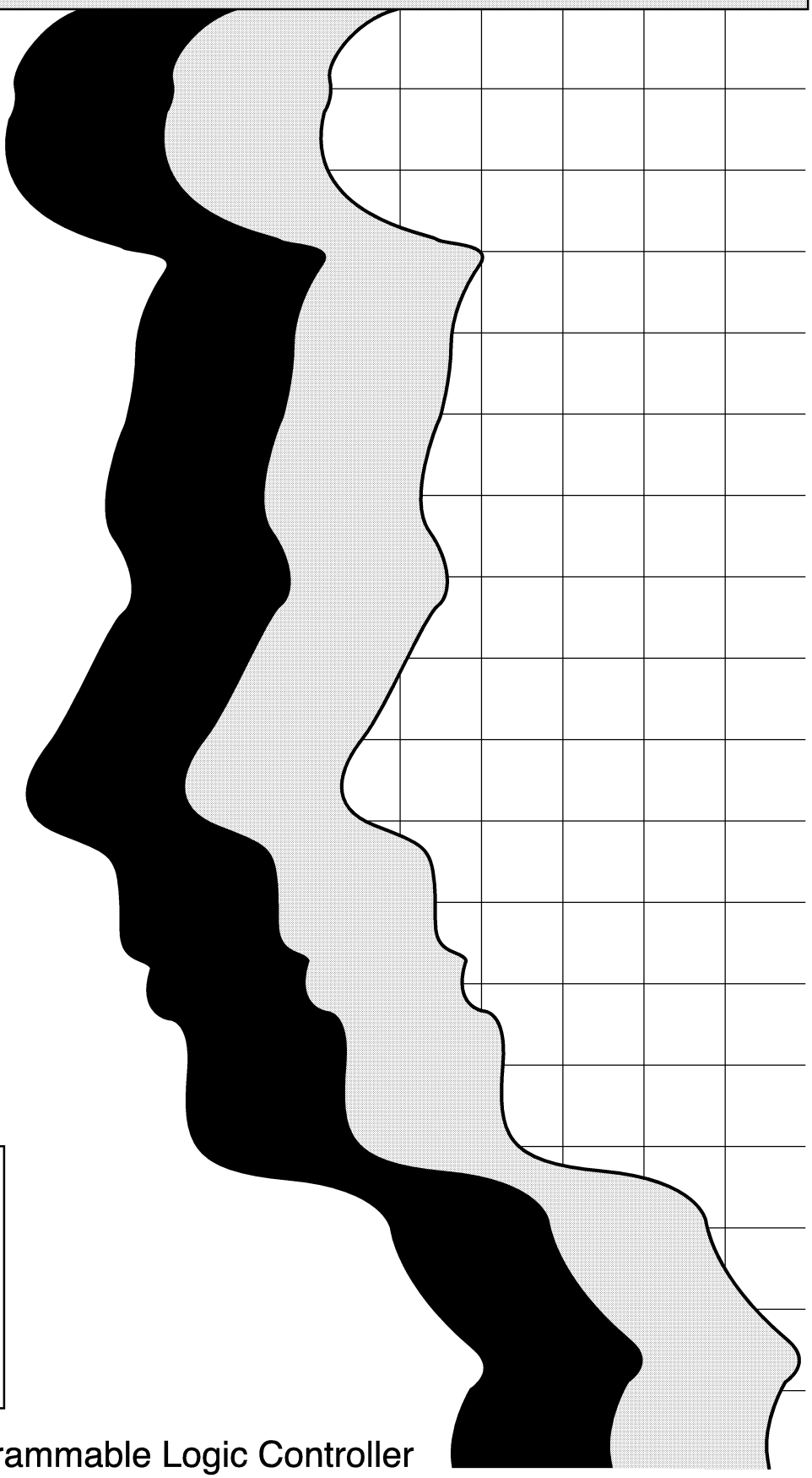
MITSUBISHI

Utility Software Package type SW0GHP-UTLPC-FN1

Operating Manual



MELSEC

Mitsubishi Programmable Logic Controller

REVISIONS

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

Print Date	※Manual Number	Revision
Sep., 1988	IB (NA) 66169-A	First edition
Sep., 1989	IB (NA) 66169-B	Correction Page 4-13

INTRODUCTION

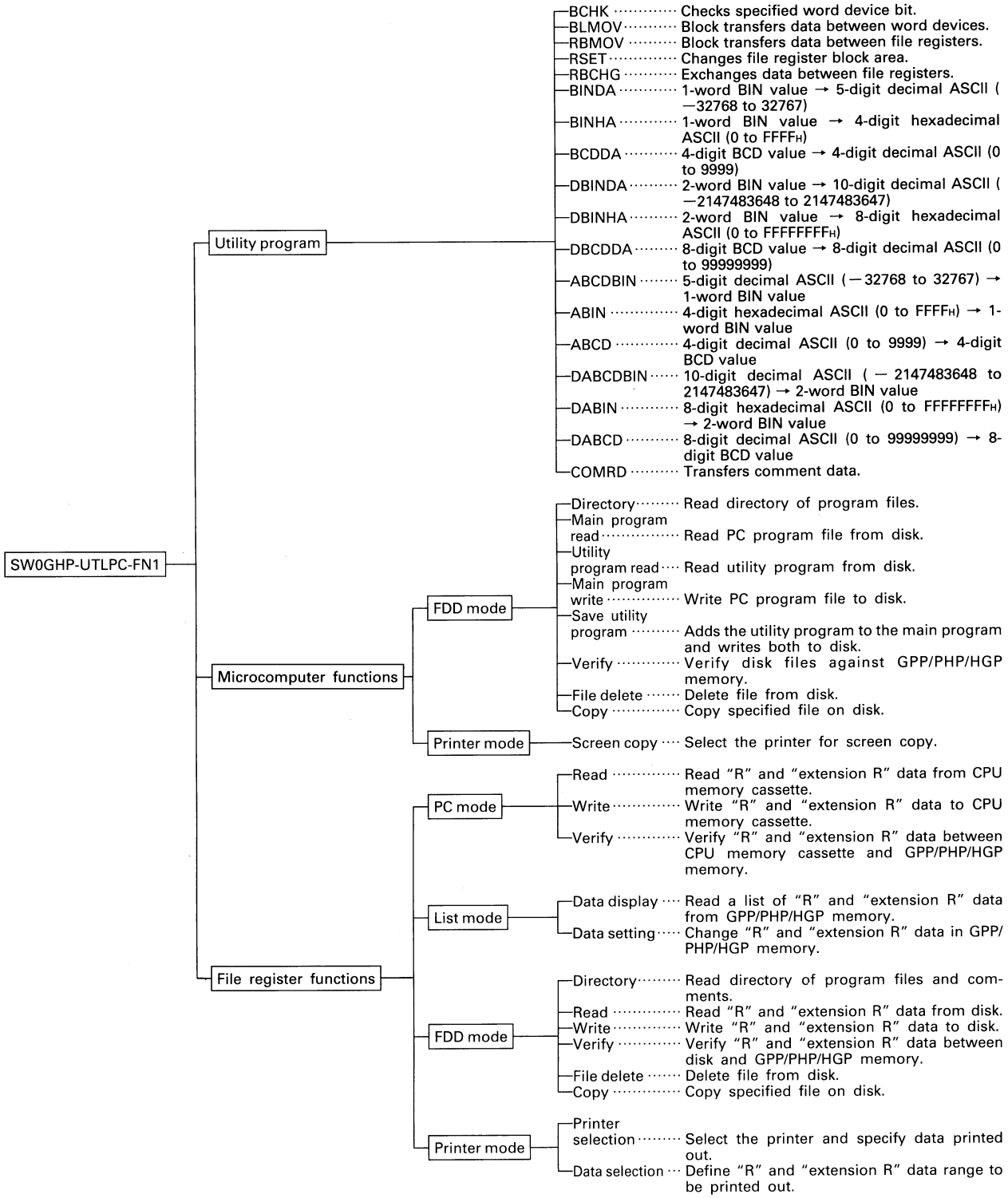
Thank you for choosing the Mitsubishi MELSEC-A Series of General Purpose Programmable Controllers. Please read this manual carefully so that the equipment is used to its optimum. A copy of this manual should be forwarded to the end User.

CONTENTS

1. INTRODUCTION

This manual gives operating instructions for the SW0GHP-UTLPC-FN1 utility package (referred to as "UTLP-FN1").
The UTLP-FN1 package may be used with any of the A-Series GPP units to allow a further set of commands, e.g. word device bit check, expanded ASCII conversion, block transfer between register memories, and use of the memory cassette empty area as extension file registers (referred to as "extension R"), to be implemented on specified MELSEC-A series programmable controllers.

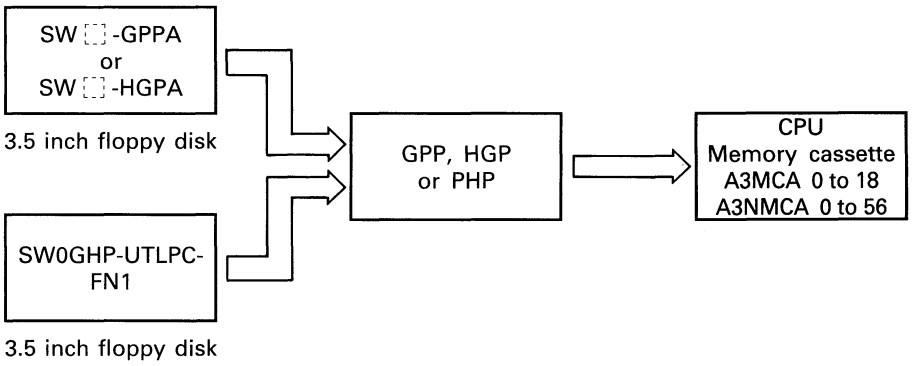
1.1 Functions of the UTLP-FN1 System Disk



IB (NA) 66169-A

2. SYSTEM CONFIGURATION

2.1 Overall Configuration



2.2 Applicable CPUs

The UTLP-FN1 utility package can be used with the following CPUs:

○ : Applicable

	Instruction	CPU Type			
		A1(E)CPU(P21/R21) A1NCPU(P21/R21)	A2(E)CPU(P21/R21) A2NCPU(P21/R21)	A3(E)CPU(P21/R21) A3NCPU(P21/R21)	A3HCPU(P21/R21)
1	BCHK	○	○	○	○
2	BLMOV	○	○	○	○
3	RBMOV	—	○	○	○
4	RSET	—	○	○	○
5	RBCHG	—	○	○	○
6	BINDA	○	○	○	○
7	BINHA	○	○	○	○
8	BCDDA	○	○	○	○
9	DBINDA	○	○	○	○
10	DBINHA	○	○	○	○
11	DBCDDA	○	○	○	○
12	ABCDBIN	○	○	○	○
13	ABIN	○	○	○	○
14	ABCD	○	○	○	○
15	DABCDBIN	○	○	○	○
16	DABIN	○	○	○	○
17	DABCD	○	○	○	○
18	COMRD	○	○	○	○

Table 2.1 Applicable CPUs

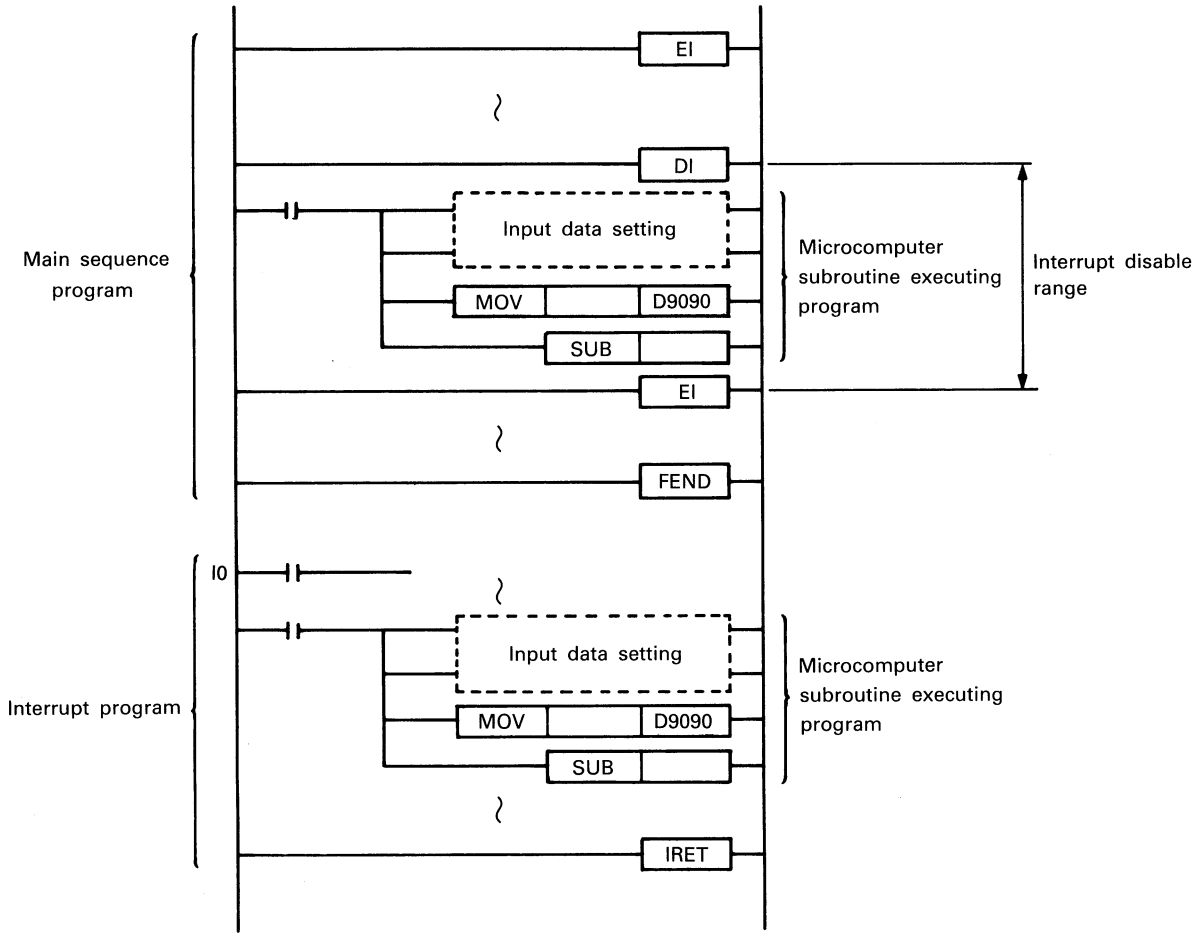
2.3 General Notes

(1) The utility package cannot be used in the sub-program area of the A3(E), A3N and A3HCPUs.

(2) When a utility package is written to the microcomputer program area, any previous user written code is cleared. If user written microcomputer programs are being used, write these to the PC memory after the utility package(s) has been loaded.

(3) A utility package and user written programs must be written to separate work areas in accordance with Section 4.2 (4).

(4) When microcomputer subroutine executing programs are used in both the main sequence program and interrupt program, the DI (interrupt disable) instruction and EI (interrupt enable) instruction must be used to disable interrupt as shown below:



3. SPECIFICATIONS

3.1 Specifications and Commands

There are five blocks of functions which may be implemented using the UTLP-FN1 package.
These function blocks may be implemented individually in the PC or all together, in which case they are referred to as A-FN1.
To execute any given operation, the required command is called from the microcomputer program area using the —[SUB][n]— instruction where n refers to the address location of the command required.

Microcomputer Program Block		Command	Function	SUB Call Address —[SUB][n]—		Program Capacity (Bytes)	
				A-FN1	A-BTST, FIDT, MCBT, ASBS, ASBD		
A-FN1	A-BTST	BCHK	Checks specified word device bit.	*A_1 + 5_H	*A_2 + 5_H	4608	1024
	A-FIDT	BLMOV	Block transfers data between word devices.	A_1 + 8_H	*A_3 + 5_H		512
	A-MCBT	RBMOV	Block transfers data between file registers.	A_1 + E_H	*A_4 + 8_H		2048
		RSET	Changes file register block area.	A_1 + B_H	A_4 + 5_H		
		RBCHG	Exchanges data between file registers.	A_1 + 38_H	A_4 + E_H		
		COMRD	Transfers comment data.	A_1 + 35_H	A_4 + B_H		
	A-ASBS	BINDA	1-word BIN value →5-digit decimal ASCII (−32768 to 32767)	A_1 + 11_H	*A_5 + 5_H		2048
		BINHA	1-word BIN value →4-digit hexadecimal ASCII (0 to FFFF$_H$)	A_1 + 14_H	A_5 + 8_H		
		BCDDA	4-digit BCD value →4-digit decimal ASCII (0 to 9999)	A_1 + 17_H	A_5 + B_H		
	A-ASBD	DBINDA	2-word BIN value →10-digit decimal ASCII (−2147483648 to 2147483647)	A_1 + 23_H	*A_6 + 5_H		2048
		DBINHA	2-word BIN value →8-digit hexadecimal ASCII (0 to FFFFFFFF$_H$)	A_1 + 26_H	A_6 + 8_H		
		DBCDDA	8-digit BCD value →8-digit decimal ASCII (0 to 99999999)	A_1 + 29_H	A_6 + B_H		
	A-ASBS	ABCDBIN	5-digit decimal ASCII (−32768 to 32767) →1-word BIN value	A_1 + $1A_H$	A_5 + E_H		2048
		ABIN	4-digit hexadecimal ASCII (0 to FFFF$_H$) →1-word BIN value	A_1 + $1D_H$	A_5 + 11_H		
		ABCD	4-digit decimal ASCII (0 to 9999) →4-digit BCD value	A_1 + 20_H	A_5 + 14_H		
	A-ASBD	DABCDBIN	10-digit decimal ASCII (−2147483648 to 2147483647) →2-word BIN value	A_1 + $2C_H$	A_6 + E_H		2048
		DABIN	8-digit hexadecimal ASCII (0 to FFFFFFFF$_H$) →2-word BIN value	A_1 + $2F_H$	A_6 + 11_H		
		DABCD	8-digit decimal ASCII (0 to 99999999) →8-digit BCD value	A_1 + 32_H	A_6 + 14_H		

Table 3.1 Specifications and Commands

REMARKS

As the individual packages have corresponding common subroutines, program capacities differ between the FN1 package and the total of individual packages.

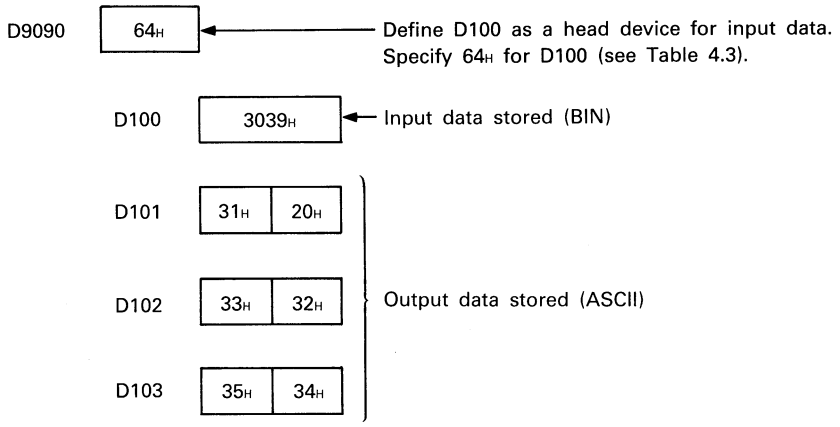
POINT

***: A1 to A6 are the head addresses (hexadecimal) of the utility program determined by the microcomputer program block used. The utility program is automatically added to the sequence program and is displayed in the START AD column of the entry sequence setting screen by pressing EDIT. See Section 6.1.7.**

3.2 Operation Inputs/Outputs

Special register D9090 is used to specify the head device number which contains the input data.
When the device number is set to D9090, the input data and output data are automatically written to the registers. The register number should not be repeated. For device number identification, see Section 4.4.

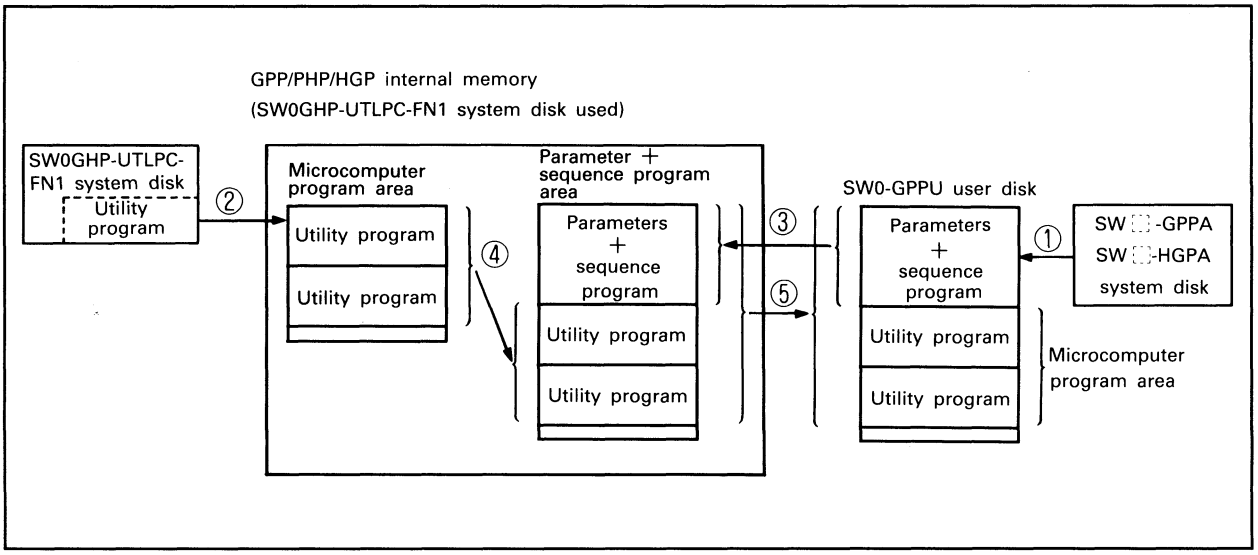
Example: To set input data to D100 and call microcomputer subroutine (BINDA command)

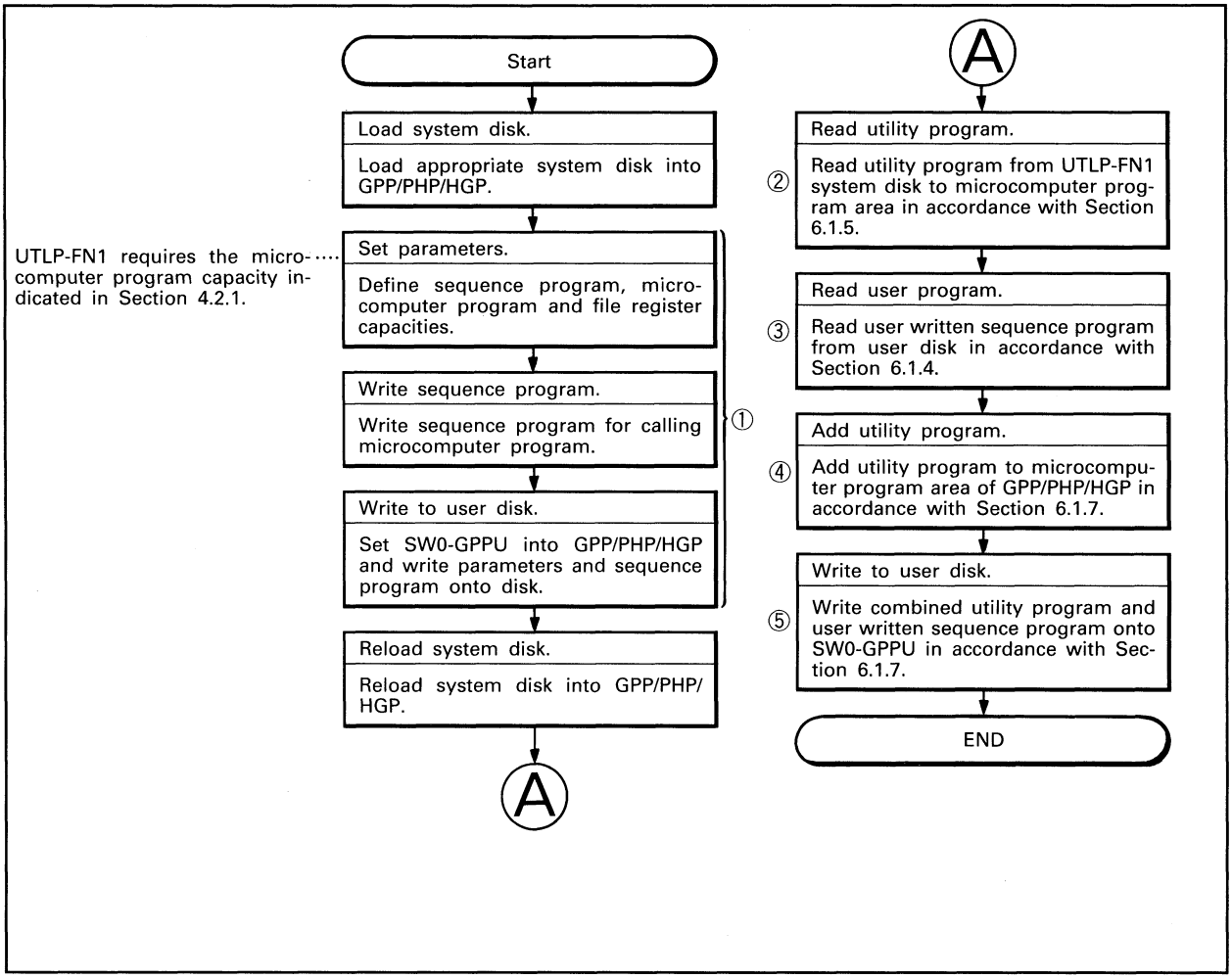


4. PROGRAMMING

4.1 Programming Procedure

Add the utility program to the user written sequence program and write both to user disk in the following procedure:





IB (NA) 66169-A

4.2 Memory Map

The head address of the FN1 utility program depends on the order written when the utility program is written to the microcomputer program area together with the other microcomputer programs. The memory map shown in Fig. 4.1 below assumes that the FN1 utility program is written independently or before several microcomputer programs. The utility program area is divided into two: address range 0_H to $6F_H$ is the common area and address range 70_H to n is the actual program area.

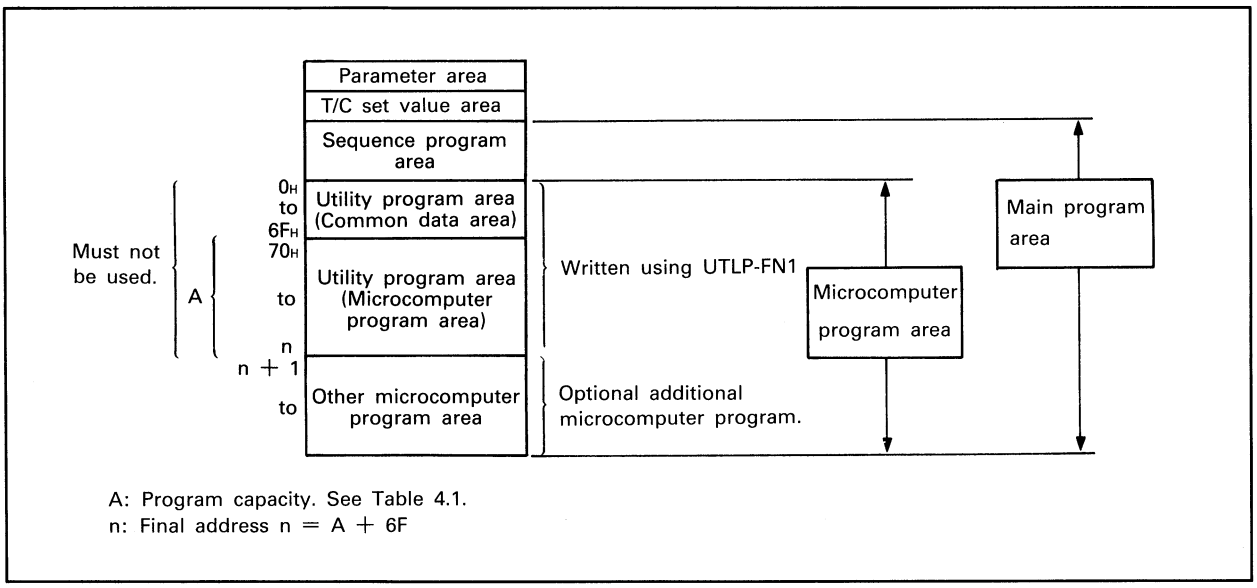


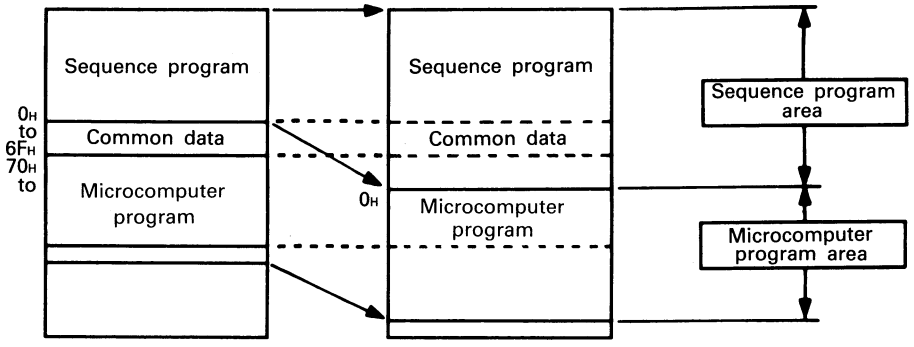
Fig. 4.1 Memory Map

(1) The head address of the utility program area is automatically set when the utility program is added to the sequence program, and it cannot be changed.

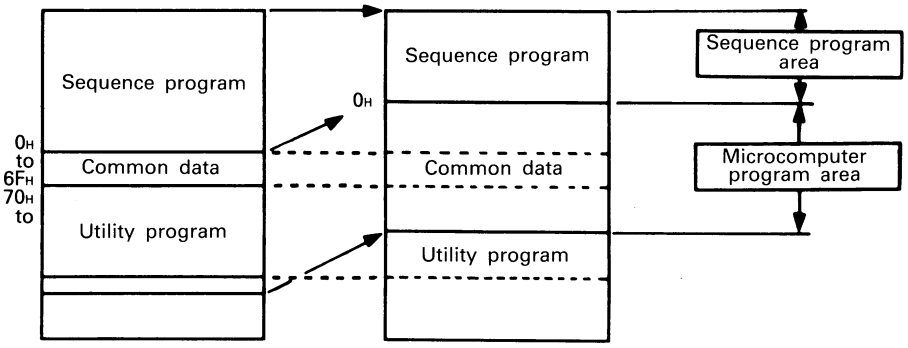
(2) When the utility program is added to the sequence program, the microcomputer program area is cleared. For this reason, any additional user written microcomputer program must be written after the utility program has been added.

(3) The utility program cannot be executed if the common data is absent from addresses 0_H to $6F_H$ in the microcomputer area.
The utility program must be re-entered if the PC sequence program capacity is changed in order to correct the microcomputer program area head address.

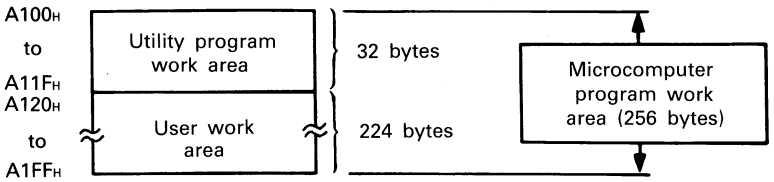
1) Capacity increased



2) Capacity reduced



(4) The microcomputer program general work area is at addresses A100H to A1FFH and the first 32 bytes of this work area are used by the utility program. This area should therefore be left free of any additional microcomputer programs which would otherwise be corrupted.
The user work area depends on the FN1 utility program used and must therefore be maximum when the program is combined with any other utility program.



(5) For full details, consult the ACPU programming manual.

4.2.1 Calculating microcomputer program capacity

Microcomputer Program Block	Program Capacity (Bytes)
A-FN1	4608 (1200H)
A-BTST	1024 (400H)
A-FIDT	512 (200H)
A-MCBT	2048 (800H)
A-ASBS	2048 (800H)
A-ASBD	2048 (800H)

Table 4.1 Utility Program Capacities

Calculate the microcomputer program capacity as follows:

Using A-FN1:

112 (Common data) + 1024 (A-BTST) = 1136 (bytes) → 2K bytes

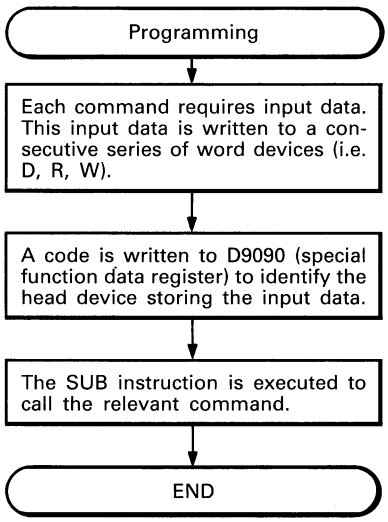
Set the microcomputer program capacity in the parameters (in blocks of 1K byte).

Using A-BTST and A-ASBS:

112 (Common data) + 1024 (A-BTST) + 2048 (A-ASBS) =3184 (bytes) → 4K bytes

4.3 Programming Procedure

To execute any microcomputer subroutine, SUB xxH must be executed after the head device number for the input data is specified at D9090.
The input data and operation result are stored to contiguous registers headed by the one set to D9090.

