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

μSxxxxAS17215

DEVICE FILE

VERSION V2.0

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

AS17215 (V2)
AS17216 (V2)
AS17217 (V1)
AS17218 (V1)

Document No. EEU-1461A
(O. D. No. EEU-925A)
Date Published June 1994 P
Printed in Japan

SIMPLEHOST is a trademark NEC Corporation.
MS-DOS is a trademark of Microsoft Corporation.
PC DOS and PC/AT are trademarks of IBM Corporation.

Major Revisions in This Version

Section	Description
Whole manual	Amalgamated with AS1726 User's Manual (EEU-918) AS17215, 17216 version changed: V1 → V2 AS17217, 17218 added
–	**PREFACE** text changed
	2.3 INSTRUCTION TABLE Operations of following instructions changed:
Page 4	MOVT DBF, @AR
Page 5	BR addr
Page 5	CALL addr
Page 5	CALL @AR
Page 6	**2.4 ASSEMBLER (AS17K) BUILT-IN MACRO INSTRUCTIONS** BANKn added
Page 11	**3.6 OTHERS** added
Page 11	**3.7.1 INSTRUCTIONS AND PSEUDO-INSTRUCTIONS** BANK1 added
Page 12	**3.7.3 MASK OPTIONS** added
Pages 13 to 14	**CHAPTER 4 MASK OPTION DEFINITION PSEUDO-INSTRUCTIONS** in previous edition changed to **CHAPTER 4 MASK OPTIONS**
Pages 16 to 19	**Figure 5-1 ICE File Format** Two files integrated into one
Pages 24 to 27	**Table 5-1 Assembly Environment Information Area Items with Different Possibilities** Program name addresses changed Linker (LK17K) version deleted Compiler (emIC17K) version deleted

The mark ★ shows major revised points.

★ PREFACE

A device file is a file that holds information specific to a 17K series device (device information), and is required when using the following 17K series software development support tools:

- AS17K assembler
- SIMPLEHOST™

μSxxxxAS17215 includes the following device files:

- μPD17215 device file (AS17215)
- μPD17216 device file (AS17216)
- μPD17217 device file (AS17217)
- μPD17218 device file (AS17218)

Each device file consists of files with .DEV and .OPT file extensions, and when used, both these files must be held in the same directory.

- File with .DEV extension: Device file (main file)
- File with .OPT extension: Option file

[List of Files included in μSxxxxAS17215]

Device File	Component File Names
AS17215	D17215.DEV, D17215.OPT
AS17216	D17216.DEV, D17216.OPT
AS17217	D17217.DEV, D17217.OPT
AS17218	D17218.DEV, D17218.OPT

Please refer to the **AS17K Assembler User's Manual (EEU-603)** for the method of using the AS17K assembler and μPD17215 subseries product device files.

CONTENTS

CONTENTS OF FIGURES

CONTENTS OF TABLES

CHAPTER 1 DEVICE INFORMATION

The AS17215, 17216, 17217 and 17218 device files supply the following information related to each μPD17215 subseries product at assembly:

(1) Program memory (ROM) capacity ★

μPD17215 : 2048 × 16 bits (0000H to 07FFH)
μPD17216 : 4096 × 16 bits (0000H to 0FFFH)
μPD17217 : 6144 × 16 bits (0000H to 17FFH)
μPD17218 : 8192 × 16 bits (0000H to 1FFFH)

(2) Data memory (RAM) capacity ★

μPD17215, 17216 : 111 × 4 bits (BANK0)
μPD17217, 17218 : 223 × 4 bits (BANK0, BANK1)

(3) Usable instructions

See **CHAPTER 2 μPD17215 SUBSERIES INSTRUCTION SET.**

(4) Register files, port registers, and peripheral registers read and write information

See **CHAPTER 3 RESERVED SYMBOLS.**

(5) Reserved symbols

See **CHAPTER 3 RESERVED SYMBOLS.**

(6) Mask option information ★

See **CHAPTER 4 MASK OPTIONS.**

(7) Device file and device number/SE board number

In the device file, the device number peculiar to each device and the SE board number which identifies the most suitable SE board for development of each product are registered. These are also included in the ICE file and the PRO flie which are output by the assembler (AS17K). They are used when checking the development environment with an in-circuit emulator and checking when a mask is ordered.

Table 1-1 Device File and Device Number/SE Board Number Correspondence Table ★

Device File (Version)	Device Name	Device Number	SE Board Number	SE Board
AS17215 (V2)	μPD17215	49	34	SE-17215
AS17216 (V2)	μPD17216	4A		
AS17217 (V1)	μPD17217	50		
AS17218 (V1)	μPD17218	4B		

CHAPTER 2 μPD17215 SUBSERIES INSTRUCTION SET

2.1 INSTRUCTION SET OUTLINE

b_{14} to b_{11} (BIN)	HEX	b_{15} = 0		b_{15} = 1	
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	AR		
		INC	IX		
		MOVT	DBF, @AR		
		BR	@AR		
		CALL	@AR		
		RET			
		RETSK			
		EI			
		DI			
		RETI			
		PUSH	AR		
		POP	AR		
		GET	DBF, p		
		PUT	p, DBF		
		PEEK	WR, rf		
		POKE	rf, WR		
		RORC	r		
		STOP	s		
		HALT	h		
		NOP			
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 (page 0)	CALL	addr
★ 1 1 0 1	D	BR	addr (page 1)	MOV	m, #4
★ 1 1 1 0	E	BR	addr (page 2)	SKT	m, #n
★ 1 1 1 1	F	BR	addr (page 3)	SKF	m, #n

2.2 LEGEND

AR : Address register
ASR : Address stack register indicated by stack pointer
addr : Program memory address (lower 11 bits)
BANK : Bank register
CMP : Compare flag
CY : Carry flag
DBF : Data buffer
h : Halt release condition
INTEF : Interrupt enable flag
INTR : Automatically saved register during interrupt
INTSK : Interrupt stack register
IX : Index register
MP : Data memory row address pointer
  MPE : Memory pointer enable flag
m : Data memory address represented by m_R and mc
  m_R : Data memory row address (higher)
  mc : Data memory column address (lower)
n : Bit position (4 bits)
n4 : Immediate data (4 bits)
PAGE : Page (program counter bits 11 and 12)
PC : Program counter
p : Peripheral address
  p_H : Peripheral address (upper 3 bits)
  p_L : Peripheral address (lower 4 bits)
r : General register column address
rf : Register file address
  rf_R : Register file row address (higher 3 bits)
  rfc : Register file column address (lower 4 bits)
SP : Stack pointer
s : Stop release condition
WR : Window register
(x) : Content addressed by x

