LD	4-6
LD (Lade Vergleich)	7-253
LD-/LDI-Anweisung	4-6
LDF	4-14
LDI	4-6
LDP	4-14
LEFT	7-229
LEN	7-224
LIMIT	7-260
LOADR	7-309
LOGR	7-319
Link- und Sonderfunktionen	9-35
Link-Funktionen	9-10
Logarithmus	
dekadischer	7-86
natürlicher	7-84
Logische Verknüpfungen	6-55

M

M8026 (RAMP-Anweisung)	6-129
M8075 (Impulsweiten-/Periodendauermessung)	8-7
M8122 (RS-Anweisung)	7-33
M8123 (RS-Anweisung)	7-33
M8130 (DHSZ-Anweisung)	6-95
M8132 (DHSZ-Anweisung)	6-97
M8160 (XCH-Anweisung)	6-40
M8161 (ASCI-Anweisung)	7-37
MC	4-25
MCR	4-25
MEAN	6-79
MEF	4-33
MEP	4-33
MIDR	7-232
MIDW	7-235
MODBUS-Kommunikation	
Sondermerker	9-26
MOV	6-32
MPP	4-22
MPS	4-22
MRD	4-22
MUL	6-49
Master Control	4-25
Master Control Reset	4-25
Merker	
adressieren	3-5
programmieren	3-6
Multiplikation	6-49

N

NEG	6-58
NEXT-Anweisung	6-25
NOP	4-35
Nesting	
Ebenen	3-42
programmieren	4-25

Numerische Daten

addieren	6-45
dividieren	6-51
multiplizieren	6-49
subtrahieren	6-47
vergleichen	6-28

Numerische Datenbereiche vergleichen 6-30

O

ODER-Verknüpfung	6-56
OR (ODER Vergleich)	7-257
ORB	4-21
ORF	4-18
ORP	4-18
OUT	4-8
Oktalzahlsystem	3-40

Operanden

Adressen	A-4
Beschreibung	6-2
Counter	3-12
Dezimalkonstante	3-25
Einführung	2-5
Erläuterung	2-5
Hexadezimalkonstante	3-25
Merker	3-5
Nesting	3-46
Pointer	3-42
Register	3-26
Schrittstatus	3-23, 3-24
Setzen und Zurücksetzen	4-28
Sonderregister	9-31
Übersicht	A-4
Übersicht FX1N	A-6
Übersicht FX1S	A-4
Übersicht FX2N und FX2NC	A-10
Übersicht FX3G/FX3GC/FX3GE	A-14
Übersicht FX3S	A-18
Übersicht FX3U und FX3UC	A-22
Übersicht aller Steuerungen	3-1

Operandenbereiche zurücksetzen 6-72
 Operandenkommentar lesen 7-186

P

PID 7-47
 PLF 4-30
 PLS 4-30
 PLSR 6-108
 PLSV 7-137
 PLSY 6-104
 POP 7-246
 PR 7-21
 PRUN 7-36
 PWM 6-106
 Paritätsprüfung 7-41
 Passwortfunktion
 bei Frequenzumrichtern 7-291
 Periodendauermessung 8-7
 Pointer
 Ausführungszeiten B-61
 adressieren 3-42
 als Sprungzieladresse 6-13
 im Programm festlegen 6-12
 programmieren 6-12
 Positionieranweisungen 7-120
 Programm
 -schritte 6-2
 Ende eines Interrupt-Programms 6-17
 Interrupt 9-7
 Leerzeile im 4-35
 Sprung in einem 6-12
 Unterprogrammaufruf 6-16
 verarbeitung 2-1
 wiederholen (FOR, NEXT) 6-25
 Programmablaufanweisungen 6-11
 Programmbereichsende 6-22
 Programmende 4-36
 Programmierung (Grundlagen) 2-1
 Programmschleife 6-25

Programmzykluszeit
 konstante 8-3
 maximale 6-23
 Prozessabbildverfahren 2-2
 Prüfsumme berechnen 7-41
 Pufferspeicher 7-23
 Pulse-Catch-Funktion
 Beschreibung 8-4
 Sondermerker 9-9

Q

Quadratwurzel berechnen 6-82
 Quelldaten 6-5

R

RAMP 6-129
 RBFM 7-299
 RCL 6-63
 RCR 6-62
 RD3A 7-163
 REF 6-85
 REFF 6-87
 RET 5-5
 RIGHT 7-226
 RND 7-189
 ROL 6-61
 ROR 6-60
 ROTC 6-131
 RS 7-30
 RS2 7-45
 RST 4-28
 RUN-/STOP-Umschaltung 8-16
 RWER 7-323
 Rampenfunktion 6-129
 Regelungsanweisung 7-47
 Register
 Aufbau 3-27
 adressieren 3-28
 Registertypen 3-26