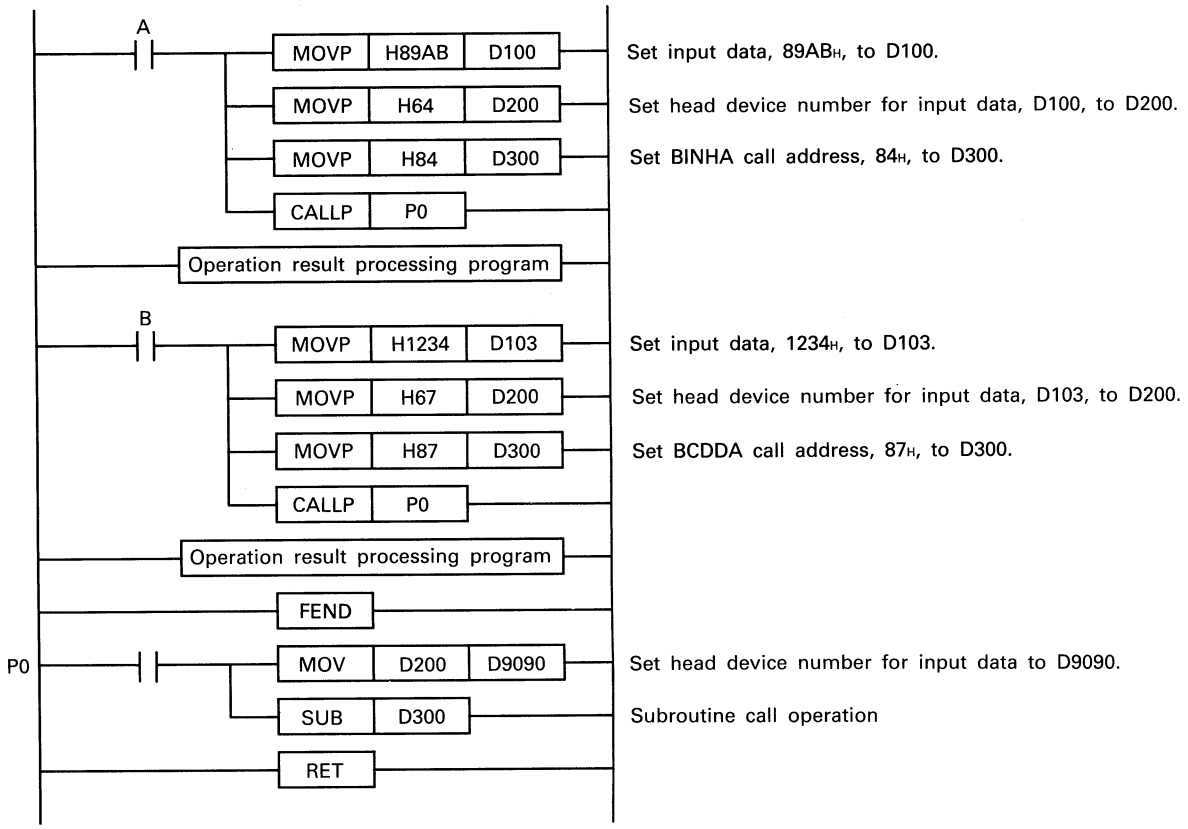


As common registers are used to execute several microcomputer subroutines, data must be set every time SUBxxH is executed. It is therefore recommended to write a subroutine program for common data.

The following example indicates a program which uses several microcomputer subroutines:

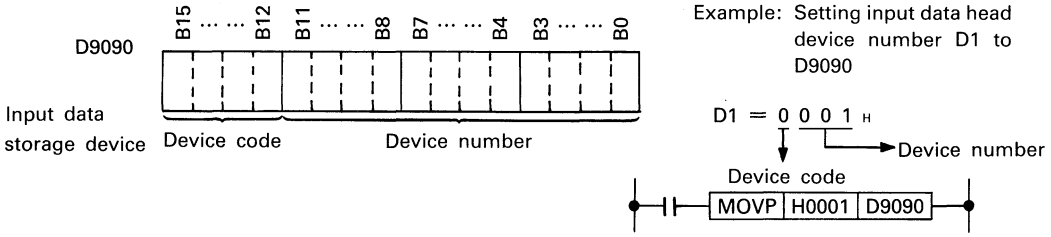
Using BINHA and BCDDA commands

D100	Input data	Work area for BINHA (for 3 words)
D101	Output data	
D102	Output data	
D103	Input data	Work area for BCDDA (for 3 words)
D104	Output data	
D105	Output data	



4.4 Identification of Device Number

Device codes are shown in Table 4.3 and the device number is simply the correct device number in hexadecimal.



Device	Device Code	Device Number	Device Code + Number	Applicable CPU			
				A1	A2	A3	A3H
Data register	0	D0 to D1023	0000H to 03FFH	○	○	○	○
		D9000 to D9255	0400H to 04FFH				
Link register	1	W0 to W3FF	1000H to 13FFH	○	○	○	○
File register	2	R0 to R4095	2000H to 2FFFH		○	○	○
	3	R4096 to R8191	3000H to 3FFFH			○	○
Timer present value	4	T0 to T255	4000H to 40FFH	○	○	○	○
Counter present value	5	C0 to C255	5000H to 50FFH	○	○	○	○
Accumulator	6	A0, A1	6000H to 60FFH	○	○	○	○
Index register	7	Z	7000H	○	○	○	○
		V	7001H	○	○	○	○
Input	8	X0 to X0FF	8000H to 80FFH	○			
		X0 to X1FF	8000H to 81FFH		○		
		X0 to X7FF	8000H to 87FFH			○	○
Output	9	Y0 to Y0FF	9000H to 90FFH	○			
		Y0 to Y1FF	9000H to 91FFH		○		
		Y0 to Y7FF	9000H to 97FFH			○	○
Internal relay	A	M(L,S)0 to M(L,S)2047	A000H to A7FFH	○	○	○	○
		M9000 to M9255	A800H to A8FFH				
Link relay	B	B0 to B3FF	B000H to B3FFH	○	○	○	○
Annunciator	C	F0 to F127	C000H to C07FH	○	○	○	○
Pointer	D	P0 to P255	D000H to D0FFH	○	○	○	○
Interrupt pointer	E	I0 to I31	E000H to E01FH	○	○	○	○

Table 4.3 Device Codes and Numbers

4.5 Special Relay and Special Registers

(1) Special relay M9091 is used as an error indication flag. Use this in conjunction with D9091.

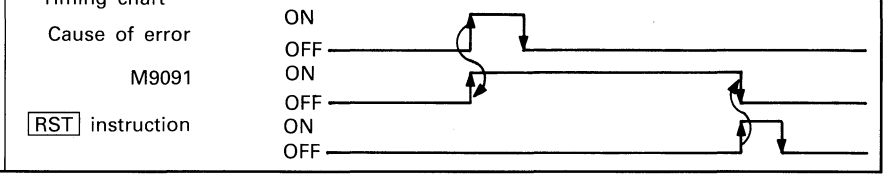
Device Number	Name	Description
M9091	Error detection	• Switched on when an error is detected. • Switched off by RST instruction. • Timing chart 

Table 4.6 Special Relay

(2) Special registers

D9090 is used to feed input data to each command as it is executed. The contents of D9090 is therefore continually changing. D9091 is used to store error codes as appropriate.

Device Number	Name	Description
D9090	Input data head device	• Used to identify the head device storing the input data.
D9091	Error code	• The error code is retained until reset using the RST, MOV instructions, etc. • An error code may be overwritten by a new code if M9091 is off. • For details, see Section 8.1.

Table 4.7 Special Registers

IB (NA) 66169-A

4.6 Bit Check Command

4.6.1 BCHK

Checks the specified word device bit.

Command	Subroutine Address	I/O Data
BCHK	A-FN1 SUB | H (A_1+5H) A-BTST SUB | H (A_2+5H) A_1, A_2: Head address for storing subroutine	D9090 ⓐ (a + 0) ← Code of device checked ⓑ (a + 1) ← Location of bit checked (0 to 15) (a + 2) ← Code of device which receives bit check result ⓒ

	Usable Devices															
	X	Y	M/L/S	B	F	T	C	D	W	R	A0	A1	Z	V	P	I
Device specified at D9090 ⓐ								○	○	○						
Device for input data ⓑ						○	○	○	○	○	○	○				
Device for output data ⓒ		○	○	○												

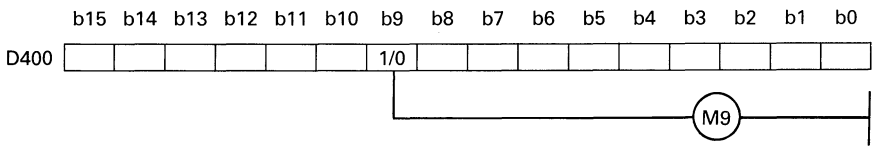
	Error Code	Description	Error Detection Condition	Operation Result at Error Occurrence
Error detection (D9091)	10050	D9009 set value error	Invalid device defined.	Operation not performed.
	10051	Input data device error	Invalid device defined during input data setting	Operation not performed.
	10052	Bit entry error	Bit specified is other than 0 to 15.	Operation not performed.
	10053	Result receive device error	Invalid device defined.	Operation not performed.

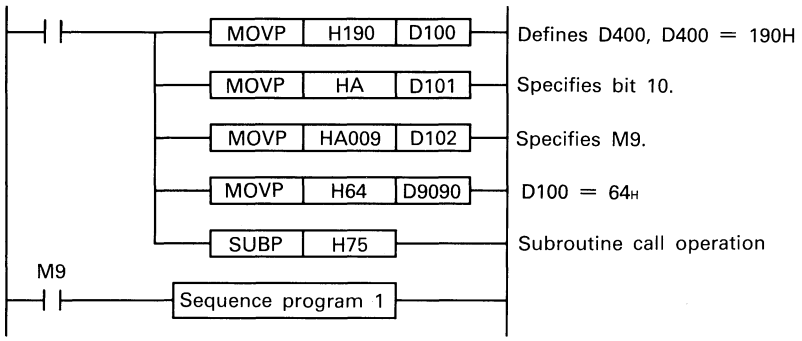
FUNCTION

(1) Checks the specified bit of the specified word device and outputs the result to the specified device.

PROGRAM EXAMPLE

The following program checks bit 10 of D400 and outputs the result to M9. Executes sequence program 1 if M9 is on.





4.7 Indirectly Specified Data Transfer Command

4.7.1 BLMOV

Transfers data in blocks between word devices.

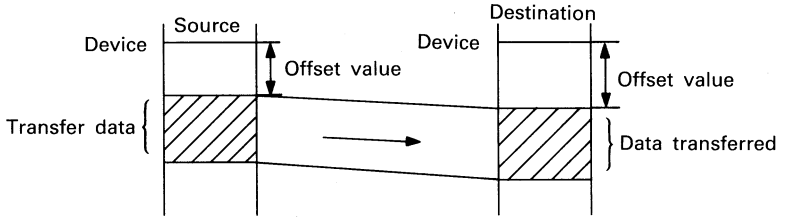
Command	Subroutine Address	I/O Data
BLMOV	A-FN1 SUB | H (A_1+8H) A-FIDT SUB | H (A_3+5H) A_1, A_3: Head address for storing subroutine	D9090 ⓐ (a + 0) ← Device for transfer source input data ⓑ (a + 1) ← Number of words transferred (0 to 8192) (b + 0) ← Transfer source device (b + 1) ← Offset value (within the specified device range) (a + 2) ← Device for transfer destination input data ⓒ (c + 0) ← Transfer destination device (c + 1) ← Offset value (within the specified device range)

	Usable Devices															
	X	Y	M/L/S	B	F	T	C	D	W	R	A0	A1	Z	V	P	I
Device specified at D9090 ⓐ								○	○	○						
Device for transfer source input data ⓑ						○	○	○	○	○						
Device for transfer destination input data ⓒ						○	○	○	○	○						

	Error Code	Description	Error Detection Condition	Operation Result at Error Occurrence
Error detection (D9091)	10054	D9090 set value error	Invalid device defined.	Data not transferred.
	10055	Transfer word number error	1. Value specified is other than 0 to 8192. 2. If within the range 0 to 8192, value specified is outside the transfer source/destination range.	1. Data not transferred. 2. Data in the specified device range only transferred.
	10056	Source input data device code error	Invalid device code defined.	Data not transferred.
	10057	Destination input data device code error	Invalid device code defined.	Data not transferred.
	10058	Source device error	Invalid device defined.	Data not transferred.
	10059	Source offset input error	Value specified is outside the specified device range.	Data not transferred.
	10060	Destination device error	Invalid device defined.	Data not transferred.
	10061	Destination offset input error	Value specified is outside the specified device range.	Data not transferred.

FUNCTION

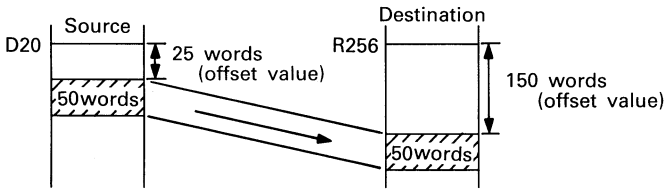
(1) Transfers the specified number of words in blocks from the source device area following the number of words specified by the offset value to the destination device area that follows the number of words specified by the offset value.

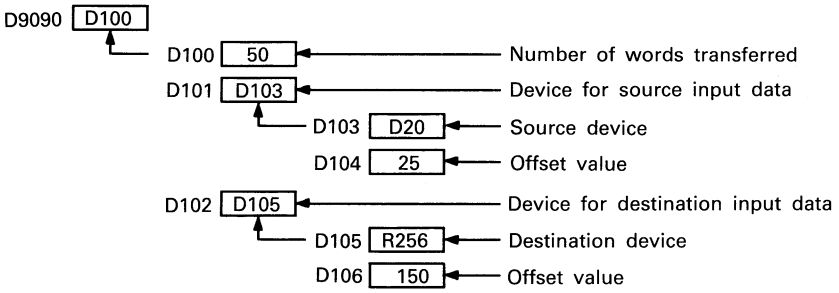


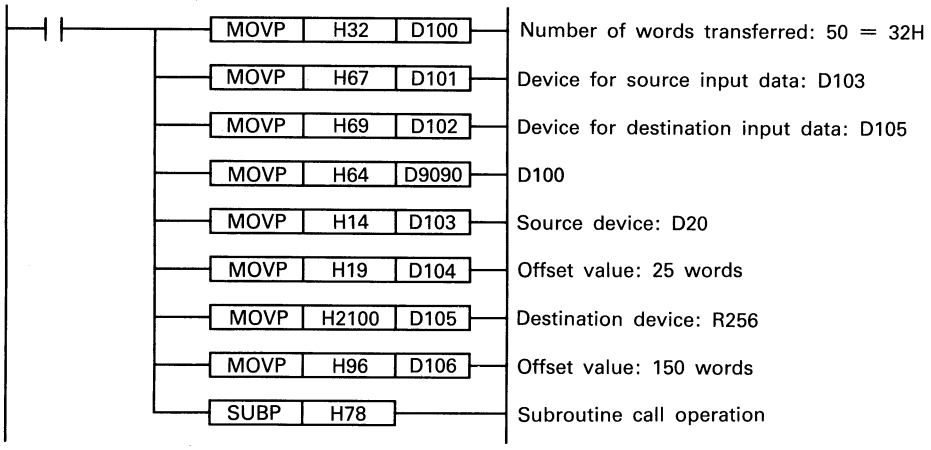
(2) The device number used as a work area must not be the same as the one storing the data.

PROGRAM EXAMPLE

The following program transfers 50 words from the source device D20 area that follows 25 words to the destination device R256 area following 150 words.

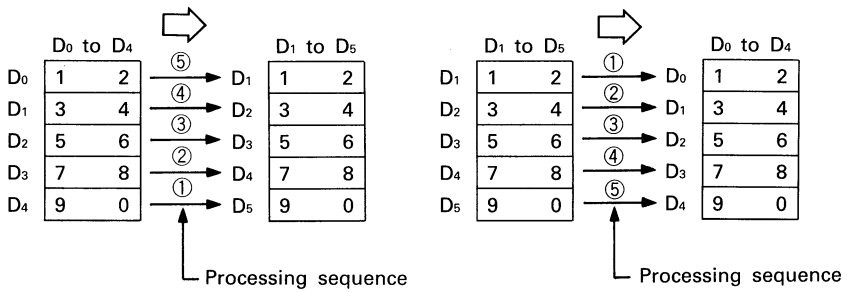






POINT

Data is not corrupted if source device numbers are repeated for destination device numbers as shown below:



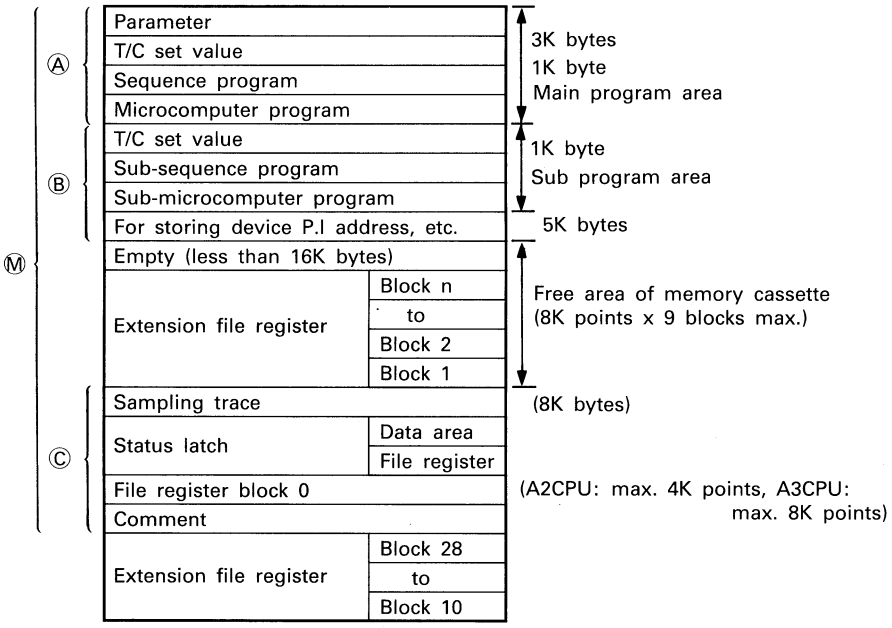
4.8 File Register Extension Commands

4.8.1 Definition

In addition to the file registers (4K points for the A2CPU, 8K points for the A3CPU), the free area of the memory cassette may be used as extension file registers (by changing the block number).

4.8.2 Assignment of block number to extension R

(1) The vacant area of the memory cassette can be defined as extension R in batches of 8K points: blocks 1 to 9 and blocks 10 to 28 Blocks 10 to 28 are automatically defined in accordance with the CPU and memory cassette used, where as blocks 1 to 9 are calculated in accordance with the CPU and memory cassette used, parameter settings, and ROM/RAM operation as indicated in (2):



(2) Calculating block number for use as extension R (blocks 1 to 9)

Calculate the free area of the memory cassette by subtracting the parameter-set capacity from the memory cassette capacity (Ⓜ: see Table 4.4.). Divide this area in increments of 8K points (16K bytes) and assign block numbers in numerical order.

RAM operation: $\frac{Ⓜ-Ⓐ-Ⓑ-Ⓒ \quad \text{K bytes}}{16} = n_1$

ROM operation: $\frac{Ⓜ-Ⓑ-Ⓒ \quad \text{K bytes}}{16} = n_2$

An integer value of n1 or n2 indicates the number of blocks used.

Example: A3MCA12 memory cassette, parameter settings as indicated below, RAM operation

Sequence program = 2K bytes, microcomputer program = 2K bytes, file register = 2K bytes
Further 4K bytes are used for parameters and T/C set values.

$$\frac{96 - (4 + 2 + 2) - 2}{16} = 5.3$$

Blocks 1 to 5 can be used as extension R.

(3) Usable extension R block numbers

Memory Cassette Type		Ⓜ	Usable Block Number (RAM/ROM operation)			Remarks (Block number without error 10064 display)
			A2N, A3NCPU	A3HCPU	A2, A3CPU	
A3NMCA-0	A3MCA-0	16K bytes	Unusable	Unusable	Unusable	
A3NMCA-2	A3MCA-2	16K bytes	Unusable	Unusable	Unusable	
A3NMCA-4	A3MCA-4	32K bytes	Block 1 only	Block 1 only	Block 1 only	
A3NMCA-8	A3MCA-8	64K bytes	Up to block 3	Up to block 3	Up to block 3	
———	A3MCA-12	96K bytes	Up to block 5	Up to block 5	Up to block 5	Blocks 10, 11
A3NMCA-16	———	96K bytes	Up to block 5 Blocks 10, 11	Up to block 5 Blocks 10, 11	Up to block 5 Blocks 10, 11	
———	A3MCA-18	144K bytes	Up to block 8	Up to block 8	Up to block 8	Blocks 10 to 28 for ANCPU, A3HCPU
A3NMCA-24	———	144K bytes	Up to block 8 Blocks 10 to 12	Up to block 8 Blocks 10 to 12	Up to block 8	Blocks 13 to 20 for ANCPU Blocks 13 to 28 for A3HCPU
A3NMCA-40	———	144K bytes	Up to block 8 Blocks 10 to 20	Up to block 8 Blocks 10 to 20	Up to block 8	Blocks 21 to 28 for A3HCPU
A3NMCA-56	———	144K bytes	Up to block 8 Blocks 10 to 20	Up to block 8 Blocks 10 to 20	Up to block 8	

Table 4.4 Usable Extension R Block Numbers

REMARKS

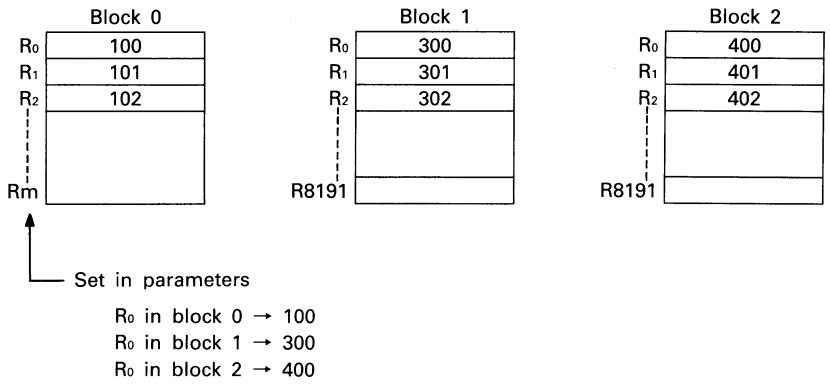
Block 9 cannot be used.

IMPORTANT

The CPU regards the A3MCA as the A3NMCA16 and regards the A3MCA18, A3NMCA24 and 40 as the A3NMCA56. Hence, an error may not be flagged if an invalid block is specified, e.g. if block 10 or 11 is specified for the A3MCA12, an error does not occur but read/write are invalid.

(4) Relation between file registers and block numbers

To access any extension file register, specify Rn (file register number) after specifying the block number. Any file register in another block cannot be accessed without specifying the corresponding block number. The block number defaults to 0.

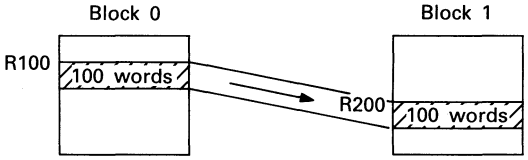


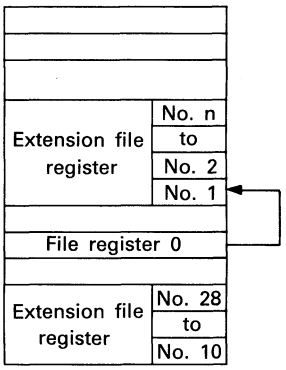
4.8.3 Commands related to file registers

(1) The following commands are available for file register block number change, data transfer and data change:

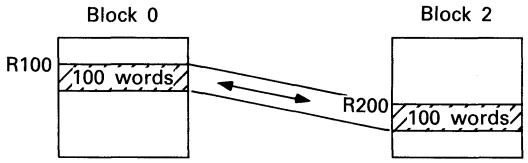
1) RSET
Changes the default file register address (block number).

2) RBMOV
Block-transfers data between file registers.





3) RBCHG
Exchanges data between file registers.



4.8.4 Notes on use of extension R

(1) Procedure
1) Set the file register capacity (block 0) in the parameter.
2) Calculate the empty memory area capacity to determine the number of blocks usable.
3) Using the RSET command, specify the block number for the extension file register used.
4) After the RSET command is executed, the other file register command accesses the block file registers specified.

(2) The capacity of file register block 0 is set in the parameter, i.e. max. 4K points for the A2CPU, max. 8K points for the A3CPU.

(3) Any external device, such as AJ71C24, or AD51E is only allowed to access file registers in block 0.
The AJ71C24-S3 and AD51E-S3 are allowed to access extension file registers by using extension file register commands.

(4) The file register facility (Section 7) allows R and extension R data to be transferred between the GPP/HGP/PHP and PC, displayed as a list, changed, stored onto disk, and printed out.

(5) Normal processing may not be performed if an invalid block number is specified.

(6) Note that when the RSET instruction is used, data is collected from the following block number by the execution of sampling trace or status latch:
• Sampling trace
Trace point setting
Step number: Block number at the execution of the END instruction
Sampling time: Block number depends on the execution timing.
• Status latch
Block number at the execution of status latch

4.8.5 RSET

Changes the file register block area.

Command	Subroutine Address	I/O Data
RSET	A-FN1 SUB | H ($A_1 + B_H$) A-MCBT SUB | H ($A_4 + 5_H$) A_1, A_4: Head address for storing subroutine	D9090 ⓐ (a + 0) ← Block number (0 to 28)

	Usable Devices														
	X	Y	M/L/S	B	F	T	C	D	W	R	A0	A1	Z	V	P	I
Device specified at D9090 ⓐ								○	○							

	Error Code	Description	Error Detection Condition	Operation Result at Error Occurrence
Error detection (D9091)	10062	D9090 set value error	Invalid device defined.	Operation not performed.
	10063	Input data error	Value set is other than 0 to 28.	Operation not performed.
	10064	Specified block not found.	Block specified is not empty.	Operation not performed.

FUNCTION

(1) Specify the file register block number to use the empty memory cassette area as extension file registers.

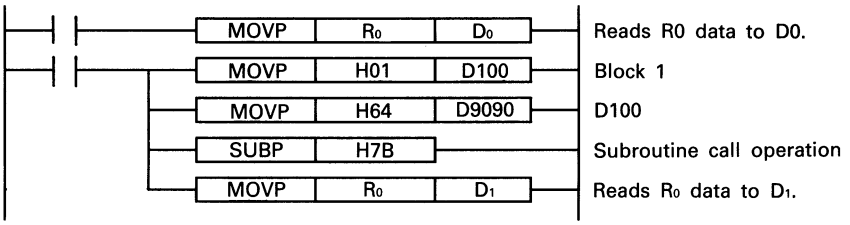
(2) Extension file register areas

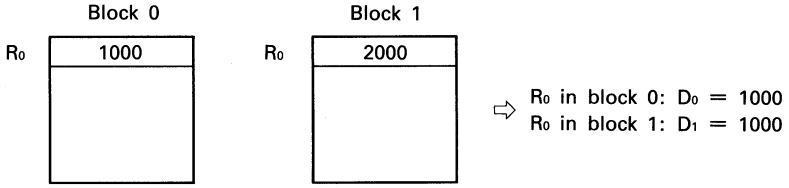
Block 0: Normal file register area
Block 1 to 28: Empty memory cassette area

(3) The block number defaults to 0.

PROGRAM EXAMPLE

The following program reads data from R0 in extension file register block 1 to D1.



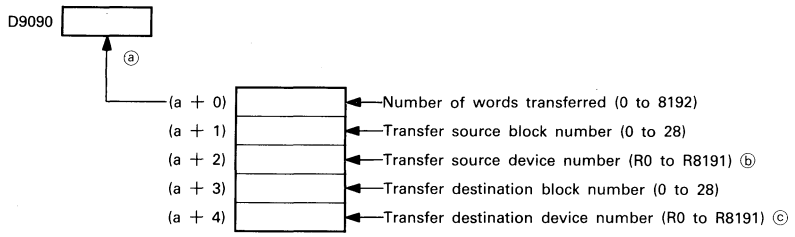


POINT

The usable block number depends on the CPU and memory cassette combination, etc.

4.8.6 RBMOV

Transfers data in blocks between file registers.

Command	Subroutine Address	I/O Data
RBMOV	A-FN1 SUB | H (A_1 + E_H) A-MCBT SUB | H (A_4 + 8_H) A_1, A_4: Head address for storing subroutine	

	Usable Devices															
	X	Y	M/L/S	B	F	T	C	D	W	R	A0	A1	Z	V	P	I
Device specified at D9090 ⓐ								○	○							
Transfer source device ⓑ										○						
Transfer destination device ⓒ										○						

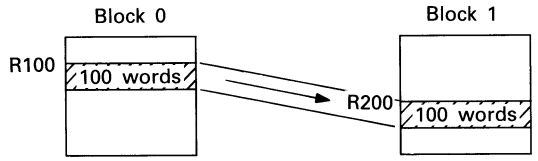
	Error Code	Description	Error Detection Condition	Operation Result at Error Occurrence
Error detection (D9091)	10065	D9090 set value error	Invalid device defined.	Data not transferred.
	10066	Transfer word number input error	1. Value specified is other than 0 to 8192. 2. Transfer source/destination range exceeded.	1. Data not transferred. 2. Data in the specified device range only transferred.
	10067	Source block number input error	Value specified is other than 0 to 28 or the block number specified does not exist.	Data not transferred.
	10068	Source device number input error	Value specified is other than R0 to R8191.	Data not transferred.
	10069	Destination block number input error	Value specified is other than 0 to 28 or the block number specified does not exist.	Data not transferred.
	10070	Destination device number input error	Value specified is other than R0 to R8191.	Data not transferred.
	10071	Specified block is memory protected.	Protected block number defined.	Data not transferred.

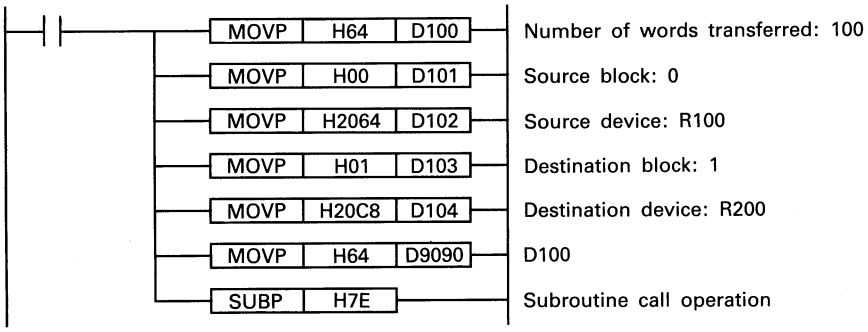
FUNCTION

(1) Transfers the specified number of words in blocks between file register/ extension file registers.
(2) Data cannot be transferred within the same block.
(3) Any source/destination block may be specified independently of the block specified by the RSET command.

PROGRAM EXAMPLE

The following program transfers 100 words from R100 in block 0 to R200 in block 1.





4.8.7 RBCHG

Exchanges data between file registers.

Command	Subroutine Address	I/O Data
RBCHG	A-FN1 SUB | H (A_1 + 38H) A-MCBT SUB | H (A_4 + EH) A_1, A_4: Head address for storing subroutine	D9090 ⓐ (a + 0) Number of words exchanged (0 to 1024) (a + 1) Exchange (1) block number (0 to 28) (a + 2) Exchange (1) device number (R0 to R8191)* (a + 3) Exchange (2) block number (0 to 28) (a + 4) Exchange (2) device number (R0 to R8191)* *Parameter-set range for block 0.

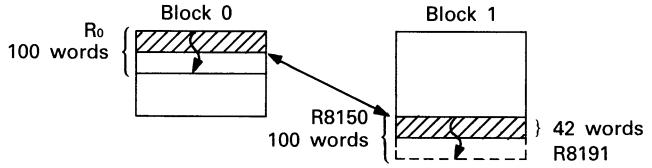
	Usable Devices															
	X	Y	M/L/S	B	F	T	C	D	W	R	A0	A1	Z	V	P	I
Device specified at D9090 ⓐ								○	○							

	Error Code	Description	Error Detection Condition	Operation Result at Error Occurrence
Error detection (D9091)	10101	D9090 set value error	Invalid device defined.	Data not exchanged.
	10102	Exchange word number input error	1. Value specified is other than 0 to 1024. 2. Device range exceeded.	1. Data not exchanged. 2. Data in the specified device range only exchanged.
	10103	Exchange (1) block number input error	Value specified is other than 0 to 28 or the block number specified does not exist.	Data not exchanged.
	10104	Exchange (1) device number input error	Value specified is other than R0 to R8191*.	Data not exchanged.
	10105	Exchange (2) block number input error	Value specified is other than 0 to 28 or the block number specified does not exist.	Data not exchanged.
	10106	Exchange (2) device number input error	Value specified is other than R0 to R8191*.	Data not exchanged.
	10107	Specified block is memory protected.	Protected block number defined.	Data not exchanged.

FUNCTION

(1) Exchanges the specified number of words in blocks between file register/ extension file registers.
(2) Data cannot be exchanged within the same block.
(3) If data exchanged exceeds the device range, only the data within the device range is exchanged.

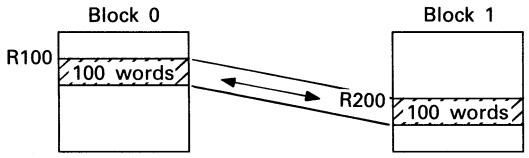
Example: Exchanging that 100 words are exchanged in blocks between R0 in block 0 and R8150 in block 1.