2.3 INSTRUCTION TABLE

Instruction Group	Mnemonic	Operands	Operation	Operation Code			
				Op. Code	Operand		
Addition	ADD	r, m	(r) ← (r) + (m)	00000	m_R	mc	r
		m, #n4	(m) ← (m) + n4	10000	m_R	mc	n4
	ADDC	r, m	(r) ← (r) + (m) + CY	00010	m_R	mc	r
		m, #n4	(m) ← (m) + n4 + CY	10010	m_R	mc	n4
	INC	AR	AR ← AR + 1	00111	000	1001	0000
		IX	IX ← IX + 1	00111	000	1000	0000
Subtraction	SUB	r, m	(r) ← (r) − (m)	00001	m_R	mc	r
		m, #n4	(m) ← (m) − n4	10001	m_R	mc	n4
	SUBC	r, m	(r) ← (r) − (m) − CY	00011	m_R	mc	r
		m, #n4	(m) ← (m) − n4 − CY	10011	m_R	mc	n4
Logical operation	OR	r, m	(r) ← (r) ∨ (m)	00110	m_R	mc	r
		m, #n4	(m) ← (m) ∨ n4	10110	m_R	mc	n4
	AND	r, m	(r) ← (r) ∧ (m)	00100	m_R	mc	r
		m, #n4	(m) ← (m) ∧ n4	10100	m_R	mc	n4
	XOR	r, m	(r) ← (r) $\overline{\vee}$ (m)	00101	m_R	mc	r
		m, #n4	(m) ← (m) $\overline{\vee}$ n4	10101	m_R	mc	n4
Decision	SKT	m, #n	CMP ← 0, if (m) ∧ n=n, then skip	11110	m_R	mc	n
	SKF	m, #n	CMP ← 0, if (m) ∧ n=0, then skip	11111	m_R	mc	n
Comparison	SKE	m, #n4	(m) − n4, skip if zero	01001	m_R	mc	n4
	SKNE	m, #n4	(m) − n4, skip if not zero	01011	m_R	mc	n4
	SKGE	m, #n4	(m) − n4, skip if not borrow	11001	m_R	mc	n4
	SKLT	m, #n4	(m) − n4, skip if borrow	11011	m_R	mc	n4
Rotation	RORC	r	┌→ CY → $(r)_{b3}$ → $(r)_{b2}$ → $(r)_{b1}$ → $(r)_{b0}$ ─┐	00111	000	0111	r
Transfer	LD	r, m	(r) ← (m)	01000	m_R	mc	r
	ST	m, r	(m) ← (r)	11000	m_R	mc	r
	MOV	@r, m	if MPE = 1: (MP, (r)) ← (m) if MPE = 0: (BANK, m_R, (r)) ← (m)	01010	m_R	mc	r
		m, @r	if MPE = 1: (m) ← (MP, (r)) if MPE = 0: (m) ← (BANK, m_R, (r))	11010	m_R	mc	r
		m, #n4	(m) ← n4	11101	m_R	mc	n4
	MOVT	DBF, @AR	SP ← SP − 1, ASR ← PC, PC ← AR, DBF ← (PC), 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	rfc
	POKE	rf, WR	(rf) ← WR	00111	rf_R	0010	rfc
	GET	DBF, p	DBF ← (p)	00111	P_H	1011	P_L
	PUT	p, DBF	(p) ← DBF	00111	P_H	1010	P_L

Instruction Group	Mnemonic	Operands	Operation	Operation Code			
				Op. Code	Operand		
Branch	BR	addr	*	*	addr		
		@AR	PC ← AR	00111	000	0100	0000
Sub-routine	CALL	addr	SP ← SP – 1, ASR ← PC, PC_{10-0} ← addr, PAGE ← 0	11100	addr		
		@AR	SP ← SP – 1, ASR ← PC, PC ← AR	00111	000	1110	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
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

★

★

* The μPD17215, 17216, 17217, and 17218 "BR addr" operations and op codes are as shown below. ★

(a) μPD17215

Operand	Operation	Op. Code
addr	PC_{10-0} ← addr	01100

(b) μPD17216

Operand	Operation	Op. Code
addr	PC_{10-0} ← addr, PAGE ← 0	01100
	PC_{10-0} ← addr, PAGE ← 1	01101

(c) μPD17217

Operand	Operation	Op. Code
addr	PC_{10-0} ← addr, PAGE ← 0	01100
	PC_{10-0} ← addr, PAGE ← 1	01101
	PC_{10-0} ← addr, PAGE ← 2	01110

(d) μPD17218

Operand	Operation	Op. Code
addr	PC_{10-0} ← addr, PAGE ← 0	01100
	PC_{10-0} ← addr, PAGE ← 1	01101
	PC_{10-0} ← addr, PAGE ← 2	01110
	PC_{10-0} ← addr, PAGE ← 3	01111

2.4 ASSEMBLER (AS17K) BUILT-IN MACRO INSTRUCTIONS

Legend
flag n : FLG type symbols
n : Bit number
< > : < > Omission possible

	Mnemonic	Operands	Operation	n
Built-in macro instruction	SKTn	flag 1, ... flag n	if (flag 1) to (flag n) = all "1", then skip	$1 \leq n \leq 4$
	SKFn	flag 1, ... flag n	if (flag 1) to (flag n) = all "0", then skip	$1 \leq n \leq 4$
	SETn	flag 1, ... flag n	(flag 1) to (flag n) ← 1	$1 \leq n \leq 4$
	CLRn	flag 1, ... flag n	(flag 1) to (flag n) ← 0	$1 \leq n \leq 4$
	NOTn	flag 1, ... flag n	if (flag n) = "0", then (flag n) ← 1 if (flag n) = "1", then (flag n) ← 0	$1 \leq n \leq 4$
	INITFLG	<NOT> flag 1, ...<<NOT> flag n>	if description = NOT flag n, then (flag n) ← 0 if description = flag n, then (flag n) ← 1	$1 \leq n \leq 4$
★	BANKn		(BANK) ← n	n = 0*

* n = 0, 1 in case of μPD17217 and μPD17218

CHAPTER 3 RESERVED SYMBOLS

The symbols which are defined by the AS17215, 17216, 17217 and 17218 device files are shown on the following pages.

