MITSUBISHI

PROGRAMMABLE CONTROLLER

MELSEC-A

User's Manual

Intelligent communication module type AD51E

MITSUBISHI ELECTRIC

REVISIONS

※The manual number is given on the bottom left of the back cover.

Print Date	*Manual number	Revision
Nov., 1986	IB (NA) 66059-A	First edition
May., 1987	IB (NA) 66059-B	Correction CONTENTS, APP-7 Addition APP-6

INTRODUCTION

Thank you for choosing the Mitsubishi MELSEC-A Series of General Purpose Programmable Controllers. Please read this manual carefully so that the equipment is used to its optimum.
A copy of this manual should be forwarded to the end User.

CONTENTS

1. INTRODUCTION

1.1 General

The AD51E Intelligent Communication Unit has two RS-232C and two RS-422 interfaces and allows multitask processing of BASIC programs.
User application programs running in the AD51E allow the following functions:

(1) Supervision of PC operating conditions

Allows graphical monitoring of the PC operating conditions on a CRT.

(2) Data transfer

Allows data to be transferred between the PC and a computer for collection and analyses.

(3) Data logging

Allows process data and fault information to be down loaded to a printer from the PC. The AD51E also includes an on-board real time clock for time, event data recording.

(4) PC CPU remote run/stop

Allows the PC CPU to be remote-controlled by the AD51E user program.

(5) UP/DOWN load of PC sequence programs

Allows PC data (including sequence programs and parameters) to be transferred between the AD51E and the PC CPU in machine code. PC sequence programs can therefore be changed according to the control operation required.

(6) Computer communication

Allows data transfer between a computer and the PC CPU. Either RS-422 or RS-232C may be used and the communications system can be configuired as a multidrop link for up to 32 stations.

1.2 Notes on Character Sets

In this manual some of the characters used may differ from those which appear on the screen, depending on the character set chosen (i.e. Japanese, English, German, Swedish).
The keyboard operations follow the standard for the character set chosen, so, for example to input @ with English characters press [SFT], [@].
Key codes are given in Appendix 4, paragraph 3.
Key operations for the different character sets are shown below.

Key	Character Set			
	English	German	Swedish	Japanese
SFT	^	^	Ü	~
	~	β	ü	^
SFT	\	Ö	Ö	¦
	¦	ö	ö	\
SFT	@	§	É	`
	`	`	é	@
SFT	[	Ä	Ä	{
	{	ä	ä	[
SFT	]	Ü	Å	}
	}	ü	à	]
SFT	£	#	#	#
SFT	$	$	ö	$

In this manual, all examples use the Japanese character set.

2. SYSTEM CONFIGURATION

2.1 Overall Configuration

(1) A1, A2, A3 type PC

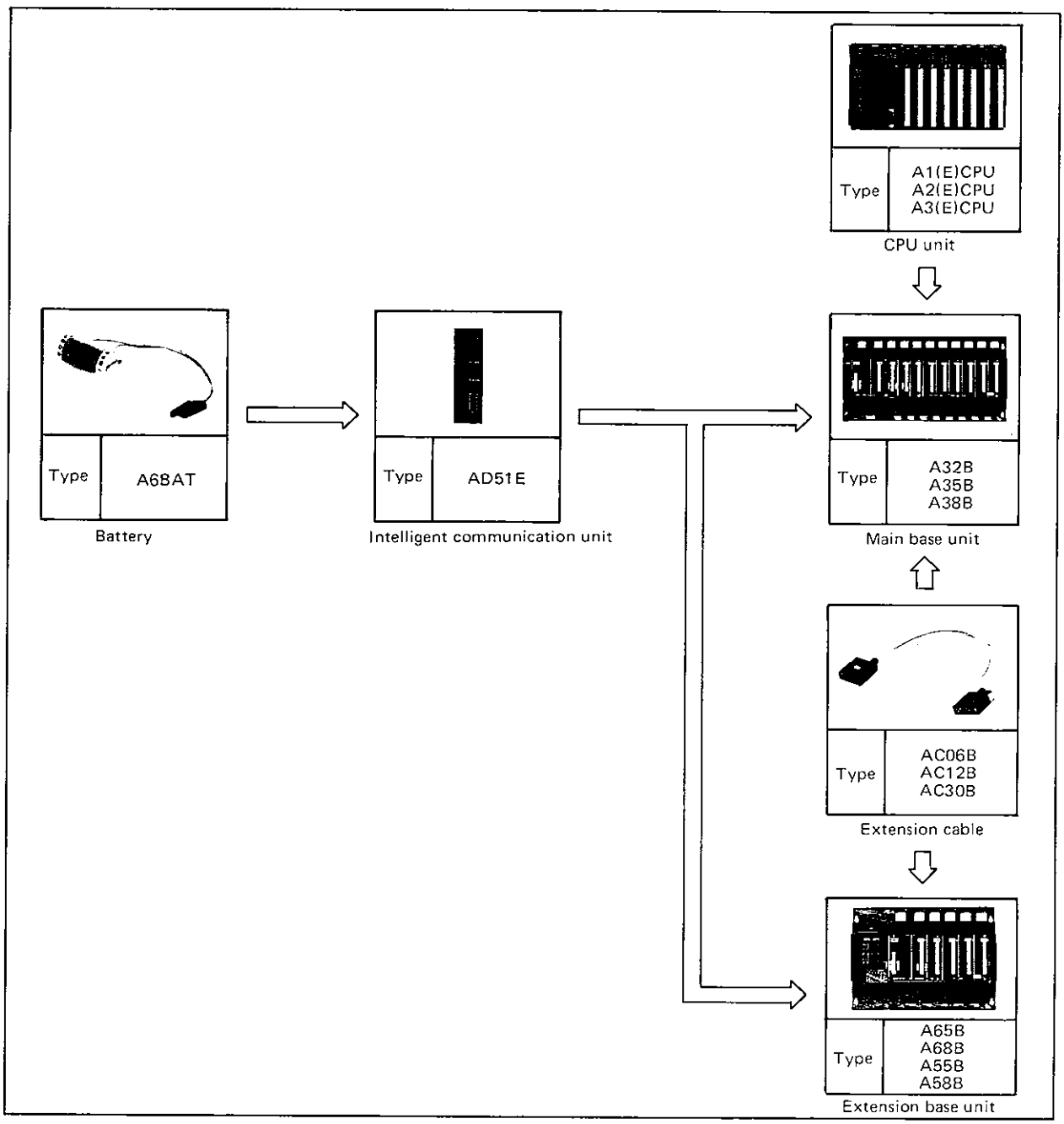


Fig. 2.1 Building Block CPU Configuration

IB (NA) 66059-A

(2) A0J2 type PC

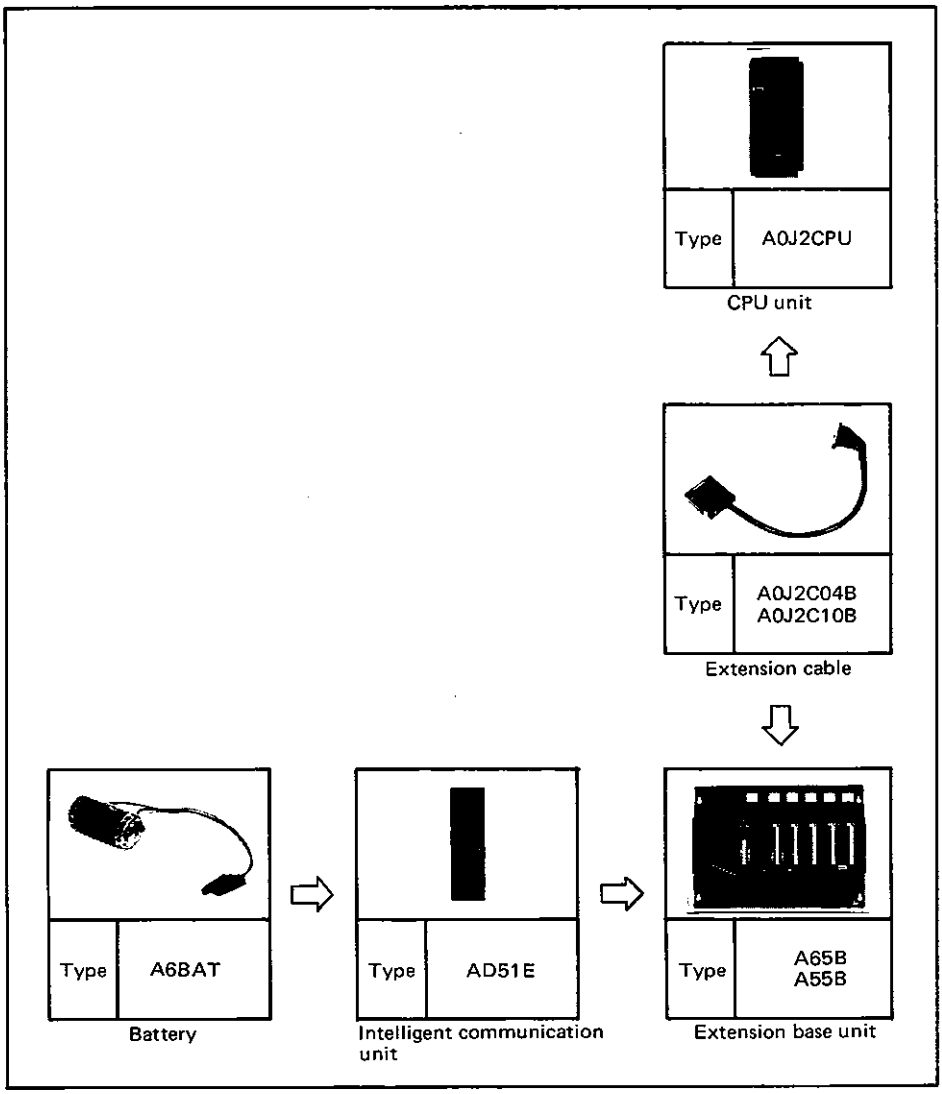


Fig. 2.2 Compact CPU Configuration

(3) Peripherals

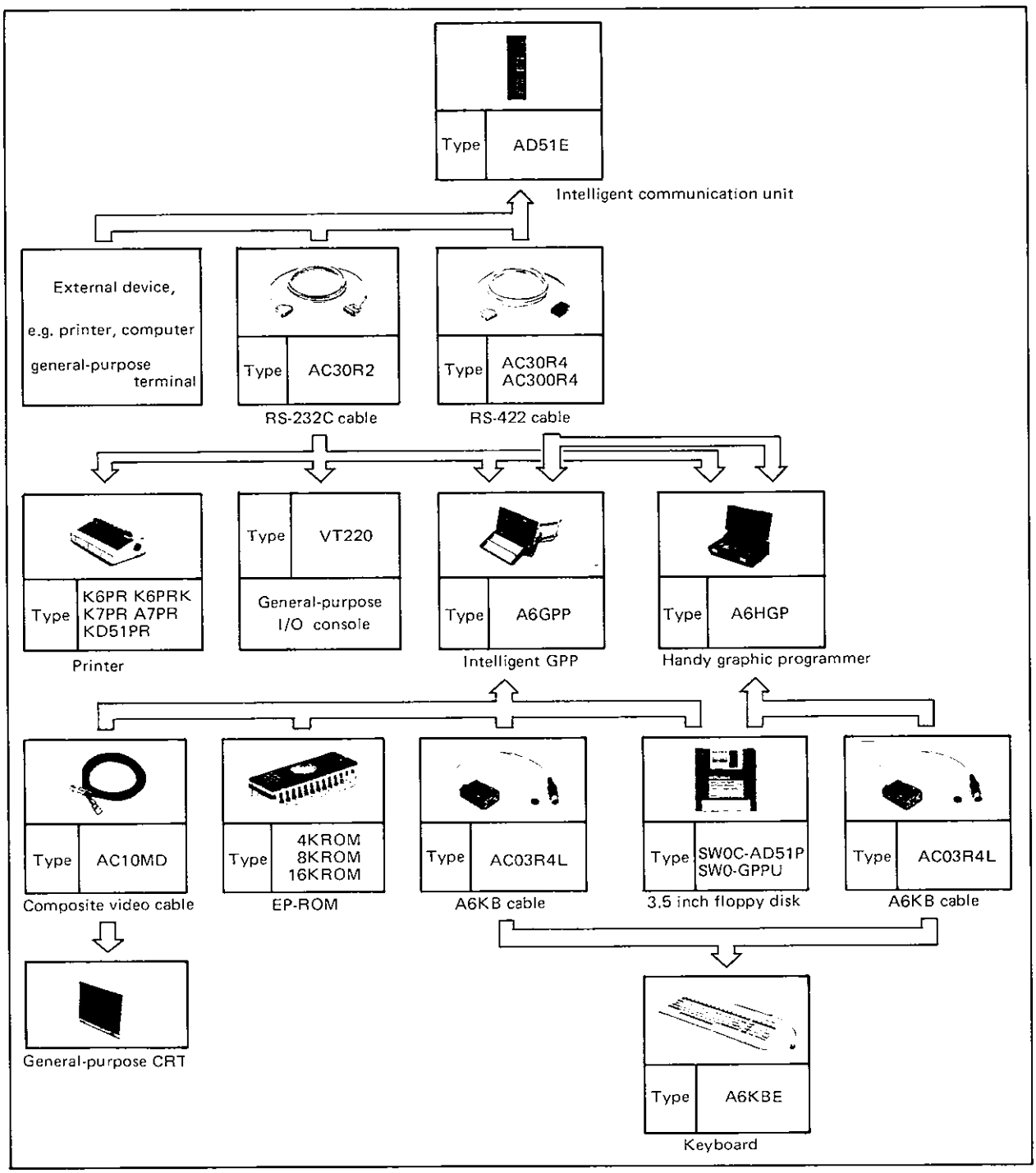


Fig. 2.3 Peripheral Device Configuration

POINT

The consul select switch must be set to determine which programming terminal is to be used.

2.2 Applicable A Series Systems

The AD51 can be used with the following CPU units.

Applicable models	A1(E)CPU A2(E)CPU A3(E)CPU

The AD51 may be loaded into any slot on the base unit with the following precautions:

(1) When using the AD51E with the A55B or A58B extension bases (i.e. those without power supplies) select the power supply for the main base unit in accordance with the A1, A2 A3CPU User's manual.

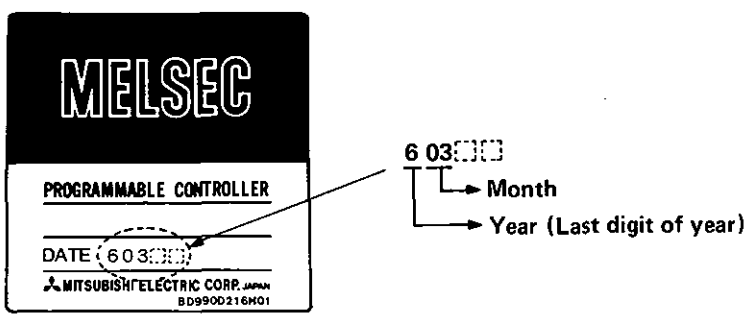
(2) The AD51 may be loaded into the master station or a local station but not into a remote I/O station in a MELSEC-NET system. The following CPU units are required for the MELSEC-NET data link system.

Applicable models (master or local stations)	A1(E)CPU P21/R21 A2(E)CPU P21/R21 A3(E)CPU P21/R21 A0J2CPU P23/R23 (local only)

(3) One AD51E or A0J2C214 (computer/multidrop link unit) may be used with the A0J2 PC. The A1, A2, A3 CPUs can accomodate two AD51Es or two AJ71C24s or one of each.

POINT

The AD51 may only be used with PC CPUs manufactured in March, 1986 and thereafter. The name plate shown below indicates the date of manufacture.



(4) The AD51E cannot be used in the last slot of the seventh extension stage in an A3CPU system.

2.3 Peripheral Equipment

The following table lists the peripheral equipment suitable for use with the AD51.

Description	Type	Remarks
Intelligent communication unit	AD51	Main unit RAM support battery supplied
EP-ROM	8KROM	16K bytes, for channels 1 and 2
	16KROM	32K bytes, for channels 1 and 2 24K bytes available to AD51 For details, refer to Section 3.5.2 (page 3-14)
Battery	A6BAT	For ICRAM support and real time clock
Intelligent GPP	A6GPP-SET	○ Consists of the following models: Type / Remarks A6GPP: • Programming unit with CRT. • Equipped with ROM writer, FDD, and printer interface functions. SW□-GPPA: A series system disk SW□-GPPK: K series system disk SW□-GPPU: User disk (3.5 inch, formatted) AC30R4: Cable for connection of AD51 and A6GPP, 3m length
Handy graphic programmer	A6HGP-SET	○ Consists of the following: Type / Remarks A6HGP: • Programming unit with LCD • Equipped with FDD and printer interface functions. SW□-HGPA: A series system disk SW□-HGPK: K series system disk SW□-GPPU: User disk (3.5 inch, formatted) AC30R4: Cable for connection of AD51E and A6HGP, ACPU, AJ71C24
System disk	SW0C-AD51PE	System software package for the A6GPP or A6HGP (back-up copy provided)
User disk	SW0-GPPU	User disk (already formatted) for storing programs.
Composite video cable	AC10MD	Optional cable for GPP external monitor 1m length
General-purpose I/O console	VT-220	Display control codes equivalent to VT-220.
Printer	K6PR K6PR-K K7PR A7PR	For program hard copy and data print out
	KD51PR	For printing data
RS-422 cable	AC300R4	Cable between AD51 and A6GPP 3m length
RS-232C cable	AC30R2	Connection cable between AD51 and printer and for VT-220. 3m length

Table 2.1 System Equipment List (Continue)

IB-NA-66059-A

Description	Type	Remarks
Ink ribbon	K6PR-R	Ink ribbon for K6PR and K6PR-K
	K7PR-R	Ink ribbon for K7PR
	A7PR-R	Ink ribbon for A7PR
	KD51PR-R	Ink ribbon for KD51PR
Printer paper	K6PR-Y	Printer paper for K6PR and K6PR-K
	KD51PR-Y	Printer paper for KD51PR
Interface connector	232-CON	Connector for RS-422 and RS-232C interfaces

Table 2.1 System Equipment List

3. SPECIFICATIONS

3.1 Performance Specifications

Item		Specification
Processor		HD64180
Program language		GPC-BASIC
Number of tasks		Maximum 8
Task start conditions		Power on
		Interrupt from PC CPU
		Real time interrupt (Set in the range 0.01 to 9.99 seconds in units of 0.01 second.)
Internal memory		Maximum: 114K bytes = 64K bytes + 2K bytes + 48K bytes Common work area Two 16KROMs loaded *For details, refer to Section 3.5 (page 3-13)
General-purpose I/O		General-purpose input : 13 points General-purpose output : 10 points *For details, refer to Section 3.6 (page 3-16).
Buffer memory		3K words (6K bytes) *For details, refer to Section 3.7 (page 3-18).
Memory protect address range		4F00 to 4FFF (system data area) 8000 to DFFF (channel 1 to 4)
Interfaces	RS-422	Conforms to EIA. RS-422. Channel 1: D shell connector. Channel 2: Terminal block Transmission distance: ⩽ 500m
	RS-232C	Conforms to EIA. RS-232C. Channel 3, 4: D shell connector. Transmission distance: ⩽ 15m
Arithmetic and logic unit (ALU)		Performs high-speed processing of BASIC's intrinsic functions (trigonometric, inverse trigonometric, logarithm, exponential, $\sqrt{\ }$, absolute value).
Clock element		Year, month, day, hour, minute, second Read/write 24 hour mode, automatic leap year compensation
Power failure compensation		Internal memory, lithium battery for back-up of real time clock Total back-up time : 130 days Battery life : 5 years
Console		A6GPP, A6HGP VT-220
Number of I/O points occupied		48
Internal current consumption (5V)		1.3A
Size		250(H) x 76(W) x 120(D)mm 9.8 x 3 x 4.7 (inch)
Weight		1.1kg

Table 3.1 AD51 Performance Specifications

IB (NA) 66059-A

3.2 Instruction Set

The AD51 is programmed in GPC-BASIC.
In addition to the BASIC commands a series of subroutines are available which can be called from the BASIC program.

3.2.1 GPC-BASIC commands

The following table lists the GPC-BASIC commands. For full information refer to the "GPC-BASIC Handbooks". The graphics commands described in the "GPC-BASIC" Handbooks are not available on the AD51.

Command			Function
Key command [KEY]	AUTO		Automatic generation of line number
	BYE		Returns to BASIC program address data screen.
	CONT		Resumes program run after BREAK
	COMPILE		Compiles multitask executable program
	DELETE		Deletes program from specified line number to specified line number
	EDIT		Corrects statement in one line
	EXECUTE		Run of program after "RUN" or "COMPILE"
	LIST	A, B, C	Displays program on screen
	LLIST		Prints out program
	NEW		Erases program
	RENUM	A, B	Renumbers line numbers
	RUN		Executes program
	ZDV		Displays I/O console
	–		Deletes line
Program command [PRG]	BREAK		Resumes program run after stop and "CONT"
	CALL		Calls machine language program
	CLS	A, B	Clears CRT screen
	CLOSE	A, B, C, D	Closes specified RS-232C/RS-422 channel.
	END		Declares end of program run
	FOR···NEXT		Repeats program run from "FOR" to "NEXT"
	GOTO		Moves to specified line number

Table 3.2 BASIC Command List (Continue)

Command			Function
Program command PRG	GOSUB····· ·····RETURN		Moves to specified subroutine Returns from subroutine
	IF		Judges result of expression
	INPUT	A	Input from keyboard
		B	
	INKEY		Assigns input from keyboard to variable
	LET		Assigns value of expression to variable
	LOCATE	A	Moves cursor position
		B	
		C	
		D	
	LPRINT	A	Prints out data
		B	
		C	
	ONGOSUB		Moves to subroutine in line number specified by value of expression
	ONGOTO		Moves to line number specified by value of expression
	OPEN	A	Opens specified RS-232C/RS-422 channel.
		B	
		C	
		D	
	PEEK		Reads 1-byte data from specified memory address
	POKE		Writes 1-byte data to specified memory address
	PRINT	A	Displays data on screen
		B	
	REM		Used to write comment.
	SIZE	A	Displays text program capacity
		B	
	STOP		Stops program run
	ZCOFF	A	Underline Type A when the console is GPP/HGP. Type B when the console is VT-220.
		B	
	ZCON	A	Resets underline (used after "ZCOFF"). Type A when the console is GPP/HGP. Type B when the console is VT-220.
		B	
	ZCRV		Reverses character color on CRT screen
	ZDATE		Reads year, month, day, hour, and minute
	ZIDV	A	Changes input console
		B	
		C	
		D	
	ZMOV		Transfers data from memory to memory
	ZNOR		Returns the character reversed after "ZCRV" to its original color

Table 3.2 BASIC Command List (Continue)

Command		Function
Program command [PRG]	ZODV [A] [B] [C]	Changes output console
	ZRD1	Reads 1-byte data from specified channel
	ZRD2	Reads 2-byte data from specified channel
	ZTIME	Stops execution for specified interval of time
	ZWR1	Writes 1-byte data to specified channel
	ZWR2	Writes 2-byte data to specified channel
Intrinsic function [INT]	ABF	Absolute value of real number in mathematical expression
	ABS	Absolute value of integer in mathematical expression
	ACOS	Arccosine ($\cos^{-1}$) of mathematical expression
	ASIN	Arcsine ($\sin^{-1}$) of mathematical expression
	ATAN	Arctangent ($\tan^{-1}$) of mathematical expression
	COS	Cosine (cos) of mathematical expression
	EXP	Value of Exponential to base "e" (e = 2.718281)
	LN	Value of natural logarithm ($\log_e$)
	LOG	Value of common logarithm ($\log_{10}$)
	NOT	Generates "1" when value of mathematical expression is "0" and generates "0" when the value is not "0".
	RND	Assigns random number to variable
	SIN	Sine (sin) of mathematical expression
	SQRT	Value of square root of mathematical expression
	TAN	Tangent (tan) of mathematical expression
Arithmetic operator [ALU]	+	Addition
	−	Subtraction
	*	Multiplication
	/	Division
	^	Exponent
	−	Sign reversion
	%	Remainder calculation

Table 3.2 BASIC Command List (Continue)

Where commands have several options (indicated as [A], [B], [C] etc.) only those shaded([ ]) may be used on the AD51.

Command		Function
Comparison operator COM	=	Is equal to
	#	Is not equal to
	<	Is less than
	>	Is greater than
	<=	Is not greater than
	>=	Is not less than
Logical operator LOG	#	Negation (NOT)
	&	Logical product (AND)
	!	Logical sum (OR)
	¥	Exclusive logical sum (EXOR)

Table 3.2 BASIC Command List

3.2.2 System subroutines

System subroutines are machine code programs used for special AD51 functions (for example PC CPU transactions etc.). They are already written in channel 0 of the AD51 at specified address locations. System subroutine operation is initiated by using the "CALL" instruction in the BASIC program.
System subroutines on the AD51 are shown in Table 3.3.

[Initializing the system subroutine]

1) The system subroutine is called from the GPC-BASIC program using the "CALL" command.

2) The format of the CALL statement is as follows.

A = CALL (variable 1, variable 2, [variable 3, variable 4])

Variable 1: Always 0. All system subroutines are located in channel 0.
Variable 2: Head address of system subroutine in channel 0 (See table 8.2)
Variable 3: Variable for system subroutine stored in (D)(E) registers.
Variable 4: Variable for system subroutine stored in (B)(C) registers.

3) For information on variable 3 and variable 4, refer to the GPC-BASIC Handbooks.

4) Before executing the CALL command, transfer variables to the work area.