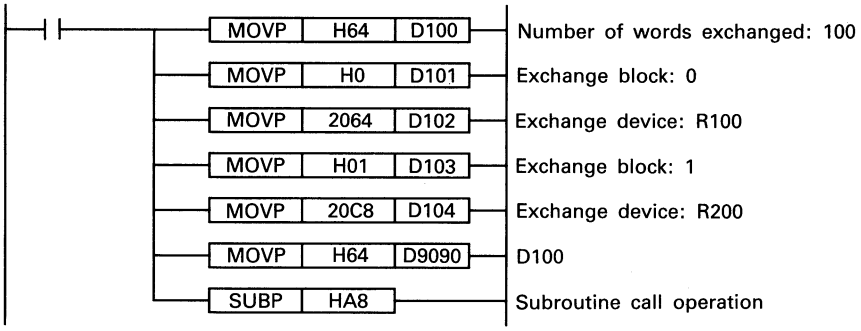


Operation results:
Error code 10102 is flagged and data within the specified device range is only exchanged as follows:
Block 0:
R0 to R41 has data transmitted from R8150 to R8191 in block 1.
R42 to R99 remain unchanged.
Block 1:
R8150 to R8191 has data transmitted from R0 to R41 in block 0.

PROGRAM EXAMPLE

The following program exchanges 100 words in blocks from R100 in block 0 to R200 in block 1.





4.9 Comment Data Transfer Command

4.9.1 COMRD

Transfers comment data.

Command	Subroutine Address	I/O Data
COMRD	A-FN1: SUB | H (A_1 + 35H) A-MCBT: SUB | H (A_4 + BH) A_1, A_4: Head address for storing subroutine	D9090 ⓐ (a + 0) ← Device number for entering comment ⓑ (a + 1) ← Main/sub designation (0/1) (a + 2) ← Transfer destination device number ⓒ

	Usable Devices														
	X	Y	M/L/S	B	F	T	C	D	W	R	A0	A1	Z	V	P	I
Device specified at D9090 ⓐ								○	○	○						
Comment entry device ⓑ	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Transfer destination device ⓒ								○	○	○						

	Error Code	Description	Error Detection Condition	Operation Result at Error Occurrence
Error detection (D9091)	10098	D9090 set value error	Invalid device defined.	Data not transferred.
	10099	Comment entry device input error	Comment device specified does not exist in the memory cassette.	Data not transferred.
	10100	Destination device input error	Invalid value specified.	Data not transferred.

FUNCTION

(1) Transfers a comment in blocks of 8 words from the specified devices in the memory cassette to the specified devices. A space code (20H) is written to each empty device.

(2) Specify main or sub program in the second word in the format above.
Main program → "0"
Sub program → "1"

PROGRAM EXAMPLE

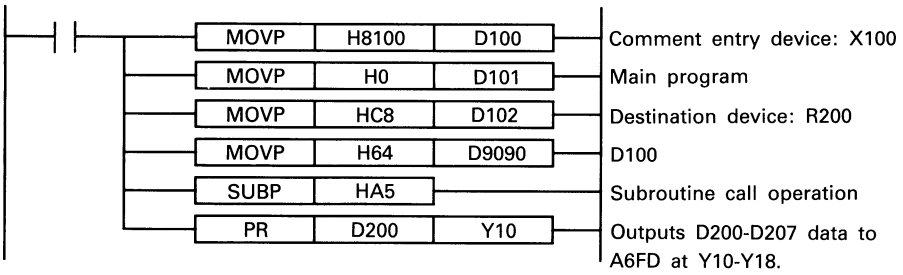
The following program transfers the comment (No. 1 LIMIT SW) from X100 to D200 and displays it on the A6FD via the output module.

D9090 [0064_H] ← D100 = 0064_H, For the setting method, see Table 4.3.

↓ Word device (D, W, R)

D100	8100_H	← Device number of X100. For the setting method, see Table 4.3.
D101	0_H	← Main program designated
D102	$C8_H$	← Device number of D200

D200	"0" $6F_H$	"N" $4E_H$	} 8 words
D201	"1" 31_H	"." $2E_H$	
D202	"I" 49_H	"L" $4C_H$	
D203	"I" 49_H	"M" AF_H	
D204	"S" 53_H	"T" 54_H	
D205	20_H	"W" 57_H	
D206	20_H	20_H	
D207	20_H	20_H	



4.10 BIN (BCD) → ASCII Conversion Commands

(1) Converts the specified data into ASCII.
(2) The following two commands are available to convert BIN data into decimal ASCII:

1-word data (−32768 to 32767) ⟶ BINDA
2-word data (−2147483648 to 2147483647) ⟶ DBINDA

Example: BINDA command

BIN value ⟶ 5-digit decimal ASCII

D100	000F$_H$		
		D101	20$_H$ 20$_H$
		D102	20$_H$ 20$_H$
		D103	35$_H$ 31$_H$

000F$_H$ = 0015

After conversion,
the data is zero suppressed.

(3) The following two commands are available to convert BIN data into hexadecimal ASCII:

1-word data (0 to FFFF) ⟶ BINHA
2-word data (0 to FFFFFFFF) ⟶ DBINHA

Example: BINHA command

BIN value ⟶ 4-digit hexadecimal ASCII

D100	000F$_H$		
		D101	30$_H$ 30$_H$
		D102	46$_H$ 20$_H$

After conversion,
the data is not zero suppressed.

(4) There are the following two commands to convert BCD data into decimal ASCII:

1-word data (0 to 9999) ⟶ BCDDA
2-word data (0 to 99999999) ⟶ DBCDDA

Example: BCDDA command

BCD value ⟶ 4-digit decimal ASCII

D100	0015		
		D101	20$_H$ 20$_H$
		D102	35$_H$ 31$_H$

After conversion,
the data is zero suppressed.

POINT

(1) Zero suppression indicates the elimination before printing of non-significant zeros, e.g. those to the left of significant digits.

Example: H0001
- **Zero suppressed → ␣␣␣1** (**Space**)
- **Not zero suppressed → 0001**

(2) Any BCD value displayed with H can be monitored as it is on the GPP, etc.

4.10.1 BINDA

Converts a 1-word BIN value into a 5-digit decimal ASCII (−32768 to 32767).

Command	Subroutine Address	I/O Data
BINDA	A-FN1: SUB | H (A_1 + 11H) A-ASBS: SUB | H (A_5 + 5H) A_1, A_5: Head address for storing subroutine	D9090 ⓐ (a + 0) Device for input data (BIN value) (−32768 to 32767) ⓑ (a + 1) 10^4 | Sign* (a + 2) 10^2 | 10^3 (a + 3) 10^0 | 10^1 Devices for converted data (ASCII) ⓒ *: 20H (SP) for positive value. 2DH (−) for negative value.

	Usable Devices															
	X	Y	M/L/S	B	F	T	C	D	W	R	A0	A1	Z	V	P	I
Device specified at D9090 ⓐ								○	○	○						
Input data device ⓑ								○	○	○						
Converted data device ⓒ								○	○	○						

	Error Code	Description	Error Detection Condition	Operation Result at Error Occurrence
Error detection (D9091)	10076	D9090 set value error	Invalid device defined.	Operation not performed.

FUNCTION

(1) Regards the specified device data as a BIN value and converts it into a 5-digit decimal ASCII code.
(2) The input data range is 16 bits. (K-32768 to 32767)
(3) The converted data is zero suppressed.

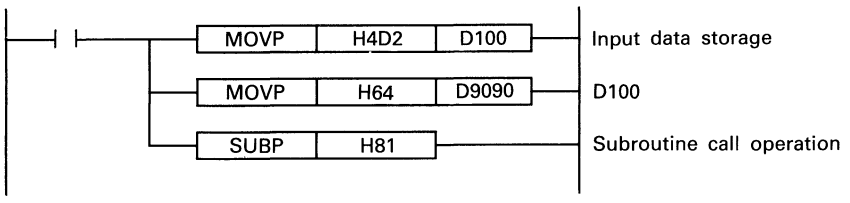
PROGRAM EXAMPLE

The following program converts 04D2H of D100 into an ASCII code.

D9090 | 0064H | ← D100 = 0064H, For the setting method, see Table 4.3.

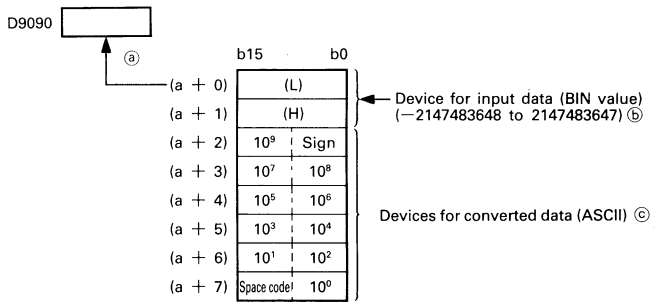
↓ Word device (D, W, R)

D100	04D2H	The 8 high-order bits of D101 are zero suppressed and change to the space code (20H).
D101	20H | 20H	
D102	32H | 31H	Data stored is monitored by GPP, etc. as shown on the left.
D103	34H | 33H	



4.10.2 DBINDA

Converts a 2-word BIN value into a 10-digit decimal ASCII (−2147483648 to 2147483647).

Command	Subroutine Address	I/O Data
DBINDA	A-FN1: SUB H (A_1 + 23H) A-MSBD: SUB H (A_6 + 8H) A_1, A_6: Head address for storing subroutine	

	Usable Devices															
	X	Y	M/L/S	B	F	T	C	D	W	R	A0	A1	Z	V	P	I
Device specified at D9090 ⓐ								○	○	○						
Input data device ⓑ								○	○	○						
Converted data device ⓒ								○	○	○						

	Error Code	Description	Error Detection Condition	Operation Result at Error Occurrence
Error detection (D9091)	10087	D9090 set value error	Invalid device defined.	Operation not performed.

FUNCTION

(1) Regards the specified device 2-word data as a BIN value and converts it into a 10-digit decimal ASCII code.
(2) The input data range is 32 bits (K−2147483648 to 2147483647).
(3) The converted data is zero suppressed.

PROGRAM EXAMPLE

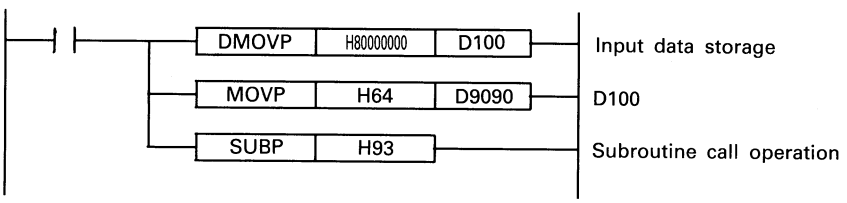
The following program converts data 80000000H of D100 into an ASCII code.

D9090 [0064H] ← D100 = 0064H, For the setting method, see Table 4.3.

↓ Word device (D, W, R)

D100	0000H	
D101	8000H	
D102	32H	2DH
D103	34H	31H
D104	34H	37H
D105	33H	38H
D106	34H	36H
D107	20H	38H

Data stored is monitored by GPP, etc. as shown on the left.



4.10.3 BINHA

Converts a 1-word BIN value into a 4-digit hexadecimal ASCII (0 to FFFF$_H$).

Command	Subroutine Address	I/O Data
BINHA	A-FN1: SUB | H (A_1 + 14$_H$) A-ASBS: SUB | H (A_5 + 8$_H$) A_1, A_5: Head address for storing subroutine	D9090 ⓐ b_{15} b_0 (a + 0) ← Device for input data (BIN value) (0 to FFFFH) ⓑ (a + 1) 16^2 | 16^3 (a + 2) 16^0 | 16^1 } Devices for converted data (ASCII) ⓒ

	Usable Devices															
	X	Y	M/L/S	B	F	T	C	D	W	R	A0	A1	Z	V	P	I
Device specified at D9090 ⓐ								○	○	○						
Input data device ⓑ								○	○	○						
Converted data device ⓒ								○	○	○						

	Error Code	Description	Error Detection Condition	Operation Result at Error Occurrence
Error detection (D9091)	10077	D9090 set value error	Invalid device defined.	Operation not performed.

FUNCTION

(1) Regards the specified device data as a BIN value and converts it into a 4-digit hexadecimal ASCII code.
(2) The input data range is 16 bits (0 to FFFF$_H$).
(3) The converted data is not zero suppressed.

PROGRAM EXAMPLE

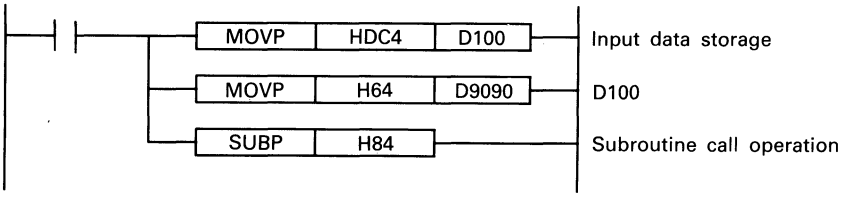
The following program converts 0DC4H of D100 into an ASCII code.

D9090 [0064$_H$] ← D100 = 0064$_H$, For the setting method, see Table 4.3.

↓ Word device (D, W, R)

D100	0DC4$_H$	
D101	44$_H$	30$_H$
D102	34$_H$	43$_H$

} Data stored is monitored by GPP, etc. as shown on the left.



4.10.4 DBINHA

Converts a 2-word BIN value into an 8-digit hexadecimal ASCII (0 to FFFFFFFFH).

Command	Subroutine Address	I/O Data
DBINHA	A-FN1: SUB | H (A1 + 26H) A-ASBD: SUB | H (A6 + 8H) A1, A6: Head address for storing subroutine	D9090 ⓐ (a + 0) (L) (a + 1) (H) (a + 2) 16^6 | 16^7 (a + 3) 16^4 | 16^5 (a + 4) 16^2 | 16^3 (a + 5) 16^0 | 16^1 b15 … b0 Device for input data (BIN value) (0 to FFFFFFFFH) ⓑ Devices for converted data (ASCII) ⓒ

	Usable Devices															
	X	Y	M/L/S	B	F	T	C	D	W	R	A0	A1	Z	V	P	I
Device specified at D9090 ⓐ								○	○	○						
Input data device ⓑ								○	○	○						
Converted data device ⓒ								○	○	○						

	Error Code	Description	Error Detection Condition	Operation Result at Error Occurrence
Error detection (D9091)	10088	D9090 set value error	Invalid device defined.	Operation not performed.

FUNCTION

(1) Regards the specified device 2-word data as a BIN value and converts it into an 8-digit hexadecimal ASCII code.
(2) The input data range is 32 bits (0 to FFFFFFFFH).
(3) The converted data is not zero suppressed.

PROGRAM EXAMPLE

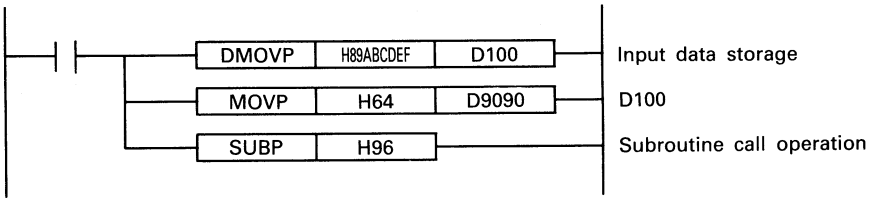
The following program converts 89ABCDEFH of D100 into an ASCII code.

D9090 | 0064H | ← D100 = 0064H, For the setting method, see Table 4.3.

↓ Word device (D, W, R)

D100	CDEFH	
D101	89ABH	
D102	39H	38H
D103	42H	41H
D104	44H	43H
D105	46H	45H

Data stored is monitored by GPP, etc. as shown on the left.



4.10.5 BCDDA

Converts a 4-digit BCD value into a 4-digit decimal ASCII (0 to 9999).

Command	Subroutine Address	I/O Data
BCDDA	A-FN1: SUB | H (A_1 + 17H) A-ASBS: SUB | H (A_5 + BH) A_1, A_5: Head address for storing subroutine	D9090 ⓐ (a + 0) b_{15} … b_0 ← Device for input data (BCD value) (0 to 9999) ⓑ (a + 1) 10^2 | 10^3 (a + 2) 10^0 | 10^1 } Devices for converted data (ASCII) ⓒ

	Usable Devices															
	X	Y	M/L/S	B	F	T	C	D	W	R	A0	A1	Z	V	P	I
Device specified at D9090 ⓐ								○	○	○						
Input data device ⓑ								○	○	○						
Converted data device ⓒ								○	○	○						

	Error Code	Description	Error Detection Condition	Operation Result at Error Occurrence
Error detection (D9091)	10078	D9090 set value error	Invalid device defined.	Operation not performed.
	10079	Input data error	Value specified is other than 0 to 9.	Operation not performed.

FUNCTION

(1) Regards the specified device data as a BCD value and converts it into a 4-digit decimal ASCII code.
(2) The input data range is 16 bits (0 to 9999).
(3) 0 to 9 may be used in the 4 digits of the BCD value.
(4) The converted data is zero suppressed.

PROGRAM EXAMPLE

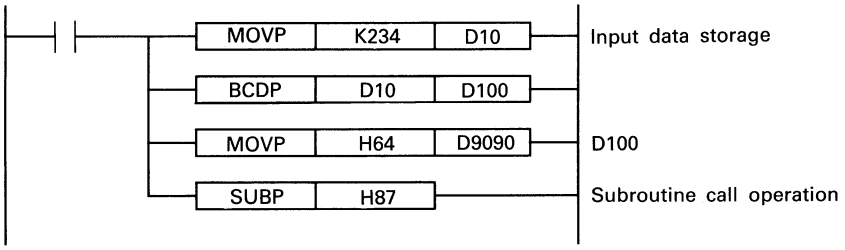
The following program converts the BCD value 0234 of D100 into an ASCII code.

D9090 [0064H] ← D100 = 0064H, For the setting method, see Table 4.3.

↓ Word device (D, W, R)

D100	0234H	
D101	32H	20H
D102	34H	33H

Data stored is monitored by GPP, etc. as shown on the left.



4.10.6 DBCDDA

Converts an 8-digit BCD value into an 8-digit decimal ASCII (0 to 99999999).

Command	Subroutine Address	I/O Data
DBCDDA	A-FN1 SUB | H (A_1 + 29H) A-ASBD SUB | H (A_6 + BH) A_1, A_6: Head address for storing subroutine	D9090 ⓐ b_{15} … b_0 (a + 0) (L) (a + 1) (H) } Device for input data (BCD value) (0 to 99999999) ⓑ (a + 2) 16^6 | 16^7 (a + 3) 16^4 | 16^5 (a + 4) 16^2 | 16^3 (a + 5) 16^0 | 16^1 } Devices for converted data (ASCII) ⓒ

	Usable Devices															
	X	Y	M/L/S	B	F	T	C	D	W	R	A0	A1	Z	V	P	I
Device specified at D9090 ⓐ								○	○	○						
Input data device ⓑ								○	○	○						
Converted data device ⓒ								○	○	○						

	Error Code	Description	Error Detection Condition	Operation Result at Error Occurrence
Error detection (D9091)	10089	D9090 set value error	Invalid device defined.	Operation not performed.
	10090	Input data error	Value specified is other than 0 to 9.	Operation not performed.

FUNCTION

(1) Regards the specified device 2-word data as a BCD value and converts it into an 8-digit decimal ASCII code.
(2) The input data range is 32 bits (0 to 99999999).
(3) 0 to 9 may be used in the 8 digits of the BCD value.
(4) The converted data is zero suppressed.

PROGRAM EXAMPLE

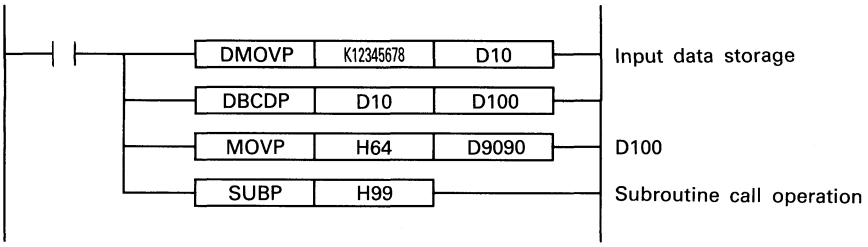
The following program converts the BCD value 12345678 of D100 into an ASCII code.

D9090 [0064H] ← D100 = 0064H, For the setting method, see Table 4.3.

↓ Word device (D, W, R)

D100	5678H	
D101	1234H	
D102	32H	31H
D103	34H	33H
D104	36H	35H
D105	38H	37H

} Data stored is monitored by GPP, etc. as shown on the left.



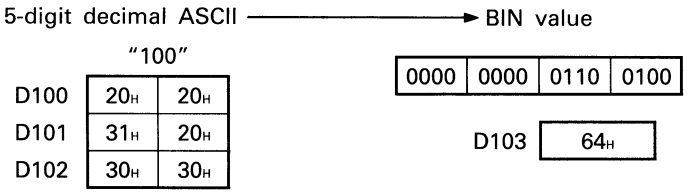
4.11 ASCII → BIN (BCD) Conversion Commands

(1) Converts the specified ASCII data into a BIN (BCD) value.

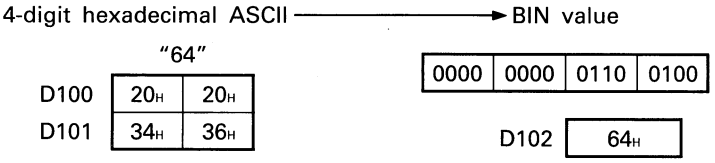
(2) The following commands are available to convert ASCII data into a BIN value:

5-digit decimal ASCII (−32768 to 32767) → ABCDBIN
4-digit hexadecimal ASCII (0 to FFFF$_H$) → ABIN
10-digit decimal ASCII (−2147483648 to 2147483647) → DABCDBIN
8-digit hexadecimal ASCII (0 to FFFFFFFF$_H$) → DABIN

Example: ABCDBIN command



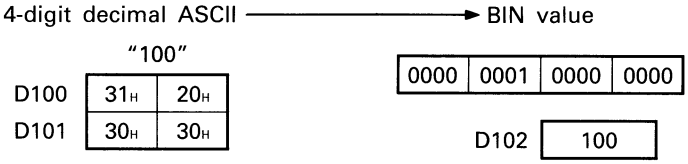
Example: ABIN command



(3) The following commands are available to convert ASCII data into a BCD value:

4-digit decimal ASCII (0 to 9999) → ABCD
8-digit decimal ASCII (0 to 99999999) → DABCD

Example: ABCD command

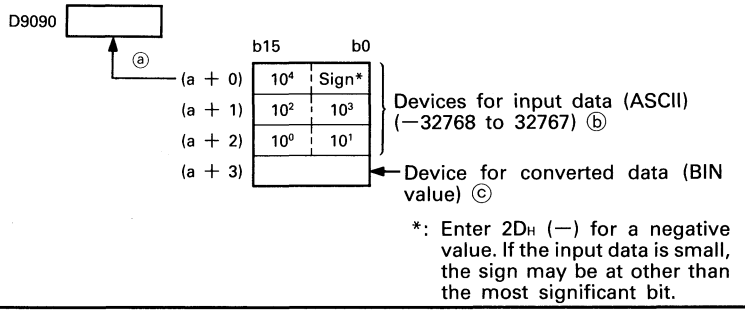


(4) 20 (space), 00 (null) in the input ASCII data are regarded as 30 ("0") at any position.

Example: Data input for ABCDBIN command

4.11.1 ABCDBIN

Converts a 5-digit decimal ASCII (−32768 to 32767) into a 1-word BIN value.

Command	Subroutine Address	I/O Data
ABCDBIN	A-FN1 SUB | H (A_1+1A_H) A-ASBS SUB | H (A_5+E$_H$) A_1, A_5: Head address for storing subroutine	

	Usable Devices															
	X	Y	M/L/S	B	F	T	C	D	W	R	A0	A1	Z	V	P	I
Device specified at D9090 ⓐ								○	○	○						
Input data devices ⓑ								○	○	○						
Converted data device ⓒ								○	○	○						

	Error Code	Description	Error Detection Condition	Operation Result at Error Occurrence
Error detection (D9091)	10080	D9090 set value error	Invalid device defined.	Operation not performed.
	10081	Input data error	Value specified is other than 0 to 9, "−".	Operation not performed.
	10082	Input data range exceeded	Value specified is outside the range −32768 to 32767.	Operation not performed.

FUNCTION

(1) Converts a 5-digit decimal ASCII code of the specified device into a BIN value.

(2) The input data range is 16 bits. (K−32768 to 32767)

(3) 20$_H$ (space), 00$_H$ (null) in the input data are regarded as 30$_H$ ("0") at any position.

(4) Any sign other than 2D$_H$ ("−") is regarded as positive. Entering 2B$_H$ ("+") results in an error.

PROGRAM EXAMPLE

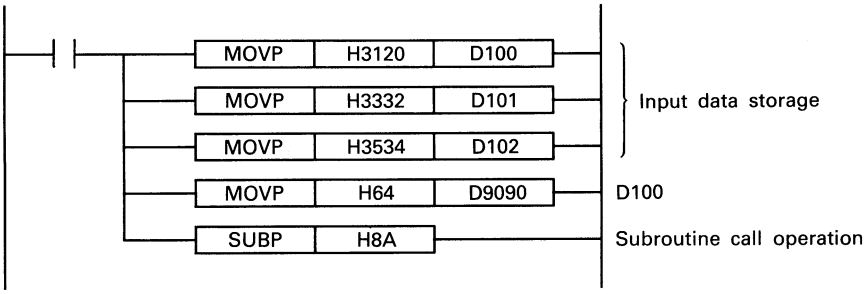
The following program converts ASCII data 12345 into a BIN value.

D9090 [0064H] ← D100 = 0064H, For the setting method, see Table 4.3.

↓ Word device (D, W, R)

D100	31H	20H
D101	33H	32H
D102	35H	34H
D103	3039H	

Data stored is monitored by GPP, etc. as shown on the left.



4.11.2 DABCDBIN

Converts a 10-digit decimal ASCII (−2147483648 to 2147483647) into a 2-word BIN value.

Command	Subroutine Address	I/O Data
DABCDBIN	A-FN1: SUB H (A_1+2C$_H$) A-ASBD: SUB H (A_6+E$_H$) A_1, A_6: Head address for storing subroutine	D9090 ⓐ (a + 0) 10^9 Sign (a + 1) 10^7 10^8 (a + 2) 10^5 10^6 (a + 3) 10^3 10^4 (a + 4) 10^1 10^2 (a + 5) Space code 10^0 Devices for input data (ASCII) (−2147483648 to 2147483647) ⓑ (a + 6) (L) (a + 7) (H) Device for converted data (BIN value) ⓒ

	Usable Devices															
	X	Y	M/L/S	B	F	T	C	D	W	R	A0	A1	Z	V	P	I
Device specified at D9090 ⓐ								○	○	○						
Input data devices ⓑ								○	○	○						
Converted data device ⓒ								○	○	○						

	Error Code	Description	Error Detection Condition	Operation Result at Error Occurrence
Error detection (D9091)	10091	D9090 set value error	Invalid device defined.	Operation not performed.
	10092	Input data error	Value specified is other than 0 to 9, "−".	Operation not performed.
	10093	Input data range exceeded	Value specified is outside the range −2147483648 to 2147483647.	Operation not performed.

FUNCTION

(1) Converts a 10-digit decimal ASCII code of the specified device into a BIN value.

(2) The input data range is 32 bits. (K−2147483648 to 2147483647)

(3) 20$_H$ (space), 00$_H$ (null) in the input data are regarded as 30$_H$ ("0") at any position.

PROGRAM EXAMPLE

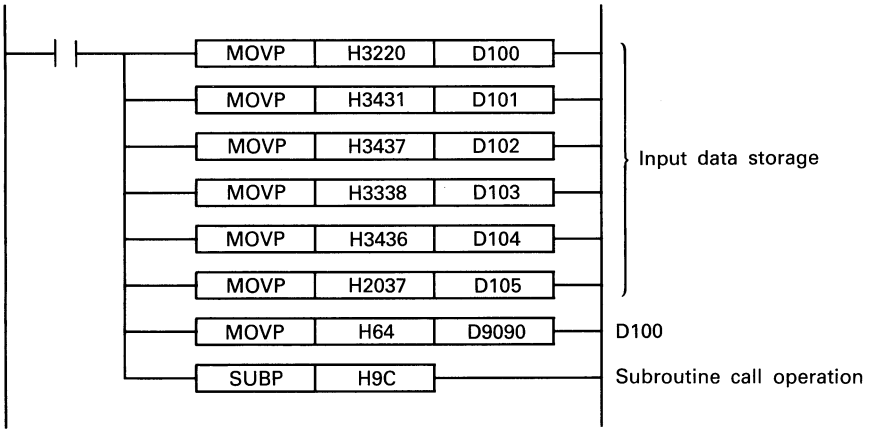
The following program converts ASCII data 2147483647 into a BIN value.

D9090 [0064H] ← D100 = 0064H, For the setting method, see Table 4.3.

↓ Word device (D, W, R)

D100	32H	20H
D101	34H	31H
D102	34H	37H
D103	33H	38H
D104	34H	36H
D105	20H	37H
D106	FFFFH	
D107	7FFFH	

Data stored is monitored by GPP, etc. as shown on the left.



4.11.3 ABIN

Converts a 4-digit hexadecimal ASCII (0 to FFFFH) into a 1-word BIN value.

Command	Subroutine Address	I/O Data
ABIN	A-FN1: SUB H (A1+1DH) A-ASBS: SUB H (A5+11H) A1, A5: Head address for storing subroutine	D9090 ⓐ b15 b0 (a + 0) 16² 16³ (a + 1) 16⁰ 16¹ (a + 2) Devices for input data (ASCII) (0 to FFFFH) ⓑ Device for converted data (BIN value) ⓒ

	Usable Devices															
	X	Y	M/L/S	B	F	T	C	D	W	R	A0	A1	Z	V	P	I
Device specified at D9090 ⓐ								○	○	○						
Input data devices ⓑ								○	○	○						
Converted data device ⓒ								○	○	○						

	Error Code	Description	Error Detection Condition	Operation Result at Error Occurrence
Error detection (D9091)	10083	D9090 set value error	Invalid device defined.	Operation not performed.
	10084	Input data error	Value specified is other than 0 to 9, A to F.	Operation not performed.

FUNCTION

(1) Converts a 4-digit hexadecimal ASCII code of the specified device into a BIN value.

(2) The input data range is 16 bits. (0 to FFFFH)

(3) 20H (space), 00H (null) in the input data are regarded as 30H ("0") at any position.

PROGRAM EXAMPLE

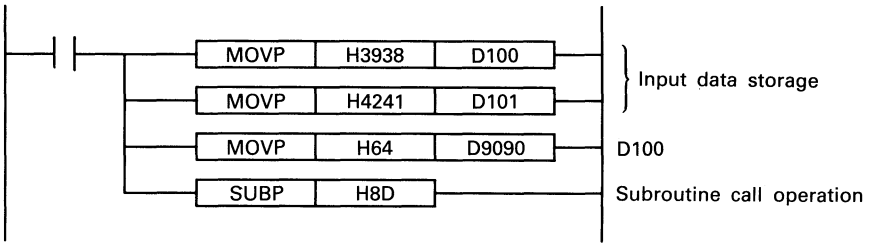
The following program converts HEX display data 89ABH of D100 into a BIN value.

D9090 [0064H] ← D100 = 0064H, For the setting method, see Table 4.3.

↓ Word device (D, W, R)

D100	39H	38H
D101	42H	41H
D102	89ABH	

Data stored is monitored by GPP, etc. as shown on the left.



4.11.4 DABIN

Converts an 8-digit hexadecimal ASCII (0 to FFFFFFFF$_H$) into a 2-word BIN value.

Command	Subroutine Address	I/O Data
DABIN	A-FN1: SUB H (A_1+2F$_H$) A-ASBD: SUB H (A_6+11$_H$) A_1, A_6: Head address for storing subroutine	D9090 ⓐ (a + 0) 16^6 | 16^7 (a + 1) 16^4 | 16^5 (a + 2) 16^2 | 16^3 (a + 3) 16^0 | 16^1 } Devices for input data (ASCII) (0 to FFFFFFFF$_H$) ⓑ (a + 4) (L) (a + 5) (H) } Device for converted data (BIN value) ⓒ

	Usable Devices															
	X	Y	M/L/S	B	F	T	C	D	W	R	A0	A1	Z	V	P	I
Device specified at D9090 ⓐ								○	○	○						
Input data devices ⓑ								○	○	○						
Converted data device ⓒ								○	○	○						

	Error Code	Description	Error Detection Condition	Operation Result at Error Occurrence
Error detection (D9091)	10094	D9090 set value error	Invalid device defined.	Operation not performed.
	10095	Input data error	Value specified is other than 0 to 9, A to F.	Operation not performed.

FUNCTION

(1) Converts an 8-digit hexadecimal ASCII code of the specified device into a BIN value.

(2) The input data range is 32 bits. (0 to FFFFFFFF$_H$)

(3) 20$_H$ (space), 00$_H$ (null) in the input data are regarded as 30$_H$ ("0") at any position.

PROGRAM EXAMPLE

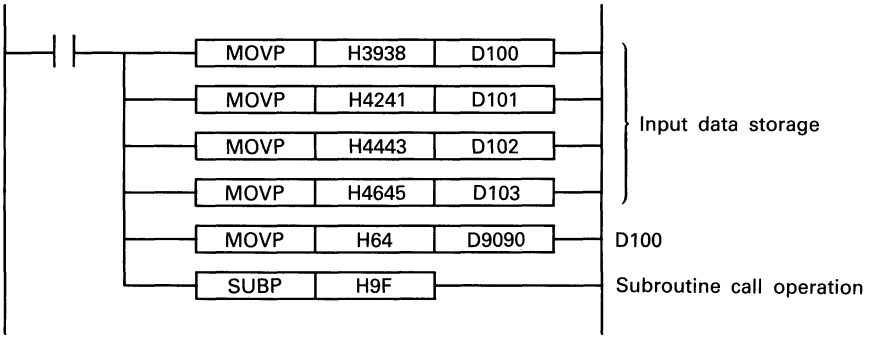
The following program converts HEX display data 89ABCDEF$_H$ of D100 into a BIN value.

