To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

USER'S MANUAL

RENESAS

μSxxxxAS17120 DEVICE FILE

PC-9800 SERIES(MS-DOS™)BASE
IBM PC/AT™(PC DOS™)BASE

Version V1

AS17120(V1)
AS17121(V1)
AS17132(V1)
AS17133(V1)

Document No. EEU−1464
(O. D. No. EEU−907)
Date Published December 1993 P
Printed in Japan

USER'S MANUAL

NEC

μSxxxxAS17120 DEVICE FILE

PC-9800 SERIES(MS-DOS™)BASE
IBM PC/AT™(PC DOS™)BASE

Version V1

AS17120(V1)
AS17121(V1)
AS17132(V1)
AS17133(V1)

SIMPLEHOST™ is a trademark of NEC Corporation.

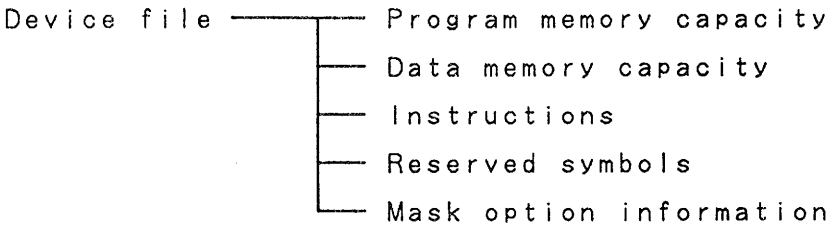
MS-DOS™ is a trademark of Microsoft Corporation.

PC/AT™ and PC DOS™ are trademarks of IBM Corporation.

PREFACE

AS17120, AS17121, AS17132, and AS17133 are device files for assembling programs for the uPD17120, uPD17121, uPD17132, and uPD17133. These device files are used together with the AS17K assembler.

The files contain the following information.



Refer to "AS17K Assembler User's Manual" (EEU-603) for the operation of the AS17K assembler and device files for the uPD17120, uPD17121, uPD17132, and uPD17133.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 DEVICE INFORMATION

Device files, AS17120, AS17121, AS17132, and AS17133, provide the following information about the uPD17120, uPD17121, uPD17132, and uPD17133 in assembling.

(1) Program memory (ROM) capacity

uPD17120 and uPD17121: 768 x 16 bits (0000H to 02FFH)

uPD17132 and uPD17133: 1024 x 16 bits (0000H to 03FFH)

(2) Data memory (RAM) capacity

uPD17120 and uPD17121: 64 x 4 bits

uPD17132 and uPD17133: 111 x 4 bits

(3) Instructions

See Chapter 2.

(4) Information about read/write for port registers and system registers

See Chapter 3.

(5) Reserved symbols

See Chapter 3.

(6) Device numbers and SE board numbers

Each device file has the number of the device and the number of a SE board which is the most suitable for developing programs for the device. These numbers are also output to the ICE and PRO files by the AS17K assembler. The in-circuit emulator uses these numbers when checking the device development environment and mask products to be ordered.

Table 1-1 Correspondence between Device Files and Device and SE Board Numbers

Device file (version)	Device	Device number	SE board number	SE board
AS17120 (V1)	uPD17120	2C	2C	SE-17120
AS17121 (V1)	uPD17121	2D		
AS17132 (V1)	uPD17132	2E		
AS17133 (V1)	uPD17133	2F		

CHAPTER 2 INSTRUCTION SET

2.1 Overview of the Instruction Set

b_{14}-b_{11} \ b_{15}		0		1	
BIN	HEX				
0 0 0 0	0	ADD	r, m	ADD	m, #n4
0 0 0 1	1	SUB	r, m	SUB	m, #n4
0 0 1 0	2	ADDC	r, m	ADDC	m, #n4
0 0 1 1	3	SUBC	r, m	SUBC	m, #n4
0 1 0 0	4	AND	r, m	AND	m, #n4
0 1 0 1	5	XOR	r, m	XOR	m, #n4
0 1 1 0	6	OR	r, m	OR	m, #n4
0 1 1 1	7	INC INC RORC MOVT PUSH POP PEEK POKE GET PUT BR CALL RET RETSK RETI EI DI STOP HALT NOP	AR IX r DBF, @AR AR AR WR, rf rf, WR DBF, p p, DBF @AR @AR s h		
1 0 0 0	8	LD	r, m	ST	m, r
1 0 0 1	9	SKE	m, #n4	SKGE	m, #n4
1 0 1 0	A	MOV	@r, m	MOV	m, @r
1 0 1 1	B	SKNE	m, #n4	SKLT	m, #n4
1 1 0 0	C	BR	addr	CALL	addr
1 1 0 1	D			MOV	m, #n4
1 1 1 0	E			SKT	m, #n
1 1 1 1	F			SKF	m, #n

2.2 Legend

AR:	Address register
$(AR)_{ROM}$:	Contents of ROM at the address specified by the AR
ASR:	Address stack register pointed to by the stack pointer
addr:	Program memory address (11 bits)
BANK:	Bank register
CMP:	Compare flag
CY:	Carry flag
DBF:	Data buffer
h:	Halt release condition
INTEF:	Interrupt enable flag
INTR:	Registers automatically saved into the stack when an interrupt occurs
INTSK:	Interrupt stack register
IX:	Index register
IXE:	Index enable flag
MP:	Data memory row address pointer
MPE:	Memory pointer enable flag
m:	Data memory address indicated by m_R and m_C
m_R:	Data memory row address (high-order)
m_C:	Data memory column address (low-order)
n:	Bit position (four bits)
n4:	Immediate data (four bits)
PC:	Program memory counter
p:	Peripheral address
p_H:	Peripheral address (high-order three bits)
p_L:	Peripheral address (low-order four bits)
RP:	General register pointer
r:	General register column address
rf:	Register file address
rf_R:	Register file address (high-order three bits)
rf_C:	Register file address (low-order four bits)
SP:	Stack pointer
s:	Stop release condition
WR:	Window register
(x):	Contents of memory addressed by x x: Direct address such as m or r, or a register such as the ASR