	System Subroutine	Function	Channel	Address
1	SAI	ASCII (hexadecimal) → BIN	0	8060H
2	SIA	BIN → ASCII (hexadecimal)	0	8063H
3	SAN	ASCII (decimal) → BIN	0	8072H
4	SNA	BIN → ASCII (decimal)	0	8075H
5	SAF	ASCII → real number	0	8066H
6	SFA	Real number → ASCII	0	8069H
7	SBF	Integer → real number	0	806CH
8	SFB	Real number → integer	0	806FH
9	SBD4	BIN → 4-digit BCD	0	8042H
10	SDB4	4-digit BCD → BIN	0	8045H
11	SBD6	BIN → 6-digit BCD	0	8048H
12	SDB6	6-digit BCD → BIN	0	804BH
13	SBA	BIN addition (24 bits)	0	804EH
14	SBS	BIN subtraction (24 bits)	0	8051H

Table 3.3 System Subroutine List (Continue)

IB (NA) 66059-A

	System Subroutine		Function	Channel	Address
15	SBM		BIN multiplication (24 bits)	0	8054H
16	SBW		BIN division (24 bits)	0	8057H
17	SCA	— B	Write to clock element (year, month, day, hour, minute, second)	0	803CH
18	SCB	— B	Read from clock element (year, month, day, hour, minute, second)	0	803FH
19	SPC	— B	Discrimination of programmable controller CPU	0	8078H
20	SKC	— B	Programmable controller CPU run/stop check	0	8030H
21	SKR	— B	Remote run of programmable controller CPU	0	8033H
22	SKP	— B	Remote stop of programmable controller CPU	0	8036H
23	SRB	A B	Receives specified byte length of data sent to specified channel	0	8009H
24	SWB	A B	Sends specified byte length of data from specified channel	0	800CH
25	SRC		Reads the number of bytes of data received by specified channel	0	800FH
26	SRF		Reads the number of vacant bytes in receive buffer of specified channel	0	8012H
27	SHX	— B	Controls send/receive data of specified channel by Xon/Xoff codes	0	8015H
28	SHD	— B	Controls send/receive data of specified channel by DR terminal	0	8018H
29	SAE		Converts all send/receive data of specified channel to EBCDIC code	0	801EH
30	SEA		No code conversion of send/receive data of specified channel	0	8021H
31	STC		Reads the number of remaining bytes in send buffer of specified channel	0	801BH
32	SRP		Reads status of specified channel	0	8027H
33	SR2		Reads data from buffer memory	0	8000H
34	SW2		Writes data to buffer memory	0	8003H
35	SADR		Reads data from data memory of programmable controller CPU	0	807BH
36	SADW		Writes data to data memory of programmable controller CPU	0	807EH
37	SADT		Randomly writes data to data memory of programmable controller CPU	0	8081H
38	SADM0		Enters data randomly read from data memory of programmable controller CPU	0	8084H
39	SADM1		Randomly reads data from data memory of programmable controller CPU	0	8087H

Table 3.3 System Subroutine List (Continue)

	System Subroutine	Function	Channel	Address
40	SAAR	Reads sequence program	0	808AH
41	SAAW	Writes sequence program	0	808DH
42	SAPR	Reads programmable controller CPU parameters	0	8090H
43	SAPW	Writes programmable controller CPU parameters	0	8093H
44	SAPS	Analysis request of programmable controller CPU parameters	0	8096H
45	SIT	Interruption to programmable controller CPU	0	802AH
46	SIR	Reads error code	0	8024H
47	SC2	Sets SR2/SW2 retry time	0	8006H

Table 3.3 System Subroutine List

Where system subroutines have several options (indicated as [A], [B]) only those shaded ([ ]) may be used on the AD51.

REMARKS

System subroutines indicated as [—/B] are covered in the GPC-BASIC Handbooks.

3.3 Software Configuration

The following shows a block diagram of the AD51 software configuration indicating how the various areas interact.

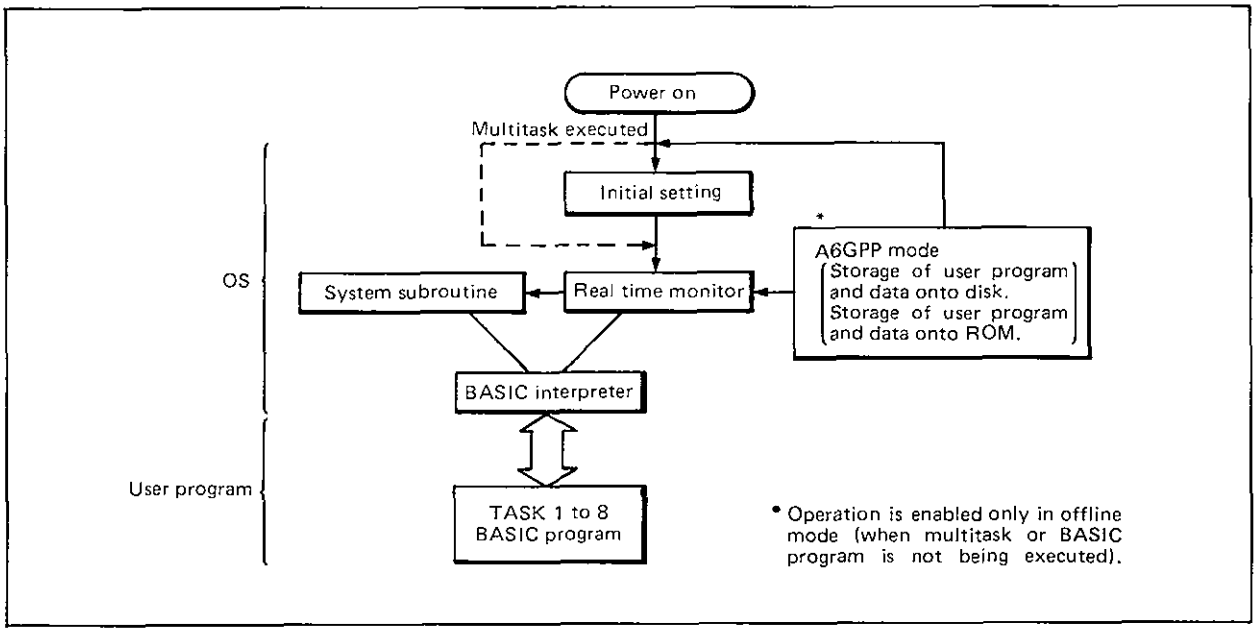


Fig. 3.1 Software Configuration

(1) As shown in Fig. 3.1, a maximum of 8 user programs may be processed in parallel under the control of the real time monitor.

(2) "Power on", "interrupt from ACPU", and "real time interrupt" are available as starting conditions for the user program.

(3) Each task can only be written in BASIC.

3.4 Memory Map

3.4.1 Memory configuration

The AD51 is a Z-80 based system. To expand the memory size from 64K bytes the second 32K bytes are duplicated in additional channels as shown in Fig. 3.2 below.

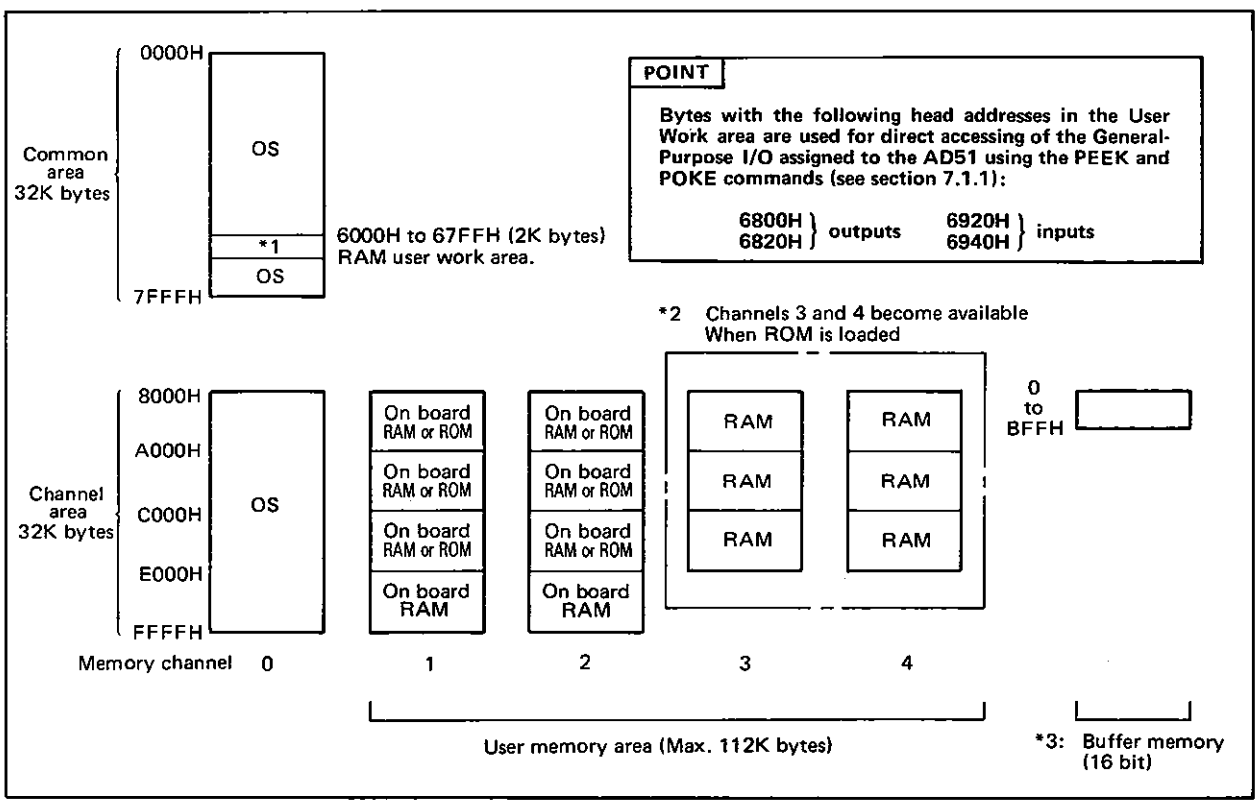


Fig. 3.2 Memory Map

POINT

***(1) The RAM user work area, (addresses 6000H to 67FFH) is in the common area and can be accessed by the user programs in any of channels 1 to 4.**

***(2) The memory area may be expanded by adding ROM to channels 1 and/or 2.**

***(3) The buffer address range is 000H to BFFH which represents 3K words (6K bytes) of buffer memory. Each buffer memory address represents 1 word. (i.e. 16 bits)**

IB (NA) 66059-A

3.4.2 Memory map with ROM loaded

The user memory area can be expanded by loading ROM into channels 1 and/or 2. The memory map will vary as shown below depending on the location and size of the ROMs.
8K and 16KROMs may be used.

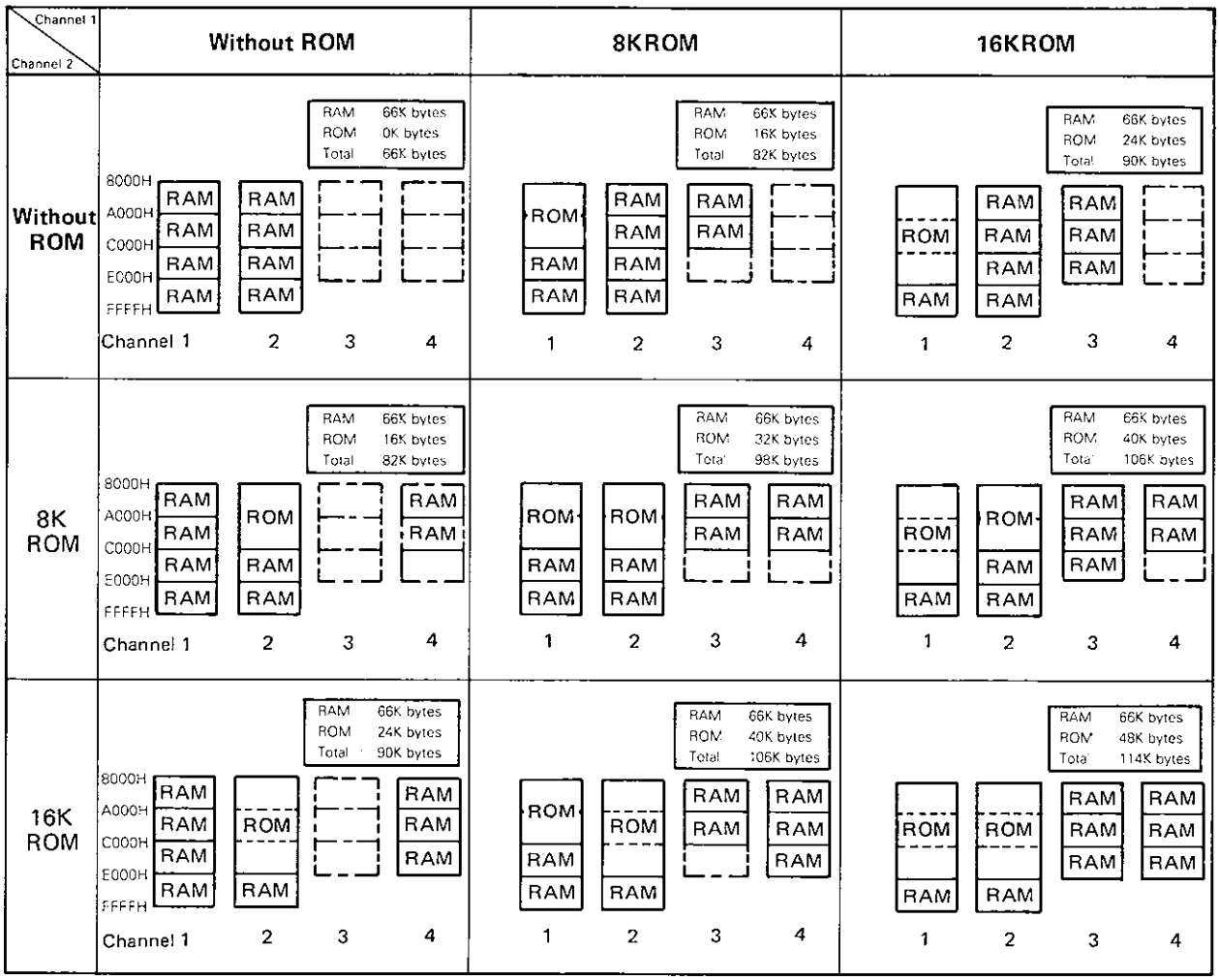


Table 3.4 Memory Map When ROM is used.

POINT

(1) The installation of a ROM shifts the corresponding RAM address range to a different channel. (e.g. Installing 8K of ROM at channel 1 addresses 8000H to BFFFH moves the RAM area to channel 3 addresses 8000H to BFFH.

(2) Two shorting pins are used to specify RAM or ROM in channels 1 and 2. The RAM area is moved as follows depending on the pin setting in each channel.

ROM loading / RAM/ ROM setting pin	With ROM	Without ROM
RAM position	RAM and ROM areas coincide.	Correct
ROM position	Correct	The RAM area changes channels as though ROM was installed.

(3) Only 24K bytes are valid (addresses 8000H to DFFFH) when 16K ROM is used.
The 8K bytes from E000H to FFFH are used in the RAM area and this address range in the ROM cannot be accessed.

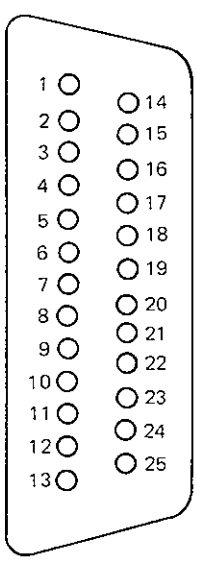
(4) For RAM area memory protect, refer to Section 4.3.1 (page 4-5).

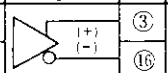
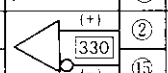
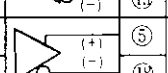
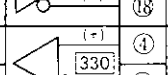
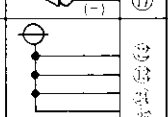
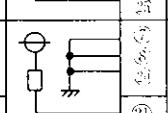
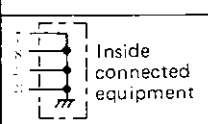
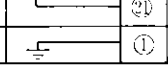
3.5 Interface Specifications

3.5.1 RS-422 connector specifications (CH1)

Item	Specifications
Connected Unit	A6GPP, printer with RS-422, personal computer, etc.
Transmission system	Conforms to EIA. RS-422.
Synchronous system	Asynchronous system
USART mode setting	Baud rate setting (300, 600, 1200, 2400, 4800, 9600 BPS selectable) Parity bit setting: Parity absent / Parity present (Even parity / Odd parity) Stop bit setting: Stop bit 1 / Stop bit 2 Character data bit setting: Data 7 bits / Data 8 bits Communication control setting: XON/XOFF control / Control with DTR terminal

Connector pin outs.



Signal		Block Diagram	Pin	Signal Direction	Remarks
Send data	SDA		③	→ outside	
	SDB		⑯		
Receive data	RDA		②	← outside	
	RDB		⑮		
Data terminal ready	CSA		⑤	→ outside	
	CSB		⑱		
Data set ready	RSA		④	→ outside	
	RSB		⑰		
DC current					Do not perform wiring
Signal ground	SGA				 Inside connected equipment
	SGB		㉑		
Frame ground	FG		①		

*Connect pin 21 to the signal ground of the external equipment.

POINT

(1) The maximum transmission speed from the AD51 is 9600 BPS, the maximum receiving speed is 4800 BPS.

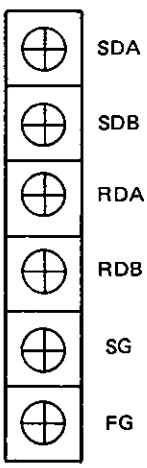
(2) When channel 1 has been set as an I/O console (DIP switch 16 set to ON), the AD51 operating system automatically sets the USART mode:
4800 BPS, even parity, stop bit 1, character data 8

IB-NA-66059-A

3.5.2 RS-422 terminal block specifications (CH2)

Item	Specifications
Connected Unit	AD51, personal computer etc.
Transmission system	Conforms to EIA. RS-422.
Synchronous system	Asynchronous system
USART mode setting	Baud rate setting (300, 600, 1200, 2400, 4800, 9600 BPS selectable) Parity bit setting — Parity absent / Parity present — Even parity / Odd parity Stop bit setting — Stop bit 1 / Stop bit 2 Character data bit setting — Data 7 bits / Data 8 bits

Terminal block pin outs.



Signal	Block Diagram	Terminal Number	Signal Direction	Remarks
Send data (SDA)	(+)	TB1	→ outside	
Send data (SDB)	(−)	TB2		
Receive data (RDA)	(+)	TB3	← outside	
Receive data (RDB)	330 (−)	TB4		
Signal ground (SG)		TB5		
Frame ground (FG)		TB6		

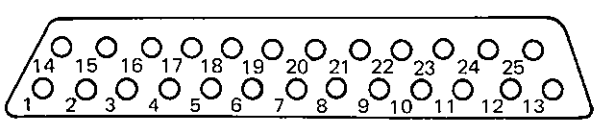
POINT

The maximum transmission baud rate from the AD51 is 9600 BPS. The maximum receiving baud rate is 4800 BPS.

3.5.3 RS-232C connector performance specifications (CH3 and 4)

Item	Specifications
Connected Unit	Console (CH3 only), computer with RS-232C interface, personal computer, printer, modem, etc.
Transmission system	Conforms to EIA. RS-232C.
Transmission speed	300, 600, 1200, 2400, 4800, 9600 selectable
Synchronous system	Asynchronous system
USART mode setting	Baud rate setting (300, 600, 1200, 2400, 4800, 9600 BPS selectable) Parity bit setting — Parity absent / Parity present — Even parity / Odd parity Stop bit setting — Stop bit 1 / Stop bit 2 Character data bit setting — Data 7 bits / Data 8 bits Communication control setting — XON/XOFF control / Control with DTR terminal *Set CH3 with the front DIP switches (SW1 to 8).

Connector pin outs.



Pin Number	Signal Abbreviation	Signal Direction Inside-outside	Description
1	FG		Frame ground
2	SD	→	Send data
3	RD	←	Receive data
4	RTS	→	Request to send
5	CTS	←	Clear to send
6	DSR	←	Data set ready
7	SG		Signal ground
20	DTR	→	Data terminal ready

POINT

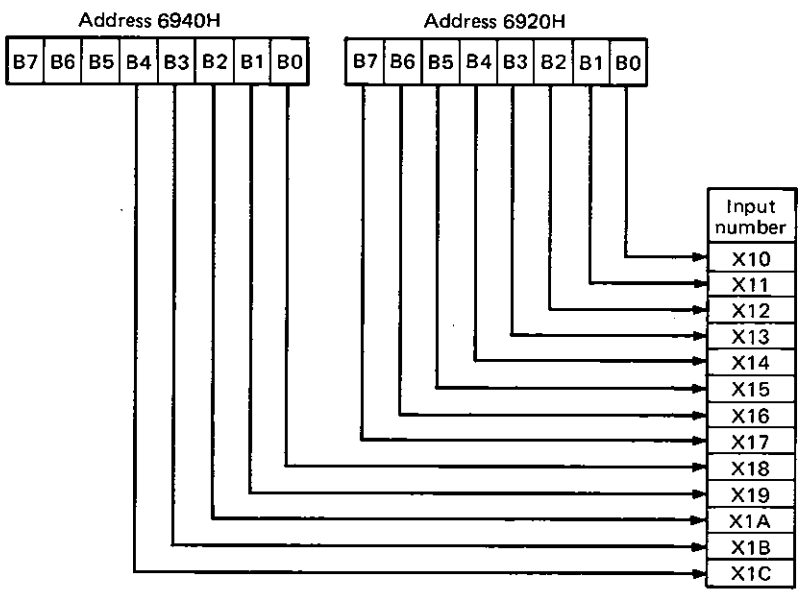
(1) The maximum transmission speed from the AD51 is 9600 BPS, the maximum receiving speed is 4800 BPS.

(2) When channel 3 has been set as an I/O console (DIP switch 16 set to OFF), the AD51 operating system automatically sets the USART mode:
4800 BPS, parity absent, stop bit 1, character data bit 8

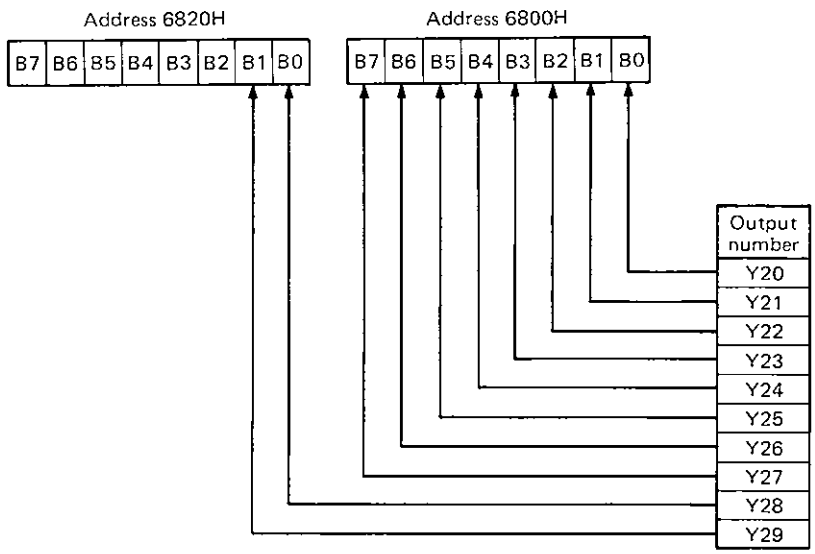
3.6 I/O Interface with Programmable Controller CPU

The digital I/O bus may be used for communication between the PC CPU and the AD51. The following table indicates the function of each signal. The drive number will vary depending on the AD51 slot location; in the table the AD51 is assumed to be in slots 0/1 of the main base unit.

(1) There are 48 input signals to the PC CPU (X00 to X2F) from the AD51.

<table>
<tr><th>Input Number</th><th>Description</th><th>Address</th></tr>
<tr><td>X00 to X0F</td><td>Unused</td><td></td></tr>
<tr><td>X10
X11
X12
X13
X14
X15
X16
X17
X18
X19
X1A
X1B
X1C</td><td>Switched on/off by the BASIC program and the contacts used in the sequence program. (Known as General-purpose inputs)</td><td rowspan="3"> 
</td></tr>
<tr><td>X1D</td><td>Switched on to indicate an AD51 CPU fault</td></tr>
<tr><td>X1E
X1F</td><td>Unused</td></tr>
<tr><td>X20 to X2F</td><td>Unused</td><td></td></tr>
</table>

(2) There are 48 output signals from the PC CPU to the AD51.

<table>
<tr><th>Output Number</th><th>Description</th><th>Address</th></tr>
<tr><td>Y00
to
Y0F</td><td>Unused</td><td rowspan="2"></td></tr>
<tr><td>Y10
to
Y1F</td><td>May be used by the PC CPU as extra internal relays (M).</td></tr>
<tr><td>Y20</td><td rowspan="8">Switched on/off in the sequence program and read by the BASIC program (known as General-purpose outputs)</td><td rowspan="9"> 
</td></tr>
<tr><td>Y21</td></tr>
<tr><td>Y22</td></tr>
<tr><td>Y23</td></tr>
<tr><td>Y24</td></tr>
<tr><td>Y25</td></tr>
<tr><td>Y26</td></tr>
<tr><td>Y27</td></tr>
<tr><td>Y28</td></tr>
<tr><td>Y29</td><td>This output may be used to start one task in the AD51 designated as an interrupt program by its task start condition.</td></tr>
<tr><td>Y2A
to
Y2F</td><td>Unused</td><td></td></tr>
</table>

POINT

(1) Input X1D is switched on when an error is detected by the AD51 hardware. This signal may be used as an interlock in the PC sequence program to control PC accessing of the AD51 buffer memory (i.e. FROM/TO instructions).

(2) Switching output Y29 on will start the task which has its start condition specified as "interrupt from ACPU". The start condition is defined during "multi task setting".

(3) Output signals Y2A to Y2F are used by the operating system and must not be switched on or off.

IB (NA) 66650-A

3.7 Buffer Memory

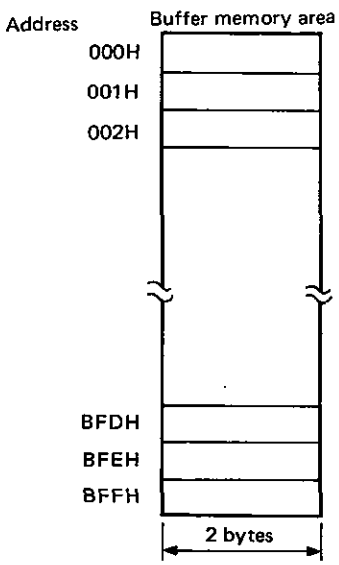
The AD51 uses a buffer memory for data communication with the PC CPU. (The buffer memory is not battery backed.)

(1) Buffer memory addresses are 000H to BFFH (3K words). See the memory map in Section 3.5 (page 3-13).

(2) Buffer memory data is made up of 16 bits per address.

(3) The buffer memory is accessed by the AD51 using system subroutines (SR2, SW2). For details, refer to Section 7.2.1 (page 7-6).

(4) The buffer memory is accessed by the PC CPU using the FROM and TO application instructions. For read and write procedures, refer to Section 7.2.2 (page 7-7). For details of the FROM/TO instructions, refer to the A1, A2, A3CPU Programming Manual.



3.8 Communication between AD51 and PC CPU

Any AD51 initiated requests for communication transactions between the AD51 and PC CPU are processed once when the END, FEND or COM instruction is executed by the PC. The time taken to process a system subroutine and the delay times caused by multiple accessing of the PC CPU are explained in this section.

(1) The following table shows the number of scans taken by the AD51 to process PC transaction subroutines.