D9090 [0064$_H$] ← D100 = 0064$_H$, For the setting method, see Table 4.3.

↓ Word device (D, W, R)

D100	39$_H$	38$_H$
D101	42$_H$	41$_H$
D102	44$_H$	43$_H$
D103	46$_H$	45$_H$
D104	CDEF$_H$	
D105	89AB$_H$	

Data stored is monitored by GPP, etc. as shown on the left.



4

4.11.5 ABCD

Converts a 4-digit decimal ASCII (0 to 9999) into a 4-digit BCD value.

Command	Subroutine Address	I/O Data
ABCD	A-FN1: SUB H (A_1+20_H) A-BTST: SUB H (A_5+14_H) A_1, A_5: Head address for storing subroutine	D9090 ⓐ (a + 0) 10^2 10^3 (b15 b0) (a + 1) 10^0 10^1 (a + 2) Devices for input data (ASCII) (0 to 9999) ⓑ Device for converted data (BCD value) ⓒ

	Usable Devices															
	X	Y	M/L/S	B	F	T	C	D	W	R	A0	A1	Z	V	P	I
Device specified at D9090 ⓐ								○	○	○						
Input data devices ⓑ								○	○	○						
Converted data device ⓒ								○	○	○						

	Error Code	Description	Error Detection Condition	Operation Result at Error Occurrence
Error detection (D9091)	10085	D9090 set value error	Invalid device defined.	Operation not performed.
	10086	Input data error	Value specified is other than 0 to 9.	Operation not performed.

FUNCTION

(1) Converts a 4-digit decimal ASCII code of the specified device into a BCD value.

(2) The input data range is 16 bits. (0 to 9999)

(3) 20_H (space), 00_H (null) in the input data are regarded as 30_H ("0") at any position.

PROGRAM EXAMPLE

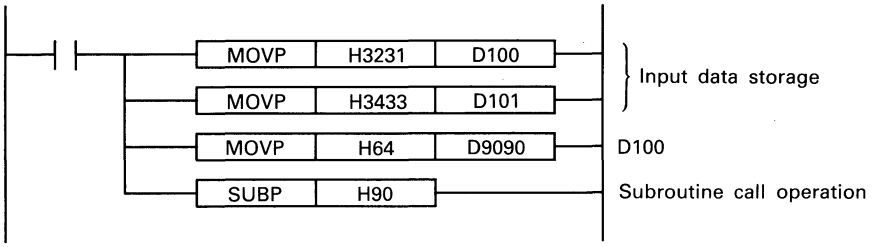
The following program converts ASCII data 1234 of D100 into a BCD value.

D9090 [0064_H] ← D100 = 0064_H, For the setting method, see Table 4.3.

↓ Word device (D, W, R)

D100	32_H	31_H
D101	34_H	33_H
D102	1234_H	

Data stored is monitored by GPP, etc. as shown on the left.



4.11.6 DABCD

Converts an 8-digit decimal ASCII (0 to 99999999) into an 8-digit BCD value.

Command	Subroutine Address	I/O Data
DABCD	A-FN1: SUB H (A_1+32_H) A-ASBD: SUB H (A_6+14_H) A_1, A_6: Head address for storing subroutine	D9090 ⓐ b15 b0 (a + 0) 10^6 10^7 (a + 1) 10^4 10^5 (a + 2) 10^2 10^3 (a + 3) 10^0 10^1 (a + 4) (L) (a + 5) (H) Devices for input data (ASCII) (0 to 99999999) ⓑ Device for converted data (BCD value) ⓒ

	Usable Devices															
	X	Y	M/L/S	B	F	T	C	D	W	R	A0	A1	Z	V	P	I
Device specified at D9090 ⓐ								○	○	○						
Input data devices ⓑ								○	○	○						
Converted data device ⓒ								○	○	○						

	Error Code	Description	Error Detection Condition	Operation Result at Error Occurrence
Error detection (D9091)	10096	D9090 set value error	Invalid device defined.	Operation not performed.
	10097	Input data error	Value specified is other than 0 to 9.	Operation not performed.

FUNCTION

(1) Converts an 8-digit decimal ASCII code of the specified device into a BCD value.

(2) The input data range is 32 bits. (0 to 99999999)

(3) 20_H (space), 00_H (null) in the input data are regarded as 30_H ("0") at any position.

PROGRAM EXAMPLE

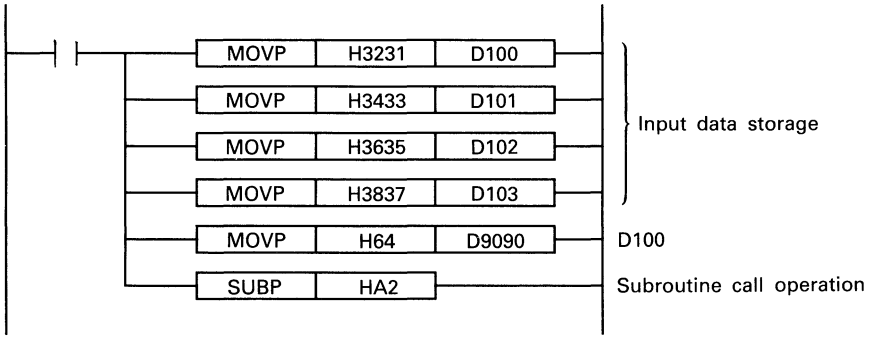
The following program converts ASCII data 12345678 of D100 into a BCD value.

D9090 [0064_H] ← D100 = 0064_H, For the setting method, see Table 4.3.

↓ Word device (D, W, R)

D100	32_H	31_H
D101	34_H	33_H
D102	36_H	35_H
D103	38_H	37_H
D104	5678_H	
D105	1234_H	

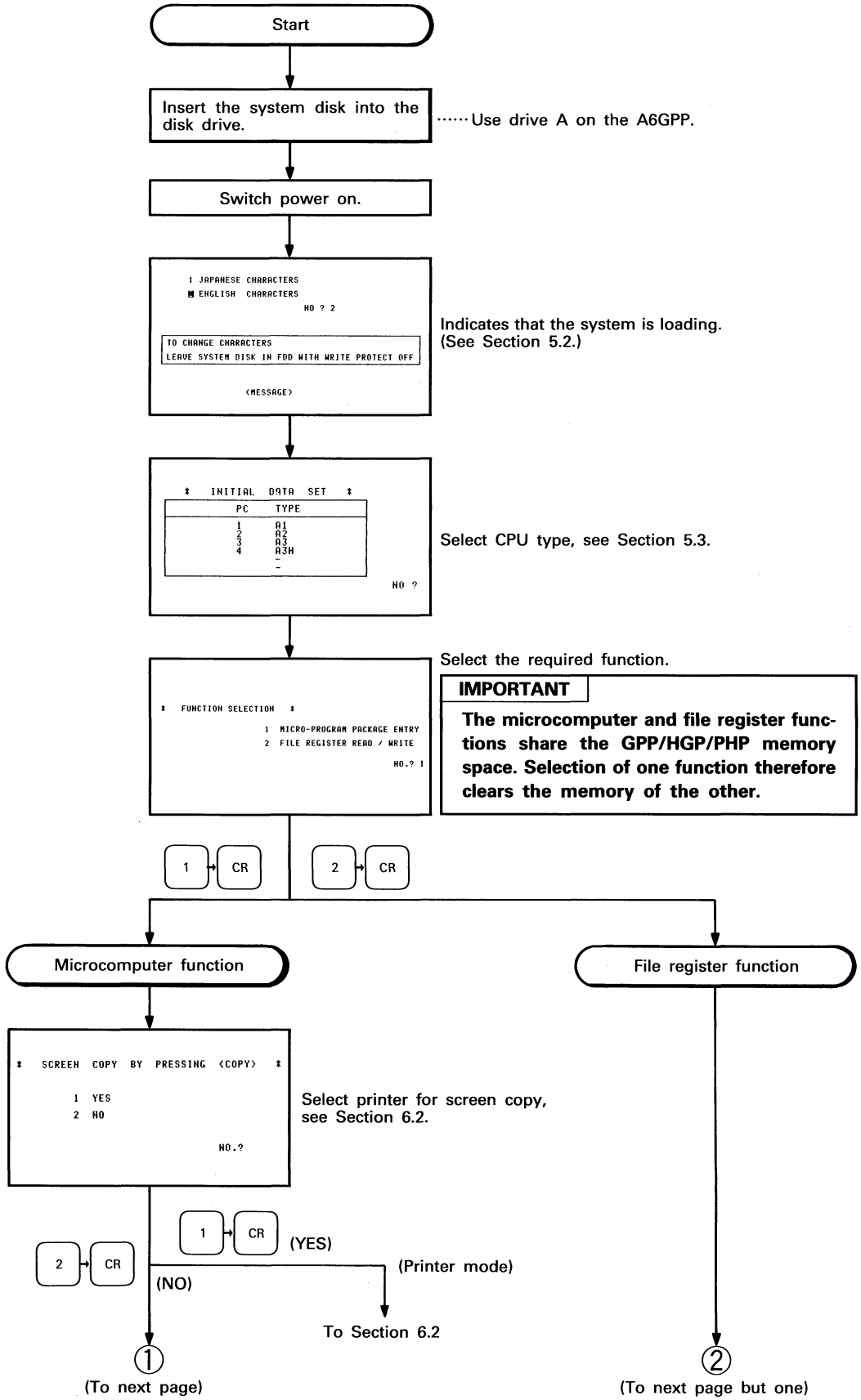
} Data stored is monitored by GPP, etc. as shown on the left.

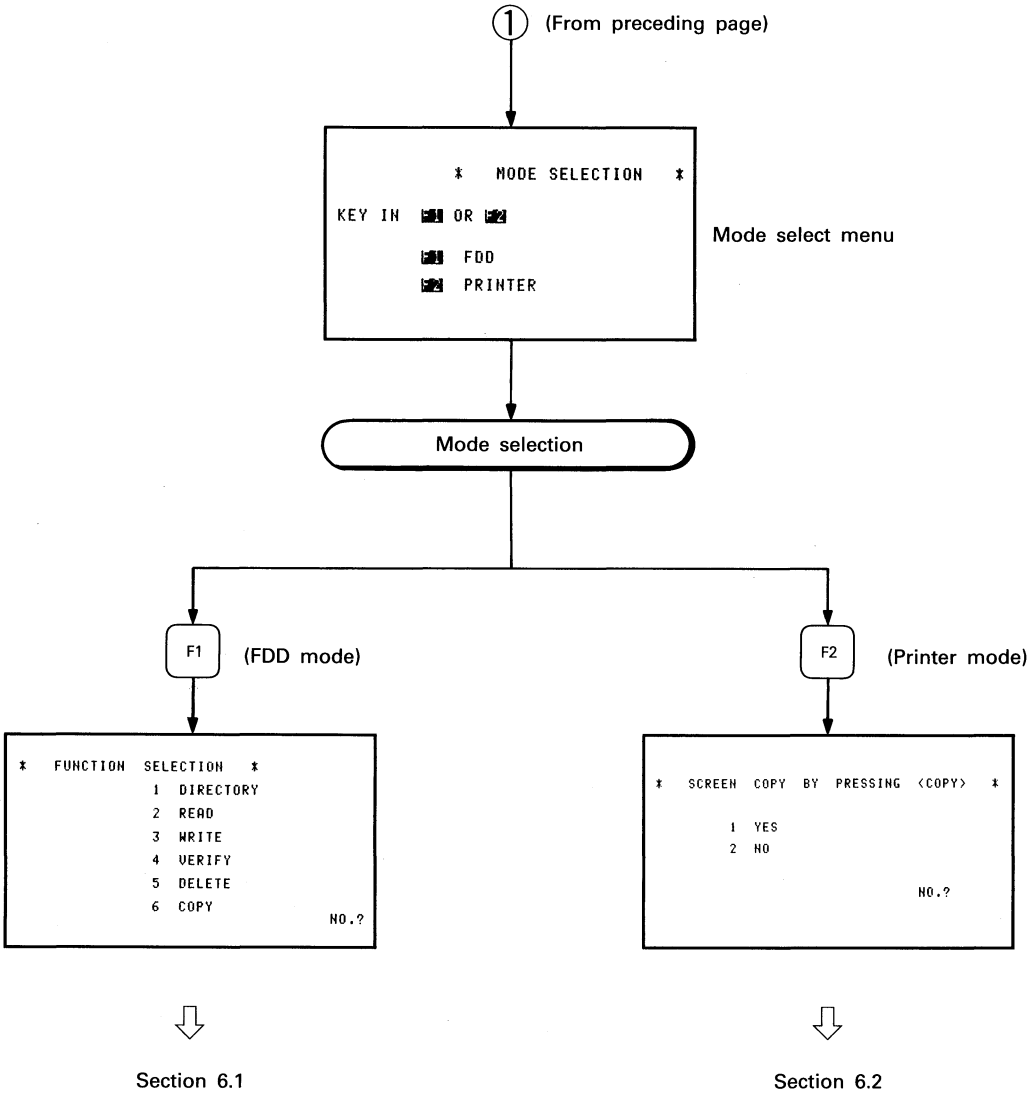


4

5. SYSTEM START-UP PROCEDURE

5.1 UTLP-FN1 Starting Procedure





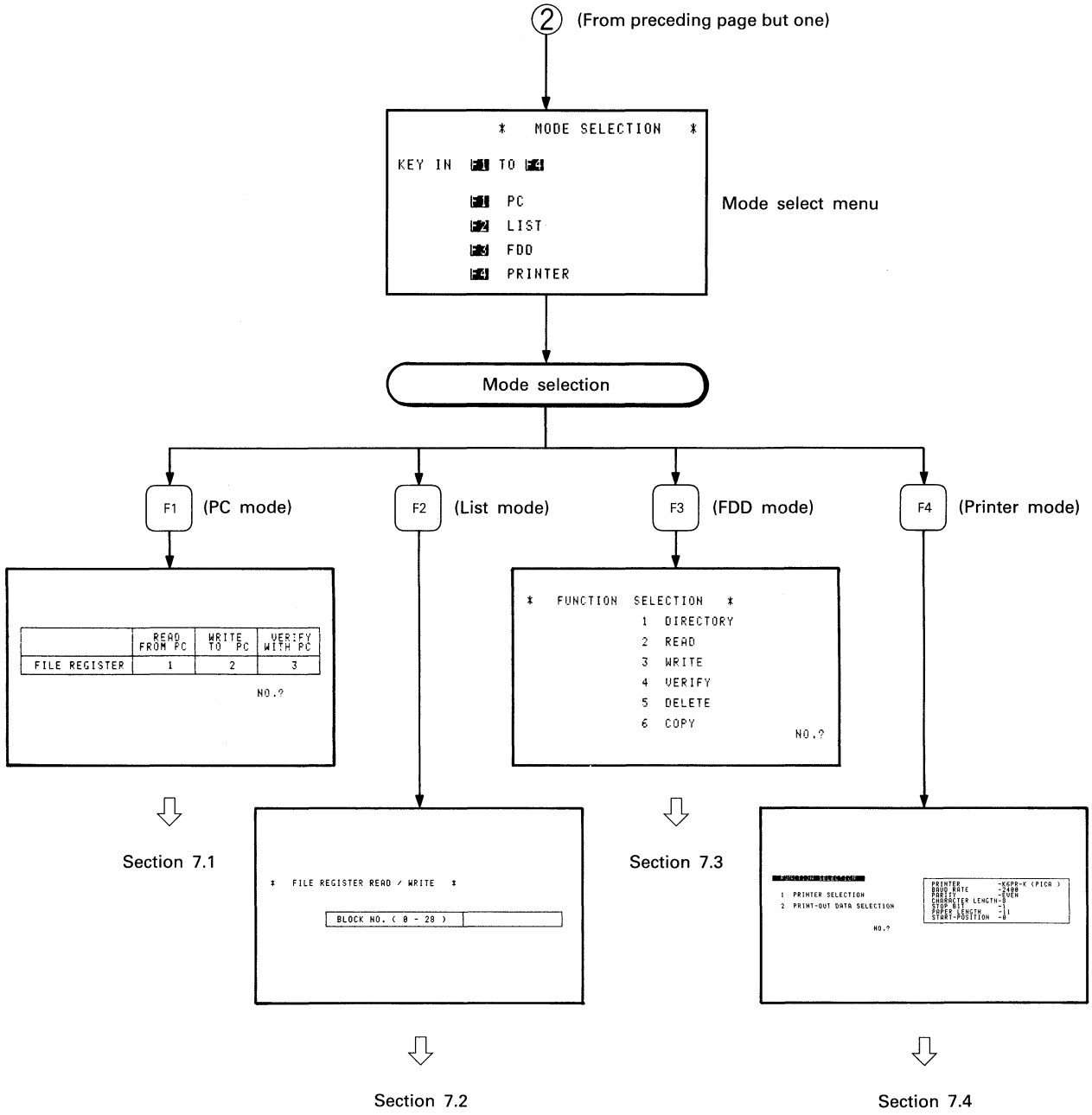
After the initial settings have been made, the mode select menu is displayed. The general functions of each mode are as follows:

FDD mode: To add the utility programs to the user program on disk.

Printer mode: To select the type of printer used for taking screen copies.

POINT

Use the upper, MELSAP, keyboard.



5

After the initial settings have been made, the mode select menu is displayed. The general functions of each mode are as follows:

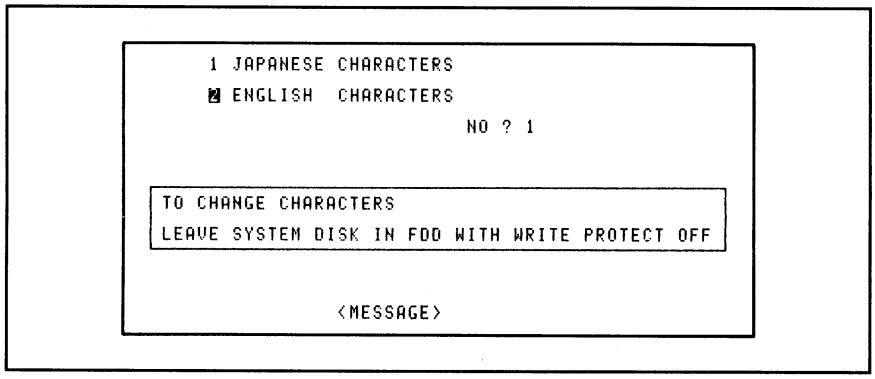
PC mode: To read, write and verify the file register/extension file register data between the CPU and GPP/HGP/PHP.

List mode: To display a list of, or change, the file register/extension file register data in the GPP/HGP/PHP.

FDD mode: To store the file register/extension file register data from the GPP/HGP/PHP memory onto disk.

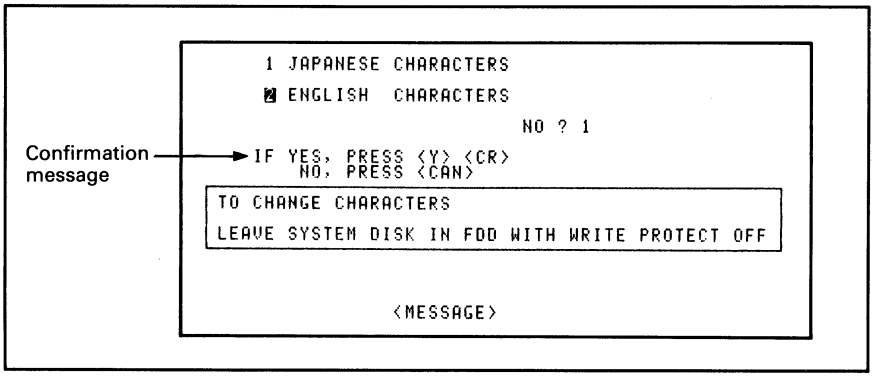
Printer mode: To print out the file register/extension file register data in the GPP/HGP/PHP memory.

5.2 Character Set Menu



(1) Specify the required character set by pressing the character number ([1] or [2]) and [CR]. As the factory-set number is 1, [CR] may only be pressed when 1 is chosen.

(2) The following screen appears after a new character set is chosen.



1) Press [Y], [CR] to specify a new character set and move onto the CPU type select screen. The data for the chosen character set is entered to the UTLP and the character set will default to the selected number from the next time.
When choosing a new character set, the UTLP must be in drive [A] and its write protect off. If the UTLP is write protected, the message "FLOPPY DISK ERROR" is displayed and the character set is not changed.

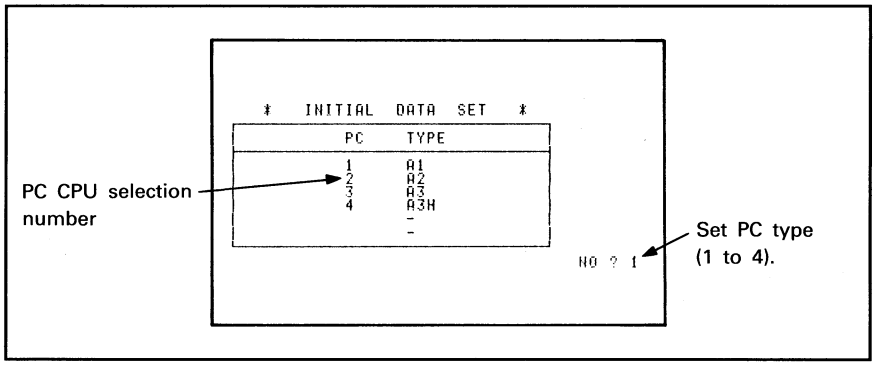
2) Press [CAN] to use the same character set and return to the character set select menu.

(3) Special characters differ as shown below between the Japanese and English character sets:

Japanese character set		English character set
¥	⟷	\
\	⟷	~
^	⟷	^
~	⟷	`
@	⟷	@
`	⟷	[
[	⟷	{
{	⟷	]
]	⟷	}
}	⟷	¥

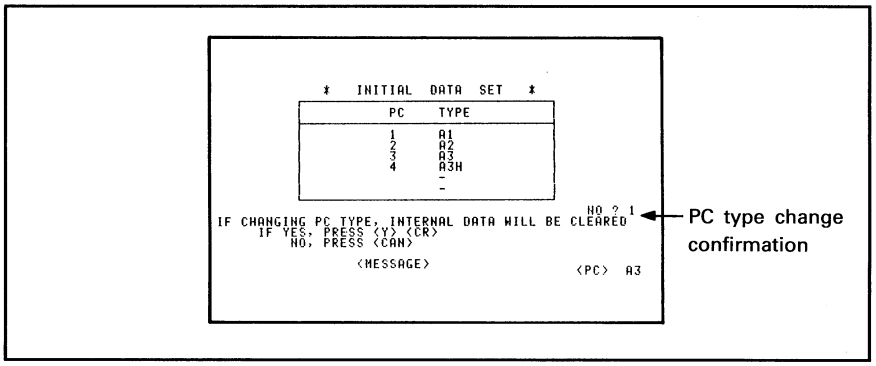
5.3 Initial Data Menu

After the UTLP-FN1 has loaded, the initial data menu is displayed as shown below:



(1) Specify the PC type being used by keying in [1] to [4] as indicated on the menu and move on to the function select menu (Section 5.1) by pressing [CR]. When the file register function has been selected, the function select menu appears if the previous setting differs from the new setting.

(2) When the microcomputer function has been selected, the following screen is displayed if the previous setting differs from the new setting.



a) Press [Y], [CR] to specify a new CPU type and move on to the screen copy select screen (Section 5.1).
b) Press [CAN] to use the same CPU and return to the initial data menu.

POINT

(1) After the microcomputer function has been selected, all data will be cleared by changing the CPU type.
(2) Press [CAN] to return to the initial data menu.

5.4 General Information on FDD Mode

5.4.1 File name structure

(1) A file name is necessary to write data onto disk.

(2) The structure of a file name is as follows:

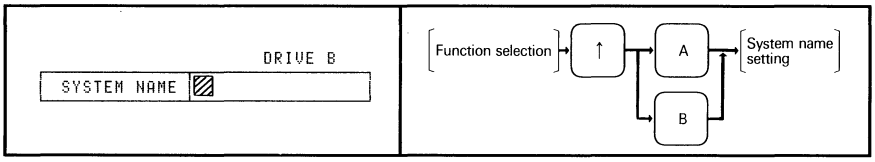
File name = [System name] . [Identifier]

(3) A system name may be specified as appropriate by the user. Up to eight characters may be entered by using alphanumeric and minus (—) keys. (Space must not be used.) Any system name must be headed by an alphabetic.

(4) An identifier is automatically placed after the system name and defines the memory contents as indicated below:

Identifier	Data
PMA	Parameters + main program data
FN1	File register, extension file register

5.4.2 Drive switching



(1) The menus in FDD mode defaults to drive B. To change the drive, press [↑] to move the cursor over the drive designation and key in A or B.

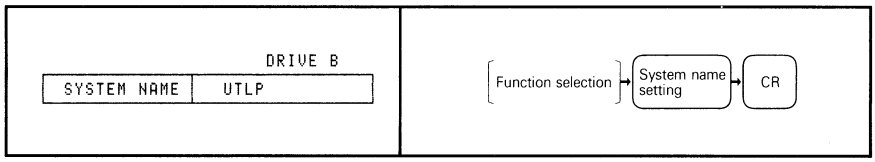
POINT

The above does not apply to the A6HGP and A6PHP which have one drive only.
When a drive has to be selected (i.e. drive A or B) from a menu, either may be specified.

IB (NA) 66169-A

5.4.3 Specifying the system name

The following applies to read, write, verify and copy functions.



(1) Valid keys

[Alphanumeric] / [−(minus)] :	The character is displayed at the cursor position.
[→], [←] :	Moves the cursor.
[BS] :	Deletes the system name and moves back one space.
[HOME CLEAR], [DEL] :	Deletes the system name and moves the cursor to the first system name character space.

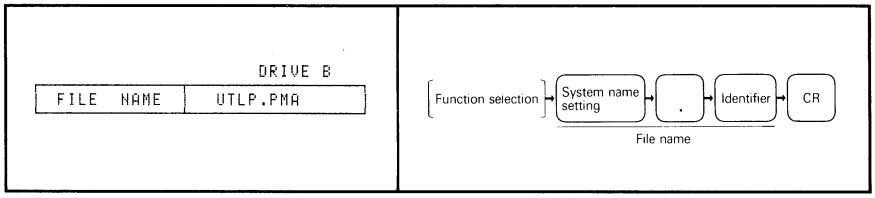
(2) A maximum of eight characters may be used (alphanumeric and minus(−)). The first character must be alphabetic.

POINT

(1) The sequence of letters ALL cannot be used in a system name as these are reserved for batch copying.
(2) Spaces may not be included in a system name.

5.4.4 Specifying the file name

The following applies to delete and directory functions.

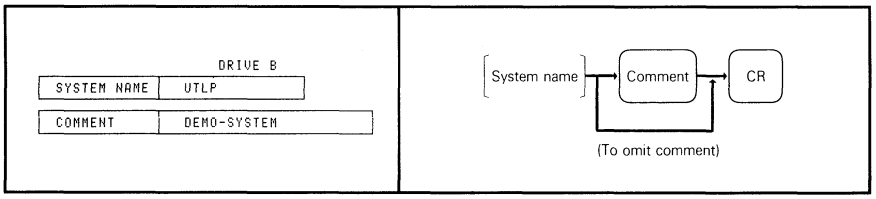


(1) Valid keys

[Alphanumeric] / [special character] :	The character is displayed at the cursor position.
[. (period)] :	Divides system name and identifier.
[✳] :	"Wild card"
[BS] :	Deletes the character and move back one space.
[HOME CLEAR], [DEL] :	Deletes the system name and moves the cursor to the first system name character space.

5.4.5 Specifying the comment

A comment may be added to a system name to help identify it. The comment is displayed next to the file name in the directory listing.



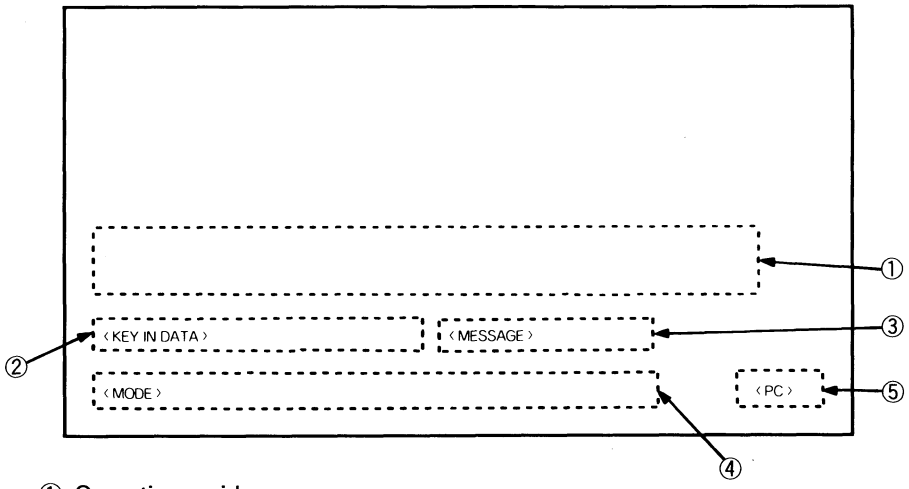
(1) Valid keys

[Alphanumeric] [special character] :	The character is displayed at the cursor position.
[→], [←] :	Moves the cursor.
[SP] :	Space key
[BS] :	Deletes the character and move back one space.
[HOME CLEAR], [DEL] :	Deletes the system name and moves the cursor to the first system name character space.

(2) A maximum of 20 characters can be used, including space and all symbols on the keyboard.

5.5 Description of Screen Data

The following describes the main display areas and the information contained there.



① Operating guide

Gives further information on settings, etc.

② KEY IN DATA area

Not used.

③ MESSAGE area

Gives prompts and error messages relevant to the mode being used.

④ MODE

Shows a menu of the different modes and indicates the current mode by highlighting it.

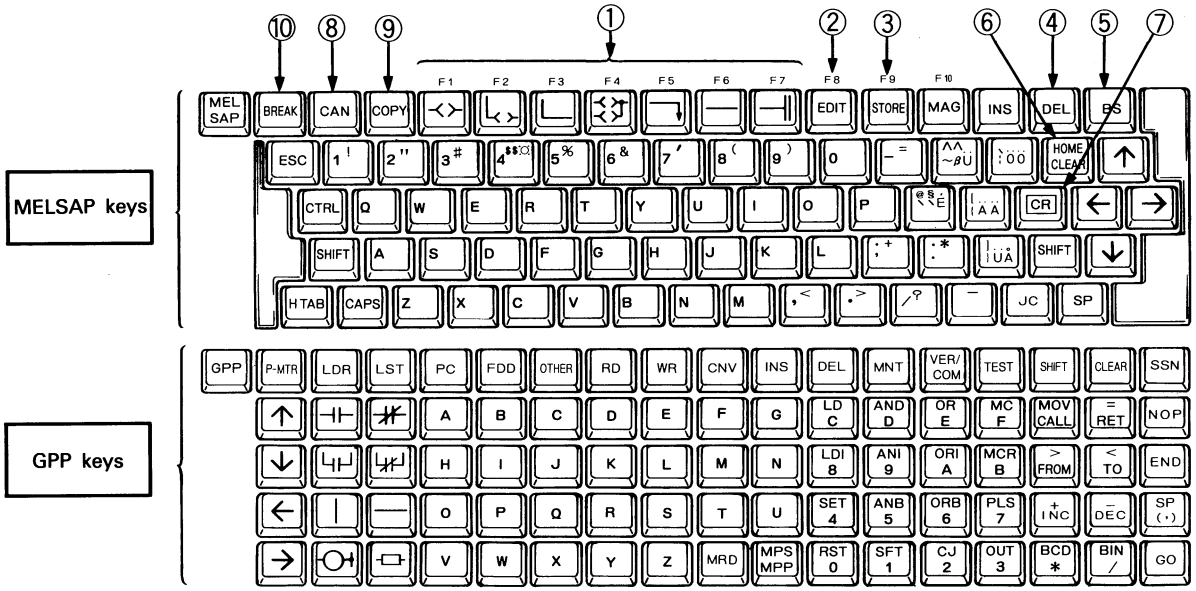
⑤ PC

Indicates the PC CPU type selected on the initial setting screen.

5.6 Keyboard

5.6.1 Description of keys

Use the upper (MELSAP) keyboard.



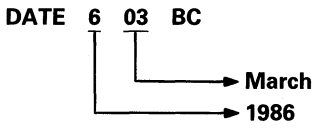
	Key	Function
①	F1 to F7	Select mode.
②	EDIT (F8)	Calculates program head addresses during utility program write.
③	STORE (F9)	Used to save utility programs.
④	DEL	Deletes entered data.
⑤	BS	Deletes a character at the cursor position and moves the cursor one character to the left.
⑥	HOME CLEAR	Deletes entered data.
⑦	CR	• Used to signal the CPU to accept entered data. • Used to stop printing during screen copy. (A screen copy routine cannot be stopped if the printer power is off, the SEL switch is off, or a cable is not connected.)
⑧	CAN	Used to return to the preceding screen.
⑨	COPY	Used to execute screen copy.
⑩	BREAK	• Used to re-start PHP/HGP after disk copy routine. • Stops processing.

5.6.2 Adjusting the buzzer volume

Press [CTRL] + [1] to switch the buzzer volume between high and low levels.

