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.

RENESAS

USER'S MANUAL

AS17062
DEVICE FILE

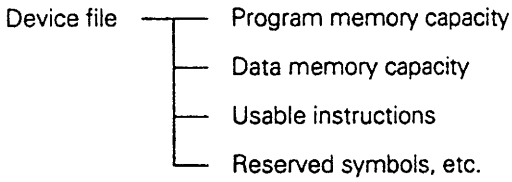
VERSION V1

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

Document No. EEU-1491
(O. D. No. EEU-936)
Date Published July 1994 P
Printed in Japan

INTRODUCTION

The AS17062 is a device file used together with the AS17K assembler to assemble the μPD17062 program. This device file contains the following information.



For how to handle the AS17K assembler and AS17062 device file, refer to the **AS17K USER'S MANUAL (EEU-603)**.

CONTENTS

CHAPTER 1 DEVICE INFORMATION

The AS17062 device file provides the following information on the μPD17062 during assembly.

(1) Program memory (ROM) capacity
3968x16 bits (0000H-0F7FH)

(2) Data memory (RAM) capacity
336x4 bits (BANK0 to BANK3)

(3) Usable instructions
Refer to **CHAPTER 2 μPD17062 INSTRUCTION SET**.

(4) Reading and writing information on register files, port registers, and peripheral registers.
Refer to **CHAPTER 3 RESERVED SYMBOLS**.

(5) Reserved symbols
Refer to **CHAPTER 3 RESERVED SYMBOLS**.

CHAPTER 2 μPD17062 INSTRUCTION SET

2.1 Outline of Instruction Set

b_{14}–b_{11} \ b_{15}		0		1	
BIN	HEX				
0000	0	ADD	r, m	ADD	m, #n4
0001	1	SUB	r, m	SUB	m, #n4
0010	2	ADDC	r, m	ADDC	m, #n4
0011	3	SUBC	r, m	SUBC	m, #n4
0100	4	AND	r, m	AND	m, #n4
0101	5	XOR	r, m	XOR	m, #n4
0110	6	OR	r, m	OR	m, #n4
0111	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		
1000	8	LD	r, m	ST	m, r
1001	9	SKE	m, #n4	SKGE	m, #n4
1010	A	MOV	@r, m	MOV	m, @r
1011	B	SKNE	m, #n4	SKLT	m, #n4
1100	C	BR	addr (Page 0)	CALL	addr (Page 0)
1101	D	BR	addr (Page 1)	MOV	m, #n4
1110	E			SKT	m, #n
1111	F			SKF	m, #n

2.2 Legends

AR	:	Address register
$(AR)_{ROM}$	:	Contents of ROM with the address specified by AR
ASR	:	Address stack register specified by the 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	:	Register automatically stored in the stack during interruption
INTSK	:	Interruption stack register
IX	:	Index register
IXE	:	Index enable flag
MP	:	Data memory row address pointer
MPE	:	Memory pointer enable flag
m	:	Data memory address represented by m_R and m_C
m_R	:	Data memory row address (Higher)
m_C	:	Data memory column address (Lower)
n	:	Bit position (4 bits)
n4	:	Immediate data (4 bits)
PAGE	:	Page (Program counter's bit 11)
PC	:	Program counter
p	:	Peripheral address
p_H	:	Peripheral address (Higher 3 bits)
p_L	:	Peripheral address (Lower 4 bits)
r	:	General register column address
rf	:	Register file address
rf_R	:	Register file address (Higher 3 bits)
rfc	:	Register file address (Lower 4 bits)
SP	:	Stack pointer
s	:	Stop release condition
WR	:	Window register
(x)	:	Contents addressed by x x: Direct addresses such as m and r or a register such as ASR.