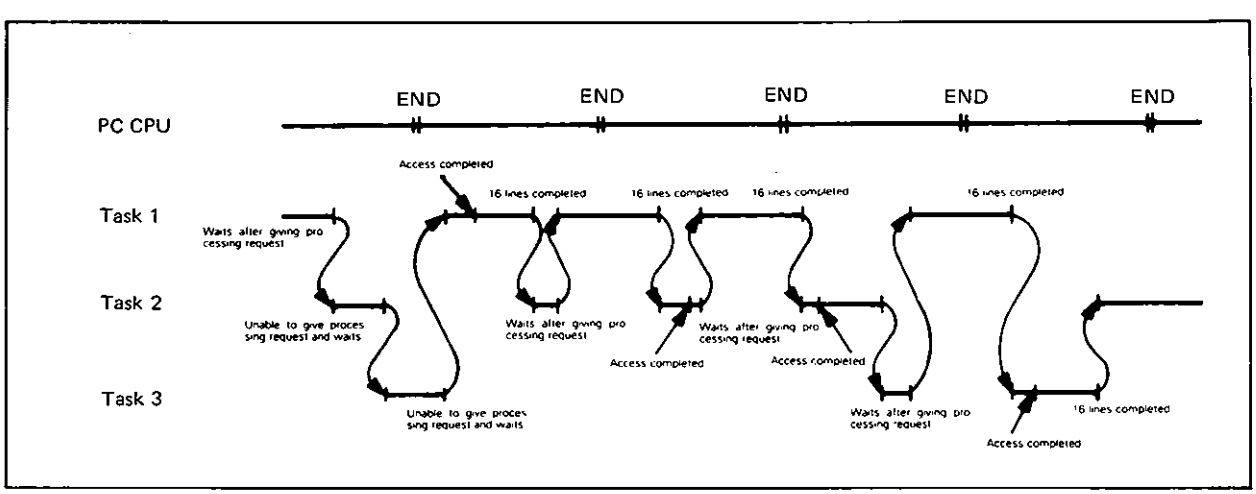
Item			System Subroutine	Number of Scans Required for Processing
Device memory	Batch read	Bit	SADR	1 scan 2 scans for device "R"
		Word		
	Batch write	Bit	SADW	
		Word		
	Test (random write)	Bit	SADT	
		Word		
	Monitor data entry	Bit	SADM0	Independent of scan
		Word		1 scan for device "R" only Independent of scan for other devices.
	Monitor	Bit	SADM1	1 scan
		Word		
Sequence program	Read	Main	SAAR	1 scan
		Sub		
	Write	Main	SAAW	2 scans 1 scan for T/C set value
		Sub		
Parameter	Read		SAPR	2 scans
	Write		SAPW	2 scans
	Analysis request		SAPS	2 scans
Parameter CPU (PC)	Remote run		SKR	1 scan
	Remote stop		SKP	
	PC type mode		SPC	
Buffer memory	Batch read		SR2	Independent of scan
	Batch write		SW2	

(2) Requests for communication transactions with the PC CPU may also come from other sources, these are listed below and are processed in the same way as AD51 transaction requests. Only one transaction may be processed per PC CPU scan so that a delay of 1 to 5 scans is possible before the AD51 transaction is processed if several of these requests overlap. The following list gives the transaction requests in priority order.

1. Programmable controller CPU OS program
2. Peripheral equipment (e.g. A6GPP)
3. Optical or coaxial data link unit incorporated in CPU unit
   A1(E)CPUP21/R21
   A2(E)CPUP21/R21, A3(E)CPUP21/R21
4. Optical or coaxial data link unit in 3 hierarchy system
   AJ71P22, AJ71R22
5. Processing request from AJ71C24 or second AD51

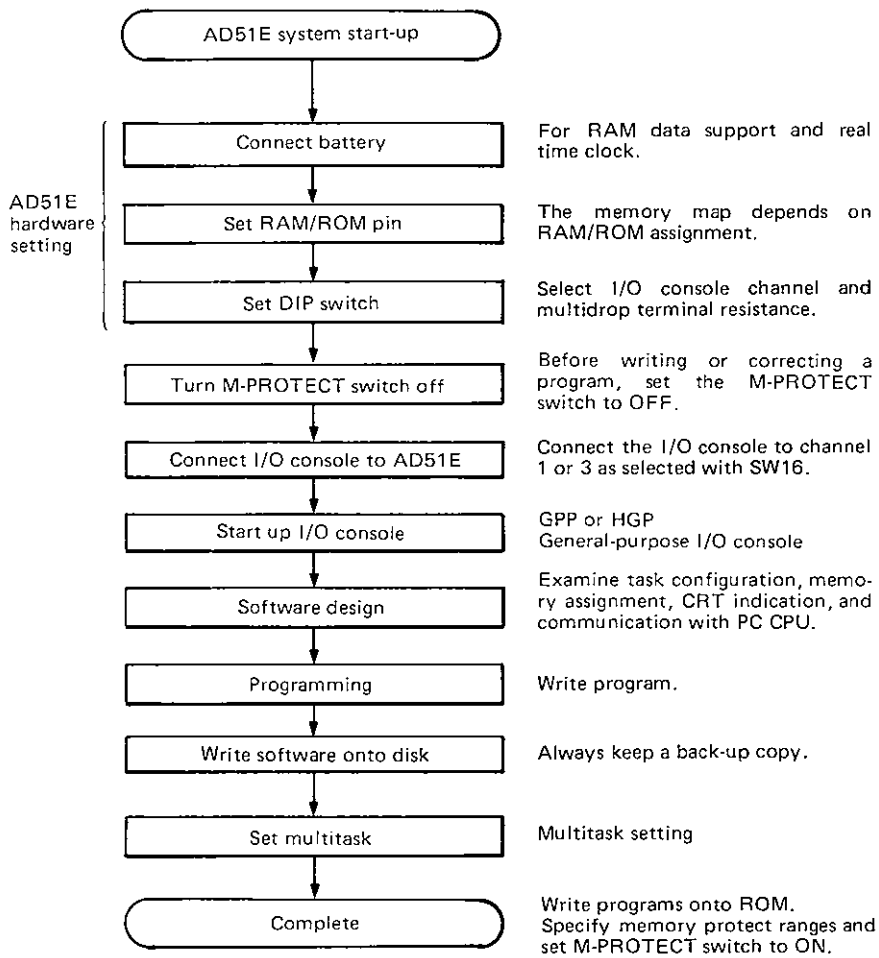
Hence, if continuous processing requests are received from the A6GPP and AJ71C24, communication between the AD51 and PC CPU is only made once every three scans.

(3) When a system subroutine is called which accesses the PC CPU there is a delay while the PC CPU prepares the appropriate data. During this delay time, the AD51 switches tasks to optimize scan time. In the example shown below, three tasks are executing subroutine which access the PC CPU. Task one provides the first processing request to the PC CPU which prepares the requested data. During this delay the AD51 switches to task 2 which is unable to pass its processing request to the PC which is still dealing with the one from task 1. The AD51 therefore switches to task 3 for which the same situation exists. Only after task 1s request has been fully processed can task 2s request be dealt with. Similarly task 3 must wait until task 2 been processed. For details of other task switching, refer to the GPC-BASIC Handbooks.



4. PRE-START UP PROCEDURES

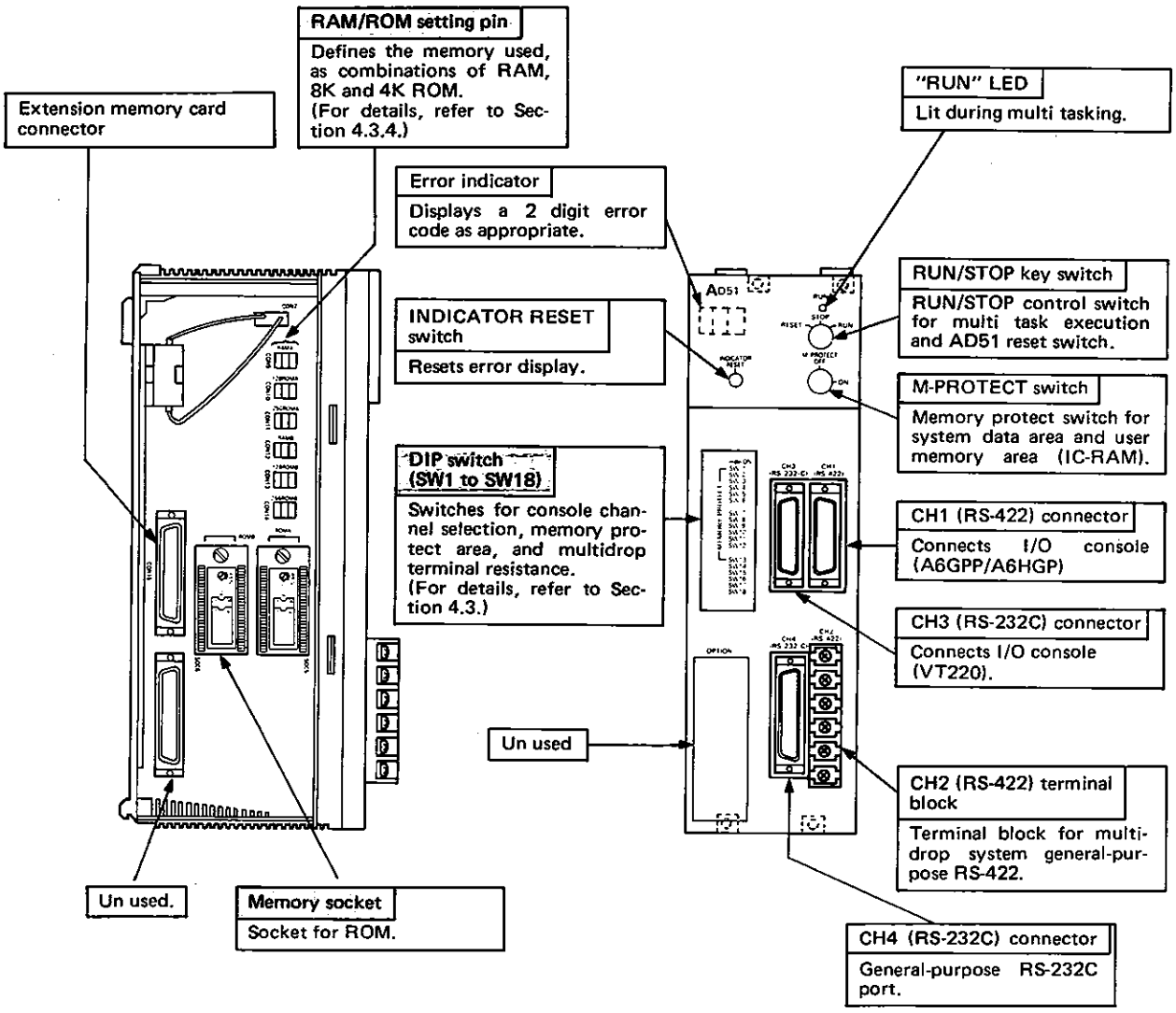
4.1 General Procedure



BNAI 66059-A

4.2 Nomenclature

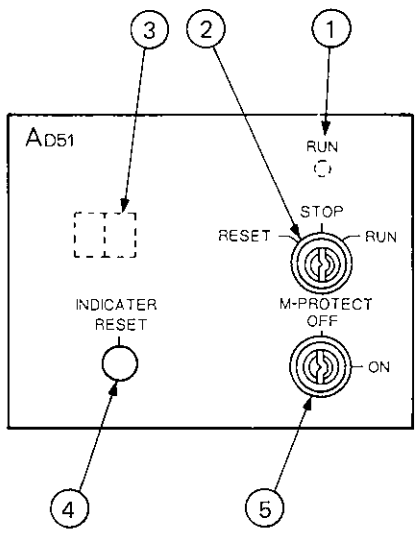
For the setting switches, refer to Section 4.3 (page 4-5).



REMARKS

It is necessary to set or load [ ] before starting operation.

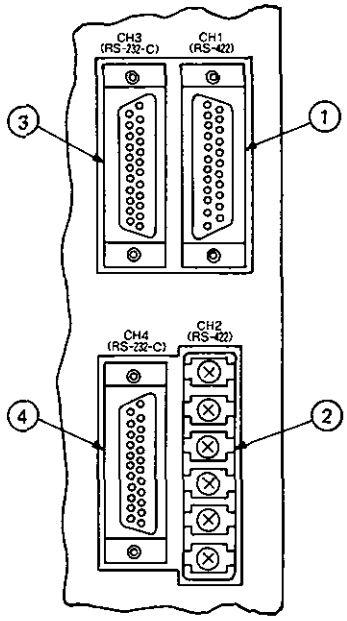
(1) Switch details



①	"RUN" LED On . . . . .During multi tasking Off . . . . .AD51 is not multitasking. (Will remain off when a single task is RUN.)
②	"RUN-STOP-RESET" switch RUN/STOP . . .○Set to RUN to enable multi tasking to be started from the input console. ○Set to RUN to enable a single task to be started by typing RUN during BASIC programming, debugging etc. ○Set to STOP to stop program execution. RESET . . . . . .Used to reset an error or to initialize multi tasking.
③	Error indicator Displays a 2 digit error code as appropriate.
④	"INDICATOR RESET" switch ○Resets the error code display after the error has been removed. The error code will remain if the error has not been cleared. ○When several errors have occurred, pressing the reset switch will display consecutive error numbers in the order that they occured.
⑤	"M-PROTECT" switch ○Memory protect for system data (the data entered during multi task setting and BASIC programming) and the user memory. ○The memory protect range depends on the DIP switch settings on the front of the unit. For DIP switch details, refer to Section 4.3 (page 4-5).

IB (NA) 66059-A

(2) Connector details



①	CH1 (RS-422) connector Set DIP switch SW16 ON to use this port for programming with the GPP or HGP. Set DIP switch SW16 OFF, to use this connector as a general-purpose port.
②	CH2 (RS-422) terminal block The terminal block is provided for use in the AD51 multidrop system. A maximum of 32 AD51 stations may be included in the multidrop link. This connector may also be used as a general-purpose RS-422 port.
③	CH3 (RS-232C) connector Set DIP switch SW16 OFF to use this port for programming with the VT-220 or GPP/HGP set DIP switch SW16 ON to use this connector as a general-purpose port.
④	CH4 (RS-232C) connector General-purpose RS-232C port.

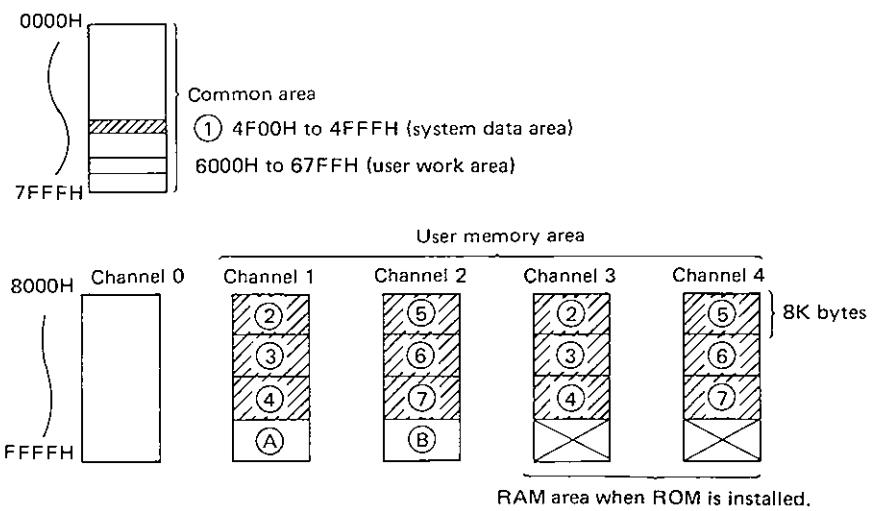
4.3 Hardware Settings

4.3.1 Memory protect range

The maximum user program memory capacity is 114K bytes (48K bytes for ROM + 66K bytes for RAM). 48K bytes of the RAM area may be memory protected in units of 8K bytes. The system data area (for multi task setting data etc.) may also be memory protected. To set memory protection to the required area, use the DIP switches on the front of the unit as described below:

(1) Memory protect area

Areas marked ▨ in the memory maps below can be memory protected.



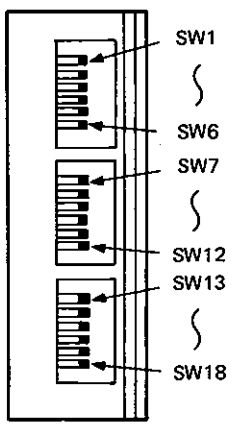
REMARKS

(1) The memory protect DIP switch number is shown as ① to ⑦ in the above memory map.

(2) The memory protect DIP switch numbers for a given RAM address range remain unchanged when ROM is loaded although the channel number has changed from channel 1 to 3 or from 2 to 4.

(2) Memory protect range

Protected RAM address ranges are shown below. The DIP switch is on when the lever points to the right, this is marked on the switch cover.

Division	DIP Switch Details	Switch Number	Memory Protect Range		
			Memory channel		Memory address
			For RAM only	ROM loaded	
①		SW1	Common area		4F00H to 4FFFH
②		SW2	1	3	8000H to 9FFFH
③		SW3	1	3	A000H to BFFFH
④		SW4	1	3	C000H to DFFFH
⑤		SW5	2	4	8000H to 9FFFH
⑥		SW6	2	4	A000H to BFFFH
⑦		SW7	2	4	C000H to DFFFH
Unused		SW8	Unused		
		SW9			
		SW10			
		SW11			
		SW12			
		SW13			

POINT

(1) BASIC program address data and multi task setting data is stored in address ranges 4F00H to 4FFFH (256 bytes). Set memory protect with SW1 after starting multi tasking.

(2) Switching the memory protect key to ON protects all areas defined by the DIP switch settings.

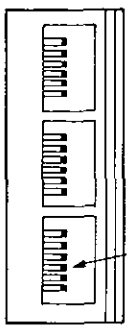
(3) Keep the memory protect switch OFF during BASIC program writing and editing.

(4) The RAM areas marked (A) and (B) on the preceeding page and the user work area cannot be memory protected.

IB (NA) 66059-A

4.3.2 Console channel

DIP switch SW16 determines which of the two channels, CH1 and CH3 is to be used for the programming console.
When the VT-220 is used SW16 should be switched OFF defining CH3 (RS-232C) as the programming console port. When the GPP/HGP is used the switch is generally switched ON defining CH1 (RS-422) as the programming console port. (RS-232C may also be used)

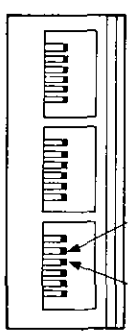
DIP switch Details	SW16 Position	CH1 (RS-422)	CH3 (RS-232C)
	ON	GPP/HGP	General-purpose port
	OFF	General-purpose port	VT-220 (GPP/HGP)

POINT

The console setting switch is valid after the AD51E is powered up or reset. When the console setting has been changed, reset the AD51E.

4.3.3 Terminal resistor

A terminal resistor is fitted to prevent distortion of the transmission signal waveform. When a number of AD51s are connected together via an RS-422 link, the two end stations should be set with "terminal resistor present", the remainder with "terminal resistance absent."
DIP switches SW14 and SW15 are used to set the terminal resistance as shown below.

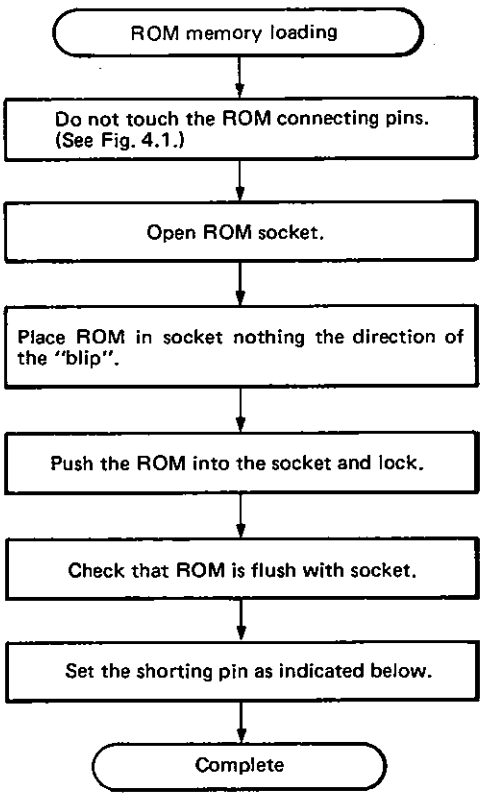
DIP Switch Details	SW14 Position	SW15 Position	Description
	ON	ON	With terminal resistor
	OFF	OFF	Without terminal resistor

POINT

Both switches should be either on or off.

4.3.4 ROM installation

This section describes the installation and settings required for using the ROM. The ROM sockets should be empty if ROM is not being used.



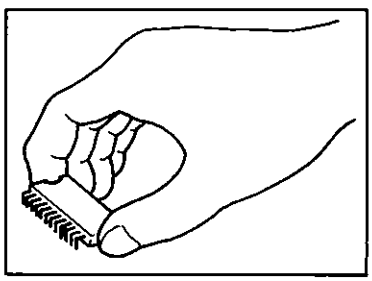


Fig. 4.1 How to Hold ROM

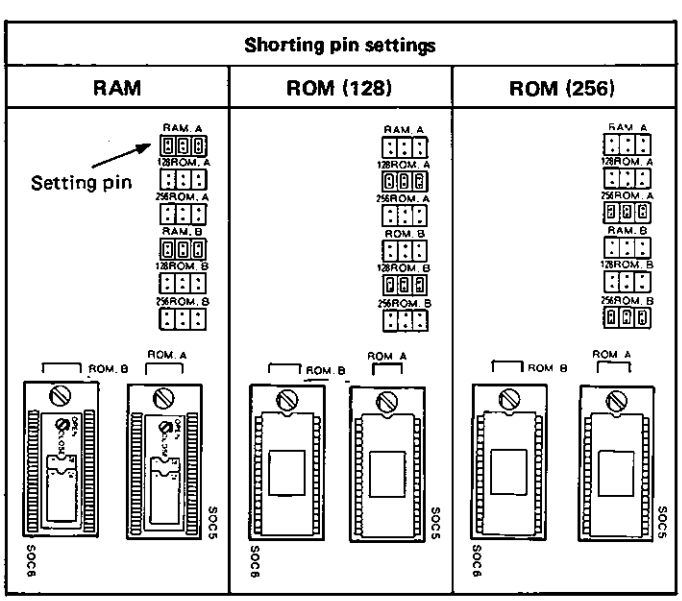


Fig. 4.2 Pin Settings Depending on Memories

POINT

(1) RAM is built into the unit, there is no need to load RAM into either socket.

(2) The correct direction of the "blip" on the ROM is indicated on the ROM socket.

(3) ROM may be loaded into either socket and ROM sizes (27128, 27256) may be mixed providing the address ranges are noted.

(4) When ROM is installed some RAM address ranges change. (For details, refer to Section 3.5 (page 3-13).

(5) Cover the EPROM window after it has been programmed.

(6) Ensure that ROMs are correctly stored and protected.

(7) Keep the ROM away from static electricity – use anti-static foam where possible.

(8) The shorting pin is factory-set to RAM (RAM.A and RAM.B connectors).

(9) Channel 1 shorting pins are marked RAM.A and channel 2, RAM.B.

4.3.5 Loading the battery

The battery is disconnected before leaving the factory to prevent unneccessary battery consumption. The battery plug should be connected to pins CON7 on the circuit board before the AD51 is used. The red wire is positive and the connector is keyed to prevent wrong connection.

5. WIRING

5.1 Wiring Instructions

All AD51 external wiring should be protected against noise.

(1) Keep cables carrying data at least 100mm away from main circuit wiring, high voltage cables and PC input and output wiring.

(2) Ground shield wires or cable shields at one point only.

(3) Use M4 solderless terminals for connection to the RS-422 terminal block.