POINT

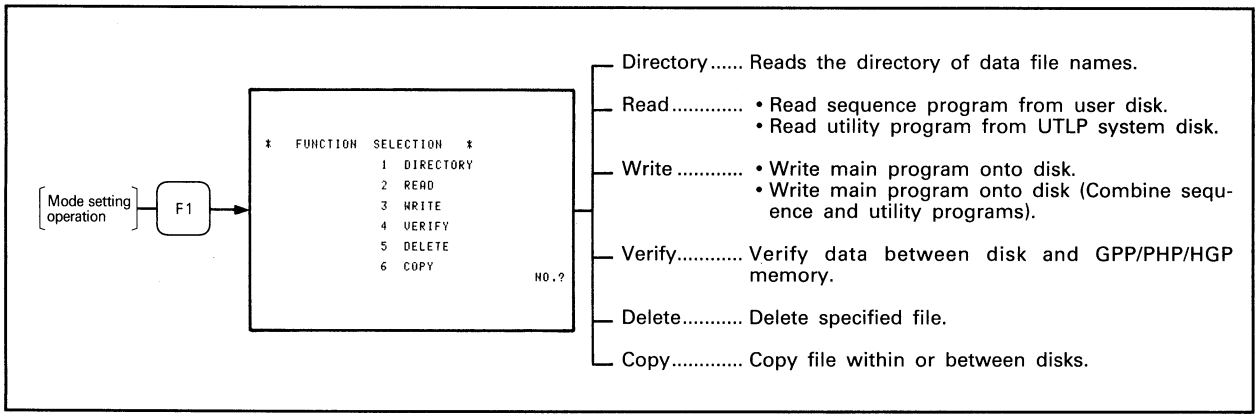
The volume can be adjusted on GPP manufactured in March, 1986 and thereafter. Check the DATE column on the rating plate.



6. MICROCOMPUTER FUNCTION

6.1 FDD Mode

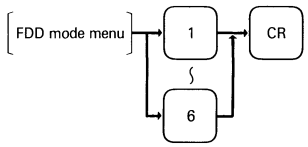
6.1.1 FDD mode functions



6.1.2 FDD mode menu selection

Press F1 to enter FDD mode and then select the required function from the menu.

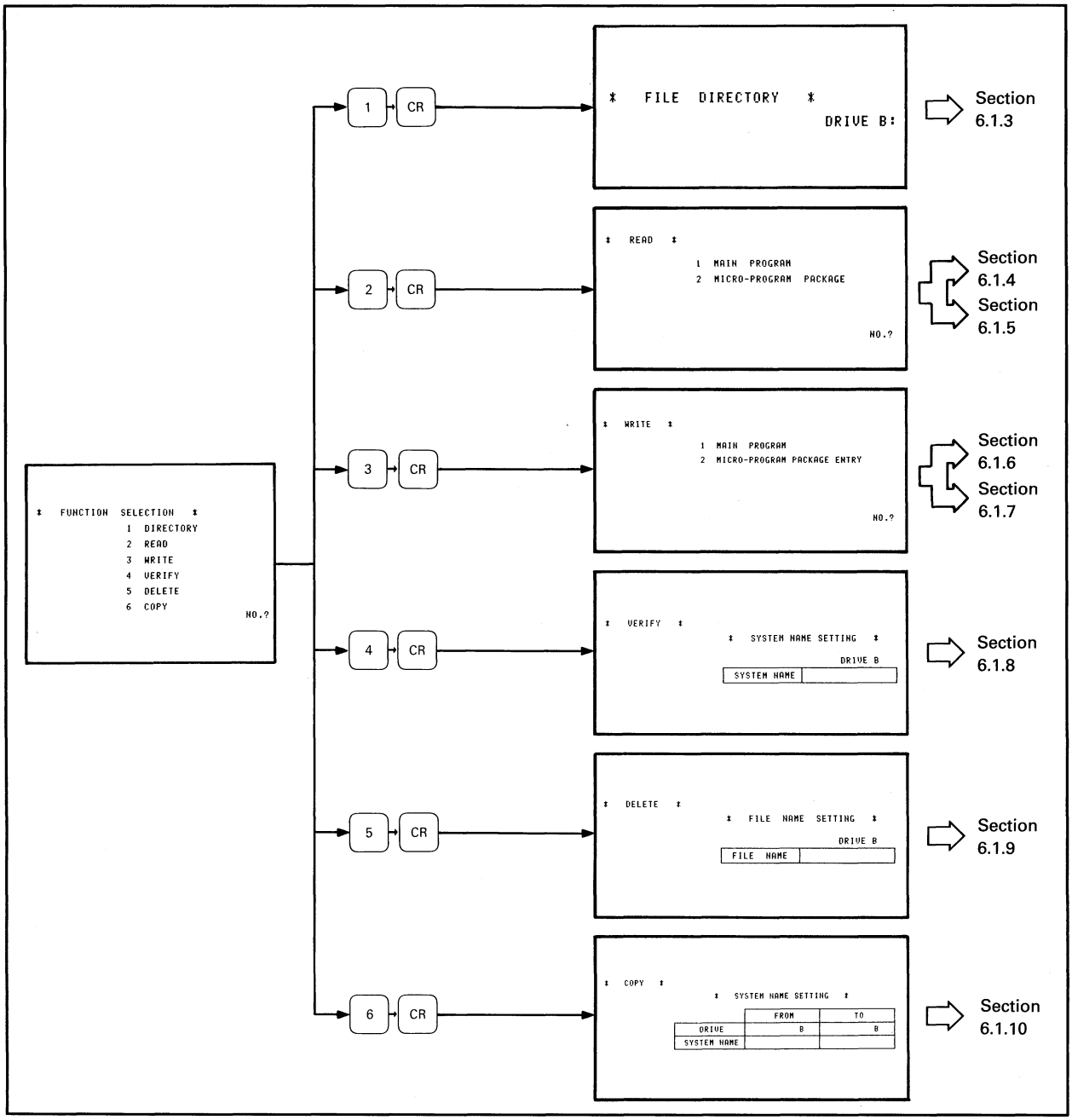
BASIC OPERATION



EXPLANATION

(1) Use keys [1] to [6] and press [CR] to select the required function. The sequence of display screens is shown below:

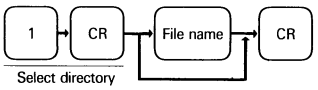
(2)The sequence of display screens is as follows:



6.1.3 Directory

Reads the directory of file names on the disk.

BASIC OPERATION



EXPLANATION

(1) The directory may be examined either for all file names or for a file name with a specified system name or identifier.

(2) To examine all file names, select the directory and press [CR].

(3) To examine a file name with a specified system name and identifier, select the directory, specify the file name, and press [CR].

(4) The character ✳ may be used in file definitions as a "wild card".

a) ✳. Identifier: Read all file names with the specified identifier.

b) ABC✳. Identifier: Read all file names with ABC as the first three characters and the specified identifier.

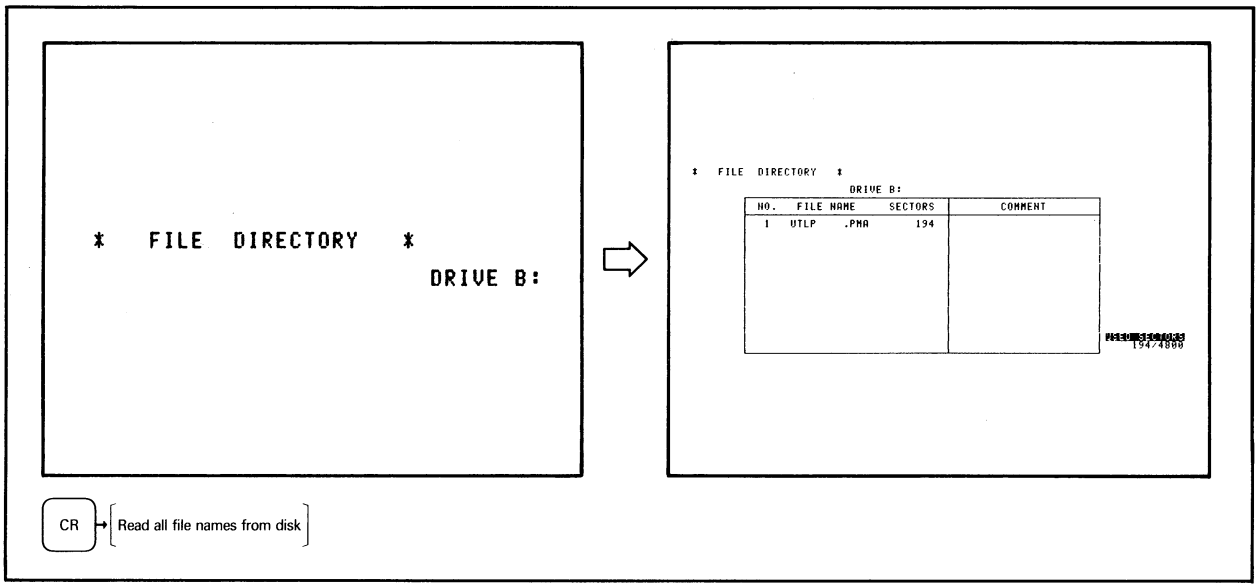
For example, set ABC✳. PMA to read all file names such as ABC1. PMA, ABC2. PMA, ABC-A. PMA, etc.

c) System name. ✳: Read all file names with the specified system name.

(5) 15 file names are read per screen, to read the next 15, press [CR].

(6) The comment is displayed next to the a file name where one has been entered.

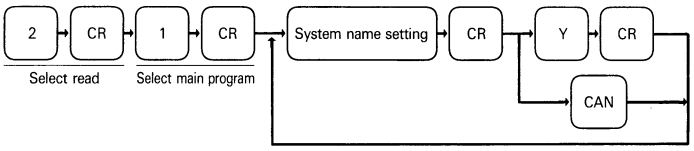
(7) Sequence of directory screens for reading all file names from the disk in drive B.



6.1.4 Main program read

In order to add the utility program to the main program, the user program must first be read from disk.

BASIC OPERATION



EXPLANATION

(1) The user program is read from disk. If the utility program has already been added to the user program, this is read as well.

(2) Select the read function from the FDD mode menu.

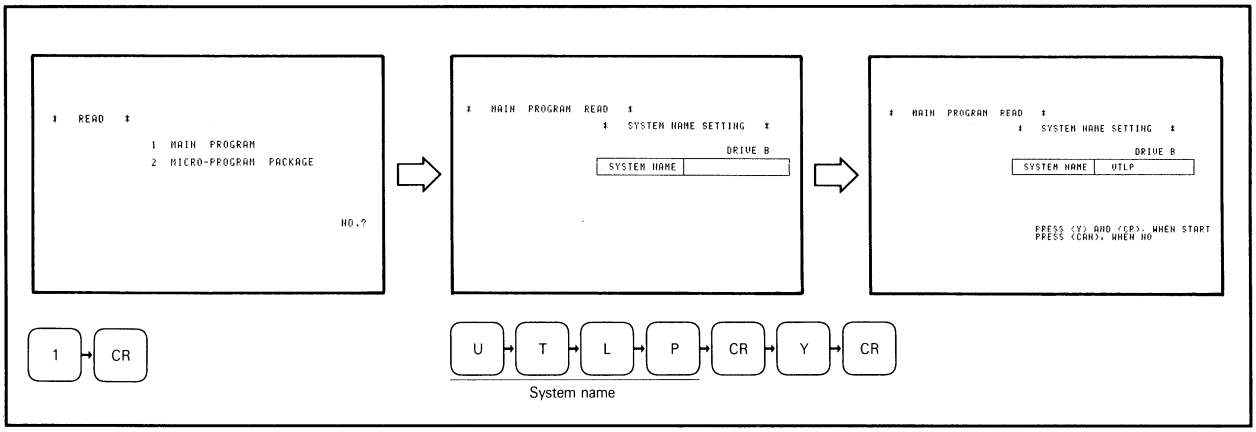
(3) Key in the system name and press [CR].

(4) Follow the screen prompts as shown below:

```
PRESS  〈Y〉  AND  〈CR〉  ,  WHEN  START
PRESS  〈CAN〉  ,  WHEN  NO
```

Press [Y] [CR] to read the specified user program.
Press [CAN] to re-enter the system name.

(5) The sequence of screens during main program read is shown below:



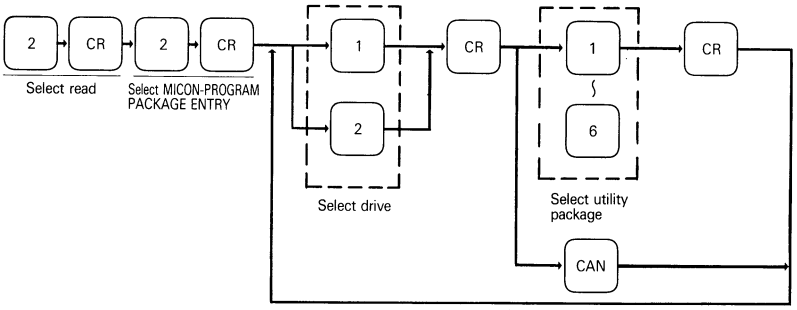
POINT

The user program cannot be read from the disk if the PC CPU type set initially does not match the type set for the program on disk. In this case change the initial PC setting by calling the initial data menu using [CAN].

6.1.5 Utility program read

The utility program is stored on the UTLP- system disk and must be read from there to the GPP/HGP/PHP memory in order to add it to the PC user program. Where any other microcomputer utility programs are to be used in the same PC, these are read in the same way.

BASIC OPERATION



EXPLANATION

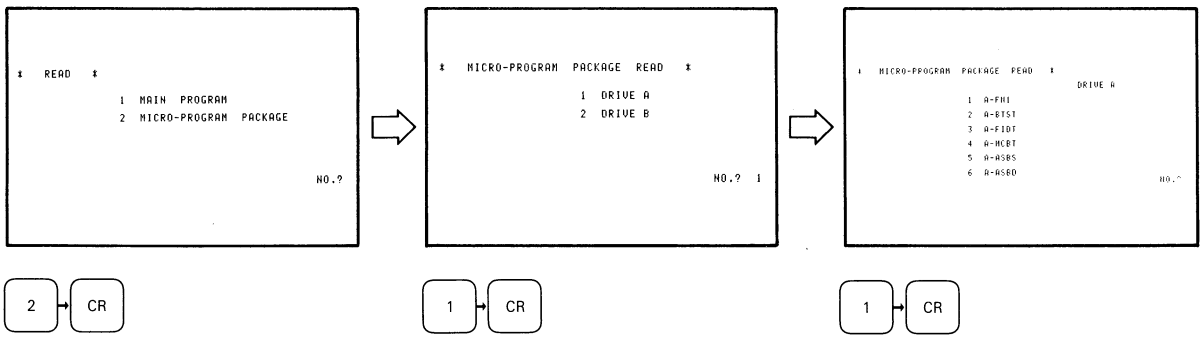
(1) Select the drive containing the system disk using keys 1 or 2 as indicated on the menu (either may be selected when using the HGP or PHP).

(2) Select the utility package required using keys 1 or 2 as indicated on the menu.

(3) The following package menu is displayed:

1. A-FN1
2. A-BTST
3. A-FIDT
4. A-MCBT
5. A-ASBS
6. A-ASBD

(4) Sequence of screens:



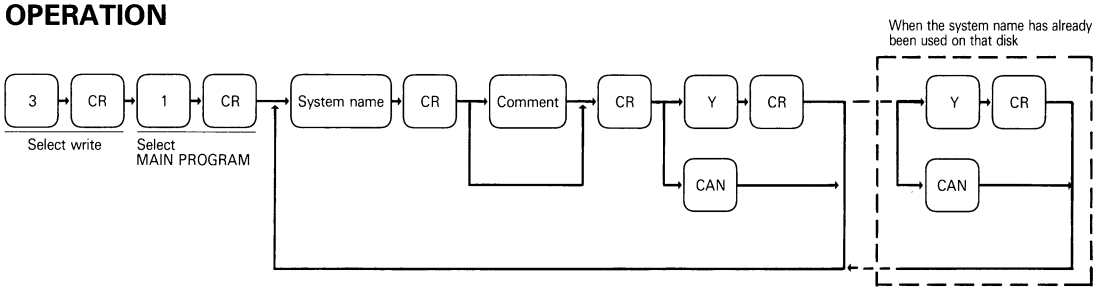
POINT

The utility package may only be read once.

6.1.6 Main program write

Write the parameters and main program from the GPP/PHP/HGP to the user disk.

BASIC OPERATION



EXPLANATION

(1) After setting the system name and comment, the following prompt is displayed.

```
PRESS  〈Y〉  AND  〈CR〉 , WHEN START
PRESS  〈CAN〉 , WHEN NO
```

Press [Y] [CR] to open a file with the specified system name and write the main program to it.
Press [CAN] to re-enter the system name.

(2) When the system name already exists on the disk the following message is displayed:

```
SYSTEM NAME ALREADY USED!!
```

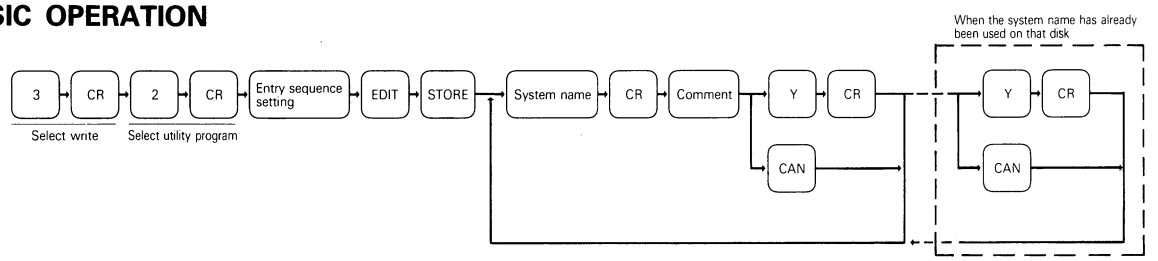
Press [Y], [CR] to overwrite this file.
Press [CAN] to re-enter the system name.

(3) The microcomputer program is not written onto disk unless it has been entered by selecting MICRO-PROGRAM PACKAGE ENTRY.

6.1.7 Utility program entry

Write the utility program from the GPP/PHP/HGP to the parameters and main program on the user disk.

BASIC OPERATION



EXPLANATION

(1) Before this operation both the sequence program and the utility program must have been read from disk.

a) The following message indicates that the user program has not been read.

```
MAIN PROGRAM NOT READ!
   PRESS 〈Y〉 AND 〈CR〉 , WHEN START
   PRESS 〈CAN〉 , WHEN NO
```

Press [Y], [CR] to progress to the entry sequence setting screen (the program cannot be saved to disk).
Press [CAN] to return to the FDD mode menu.

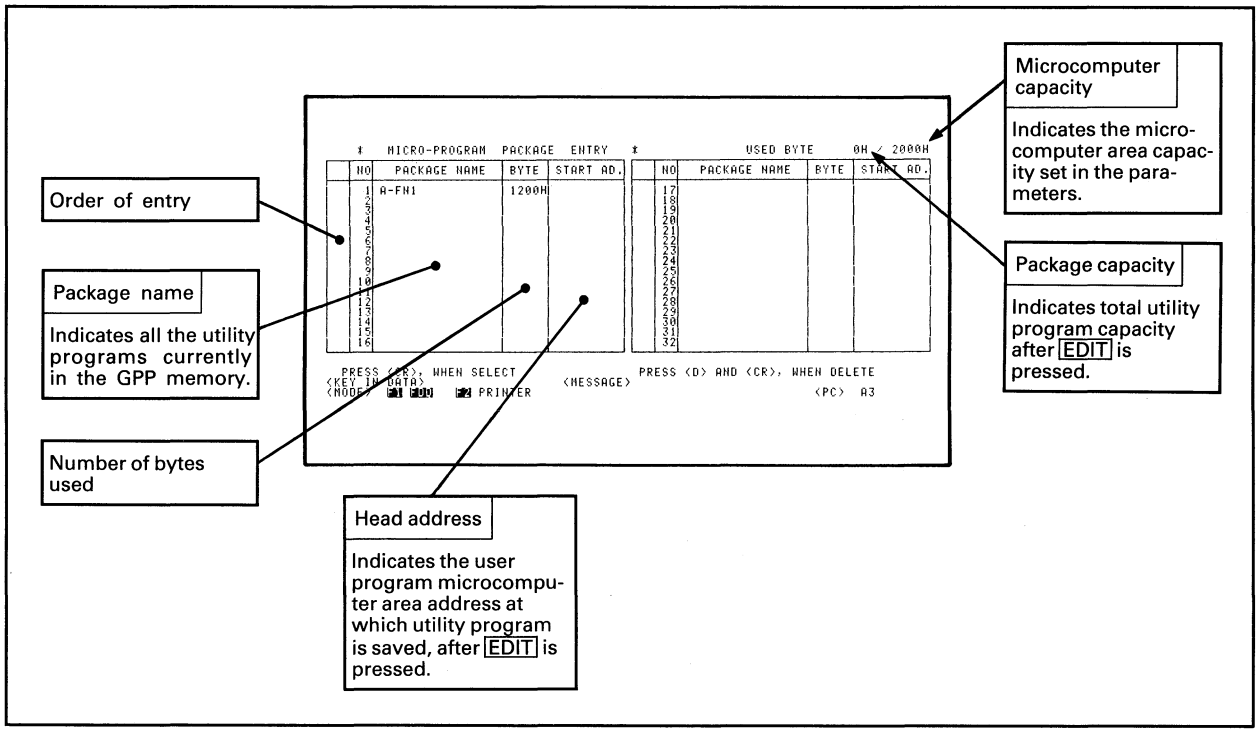
b) The following message indicates that the microcomputer capacity has not been set in the PC parameters.

```
MICON-CAPACITY NOT SET!
   PRESS 〈Y〉 AND 〈CR〉 , WHEN START
   PRESS 〈CAN〉 , WHEN NO
```

Pressing [Y], [CR] to progress to the entry sequence setting screen (the program cannot be saved to disk).
Press [CAN] to return to the FDD mode menu.

The entry sequence setting screen without any data in the package name area indicates that the utility program has not been read.

(2) In some cases, more than one microcomputer utility package is required per sequence program. In this case read the relevant main program from the user disk (i.e. sequence program and micro program(s)) into the GPP/HGP/PHP memory. The utility program is then added to the main program in the GPP/HGP/PHP memory. Finally the combined programs are written back onto disk with the microcomputer programs (and hence their head addresses) arranged in the specified order.



(3) Entry sequence setting

The order in which the microcomputer utility programs are entered into the PC (and hence their head addresses in the microcomputer area) must be specified on the sequence setting table as follows:

1) Use the ↓ and/or ↑ key to move the cursor to the utility program to be entered at the first address in the microcomputer area.
2) Press CR. "1" is displayed in the order of entry column next to the selected utility program.
3) Using the ↓ and/or ↑ key, move the cursor to the utility program to be entered next and press CR.
   "2" is displayed in the order of entry column next to the selected utility program.
4) Continue specifying the order of entry for each microcomputer program.
5) To re-enter the order, press DEL, SP or HOME CLEAR. All the order of entry numbers are canceled.
6) To delete a utility program from the table, move the cursor to that program and press D, CR.
7) Do not perform the entry sequence setting for the utility program which will not be combined with the user program.

(4) After completing the order of entry table, press [EDIT].
The head address of each utility program is displayed in the START AD. column.
The message "EXCESS OF MICON-CAPACITY" indicates that the total number of bytes of utility program exceeds the microcomputer capacity set in the parameters. START addresses which are outside the microcomputer area are shown as "✳✳✳✳✳".
In this case, change the microcomputer program capacity parameter.
Press [EDIT] to combine the utility program and the user program in the GPP/PHP/HGP memory.

POINT

Addresses 0_H to $6F_H$ in the microcomputer area are automatically reserved as a work area.
The first head address is therefore 70_H.

(5) The following message indicates that the microcomputer area has sufficient capacity for the specified utility programs:

```
PRESS  〈STORE〉 , WHEN  ENTRY
```

(6) Pressing [STORE] combines the utility program and the user program in the GPP/PHP/HGP memory. The screen is now prepared for system name entry.

(7) Key in the system name and comment (if required) and press [CR] to display the following prompt:

```
PRESS  〈Y〉  AND  〈CR〉 , WHEN  START
PRESS  〈CAN〉 , WHEN NO
```

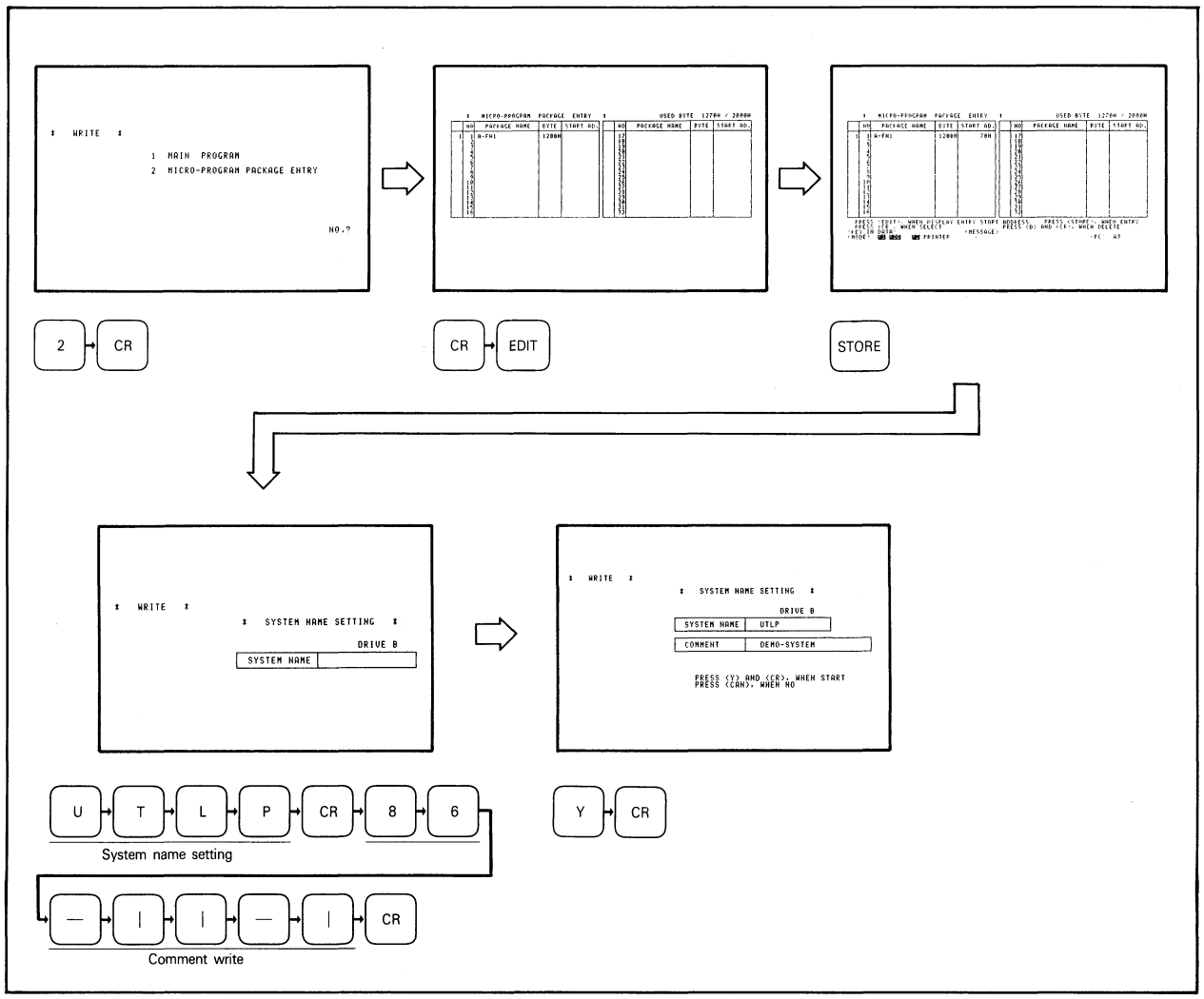
Press [Y], [CR] to write data to the file name.
Press [CAN] to re-enter system name.

(8) The message:

```
SYSTEM  NAME  ALREADY  USED!!
```

indicates that the specified file name already exists on that disk. In this case, press [Y], [CR] to overwrite or press [CAN] to re-enter the system name.

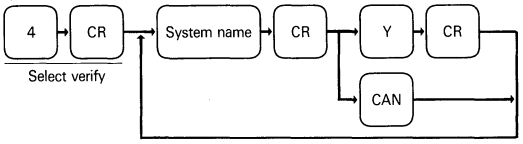
(9) The sequence of display screens is as follows:



6.1.8 Verify

Compares the programs stored in the GPP memory and on disk and verifies that they are identical.

BASIC PROGRAM



EXPLANATION

(1) Only user programs (i.e. those files with identifier "PMA") can be verified.

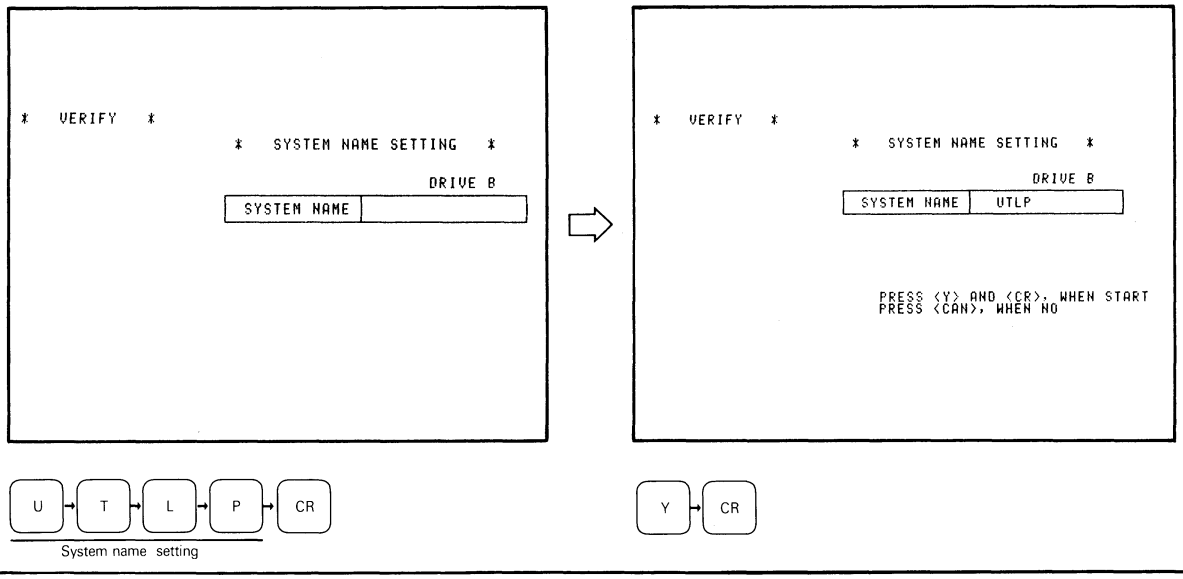
(2) Identify the program on disk by its system name and press [CR]. Follow the resulting prompts as shown:

```
PRESS  〈Y〉  AND  〈CR〉 , WHEN  START
PRESS  〈CAN〉 , WHEN  NO
```

Press [Y] [CR] to verify.
Press [CAN] to re-enter the system name.

(3) Any mismatch between the two programs is indicated by the message "VERIFY ERROR."

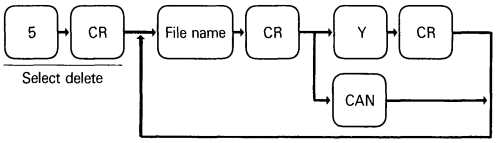
(4) The sequence of display screens is as follows:



6.1.9 Delete

Deletes files from the disk.

BASIC OPERATION



EXPLANATION

(1) The character "✳" may be used as a "wild card".

a) System name. ✳ ····· Delete all files with the specified system name.

For example, setting ABC. ✳ deletes ABC. PMA, ABC. COT, ABC. STR, etc.

b) ✳. Identifier ····· Delete all file names with the specified identifier.

For example, setting ✳. PMA deletes ABC. PMA, BCD. PMA, CDE. PMA, etc.

c) [ ][ ][ ]✳. Identifier ····· Delete all file names with [ ][ ][ ] as the first three characters and the specified identifier.

For example, setting ABC✳. PMA deletes ABC-1. PMA, ABC-2. PMA, ABC-3. PMA, etc.

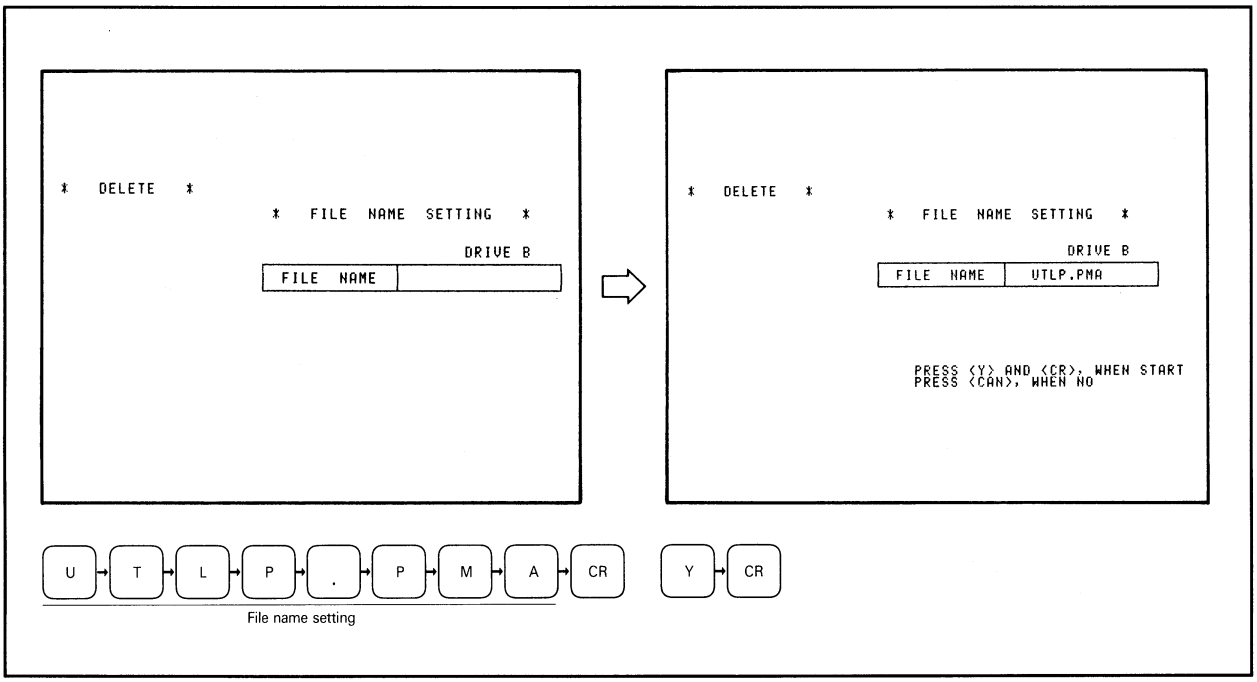
d) ✳. ✳ ····· Delete all files.