2.3 List of Instructions

Instruction Set	Mnemonic	Operand	Operation	Machine Code			
				Operation Code	Operand		
Addition	ADD	r, m	(r)←(r)+(m)	00000	mR	mc	r
		m, #n4	(m)←(m)+n4	10000	mR	mc	n4
	ADDC	r, m	(r)←(r)+(m)+CY	00010	mR	mc	r
		m, #n4	(m)←(m)+n4+CY	10010	mR	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	mR	mc	r
		m, #n4	(m)←(m)–n4	10001	mR	mc	n4
	SUBC	r, m	(r)←(r)–(m)–CY	00011	mR	mc	r
		m, #n4	(m)←(m)–n4–CY	10011	mR	mc	n4
Logical Operation	OR	r, m	(r)←(r)∨(m)	00110	mR	mc	r
		m, #n4	(m)←(m)∨n4	10110	mR	mc	n4
	AND	r, m	(r)←(r)∧(m)	00100	mR	mc	r
		m, #n4	(m)←(m)∧n4	10100	mR	mc	n4
	XOR	r, m	(r)←(r)∀(m)	00101	mR	mc	r
		m, #n4	(m)←(m)∀n4	10101	mR	mc	n4
Decision	SKT	m, #n	CMP←0, if(m)∧n=n, then skip	11110	mR	mc	n
	SKF	m, #n	CMP←0, if(m)∧n=0, then skip	11111	mR	mc	n
Comparison	SKE	m, #n4	(m)–n4, skip if zero	01001	mR	mc	n4
	SKNE	m, #n4	(m)–n4, skip if not zero	01011	mR	mc	n4
	SKGE	m, #n4	(m)–n4, skip if not borrow	11001	mR	mc	n4
	SKLT	m, #n4	(m)–n4, skip if borrow	11011	mR	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	mR	mc	r
	ST	m, r	(m)←(r)	11000	mR	mc	r
	MOV	@r, m	if MPE=1: (MP, (r))←(m) if MPE=0: (BANK, mR, (r))←(m)	01010	mR	mc	r
		m, @r	if MPE=1: (m)←(MP, (r)) if MPE=0: (m)←(BANK, mR, (r))	11010	mR	mc	r
		m, #n4	(m)←n4	11101	mR	mc	n4

Instruction Set	Mnemonic	Operand	Operation	Machine Code			
				Operation Code	Operand		
Transfer	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, PAGE←0	01100	addr		
			PC$_{10-0}$←addr, PAGE←1	01101			
		@AR	PC←AR	00111	000	0100	0000
Sub-Routine	CALL	addr	SP←SP−1, ASR←PC+1, PC$_{11}$←0, 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
Interrup-tion	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

CHAPTER 3 RESERVED SYMBOLS

The symbols defined by the μPD17062 device file are shown from the next page.
These symbols are as follows.

- Data buffer
- System register
- Port register
- Register file (Control register)
- Peripheral register

3.1 Data Buffer

Symbol	Attribute	Value	R/W	Function
DBF3	MEM	0.0CH	R/W	Bits 15 to 12 of the data buffer
DBF2	MEM	0.0DH	R/W	Bits 11 to 8 of the data buffer
DBF1	MEM	0.0EH	R/W	Bits 7 to 4 of the data buffer
DBF0	MEM	0.0FH	R/W	Bits 3 to 0 of the data buffer

3.2 System Register

Symbol	Attribute	Value	R/W	Function
AR3	MEM	0.74H	R	Bits 15 to 12 of the address register
AR2	MEM	0.75H	R	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	Bits 10 to 8 of the index register
MPH	MEM	0.7AH	R	Bits 6 to 4 of the memory pointer
MPE	FLG	0.7AH.3	R/W	Memory pointer enable flag
IXM	MEM	0.7BH	R/W	Bits 7 to 4 of the index register
MPL	MEM	0.7BH	R/W	Bits 3 to 0 of the memory pointer
IXL	MEM	0.7CH	R/W	Bits 3 to 0 of the index register
RPH	MEM	0.7DH	R	Bits 6 to 3 of the register pointer
RPL	MEM	0.7EH	R/W	Bits 2 to 0 of the register pointer
PSW	MEM	0.7FH	R/W	Program status word
BCD	FLG	0.7EH.0	R/W	BCD operation 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 Register