The symbols which are defined are:

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

3.1 DATA BUFFERS (DBF)

Symbol	Attribute	Value	R/W	Description
DBF3	MEM	0.0CH	R/W	Data buffer bit 15 to bit 12
DBF2	MEM	0.0DH	R/W	Data buffer bit 11 to bit 8
DBF1	MEM	0.0EH	R/W	Data buffer bit 7 to bit 4
DBF0	MEM	0.0FH	R/W	Data buffer bit 3 to bit 0

3.2 SYSTEM REGISTERS (SYSREG)

Symbol	Attribute	Value	R/W	Description
AR3	MEM	0.74H	R	Address register bit 15 to bit 12
AR2	MEM	0.75H	R/W	Address register bit 11 to bit 8
AR1	MEM	0.76H	R/W	Address register bit 7 to bit 4
AR0	MEM	0.77H	R/W	Address register bit 3 to bit 0
WR	MEM	0.78H	R/W	Window register
BANK	MEM	0.79H	R	Bank register
IXH	MEM	0.7AH	R	Index register high
MPH	MEM	0.7AH	R	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	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

Symbol	Attribute	Value	R/W	Description
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
P0E3	FLG	0.6FH.3	R/W	Port 0E bit 3
P0E2	FLG	0.6FH.2	R/W	Port 0E bit 2
P0E1	FLG	0.6FH.1	R/W	Port 0E bit 1
P0E0	FLG	0.6FH.0	R/W	Port 0E bit 0

3.4 REGISTER FILES

Symbol	Attribute	Value	R/W	Description
SP	MEM	0.81H	R/W	Stack pointer
SYSCK	FLG	0.82H.0	R/W	System clock select flag
WDTRES	FLG	0.83H.3	R/W	Watchdog timer reset flag
BTMCK	FLG	0.83H.2	R/W	Basic interval timer mode selct flag
BTMRES	FLG	0.83H.1	R/W	Basic interval timer reset flag
INT	FLG	0.8FH.0	R	INT pin status flag
NRZBF	FLG	0.91H.0	R/W	NRZ buffer data flag
NRZ	FLG	0.92H.0	R/W	NRZ data flag
P0EBPU3	FLG	0.97H.3	R/W	$P0E_3$ pull-up setting flag
P0EBPU2	FLG	0.97H.2	R/W	$P0E_2$ pull-up setting flag
P0EBPU1	FLG	0.97H.1	R/W	$P0E_1$ pull-up setting flag
P0EBPU0	FLG	0.97H.0	R/W	$P0E_0$ pull-up setting flag
IEG	FLG	0.9FH.0	R/W	INT pin interrupt edge select flag
P0EBIO3	FLG	0.0A7H.3	R/W	$P0E_3$ input/output setting flag
P0EBIO2	FLG	0.0A7H.2	R/W	$P0E_2$ input/output setting flag
P0EBIO1	FLG	0.0A7H.1	R/W	$P0E_1$ input/output setting flag
P0EBIO0	FLG	0.0A7H.0	R/W	$P0E_0$ input/output setting flag
IPBTM	FLG	0.0AFH.2	R/W	Basic interval timer interrupt enable flag
IP	FLG	0.0AFH.1	R/W	INT pin interrupt enable flag
IPTM	FLG	0.0AFH.0	R/W	Timer interrupt enable flag
TMEN	FLG	0.0B3H.3	R/W	Timer enable flag
TMRES	FLG	0.0B3H.2	R/W	Timer reset flag
TMCK1	FLG	0.0B3H.1	R/W	Timer clock select flag
TMCK0	FLG	0.0B3H.0	R/W	Timer clock select flag
IRQBTM	FLG	0.0BDH.0	R/W	Basic interval timer interrupt request flag
IRQ	FLG	0.0BEH.0	R/W	INT pin interrupt request flag
IRQTM	FLG	0.0BFH.0	R/W	Timer interrupt request flag

3.5 PERIPHERAL HARDWEARE REGISTERS

Symbol	Attribute	Value	R/W	Description
NRZLTMM	DAT	03H	R/W	NRZ low level timer modulo register
NRZHTMM	DAT	04H	R/W	NRZ high level timer modulo register
TMC	DAT	05H	R	Timer count register
TMM	DAT	06H	W	Timer modulo register
AR	DAT	40H	R/W	Address register

3.6 OTHERS ★

Symbol	Attribute	Value	Description
DBF	DAT	0FH	PUT instruction, GET instruction, and MOVT instruction fixed operand value
IX	DAT	01H	INC instruction fixed operand value

3.7 RESERVED WORDS TABLE (ALPHABETICAL ORDER)

3.7.1 Instructions and Pseudo-Iinstructions

ADD	EXITR	NIBBLE6	SET1
ADDC	EXTRN	NIBBLE6V	SET2
AND	FLG	NIBBLE7	SET3
BANK0	GET	NIBBLE7V	SET4
BANK1*	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	

* μPD17217, 17218 only

3.7.2 Registers and Flags

AR
AR0
AR1
AR2
AR3
AR_EPA0
AR_EPA1
BANK
BCD
BTMCK
BTMRES
CMP
CY
DBF
DBF0
DBF1
DBF2
DBF3
IEG
INT
IP
IPBTM
IPTM
IRQ
IRQBTM
IRQTM
IX
IXE
IXH
IXL
IXM
MPE
MPH
MPL
NOUSEPOC
NRZ
NRZBF
NRZHTMM
NRZLTMM
OPEN
P0A0
P0A1
P0A2
P0A3
P0B0
P0B1
P0B2
P0B3
P0C0
P0C1
P0C2
P0C3
P0D0
P0D1
P0D2
P0D3
P0E0
P0E1
P0E2
P0E3
P0EBIO0
P0EBIO1
P0EBIO2
P0EBIO3
P0EBPU0
P0EBPU1
P0EBPU2
P0EBPU3
PSW
PULLUP
RPH
RPL
SP
SYSCK
TMC
TMCK0
TMCK1
TMEN
TMM
TMRES
USEPOC
WDTRES
WR
Z
ZZZ0
ZZZ1
ZZZ2
ZZZ3
ZZZ4
ZZZ5
ZZZ6
ZZZ7
ZZZ8
ZZZ9
ZZZALBMAC
ZZZALMAC
ZZZARGC
ZZZDEVID
ZZZEPA
ZZZLINE
ZZZLSARG
ZZZPRINT
ZZZSKIP
ZZZSYDOC

★ 3.7.3 Mask Options

ENDOP
NOUSEPOC
OPEN
OPTION
OPTRES
OPTPOC
PULLUP
USEPOC

CHAPTER 4 MASK OPTIONS ★

μPD17215 subseries products have the following mask options:

- $\overline{\text{RESET}}$ pin internal pull-up resistor
- On-chip POC circuit

When writing a program, all the above mask options must be specified in the source program using a mask option definition pseudo-instruction.
The device file consists of files with .DEV and .OPT file extensions.