Remanente Timer 3-11
 Rücksetzen von Operanden 4-28

S

SAVER 7-311
 SCL 7-269
 SCL2 7-279
 SEGD 7-13
 SEGL 7-14
 SER 6-118
 SET 4-28
 SFL 7-251
 SFR 7-249
 SFRD 6-69
 SFTL 6-64
 SFTR 6-64
 SFWR 6-68
 SMOV 6-33
 SORT 6-134
 SORT2 7-117
 SPD 6-99
 SPS-Modus
 Sondermerker 9-5
 Sonderregister 9-33
 SPS-Status
 Sondermerker 9-2
 Sonderregister 9-31
 SQR 6-82
 SRET 6-17
 STL 5-5
 STL-Anweisungen 5-1
 STL-Status
 Sondermerker 9-6
 Sonderregister 9-34

STL-Verzweigungen
 Einfachverlauf 5-11
 Leerstatus 5-17
 Sprung 5-18
 parallele 5-14
 selektive 5-12
 STMR 6-125
 STO 7-154
 STOP-Modus
 Datenerhalt im 8-2
 STR 7-212
 SUB 6-47
 SUM 6-77
 SWAP 7-116
 Schrittstatus 3-23, 3-24
 adressieren 3-23
 initialisieren 5-10
 initialisieren mit IST-Anweisung . . . 6-112
 Schrittstatusoperanden
 Adressierung 3-23, 3-24
 S0 bis S9 5-1
 Schrittsteuerung
 Ausgänge mehrfach belegen 5-7
 Flussdiagramm 5-4
 Schematischer Ablauf 5-3
 Timer mehrfach belegen 5-8
 Serielle Kommunikation (Anweisungen) . . . 7-29
 Setzen von Operanden 4-28
 Shift-Transfer 6-33
 Skalieren von Werten
 mit Tabelle (SCL-Anweisung) 7-269
 mit Tabelle (SCL2-Anweisung) 7-279
 Sonderfunktionen 8-1

Sondermerker

Fehlererkennung	10-1
Flag	9-4
Interrupt-Programm	9-7
M8026 (RAMP-Anweisung)	6-129
M8027 (PR-Anweisung)	7-21
M8075 (Impulsweiten-/Perioden- dauermessung)	8-7
M8090 (Datenblockvergleich)	7-207
M8122 (RS-Anweisung)	7-33
M8123 (RS-Anweisung)	7-33
M8130 (DHSZ-Anweisung)	6-95
M8132 (DHSZ-Anweisung)	6-97
M8160 (XCH-Anweisung)	6-40
M8161 (8-Bit-Modus)	7-20
M8161 (ASCII-Anweisung)	7-37
SPS-Modus	9-5
SPS-Status	9-2
STL-Status	9-6

Sondermodul

Daten aus Pufferspeicher lesen	7-25
Daten aus Pufferspeicher lesen (RBFM)	7-299
Daten in Pufferspeicher schreiben	7-27
Daten in Pufferspeicher schreiben (WBFM)	7-301
Datenaustausch mit Grundgerät	7-23
direkte Adressierung	3-47

Sonderregister

D8008 (Spannungsausfall)	8-17
D8067 (PID-Anweisung)	7-51
D8120 (RS-Anweisung)	7-31
EingangsfILTER	9-33
Fehlererkennung	10-2
Impulsweiten-/Periodendauer- messung	8-7
Kommunikation mit Frequenzumrichtern	9-39
Kommunikationsadapter	9-37
Ringzähler	9-48
SPS-Status	9-31

STL-Status	9-34
Zeittakt	9-32
für Analogmodule	9-43
für Anzeigemodul FX3G-5DM	9-44
für Anzeigemodul FX3U-7DM	9-44
für Indexregister	9-33
für Positionierung	9-45
für Zufallszahl	9-44
integrierte Potentiometer	8-13
n:n-Netzwerk	9-41
Sonstige Sonderregister	9-36
Sortieranweisung	6-134
Source	6-5
Sprung innerhalb eines Programms	6-12
Statisches Signal	6-6
Steuerungsanweisungen	
Aufbau	2-4
Darstellungsarten	2-6
Subtraktion	6-47
Suchanweisung	6-118

T

TADD	7-148
TCMP	7-144
TKY	7-6
TO	7-27
TRD	7-156
TSUB	7-150
TTMR	6-124
TWR	7-158
TZCP	7-146
Tastatur	
Zehner	7-6
hexadezimal	7-8
Technische Daten	
Operanden	A-4