Symbol	Attribute	Value	R/W	Function
P0A3	FLG	0.70H.3	R/W	Bit 3 of port 0A
P0A2	FLG	0.70H.2	R/W	Bit 2 of port 0A
P0A1	FLG	0.70H.1	R/W	Bit 1 of port 0A
P0A0	FLG	0.70H.0	R/W	Bit 0 of port 0A
P0B3	FLG	0.71H.3	R/W	Bit 3 of port 0B
P0B2	FLG	0.71H.2	R/W	Bit 2 of port 0B
P0B1	FLG	0.71H.1	R/W	Bit 1 of port 0B
P0B0	FLG	0.71H.0	R/W	Bit 0 of port 0B
P0C3	FLG	0.72H.3	R/W	Bit 3 of port 0C
P0C2	FLG	0.72H.2	R/W	Bit 2 of port 0C
P0C1	FLG	0.72H.1	R/W	Bit 1 of port 0C
P0C0	FLG	0.72H.0	R/W	Bit 0 of port 0C
P0D3	FLG	0.73H.3	R[Note]	Bit 3 of port 0D
P0D2	FLG	0.73H.2	R[Note]	Bit 2 of port 0D
P0D1	FLG	0.73H.1	R[Note]	Bit 1 of port 0D
P0D0	FLG	0.73H.0	R[Note]	Bit 0 of port 0D
P1A3	FLG	1.70H.3	R/W	Bit 3 of port 1A
P1A2	FLG	1.70H.2	R/W	Bit 2 of port 1A
P1A1	FLG	1.70H.1	R/W	Bit 1 of port 1A
P1A0	FLG	1.70H.0	R/W	Bit 0 of port 1A
P1B3	FLG	1.71H.3	R/W	Bit 3 of port 1B
P1B2	FLG	1.71H.2	R/W	Bit 2 of port 1B
P1B1	FLG	1.71H.1	R/W	Bit 1 of port 1B
P1B0	FLG	1.71H.0	R/W	Bit 0 of port 1B
P1C3	FLG	1.72H.3	R/W	Bit 3 of port 1C
P1C2	FLG	1.72H.2	R/W	Bit 2 of port 1C
P1C1	FLG	1.72H.1	R/W	Bit 1 of port 1C

Note Although these ports are for inputting only, the IE-17K assembler will not output error messages even if output instructions are written into these ports.
No problems will occur when they are executed on the device.

3.4 Register File

Symbol	Attribute	Value	R/W	Function
IDCDMAEN	FLG	0.80H.1	R/W	DMA enable flag
SP	MEM	0.81H	R/W	Stack pointer
CE	FLG	0.87H.0	R	CE pin status flag
SIO0CH	FLG	0.88H.3	R/W	SIO0 channel select flag
SB	FLG	0.88H.2	R/W	SIO0 mode select flag
SIO0MS	FLG	0.88H.1	R/W	SIO0 clock mode select flag
SIO0TX	FLG	0.88H.0	R/W	SIO0 TX/RX select flag
BTM0ZX	FLG	0.89H.3	R/W	Timer 0 interrupt mode select flag
BTM0CK2	FLG	0.89H.2	R/W	Timer 0 carry FF mode select flag
BTM0CK1	FLG	0.89H.1	R/W	Timer 0 carry FF mode select flag
BTM0CK0	FLG	0.89H.0	R/W	Timer 0 carry FF mode select flag
INTVSYN	FLG	0.8FH.2	R	Vsync pin status flag
INTNC	FLG	0.8FH.0	R	RMC pin status
HSCGT3	FLG	0.91H.3	R/W	Hsync counter mode select flag (Dummy:0)
HSCGT2	FLG	0.91H.2	R/W	Hsync counter mode select flag (Dummy:0)
HSCGT1	FLG	0.91H.1	R/W	Hsync counter mode select flag
HSCGT0	FLG	0.91H.0	R/W	Hsync counter mode select flag
HSCGOSTT	FLG	0.92H.3	R	Hsync counter gate open flag
PLLRFCK3	FLG	0.93H.3	R/W	PLL reference clock select flag
PLLRFCK2	FLG	0.93H.2	R/W	PLL reference clock select flag
PLLRFCK1	FLG	0.93H.1	R/W	PLL reference clock select flag
PLLRFCK0	FLG	0.93H.0	R/W	PLL reference clock select flag
INTNCMD3	FLG	0.95H.3	R/W	RMC pin status flag (Dummy)
INTNCMD2	FLG	0.95H.2	R/W	RMC pin status flag
INTNCMD1	FLG	0.95H.1	R/W	RMC pin status flag
INTNCMD0	FLG	0.95H.0	R/W	RMC pin status flag
BTM0CY	FLG	0.97H.0	R	Timer 0 carry FF status flag