- .DEV file: Device file (main file)
- .OPT file: Option file

The option file is required to specify mask options.
Also, when assembly is performed, both these files must be held in the same directory.

Remarks If the device file full path name is specified in a sequential file (.SEQ file), the option file will be searched for in the same directory as the device file.

4.1 MASK OPTION SPECIFICATION METHOD

Mask options are described in the source program by means of the following pseudo-instructions:

- OPTION pseudo-instruction & ENDOP pseudo-instruction
- Mask option definition pseudo-instruction

(1) OPTION pseudo-instruction & ENDOP pseudo-instruction

These pseudo-instructions specify the range in which the mask options are described (mask option definition block).

Mask options are specified by writing mask option definition pseudo-instructions within the area enclosed by the OPTION pseudo-instruction and the ENDOP pseudo-instruction.

Description Format

Symbol Field	Mnemonic Field	Operand Field	Comment Field
[Label:]	OPTION		[; Comment]
	⋮		
	ENDOP		

(2) Mask option definition pseudo-instruction

Table 4-1 Mask Option Definition Pseudo-Instructions

Option	Definition Pseudo-Instruction & Format	Operand	Definition Content
$\overline{\text{RESET}}$ pin internal pull-up resistor	OPTRES <operand>	OPEN	No
		PULLUP	Yes
On-chip POC circuit	OPTPOC <operand>	NOUSEPOC	Not used
		USEPOC	Used

(3) Example of mask option description

;Example of μPD17215 mask option description

MASK_OPTION:

```
OPTION              ;     Start of mask option definition block
OPTRES   PULLUP     ;     RESET pin has internal pull-up resistor
OPTPOC   NOUSEPOC   ;     On-chip POC circuit not used
ENDOP               ;     End of mask option definition block
```

Note
1. **The specified number of parameters must be specified in the operand. An error will be flagged if there are too few or too many. "Operand count error"**
2. **A symbol defined as a reserved word should be written in the operand. An error will be flagged if a symbol with a different type or value from that of the specified symbol is written. "Invalid value for xxxxx terminal"**
3. **The option file (.OPT file) must be located in the same directory as the device file (.DEV file). If there is no option file in the same directory, an error will be flagged and the mask options will not be set.**
   "D17215.OPT:LIB:No such file"
   If this error is flagged, an error will also be flagged when linkage is performed, since there is no mask option specification. "No option specification block"

CHAPTER 5 LOAD MODULE FILE FORMAT

There are two types of output formats of the HEX format load module files output by the assembler (AS17K), ICE file and PRO file.

These two files should be used in accordance with their purposes, respectively. They contain the assembly environment information area, the in-circuit emulator operating environment information area, etc., in addition to the user program area.

(1) HEX format load module file formatting

Each data in the HEX format load module files output by the assembler is output as shown in the format examples below.

[HEX format load module file formatting examples]

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

① Record mark
Indicates the start of a record.

② Number of codes (two digits)
Indicates the number of codes (byte data) contained in the record. The maximum of 10H (for 16 codes) is represented in hexadecimal numbers. For the last record, the value is 00H.

③ Address (4 digits)
Indicates the start address of the code represented by the record. At the final record, it becomes 0000H, which has no relation to the address.

④ Record type (2 digits)
00H indicates that the record is a data record, and 01H means that it is the final record.

⑤ Code (maximum 32 digits (16 bytes))
Up to 16 bytes are output to this field byte-wise.

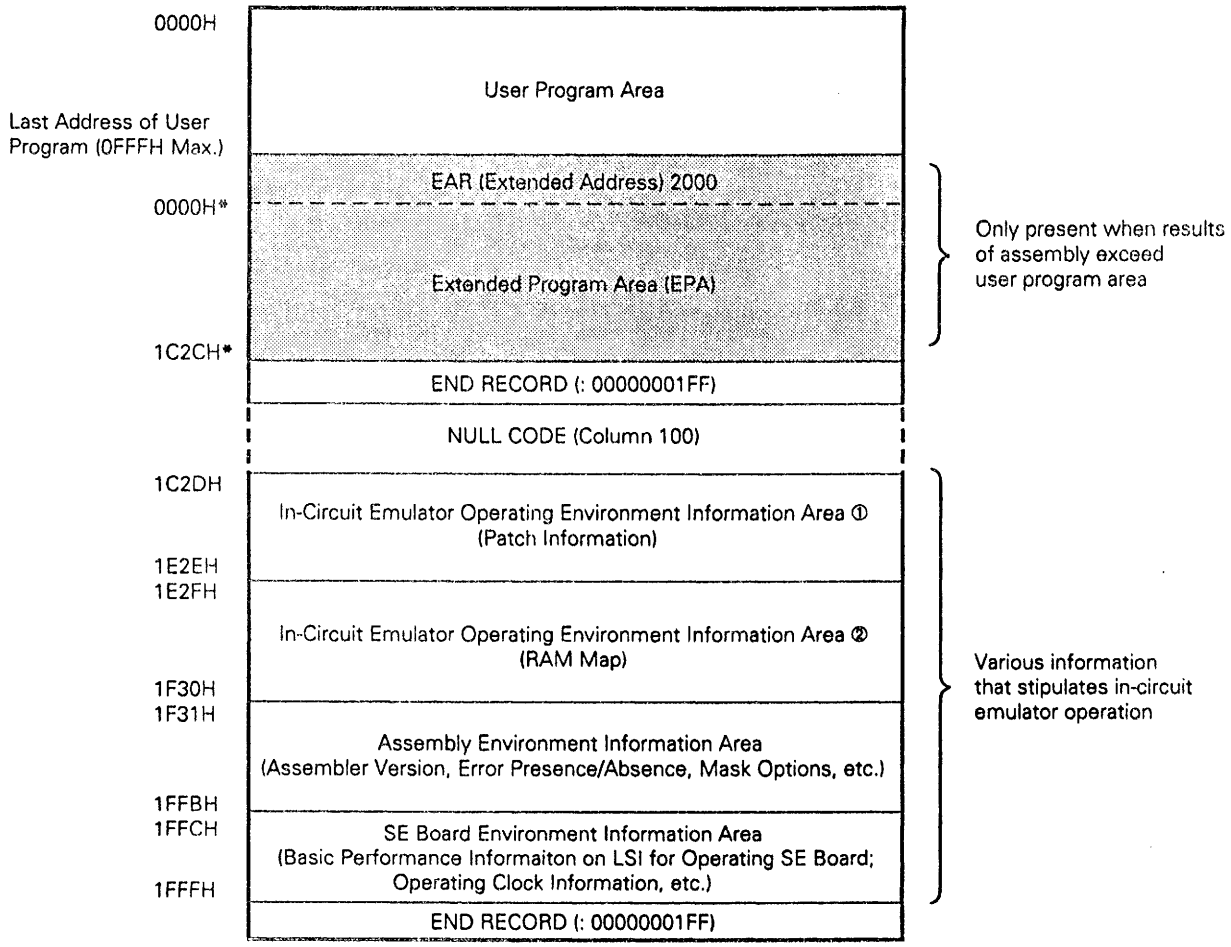
⑥ Check sum (2 digits)
Byte data is output to ⑥ so that the least significant byte of the sum of data ②, ③, ④, ⑤ and ⑥ totaled in 1-byte units is 00H (even parity).