5.2 RS-232C Connection

RS-232C connection

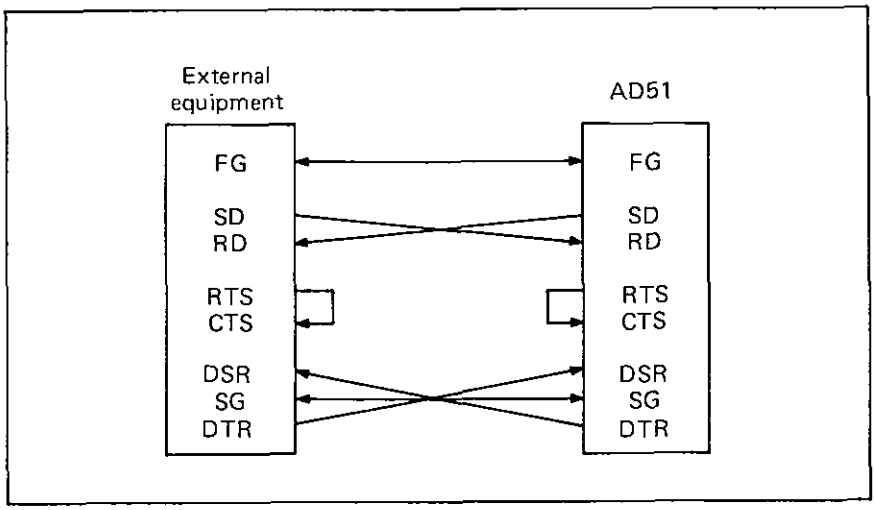
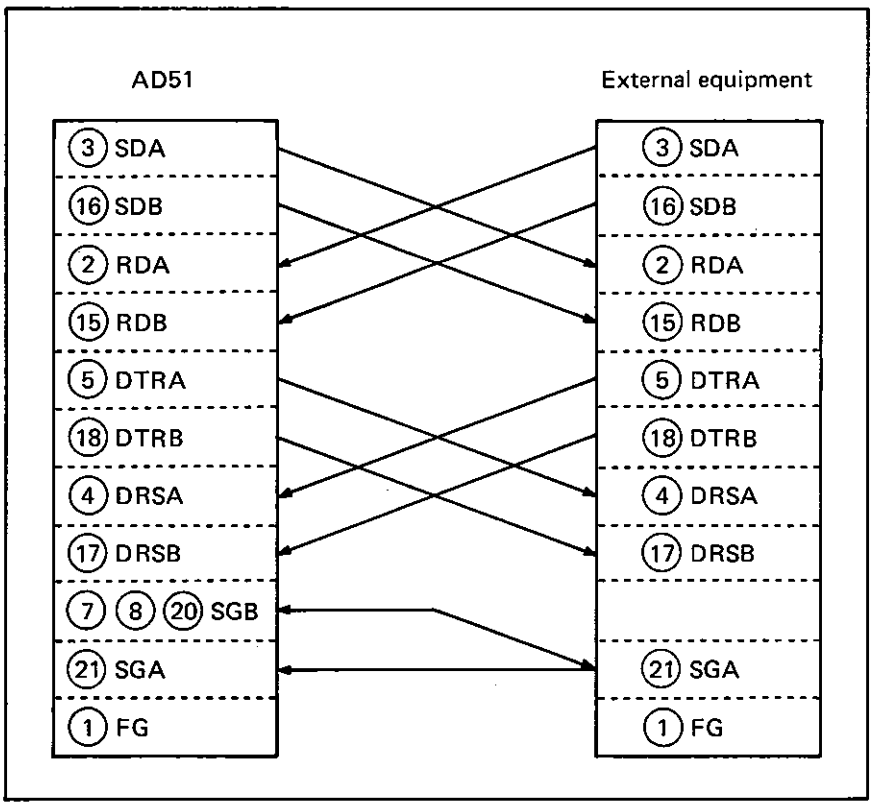


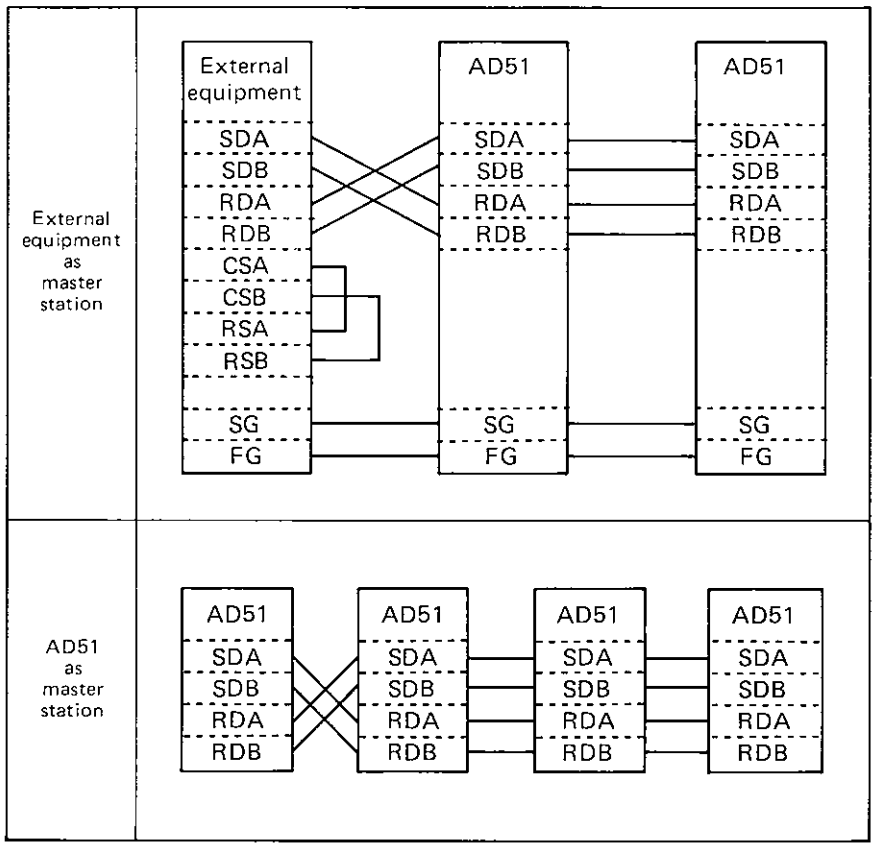
Fig. 5.1 RS-232C Connection Diagram

5.3 RS-422 Connection

(1) RS-422 connector



(2) RS-422 multidrop connection.



POINT

A maximum of 32 stations may be included in the multidrop system with an overall link distance of 500m.

REMARKS

RS-422 multidrop cabling should conform to the following specifications.

Item	Specifications
Cable type	Shielded cable
Conductor resistance (20°C)	88.0Ω/km or less
Insulation resistance	10,000MΩ·km or less
Dielectric strength	500V DC for 1 minute
Electrostatic capacity (1KHz)	60nF/km or less on average
Characteristic impedance (100KHz)	110 ± 10Ω

6. AD51 PROGRAMMING NOTES

6.1 BASIC Program Address Data

The following information must be specified before a GPC-BASIC program can be written: program number, program head address, program last address, additional program head address, work area head address, and channel. For further details on setting the data, refer to AD51E Operating Manual.

REMARKS

The following table indicates the function of each of the addresses. Before the BASIC program can be written a BASIC text area must be defined as well as an interpreter work area. The operating system automatically asigns the additional program head address depending on how much of the BASIC text area is vacant.

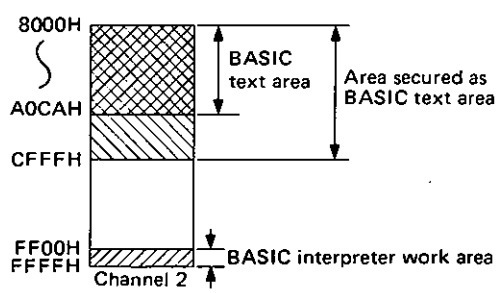
Item	Description
Program number	BASIC text number (1 to 8)
Program head address	The first address of the BASIC text area (8000H onwards)
Program last address	The last address of the BASIC text area
Additional program head address	Head address of vacant area in BASIC text area. (Automatically set by the O.S.)
Work area head address	Work area used for BASIC interpreter. Fixed to 256 bytes. (Not available for user)
Channel	Channel for the BASIC text

(1) Direct variables (A to Z) are allotted in the BASIC interpreter work area.

(2) Use the address range D000H to FEFFH for @ array variables and indirect variables. The hatched areas in the example below may not be used.

Example:
Channel 2 (8000H to FFFFH) data has been set as follows:

1. Program head address 8000H
2. Program last address CFFFH
3. Additional program head address A0CBH
4. Work area head address FF00H
5. Channel 2

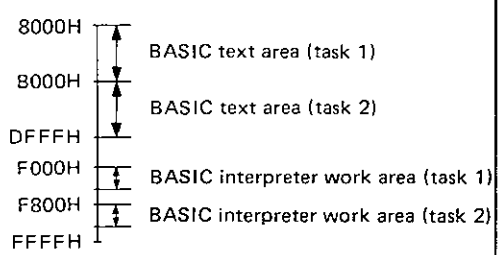
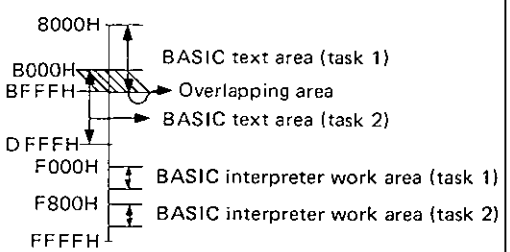
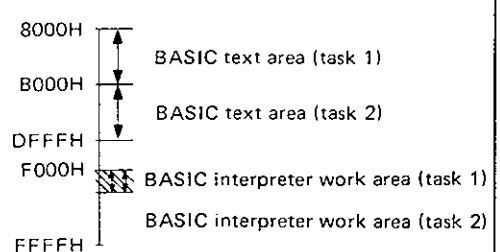


(3) The work area must come after the text area and the work area head address must have 00 in the two least significant digits. 256 bytes are used for the BASIC interpreter work area starting at the work area head address.

(4) When two or more tasks are written in the same channel make sure that the program areas and BASIC interpreter work areas do not overlap each other.

a) Program data will be corrupted in overlapping memory areas.

b) Multitasking results will be invalid if the BASIC interpreter work area for a given task is overlapped by program data from a different task. Independent running of that task however is valid.

	BASIC Program Addresses	TASK 1	TASK 2	Memory Map
Correct example	Task	1	2	
	Program head address	8000H	B000H	
	Program last address	AFFFH	DFFFH	
	Additional program head address	A74CH	C851H	
	Work area head address	F000H	F800H	
	Channel	1	1	
Over-lapping program area	Task	1	2	
	Program head address	8000H	A000H	
	Program last address	AFFFH	DFFFH	
	Additional program head address	A74CH	B851H	
	Work area head address	F000H	F800H	
	Channel	1	1	
Over-lapping interpreter work area	Task	1	2	
	Program head address	8000H	B000H	
	Program last address	AFFFH	DFFFH	
	Additional program head address	A74CH	C851H	
	Work area head address	F000H	F000H	
	Channel	1	1	

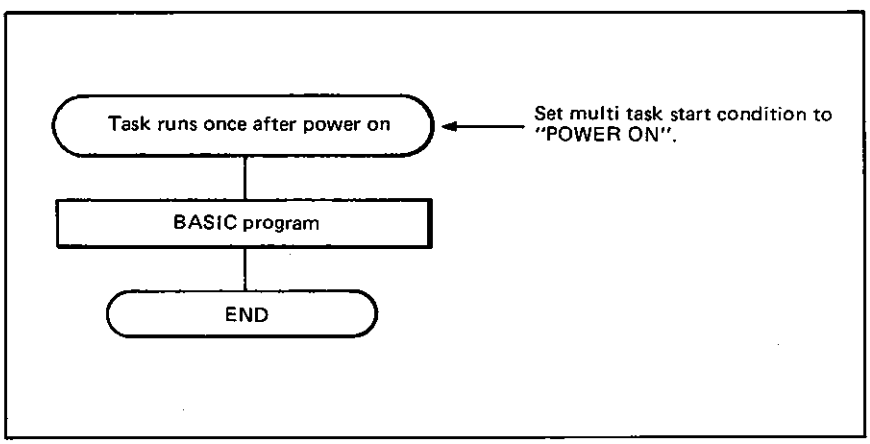
6.2 Start Conditions

There are 4 types of BASIC program execution formats:

(1) Program runs once after power on.
(2) Program runs continuously after power on.
(3) Program runs after an interrupt signal from the PC CPU.
(4) Program runs at preset intervals in real time.

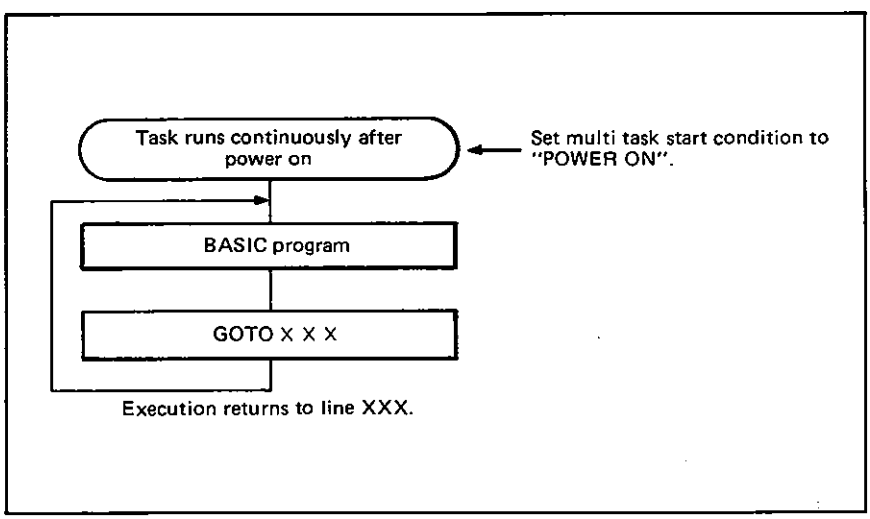
6.2.1 Program runs once after power on

Write the BASIC program so that "END" is executed as the final instruction and set the task start condition to "POWER ON".



6.2.2 Program runs continuously after power on

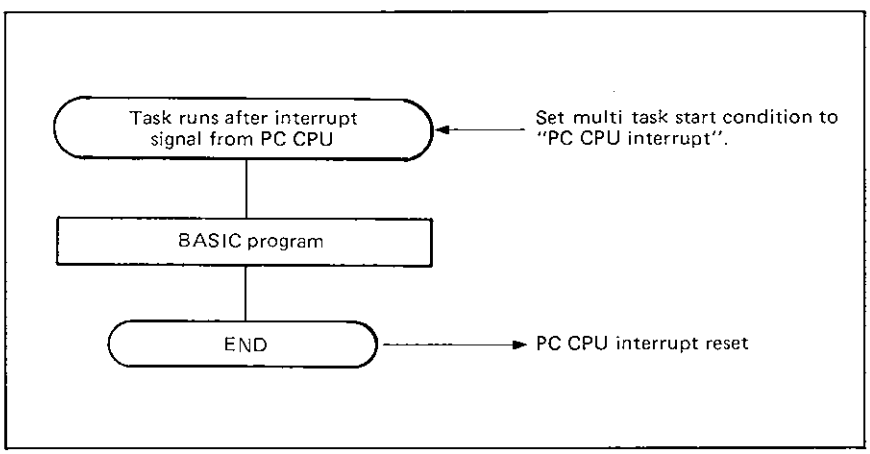
Write the BASIC program using the "GOTO" instruction to continue program execution and set the task start condition to "POWER ON".



6.2.3 Program runs after an interrupt signal from the PC CPU

Set the task start condition to "CPU INT" (CPU interrupt). The program is then run when the rising edge of the AD51 interrupt signal is received from the PC CPU. For programming information see Section 7.4.

(1) Write the BASIC program so that "END" is executed as the final instruction. When "END" is executed the AD51 interrupt condition is reset. The interrupt program will not run again until the rising edge of the AD51 interrupt signal is received from the PC CPU.



(2) Only one task may be defined as PC interrupt start. More than one will generally lead to "ORST" error.

6.2.4 Program runs at preset intervals in real time

Select starting condition "REAL TIME INT" (real time interrupt).

The real time interrupt interval (i.e. the time between interrupt signals) should be longer than the total time taken for the interrupt program to reach the END instruction including the time required by other tasks. An "ORST" error is detected if a second real time interrupt signal is given before the interrupt program has executed its END command.

6.3 Notes on the Use of BASIC Commands

6.3.1 Key input commands

Key inputs (INPUT and INKEY commands) to the AD51 via one channel (as specified by the ZIDV command) should only be made to one task.
Since tasks are executed in order of task numbers, any data keyed in to a task can only be read at certain intervals. If a key is pressed and more than one task is waiting for data from the specified channel, only the first task to execute the INKEY or INPUT instruction will read the key input. The other tasks will then continue waiting until a key is pressed while they are being run.

```
Task 1                          Task 2

100  REM  "TASK 1"              100  REM  "TASK 2"
      ⟨                               ⟨
200  ZIDV 1                     500  ZIDV 1
210  A=INKEY                    510  B=INKEY
      ⟨                               ⟨
```

Example: When both task 1 and task 2 are waiting for key input from channel 1, pressing a key will only write data to one of variables A or B.

6.3.2 Printing commands

The printing commands are "PRINT" and "LPRINT".

(1) Difference between PRINT and LPRINT commands

[PRINT]

Used when the printer is connected to the console channel (channel 1 or 3 set with DIP switch 16) or the channel specified with ZODV.

[LPRINT]

Used when the printer is connected to the channel specified in the printer setting.

(2) Sharing of a single printer between tasks.

When several tasks are sharing the use of a printer ensure that interlock flags are provided in the work area to prevent two or more tasks attempting to access the printer simultaneously.

(3) Note that with printers that use the CR code (0DH) to initiate printing (K6PR, K6PR-K, K7PR, A7PR etc.), writing a comma (,) after the statement in the PRINT or LPRINT command, stops the AD51 from sending the CR code.
Printing is therefore not initiated.

(4) Notes on the use of the KD51PR

The KD51PR may be connected to either of the two AD51 RS-232C ports. When using the KD51PR note the following:

- The KD51PR will print "?" if data is sent from the AD51 while it is printing or during paper feed. This may be avoided by using the program shown in Example 2.

- Example 1 shows a program which repeatedly prints the letters "ABCDE" and the resultant KD51PR print out.

{Example 1}

Use of KD51PR in "2K buffer OFF, buffer full set" mode

BASIC program

```
100 LPRINT "ABCDE"
110 LPRINT CHR$(&H03),
120 GOTO 100
```

Print data and control codes (CR code, LF code) are sent from AD51.

Print control code 03H is sent from AD51.

Upon execution of line 120, execution returns to line 100. Data is continuously written to the KD51PR while it is still printing.

Print result

```
ABCDE
?A?B?D?
?
?E?
?E??D?
?D?E??A?C?E?
```

{Example 2}

KD51PR setting .....2K buffer ON, buffer full set

BASIC program

```
100 LPRINT "ABCDE"
110 LPRINT ¥$03,
120 ZTIME 400
130 GOTO 100
```

Print result

```
ABCDE
ABCDE
ABCDE
```

Print data and control codes (CR code, LF code) are sent from AD51.

Print control code 03H is sent from AD51.

The KD51PR starts printing after it receives the 03H code. The ZTIME instruction allows a time delay before the next set of data is sent.

In this case, it takes 8 seconds from 1-time print command to print termination.

"ABCDE"	print	2 seconds
CR, LF	(line feed)	2 seconds
"ABCDE"	print	2 seconds
CR, LF	(line feed)	2 seconds

- When the 2K buffer is set to OFF in the KD51PR, any string of characters sent which is more than one line long will lead to an overrun error when the LF code is given. (The receive buffer is 32 characters long). See below:

```
100 LRPINT "ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEF"
           └──────────( 32 characters )──────────┘
```

To overcome this, write a comma (,) after the PRINT statement to stop the LF code from being given as below:

```
100 LPRINT "ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEF,"
```

6.3.3 CRT display commands

Any commands addressed to a CRT on a given channel should come from one task only. There is no management of display commands between tasks.

[CRT display commands]

CLS, ZCON, ZCOFF, ZNOR, ZCRV, PRINT, LOCATE

POINT

Any program controlling the display on one CRT should be written in one task only.

6.3.4 OPEN and CLOSE commands

Note the following precautions when using the OPEN and CLOSE commands.

(1) The channel specified as that for the console by DIP switch SW16 and the channel selected for the printer on the printer setting screen are automatically opened by the AD51 OS.
The communication mode for other channels must be set in the user program using the OPEN command. These two channels default to closed.

(2) The OPEN command is used to specify the communication control at the RS-232C or RS-422 port. It also initializes the AD51 send and receive buffers. Executing this command therefore re-defines the communication mode for the specified port and clears both AD51 buffers at that port.

(3) The CLOSE command initializes the AD51 communication control and buffers in the same way as the OPEN command however in this case the port is left in a read/write disable state.

POINT

Before using the CLOSE command, ensure that the transmit buffer is empty be using subroutine "STC". Data will be lost if the buffer is initialized while it still contains data.

(4) OPEN and CLOSE commands are common to all tasks. Therefore, any channel opened by one task can be used in subsequent tasks without having to repeat the OPEN command.

6.3.5 Z commands

The ZMOV, ZRD1, ZRD2, ZWR1 and ZWR2 commands are not available for AD51, PC CPU transactions. Access to the PC CPU data is via system subroutines. For details refer to section 3.2.2.

6.4 AD51 and PC CPU Reset

The following explains the effects of resetting the AD51 and the PC CPU.

(1) AD51 reset operation

1) The AD51 processes its programs as though the power has been switched on.

   With multi task start already set .....Executes multi task.
   Without multi task start .....Displays the mode select menu on the console.

2) All the AD51 general-purpose inputs are switched off.

3) During reset, there is no accessing of the PC CPU.

4) There is no signal by which the PC CPU can know that the AD51 has been reset.

5) Any FROM or TO instructions executed by the PC CPU when the AD51 is reset will be invalidated.

(2) PC CPU reset operation

1) All the AD51 general-purpose outputs are switched off.

2) Resetting the PC CPU disables access by the AD51E to the PC CPU for about five seconds after the PC CPU is set to RUN. If the PC CPU is accessed during this period, "PC DOWN ERROR" or "TIME OUT ERROR" is detected.

6.5 Notes on BASIC Programming

(1) Before making additions or corrections to the BASIC program or changing the program to its final format with the RUN or COMPILE command, set the "M-PROTECT" switch to OFF.

(2) Before executing multi task, remove all STOP and BREAK commands.

(3) Always RUN or COMPILE the BASIC program after it has been completed.
The RUN or COMPILE commands change the program into a format suitable for multitasking. If the program is not formatted the CPU may misoperate. Remember to RUN or COMPILE programs before writing them to ROM.

7. COMMUNICATION WITH PROGRAMMABLE CONTROLLER CPU

The AD51 occupies 48 I/O points and is provided with 13 digital inputs and 10 digital outputs as well as a 3K word buffer memory. The following section describes communication between the PC CPU and the AD51. It is assumed that the AD51 is located at slot 0 and 1 of the main base unit and that any BASIC program is written in channel 1.

POINT

The AD51 can only communicate with the PC CPU at its own station. When using the system subroutines therefore, always specify the station number as FFH (i.e. host).

7.1 General-Purpose I/O Read/Write

The AD51 can access the general-purpose I/O immediately and continuously using the "PEEK" and "POKE" commands. This eliminates the need to use the data memory read/write system subroutines which are only processed when the "END" or "COM" instruction is executed in the PC CPU.
Output Y29 from the PC CPU can be used as an interrupt signal to the AD51, see Section 7.5.

7.1.1 General-purpose I/O addresses

The general-purpose I/O assigned to the AD51 are accessed via the following addresses in the user work area:

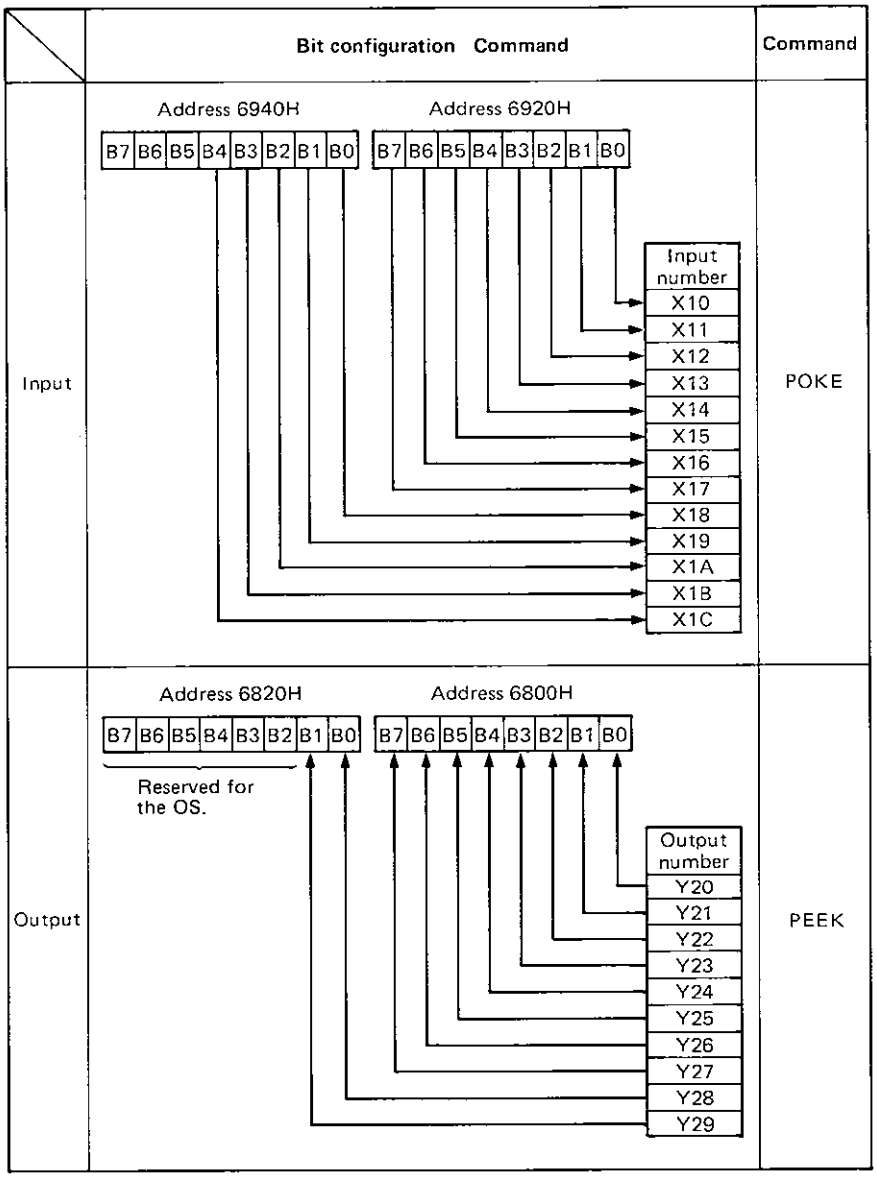
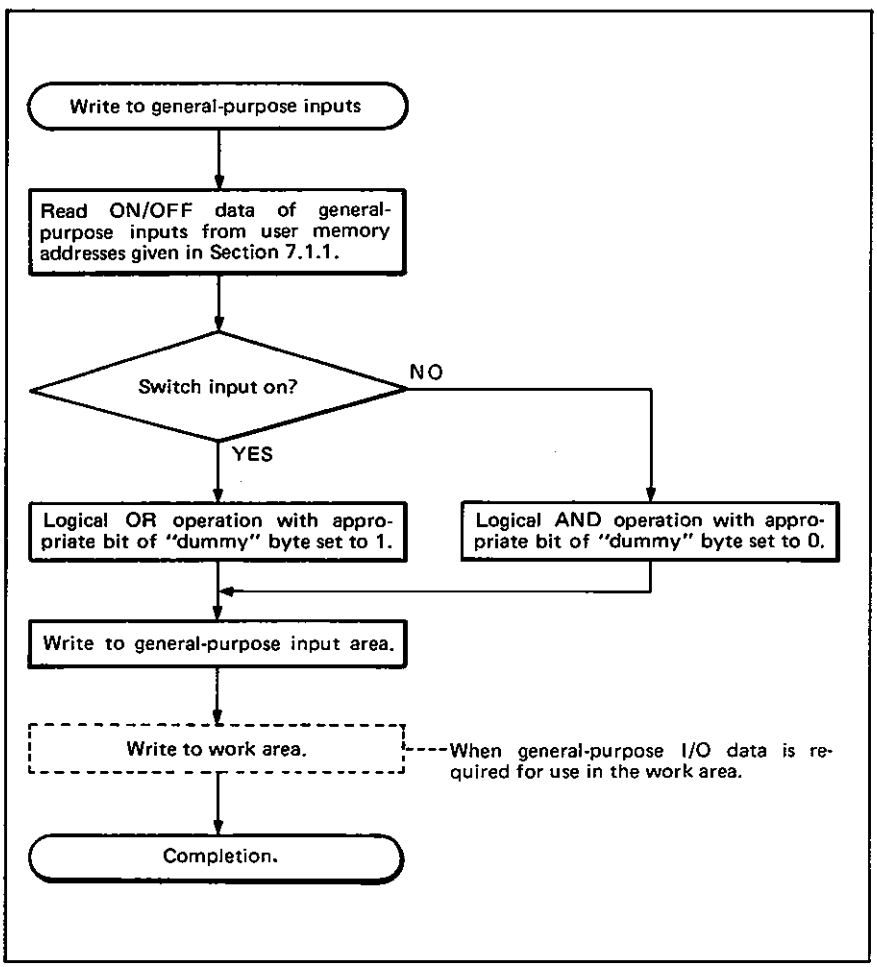


Table 7.1 General-purpose I/O Data Configuration

7.1.2 Write to general-purpose input

To write to a specified general-purpose input the appropriate bit in the relevant byte of memory must be set or reset. The memory addresses for general-purpose inputs and outputs are given in Section 7.1.1.