(2) Specify the file name and follow the prompts as shown:

```
PRESS  〈Y〉  AND  〈CR〉 , WHEN  START
PRESS  〈CAN〉 , WHEN  NO
```

Press [Y], [CR] to delete.
Press [CAN] to re-enter the system name.

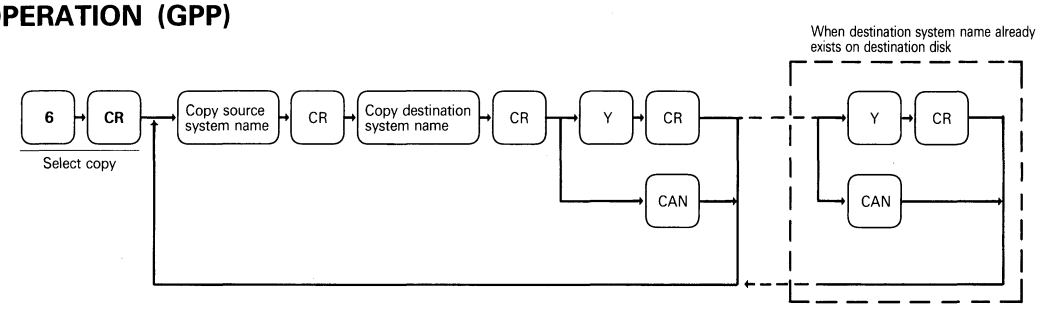
(3) The sequence of display screens is as follows:



6.1.10 Copy

Allows program data to be copied from one disk to another or from one file name to another on the same disk.

BASIC OPERATION (GPP)



EXPLANATION

(1) The A6GPP is fitted with two drives so any of the following copy combinations are possible:

1) From FDD [A] to FDD [B]
2) From FDD [B] to FDD [A]
3) From FDD [A] to FDD [A] (on the same FD)
4) From FDD [B] to FDD [B] (on the same FD)

The source or destination drive specification can be changed as described in Section 6.3.1.

(2) The copy source and destination system names must be different when files are being copies within the same disk.
The destination disk must be formatted.

(3) To batch copy all data from one disk to another, change the drive setting as appropriate (See Section 6.3.1) and specify the system name "ALL" to both the source and destination.

POINT

The system disk UTLP cannot be copied.

(4) Specify the system name and follow the prompts as shown:

```
PRESS  〈Y〉  AND  〈CR〉 , WHEN  START
PRESS  〈CAN〉 , WHEN  NO
```

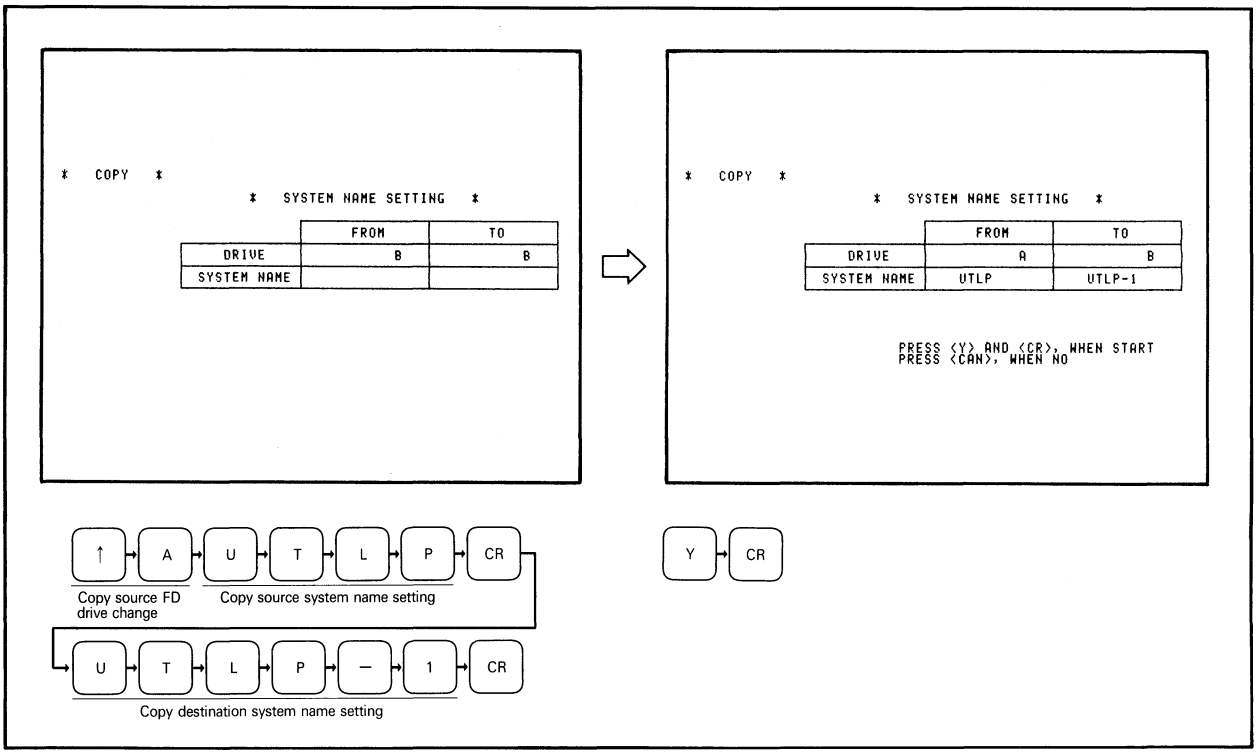
Press [Y] [CR] to copy.
Press [CAN] to re-enter the system names.

(5) The message:

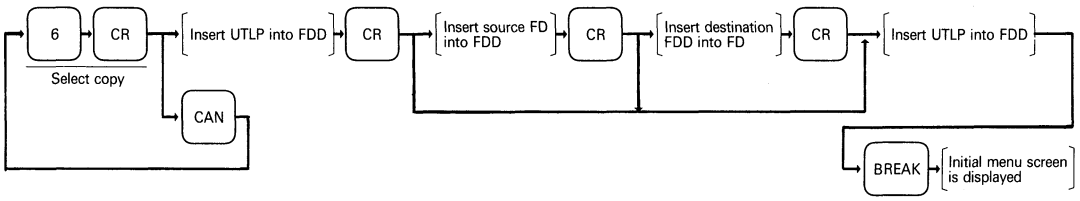
```
SYSTEM  NAME  ALREADY  USED!!
```

indicates that the specified system name already exists on the destination disk.
Press [Y], [CR] to overwrite.
Press [CAN] to re-enter the system name.

(6) The sequence of display screens is as follows:



BASIC OPERATION (PHP/HGP)



POINT

(1) The copy destination disk must be formatted.

(2) Prior to copy, save the PHP/HGP memory data onto disk. Data is cleared from the memory during copying.

(3) The UTLP system disk cannot be copied.

(1) After selecting the copy function on the FDD mode initial screen, press CR to call the following screen.

```
*  COPY  *

        *  DATA OF HGP/PHP WILL BE LOST BY EXECUTING COPY
           PRIOR TO COPY, SAVE DATA ONTO FD
        *  USE FORMATTED FD FOR COPY DESTINATION

              LOAD SYSTEM FD AND PRESS <CR>,
              WHEN EXECUTE COPY
              PRESS <CAN>, WHEN CANCEL COPY
```

(2) Insert the UTLP-FN1 into the drive and press CR.
The copy function is loaded from the UTLP-FN1 and the following screen is displayed.

```
*  COPY  *

      PRESS <CR>, WHEN READ FROM COPY SOURCE FD



           INSERT SYSTEM FD
           AND PRESS <BREAK>, WHEN TERMINATE COPY
```

(3) Insert the source disk into the drive and press CR.

(4) The following is displayed during data reading from the source disk.

```
Reading  TRACK=[  ]
```

← Indicates the track number.

(5) When the following is displayed, insert the destination disk into the drive and press CR.

```
PRESS  <CR>  ,  WHEN  WRITE  TO  DESTINATION FD
```

(6) During writing to the destination disk

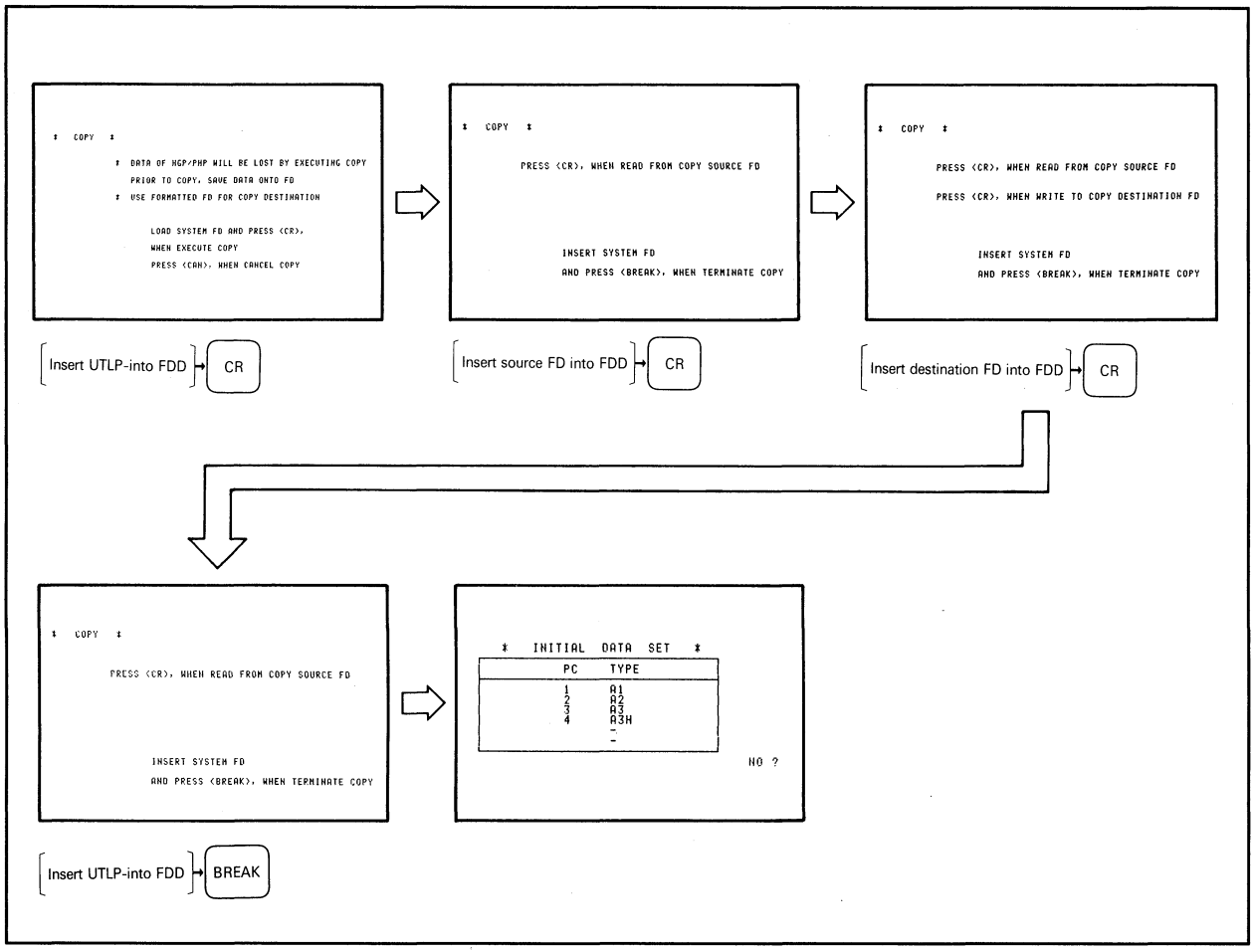
Writing TRACK=[ ] ← Indicates the track number.

is displayed. The data is automatically verified upon completion.

Verifying TRACK=[ ] ← Indicates the track number.

(7) "COMPLETED" is displayed on the INITIAL DATA SET screen to indicate that write and verify are complete.

(8) Copy operation sequence



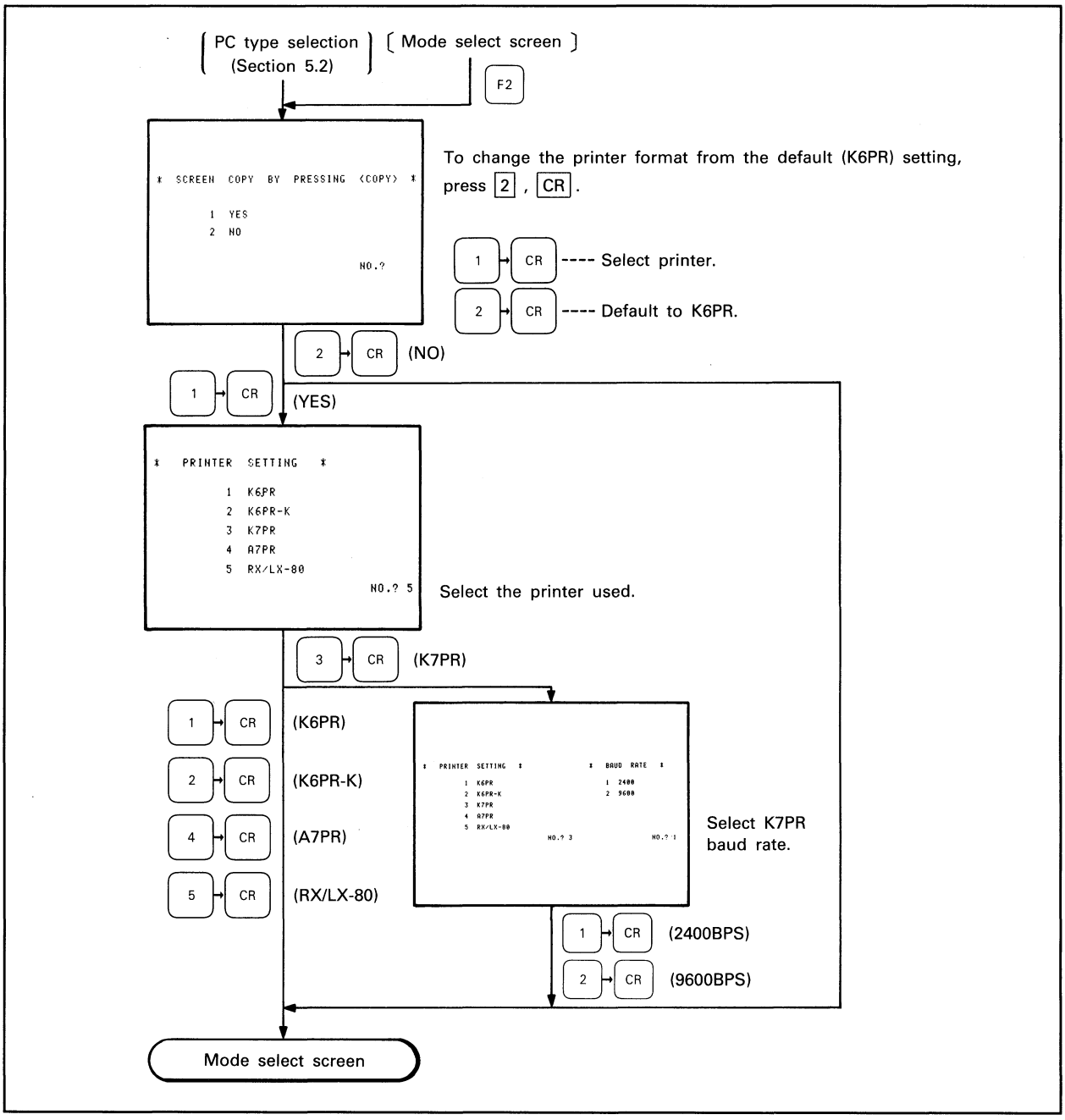
6

6.2 Printer Mode

Select the printer for screen copy as described below.

6.2.1 Selecting the printer

When the microcomputer function has been selected, all print outs of data are obtained by hard copy screen dumps for which the type of printer used must be specified.
The printer is connected to the RS232C interface.



(1) The printer selected in the above procedure remains valid until the GPP/PHP/HGP is powered down or reset.

(2) Pressing the [COPY] key starts the screen dump.

POINT

(1) The GPP/HGP/PHP will default to K6PR baud rate, etc.

(2) The RX/LX-80 selection sets the following:
Communication speed = 9600BPS, data length = 8 bits, stop bit = 2, parity = none

7. FILE REGISTER FUNCTION

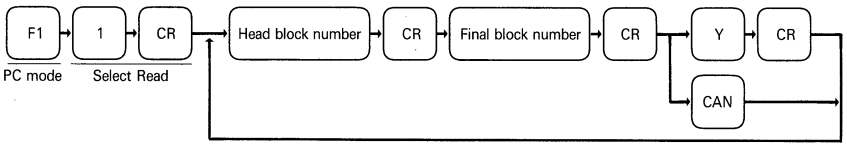
Allows file register (R) and extension file register (extension R) data to be transferred between the GPP/HGP/PHP and PC, displayed as a list, changed, stored onto disk, and printed out.

7.1 PC Mode

7.1.1 Read from PC

Batch-read R and extension R data in the specified blocks from the PC CPU memory cassette to the GPP/HGP/PHP memory.

BASIC OPERATION



EXPLANATION

(1) Specify the head and final block numbers for the file register data to be read from the memory cassette.
Enter the same block number to read data from one block.
The head block number must not be greater than the final block number.

(2) Press [Y], [CR] to check the PC type, parameters, and memory cassette capacity. Read is started if the results are OK.

(3) The error message "BLOCK NO. ERROR" indicates that an invalid block number has been specified. For more information, see Section 4.8.2.

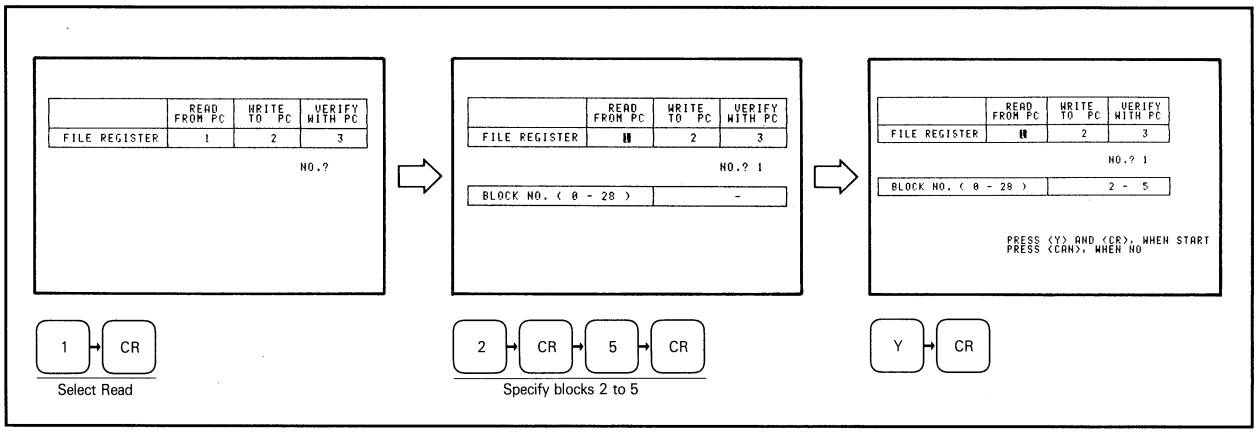
(4) The number of remaining bytes is displayed during read and is decremented by 1 every time 1K bytes are read.

(5) Press [BREAK] to stop read and return the cursor to the block number setting area.

(6) Press [CAN] to change the block number specified.

(7) The error message "PC SELECTION ERROR" indicates that the CPU specified is different from that connected. Read the initial data menu to specify the correct CPU type.

(8) Sequence of display screens:



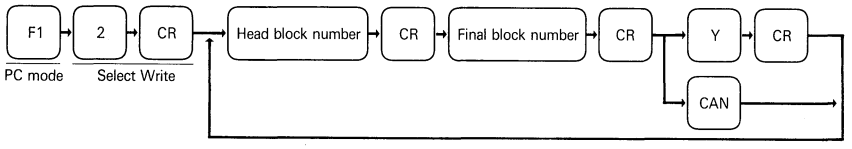
POINT

In PC mode, read, write and verify may be performed during PC RUN. As verify may result in an error during RUN, verify after read should be performed after setting the PC CPU to STOP.

7.1.2 Write to PC

Batch-write R and extension R data in the specified blocks from the GPP/HGP/PHP memory to the CPU memory cassette.

BASIC OPERATION



EXPLANATION

(1) Specify the head and final block numbers for the file register data to be written to the memory cassette.
The head block number must not be greater than the final block number.

(2) Press [Y], [CR] to check the PC type, parameters, memory cassette capacity and memory protect. Write is started if the results are OK.
Memory protect check is only performed on blocks 0 to 9 and A3NMCA-16 blocks 10 and 11. Automatic verify is not performed.

(3) The error message "BLOCK NO. ERROR" indicates that an invalid block number has been specified. For more information, see Section 4.8.2.

(4) The number of remaining bytes is displayed during write and is decremented by 1 every time 1K byte are written.

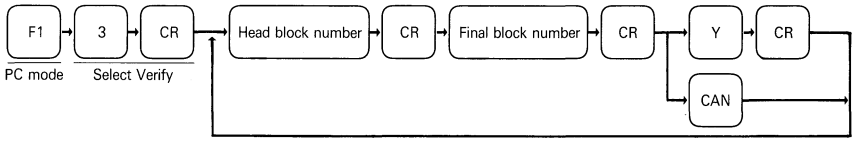
(5) Press [BREAK] to stop write and return the cursor to the block number setting area.

(6) Press [CAN] to change the block number specified.

7.1.3 Verify with PC memory

Compares the R and extension R data in the CPU memory cassette and in the GPP/HGP/PHP memory and verifies that they are identical.

BASIC OPERATION



EXPLANATION

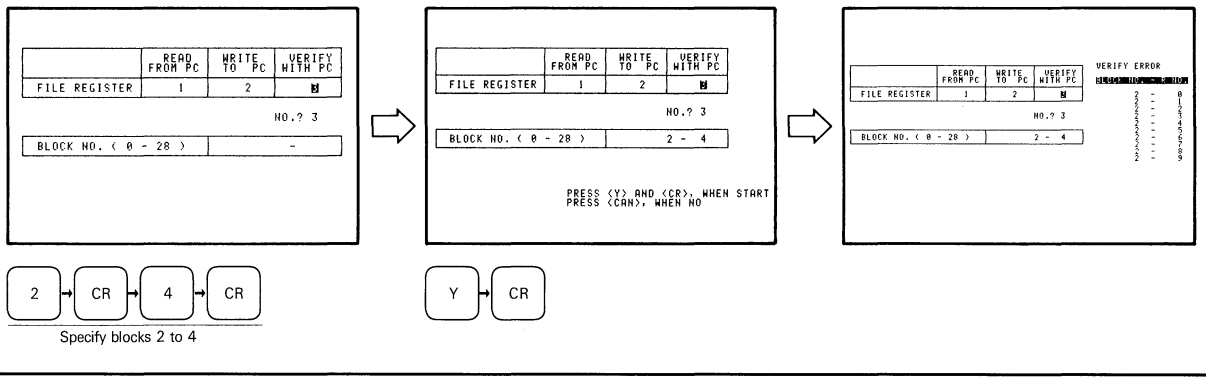
(1) Specify the head and final block numbers for the file register data to be verified.

(2) Although verify may be performed during run, it is suggested to verify data during stop because a mismatch error may occur during run.

(3) If mismatches are detected, up to 10 file register numbers are displayed on the screen together with the corresponding block numbers. The message "MORE THAN 10 ERRORS" indicates that there are more than 10 mismatches.

(4) The number of remaining bytes is displayed during verify and is decremented by 1 every time 1K byte are verified.

(5) Sequence of display screens:

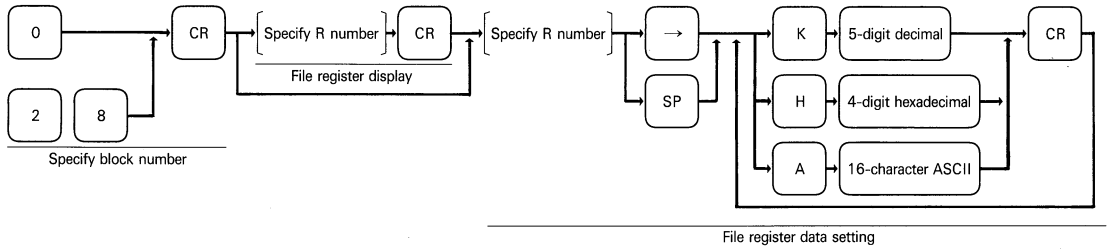


7.2 List Mode

Displays a list of, and changes, the R and extension R data in the GPP/HGP/PHP memory. As the GPP/HGP/PHP is put offline in list mode, the R and extension R areas in its memory can be accessed independently of the CPU memory cassette capacity.

7.2.1 File register display, data setting

BASIC OPERATION



EXPLANATION

(1) Display

(1) Key in the block number (0 to 28) for the file registers to be listed. Any of blocks 0 to 28 can be specified independently of the memory cassette capacity.

(2) When the file register setting screen appears, enter the head file register number to read 72 Rs headed by the specified one. The Rs displayed on the screen remain unchanged if the specified R already exists on the screen. The rightmost ASCII area displays an ASCII value of all data in each line.

(3) Press [CR] to call the next 72 Rs.

(2) Data setting

(1) Enter the required R number and data. Press [K] [ ] before decimal data, [H] [ ] before hexadecimal data, and [A] [ ] before ASCII data. Press [CR] to write the data, call the next R number and move the cursor to the data area. To change the R number, K, H and/or A, move the cursor by the [→] key. Press [HOME CLEAR] to clear the keyed-in data. [CR] must be pressed after any data has been changed.

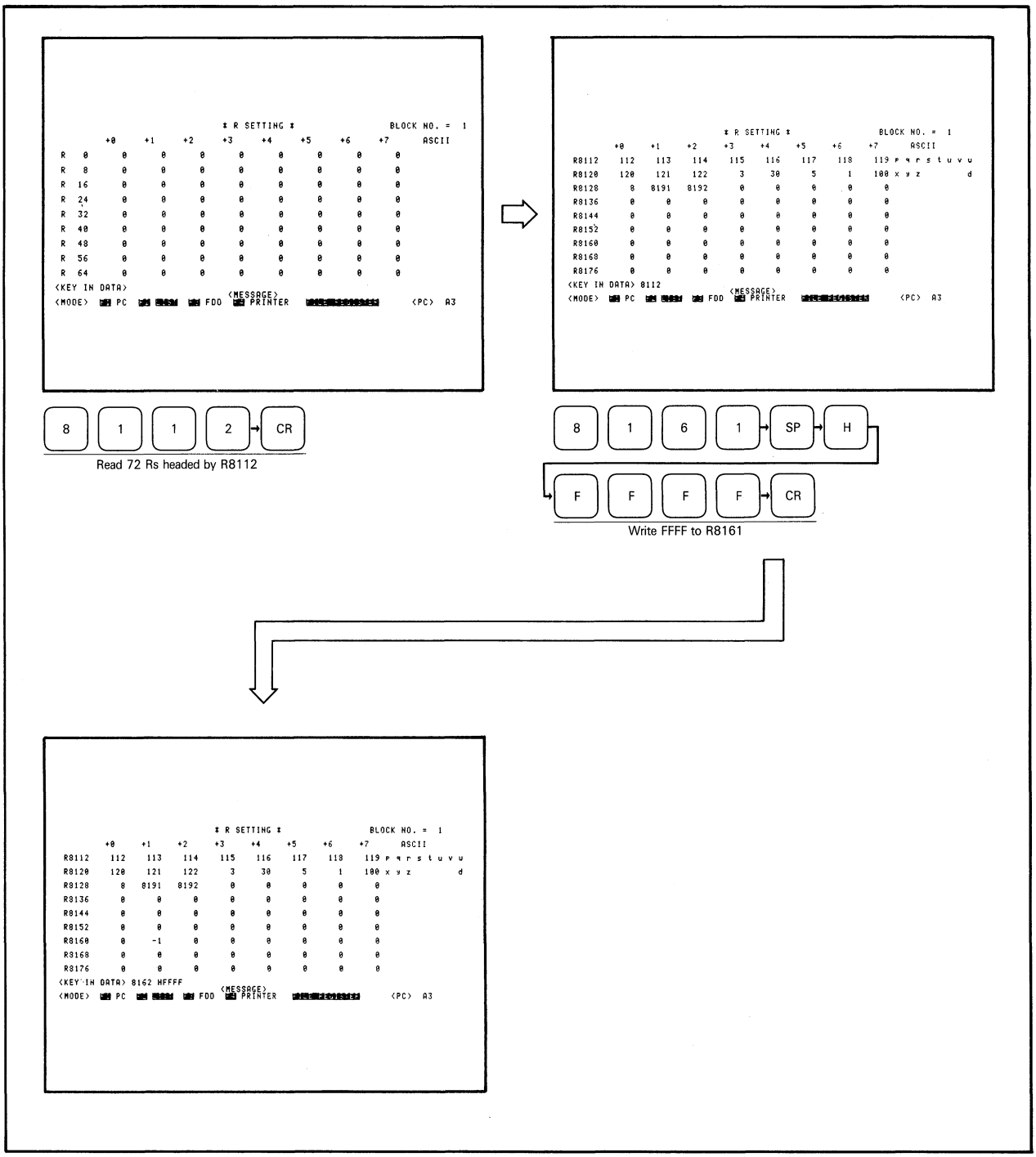
(2) Up to 5 digits (−32768 to 32767) can be entered for a decimal, up to 4 digits (0000 to FFFF) for a hexadecimal, and up to 16 characters for ASCII.

(3) Press [H], [CR] to change a decimal into a hexadecimal and [K], [CR] to get back to the decimal.

(4) Press keys as indicated below to call data of the Rs which are not present on the screen. This example calls data of 72 Rs headed by R123:

[2] → [CR] → [1] [2] [3] → [SP] → [K] → [1] [0] [0] → [CR]

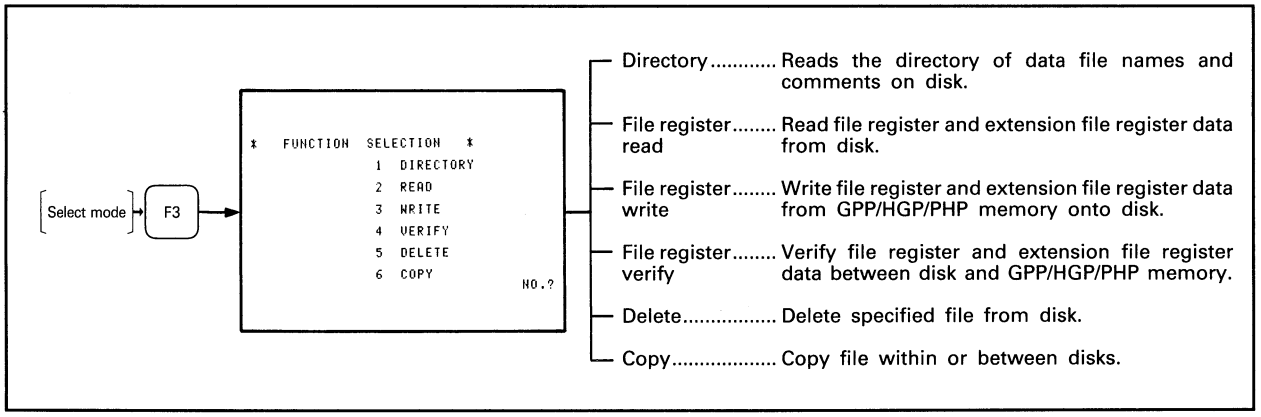
(5) Sequence of display screens:



7.3 FDD Mode

Allows the file register and extension file register data in the GPP/HGP/PHP memory to be written onto disk, read, verified and copied.

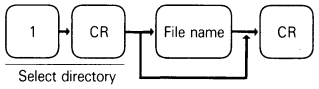
7.3.1 FDD mode functions



7.3.2 Directory

Reads the directory of file names on the disk.

BASIC OPERATION



EXPLANATION

(1) The directory may be examined either for all file names or for a file name with a specified system name or identifier.

(2) To examine all file names, select the directory and press CR.

(3) To examine a file name with a specified system name and identifier, select the directory, specify the file name, and press CR.

(4) The character ✳ may be used in file definitions as a "wild card".

a) ✳. Identifier: Read all file names with the specified identifier.

b) ABC✳. Identifier: Read all file names with ABC as the first three characters and the specified identifier.

For example, set ABC✳. PMA to read all file names such as ABC1. PMA, ABC2. PMA, ABC-A. PMA, etc.

c) System name. ✳: Read all file names with the specified system name.

(5) 15 file names are read per screen, to read the next 15, press CR.