(2) ICE file

This file is output in the HEX format dedicated to the in-circuit emulator (IE-17K or IE-17K-ET) to which the AS17K outputs. The output format in the case of assembly using the μSxxxxAS17215 is shown in Fig. 5-1.

★ **Fig. 5-1 ICE File Format (1/4)**

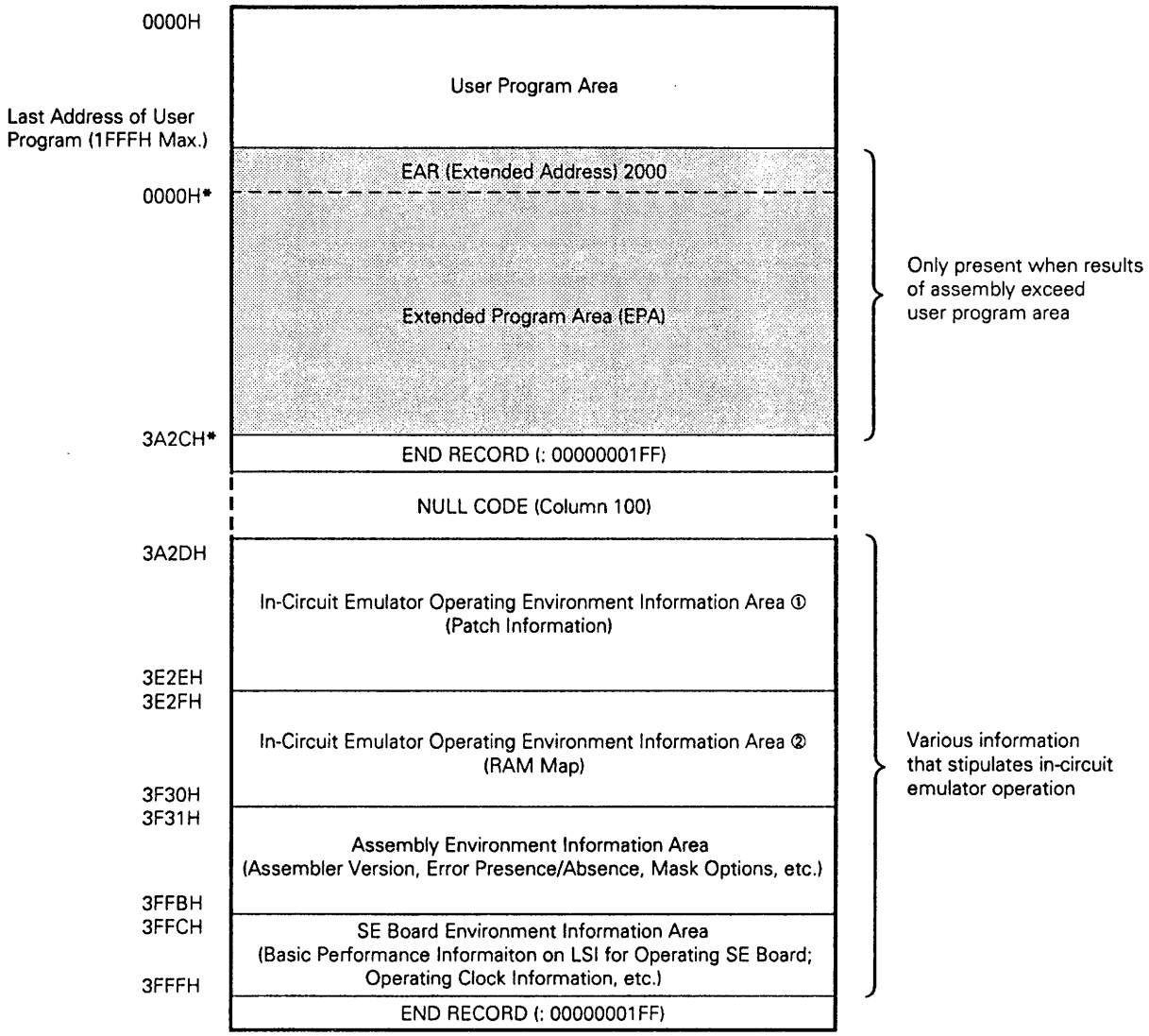
(a) AS17215



* With the in-circuit emulator, 10000H to 11C2CH.