2.3 Instruction List

Instruction set	Mnemonic	Operand	Operation	Machine code			
				Op code	Operand		
Add	ADD	r,m	(r) ← (r) + (m)	00000	m_R	m_C	r
		m,#n4	(m) ← (m) + n4	10000	m_R	m_C	n4
	ADDC	r,m	(r) ← (r) + (m) + CY	00010	m_R	m_C	r
		m,#n4	(m) ← (m) + n4 + CY	10010	m_R	m_C	n4
	INC	AR	AR ← AR + 1	00111	000	1001	0000
		IX	IX ← IX + 1	00111	000	1000	0000
Subtract	SUB	r,m	(r) ← (r) - (m)	00001	m_R	m_C	r
		m,#n4	(m) ← (m) - n4	10001	m_R	m_C	n4
	SUBC	r,m	(r) ← (r) - (m) - CY	00011	m_R	m_C	r
		m,#n4	(m) ← (m) - n4 - CY	10011	m_R	m_C	n4
Logical operation	OR	r,m	(r) ← (r) ∨ (m)	00110	m_R	m_C	r
		m,#n4	(m) ← (m) ∨ n4	10110	m_R	m_C	n4
	AND	r,m	(r) ← (r) ∧ (m)	00100	m_R	m_C	r
		m,#n4	(m) ← (m) ∧ n4	10100	m_R	m_C	n4
	XOR	r,m	(r) ← (r) ∀ (m)	00101	m_R	m_C	r
		m,#n4	(m) ← (m) ∀ n4	10101	m_R	m_C	n4
Test	SKT	m,#n	CMP ← 0, if (m) ∧ n = n, then skip	11110	m_R	m_C	n
	SKF	m,#n	CMP ← 0, if (m) ∧ n = 0, then skip	11111	m_R	m_C	n
Compare	SKE	m,#n4	(m) - n4, skip if zero	01001	m_R	m_C	n4
	SKNE	m,#n4	(m) - n4, skip if not zero	01011	m_R	m_C	n4
	SKGE	m,#n4	(m) - n4, skip if not borrow	11001	m_R	m_C	n4
	SKLT	m,#n4	(m) - n4, skip if borrow	11011	m_R	m_C	n4

(to be continued)

(Cont'd)

Instruction set	Mnemonic	Operand	Operation	Machine code			
				Op code	Operand		
Shift	RORC	r	$\rightarrow CY \rightarrow (r)_{b3} \rightarrow (r)_{b2} \rightarrow (r)_{b1} \rightarrow (r)_{b0} \rightarrow$	00111	000	0111	r
Transfer	LD	r,m	(r) ← (m)	01000	m_R	m_C	r
	ST	m,r	(m) ← (r)	11000	m_R	m_C	r
	MOV	@r,m	if MPE = 1: (MP,(r)) ← (m) if MPE = 0: (BANK, m_R,(r)) ← (m)	01010	m_R	m_C	r
		m,@r	if MPE = 1: (m) ← (MP,(r)) if MPE = 0: (m) ← (BANK,m_R,(r))	11010	m_R	m_C	r
		m,#n4	(m) ← n4	11101	m_R	m_C	n4
	MOVT	DBF,@AR	SP ← SP - 1, ASR ← PC, PC ← AR, DBF ← $(AR)_{ROM}$, PC ← ASR, SP ← SP + 1	00111	000	0001	0000
	PUSH	AR	SP ← SP - 1, ASR ← AR	00111	000	1101	0000
	POP	AR	AR ← ASR, SP ← SP + 1	00111	000	1100	0000
	PEEK	WR,rf	(WR) ← (rf)	00111	rf_R	0011	rf_C
	POKE	rf,WR	(rf) ← WR	00111	rf_R	0010	rf_C
	GET	DBF,p	DBF ← (p)	00111	p_H	1011	p_L
	PUT	p,DBF	(p) ← DBF	00111	p_H	1010	p_L
Branch	BR	addr	PC_{10-0} ← addr	01100	addr		
		@AR	PC ← AR	00111	000	0100	0000
Subroutine	CALL	addr	SP ← SP - 1, ASR ← PC + 1, PC_{10-0} ← addr	11100	addr		
		@AR	SP ← SP - 1, ASR ← PC + 1, PC ← AR	00111	000	0101	0000
	RET		PC ← ASR, SP ← SP + 1	00111	000	1110	0000
	RETSK		PC ← ASR, SP ← SP + 1 and skip	00111	001	1110	0000
	RETI		PC ← ASR, INTR ← INTSK, SP ← SP + 1	00111	100	1110	0000

(to be continued)

(Cont'd)

Instruction set	Mnemonic	Operand	Operation	Machine code			
				Op code	Operand		
Interrupt	EI		INTEF ← 1	00111	000	1111	0000
	DI		INTEF ← 0	00111	001	1111	0000
Others	STOP	s	STOP	00111	010	1111	s
	HALT	h	HALT	00111	011	1111	h
	NOP		No operation	00111	100	1111	0000

2.4 Macro Instructions Built in the Assembler (AS17K)

Legend

flagn: FLG-type symbol

< >: Omissible

	Mnemonic	Operand	Operation	n
Built-in macros	SKTn	flag1, ...flagn	if (flag1) to (flagn) = all "1", then skip	1≦n≦4
	SKFn	flag1, ...flagn	if (flag1) to (flagn) = all "0", then skip	1≦n≦4
	SETn	flag1, ...flagn	(flag1) to (flagn) ← 1	1≦n≦4
	CLRn	flag1, ...flagn	(flag1) to (flagn) ← 0	1≦n≦4
	NOTn	flag1, ...flagn	if (flagn) = "0", then (flagn) ← 1 if (flagn) = "1", then (flagn) ← 0	1≦n≦4
	INITFLG	<Not> flag1, ...<<Not> flagn>	if description = NOT flagn, then (flagn) ← 0 if description = flagn, then (flagn) ← 1	1≦n≦4

CHAPTER 3 RESERVED SYMBOLS

The reserved symbols defined in the device files, AS17120, AS17121, AS17132, and AS17133, are listed on the subsequent pages.