Symbol	Attribute	Value	R/W	Function
SBACK	FLG	0.98H.3	R/W	Serial bus acknowledge flag
SIO0NWT	FLG	0.98H.2	R/W	SIO0 no wait flag
SIO0WRQ1	FLG	0.98H.1	R/W	SIO0 wait request flag
SIO0WRQ0	FLG	0.98H.0	R/W	SIO0 wait request flag
IEGVSYN	FLG	0.9FH.2	R/W	Vsync interrupt edge select flag
IEGNC	FLG	0.9FH.0	R/W	RMC interrupt edge select flag
ADCCH2	FLG	0.0A1H.3	R/W	A/D converter channel select flag
ADCCH1	FLG	0.0A1H.2	R/W	A/D converter channel select flag
ADCCH0	FLG	0.0A1H.1	R/W	A/D converter channel select flag
ADCCMP	FLG	0.0A1H.0	R/W	A/D converter judge flag
PLLUL	FLG	0.0A2H.0	R	PLL unlock FF flag
P1CGIO	FLG	0.0A7H.0	R/W	Port 1C I/O select flag
SIO0SF8	FLG	0.0A8H.3	R	SIO0 shift 8 clock flag
SIO0SF9	FLG	0.0A8H.2	R	SIO0 shift 9 clock flag
SBSTT	FLG	0.0A8H.1	R	Serial bus start test flag
SBBSY	FLG	0.0A8H.0	R	Serial bus busy flag
IPSIO0	FLG	0.0AFH.3	R/W	SIO0 interrupt permission flag
IPVSYN	FLG	0.0AFH.2	R/W	Vsync interrupt permission flag
IPBTM0	FLG	0.0AFH.1	R/W	Timer 0 interrupt permission flag
IPNC	FLG	0.0AFH.0	R/W	RMC interrupt permission flag
CROMBNK	FLG	0.0B0H.0	R/W	CROM bank select flag
IDCEN	FLG	0.0B1H.0	R/W	IDC enable flag
PLULSEN3	FLG	0.0B2H.3	R/W	PLL unlock time select flag (Dummy:0)
PLULSEN2	FLG	0.0B2H.2	R/W	PLL unlock time select flag (Dummy:0)
PLULSEN1	FLG	0.0B2H.1	R/W	PLL unlock time select flag
PLULSEN0	FLG	0.0B2H.0	R/W	PLL unlock time select flag
P1BBIO3	FLG	0.0B5H.3	R/W	P1B3 I/O select flag
P1BBIO2	FLG	0.0B5H.2	R/W	P1B2 I/O select flag
P1BBIO1	FLG	0.0B5H.1	R/W	P1B1 I/O select flag
P1BBIO0	FLG	0.0B5H.0	R/W	P1B0 I/O select flag