Fig. 5-1 ICE File Format (2/4) ★

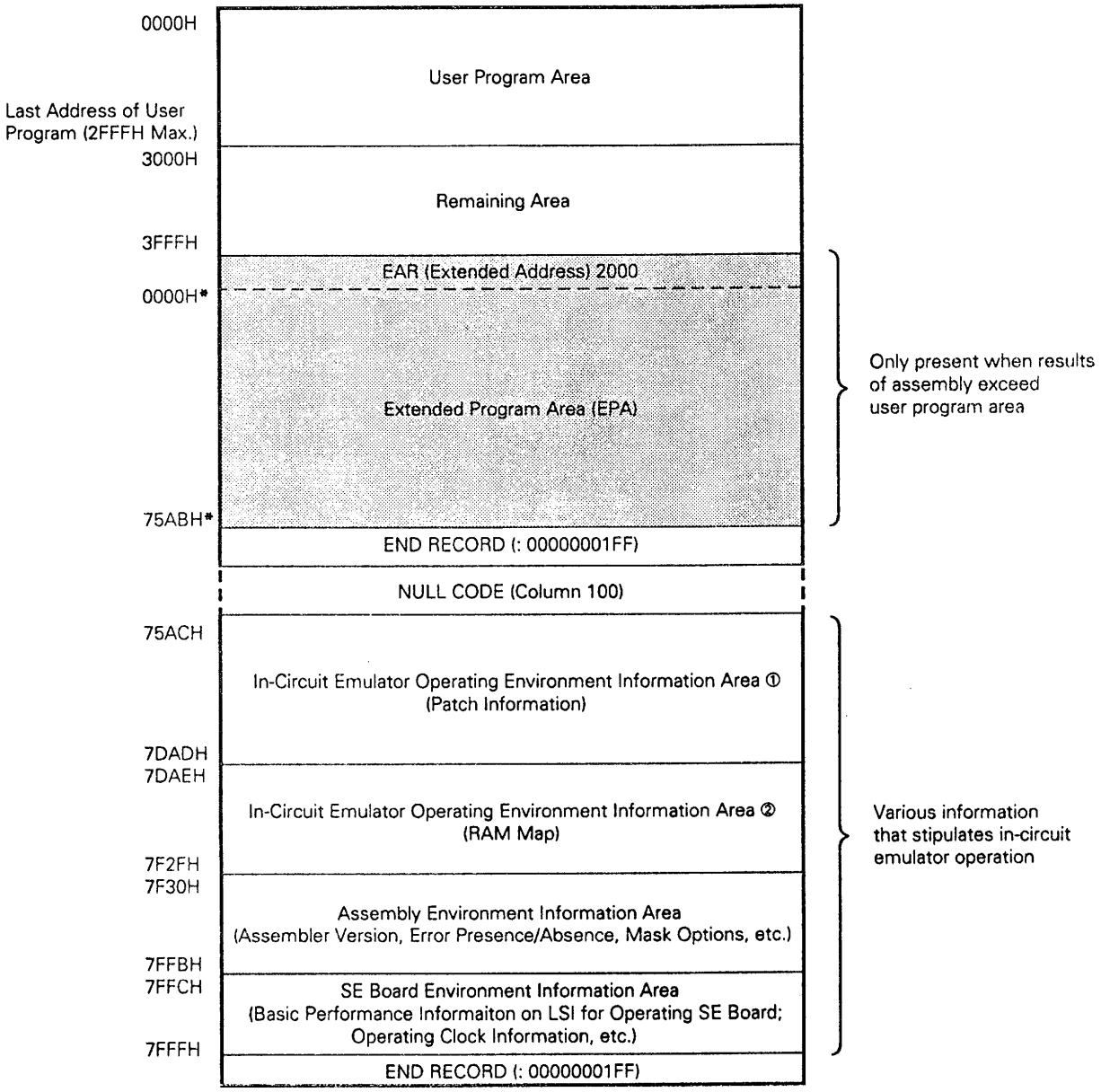
(b) AS17216



* With the in-circuit emulator, 10000H to 13A2CH.

★ **Fig. 5-1 ICE File Format (3/4)**

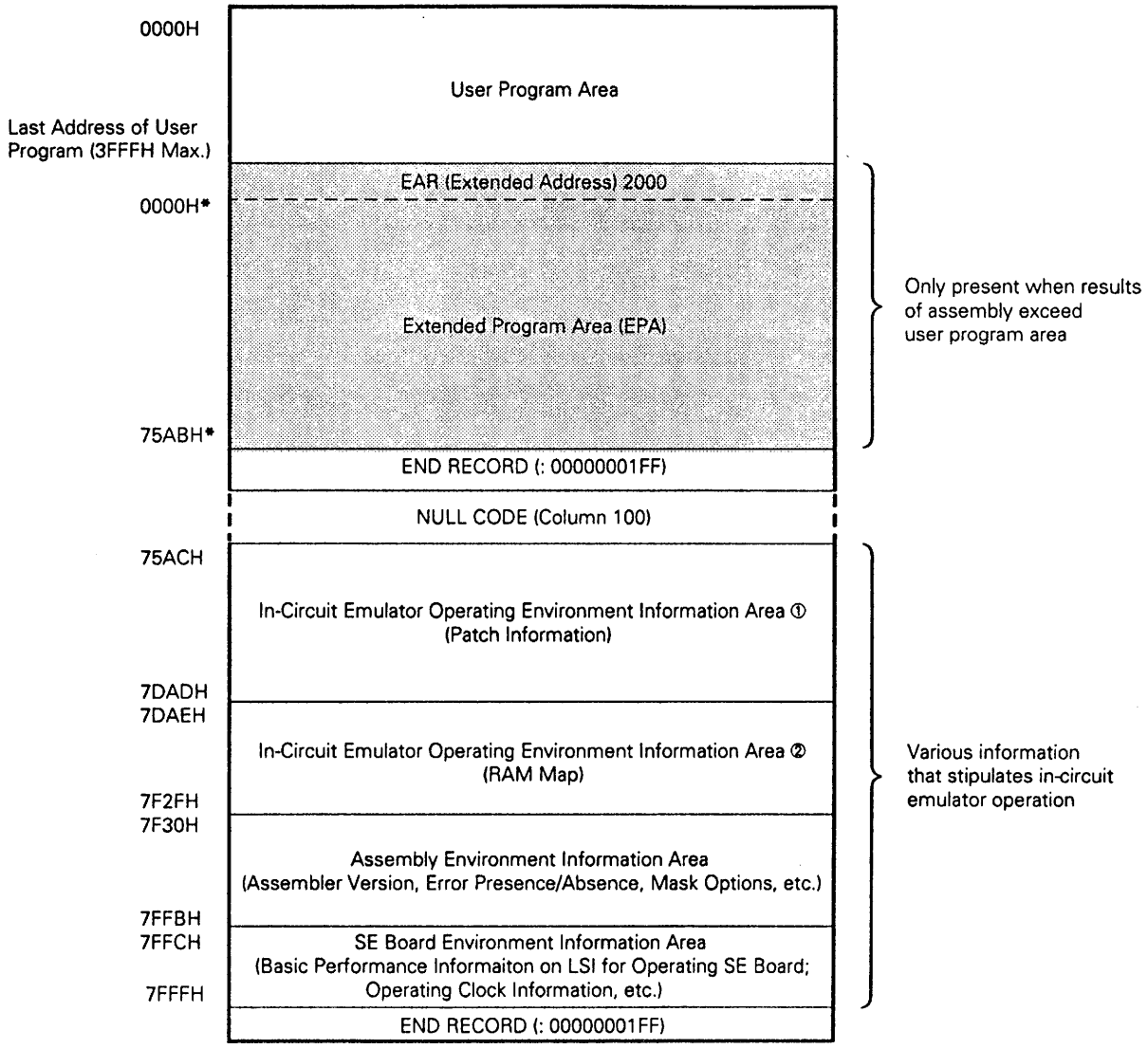
(c) AS17217



* With the in-circuit emulator, 10000H to 175ABH.

Fig. 5-1 ICE File Format (4/4) ★

(d) AS17218



* With the in-circuit emulator, 10000H to 175ABH.