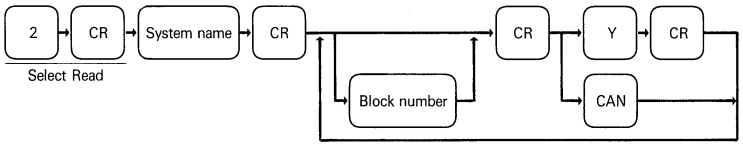
(6) The comment is displayed next to the a file name where one has been entered.

(7) Sequence of directory screens for reading all file names from the disk in drive B.

7.3.3 File register read

Read R and extension R data from disk to the GPP/HGP/PHP memory headed by the specified block.

BASIC OPERATION



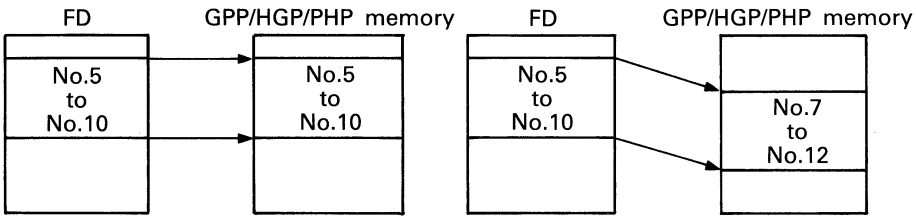
EXPLANATION

(1) Select the drive containing the user disk which stores R and extension R data.

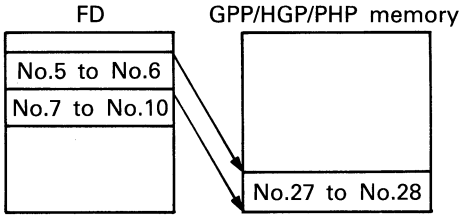
(2) Specify the system name of the file which stores the required R and extension R. Files with identifier "FN1" may only be read. Press [CR] to define the system name and call the block number setting screen.

(3) Press [CR] to store the file to the GPP with the same block numbers. Specify the head block number to store the file to the GPP with different block numbers.

(a) No head block number specified (b) Head block number "7" specified



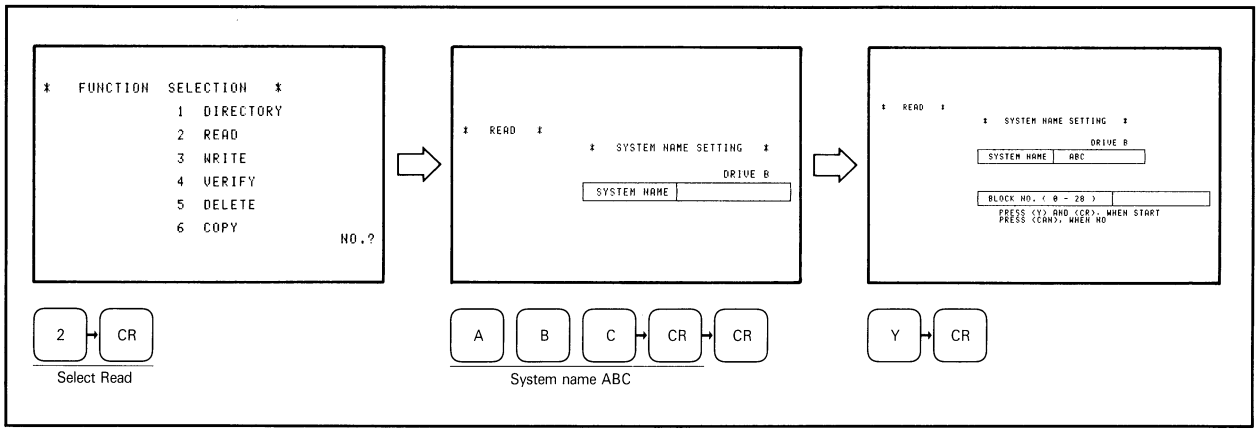
(c) Head block number "27" specified



In this case, the error message "MEMORY CAPACITY EXCEEDED" is displayed and data of blocks 7 to 10 is not read by pressing [Y], [CR].

(4) Press [Y], [CR] to read data. When read is complete, the cursor returns to the block number setting area.
Press [CAN] to return the cursor to the block number setting area.

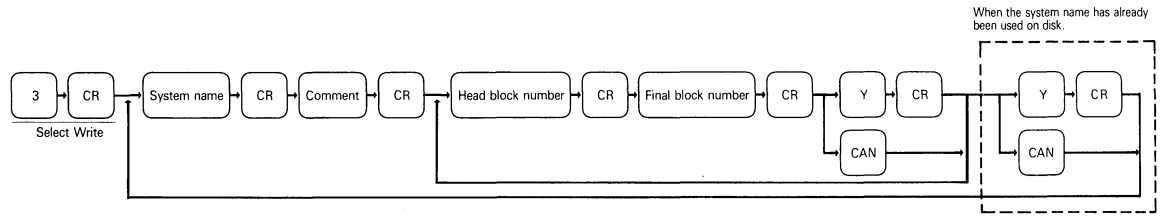
(5) Sequence of display screens:



7.3.4 File register write

Write the R and extension R data from the specified block area of the GPP/HGP/PHP memory onto disk.

BASIC OPERATION



EXPLANATION

(1) After specifying the system name, comment, and block numbers, the following prompt is displayed.

```
PRESS  〈Y〉  AND  〈CR〉  ,  WHEN  START
PRESS  〈CAN〉  ,  WHEN  NO
```

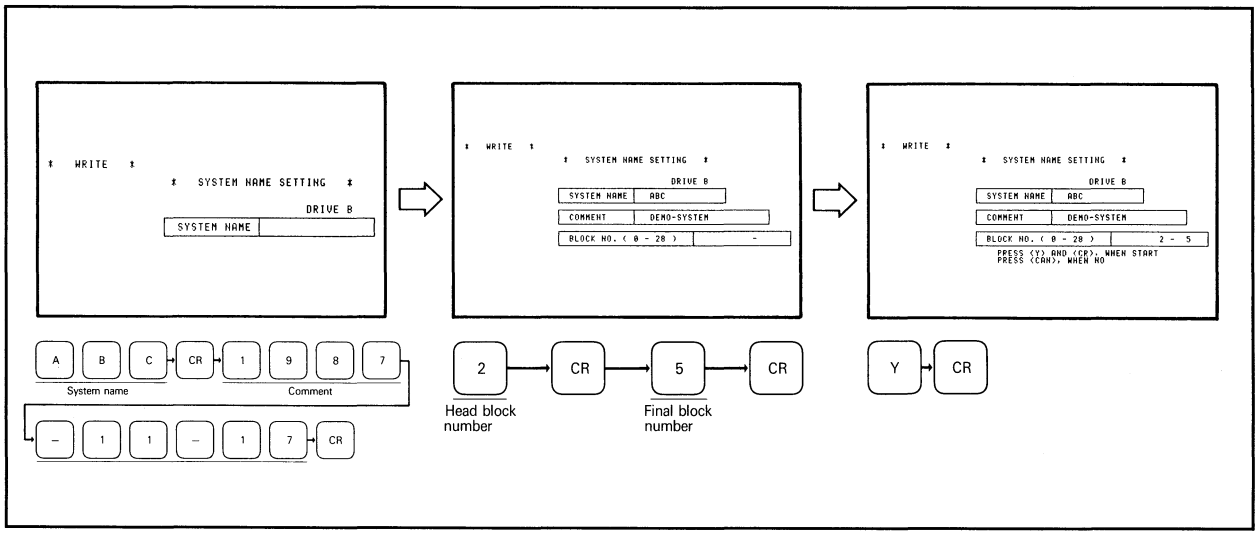
Press [Y], [CR] to open a file with the specified system name and write the data of the specified blocks.
Press [CAN] to return the cursor to the block number setting area.

(2) When the system name already exists on the disk, the following message is displayed:

```
SYSTEM  NAME  ALREADY  USED!!
```

Press [Y], [CR] to overwrite this file.
Press [CAN] to re-enter the system name.

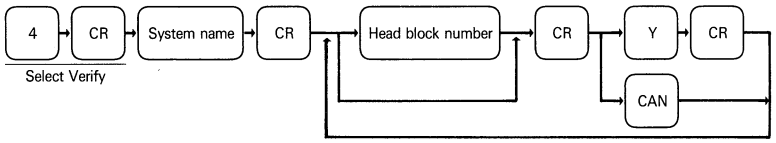
(3) Sequence of display screens:



7.3.5 File register verify

Compares the specified R and extension R data on disk and in the GPP/HGP/PHP memory and verifies that they are identical.

BASIC OPERATION



EXPLANATION

(1) Only file register data (i.e. files with identifier "FN1") can be verified.

(2) Identify the R and extension R data by its system name and press [CR].

(3) When the block number setting screen appears, specify the head block number in the GPP/HGP/PHP memory and press [CR].
Press only [CR] when the block number is not specified.

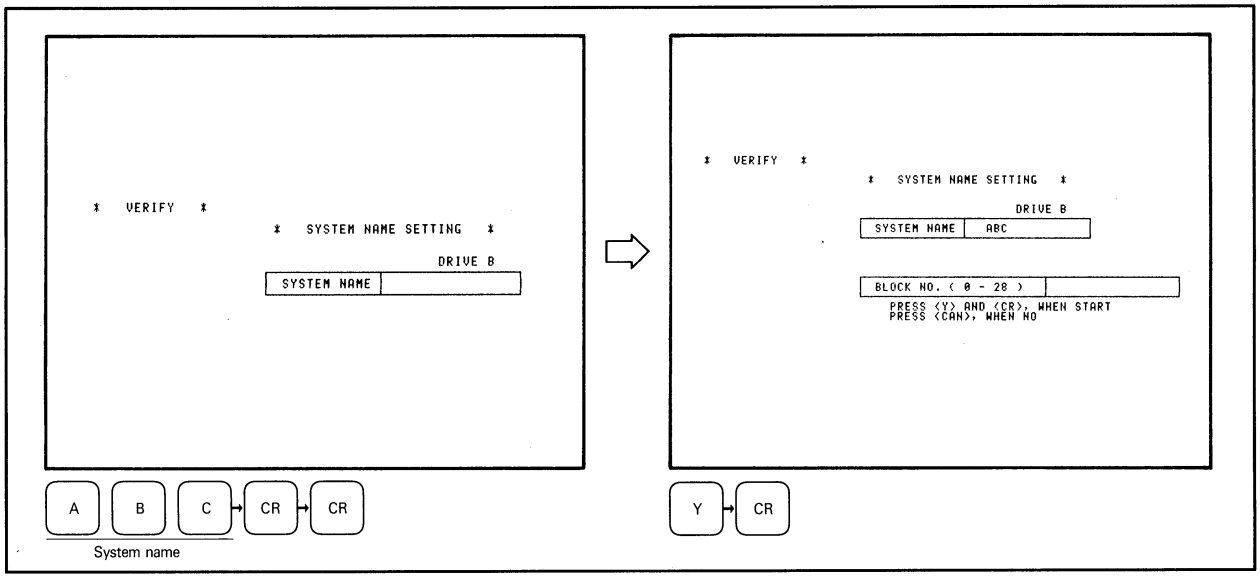
(4) Follow the resulting prompts as shown:

```
PRESS  〈Y〉  AND  〈CR〉 , WHEN  START
PRESS  〈CAN〉 , WHEN  NO
```

Press [Y], [CR] to verify.
Press [CAN] to return the cursor to the block number setting area.

(5) The message "VERIFY ERROR" indicates mismatch between the R and extension R data in the GPP/HGP/PHP memory and on disk.

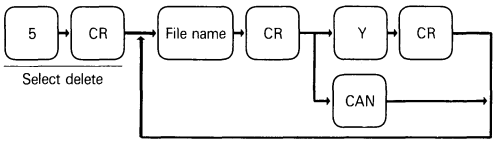
(6) Sequence of display screens:



7.3.6 Delete

Deletes files from the disk.

BASIC OPERATION



EXPLANATION

(1) The character "✳" may be used as a "wild card".

a) System name. ✳ ····· Delete all files with the specified system name.

For example, setting ABC. ✳ deletes ABC. PMA, ABC. COT, ABC. STR, etc.

b) ✳. Identifier ····· Delete all file names with the specified identifier.

For example, setting ✳. PMA deletes ABC. PMA, BCD. PMA, CDE. PMA, etc.

c) ⬚⬚⬚✳. Identifier ····· Delete all file names with ⬚⬚⬚ as the first three characters and the specified identifier.

For example, setting ABC✳. PMA deletes ABC-1. PMA, ABC-2. PMA, ABC-3. PMA, etc.

d) ✳. ✳ ····· Delete all files.

(2) Specify the file name and follow the prompts as shown:

```
PRESS  〈Y〉  AND  〈CR〉  , WHEN  START
PRESS  〈CAN〉  , WHEN  NO
```

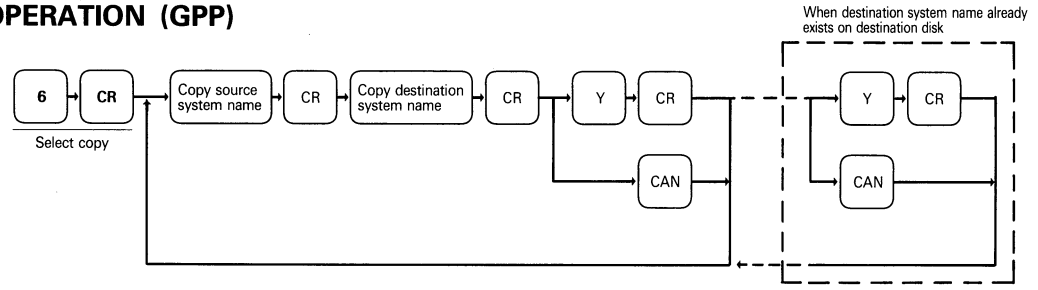
Press [Y], [CR] to delete.
Press [CAN] to re-enter the system name.

7

7.3.7 Copy

Allows program data to be copied from one disk to another or from one file name to another on the same disk.

BASIC OPERATION (GPP)



EXPLANATION

(1) The A6GPP is fitted with two drives so any of the following copy combinations are possible:

1) From FDD [A] to FDD [B]
2) From FDD [B] to FDD [A]
3) From FDD [A] to FDD [A] (on the same FD)
4) From FDD [B] to FDD [B] (on the same FD)

The source or destination drive specification can be changed as described in Section 6.3.1.

(2) The copy source and destination system names must be different when files are being copies within the same disk.
The destination disk must be formatted.

(3) To batch copy all data from one disk to another, change the drive setting as appropriate (See Section 6.3.1) and specify the system name "ALL" to both the source and destination.

POINT

The system disk UTLP cannot be copied.

(4) Specify the system name and follow the prompts as shown:

```
PRESS  〈Y〉  AND  〈CR〉 , WHEN  START
PRESS  〈CAN〉 , WHEN  NO
```

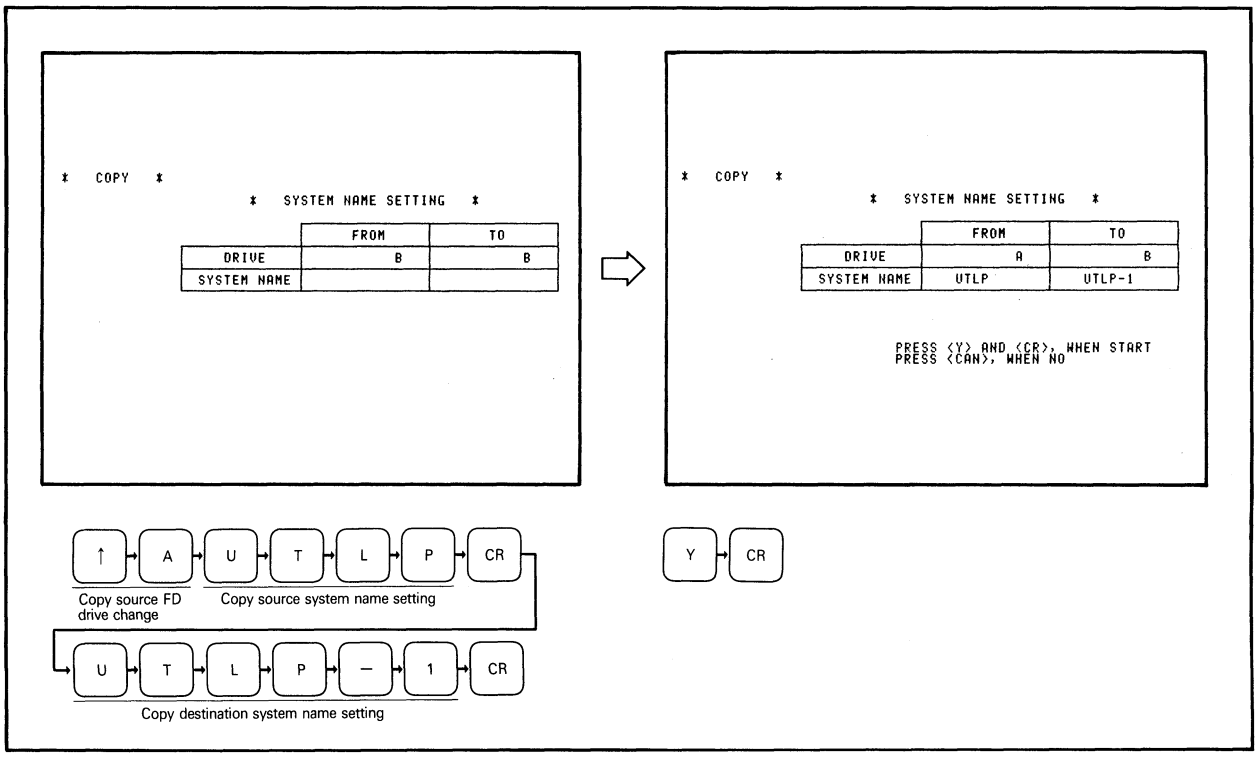
Press [Y] [CR] to copy.
Press [CAN] to re-enter the system names.

(5) The message:

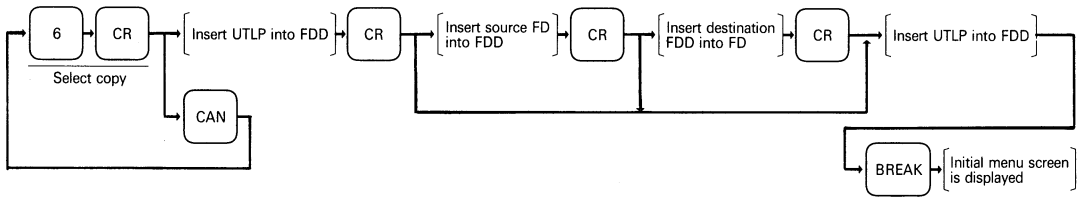
```
SYSTEM  NAME  ALREADY  USED!!
```

indicates that the specified system name already exists on the destination disk.
Press [Y], [CR] to overwrite.
Press [CAN] to re-enter the system name.

(6) The sequence of display screens is as follows:



BASIC OPERATION



POINT

(1) The copy destination disk must be formatted.

(2) Prior to copy, save the HGP/PHP memory data onto disk. Data is cleared from the memory during copying.

(3) The UTLP system disk cannot be copied.

(1) After selecting the copy function on the FDD mode initial screen, insert the UTLP disk into the drive and press CR.
The copy function is loaded from the UTLP and the following screen is displayed.

```
*  COPY  *

        *  DATA OF HGP/PHP WILL BE LOST BY EXECUTING COPY
           PRIOR TO COPY, SAVE DATA ONTO FD
        *  USE FORMATTED FD FOR COPY DESTINATION

              LOAD SYSTEM FD AND PRESS <CR>,
              WHEN EXECUTE COPY
              PRESS <CAN>, WHEN CANCEL COPY
```

```
*  COPY  *

      PRESS <CR>, WHEN READ FROM COPY SOURCE FD



          INSERT SYSTEM FD
          AND PRESS <BREAK>, WHEN TERMINATE COPY
```

(2) Insert the source disk into the drive and press CR.

(3) The following is displayed during data reading from the source disk.

```
Reading  TRACK=[  ]
```

← Indicates the track number.

(4) When the following is displayed, insert the destination disk into the drive and press CR.

```
PRESS  〈CR〉 , WHEN  WRITE  TO  DESTINATION FD
```

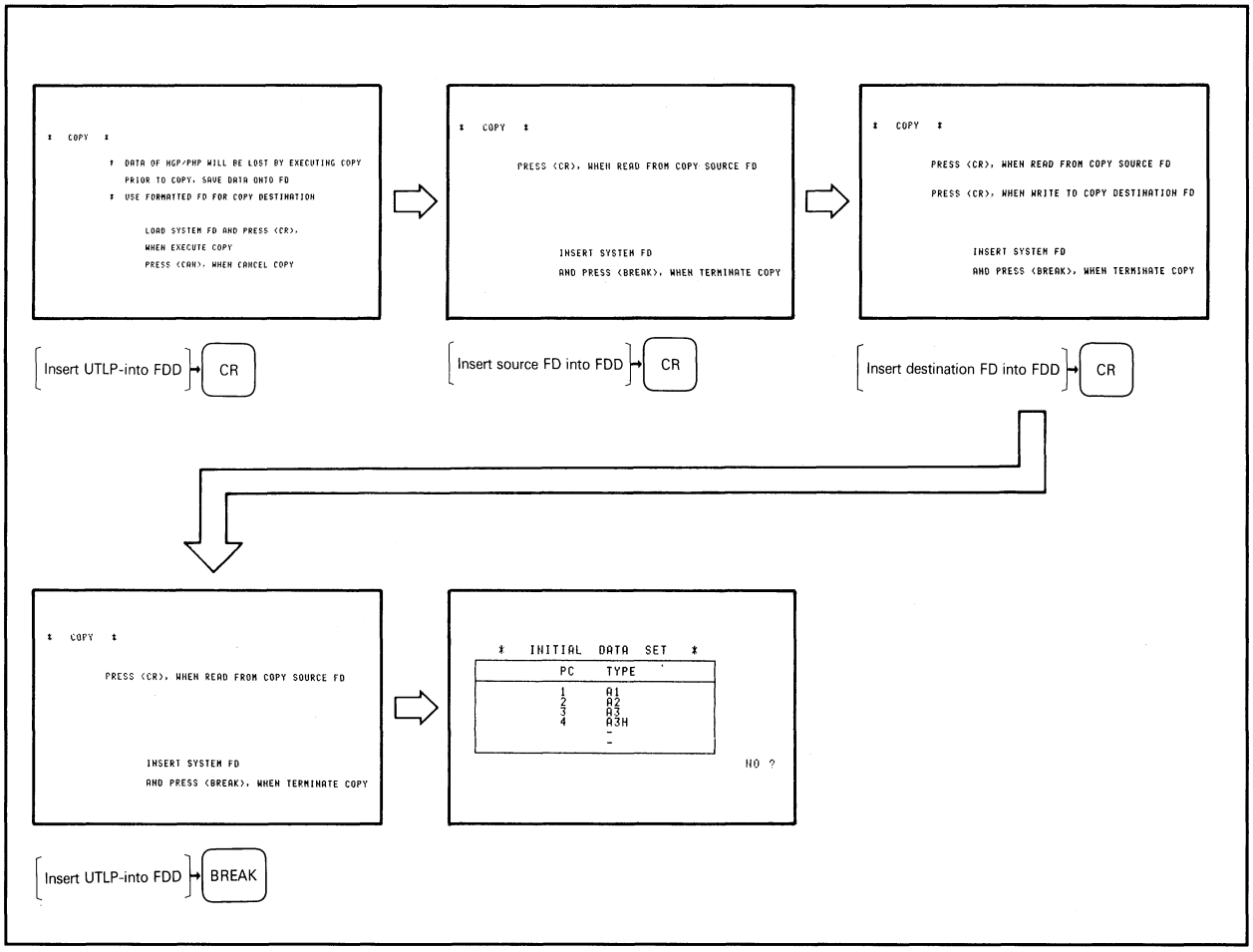
(5) During writing to the destination disk

Writing TRACK=[ ] ← Indicates the track number.

is displayed. The data is automatically verified upon completion.

Verifying TRACK=[ ] ← Indicates the track number.

(6) Copy operation sequence

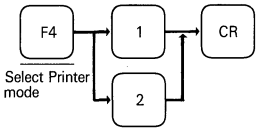


7.4 Printer Mode

Print out R and extension R data in the GPP/HGP/PHP memory.

7.4.1 Selecting function

BASIC OPERATION

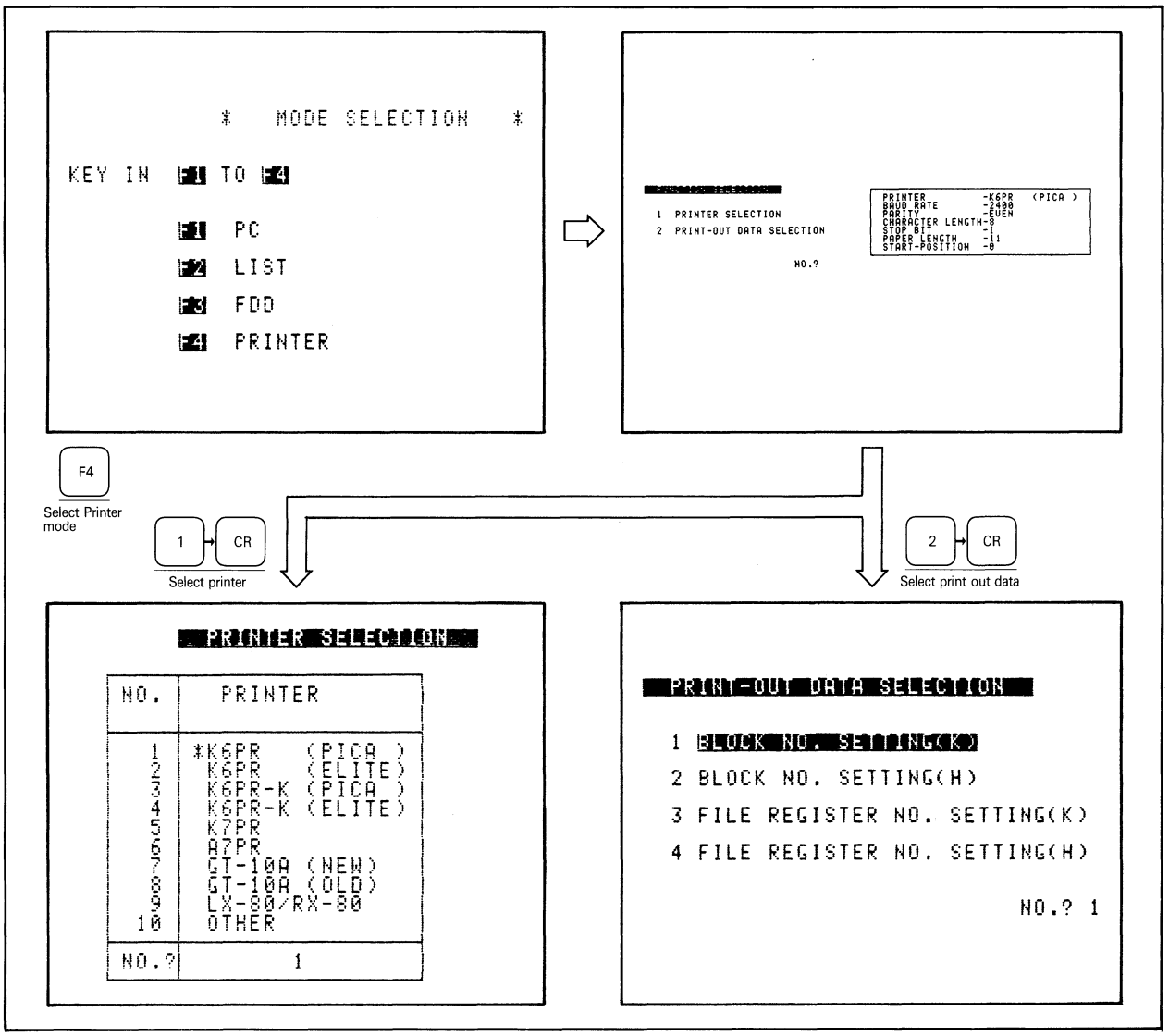


EXPLANATION

(1) Press [1], [CR] to change the printer displayed on the screen.
The default setting is K6PR.
Data of the printer used appears on the screen.

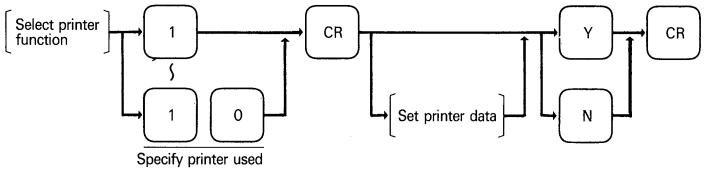
(2) Press [2], [CR] to specify the data printed out.

(3) Sequence of display screens:



7.4.2 Specifying the printer

BASIC OPERATION



EXPLANATION

(1) Type the number (1 to 10) corresponding to the printer used, and press CR.

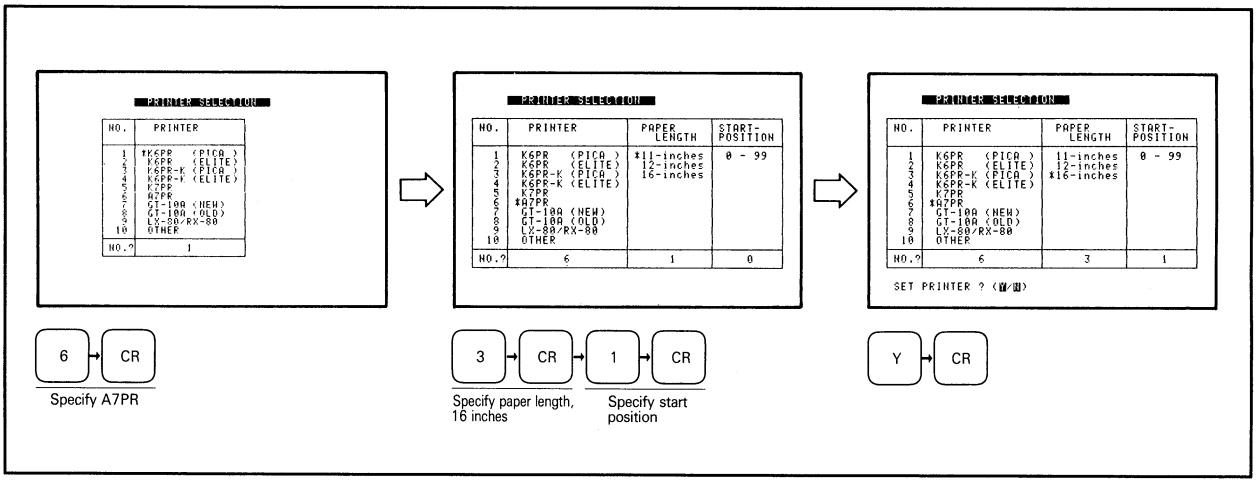
(2) When the printer data setting table appears, key in the corresponding number and press CR to move the cursor to the next data. "✳" indicates that the data has been set. (○ in the following table indicates an item to be specified.)

Printer	Printer Data Setting				
	Baud rate	Parity	Character length, stop bit	Paper length	Start position
K6PR (PICA)	—	—	—	—	—
K6PR (ELITE)	—	—	—	—	—
K6PR-K (PICA)	—	—	—	—	—
K6PR-K (ELITE)	—	—	—	—	—
K7PR	○	—	—	○	○
A7PR	—	—	—	○	○
GT-10A (NEW)	—	—	—	—	—
GT-10A (OLD)	—	—	—	—	—
LX-80/RX-80	—	—	—	—	—
General-purpose printer	○	○	○	○	○

(3) The message "SET PRINTER? (Y/N)" indicates that printer data setting is complete. Press Y, CR to define the printer setting and move to the print out data select screen. Press N, CR to return to the printer select screen.

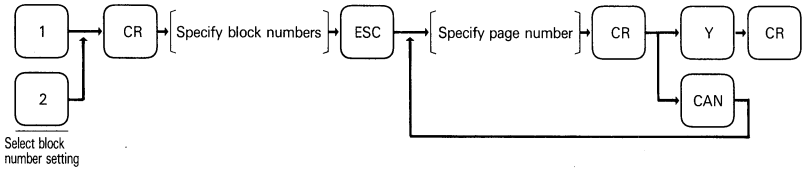
(4) The printer selected remains valid unless the GPP/HGP/PHP is powered down or reset.

(5) Sequence of display screens:



7.4.3 Selecting print out data

BASIC OPERATION 1 (BLOCK NUMBER SETTING)



EXPLANATION

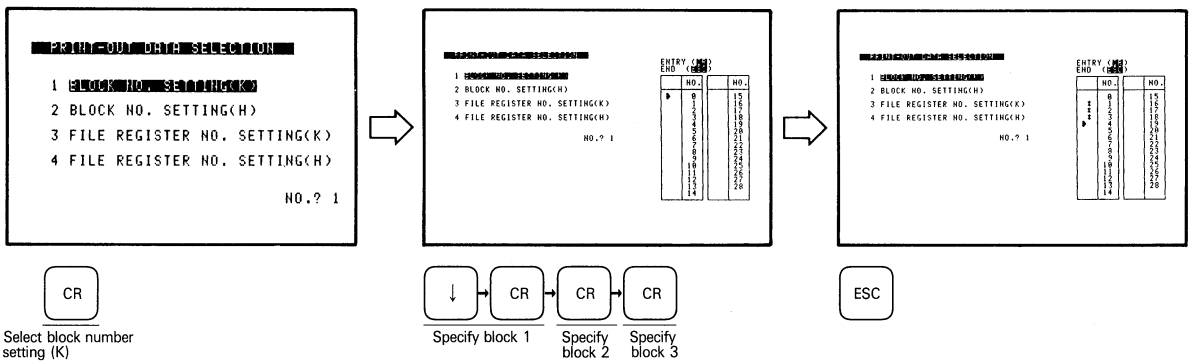
(1) Block number setting

(1) Type 1 or 2 and press CR to print out all data of the specified block. Pressing 1, CR selects print out data displayed in decimal, and 2, CR in hexadecimal.