This section describes the direct accessing of general-purpose I/O data using the PEEK and POKE commands. The system subroutine "SADR" may also be used to access this data but system subroutines are only processed after the "END" or "COM" instruction has been executed in the PC CPU.



[Program example]

Inputs X13 and X1B are turned on and off respectively and the resultant on/off condition is stored in head addresses 6000H and 6001H.

(1) Program to switch on X13

```
100 A=PEEK($6000) . . .Reads address 6000H from the common area
                        (This byte is all '0's).
110 B=A / $08. . . . . .Executes a logical OR operation with address
                        6000H and a "dummy" byte containing the
                        binary representation of 8 as follow:
```

	Bit pattern
Address 6000H	0 0 0 0 0 0 0 0
8_2	0 0 0 0 1 0 0 0
OR result	0 0 0 0 1 0 0 0

The result is written to variable B.

```
120 POKE $6920, B . . . .Variable B is written to address 6920H (i.e. the
                         user work area reserved for accessing the general-
                         purpose I/O).
130 POKE $6000, B . . . .The result is also written to address 6000H.
140 END
```

(2) Program to switch off X1B

```
100 A=PEEK($6001) . . .Reads address 6001H from the common area
                        (This byte is all '0's).
110 B=A&*$F7. . . . . .Executes a logical AND operation with address
                        6000H and a dummy byte containing the binary
                        representation of F7 as follows:
```

	Bit pattern
Address 6000H	0 0 0 0 0 0 0 0
$F7_2$	1 1 1 1 0 1 1 1
AND result	0 0 0 0 0 0 0 0

The result is written into variable B.

```
120 POKE $6940, B . . . .Variable B is written to address 6940H (i.e. the
                         user work area reserved for accessing the general-
                         purpose I/O).
130 POKE $6001, B . . . .The result is also written to address 6001H.
140 END
```

POINT

***: X1D to 1F cannot be turned on/off in the BASIC program. The value marked * in line 110 of (2) may also be 17H (bits 5 to 7 = 0). If AND operation is performed, the same result is obtained.**

7.1.3 Read from general-purpose output

A similar procedure may be used to read the general-purpose outputs using the logical AND operation. The work area addresses are given in Section 7.1.1. Note that bits 2 to 7 of address 6820H are reserved for the operating system and should be masked.

[Program example]

Program for ON/OFF monitor of Y20 to 27

```
100 A=PEEK($6800) . . .Reads address 6800H to variable A.
110 B=1 . . . . . .Variable B set.to 1.
120 FOR I=0 TO 7
130 IF (A&B)#0 PRINT "ON" ; GOTO 150 . . . .
                    . . . . .Performs AND operation of variable A and
                             variable B, and if relevant bit is on, displays
                             "ON".
140 PRING "OFF"
150 B=B*2 . . . . . .Shifts the contents of variable B one bit to the
                             left.
160 NEXT I
170 END
```

7.2 Read/Write of Buffer Memory

The buffer memory can be accessed by both the BASIC program and the sequence program. A maximum of 3K words can be transferred with one instruction.

7.2.1 Read/write with BASIC program

The following system subroutines are used to access the buffer memory. For details refer to the GPC-BASIC Supplementary Handbook.

Read from buffer memory	SR2
Write to buffer memory	SW2
Read/write retry time	SC2

REMARKS

The buffer memory cannot be accessed by the AD51 if the PC CPU is already executing a FROM or TO instruction. In this case the AD51 will retry communication according to its retry time setting. The retry time is set at 10ms unless changed using the SC2 system subroutine.

[Program example]

(1) Program to read 256 words from buffer memory head address 200H to AD51 memory head address F000H.

```
100 A=$E000 . . . . . .Indirect variable head address
110 A(0)=$200 . . . . . .Data source buffer memory head
                         address.
120 A(1)=$F000 . . . . . .Data destination head address
130 A(2)=$100 . . . . . .Number of words
140 B=CALL(0, $8000, 1, A) . . .System subroutine SR2 reads buffer
                               memory.
150 IF B#0 PRINT "ERROR",
    B; GOTO 140 . . . . . .Checks for errors in SR2 execution.
160 END
```

(2) Program to write data from work area addresses E700H to E77FH in the BASIC program to buffer memory addresses 350H to 38FH

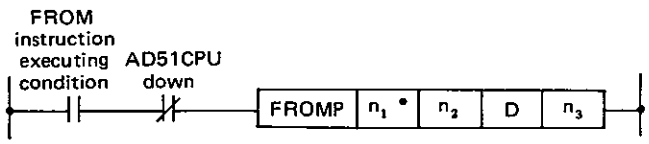
```
100 C=$E100 . . . . . .Indirect variable head address
110 C(0)=$350 . . . . . .Data destination buffer memory head
                         address
120 C(1)=$E700 . . . . . .Data source buffer memory head ad-
                          dress
130 C(2)=$40 . . . . . .Number of data words
140 D=CALL(0, $8003, 1, C) . . .System subroutine SW2 writes data to
                               buffer memory.
150 IF D#0 PRINT "ERROR",
    D; GOTO 140 . . . . . .Checks for errors in SW2 execution.
160 END
```

7.2.2 Read/write with sequence program

Access to the buffer memory from the PC program is via the FROM and TO application instructions.

(1) Read from buffer memory. . . . .
. . . . .FROM, FROMP, DFRO, DFROP instructions

Format



Symbol	Description	Usable Devices
n_1	16th I/O address of the AD51 omitting the least significant digit. *1	K, H
n_2	Buffer memory head address of data source.	K, H
D	Head device number of data destination.	T, C, D, W, R
n_3	Number of words of data to be read	K, H

POINT

***1: Note that the location specification for the AD51 is different from other I/O modules. To specify the slot location use the number representing the 16th I/O point, omitting the "units". (So for example, I/O point 120H is represented as 12H.) Hence, if the AD51 is located at I/O addresses. X60 to X8F and Y60 to Y8F, n_1 is defined as H7.**

[Program example]

Fig. 7.3 shows a program to read the data from buffer addresses 500H to 54FH to data registers D100 to D179 in the PC CPU. When X13 turns with system configuration shown in Fig. 7.2.

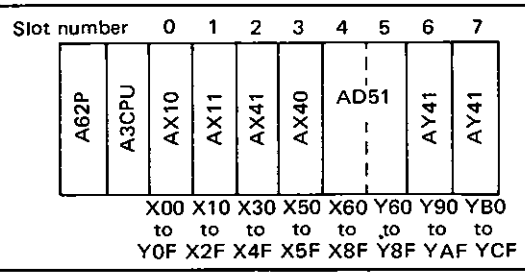


Fig. 7.2 Unit Configuration

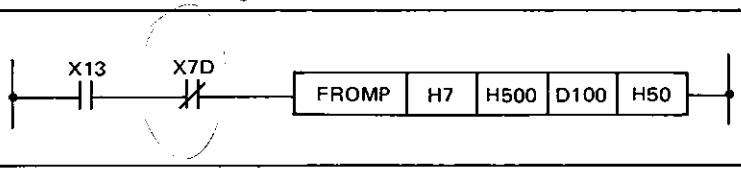
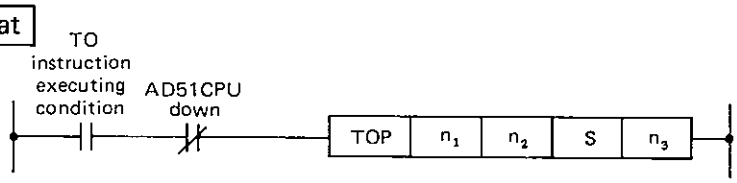


Fig. 7.3 Buffer Memory Read

(2) Write to buffer memory .... TO, TOP, DTO, DTOP instructions

Format



Symbol	Description	Usable Devices
n_1	16th I/O address of the AD51 omitting the least significant digit.	K, H
n_2	Buffer memory head address of data destination.	K, H
S	Head device number of data source	T, C, D, W, R, K, H
n_3	Number of words of data to be written	K, H

[Program example]

Fig. 7.4 shows a program to write data from PC data registers D200 to D263 to buffer addresses 100H to 13FH when X1F turns on with the system configuration shown in Fig. 7.2.

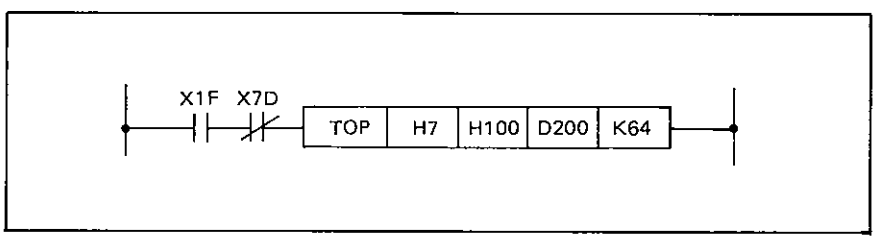


Fig. 7.4 Buffer Memory Write

7.3 Device Memory Read/Write

This section explains how the AD51 communicates with the PC CPU device memory to read present value data and to write new values.

7.3.1 System subroutines and device ranges

The following system subroutines are used by the AD51 to access the PC CPU device memory.

(1) System subroutine types and functions

Item			System Subroutine	Processing	Number of points processed per PC CPU, AD51 transaction.	PC CPU State	
						During STOP	During RUN
Device memory	Batch read	Bit	SADR	Reads data from bit device (such as Y and M) (for 1 point).	256 points	○	○
		Word		Reads data from bit devices (such as Y and M) (for 16 points).	32 words (512 points)		
				Reads data from word device (such as D and R) (for 1 point).	64 points		
	Batch write	Bit	SADW	Write data to bit device (such as Y and M) (for 1 point).	160 points	○	○
		Word		Writes data to bit devices (such as Y and M) (for 16 points).	10 words (160 points)		
				Writes data to word device (such as D and R) (for 1 point).	64 points		
	Test (During random write)	Bit	SADT	Sets/resets any specified bit device (such as Y and M) and device number (for 1 point).	20 points	○	○
		Word		Sets/resets any specified number of blocks of sixteen bit devices (such as Y and M) and device number (for 16 points).	10 words (160 points)		
				Writes data to any specified word device (such as Y and M) and device number (for 1 point).	10 points		
	Monitor data entry	Bit	SADM0	Defines the bit device (such as Y and M) to be monitored (for 1 point).	40 points*	○	○
		Word		Defines the bit devices (such as Y and M) to be monitored (for 16 points).	20 words* (320 points)		
				Defines the word device (such as D and R) to be monitored (for 1 point).	20 points		
	Monitor	Bit	SADM1	Monitors the device specified in monitor data entry.		○	○
		Word					

Key, ○ : indicates available

* Half the number of points shown above can be processed for device X.

(2) Valid device ranges for device memory transactions are given below.

Bit Devices			Word Devices		
Device	Device number range	Represented in: Decimal/ Hexadecimal	Device	Device number range	Represented in: Decimal/ Dexadecimal
Input X	X0000 to X07FF	Hexadecimal	Timer (present value) T	TN000 to TN255	Decimal
Output Y	Y0000 to Y07FF	Hexadecimal	Counter (present value) C	CN000 to CN255	Decimal
Internal relay M	M0000 to M2047	Decimal	Data register D	D0000 to D1023	Decimal
Latch relay L	L0000 to L2047	Decimal	Link register W	W0000 to W03FF	Hexadecimal
Link relay B	B0000 to B03FF	Hexadecimal	File register R	R0000 to R8191	Decimal
Annunciator F	F0000 to F0255	Decimal	Special register D	D9000 to D9255	Decimal
Special relay M	M9000 to M9255				
Timer (Contact) T	TS000 to TS255				
Timer (Coil) T	TC000 to TC255				
Counter (Contact) C	CS000 to CS255				
Counter (Coil) C	CC000 to CC255				

POINT

(1) Bit devices and word devices are differentiated as follows.

Bit device . . .X, Y, M, L, B, F, T (contact), T (coil), C (contact), C (coil)
Word device. .T (present value), C (present value), D, W, R

(2) When reading 16 consecutive bit devices as word data, the bit device number must be a multiple of 16.

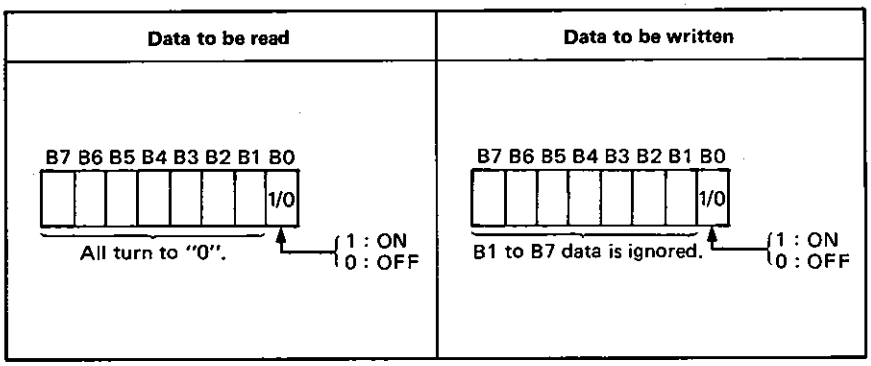
(3) The "0" in the most significant digits of the device number may be specified as spaces (20H).

(4) M and L ranges are specified in the PC CPU parameters however AD51 device memory transactions regard the two as the same.

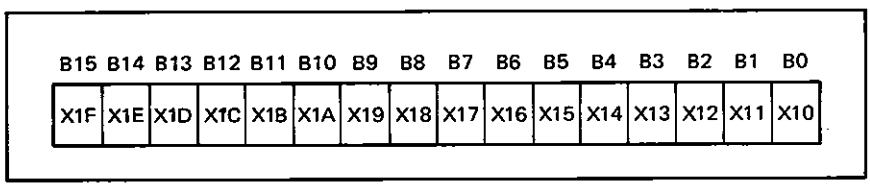
(5) Different device ranges apply when using the AD51 with the A0J2CPU. (See A0J2 Programming Manual)

(3) Bit/word specification

(a) When a system subroutine has been called with "bit" processing specified, one byte of AD51 memory is used per bit. This is illustrated below, the least significant bit indicates the state of the specified bit device for both reading and writing.



(b) When a system subroutine has been called with "word" specified, and bit devices are to be processed (i.e. in batches of 16 devices) then one word of AD51 memory is used per 16 bits. This is illustrated below. The head device number is X10 and state of this device is indicated by the least significant bit. The states of the next 15 bits are stored in consecutive bits from Bit B1 to Bit B15.



7.3.2 BASIC program examples

This section gives some BASIC program examples which use system subroutines SADR (batch read), SADW (batch write), SADM0 (monitor data entry), SADM1 (monitor), and SADT (test). The program examples use channel 1. For details of the system subroutines, refer to the GPC-BASIC Handbooks.

(1) Batch read from device memory (SADR)

[Program example 1]

Program to read data from 16 points, X100 to X10F, to AD51 addresses E000H to E00FH as individual bits.

```
100 A=$F000 . . . . . .Head address for system subroutine
                       INPUT data.
110 A:0)=$FF . . . . . .Defines PC station number as host.
120 A:1)="B" . . . . . .Specifies bit read.
130 B=$F002 . . . . . .Sets indirect variable head address.
140 B$="X0100". . . . . .Sets head device to be read using character
                       string variable.
150 C=$F007 . . . . . .Sets indirect variable head address.
160 C(0)=16 . . . . . .Sets the number of points to be read as
                       16.
170 C(1)=$E000 . . . . . .Sets destination head address for data.
180 C(2)=60 . . . . . .Sets time check period to 60 (600msec).
190 Z=CALL(0, $807B, 1, A). . .Calls system subroutine SADR.
200 IF Z#0 PRINT "ERROR",
    Z; GOTO 190 . . . . . .Checks for errors in SADR execution.
210 END
```

POINT

(1) When specifying the head device using a character string variable, define the number of points to be read after setting the head device. (Reason: The character string variable overwrites the bits at the end of the data with "0". This would delete the "number of points" data if this was written first.)

(2) The time check period in line 180 should be set in accordance with the "number of scans required for processing" given in Section 3.8 (page 3-19) after taking into account any delays which may occur due to other devices accessing the PC CPU.

[Program example2]

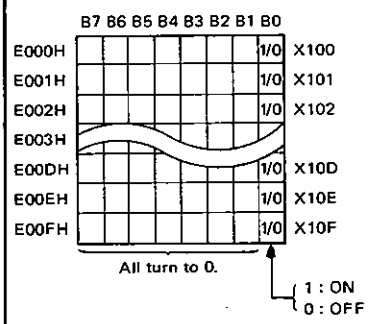
Program to read data from 16 points, X100 to X10F to AD51 addresses E000H to E001H in word units. (i.e. batches of 16 bits)

```
100 A=$F000 . . . . . .Head address for system subroutine
                       INPUT data
110 A:0)=$FF. . . . . .Defines PC station number as host.
120 A:1)="W". . . . . .Specifies word read.
130 B=$F002 . . . . . .Sets indirect variable head address.
140 B$="X0100". . . . . .Sets head device to be read using character
                         string variable.
150 C=$F007 . . . . . .Sets indirect variable head address.
160 C(0)=1 . . . . . .Sets the number of points to be read as 1.
170 C(1)=$E000 . . . . . .Sets destination head address for data.
180 C(2)=60 . . . . . .Sets time check period to 60 (600msec).
190 Z=CALL(0, $807B, 1, A). . .Calls system subroutine SADR.
200 IF Z#0 PRINT "ERROR",
    Z; GOTO 190 . . . . . .Checks for errors in SADR execution.
210 END
```

REMARKS

When "bit read" is specified, one byte of AD51 memory is required per bit of data.
When "word read" is specified, one byte of AD51 memory contains 8 bits of data.
See below:

Bit Units (Example 1)



Word Units (Example 2)

	B7	B6	B5	B4	B3	B2	B1	B0
E000H	X107	X106	X105	X104	X103	X102	X101	X100
E001H	X10F	X10E	X10D	X10C	X10B	X10A	X109	X108

(2) Batch write to device memory (SADW)

[Program example 1]

Program to write on/off data to Y80 to Y9F from the AD51 as individual bits.

Switch on Y80 to Y8F Switch off Y90 to Y9F

```
100 A=$E800 ..... .Head address for system subroutine
                   INPUT data
110 A:0)=$FF ..... .Defines PC station number as host.
120 A:1)="B" ..... .Specifies bit write.
130 B=$E802 ..... .Sets indirect variable head address.
140 B$="Y0080". ..... .Sets destination head device using charac-
                   ter string variable.
150 C=$E807 ..... .Sets indirect variable head address.
160 C(0)=32 ..... .Sets the number of points to be written
                   to 32.
170 C(1)=$E000 ..... .Sets source data head address.
180 C(2)=30 ..... .Sets time check period to 30 (300msec).
190 D=$E000
200 FOR I=0 TO 15
210 D:I)=1          ..... .Generates data to be written to addresses
220 D:(+16)=0               E000H to E01FH.
230 NEXT I
240 Z=CALL(0, $807E, 1, A). . .Calls system subroutine SADW.
250 IF Z#0 PRINT "ERROR",
    Z; GOTO 240 ..... .Check for errors in SADW execution.
260 END
```

[Program example 2]

Program to write on/off data to Y80 to Y9F from the AD51 as individual bits.

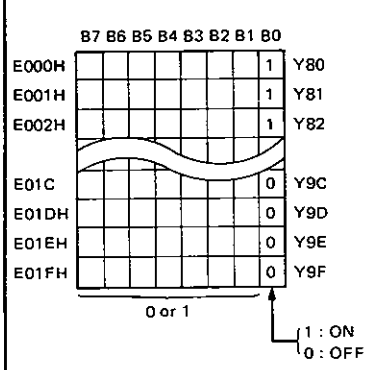
Switch on Y80 to Y8F Switch off Y90 to Y9F

```
100 A=$E800 ..... .Head address for system subroutine
                   INPUT data
110 A:0)=$FF ..... .Defines PC station number as host.
120 A:1)="W" ..... .Specifies word write.
130 B=$E802 ..... .Sets indirect variable head address.
140 B$="Y0080" ..... .Sets destination head device using charac-
                   ter string variable.
150 C=$E807 ..... .Sets indirect variable head address.
160 C(0)=2 ..... .Sets the number of words to be written
                   to 2.
170 C(1)=$E000 ..... .Sets source data head address.
180 C(2)=30 ..... .Sets time check period to 30 (300msec).
190 @($E000)=SFFFF  }
200 @($E002)=0      } .... Generates data to be written to addresses
                           E000H to E003H.
210 Z=CALL(0, $807E, 1, A). . .Calls system subroutine SADW.
220 IF Z#0 PRINT "ERROR",
    Z; GOTO 210 ..... .Checks for errors in SADW execution.
230 END
```

REMARKS

When "bit write", is specified, one byte of AD51 memory is required per bit of data. When "word write" is specified, one byte of AD51 memory contains 8 bits of data.
See below:

Bit Units (Example 1)



*Only B0 is valid and B1 toB7 are ignored.

Word Units (Example 2)

	B7	B6	B5	B4	B3	B2	B1	B0
E000H	Y87	Y86	Y85	Y84	Y83	Y82	Y81	Y80
E001H	Y8F	Y8E	Y8D	Y8C	Y8B	Y8A	Y89	Y88
E002H	Y97	Y96	Y95	Y94	Y93	Y92	Y91	Y90
E003H	Y9F	Y9E	Y9D	Y9C	Y9B	Y9A	Y99	Y98

(3) Data write to any PC device data memory (SADT)

[Program example 1]

Program to write on/off data to the following random bit devices:

Switch on Y115, switch off M340. Switch off B24C, switch on L875.

```
100 A=$E000 . . . . . .Head address for system subroutine
                       INPUT data
110 FOR I=0 TO 3
120 A(I)=$E100+I*6     . . . .Sets character string variable head address
130 NEXT I                    for destination device.
140 C=$E100 . . . . . .Sets indirect variable head address.
150 A$(0)="Y0115". . . . . .Sets device name "Y0115".
160 C:5)=1 . . . . . .Source data for Y115.
170 A$(1)="M0340" . . . . . .Sets device name "M0340".
180 C:11)=0 . . . . . .Source data for M340.
190 A$(2)="B024C". . . . . .Sets device name "B024C".
200 C:17)=0 . . . . . .Source data for B24C.
210 A$(3)="L0875". . . . . .Sets device name "L0875".
220 C:23)=1 . . . . . .Source data for L875.
230 B=$E200 . . . . . .Sets indirect variable head address.
240 B:0)=$FF . . . . . .Defines PC station number as host.
250 B:1)="B" . . . . . .Specifies "bit write".
260 B(1)=4 . . . . . .Specifies 4 pieces of data to be written.
270 B(2)=$E100 . . . . . .Sets source data head address.
280 B(3)=70 . . . . . .Sets time check period to 70(700msec).
290 Z=CALL(0, $8081, 1, B) . . .Calls system subroutine SADT.
300 IF Z#0 PRINT "ERROR",
    Z; GOTO 290 . . . . . .Checks for errors in SADT execution.
310 END
```

[Program example 2]

Program to write numerical data to the following random word devices.

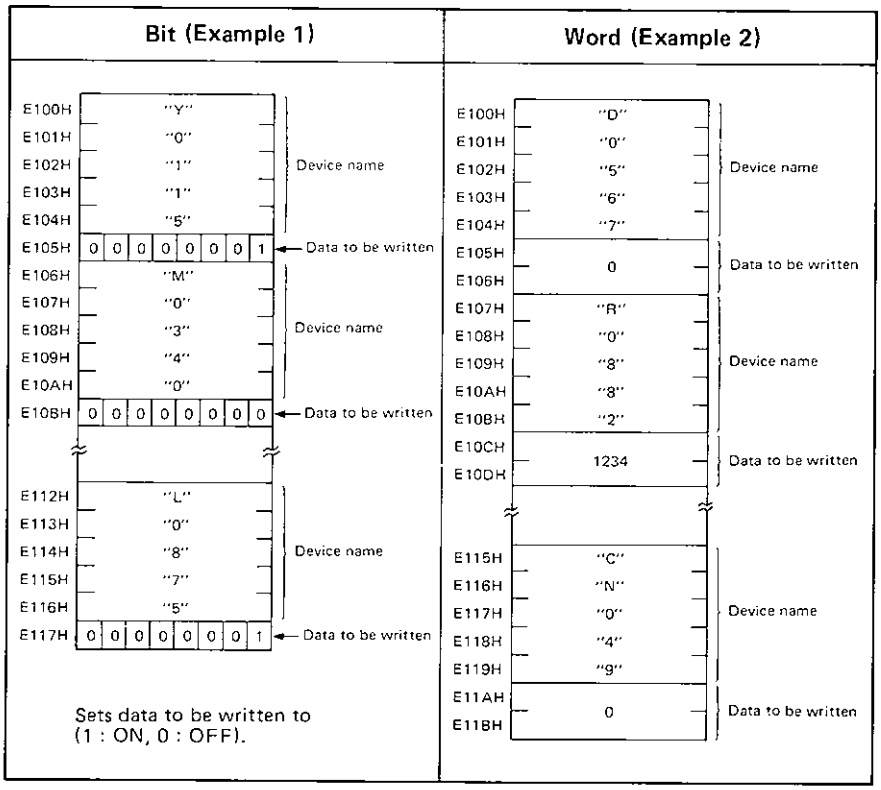
```
· D567 ← 0   , R882 ← 1234
  W187 ← 751, C49  ← 0
```

```
100 A=$E000 . . . . . .Sets indirect variable head address.
110 FOR I=0 TO 3
120 A(I)=$E100+IX7  . . . . . .Character string variable head address for
130 NEXT I                     destination device.
140 C=$E100 . . . . . .Sets indirect variable head address.
150 A$(0)="D0567". . . . . .Sets device name "D0567".
160 @(C+5)=0 . . . . . .Sets source data for D567.
170 A$(1)="R0882". . . . . .Sets device name "R0882".
180 @(C+12)=1234. . . . . .Sets source data for R882.
190 A$(2)="W0187" . . . . . .Sets device name "W0187".
200 @(C+19)=751 . . . . . .Sets source data for W187.
210 A$(3)="CN049" . . . . . .Sets device name "CN049".
220 @(C+26)=0 . . . . . .Sets source data for CN49.
230 B=$E200 . . . . . .Sets indirect variable head address.
240 B:0)=$FF . . . . . .Defines PC station number as host.
250 B:1)="W". . . . . .Specifies "word write".
260 B(1)=4 . . . . . .Specifies 4 pieces of data to be written.
270 B(2)=$E100 . . . . . .Sets source data head address.
280 B(3)=70 . . . . . .Sets time check period to 70(700msec).
290 Z=CALL(0, $8081, 1, B) . . .Calls system subroutine SADT.
300 IF Z#0 PRINT "ERROR",
    Z; GOTO 290 . . . . . .Checks for errors in SADT execution.
310 END
```


IB (NA) 66059-A

REMARKS

The data is stored in the AD51 in the following formats depending on whether "bit" or "word" has been specified.