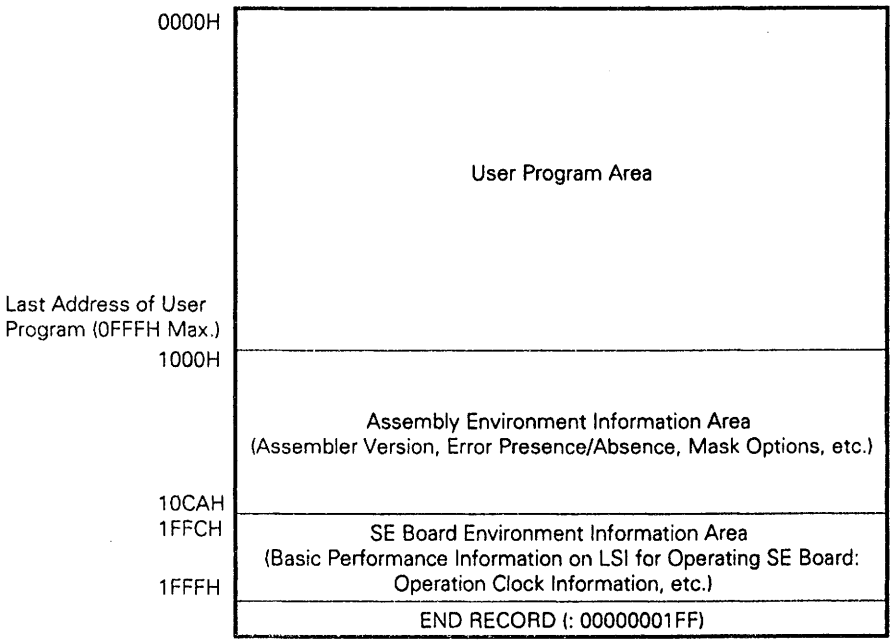
(3) PRO file

This is the HEX data dedicated PROM/one-time PROM products (μPD17P218*) which are used for mask ordering or evaluation in a standalone SE board. During assembly, the PRO file is output by specifying it as /PRO by the assembly option. The output format in the case of assembly using the μSxxxxAS17215 is shown in Fig. 5-2.

* Under development

Fig. 5-2 PRO File Format (1/4)

(a) AS17215

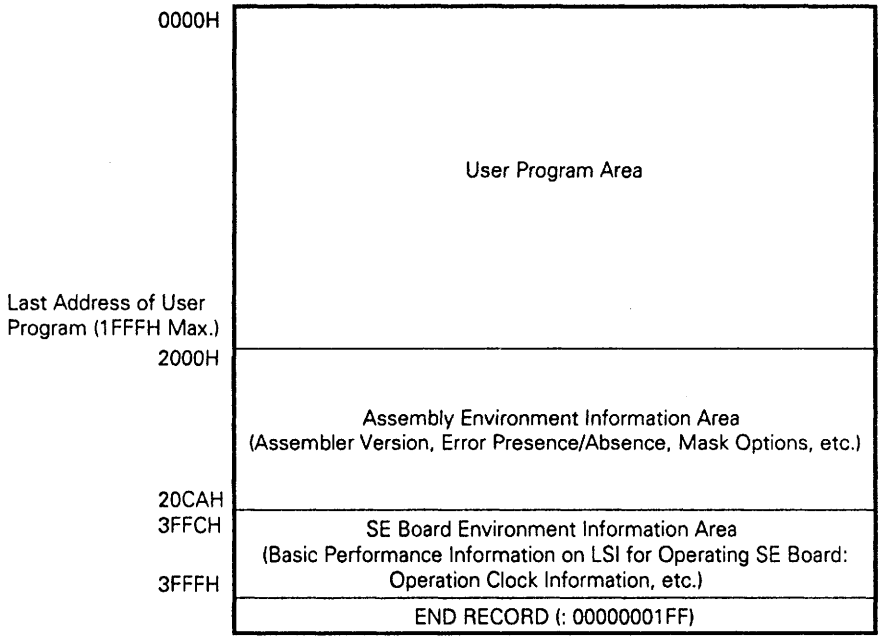


Remarks 1. The assembly environment information area contains mask option information used for LSI mask creation. Therefore, documentation concerning mask options is not required when the mask is ordered.

2. The area 10CBH to 1FFBH is not present in the PRO file.

Fig. 5-2 PRO File Format (2/4) ★

(b) AS17216

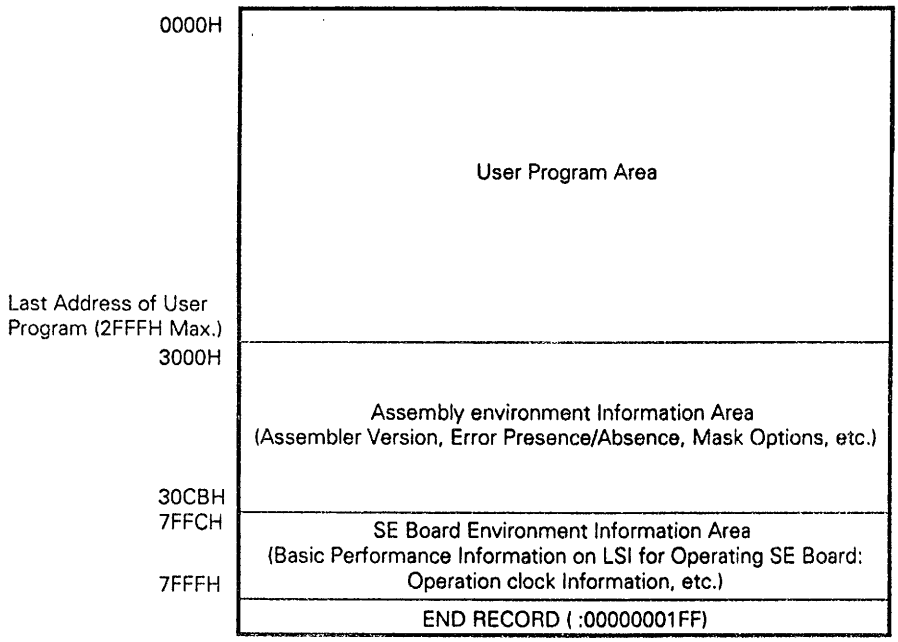


Remarks 1. The assembly environment information area contains mask option information used for LSI mask creation. Therefore, documentation concerning mask options is not required when the mask is ordered.