The reserved symbols are defined for the following buffers and registers:

- Data buffers (DBF)
- System registers (SYSREG)
- Port registers
- Register files (control registers)
- Peripheral registers

3.1 Data Buffers (DBF)

Symbolic name	Attribute	Value	Read/ write	Description
DBF3	MEM	0.0CH	R/W	DBF bits 15 to 12
DBF2	MEM	0.0DH	R/W	DBF bits 11 to 8
DBF1	MEM	0.0EH	R/W	DBF bits 7 to 4
DBF0	MEM	0.0FH	R/W	DBF bits 3 to 0

3.2 System Registers (SYSREG)

Symbolic name	Attribute	Value	Read/ write	Description
AR3	MEM	0.74H	R	Bits 15 to 12 of the address register
AR2	MEM	0.75H	R/W	Bits 11 to 8 of the address register
AR1	MEM	0.76H	R/W	Bits 7 to 4 of the address register
AR0	MEM	0.77H	R/W	Bits 3 to 0 of the address register
WR	MEM	0.78H	R/W	Window register
BANK	MEM	0.79H	R/W	Bank register
IXH	MEM	0.7AH	R/W	Index register high
MPH	MEM	0.7AH	R/W	Data memory row address pointer high
MPE	FLG	0.7AH.3	R/W	Memory pointer enable flag
IXM	MEM	0.7BH	R/W	Index register middle
MPL	MEM	0.7BH	R/W	Data memory row address pointer low
IXL	MEM	0.7CH	R/W	Index register low
RPH	MEM	0.7DH	R/W	General register pointer high
RPL	MEM	0.7EH	R/W	General register pointer low
PSW	MEM	0.7FH	R/W	Program status word
BCD	FLG	0.7EH.0	R/W	BCD flag
CMP	FLG	0.7FH.3	R/W	Compare flag
CY	FLG	0.7FH.2	R/W	Carry flag
Z	FLG	0.7FH.1	R/W	Zero flag
IXE	FLG	0.7FH.0	R/W	Index enable flag

3.3 Port Registers

Symbolic name	Attribute	Value	Read/ write	Description
P0E1	FLG	0.6FH.1	R/W	Port 0E bit 1
P0E0	FLG	0.6FH.0	R/W	Port 0E bit 0
P0A3	FLG	0.70H.3	R/W	Port 0A bit 3
P0A2	FLG	0.70H.2	R/W	Port 0A bit 2
P0A1	FLG	0.70H.1	R/W	Port 0A bit 1
P0A0	FLG	0.70H.0	R/W	Port 0A bit 0
P0B3	FLG	0.71H.3	R/W	Port 0B bit 3
P0B2	FLG	0.71H.2	R/W	Port 0B bit 2
P0B1	FLG	0.71H.1	R/W	Port 0B bit 1
P0B0	FLG	0.71H.0	R/W	Port 0B bit 0
P0C3	FLG	0.72H.3	R/W	Port 0C bit 3
P0C2	FLG	0.72H.2	R/W	Port 0C bit 2
P0C1	FLG	0.72H.1	R/W	Port 0C bit 1
P0C0	FLG	0.72H.0	R/W	Port 0C bit 0
P0D3	FLG	0.73H.3	R/W	Port 0D bit 3
P0D2	FLG	0.73H.2	R/W	Port 0D bit 2
P0D1	FLG	0.73H.1	R/W	Port 0D bit 1
P0D0	FLG	0.73H.0	R/W	Port 0D bit 0

3.4 Register Files (Control Registers)

(1) AS17120 and AS17121

Symbolic name	Attribute	Value	Read/ write	Description
SP	MEM	0.81H	R/W	Stack pointer
SIOEN	FLG	0.8BH.0	R/W	SIO enable flag
INT	FLG	0.8FH.0	R	INT pin status flag
PDRESEN	FLG	0.90H.0	R/W	Power-down reset enable flag
TMEN	FLG	0.91H.3	R/W	Timer enable flag
TMRES	FLG	0.91H.2	R/W	Timer reset flag
TMCK1	FLG	0.91H.1	R/W	Timer source clock selection flag bit 1
TMCK0	FLG	0.91H.0	R/W	Timer source clock selection flag bit 0
TMOSEL	FLG	0.92H.0	R/W	POD3/$\overline{\text{TMOUT}}$ selection flag
SIOTS	FLG	0.9AH.3	R/W	SIO start flag
SIOHIZ	FLG	0.9AH.2	R/W	SO pin status
SIOCK1	FLG	0.9AH.1	R/W	SIO source clock selection flag bit 1
SIOCK0	FLG	0.9AH.0	R/W	SIO source clock selection flag bit 0
IEGMD1	FLG	0.9FH.1	R/W	INT pin edge detection selection flag bit 1
IEGMD0	FLG	0.9FH.0	R/W	INT pin edge detection selection flag bit 0
POBGIO	FLG	0.A4H.0	R/W	POB group input/output selection flag (1 = All POBs are output ports.)
IPSIO	FLG	0.AFH.2	R/W	SIO interrupt enable flag
IPTM	FLG	0.AFH.1	R/W	Timer interrupt enable flag
IP	FLG	0.AFH.0	R/W	INT pin interrupt enable flag

(to be continued)

(Cont'd)