Symbol	Attribute	Value	R/W	Function
P0BBIO3	FLG	0.0B6H.3	R/W	P0B3 I/O select flag
P0BBIO2	FLG	0.0B6H.2	R/W	P0B2 I/O select flag
P0BBIO1	FLG	0.0B6H.1	R/W	P0B1 I/O select flag
P0BBIO0	FLG	0.0B6H.0	R/W	P0B0 I/O select flag
P0ABIO3	FLG	0.0B7H.3	R/W	P0A3 I/O select flag
P0ABIO2	FLG	0.0B7H.2	R/W	P0A2 I/O select flag
P0ABIO1	FLG	0.0B7H.1	R/W	P0A1 I/O select flag
P0ABIO0	FLG	0.0B7H.0	R/W	P0A0 I/O select flag
SIO0IMD3	FLG	0.0B8H.3	R/W	SIO0 interrupt mode select flag (Dummy:0)
SIO0IMD2	FLG	0.0B8H.2	R/W	SIO0 interrupt mode select flag (Dummy:0)
SIO0IMD1	FLG	0.0B8H.1	R/W	SIO0 interrupt mode select flag
SIO0IMD0	FLG	0.0B8H.0	R/W	SIO0 interrupt mode select flag
SIO0CK3	FLG	0.0B9H.3	R/W	SIO0 shift clock select flag (Dummy:0)
SIO0CK2	FLG	0.0B9H.2	R/W	SIO0 shift clock select flag (Dummy:0)
SIO0CK1	FLG	0.0B9H.1	R/W	Serial clock selection
SIO0CK0	FLG	0.0B9H.0	R/W	Serial clock selection
IRQSIO0	FLG	0.0BFH.3	R	SIO0 interruption request flag
IRQVSYN	FLG	0.0BFH.2	R	Vsync interruption request flag
IRQBTM0	FLG	0.0BFH.1	R	Timer 0 interruption request flag
IRQNC	FLG	0.0BFH.0	R	RMC pin interruption request flag

3.5 Peripheral Registers

Symbol	Attribute	Value	R/W	Function
IDCORG	DAT	01H	R/W	IDC start position setting register
ADCR	DAT	02H	R/W	A/D converter V_{REF} data register
SIO0SFR	DAT	03H	R/W	SIO0 register
HSC	DAT	04H	R	Hsync counter data register
PWMR0	DAT	05H	R/W	PWM data register 0
PWMR1	DAT	06H	R/W	PWM data register 1
PWMR2	DAT	07H	R/W	PWM data register 2
PWMR3	DAT	08H	R/W	PWM data register 3
AR	DAT	40H	R/W	Address register
PLLR	DAT	41H	R/W	PLLR data register
DBF	DAT	0FH	—	Data buffer (Can be used only as GET/PUT/MOVT instruction operand.)
IX	DAT	01H	—	Index register (Can be used only as INC instruction operand.)
AR_EPA1	DAT	8040H	—	CALL/BR/MOVT instruction operand (EPA bit on)
AR_EPA0	DAT	4040H	—	CALL/BR/MOVT instruction operand (EPA bit off)

3.6 List of Reserved Words (Alphabetical order)

3.6.1 Instructions and pseudo instructions

ADD	ADDC	AND	BANK0
BANK1	BANK2	BELOW	BR
C14344	C4444	CALL	CASE
CLR1	CLR2	CLR3	CLR4
CSEG	DAT	DB	DCP
DI	DW	EI	EJECT
ELSE	END	ENDCASE	ENDIF
ENDIFC	ENDIFNC	ENDM	ENDP
ENDR	EOF	EXIT	EXITR
EXTRN	FLG	GET	GLOBAL
HALT	IF	IFCHAR	IFNCHAR
INC	INCLUDE	INITFLG	IRP
LAB	LBMAC	LD	LFCOND
LIST	LITERAL	LMAC	MACRO
MEM	MOV	MOVT	NIBBLE
NIBBLE1	NIBBLE2	NIBBLE2V	NIBBLE3
NIBBLE3V	NIBBLE4	NIBBLE4V	NIBBLE5
NIBBLE5V	NIBBLE6	NIBBLE6V	NIBBLE7
NIBBLE7V	NIBBLE8	NIBBLE8V	NOBMAC
NOLIST	NOMAC	NOP	NOT1
NOT2	NOT3	NOT4	OBMAC
OMAC	OR	ORG	OTHER
PEEK	POKE	POP	PUBLIC
PURGE	PUSH	PUT	REPT
RET	RETI	RETSK	RORC
SBMAC	SET	SET1	SET2
SET3	SET4	SFCOND	SKE
SKF	SKF1	SKF2	SKF3
SKF4	SKGE	SKLT	SKNE
SKT	SKT1	SKT2	SKT3
SKT4	SMAC	ST	STOP
SUB	SUBC	SUMMARY	TAG
TITLE	XOR	ZZZERROR	ZZZMCHK
ZZZMSG			