(4) Defining device numbers to be monitored. (Before a given device can be monitored using the SADM1 subroutine, it must be specified and "entered" using the following procedure.)

[Program example 1]

Program to specify the following bit devices for monitoring (i.e. monitor data entry)

X7D , Y201 , M178 B3A0, T46 contact, C85 coil

```
100 A=$F000 . . . . . .Sets indirect variable head address.
110 A$="X007DY0201M0178B03A0TS046CC085" . . .
                        . . . .Stores device numbers into work area.
120 B=$E800 . . . . . .Sets indirect variable head address.
130 B:0)=$FF . . . . . .Defines PC station number as host.
140 B:1)="B" . . . . . .Specifies "bit entry".
150 C=$E802 . . . . . .Sets indirect variable head address.
160 C(0)=6 . . . . . .Sets the number of points to be entered
                        to 6.
170 C(1)=A. . . . . .Sets "entry" data head address.
180 C(2)=50 . . . . . .Sets time check period to 50 (500msec).
190 Z=CALL(0, $8084, 1, B) . . .Calls system subroutine SADM0.
200 IF Z#0 PRINT "ERROR",
    Z; GOTO 190 . . . . . .Checks for errors in SADM0 execution.
210 END
```

[Program example 2]

Program to specify the following word devices for monitoring (or bit devices entered as 16 consecutive device numbers).

X60 to X6F, T80 to T95 contacts, C183 present value D260 , W175 , R700

```
100 A=$F000 . . . . . .Sets indirect variable head address.
110 A$="X0600TS080CN183D0260W0175R0700". . .
                        . . . .Stores device numbers into work area.
120 B=$E800 . . . . . .Sets indirect variable head address.
130 B:0)=$FF . . . . . .Defines PC station number as host.
140 B:1)="W". . . . . .Specifies "word entry".
150 C=$E802 . . . . . .Sets indirect variable head address.
160 C(0)=6 . . . . . .Sets the number of points to be entered
                        to 6.
170 C(1)=A. . . . . .Sets "entry" data head address.
180 C(2)=70 . . . . . .Sets time check period to 70 (700msec).
190 Z=CALL(0, $8084, 1, B) . . .Calls system subroutine SADM0.
200 IF Z#0 PRINT "ERROR",
    Z; GOTO 190 . . . . . .Checks for errors in SADM0 execution.
210 END
```

POINT

The SADM0 system subroutine (Monitor data entry) is used for both bit and word device and, once entered, is valid for all tasks. These system subroutine enter the devices specified into the OS area where they remain valid until new ones are entered.

(5) Monitoring of devices (SADM1) specified by the monitor data entry (SADM0)

[Program example 1]

Program to specify and monitor the following bit devices:

X7D , Y201 , M178 B3A0, T46 contact, C85 coil

```
100 A=$F000 . . . . . .Sets indirect variable head address.
110 A$="X007DY0201M0178B03A0TS046CC085". . .
                        . . . .Stores device numbers into work area.
120 B=$E800 . . . . . .Sets indirect variable head address.
130 B:0)=$FF . . . . . .Defines PC station number as host.
140 B:1)="B" . . . . . .Specifies "bit entry".
150 C=$E802 . . . . . .Sets indirect variable head address.
160 C(0)=6 . . . . . .Sets the number of points to be entered
                        to 6.
170 C(1)=A. . . . . .Sets "entry data" head address.
180 C(2)=50 . . . . . .Sets time check period to 50 (500msec).
190 Z=CALL(0, $8084, 1, B) . . .Calls system subroutine SADM0.
200 IF Z#0 PRINT "ERROR",
    Z; GOTO 190 . . . . . .Checks for errors in SADM0 execution.
210 D=$E900 . . . . . .Sets indirect variable head address.
220 D:0)=$FF . . . . . .Defines PC station number as host.
230 D:1)="B" . . . . . .Specifies "bit monitor".
230 D(1)=$EA00 . . . . . .Data destination head address.
240 D(2)=50 . . . . . .Sets time check period to 50 (500msec).
240 Z=CALL(0, $8087, 1, D) . . .Calls system subroutine SADM1.
250 IF Z#0 PRINT "ERROR",
    Z; GOTO 260 . . . . . .Checks for errors in SADM1 execution.
260 END
```


IB (NA) 66059-A

[Program example 2]

Program to specify and monitor the following bit devices (and bit devices entered as 16 consecutive device numbers).

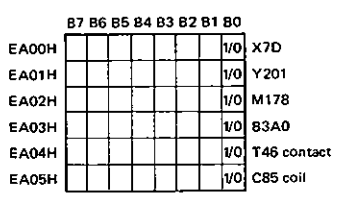
X60 to X6F, T80 to T95 contacts, C183 present value D260 , W175 , R700

```
100 A=$F000 . . . . . .Sets indirect variable head address.
110 A$="X0060TS080CN183D0260W0175R0700". . .
                       . . . .Stores device numbers into work area.
120 B=$E800 . . . . . .Sets indirect variable head address.
130 B:0)=$FF . . . . . .Defines PC station number as host.
140 B:1)="W". . . . . .Specifies "word entry".
150 C=$E802 . . . . . .Sets indirect variable head address.
160 C(0)=6 . . . . . .Sets the number of points to be entered
                       to 6.
170 C(1)=A. . . . . .Sets "entry" data head address.
180 C(2)=50 . . . . . .Sets time check period to 50 (500msec).
190 Z=CALL(0, $8084, 1, B) . . .Calls system subroutine SADM0.
200 IF Z#0 PRINT "ERROR",
    Z; GOTO 190 . . . . . .Checks for errors in SADM0 execution.
210 D=$E900 . . . . . .Sets indirect variable head address.
220 D:0)=$FF . . . . . .Defines PC station number as host.
230 D:1)="W". . . . . .Specifies "word monitor".
240 D(1)=$EA00 . . . . . .Data destination head address.
250 D(2)=50 . . . . . .Sets time check period to 50 (500msec).
260 Z=CALL(0, $8087, 1, D) . . .Calls system subroutine SADM1.
270 IF Z#0 PRINT "ERROR",
    Z; GOTO 260 . . . . . .Checks for errors in SADM1 execution.
280 END
```

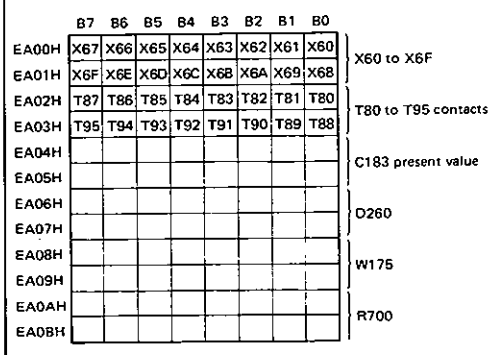
REMARKS

The data is stored in the AD51 in the following formats depending on whether "bit" or "word" has been specified.

Bit (Example 1)



Word (Example 2)



7.4 Interrupt from PC CPU to AD51

The interrupt signal from the PC CPU to the AD51 is valid on its rise (i.e. the AD51 waits for the leading edge of the signal).

[How to use]

To enable the interrupt facility, the relevant task should be set to "start at interrupt from PC CPU" on the multi task setting screen. The AD51 will then run that task when it receives the interrupt signal. Once it has been started the interrupt task will operate until the END instruction is executed. The interrupt is re-enabled after the END instruction is executed.

[Program example]

The following P.C. program will call the designated interrupt task when X01 turns on.

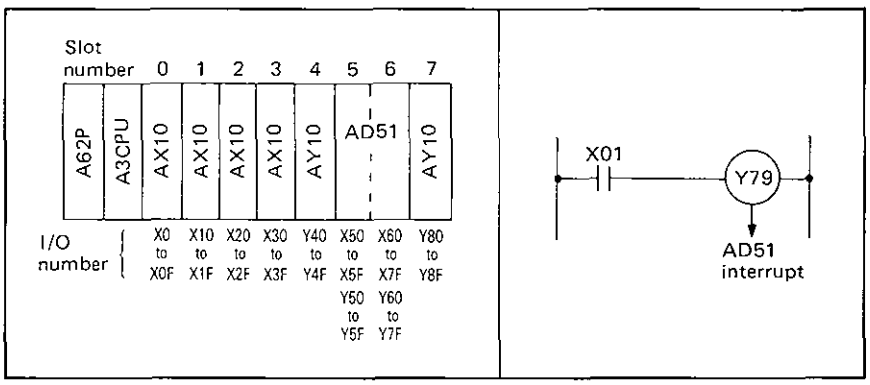


Fig. 7.5 Unit Configuration and Sequence Program

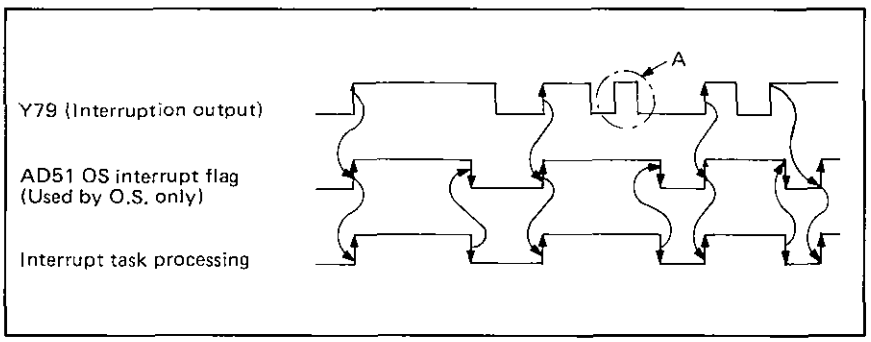


Fig. 7.6 Timing Chart

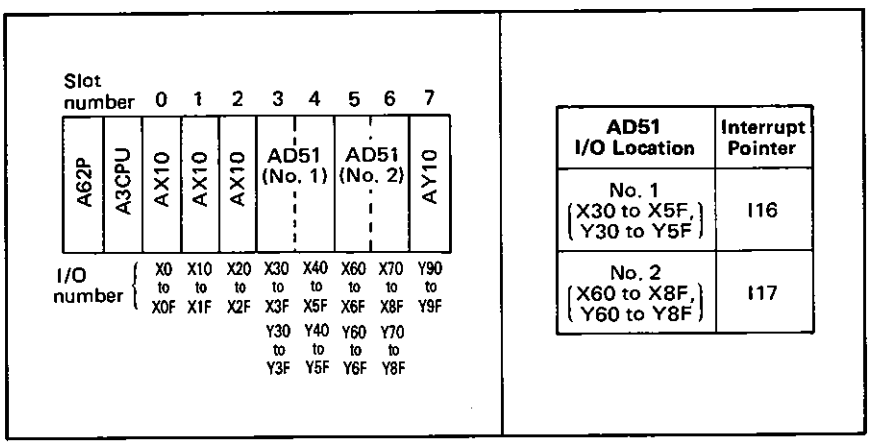
POINT

(1) Any interrupt signal given to the AD51 while an interrupt program is running (i.e. interrupt flag is ON) will be ignored (See area A in Fig. 7.6.)

(2) Only one task may be specified as "interrupt start". Setting any more generally leads to ORST error.

7.5 Interrupt from AD51 to PC CPU

System subroutine "SIT" causes the AD51E to interrupt the A1, A2, or A3CPU and allows AD51E interrupt sequence programs to be executed. The A0J2CPU cannot be interrupted by the AD51E. The A1, A2, or A3CPU has interrupt pointers, I16 to I23, which are assigned to interrupt signals generated by special function units in order of I/O allocation. For details, refer to the CPU Unit User's Manual and Programming Manual.
In the system configuration shown in Fig. 7.7, when AD51 No. 1 interrupts the PC CPU, the interrupt program designated by pointer I16 is executed. When AD51 No. 2 interrupts the PC CPU, interrupt program I17 is executed. (I16 has higher priority.)



AD51 I/O Location	Interrupt Pointer
No. 1 (X30 to X5F, Y30 to Y5F)	I16
No. 2 (X60 to X8F, Y60 to Y8F)	I17

Fig. 7.7 System Configuration and Interrupt Pointers

7.6 Read/Write of Sequence Program and T/C Set Values

This section describes procedures for reading and writing PC CPU sequence programs and T/C set values from the AD51.

7.6.1 System subroutines and functions

The following system subroutines are used to read and write sequence programs and T/C set values.

(1) System subroutines and functions

Item			System Subroutine	Processing	Amount of information processed per PC – AD51 transaction.	PC CPU State	
						Stop	Run
Sequence program	Read	Main	SAAR	Reads main sequence program.	64 steps	○	○
		Sub		Reads subsequence program.			
	Write	Main	SAAW	Writes main sequence program.		○	×•
		Sub		Writes subsequence program.			
Parameter	Read		SAPR	Reads parameters of PC CPU.	128 bytes	○	○
	Write		SAPW	Writes parameters of PC CPU.		○	×
	Analysis request		SAPS	Causes PC CPU to recognize and check rewritten parameters.		○	×

Key : ○ Available
× Unavailable
• For T/C only

Table 7.2 System Subroutines and Functions

* A subprogram may be written while a main program is running and vice versa using appropriate control of M9050 and M9051.

POINT

(1) Sequence programs should be read and written in the range set in the parameters. PC CPU data may be corrupted if programs are written outside the set range.

(2) An input data error is returned if the specified parameter read/write capacity is outside the allowed range (16 bytes in the A0J2CPU, 3K bytes in the A1, A2, and A3CPUs).

IB (NA) 66059-A

(2) Read head step and write head step

Specify T/C set values and program step numbers as shown in the following table.

Sequence Program	Step
T0 set value T1 set value to T255 set value	FE00H FE01H to FEFFH
C0 set value C1 set value to C255 set value	FF00H FF01H to FFFFH
Step 0 Step 1 to Step 30719 (30K)	0000H 0001H to 77FFH

Calculation of specified step:

Timer Tm = FE00H + n
Counter Cm = FF00H + n

where, m = device number
n = hexadecimal value of device number

(3) T/C set value

Read/write data of T/C set values is expressed in hexadecimal as shown in the following table.

Specification with Constant	Set Value	Specification with D Register	Set Value
K0 K1 K2 to K32766 K32767	0000H 0001H 0002H to 7FFEH 7FFFH	D0 D1 D2 to D1022 D1023	8000H 8002H 8004H to 81FCH 81FEH

Calculation of set value:

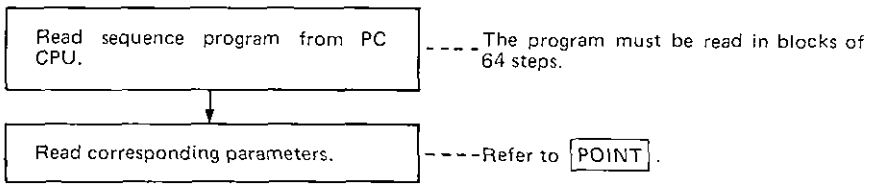
Km = 000H + n
Dm = 8000H + 2n

where, m = device number
n = hexadecimal value of device number

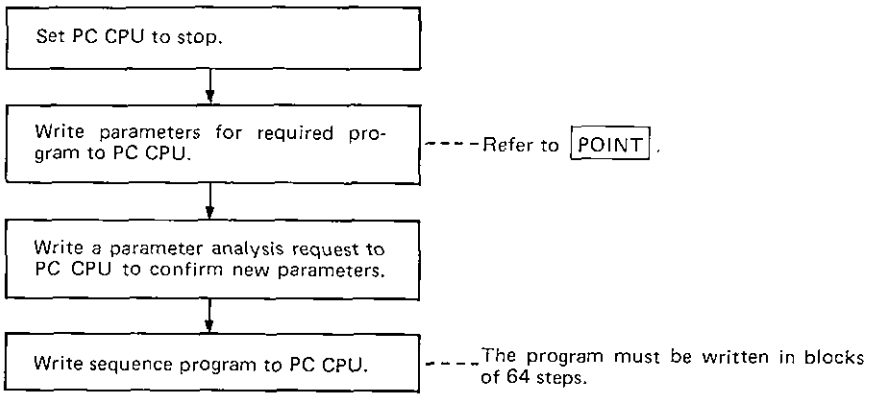
7.6.2 Read and write procedures

Parameter data should not be separated from the sequence program when it is read or written by the AD51. Use the following procedures:

(1) Sequence program read



(2) Write of sequence programs



POINT

When reading or writing sequence programs, always, ensure that the parameters match the program. Mismatches of parameters and programs will result in misoperation.

IB (NA) 66059-A

7.6.3 BASIC program example

This section gives the procedure for transferring sequence programs and T/C set values between the PC CPU and the AD51.

(1) Sequence program read (SAAR, SAPR)

[Program example]

Program to read the PC CPU main program to the following AD51 memory areas.

Sequence program (2K steps) → Channel 2 E000H to EFFFH Parameter (3K bytes) → Channel 2 D000H to DBFFH

```
100 Z=ZWR1(2, $F800, $FF) . . . . . .Defines PC station number as host.
110 Z=ZWR1(2, $F801, $4D) . . . . . .Sets main, "M", sequence program.
120 Z=ZWR2(2, $F804, 64) . . . . . .Sets the number of steps to be read to 64.
130 Z=ZWR2(2, $F808, 60) . . . . . .Sets time check period to 60 (600m sec).
140 A=$E000. . . . . .Sets data destination head address to
                      variable A.
150 FOR I=O TO 2047 STEP 64
160 Z=ZWR2(2, $F802, I) . . . . . .Sets head step number of sequence pro-
                                   gram to be read.
170 Z=ZWR2(2, $F806, A). . . . . .Sets data destination head address.
180 Z=CALL(O, $808A, 2, $F800) . .Reads sequence program.
185 IF Z#0 PRINT "ERROR",
    Z; GOTO 180. . . . . .Checks for errors in SAAR execution.
190 A=A+128 . . . . . .Adds 128 (64 steps) to head address of
                       area to be read.
200 NEXT I
210 Z=ZWR1(2, $F810, $FF) . . . . . .Defines PC station number as host.
220 Z=ZWR2(2, $F814, 128) . . . . . .Sets read byte length to 128.
230 Z=ZWR2(2, $F818, 60) . . . . . .Sets time check period to 60 (600msec).
240 FOR I=O TO $BFF STEP $80
250 Z=ZWR2 (2, $F811, I)  } . . . . .Sets parameter read head address.
260 Z=ZWR1(2, $F813, O)   }
270 Z=ZWR2(2, $F816, $D000+I). . .Sets data destination head address.
280 Z=CALL(O, $8090, 2, $F810). . .Reads parameters.
285 IF Z#0 PRINT "ERROR",
    Z; GOTO 280. . . . . .Checks for errors in SAPR execution.
290 NEXT I
300 END
```

(2) Sequence program write

[Program example]

Program to write the main sequence program and parameters from the AD51 to the PC CPU.

Channel 2 E000H to EFFFH → Sequence program (2K steps) Channel 2 D000H to DBFFH → Parameters (3K bytes)

```
100 Z=CALL(0, $8030, $FF, 70). . . .Checks PC CPU RUN/STOP status.
110 IF Z=1 LOCATE 5, 10; PRINT "CPU RUN"; GOTO 100. . .
                                .....Indicates whether the CPU is running.
120 IF Z#0 GOTO  To error processing . . .
                                .....Detects any error in SKC execution and
                                     moves to a suitable part of the program.
130 Z=ZWR1(2, $F000, $FF). . . . .Defines the PC station number as host.
140 Z=ZWR2(2, $F004, 128) . . . . .Sets byte length to be written to 128.
150 Z=ZWR2(2, $F008, 70) . . . . . .Sets time check period to 70 (700msec).
160  FOR I=O TO $BFF STEP $80
170 Z=ZWR2(2, $F001, I)  }
180 Z=ZWR1(2, $F003, O)  } . . . . .Sets parameter write head address.
190 Z=ZWR2(2, $F006, $D000+I). . .Sets source data head address.
200 Z=CALL(0, $8093, 2, $F000) . .Writes parameters.
210 IF Z#0 PRINT "ERROR",
    Z; GOTO 200. . . . . .Checks for errors in SAPW execution.
220 NEXT I
230 Z=CALL(0, $8096, $FF, 70). . . .Parameter analysis request.
240 IF Z#0 PRINT "ERROR",
    Z; GOTO 225. . . . . .Checks for errors in SAPS execution.
250 Z=ZWR1(2, $F010, $FF). . . . .Defines PC station number as host.
260 Z=ZWR1(2, $F011, $4D). . . . .Sets main, "M", sequence program.
270 Z=ZWR2(2, $F014, 64) . . . . . .Sets the number of steps to be written
                                     to 64.
280 Z=ZWR2(2, $F018, 70) . . . . . .Sets time check period to 70 (700msec).
290 A=$E000. . . . . .Sets data source head address to variable
                                     A.
300 FOR I=O TO 2047 STEP 64
310 Z=ZWR2(2, $F012, I) . . . . . .Sets head step number of sequence pro-
                                     gram to be written.
320 Z=ZWR2(2, $E014, A). . . . . .Sets data source head address.
330 Z=CALL(0, $F016, 2, $F010). . .Writes sequence program.
340 IF Z#0 PRINT "ERROR",
    Z; GOTO 300. . . . . .Checks for errors in SAAW execution.
350 A=A+128 . . . . . .Adds 128 to sequence program write
                                     head address.
360 NEXT I
370 END
```

(3) T/C set values read.

[Program example]

Program for reading set values of T0 to T31 and C64 to C79 to memory addresses B000H to B03FH and B040H to B05FH in channel 2.

```
100 Z=ZWR1(2, $E700, $FF) . . . . .Defines PC station number as host.
110 Z=ZWR1(2, $E701, $4D) . . . . .Sets main ,"M", sequence program.
120 Z=ZWR2(2, $E702, $FE00) . . . .Sets read head step to T0.
130 Z=ZWR2(2, $E704, 32) . . . . . .Sets the number of points to be read to
                                    32.
140 Z=ZWR2(2, $E706, $B000) . . . .Sets data destination head address.
150 Z=ZWR2(2, $E708, 50) . . . . . .Sets time check period to 50 (500msec).
160 Z=CALL(0, $808A, 2, $E700). . .Reads T0 to T31 set values.
170 Z=ZWR2(2, $E702, $FF40) . . . .Sets read head step to C64.
180 Z=ZWR2(2, $E704, 16) . . . . . .Sets the number of points to be read to
                                    16.
190 Z=ZWR2(2, $E706, $B040) . . . .Sets data destination head address.
200 Z=CALL(0, $808A, 2, $E700). . .Reads C64 to C79 set values.
210 IF Z#0 PRINT "ERROR",
    Z; GOTO 200. . . . . .Checks for errors in SAAR execution.
220 END
```

7.7 Remote RUN/STOP of PC CPU

The PC CPU can be switched between RUN and STOP by the AD51 using the following system subroutines.

Item		System Subroutine	Processing
PC CPU	Remote RUN	SKR	Requests remote RUN of PC CPU.
	Remote STOP	SKP	Requests remote STOP of PC CPU.

(1) Precautions for remote RUN/STOP

1) Note that a "remote error" is flagged if a remote RUN (or STOP) signal is given to a PC CPU which has already received a remote STOP (or RUN) signal from a separate unit, e.g. AJ71C24.

2) Remote RUN/STOP commands from the AD51 are valid as follows for different CPU key switch positions:

		PC CPU Key Switch Position			
		RUN	STOP	PAUSE	STEP-RUN
Command from AD51	Remote RUN	RUN	STOP	PAUSE	STEP-RUN
	Remote STOP	STOP	STOP	STOP	STOP

3) The clearing of data memories on receiving a remote run instruction depends on the states of special relays M9016 and M9017.

Special Relay		Data Memory State
M9016	M9017	
OFF	OFF	CPU is run without clearing data memory.
OFF	ON	Data memory is cleared outside the latch range set in parameters. (Link image is not cleared.)
ON	ON/OFF	CPU is run after data memory is cleared.

REMARKS

Always reset special relays M9016 and M9017 where data memory clearing is not required.

4) Resetting the PC CPU during remote RUN/STOP control (either with the keyswitch or by powering down and up) causes the remote signal to be removed and the PC CPU to revert to the mode detected by its key switch.

(2) BASIC program example

[Program example]

Program for remote RUN/STOP of PC CPU through key inputs

```
100 LOCATE 20, 10 . . . . . .Specifies cursor position.
110 PRINT"STOP PC CPU? Y/N" . . . . . .Displays message.
120 A=INKEY . . . . . .Waits for key input.
130 IF A#"Y" GOTO 120 . . . . . .Checks key input characters.
140 Z=CALL(0, $8036, $FF, 60). . . . . .Executes remote STOP.
145 IF Z#0 PRINT "ERROR", Z; GOTO 140 . .Checks for errors in SKP execution.
150 LOCATE 20, 10 . . . . . .Specifies cursor position.
160 PRINT"REMOTE-RUN PC CPU? Y/N" . . .Displays message.
170 A=INKEY . . . . . .Waits for key input.
180 IF A#$"Y" GOTO 170 . . . . . .Checks key input characters.
190 Z=CALL(0, $8033, $FF, 60). . . . . .Executes remote RUN.
200 IF Z#0 PRINT"ERROR", Z; GOTO 190. . .Checks for errors in SKR execution.
210 GOTO 100
```

8. TROUBLESHOOTING

This section lists error messages and troubleshooting procedures.

8.1 Screen Error Messages

The following messages may be generated during operation of the AD51 with its programming console.