Symbolic name	Attribute	Value	Read/ write	Description
P0EB101	FLG	0.B2H.1	R/W	$P0E_1$ input/output selection flag (1 = output port)
P0EB100	FLG	0.B2H.0	R/W	$P0E_0$ input/output selection flag (1 = output port)
P0DB103	FLG	0.B3H.3	R/W	$P0D_3$ input/output selection flag (1 = output port)
P0DB102	FLG	0.B3H.2	R/W	$P0D_2$ input/output selection flag (1 = output port)
P0DB101	FLG	0.B3H.1	R/W	$P0D_1$ input/output selection flag (1 = output port)
P0DB100	FLG	0.B3H.0	R/W	$P0D_0$ input/output selection flag (1 = output port)
P0CB103	FLG	0.B4H.3	R/W	$P0C_3$ input/output selection flag (1 = output port)
P0CB102	FLG	0.B4H.2	R/W	$P0C_2$ input/output selection flag (1 = output port)
P0CB101	FLG	0.B4H.1	R/W	$P0C_1$ input/output selection flag (1 = output port)
P0CB100	FLG	0.B4H.0	R/W	$P0C_0$ input/output selection flag (1 = output port)
P0AB103	FLG	0.B5H.3	R/W	$P0A_3$ input/output selection flag (1 = output port)
P0AB102	FLG	0.B5H.2	R/W	$P0A_2$ input/output selection flag (1 = output port)
P0AB101	FLG	0.B5H.1	R/W	$P0A_1$ input/output selection flag (1 = output port)
P0AB100	FLG	0.B5H.0	R/W	$P0A_0$ input/output selection flag (1 = output port)
IRQSIO	FLG	0.BDH.0	R/W	SIO interrupt request flag
IRQTM	FLG	0.BEH.0	R/W	Timer interrupt request flag
IRQ	FLG	0.BFH.0	R/W	INT pin interrupt request flag

(2) AS17132 and AS17133

Symbolic name	Attribute	Value	Read/ write	Description
SP	MEM	0.81H	R/W	Stack pointer
SIOEN	FLG	0.8AH.0	R/W	SIO enable flag
INT	FLG	0.8FH.0	R	INT pin status flag
PDRESEN	FLG	0.90H.0	R/W	Power-down reset enable flag
TMEN	FLG	0.91H.3	R/W	Timer enable flag
TMRES	FLG	0.91H.2	R/W	Timer reset flag
TMCK1	FLG	0.91H.1	R/W	Timer source clock selection flag bit 1
TMCK0	FLG	0.91H.0	R/W	Timer source clock selection flag bit 0
TMOSEL	FLG	0.92H.0	R/W	$POD_3/\overline{TMOUT}$ selection flag
SIOTS	FLG	0.9AH.3	R/W	SIO start flag
SIOHIZ	FLG	0.9AH.2	R/W	SO pin status
SIOCK1	FLG	0.9AH.1	R/W	SIO source clock selection flag bit 1
SIOCK0	FLG	0.9AH.0	R/W	SIO source clock selection flag bit 0
CMPCH1	FLG	0.9CH.1	R/W	Comparator input channel selection flag bit 1
CMPCH0	FLG	0.9CH.0	R/W	Comparator input channel selection flag bit 0
CMPVREF3	FLG	0.9DH.3	R/W	Comparator reference voltage selection flag bit 3
CMPVREF2	FLG	0.9DH.2	R/W	Comparator reference voltage selection flag bit 2
CMPVREF1	FLG	0.9DH.1	R/W	Comparator reference voltage selection flag bit 1
CMPVREF0	FLG	0.9DH.0	R/W	Comparator reference voltage selection flag bit 0

(to be continued)

(Cont'd)

Symbolic name	Attribute	Value	Read/ write	Description
CMPSTRT	FLG	0.9EH.1	R/W	Comparator start flag
CMPRSLT	FLG	0.9EH.1	R	Comparator result flag
IEGMD1	FLG	0.9FH.1	R/W	INT pin edge detection selection flag bit 1
IEGMD0	FLG	0.9FH.0	R/W	INT pin edge detection selection flag bit 0
POC3IDI	FLG	0.A3H.3	R/W	POC_3 input port disable flag (POC_3/Cin_3 selection)
POC2IDI	FLG	0.A3H.2	R/W	POC_2 input port disable flag (POC_2/Cin_2 selection)
POC1IDI	FLG	0.A3H.1	R/W	POC_1 input port disable flag (POC_1/Cin_1 selection)
POC0IDI	FLG	0.A3H.0	R/W	POC_0 input port disable flag (POC_0/Cin_0 selection)
POBGIO	FLG	0.A4H.0	R/W	POB group input/output selection flag (1 = All POBs are output ports.)
IPSIO	FLG	0.AFH.2	R/W	SIO interrupt enable flag
IPTM	FLG	0.AFH.1	R/W	Timer interrupt enable flag
IP	FLG	0.AFH.0	R/W	INT pin interrupt enable flag
POEBIO1	FLG	0.B2H.1	R/W	POE_1 input/output selection flag (1 = output port)
POEBIO0	FLG	0.B2H.0	R/W	POE_0 input/output selection flag (1 = output port)
PODBIO3	FLG	0.B3H.3	R/W	POD_3 input/output selection flag (1 = output port)
PODBIO2	FLG	0.B3H.2	R/W	POD_2 input/output selection flag (1 = output port)
PODBIO1	FLG	0.B3H.1	R/W	POD_1 input/output selection flag (1 = output port)
PODBIO0	FLG	0.B3H.0	R/W	POD_0 input/output selection flag (1 = output port)

(to be continued)

(Cont'd)

Symbolic name	Attribute	Value	Read/ write	Description
P0CBIO3	FLG	0.B4H.3	R/W	$P0C_3$ input/output selection flag (1 = output port)
P0CBIO2	FLG	0.B4H.2	R/W	$P0C_2$ input/output selection flag (1 = output port)
P0CBIO1	FLG	0.B4H.1	R/W	$P0C_1$ input/output selection flag (1 = output port)
P0CBIO0	FLG	0.B4H.0	R/W	$P0C_0$ input/output selection flag (1 = output port)
P0ABIO3	FLG	0.B5H.3	R/W	$P0A_3$ input/output selection flag (1 = output port)
P0ABIO2	FLG	0.B5H.2	R/W	$P0A_2$ input/output selection flag (1 = output port)
P0ABIO1	FLG	0.B5H.1	R/W	$P0A_1$ input/output selection flag (1 = output port)
P0ABIO0	FLG	0.B5H.0	R/W	$P0A_0$ input/output selection flag (1 = output port)
IRQSIO	FLG	0.BDH.0	R/W	SIO interrupt request flag
IRQTM	FLG	0.BEH.0	R/W	Timer interrupt request flag
IRQ	FLG	0.BFH.0	R/W	INT pin interrupt request flag