(2) When the block number setting screen is displayed, move the cursor to the required block number and press CR. "✳" indicates that the setting of the current line is complete and the cursor moves to the next line.
Press HOME CLEAR to clear all "✳"s.
Press DEL to delete "✳" at the cursor.

(3) When the required block numbers have been specified, press ESC to move to the print out start page setting screen.

(4) Sequence of display screens:



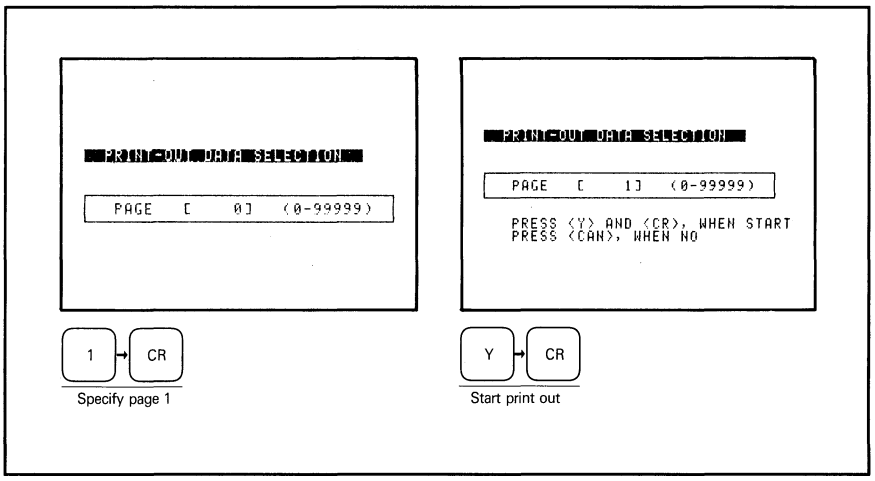
(2) Page number setting

(1) Type the print out start page (0 to 99999).
The page defaults to 0 and is incremented by 1 each time one page is printed out.

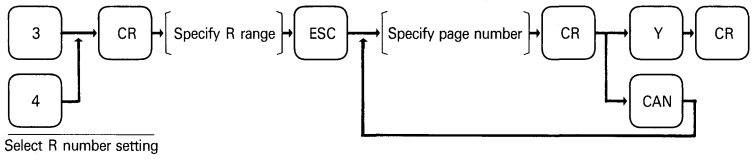
(2) When page setting is complete, a confirmation message is displayed.
Press Y, CR to start print out.
Press CAN to return the cursor to the page setting area.

(3) Press BREAK to stop print out.
Press CR to resume print out.
Press BREAK again to terminate print out.

(4) Sequence of display screens:



BASIC OPERATION 2 (R RANGE SETTING)



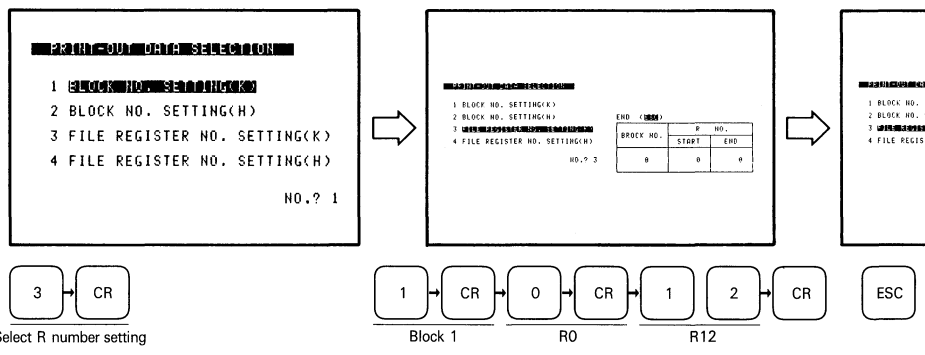
EXPLANATION

(1) Type 3 or 4 and press CR to print out the data of the specified block and R range. Pressing 3 , CR selects print out data displayed in decimal, and 4 , CR in hexadecimal.

(2) When the R number setting screen is displayed, specify the required block number and R number range (head and final numbers) and press CR . The R numbers specified must satisfy the following condition:
Head number ≦ final number ≦ 8191

(3) When the R number range has been specified, press ESC to call the print out start page setting screen.

(4) Sequence of display screens:



8. ERROR MESSAGES

8.1 Utility Program Error Messages

(1) The error flag M9091 is switched on to indicate an error in the processing of any of the UTLP-FN1 commands.

(2) The following error codes are written to D9091 when M9091 switches on.

Error Code	Command	Description
10050	BCHK	Invalid device specified in D9090.
10051	BCHK	Specified device is not D, W, R, T, C or A.
10052	BCHK	Specified bit is outside the allowed range (0 to 15).
10053	BCHK	Specified output device is not M, B or Y.
10054	BLMOV	Invalid device specified in D9090.
10055	BLMOV	Specified number of words transferred is outside the allowed range (0 to 8192).
10056	BLMOV	Invalid number specified for transfer source input data device.
10057	BLMOV	Invalid number specified for transfer destination input data device.
10058	BLMOV	Invalid transfer source device specified.
10059	BLMOV	Invalid transfer source offset value specified.
10060	BLMOV	Invalid transfer destination device specified.
10061	BLMOV	Invalid transfer destination offset value specified.
10062	RSET	Invalid device specified in D9090.
10063	RSET	Input data is outside the allowed range (0 to 28).
10064	RSET	Specified block not found.
10065	RBMOV	Invalid device specified in D9090.
10066	RBMOV	Specified number of words transferred is outside the allowed range (0 to 8192).
10067	RBMOV	Specified transfer source block number is outside the allowed range (0 to 28).
10068	RBMOV	Specified transfer source device number block is outside the allowed range (0 to 8191).
10069	RBMOV	Specified transfer destination block number is outside the allowed range (0 to 28).
10070	RBMOV	Specified transfer destination device number block is outside the allowed range (0 to 8191).
10071	RBMOV	Specified block protected.
10076	BINDA	Invalid device specified in D9090.
10077	BINHA	Invalid device specified in D9090.
10078	BCDDA	Invalid device specified in D9090.
10079	BCDDA	Input data is outside the allowed range (0 to 9).

Table 8.1 Error Code List (Continue)

Error Code	Command	Description
10080	ABCDBIN	Invalid device specified in D9090.
10081	ABCDBIN	Input data is outside the allowed range (0 to 9, "—").
10082	ABCDBIN	Input data range (—32768 to 32767) exceeded.
10083	ABIN	Invalid device specified in D9090.
10084	ABIN	Input data is outside the allowed range (0 to 9, A to F).
10085	ABCD	Invalid device specified in D9090.
10086	ABCD	Input data is outside the allowed range (0 to 9).
10087	DBINDA	Invalid device specified in D9090.
10088	DBINHA	Invalid device specified in D9090.
10089	DBCDDA	Invalid device specified in D9090.
10090	DBCDDA	Input data is outside the allowed range (0 to 9).
10091	DABCDBIN	Invalid device specified in D9090.
10092	DABCDBIN	Input data is outside the allowed range (0 to 9, "—").
10093	DABCDBIN	Input data range (—2147483648 to 2147483647) exceeded.
10094	DABIN	Invalid device specified in D9090.
10095	DABIN	Input data is outside the allowed range (0 to 9, A to F).
10096	DABCD	Invalid device specified in D9090.
10097	DABCD	Input data is outside the allowed range (0 to 9).
10098	COMRD	Invalid device specified in D9090.
10099	COMRD	Invalid comment entry device specified.
10100	COMRD	Invalid transfer destination device specified.
10101	RGCHG	Invalid device specified in D9090.
10102	RBCHG	Specified number of words exchanged is outside the allowed range (0 to 1024).
10103	RBCHG	Specified exchange (1) block number is outside the allowed range (0 to 28).
10104	RBCHG	Specified exchange (1) device number is outside the allowed range (0 to 8191).
10105	RBCHG	Specified exchange (2) block number is outside the allowed range (0 to 28).
10106	RBCHG	Specified exchange (2) device number is outside the allowed range (0 to 8191).
10107	RBCHG	Specified block protected.

Table 8.1 Error Code List

8.2 GPP/HGP/PHP Error Messages

The following error messages are displayed in the GPP/HGP/PHP screen message area.

No.	Error Message	Definition	Corrective Action
1	BLOCK NO. ERROR	Blocks 1 to 9: Invalid block number specified.	Specify an empty block number.
		Blocks 10, 11: A3MCA-12, 18, A3NMCA-16, 24, 40, 56 not used. *1	Use A3NMCA-16, 24, 40, 56.
		Blocks 12 to 20: A3MCA-18, A3NMCA-24, 40, 56 not used. *2	Use A3NMCA-40, 56 for blocks 13 to 20. Use A3NMCA-24, 40, 56 for block 12.
		Blocks 21 to 28: A3MCA-18, A3NMCA-24, 40, 56 not used or CPU is not A3HCPU. *3	Use A3NMCA-56 and A3HCPU.
2	CAPACITY NOT SET	Block 0 specified without setting file register capacity in the parameters.	Set file register capacity.
		A1, A1NCPU specified during PC selection.	Specify A2, A3, A2N, A3N or A3HCPU.
3	CHECK 'PRINTER'	Printer power or SEL switch off.	Switch on.
		No printer paper.	Set paper.
		Cable not connected.	Connect.
4	DEVICE NO. ERROR	File register number specified in list mode is outside the allowed range.	Specify valid number.
5	EXCESS OF DISK CAPACITY	Remaining disk memory capacity is too small, data cannot be written.	• Change disk. • Delete unnecessary data on disk.
6	EXCESS OF MICON-CAPACITY	The utility program capacity is larger than the microcomputer area capacity set in the parameters.	• Change the microcomputer capacity parameter. • Cancel unnecessary utility programs.
7	EXCESS OF PACKAGE MEMORY	Read data exeeds utility program capacity.	A maximum of 32 utility programs or 58K bytes can be read.
8	FILE MISSING	The specified file name cannot be found on the disk.	Read the directry and check for the file name.
9	FILE NAME ERROR	An invalid file name has been set during FD delete or directory operations.	Specify correct name.
10	FLOPPY DISK ERROR	No disk in accessed drive.	• Insert disk. • Change the drive number.
		Disk is in write forbid mode.	• Set the write protect tab to OK. • Change disk.
		Disk is defective.	Change disk.
11	MAIN PROGRAM NOT READ	The user program has not been read into the GPP/PHP/HGP memory.	Read the user progroam.
12	MEMORY CASSETTE CHECK	Memory cassette connected incorrectly.	Correct.
13	MEMORY CAPACITY EXCEEDED	GPP/PHP/HGP memory capacity is exceeded during read of file register data from disk.	Perform read to only read the area within memory capacity, or specify the head block number which meets data capacity on disk.
14	MEMORY PROTECT	Write was attempted to memory protected area. (Blocks 0 to 9. Blocks 10 and 11 when A3NMCA-16 is used)	Set memory protect switch in memory cassette to OFF.

Table 8.2 Error Messages (Continue)

No.	Error Message	Definition	Corrective Action
15	MICRO-CAPACITY NOT SET	Microcomputer capacity not set in the user program read in FDD mode.	Set microcomputer capacity in the parameters.
16	MORE THAN 10 ERRORS	More than 10 mismatches found during verify in PC mode.	Correct mismatches.
17	OPERATOR ERROR	An invalid key has been pressed.	Check operation.
18	PACKAGE NAME ALREADY USED	The utility program has already read been into the GPP/PHP/HGP memory.	It is not necessary to repeat.
19	PC COMMUNICATION ERROR	Cable is not connected or is defective.	Check cable. Change cable.
		PC CPU reset during communication.	Make communication again.
		PC power off.	Switch on.
20	PC SELECTION ERROR	The PC CPU set in the user program does not match the CPU specified in the initial data setting.	Select correct PC type.
21	READING ERROR	Data cannot be read from disk.	If data cannot be read a second time, format the disk.
22	SETTING ERROR	The entered data is outside the setting range.	Use the specified setting range.
23	SYSTEM NAME ALREADY USED	The destination system name used fow write or copy has already been used on the destination disk.	Continue operation to overwrite, or specify a new system name.
24	SYSTEM NAME ERROR	An invalid system name has been set in FDD mode.	Set a correct name.
25	VERIFY ERROR	GPP/PHP/HGP memory data does not match disk data during verification.	Check the data.
26	WRITE-IN ERROR	Data cannot be written onto disk.	Change disk.
27	WRITE-IN FORBIDDEN	Data written in PC mode with GPP WRITE-IN CPU switch in FORBID position.	Set WRITE-IN CPU switch to PERMIT.

Table 8.2 Error Messages

Note

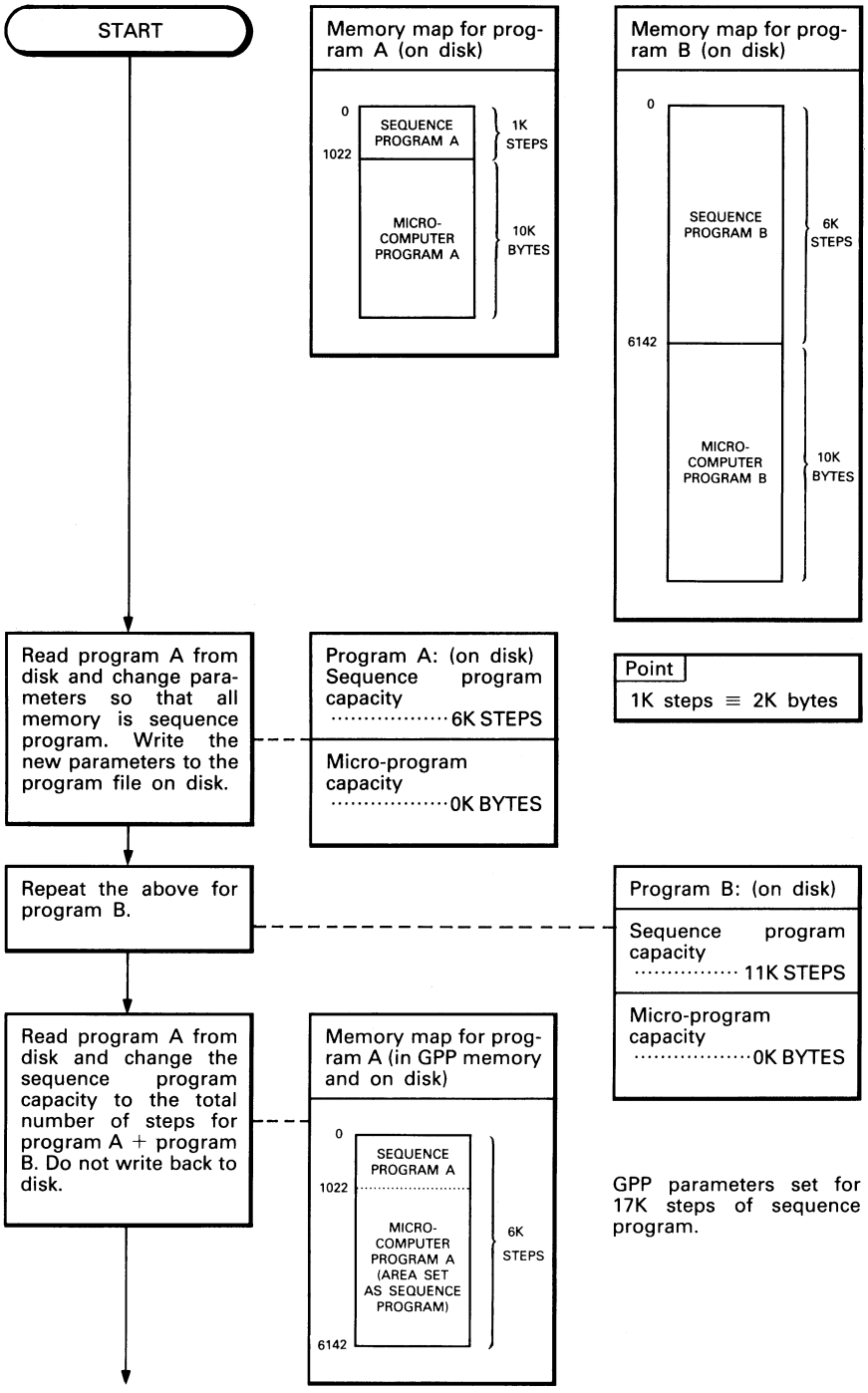
*1: Use of the A3MCA-12, 18 results in a block number error, which is not displayed.

*2: Use of the A3MCA-18, A3NMCA-24 for blocks 13 to 20 causes a block number error, which is not displayed.

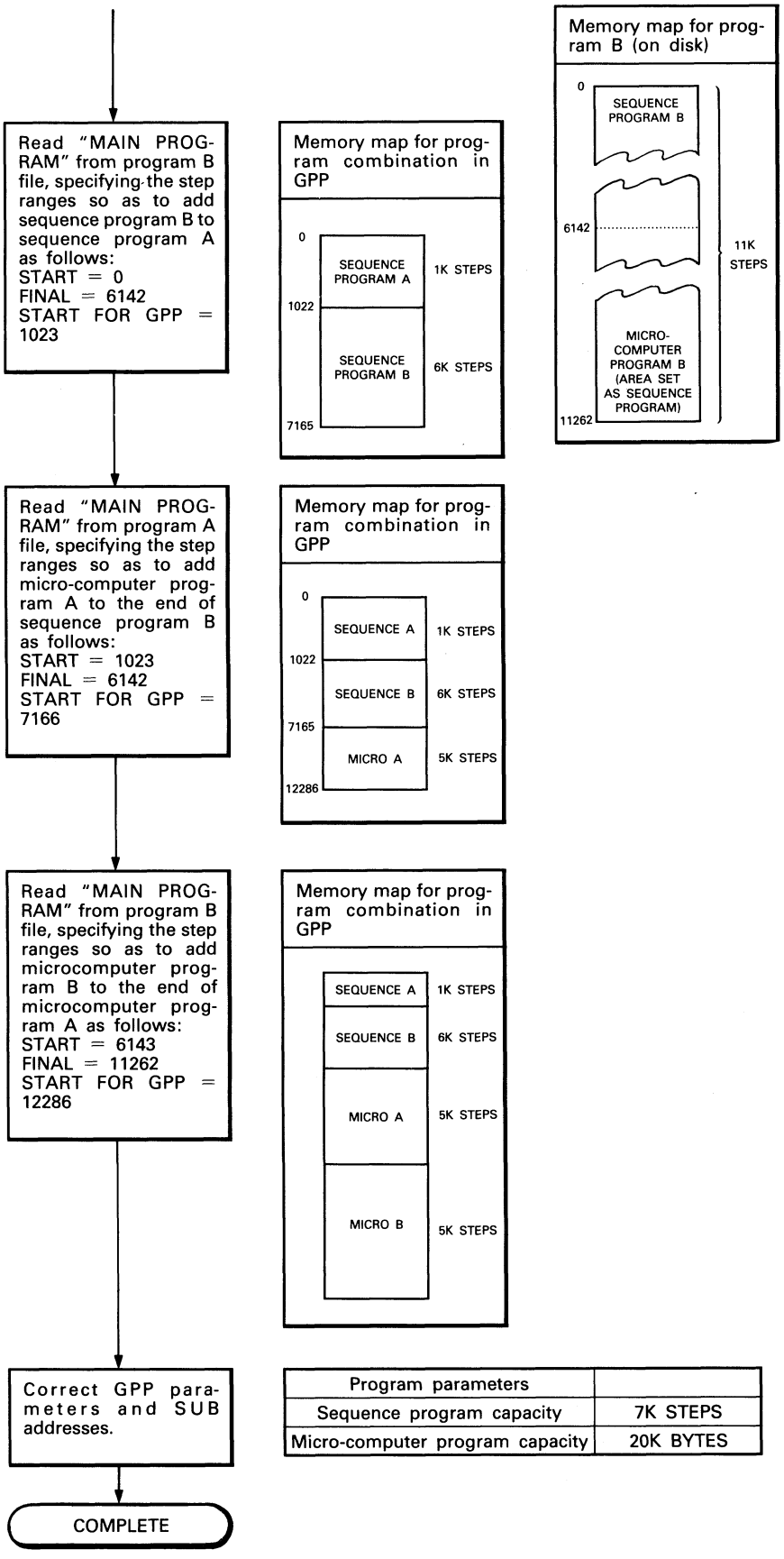
APPENDICES

APPENDIX 1 Combining Together Programs with Utility Programs

The following method may be used to edit together programs (which include utility packages) using only the GPP/PHP/HGP system software.
In the example that follows, there are two programs, A and B. Both contain a sequence program area and a microcomputer program area. This is illustrated as follows:



APP



Program parameters	
Sequence program capacity	7K STEPS
Micro-computer program capacity	20K BYTES

APP

APPENDIX 2 Hexadecimal-Decimal Conversion Table

A three-digit hexadecimal is represented as H_3 H_2 H_1.

H_3H_2 \ H_1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010	0011	0012	0013	0014	0015
01	0016	0017	0018	0019	0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	0030	0031
02	0032	0033	0034	0035	0036	0037	0038	0039	0040	0041	0042	0043	0044	0045	0046	0047
03	0048	0049	0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	0060	0061	0062	0063
04	0064	0065	0066	0067	0068	0069	0070	0071	0072	0073	0074	0075	0076	0077	0078	0079
05	0080	0081	0082	0083	0084	0085	0086	0087	0088	0089	0090	0091	0092	0093	0094	0095
06	0096	0097	0098	0099	0100	0101	0102	0103	0104	0105	0106	0107	0108	0109	0110	0111
07	0112	0113	0114	0115	0116	0117	0118	0119	0120	0121	0122	0123	0124	0125	0126	0127
08	0128	0129	0130	0131	0132	0133	0134	0135	0136	0137	0138	0139	0140	0141	0142	0143
09	0144	0145	0146	0147	0148	0149	0150	0151	0152	0153	0154	0155	0156	0157	0158	0159
0A	0160	0161	0162	0163	0164	0165	0166	0167	0168	0169	0170	0171	0172	0173	0174	0175
0B	0176	0177	0178	0179	0180	0181	0182	0183	0184	0185	0186	0187	0188	0189	0190	0191
0C	0192	0193	0194	0195	0196	0197	0198	0199	0200	0201	0202	0203	0204	0205	0206	0207
0D	0208	0209	0210	0211	0212	0213	0214	0215	0216	0217	0218	0219	0220	0221	0222	0223
0E	0224	0225	0226	0227	0228	0229	0230	0231	0232	0233	0234	0235	0236	0237	0238	0239
0F	0240	0241	0242	0243	0244	0245	0246	0247	0248	0249	0250	0251	0252	0253	0254	0255
10	0256	0257	0258	0259	0260	0261	0262	0263	0264	0265	0266	0267	0268	0269	0270	0271
11	0272	0273	0274	0275	0276	0277	0278	0279	0280	0281	0282	0283	0284	0285	0286	0287
12	0288	0289	0290	0291	0292	0293	0294	0295	0296	0297	0298	0299	0300	0301	0302	0303
13	0304	0305	0306	0307	0308	0309	0310	0311	0312	0313	0314	0315	0316	0317	0318	0319
14	0320	0321	0322	0323	0324	0325	0326	0327	0328	0329	0330	0331	0332	0333	0334	0335
15	0336	0337	0338	0339	0340	0341	0342	0343	0344	0345	0346	0347	0348	0349	0350	0351
16	0352	0353	0354	0355	0356	0357	0358	0359	0360	0361	0362	0363	0364	0365	0366	0367
17	0368	0369	0370	0371	0372	0373	0374	0375	0376	0377	0378	0379	0380	0381	0382	0383
18	0384	0385	0386	0387	0388	0389	0390	0391	0392	0393	0394	0395	0396	0397	0398	0399
19	0400	0401	0402	0403	0404	0405	0406	0407	0408	0409	0410	0411	0412	0413	0414	0415
1A	0416	0417	0418	0419	0420	0421	0422	0423	0424	0425	0426	0427	0428	0429	0430	0431
1B	0432	0433	0434	0435	0436	0437	0438	0439	0440	0441	0442	0443	0444	0445	0446	0447
1C	0448	0449	0450	0451	0452	0453	0454	0455	0456	0457	0458	0459	0460	0461	0462	0463
1D	0464	0465	0466	0467	0468	0469	0470	0471	0472	0473	0474	0475	0476	0477	0478	0479
1E	0480	0481	0482	0483	0484	0485	0486	0487	0488	0489	0490	0491	0492	0493	0494	0495
1F	0496	0497	0498	0499	0500	0501	0502	0503	0504	0505	0506	0507	0508	0509	0510	0511
20	0512	0513	0514	0515	0516	0517	0518	0519	0520	0521	0522	0523	0524	0525	0526	0527
21	0528	0529	0530	0531	0532	0533	0534	0535	0536	0537	0538	0539	0540	0541	0542	0543
22	0544	0545	0546	0547	0548	0549	0550	0551	0552	0553	0554	0555	0556	0557	0558	0559
23	0560	0561	0562	0563	0564	0565	0566	0567	0568	0569	0570	0571	0572	0573	0574	0575
24	0576	0577	0578	0579	0580	0581	0582	0583	0584	0585	0586	0587	0588	0589	0590	0591
25	0592	0593	0594	0595	0596	0597	0598	0599	0600	0601	0602	0603	0604	0605	0606	0607
26	0608	0609	0610	0611	0612	0613	0614	0615	0616	0617	0618	0619	0620	0621	0622	0623
27	0624	0625	0626	0627	0628	0629	0630	0631	0632	0633	0634	0635	0636	0637	0638	0639
28	0640	0641	0642	0643	0644	0645	0646	0647	0648	0649	0650	0651	0652	0653	0654	0655
29	0656	0657	0658	0659	0660	0661	0662	0663	0664	0665	0666	0667	0668	0669	0670	0671
2A	0672	0673	0674	0675	0676	0677	0678	0679	0680	0681	0682	0683	0684	0685	0686	0687
2B	0688	0689	0690	0691	0692	0693	0694	0695	0696	0697	0698	0699	0700	0701	0702	0703
2C	0704	0705	0706	0707	0708	0709	0710	0711	0712	0713	0714	0715	0716	0717	0718	0719
2D	0720	0721	0722	0723	0724	0725	0726	0727	0728	0729	0730	0731	0732	0733	0734	0735
2E	0736	0737	0738	0739	0740	0741	0742	0743	0744	0745	0746	0747	0748	0749	0750	0751
2F	0752	0753	0754	0755	0756	0757	0758	0759	0760	0761	0762	0763	0764	0765	0766	0767
30	0768	0769	0770	0771	0772	0773	0774	0775	0776	0777	0778	0779	0780	0781	0782	0783
31	0784	0785	0786	0787	0788	0789	0790	0791	0792	0793	0794	0795	0796	0797	0798	0799
32	0800	0801	0802	0803	0804	0805	0806	0807	0808	0809	0810	0811	0812	0813	0814	0815
33	0816	0817	0818	0819	0820	0821	0822	0823	0824	0825	0826	0827	0828	0829	0830	0831
34	0832	0833	0834	0835	0836	0837	0838	0839	0840	0841	0842	0843	0844	0845	0846	0847
35	0848	0849	0850	0851	0852	0853	0854	0855	0856	0857	0858	0859	0860	0861	0862	0863
36	0864	0865	0866	0867	0868	0869	0870	0871	0872	0873	0874	0875	0876	0877	0878	0879
37	0880	0881	0882	0883	0884	0885	0886	0887	0888	0889	0890	0891	0892	0893	0894	0895
38	0896	0897	0898	0899	0900	0901	0902	0903	0904	0905	0906	0907	0908	0909	0910	0911
39	0912	0913	0914	0915	0916	0917	0918	0919	0920	0921	0922	0923	0924	0925	0926	0927
3A	0928	0929	0930	0931	0932	0933	0934	0935	0936	0937	0938	0939	0940	0941	0942	0943
3B	0944	0945	0946	0947	0948	0949	0950	0951	0952	0953	0954	0955	0956	0957	0958	0959
3C	0960	0961	0962	0963	0964	0965	0966	0967	0968	0969	0970	0971	0972	0973	0974	0975
3D	0976	0977	0978	0979	0980	0981	0982	0983	0984	0985	0986	0987	0988	0989	0990	0991
3E	0992	0993	0994	0995	0996	0997	0998	0999	1000	1001	1002	1003	1004	1005	1006	1007
3F	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023

APPENDIX 3 Processing Time List

(1) Utility package commands

Command	Processing Time (ms)			Remarks
	A1(E), A2(E), A3(E)CPU	A1N, A2N, A3NCPU	A3HCPU	
BCHK	0.25	0.21	0.18	
BLMOV	0.35	0.29	0.24	Number of words transferred = 1
	21.0	16.4	6.5	Number of words transferred = 8192
RBMOV	0.52	0.42	0.33	Number of words transferred = 1
	21.5	16.6	6.6	Number of words transferred = 8192
RSET	0.26	0.22	0.19	
RBCHG	0.57	0.46	0.37	Number of words exchanged = 1
	13.8	11.0	6.9	Number of words exchanged = 1024
BINDA	0.57	0.46	0.36	Conversion data = 0000_H
	0.81	0.66	0.52	Conversion data = 8000_H
BINHA	0.27	0.23	0.19	
BCDDA	0.34	0.28	0.23	
DBINDA	0.58	0.47	0.36	Conversion data = 00000000_H
	0.91	0.73	0.56	Conversion data = 80000000_H
DBINHA	0.30	0.24	0.21	
DBCDDA	0.36	0.29	0.23	
ABCDBIN	0.78	0.64	0.53	
ABIN	0.24	0.21	0.18	
ABCD	0.26	0.22	0.19	
DABCDBIN	0.87	0.70	0.54	
DABIN	0.33	0.27	0.22	
DABCD	0.35	0.29	0.23	
COMRD	0.28	0.23	0.19	

(2) PC mode

(Seconds)

Number of Blocks	Read	Write*	Verify
1 (16K bytes)	23	23	24
All blocks (464K bytes)	625	624	654

*: Without automatic verify

APP

IB (NA) 66169-A

(3) FDD mode (file register)

(Seconds)

Number of Blocks	Read	Write*	Verify	Copy
1 (16K bytes)	6	13.5	6	29
All blocks (464K bytes)	70	140	93	311

(4) File register print out

Printer		Processing Time (seconds/page)
K6PR	Pica	75
	Elite	94
A7PR		19

IB (NA) 66169-A

APPENDIX 4 ASCII Code Table

Bit number b7	b6	b5	b4	b3	b2	b1	Column / Line	0	0	0	0	1	1	1	1
								0	0	1	1	0	0	1	1
								0	1	0	1	0	1	0	1
							0	1	2	3	4	5	6	7
			0	0	0	0	0	NUL	(TC7)DLE	SP	0	@	P	‘	p
			0	0	0	1	1	(TC1)SOH	DC1	!	1	A	Q	a	q
			0	0	1	0	2	(TC2)STX	DC2	"	2	B	R	b	r
			0	0	1	1	3	(TC3)ETX	DC3	#	3	C	S	c	s
			0	1	0	0	4	(TC4)EOT	DC4	$	4	D	T	d	t
			0	1	0	1	5	(TC5)ENQ	(TC8)NAK	%	5	E	U	e	u
			0	1	1	0	6	(TC6)ACK	(TC9)SYN	&	6	F	V	f	v
			0	1	1	1	7	BEL	(TC10)ETB	’	7	G	W	g	w
			1	0	0	0	8	FE0(BS)	CAN	(	8	H	X	h	x
			1	0	0	1	9	FE1(HT)	EM	)	9	I	Y	i	y
			1	0	1	0	10	FE2(LF/NL)	SUB	*	:	J	Z	j	z
			1	0	1	1	11	FE3(VT)	ESC	+	;	K	[	k	{
			1	1	0	0	12	FE4(FF)	IS4(FS)	,	<	L	\	l	|
			1	1	0	1	13	FE5(CR)	IS3(GS)	-	=	M	]	m	}
			1	1	1	0	14	SO	IS2(RS)	.	>	N	^	n	~
			1	1	1	1	15	SI	IS1(US)	/	?	O	_	o	DEL

ASCII Codes (Control codes)

NUL	(Blank)
SOH	(Start of Heading)
STX	(Start of Text)
ETX	(End of Text)
EOT	(End of Transmission)
ENQ	(Enquiry)
ACK	(Acknowledge 〈Positive〉)
BEL	(Bell)
BS	(Backspace)
HT	(Horizontal Tabulation)
LF	(Line Feed)
VT	(Vertical Tabulation)
FF	(Form Feed)
CR	(Carriage Return)
SO	(Shift Out)
SI	(Shift In)
DLE	(Data Link Escape)
DC1	(Device Control 1)
DC2	(Device Control 2)
DC3	(Device Control 3)
DC4	(Device Control 4-Stop)
NAK	(Negative Acknowledge)
SYN	(Synchronous Idle)
ETB	(End of Transmission Block)
CAN	(Cancel)
EM	(End of Medium)
SUB	(Substitute Character)
ESC	(Escape)
FS	(File Separator)
GS	(Group Separator)
RS	(Record Separator)
US	(Unit Separator)
SP	(Space)
DEL	(Delete/Rubout)

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 satisfy yourself as to its suitability for your specific application.

Utility Software Package type SWOGHP-UTLPC-FN1

Operating Manual

MODEL	SW0-UTLPC-FN1-O-E
MODEL CODE	13J715
IB(NA)-66169-B(8909)MEE	

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE : 1-8-12, OFFICE TOWER Z 14F HARUMI CHUO-KU 104-6212,JAPAN
NAGOYA WORKS : 1-14 , YADA-MINAMI 5 , HIGASHI-KU, NAGOYA , JAPAN