Error Message	Display Screen	Description	Corrective Action
CANNOT SET	Mode select menu	1) Invalid number has been set. 2) "1" (MULTI TASK GO) has been pressed before multitask setting, or there is an error in the multitask data.	1) Correct the number. 2) Set or correct multitask data.
	BASIC program address setting	Invalid number has been set.	Correct the number.
	Date and time setting	Invalid value has been set.	Correct.
	GPP mode	1) GPP mode selected without connecting the GPP to CH1. 2) Invalid number has been set.	Correct.
MEMORY PROTECT ERROR	Mode select menu	System data area is memory protected.	Set the memory protect switch to OFF.
DATA [n] SET ERROR	BASIC program address setting	The data indicated on the menu by the number n is wrong.	Correct.
	Printer setting		Correct.
ERROR	Multitask setting	Value above "ERROR" is wrong.	For ERROR displayed in the TYPE, START CONDITION, or INTERVAL, columns, correct the data on the screen. For other columns correct the data on the BASIC program address setting screen.
AD51 BUS ERROR	GPP mode	GPP inaccessible to AD51 buffer.	Usually caused by the PC CPU accessing the AD51 buffer memory too frequently or with too much data. STOP the PC CPU.
AD51 COMMUNICATION ERROR		Communication error between AD51E and GPP	Check cable connection and start up again.
AD51 WRITE ERROR		Memory area is ROM or memory protected.	Select RAM area channel or reset memory protect.
ADDRESS ERROR		Address is not in the allowed range.	Correct.
CANNOT USE KANA!!		"Kana" (i.e. Japanese characters) in the system name.	Select appropriate character set and use alphanumerics for system name specification.
DISK FULL		FD capacity exceeded.	Insert new FD.
FILE NAME ERROR		Invalid file name for file directory or delete function.	Correct.
FLOPPY ERROR		1) No FD in accessed drive. 2) FD is write protected. 3) FD is defective.	1) Insert FD. 2) Set FD write protect tab to OK. 3) Change FD.
FLOPPY WRITE PROTECT		FD is write protected.	Set FD write protect tab to OK.
IDENTICAL NAME		The same file name exists.	Change the file name.
MEMORY NOTHING		Invalid area number has been specified.	Correct the area number.
NO FILE		Specified file is not on FD.	Correct.
ROM ERASING ERROR		ROM has not been erased.	Erase ROM data or use a new ROM.

IB (NA) 66059-A

Error Message	Display Screen	Description	Corrective Action
ROM WRITE ERROR	GPP mode	1) ROM is wrongly or not loaded. 2) ROM is defective.	1) Check ROM. 2) Write several times. Try again, if data cannot be written, change ROM.
SIZE UNMATCH (ROM<FILE)		ROM capacity is smaller than file capacity.	Select appropriate ROM.
SYSTEM NAME ERROR		Invalid name has been specified. (The name includes non-alpha-numeric character or blank or the first character is not a letter.)	Correct.
VERIFY ERROR		Data unmatched.	Correct.
STACK ERROR! AD51 STOP!	Multitask execution	Stack has been used outside the set area.	In BASIC, a maximum of ten levels of GOSUB or FOR/NEXT instructions are allowed.
BTWF ERROR! AD51 STOP!		Task scheduling RAM data has been changed.	Check whether the system memory has been accessed by the user program.
WAIT ERROR! AD51 STOP!			
AD51 STOP! TASK NO.		There is a BASIC statement which cannot be translated by the interpreter.	Correct BASIC program.
STOP COMMAND AD51 STOP! TASK NO.		STOP command executed.	Remove STOP command or change to END, GOTO, GOSUB, RETURN, ONGOTO or ONGOSUB command.
BREAK COMMAND AD51 STOP! TASK NO.		BREAK command executed.	Remove BREAK command.
TEXT END AD51 STOP! TASK NO.		BASIC program does not finish with END, GOTO, GOSUB, ONGOTO, ONGOSUB or RETURN command.	Correct.
WHAT?	BASIC programming mode	BASIC programming mode error detected in BASIC program. *1	Correct.
HOW?			
SORRY		Program area insufficient.	Expand.
ROM OR MEMORY PROTECT AREA! PLEASE DO NOT CORRECT PROGRAM		Program area is ROM or memory protected.	Alarm message *2

POINT

***1: "WHAT" and "HOW" are indicated when:**

1) **An undefined command is used;**
2) **A command description format is wrong;**
3) **A line number is not specified on the left of the GOTO, GOSUB, ONGOTO, or ONGOSUB command; and**
4) **The RETURN command is used without the GOSUB or ONGOSUB command.**

***2: When this message is indicated, never correct the program. Correction will corrupt the BASIC program memory area data. With this message displayed, only LIST, LLIST and BYE commands should be used.**
To allow correction of a protected program, switch the memory protect off.

8.2 Error Code List

The occurance of any error during AD51 operation will cause the appropriate error code to be displayed on the two digit annunciator. Code definitions are as follows:

Error Number	Error		Description	Location	Corrective Action
00	Battery error		Battery is not loaded. Battery voltage low.	--	Load battery. Change battery.
10	Multi task setting error		Although multi task setting is wrong, multi task has been executed.	--	Re-set multi task.
11	BASIC program error		Grammatical error in BASIC program.	Task 1	Correct program.
12				Task 2	
13				Task 3	
14				Task 4	
15				Task 5	
16				Task 6	
17				Task 7	
18				Task 8	
21	STOP error	BASIC	STOP command has been executed during multi task execution.	Task 1	Remote STOP command or change to END, GOTO, GOSUB, ONGOTO, ONGOSUB, or RETURN command.
22				Task 2	
23				Task 3	
24				Task 4	
25				Task 5	
26				Task 6	
27				Task 7	
28				Task 8	
31	BREAK error	BASIC	BREAK command has been executed during multi task execution.	Task 1	Remove BREAK command.
32				Task 2	
33				Task 3	
34				Task 4	
35				Task 5	
36				Task 6	
37				Task 7	
38				Task 8	
41	Text end error		BASIC program does not end with END, GOTO, GOSUB, ONGOTO, ONGOSUB, or RETURN command.	Task 1	Correct program.
42				Task 2	
43				Task 3	
44				Task 4	
45				Task 5	
46				Task 6	
47				Task 7	
48				Task 8	

Error Number	Error	Description	Location	Corrective Action
51	ORST error	A task has been re-started before it has completed.	Task 1	Correct task start condition.
52			Task 2	
53			Task 3	
54			Task 4	
55			Task 5	
56			Task 6	
57			Task 7	
58			Task 8	
60	Stack error	Stack used is outside the system stack area.	—	In BASIC a maximum of ten levels of GOSUB or FOR/NEXT instructions are allowed.
70	Duplex WAIT error BTWF error	RAM contents for system's task schedule have been rewritten.	—	Check whether the system memory has been accessed by the user program.
81	Receive buffer full error	511 bytes of received data in receive buffer.	RS-422 CH1	Do not send more than 512 bytes at one time.
82			RS-422 CH2	
83			RS-232C CH3	
84			RS-232C CH4	
91	Send buffer full error	127 bytes of send data in send buffer.	RS-422 CH1	Check cables. Empty the external equipment reveive buffer.
92			RS-422 CH2	
93			RS-232C CH3	
94			RS-232C CH4	
99	PC CPU error	1) PC CPU has been reset during communication. 2) Time out error has occurred during PC accessing by system subroutine. 3) PC CPU error detected by WDT and communication has stopped. *Note: Error code 99 is sometimes displayed after an instantaneous power failure.*	—	AD51 program execution not directly affected.

POINT

(1) The AD51 continues operating in the event of "battery error", "ORST error", "receive buffer full error", "send buffer full error", or "PC CPU error".

(2) The receive buffer capacity is 511 bytes per channel. Overflow data is ignored.

(3) When data is sent from one task to the buffer memory and the buffer memory becomes full, the AD51 switches tasks. After that task has run the AD51 checks the buffer, if it is now vacant the original task is allowed to continue and if it is still full the AD51 will switch to a third task (where used). This checking and switching procedure will continue for 1 minute if the buffer remains full after which 1 byte of the excess data will be deleted. The one minute cycle is repeated until there is no excess data left (The 1 minute cycle time may be charged using the SWB system subroutine.)

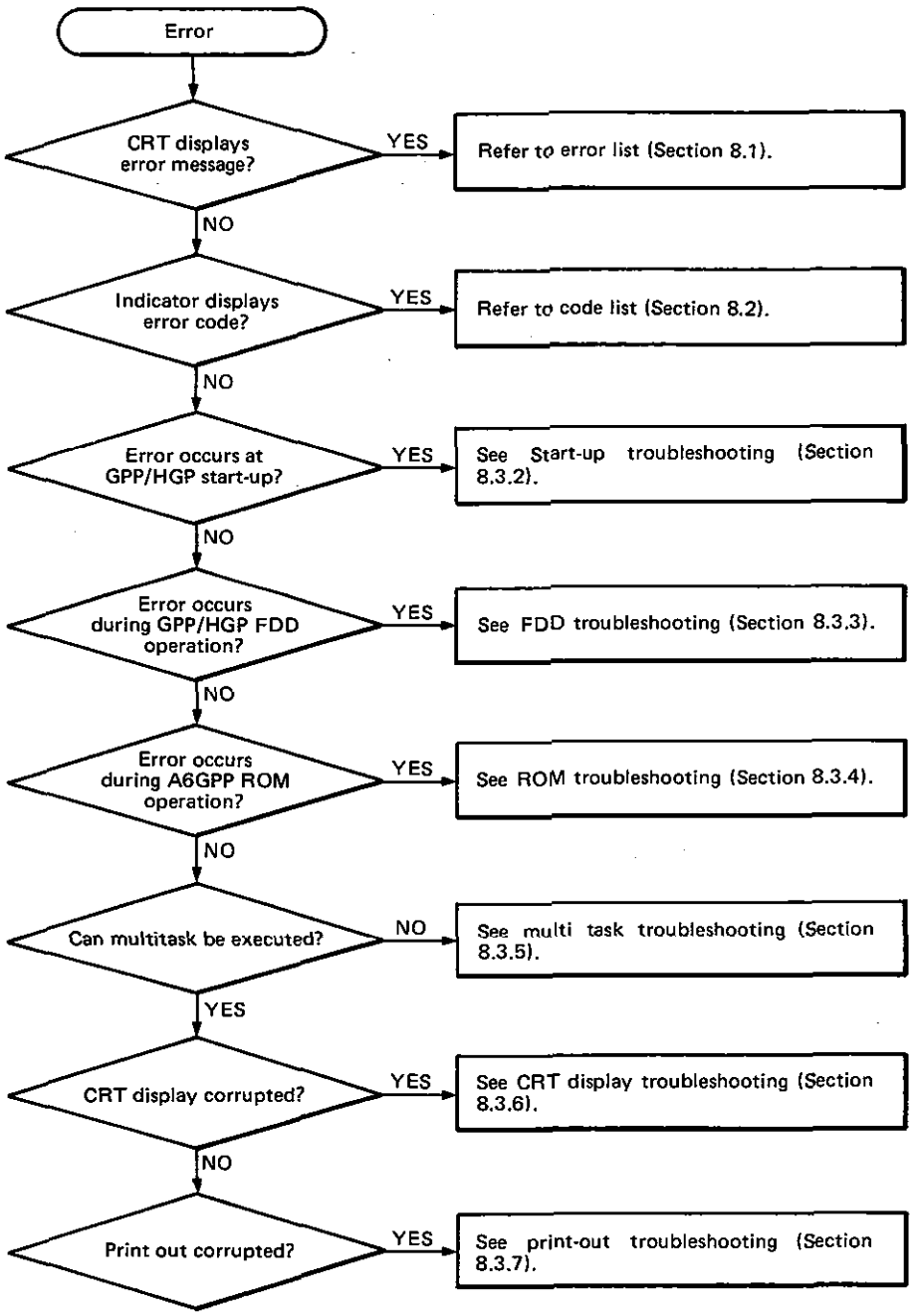
(4) Possible causes of send buffer full error are as follows:

- ○ The DTR signal from the external equipment (Pin 6 of the RS-232C connector) is low.
- ○ X ON code is not received from the external equipment after X OFF has been received.

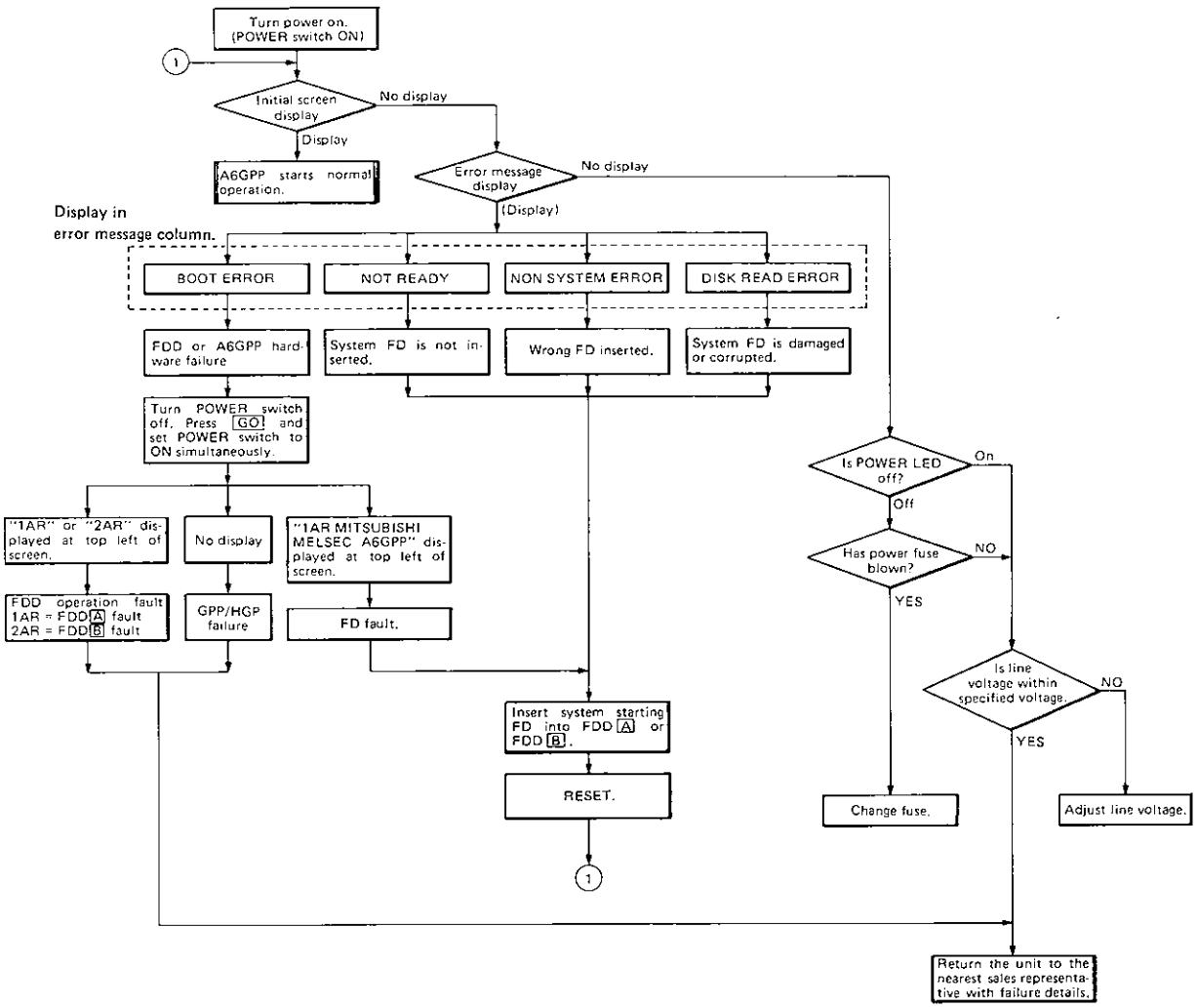
8.3 Troubleshooting

This section gives simple AD51 troubleshooting procedures. For PC CPU troubleshooting refer to the PC CPU User's Manual.

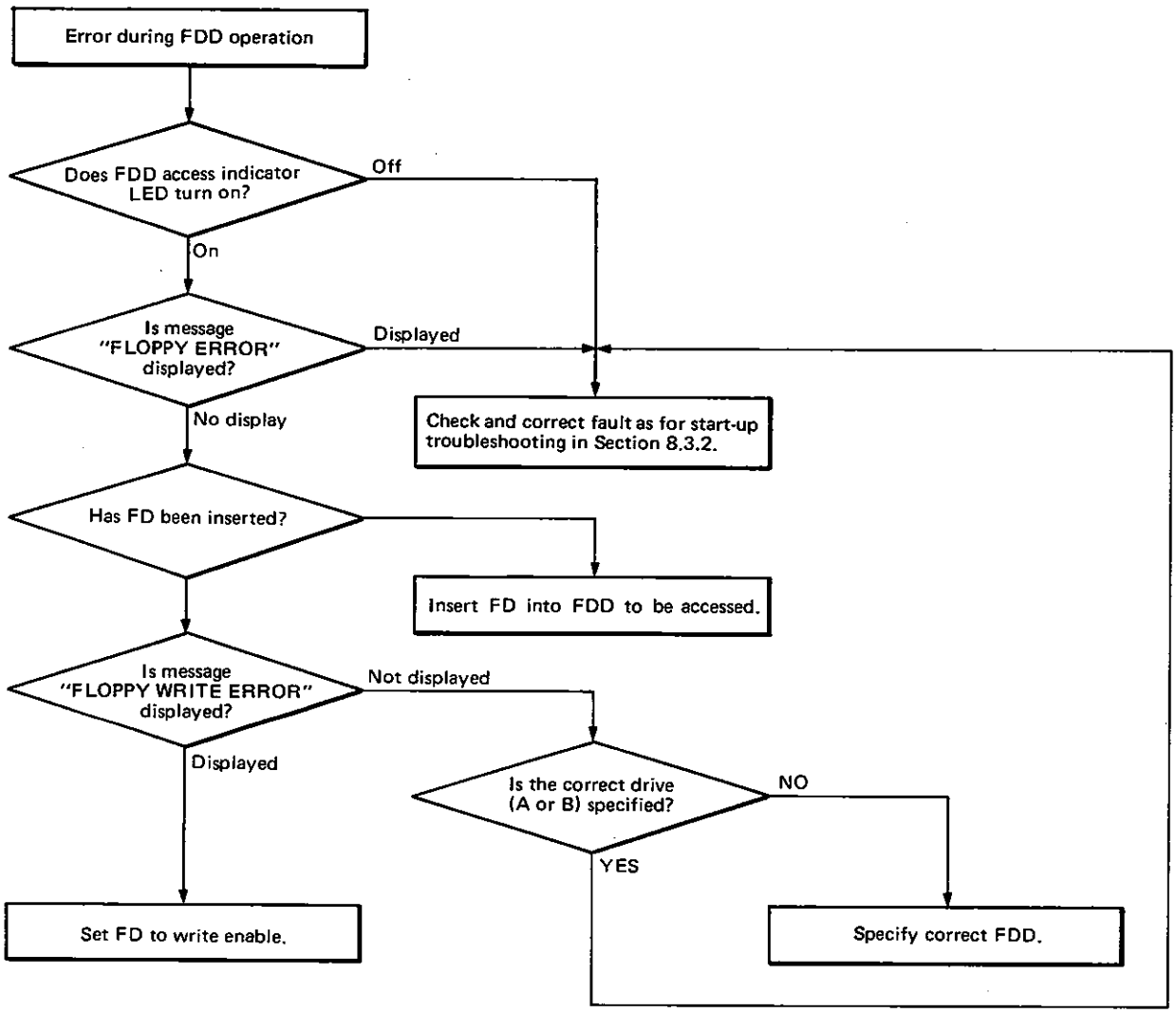
8.3.1 Troubleshooting flow chart



8.3.2 Start-up troubleshooting



8.3.3 FDD troubleshooting

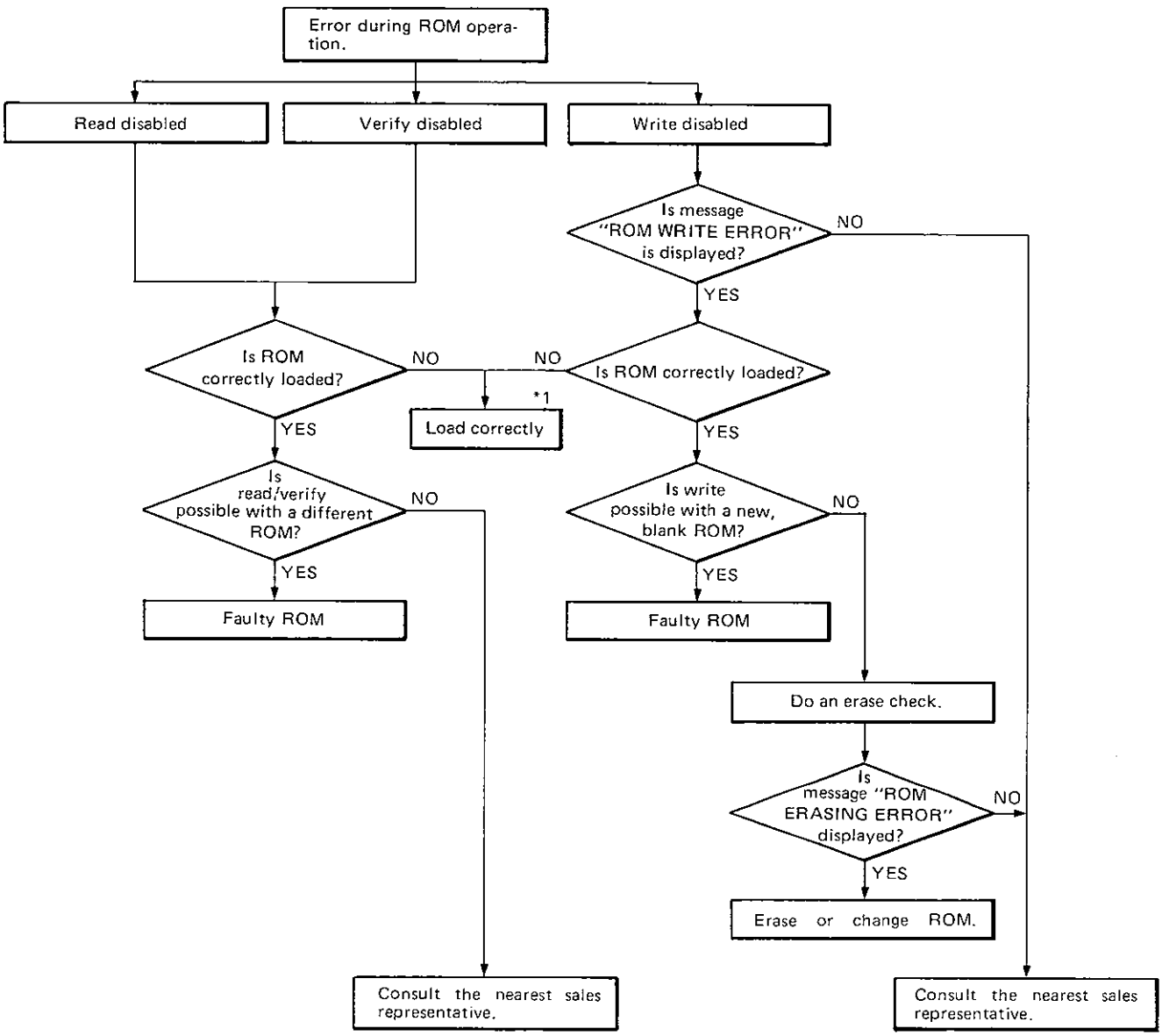


POINT

(1) Note the correct direction for inserting the disk and never force it.

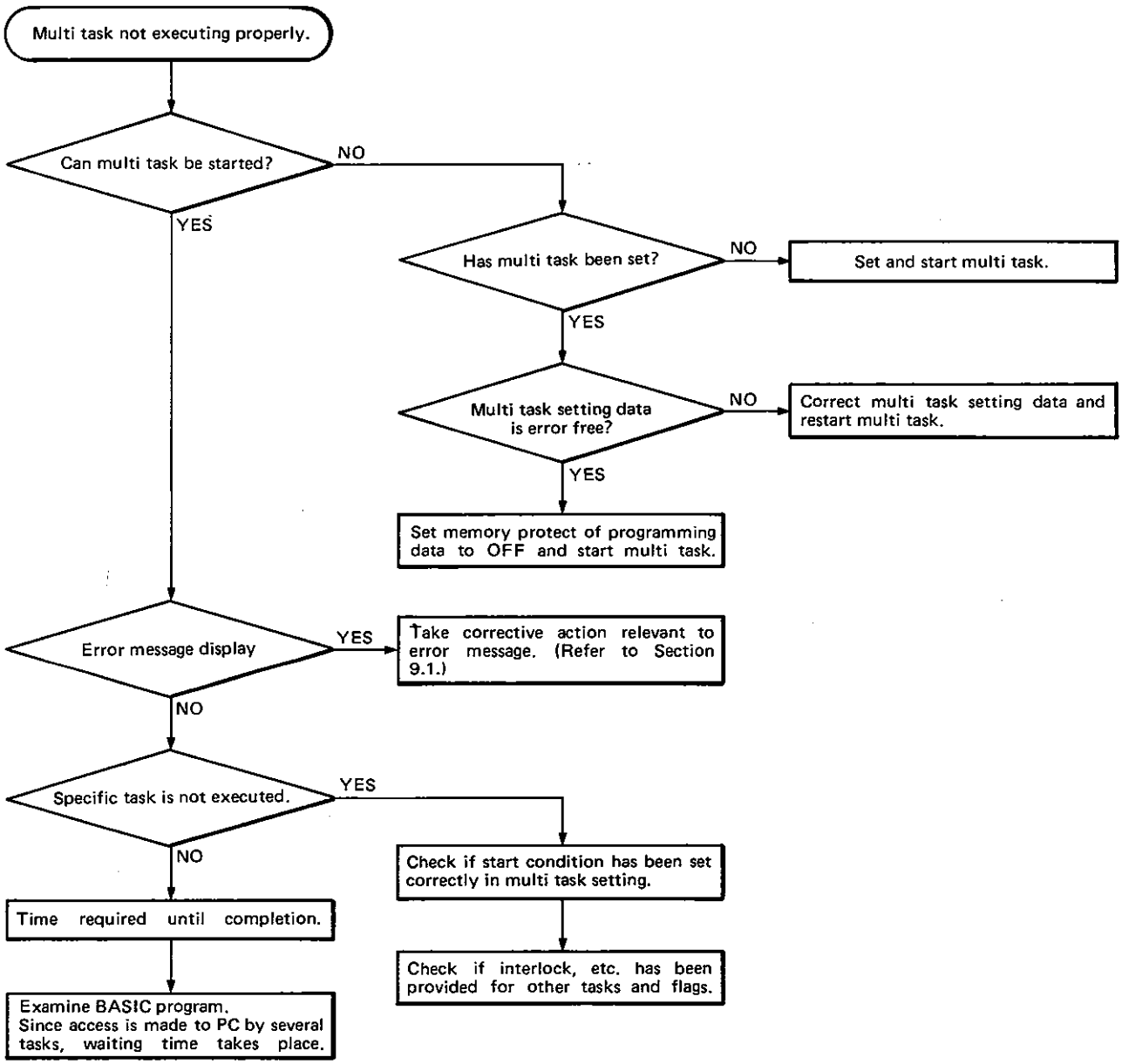
(2) Formatting a disk clears all the data on it.