3.5 Peripheral Registers

Symbolic name	Attribute	Value	Read/ write	Description
SIOSFR	DAT	01H	R/W	Peripheral address of the shift register
TMC	DAT	02H	W	Peripheral address of the timer count register
TMM	DAT	03H	W	Peripheral address of the timer modulo register
DBF	DAT	0FH	R/W	Peripheral address of the GET/PUT instruction data buffer
IX	DAT	01H	R/W	Peripheral address of the INC instruction index register
AR	DAT	40H	R/W	Peripheral address of the address register for GET, PUT, PUSH, CALL, BR, MOVT, and INC instructions

3.6 Reserved Words (In Alphabetic Order)

3.6.1 Instructions and pseudo instructions

ADD	EXTRN	NIBBLE6V	SET2
ADDC	FLG	NIBBLE7	SET3
AND	GET	NIBBLE7V	SET4
BANK0	GLOBAL	NIBBLE8	SFCOND
BELOW	HALT	NIBBLE8V	SKE
BR	IF	NOBMAC	SKF
C14344	IFCHAR	NOLIST	SKF1
C4444	IFNCHAR	NOMAC	SKF2
CALL	INC	NOP	SKF3
CASE	INCLUDE	NOT1	SKF4
CLR1	INITFLG	NOT2	SKGE
CLR2	IRP	NOT3	SKLT
CLR3	LAB	NOT4	SKNE
CLR4	LBMAC	OBMAC	SKT
CSEG	LD	OMAC	SKT1
DAT	LFCOND	OPTION	SKT2
DB	LIST	OR	SKT3
DI	LITERAL	ORG	SKT4
DW	LMAC	OTHER	SMAC
EI	MACRO	PEEK	ST
EJECT	MEM	POKE	STOP
ELSE	MOV	POP	SUB
END	MOVT	PUBLIC	SUBC
ENDCASE	NIBBLE	PURGE	SUMMARY
ENDIF	NIBBLE1	PUSH	TAG
ENDIFC	NIBBLE2	PUT	TITLE
ENDIFNC	NIBBLE2V	REPT	XOR
ENDM	NIBBLE3	RET	ZZZERROR
ENDOP	NIBBLE3V	RETI	ZZZMCHK
ENDP	NIBBLE4	RETSK	ZZZMSG
ENDR	NIBBLE4V	RORC	ZZZOPT
EOF	NIBBLE5	SBMAC	
EXIT	NIBBLE5V	SET	
EXITR	NIBBLE6	SET1	

3.6.2 Registers and flags

(1) AS17120 and AS17121

AR	IXH	POCBIO3	TMCK1
AR0	IXL	POD0	TMEN
AR1	IXM	POD1	TMM
AR2	MPE	POD2	TMOSEL
AR3	MPH	POD3	TMRES
AR_EPA0	MPL	PODBIO0	WR
AR_EPA1	OPEN	PODBIO1	Z
BANK	POA0	PODBIO2	ZZZ0
BCD	POA1	PODBIO3	ZZZ1
CMP	POA2	POE0	ZZZ2
CY	POA3	POE1	ZZZ3
DBF	POABIO0	POEBIO0	ZZZ4
DBF0	POABIO1	POEBIO1	ZZZ5
DBF1	POABIO2	PDRESEN	ZZZ6
DBF2	POABIO3	PSW	ZZZ7
DBF3	POB0	PULLUP	ZZZ8
IEGMD0	POB1	RPH	ZZZ9
IEGMD1	POB2	RPL	ZZZALBMAC
INT	POB3	SIOCK0	ZZZALMAC
IP	POBGIO	SIOCK1	ZZZARGC
IPSIO	POC0	SIOEN	ZZZDEVID
IPTM	POC1	SIOHIZ	ZZZEPA
IRQ	POC2	SIOSFR	ZZZLINE
IRQSIO	POC3	SIOTS	ZZZLSARG
IRQTM	POCBIO0	SP	ZZZPRINT
IX	POCBIO1	TMC	ZZZSKIP
IXE	POCBIO2	TMCK0	ZZZSYDOC

(2) AS17132 and AS17133

AR	IRQ	POC2IDI	SP
AR0	IRQSIO	POC3	TMC
AR1	IRQTM	POC3IDI	TMCK0
AR2	IX	POCBIO0	TMCK1
AR3	IXE	POCBIO1	TMEN
AR_EPA0	IXH	POCBIO2	TMM
AR_EPA1	IXL	POCBIO3	TMOSEL
BANK	IXM	POD0	TMRES
BCD	MPE	POD1	WR
CMP	MPH	POD2	Z
CMPCH0	MPL	POD3	ZZZ0
CMPCH1	OPEN	PODBIO0	ZZZ1
CMPRSLT	POA0	PODBIO1	ZZZ2
CMPSTRT	POA1	PODBIO2	ZZZ3
CMPVREF0	POA2	PODBIO3	ZZZ4
CMPVREF1	POA3	POE0	ZZZ5
CMPVREF2	POABIO0	POE1	ZZZ6
CMPVREF3	POABIO1	POEBIO0	ZZZ7
CY	POABIO2	POEBIO1	ZZZ8
DBF	POABIO3	PDRESEN	ZZZ9
DBF0	POB0	PSW	ZZZALBMAC
DBF1	POB1	PULLUP	ZZZALMAC
DBF2	POB2	RPH	ZZZARGC
DBF3	POB3	RPL	ZZZDEVID
IEGMD0	POBGIO	SIOCK0	ZZZEPA
IEGMD1	POC0	SIOCK1	ZZZLINE
INT	POC0IDI	SIOEN	ZZZLSARG
IP	POC1	SIOHIZ	ZZZPRINT
IPSIO	POC1IDI	SIOSFR	ZZZSKIP
IPTM	POC2	SIOTS	ZZZSYDOC

CHAPTER 4 MASK OPTION DEFINITION PSEUDO INSTRUCTIONS

To create programs for the products, uPD17120, uPD17121, uPD17132, and uPD17133, it is necessary to specify mask options in source programs to be assembled using mask option definition pseudo instructions.

The uSxxxxAS17120 contains the device file (D171xx.DEV) and option file (D171xx.OPT), which correspond to the product, uPD17120, uPD17121, uPD17132, or uPD17133. If these device and option files are registered in the same current directory, the device and option files are automatically loaded, and assembling is performed by specifying a device file name in a sequential file in assembling or specifying a device file name when starting the assembling.