Timer	
Genauigkeit	3-11
Programmierbeispiel	4-44
Spezial	6-125
Teaching	6-124
Watch Dog	6-23
Transferanweisungen	6-27

U

UNI	7-112
Uhrzeit	
Einheit wandeln	7-152
addieren	7-148
einstellen	7-158
subtrahieren	7-150
vergleichen	7-144
Umwandlung	
ASCII in hexadezimal	7-39
Gleitkommazahl in Zeichenfolge	7-63
Grad in Radiant	7-100
Radiant in Grad	7-102
Zeichenfolge in Binärwert	7-217
Zeichenfolge in Gleitkommazahl	7-71
hexadezimal in ASCII	7-37
Umwandlung des Zahlenformats	6-83
Unterprogramm	
Aufruf	6-16
Ende	6-17

V

VAL	7-217
VRRD	7-43
VRSC	7-44
Vergleich	
numerische Daten	6-28
von Datenblöcken	7-206
Vergleichsanweisungen II	7-253

Verknüpfungen	
ODER	4-12
ODER-Block	4-21
OR	4-12
ORI	4-12
UND	6-55
UND-Block	4-20
Verknüpfungsbeginn	4-6
Verknüpfungsebene	4-22
Verknüpfungsergebnis	
ausgeben	4-8
in Impuls wandeln	4-33
speichern	4-22
Verschiebeanweisungen	6-59

W

WAND	6-55
WBFM	7-301
WDT	6-23
WOR	6-56
WSFL	6-67
WSFR	6-66
WSUM	7-106
WTOB	7-108
WXOR	6-57
Weiterschaltbedingung	5-7
Wiederholung	
Programmteile	6-25
Wort	6-2
Wortoperanden	
Summenbildung	7-106
aus 4-Bit-Gruppen bilden	7-112
aus Bytes bilden	7-110
in Bytes aufteilen	7-108
Übersicht	6-2

X

XCH	6-39
-----	------

Z

- ZCP-Anweisung 6-30
- ZPUSH 7-54
- ZRN 7-135
- ZRST 6-72
- Zahlen im wissenschaftlichen Format 3-37
- Zahlenformate 3-36
- Zehnergertastatur 7-6
- Zeichenfolgen
- Länge ermitteln 7-224
 - Zeichen ab dem Anfang
herausziehen 7-229
 - Zeichen ab dem Ende
herausziehen 7-226
 - Zeichen kopieren 7-232
 - aus Binärdaten erzeugen 7-212
 - ersetzen 7-235
 - in Binärdaten wandeln 7-217
 - suchen 7-238
 - transferieren 7-240
 - zusammenfügen 7-222
 - Zeitintervall starten 6-80
- Zeitsollwert 3-9
- Zeitsollwertangabe
- direkt 3-9
 - indirekt 3-9
- Zeittakte
- Sondermerker 9-3
 - Sonderregister 9-32
- Zero Flag 6-47
- Zieldaten 6-5
- Zufallszahl erzeugen 7-189
- Zuordnungsliste 2-8
- Zurücksetzen von Operanden 4-28
- Zykluszeit 2-1
- Zähler
- Siehe Counter
- Zählersollwert 3-13
- Zählersollwertangabe
- direkt 3-14
 - indirekt 3-14

Deutschland

Mitsubishi Electric Europe B.V.
Gothaer Straße 8
D-40880 Ratingen
Telefon: (0 21 02) 4 86-0
Telefax: (0 21 02) 4 86-11 20
<https://de3a.MitsubishiElectric.com>

Kunden-Technologie-Center

Mitsubishi Electric Europe B.V.
Revierstraße 21
D-44379 Dortmund
Telefon: (02 31) 96 70 41-0
Telefax: (02 31) 96 70 41-41

Mitsubishi Electric Europe B.V.
Kurze Straße 40
D-70794 Filderstadt
Telefon: (07 11) 77 05 98-0
Telefax: (07 11) 77 05 98-79

Mitsubishi Electric Europe B.V.
Lilienthalstraße 2 a
D-85399 Hallbergmoos
Telefon: (08 11) 9 98 74-0
Telefax: (08 11) 9 98 74-10

Österreich

GEVA
Wiener Straße 89
A-2500 Baden
Telefon: +43 (0) 22 52 / 85 55 20
Telefax: +43 (0) 22 52 / 4 88 60

Schweiz

OMNI RAY AG
Im Schörlü 5
CH-8600 Dübendorf
Telefon: +41 (0)44 / 802 28 80
Telefax: +41 (0)44 / 802 28 28