8.3.4 ROM troubleshooting

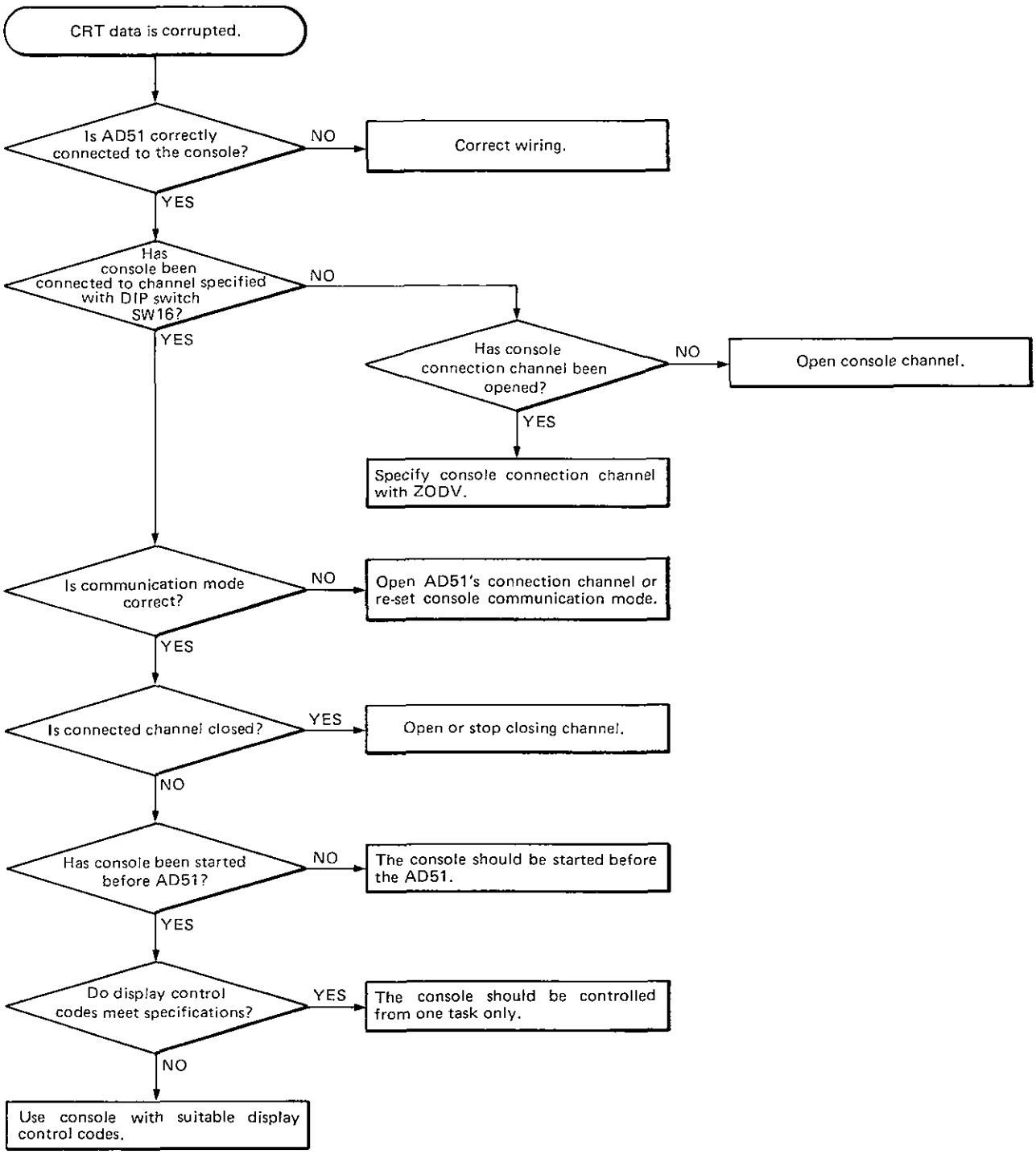


*1: Load correctly.
1) Is ROM in the right direction?
2) Is ROM securely loaded in socket?
3) Is ROM socket lever set correctly?

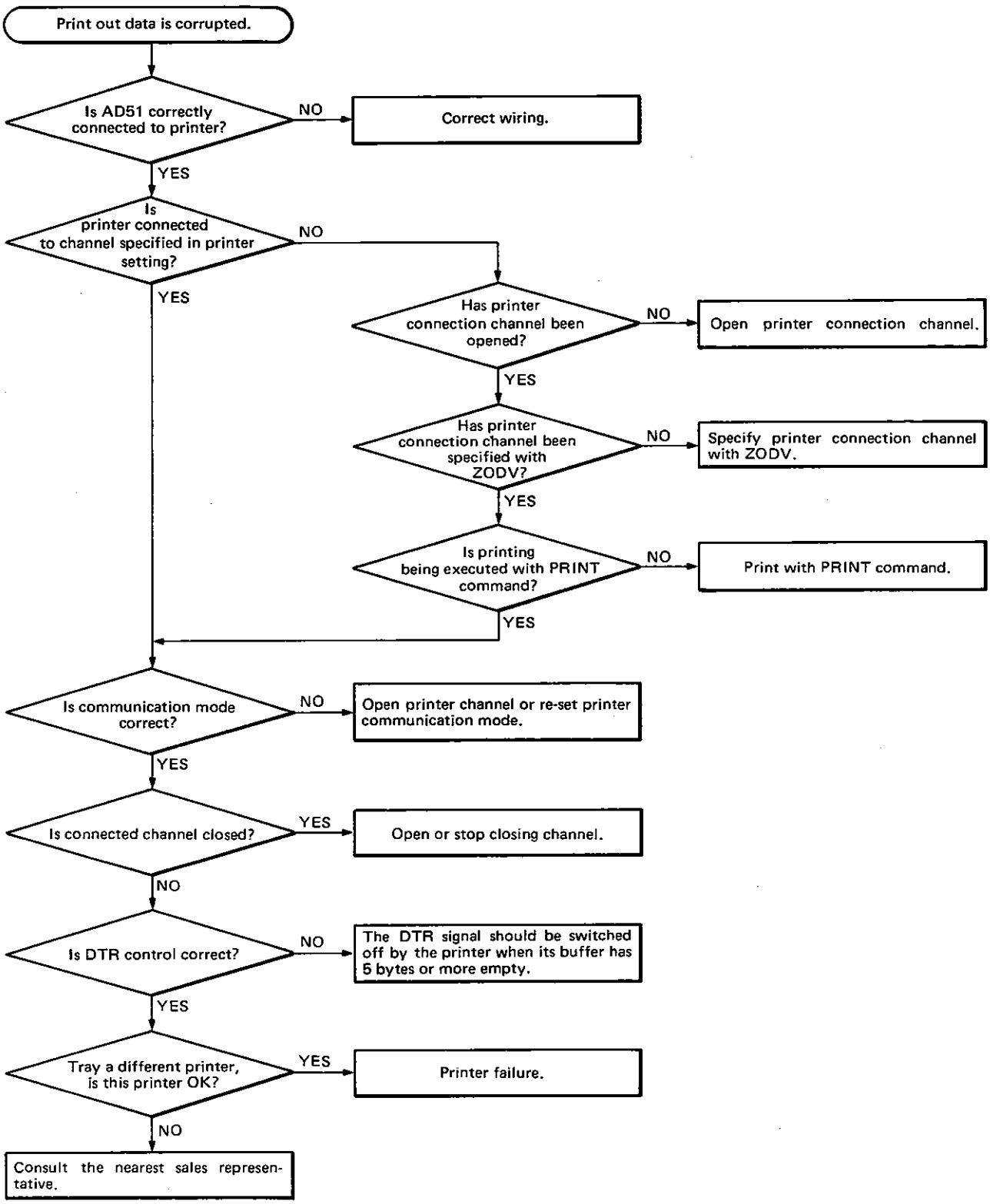
8.3.5 Multi task troubleshooting



8.3.6 CRT troubleshooting



8.3.7 Print-out troubleshooting



9. MAINTENANCE

9.1 Battery Life

When the battery voltage drops, the error indicator on the front of the AD51 displays "00". The error can also be read using the SIR system subroutine.
After this error message is displayed the battery has a further life of 65 days (1560 hours).

	Guaranteed Value	Actual Operation Period (Av.)
Back-up by battery	130 days (3120 hours)	300 days (7200 hours)
Back-up after battery error	65 days (1560 hours)	–
Back-up by capacitor	11 minutes	25 minutes

Preventitive maintenance guide.

(1) Change the battery after four years if the total battery back-up time during that period has been a maximum of 130 days.

(2) For back-up periods exceeding a total of 130 days in four years, calculate the battery life as follows:

Example

Assume that the power is off for 14 hours 5 days a week, and all day for the remaining 2 days per week. Under these conditions, the power is off for:

14(hours) x 5(days) = 70 hours
24(hours) x 2(days) = 48 hours
118 hours per week

The total battery life is 3120 hours which at 118 hours per week gives

$$\frac{3120}{118} = 26.4 \text{ weeks}$$
= Approx. 6 months.

Therefore,

it is necessary to change the battery every 6 months.

IB (NA) 66059-A

9.2 Battery Changing Procedure

Fig. 9.1 shows the battery changing procedure.

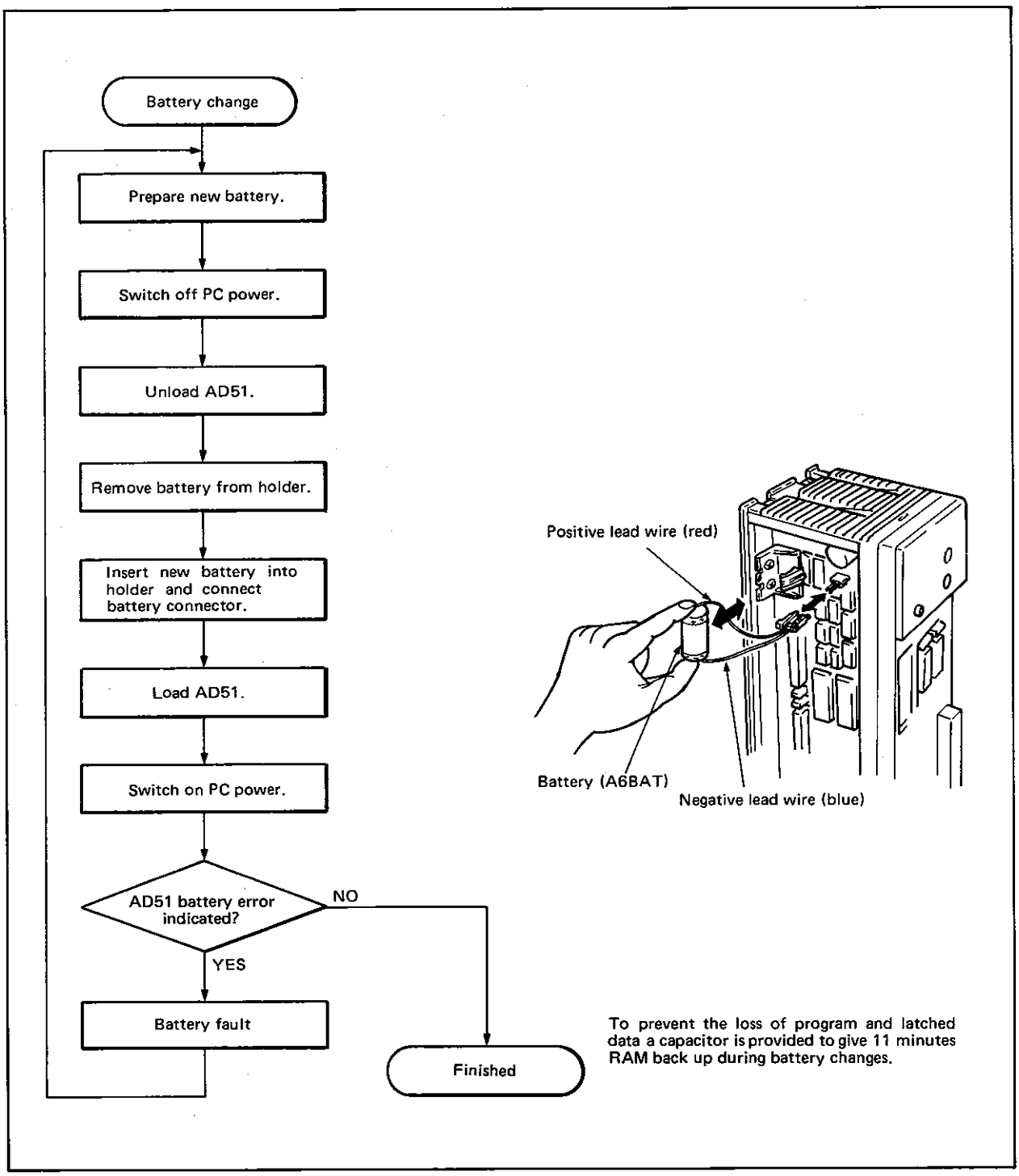


Fig. 9.1 Battery Changing Procedure

REMARKS

The battery is common to all the MELSEC-A series.
Battery storage life is 5 years. Total memory back-up guarantee period is 130 days. Battery used is as follows.

Description : Lithium battery
Type and rating : Type A6BAT (3.6V with plug and leads)

Handling instructions

(1) Do not short.
(2) Do not disassemble.
(3) Do not burn.
(4) Do not heat.
(5) Do not solder electrodes.
(6) Do not measure voltage with an analog voltmeter.

APPENDICES

APPENDIX 1 GPP/HGP Display Control Codes

Function	Description	Code (ASCII)	BASIC Command
Line feed	Carriage return and line feed	CR, LF codes (0DH, 0AH)	–
Screen clear	All screen clear	FF code (0CH)	CLS
XON	Enable transfer from external device.	DC1 code (11H)	–
XOFF	Disable transfer from external device.	DC3 code (13H)	–
Escape	Initiate escape sequence	ESC code (1BH)	–
Back space	Cursor back one space	BS code (08H)	–
Cursor addressing	Set cursor position absolutely.	ESC + Y (59H) + line specification code (20H to 9FH) + column specification code (20H to 9FH)	LOCATE
Character qualification	Character highlight stop	ESC + O (4FH)	ZNOR
	Character highlight	ESC + R (52H)	ZCRV
	Cursor ON	ESC + S (53H)	ZCON
	Cursor OFF	ESC + T (54H)	ZCOFF
Audible alarm	Bell	BEL code (07H)	–

Display Control Code List

APPENDIX 2 VT220 Control Codes

The following VT control codes are used in the AD51E system.

	Function	Description	Code
1	Bell	Audible warning/confirmation (specified at SET-UP).	BEL code (07H)
2	Back space	Move the cursor one column to the left. In column 1, the cursor does not move.	BS code (08H)
3	Line feed	Move the cursor to the next line in the same column (not when set in "new line mode")	LF code (0AH)
4	Carriage return	Move the cursor to the first place in the same line in other than new line mode.	CR code (0DH)
5	XON	Enable transmission.	DC1 code (11H)
6	XOFF	Disable transmission.	DC3 code (13H)
7	Escape	Initiate escape sequence	ESC code (1BH)

APPENDIX 3 GPP/HGP Key Codes and Character Codes

(1) GPP/HGP key codes

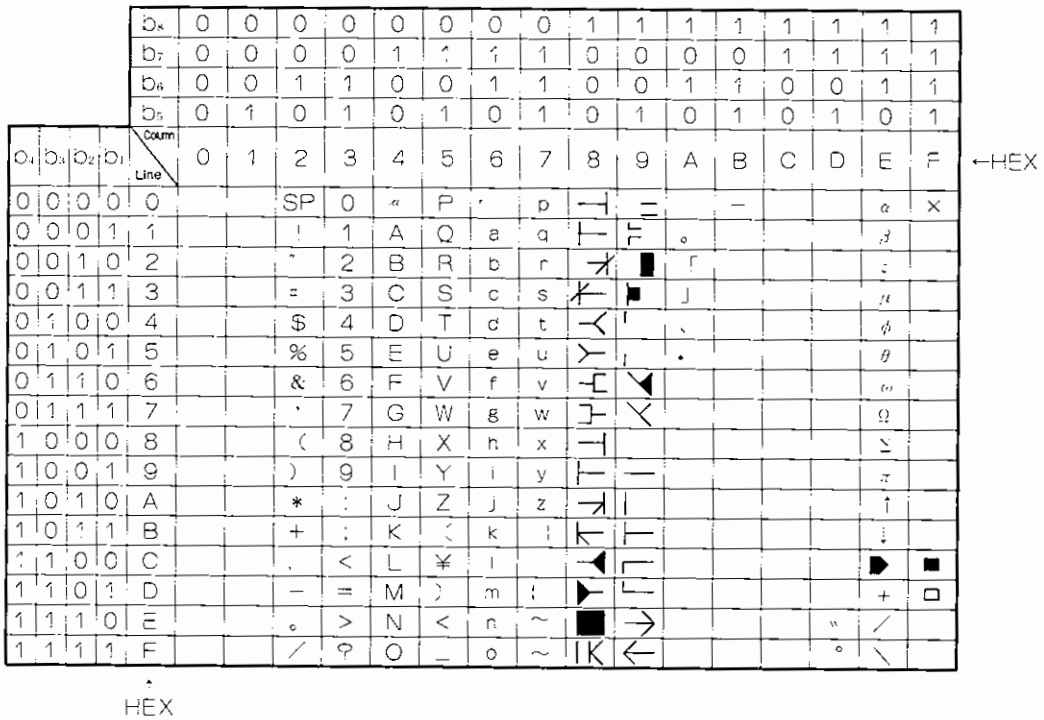
	b_x	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	
	b_7	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	
	b_6	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	
	b_5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
b_4 b_3 b_2 b_1	Column / Line	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	←HEX
0 0 0 0	0		CTRL P	SP	0	‴	P		p	INS			−					
0 0 0 1	1	CTRL A	CTRL Q	!	1	A	Q	a	q	↑	F1	。						
0 0 1 0	2	CTRL B	CTRL R	"	2	B	R	b	r	↓	F2	「						
0 0 1 1	3	BREAK	CTRL S	=	3	C	S	c	s	→	F3	」						
0 1 0 0	4	CTRL D	CTRL T	$	4	D	T	d	t	←	F4	、						
0 1 0 1	5	CTRL E	CTRL/U	%	5	E	U	e	u		F5	・						
0 1 1 0	6	CTRL F	CTRL/V	&	6	F	V	f	v		F6							
0 1 1 1	7	CTRL G	CTRL/W	'	7	G	W	g	w		F7							
1 0 0 0	8	BS	CAN	(	8	H	X	h	x		F8							
1 0 0 1	9	HTAB	CTRL/Y	)	9	I	Y	i	y		F9							
1 0 1 0	A	LF	CTRL/Z	*	:	J	Z	j	z		F10							
1 0 1 1	B	CTRL/K	ESC	+	;	K	[	k	{									
1 1 0 0	C	CTRL/L	CTRL/¥	,	<	L	¥	l	|									
1 1 0 1	D	CR	CTRL/]	−	=	M	]	m	}									
1 1 1 0	E	CTRL/N	CTRL/^	.	<	N	>	n	~							〃		
1 1 1 1	F	CTRL/O	CTRL/		?	O	_	o	DEL							°		

↑ HEX

*For CTRL/[ ], press [ ] and CTRL simultaneously.

Key	Key Code
MELSAP	ESC + "(" (28H) + "M" (4DH)
GPP	ESC + "(" (28H) + "G" (37H)

(2) GPP/HGP display character code



(3) Selected character list

Key code		English	German	Swedish	Japanese
1	5e	^	^	Ü	^
	7e	~	β	ü	~
2	5c	\	Ö	Ö	¥
	7c	¦	ö	ö	¦
3	40	@	§	É	@
	60	`	`	é	`
4	5b	[	Ä	Ä	[
	7b	{	ä	ä	{
5	5d	]	Ü	Å	]
	7d	}	ü	å	}
6	24	$	$	¤	$
7	23	£	#	#	#

Escape Sequence

	Function	Description	Code
1	Screen clear	All screen erase	ESC + [+ 2 + J
2	Cursor addressing	Set cursor position.	ESC + [+ (line specification) +; + (column specification) + H (*)
3	Character *qualification*	Character qualification OFF	ESC + [+ 0 + m
		Character highlight	ESC + [+ 7 + *m*
4	Cursor home	Move the cursor to home position.	ESC + [+ H

*: Line specification . . . . . .1 to 24
Column specification . . . . .1 to 80

Example: To specify line 5 and column 10

ESC + [+ 5 (35H) + ; + 1 (31H) + 0 (30H) + H

Line specification Column specification

Note: The LOCATE command counts the line and column, starting at 0. If "LOCATE 0, 0" is executed, code "ESC + [+ 1 + ; + 1 + H" is transmitted to the VT220.

APPENDIX 4 Storing the AD51E Memory Data into ROM Using the A6WU

The AD51E internal memory and buffer memory data can be stored into the ROM with the A6WU P-ROM writer unit connected with the PC CPU. For the operating procedure, see the A6WU Operating Manual.

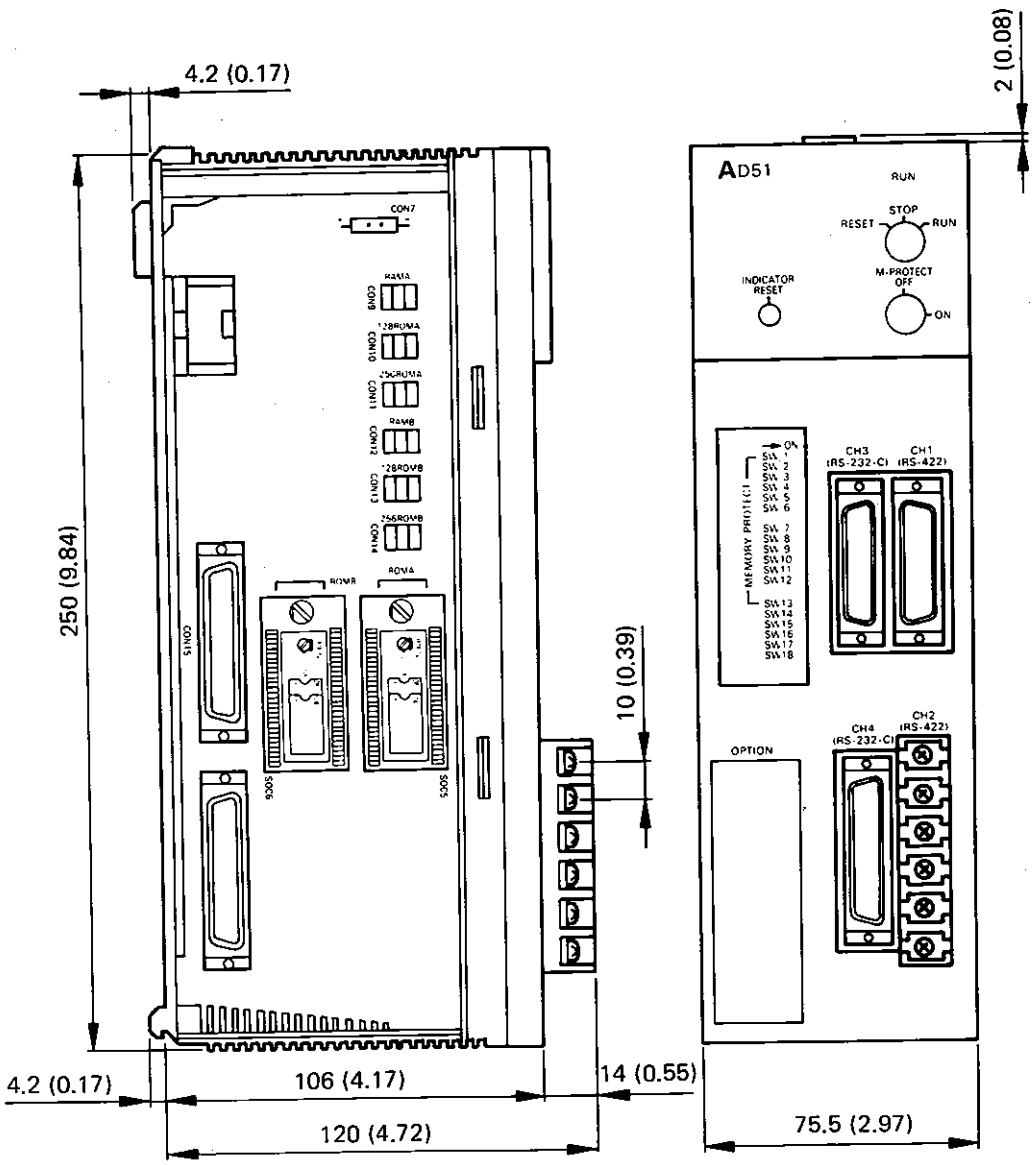
The addresses must be set as follows when the AD51E data is written to the ROM using the A6WU.

		AD51E Addresses	Addresses Set by A6WU
Programming data		4F81H to 4FD0H	4F81H to 4FD0H
Common area		6000H to 67FFH	6000H to 67FFH
Channel area	CH1	8000H to FFFFH	8000H to FFFFH
	CH2	8000H to FFFFH	10000H to 17FFFH
	CH3	8000H to DFFFH	18000H to 1DFFFH
	CH4	8000H to DFFFH	20000H to 25FFFH
Buffer memory		000H to BFFH (0 to 3071)	0 to 3071

REMARKS

Data may only be stored to the ROM if the AD51E is connected with the A1, A2, A3 or A3HCPU.

APPENDIX 5 External View



IMPORTANT

The components on the printed circuit boards will be damaged by static electricity, so avoid handling them directly. If it is necessary to handle them take the following precautions.

(1) Ground human body and work bench.

(2) Do not touch the conductive areas of the printed circuit board and its electrical parts with any non-grounded tools etc.

Under no circumstances will Mitsubishi Electric be liable or responsible for any consequential damage that may arise as a result of the installation or use of this equipment.

All examples and diagrams shown in this manual are intended only as an aid to understanding the text, not to guarantee operation. Mitsubishi Electric will accept no responsibility for actual use of the product based on these illustrative examples.

Owing to the very great variety in possible applications of this equipment, you must satisty yourself as to its suitability for your specific application.

MITSUBISHI ELECTRIC CORPORATION
HEAD OFFICE: MITSUBISHI DENKI BLDG MARUNOUCHI TOKYO 100 TELEX: J24532 CABLE MELCO TOKYO
NAGOYA WORKS : 1-14 , YADA-MINAMI 5 , HIGASHI-KU , NAGOYA , JAPAN

IB (NA) 66059-B (8705) REC Printed in Japan