To specify mask options, register the device file (D171xx.DEV) and option file (D171xx.OPT) in the same current directory before assembly. For the uPD17120, for instance, register the D17120.DEV and D17120.OPT files in the same current directory.

Specify mask options for the following pins:

- $\overline{\text{RESET}}$ pin
- Port OD (POD_3, POD_2, POD_1, POD_0)
- Port OE (POE_1, POE_0)

4.1 OPTION and ENDOP Pseudo Instructions

The block from the OPTION pseudo instruction to the ENDOP pseudo instruction is defined as the mask option definition block.

The format for the mask option definition block is shown below. Only the three pseudo instructions listed in Table 4-1 can be described in this block.

[Format]

Symbol	Mnemonic	Operand	Comment
[label:]	OPTION		[;comment]
	:		
	:		
	:		
	ENDOP		

4.2 Mask Option Definition Pseudo Instructions

Table 4-1 lists the pseudo instructions which define the mask options for each pin.

Table 4-1 Mask Option Definition Pseudo Instructions

Pin	Mask option definition pseudo instruction	Number of operands	Parameter name
$\overline{\text{RESET}}$	OPTRES	1	OPEN (without pull-up resistor) PULLUP (with pull-up resistor)
POD_3-POD_0	OPTPOD	4	OPEN (without pull-up resistor) PULLUP (with pull-up resistor)
POE_1 and POE_0	OPTPOE	2	OPEN (without pull-up resistor) PULLUP (with pull-up resistor)

The OPTRES format is shown below. Specify the $\overline{\text{RESET}}$ mask option in the operand field.

Symbol	Mnemonic	Operand	Comment
[label:]	OPTRES	($\overline{\text{RESET}}$)	[;comment]

The OPTPOD format is shown below. Specify mask options for all pins of port OD in the order of POD_3, POD_2, POD_1, and POD_0 starting at the first operand in the operand field.

Symbol	Mnemonic	Operand	Comment
[label:]	OPTPOD	(POD_3), (POD_2), (POD_1), (POD_0)	[;comment]

The OPTPOE format is shown below. Specify mask options for all pins of port OE in the order of POE_1 and POE_0 starting at the first operand in the operand field.

Symbol	Mnemonic	Operand	Comment
[label:]	OPTPOE	(POE_1), (POE_0)	[;comment]

Example of describing mask options

Specify the following mask options in a source file to be assembled for the uPD17120.

- $\overline{\text{RESET}}$ pin: Pull-up
- POD_3: Open, POD_2: Open, POD_1: Pull-up, POD_0: Pull-up
- POE_1: Pull-up, POE_0: Open

```
;uPD17120
Setting mask options:   OPTION
            ;
                        OPTRES      PULLUP
                        OPTPOD      OPEN,OPEN,PULLUP,PULLUP
                        OPTPOE      PULLUP, OPEN
            ;
                        ENDOP
```

CHAPTER 5 FORMATS OF LOAD MODULE FILES

Hexadecimal load module files to be output by the AS17K assembler are classified into two output format types: an ICE file and a PRO file.

The ICE and PRO files must be used according to their applications. These files contain their user program areas, assembling environment information areas, and in-circuit emulator operating environment information areas.

(1) Format of a hexadecimal load module file

The assembler outputs data in hexadecimal load module files in the following sample format:

[Example of the format of a hexadecimal load module file]

```
: 10 0002 00 2B41000BFC80F ··· 3A20 EC
|  |   |  |        |              |
①  ②   ③  ④        ⑤              ⑥
:  00 0000 01 FF
|   |   |  |  |
①   ②   ③  ④  ⑥
```

① Record mark

Start of a record

② Number of codes (two digits)

Number of codes (data items in bytes) stored in a record. The number is represented in hexadecimal up to 10H (corresponding to 16 codes). The number is 00H for the last record.

③ Address (four digits)

Start address of codes in a record. 0000H, which is used for the last record, is not related to the address.

④ Record type (two digits)

Record type 00H indicates that the record is a data record. Record type 01H indicates that the record is the last record.

⑤ Code (up to 32 digits (16 bytes))

Data of up to 16 bytes is output to this field byte by byte.

⑥ Checksum (two digits)

The byte data is output to field ⑥ so that the lowest-order byte of the sum of the data items in ②, ③, ④, ⑤, and ⑥ in bytes is 00H (even parity).

(2) ICE file

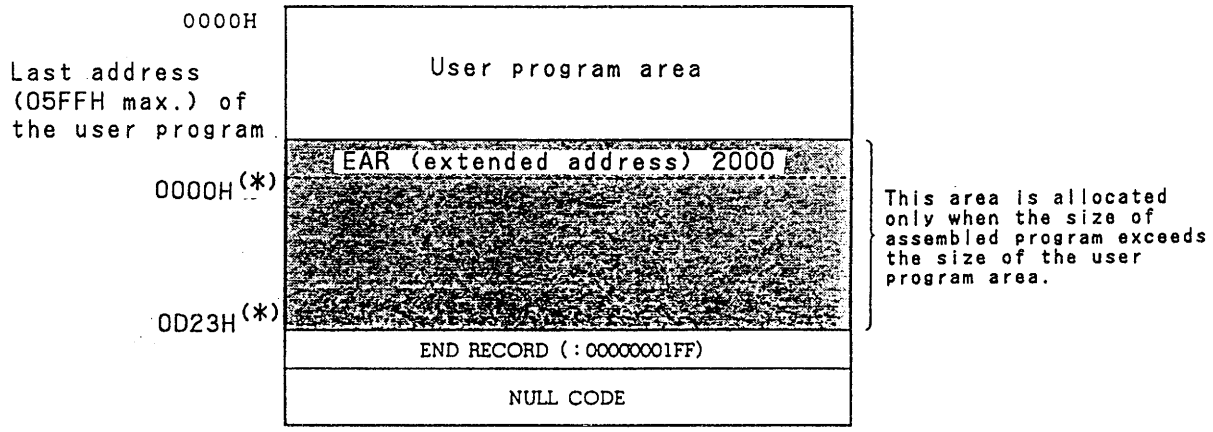
The ICE file contains hexadecimal data to be output by the AS17K assembler. The data is used for an in-circuit emulator (IE-17K or IE-17K-ET) only. Figure 5-1 shows the output format of the file data assembled using the uSxxxxAS17120.