2. The area 20CBH to 3FFBH is not present in the PRO file.

★ **Fig. 5-2 PRO File Format (3/4)**

(c) AS17217

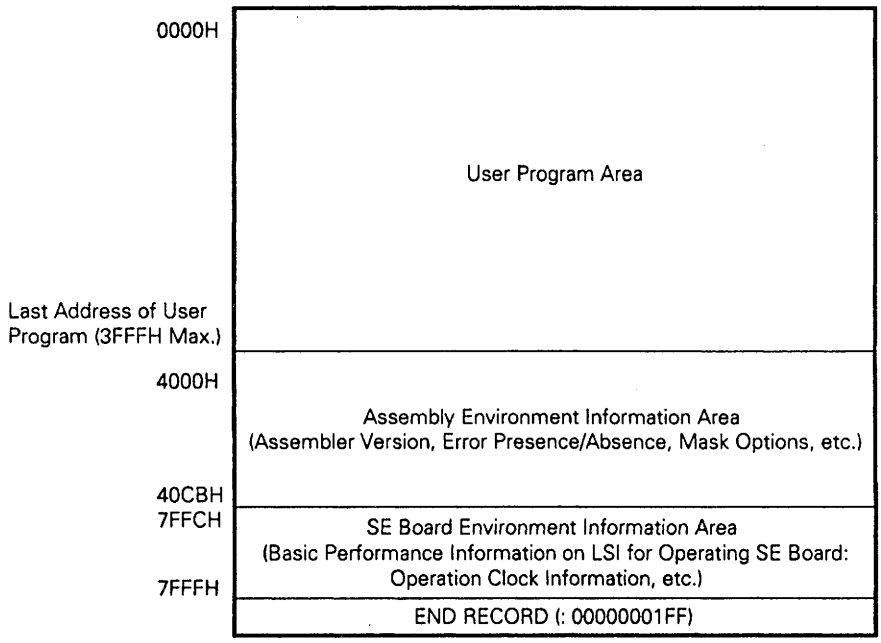


Remarks 1. The assembly environment information area contains mask option information used for LSI mask creation. Therefore, documentation concerning mask options is not required when the mask is ordered.

2. The area 30CCH to 7FFBH is not present in the PRO file.

Fig. 5-2 PRO File Format (4/4) ★

(d) AS17218



Remarks 1. The assembly environment information area contains mask option information used for LSI mask creation. Therefore, documentation concerning mask options is not required when the mask is ordered.

2. The area 40CCH to 7FFBH is not present in the PRO file.

(4) Load module file update contents

Even if the source file has not been changed, if reassembly is performed, the contents of part of the load module file (the assembly environment information area) may be different. This is because the assembly environment information area contains information such as the date and time at which the source file was created.

★ **Table 5-1 Assembly Environment Information Area Items with Different Possibilities (1/4)**

(a) AS17215

Item	Addresses	
	ICE File	PRO File
Program name (Max. 64-byte character string specified by assembly option (/'PROG=')	1F31H to 1F70H	1000H to 103FH
Mask option data	1F71H, 1F72H	1040H, 1041H
Information for SIMPLEHOST	1FADH	107CH
Presence/absence of errors or warnings	1FB0H	107FH
Source file creation year, month, day, hour, minute*	1FBEH to 1FC7H	108DH to 1096H
Device name	1FC8H to 1FD7H	1097H to 10A6H
Device file version	1FDCH, 1FDDH	10ABH, 10ACH
Assembler version	1FDEH to 1FE1H	10ADH to 10B0H

* If the source file is divided into a number of modules, the latest values for the creation year, month, day, hour, minute, are written in those modules.

Note **The load module file should not be changed directly, as this will result in inconsistency between the load module file and other file histories, and will cause a bug. Reassembly must be performed when the load module file is changed.**

Table 5-1 Assembly Environment Information Area Items with Different Possibilities (2/4) ★

(b) AS17216

Item	Addresses	
	ICE File	PRO File
Program name (Max. 64-byte character string specified by assembly option (/'PROG=')	3F31H to 1F70H	2000H to 203FH
Mask option data	3F71H, 3F72H	2040H, 2041H
Information for SIMPLEHOST	3FADH	207CH
Presence/absence of errors or warnings	3FB0H	207FH
Source file creation year, month, day, hour, minute*	3FBEH to 3FC7H	208DH to 2096H
Device name	3FC8H to 3FD7H	2097H to 20A6H
Device file version	3FDCH, 3FDDH	20ABH, 20ACH
Assembler version	3FDEH to 3FE1H	20ADH to 20B0H

* If the source file is divided into a number of modules, the latest values for the creation year, month, day, hour, minute, are written in those modules.

Note The load module file should not be changed directly, as this will result in inconsistency between the load module file and other file histories, and will cause a bug. Reassembly must be performed when the load module file is changed.

★ **Table 5-1 Assembly Environment Information Area Items with Different Possibilities (3/4)**

(c) AS17217

Item	Addresses	
	ICE File	PRO File
Program name (Max. 64-byte character string specified by assembly option (/'PROG=')	7F30H to 7F60H	3000H to 303FH
Mask option data	7F70H, 7F71H	3040H, 3041H
Information for SIMPLEHOST	7FADH	307DH
Presence/absence of errors or warnings	7FB0H	3080H
Source file creation year, month, day, hour, minute*	7FBEH to 7FC7H	308EH to 3097H
Device name	7FC8H to 7FD7H	3098H to 30A7H
Device file version	7FDCH, 7FDDH	30ACH, 30ADH
Assembler version	7FDEH to 7FE1H	30AEH to 30B1H

* If the source file is divided into a number of modules, the latest values for the creation year, month, day, hour, minute, are written in those modules.

Note **The load module file should not be changed directly, as this will result in inconsistency between the load module file and other file histories, and will cause a bug. Reassembly must be performed when the load module file is changed.**

Table 5-1 Assembly Environment Information Area Items with Different Possibilities (4/4) ★

(d) AS17218

Item	Addresses	
	ICE File	PRO File
Program name (Max. 64-byte character string specified by assembly option (/'PROG=')	7F30H to 7F6FH	4000H to 403FH
Mask option data	7F70H, 7F71H	4040H, 4041H
Information for SIMPLEHOST	7FADH	407DH
Presence/absence of errors or warnings	7FB0H	4080H
Source file creation year, month, day, hour, minute*	7FBEH to 7FC7H	408EH to 4097H
Device name	7FC8H to 7FD7H	4098H to 40A7H
Device file version	7FDCH, 7FDDH	40ACH, 40ADH
Assembler version	7FDEH to 7FE1H	40AEH to 40B1H

* If the source file is divided into a number of modules, the latest values for the creation year, month, day, hour, minute, are written in those modules.

Note The load module file should not be changed directly, as this will result in inconsistency between the load module file and other file histories, and will cause a bug. Reassembly must be performed when the load module file is changed.