3.6.2 Registers and flags

ADCCH0	ADCCH1	ADCCH2	ADCCMP
ADCR	AR	AR0	AR1
AR2	AR3	AR_EPA0	AR_EPA1
BANK	BCD	BTM0CK0	BTM0CK1
BTM0CK2	BTM0CY	BTM0ZX	CE
CMP	CROMBNK	CY	DBF
DBF0	DBF1	DBF2	DBF3
HSC	HSCGOSTT	HSCGT0	HSCGT1
HSCGT2	HSCGT3	IDCDMAEN	IDCEN
IDCORG	IEGNC	IEGVSYN	INTNC
INTNCMD0	INTNCMD1	INTNCMD2	INTNCMD3
INTVSYN	IPBTM0	IPNC	IPSIO0
IPVSYN	IRQBTM0	IRQNC	IRQSIO0
IRQVSYN	IX	IXE	IXL
IXM	MPE	MPL	P0A0
P0A1	P0A2	P0A3	P0ABIO0
P0ABIO1	P0ABIO2	P0ABIO3	P0B0
P0B1	P0B2	P0B3	P0BBIO0
P0BBIO1	P0BBIO2	P0BBIO3	P0C0
P0C1	P0C2	P0C3	P0D0
P0D1	P0D2	P0D3	P1A0
P1A1	P1A2	P1A3	P1B0
P1B1	P1B2	P1B3	P1BBIO0
P1BBIO1	P1BBIO2	P1BBIO3	P1C1
P1C2	P1C3	P1CGIO	PLLR
PLLRFCK0	PLLRFCK1	PLLRFCK2	PLLRFCK3
PLLUL	PLULSEN0	PLULSEN1	PLULSEN2
PLULSEN3	PSW	PWMR0	PWMR1
PWMR2	PWMR3	RPH	RPL
SB	SBACK	SBBSY	SBSTT
SIO0CH	SIO0CK0	SIO0CK1	SIO0CK2
SIO0CK3	SIO0IMD0	SIO0IMD1	SIO0IMD2
SIO0IMD3	SIO0MS	SIO0NWT	SIO0SF8
SIO0SF9	SIO0SFR	SIO0TX	SIO0WRQ0
SIO0WRQ1	SP	WR	Z
ZZZ0	ZZZ1	ZZZ2	ZZZ3
ZZZ4	ZZZ5	ZZZ6	ZZZ7
ZZZ8	ZZZ9	ZZZALBMAC	ZZZALMAC
ZZZARGC	ZZZDEVID	ZZZEPA	ZZZLINE
ZZZLSARG	ZZZPRINT	ZZZSKIP	ZZZSYDOC

CHAPTER 4 CHARACTER PATTERN DEFINITION PSEUDO INSTRUCTIONS

The AS17062 has DCP (Define Character Pattern) pseudo instructions for defining character patterns for the IDC (Image Display Controller).

(1) Description Format

Symbol Column	Mnemonic Column	Operand Column	Comment Column
[Label:]	DCP	Equation, Display Pattern	[;Comment]

(2) Explanation

(a) The equation is represented by value "0" or "1" and they specify whether or not to fringe the display pattern described by the 2nd operand.

"0": Does not fringe
"1": Fringes

When the evaluation value of the equation is not "0" or "1", it indicates an error.

(b) Only three types of characters are used for the display pattern. They are "○", "#", " " (space). The pattern is described in 10 characters.
When the pattern is described using characters other than these, or more/less than 10 characters are used, errors will result. These three characters are equivalent to one dot of the display pattern and result in the following.