This file consists of two subfiles. The first subfile contains a program area which consists of a user program area and patch area. The patch area is allocated only when patching is performed in the in-circuit emulator. The second subfile contains an in-circuit emulator operating environment information area, an assembling environment information area, and an SE board environment information area. Various data items specifying the operation of the in-circuit emulator are contained in these areas.

Fig. 5-1 Format of the ICE File

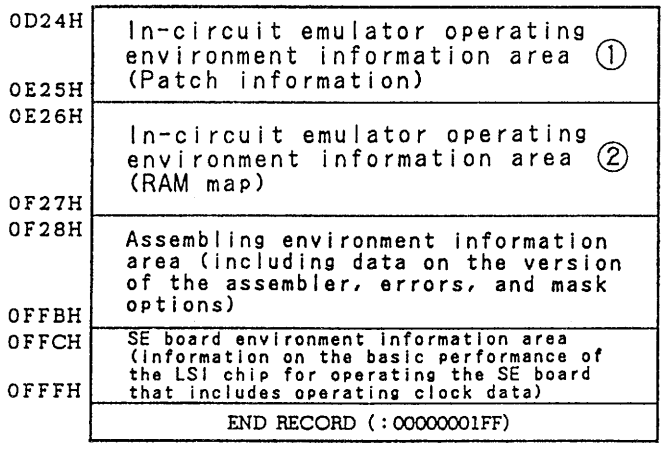
(a) AS17120 and AS17121

First subfile: Program area



* 8000H to 8D23H for the in-circuit emulator

Second subfile: In-circuit emulator operating environment information area and assembling environment information area

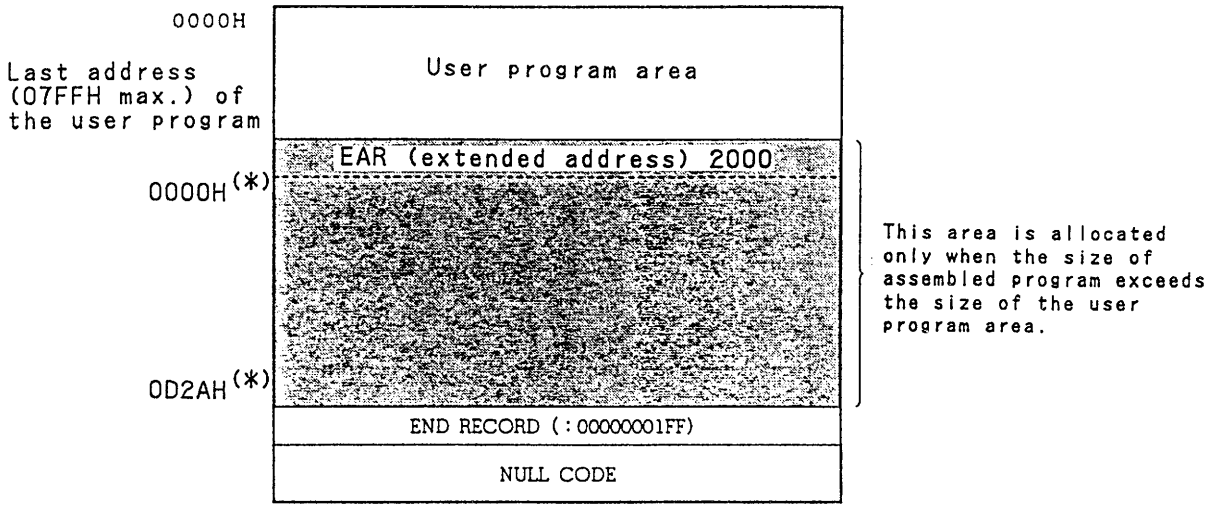


(to be continued)

Fig. 5-1 Format of the ICE File (Cont'd)

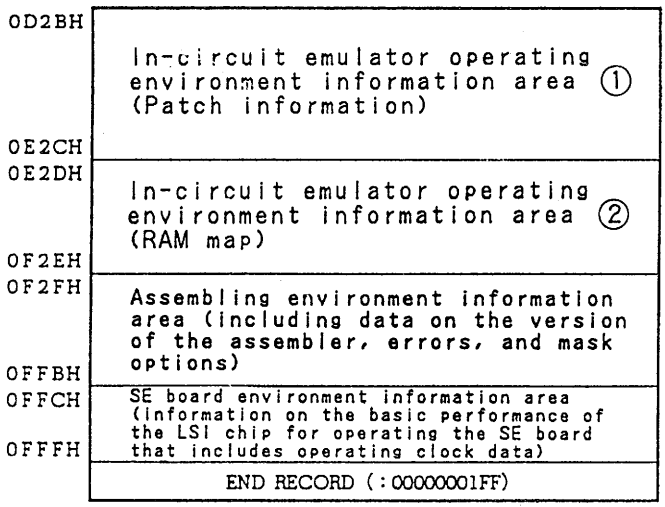
(b) AS17132 and AS17133

First subfile: Program area



* 8000H to 8D2AH for the in-circuit emulator

Second subfile: In-circuit emulator operating environment information area and assembling environment information area



(3) PRO file

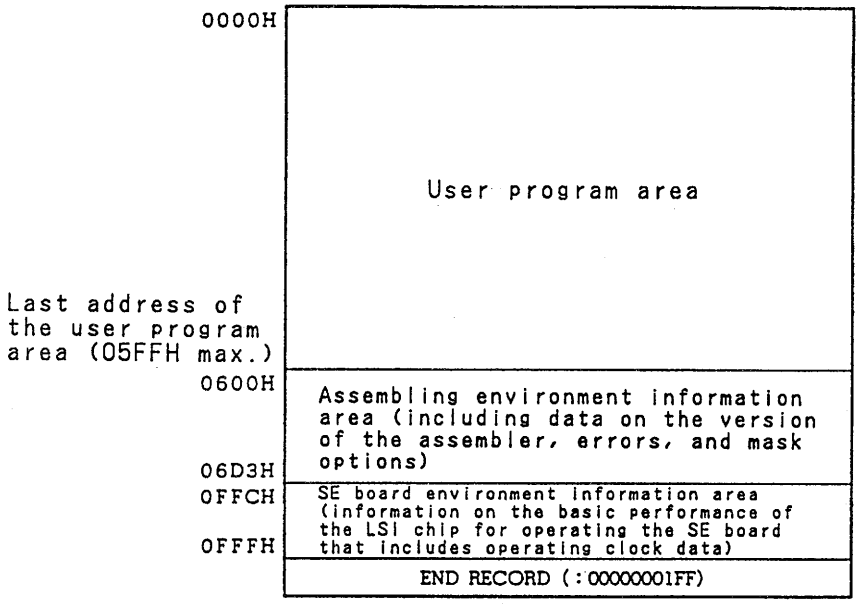
The PRO file contains hexadecimal data to be output by the AS17K assembler. The data is used only for mask products to be ordered, PROM products to be evaluated with a single SE board, and one-time PROM products (uPD17P132 and uPD17P133). To output the PRO file data, /PRO must be specified with an assemble option during assembly.

Figure 5-2 shows the output format of the file data assembled using the uSxxxxAS17120.

The PRO file contains only one file. It consists of a user program area, an assembling environment information area, and an SE board environment information area.

Fig. 5-2 Format of the PRO File

(a) AS17120 and AS17121



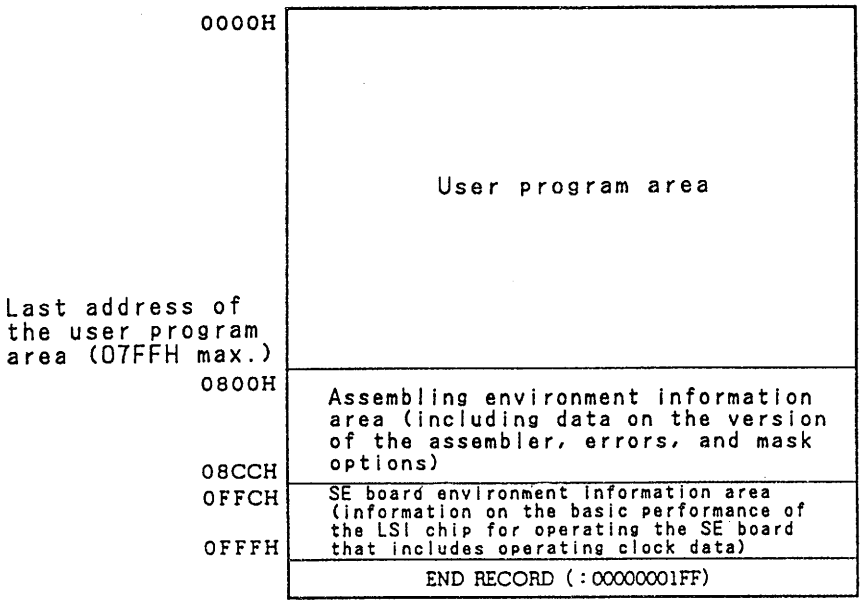
Remarks 1. The assembling environment information area also contains mask option information for LSI chip masking. Documents on mask options are unnecessary when mask products are ordered.

2. Addresses 06D4H to 0FFBH are not assigned to the PRO file.

(to be continued)

Fig. 5-2 Format of the PRO File (Cont'd)

(b) AS17132 and AS17133



Remarks 1. The assembling environment information area also contains mask option information for LSI chip masking. Documents on mask options are unnecessary when mask products are ordered.

2. Addresses 08CDH to 0FFBH are not assigned to the PRO file.

(4) Comparison between load module files: ICE and PRO files

Even when no source files are changed, the ICE and PRO files may differ in the assembler output results; namely, the contents of the assembling environment information areas in these files may be different. This is because these areas contain information on the dates when their source files were created.

Table 5-1 Data Items Which May Differ between the ICE and PRO Files Even If the Source File Does Not Change

(a) AS17120 and AS17121

Data item	Address	
	ICE file	PRO file
Program name (character string of up to 64K bytes which is specified with an assemble option (/'PROG='))	0F28H - 0F67H	0600H - 063FH
Data on mask options	0F68H - 0F6AH	0640H - 0642H
Information on SIMPLEHOST™	0FADH	0685H
Information of whether an error has occurred or whether a warning has been issued	0FB0H	0688H
Time, day, month, and year source file was created(*)	0FBEH - 0FC7H	0696H - 069FH
Version of a device file	0FDDH	06B5H
Version of the assembler	0FE1H	06B9H

* If a source file is divided into multiple modules and the modules are updated, time, day, month, and year are also updated.

Caution: Do not change a load module directly. To change a load module, change and reassemble the source file. If a load module is directly changed, the history of the load module file does not agree with those of other files, causing a software bug.

(to be contidued)

Table 5-1 Data Items Which May Differ between the ICE and PRO Files Even If the Source File Does Not Change (Cont'd)

(b) AS17132 and AS17133

Data item	Address	
	ICE file	PRO file
Program name (character string of up to 64K bytes which is specified with an assemble option (/'PROG='))	0F2FH - 0F6EH	0800H - 083FH
Data on mask options	0F6FH - 0F71H	0840H - 0842H
Information on SIMPLEHOST™	0FADH	087EH
Information of whether an error has occurred or whether a warning has been issued	0FB0H	0881H
Time, day, month, and year source file was created(*)	0FBEH - 0FC7H	088FH - 0898H
Version of a device file	0FDDH	08AEH
Version of the assembler	0FE1H	08B2H

* If a source file is divided into multiple modules and the modules are updated, time, day, month, and year are also updated.

Caution: Do not change a load module directly. To change a load module, change and reassemble the source file. If a load module is directly changed, the history of the load module file does not agree with those of other files, causing a software bug.