"○": Lights up
"#" : Fringes
" " : Blank

When the evaluation value obtained from the equation of the 1st operand is "0", "#" cannot be used for the display pattern.

CHAPTER 5 PRECAUTIONS FOR USING THE DEVICE FILE

5.1 Precautions for Using Writing Instructions for the IRQxxx Flag

5.1.1 Precautions

Do not write data in address 3FH (BFH) of register files containing IRQxxx (IRQNC, IRQBTM0, IRQVSYN, IRQSIO0) flags using the following instructions because assemble errors will result.

- Writing instructions (POKE)
- On-chip macro instructions (SETn, CLRn, NOTn, INITFLG)

5.1.2 Writing instruction prohibition reason

The IRQxxx flag is set when interruption requests are issued. When interruptions can be received, these flags will be reset.

Use the following programs to set or reset the IRQxxx flag (E.g. when initializing the interruption request or processing polling for the interruption request).

Example To reset the IRQSIO0 flag

```
PEEK   WR,.MF. IRQSIO0 SHR 4              ; ①
AND    WR, #.DF. (NOT IRQSIO0) AND 0FH;   ②
POKE   .MF. IRQSIO0 SHR 4, WR             ; ③
```

In the above example, even if the IRQxxx flag in the same address is changed while instructions ① and ② are executed, the IRQxxx flag will return to its original state since the value read with the ① instruction is written into the ③ instruction. To prevent such problems, assemble errors are generated when data is written in the IRQxxx flag.

5.1.3 Writing

When executing writing instructions for the IRQxxx flag, assemble errors result. However, if this flag must be changed for an application program, use the following method.

First, read the IRQ. MAC file of the device file using the INCLUDE instruction. Execute the macro instructions defined by the IRQ. MAC file. These macro instructions are as follows.

```
SETIRQn
CLRIRQn
NOTIRQn
INITIRQ
```

Assemble errors do not result when these macro instructions are executed. However, error messages calling for caution will be output.

Remark For details on the on-chip macro instructions, refer to the **"AS17K assembler user's manual (EEU-603)"**.

CHAPTER 6 LOAD MODULE FILE FORMAT

The HEX load module file output by the AS17K assembler has two types of output formats—ICE file and PRO file.

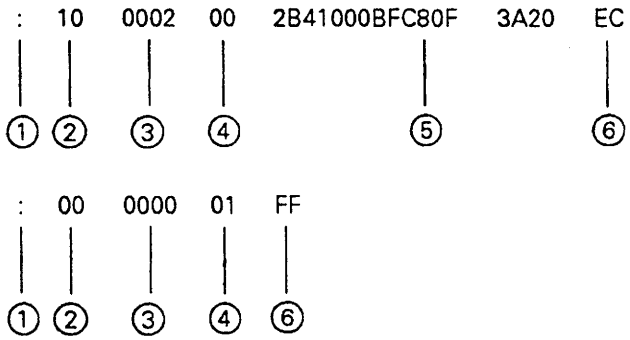
These files must be selected according to the purpose of use.

They have areas for user programs, assembler environment information, in-circuit emulator operation environment information, etc.

(1) Format of HEX Load Module File

The data in each HEX load module file output by the assembler are output in the following format.

[Example of HEX Load Module File Format]



① Record Mark

Indicates the start of recording.

② Code Number (2 Digits)

Indicates the number of codes (byte data) stored in the record. It is displayed as an hexadecimal number and is 10H at the maximum (16 codes). It is 00H for the last record.

③ Address (4 Digits)

Indicates the start address of the code representing the record. It is 0000H for the last record and is not related to addresses.

④ Record Type (2 Digits)

When it is 00H, it indicates that the record is a data record. When it is 01H, it indicates that the record is the last one.

⑤ Code (Max. 32 Digits (16 Bytes))

Up to 16 bytes will be output to this field one byte at a time.

⑥ Check Sum (2 Digits)

Outputs to ⑥ byte data that makes the lowermost byte become 00H when the data of ②, ③, ④, ⑤, and ⑥ are totaled up in bytes. (Even parity)

(2) ICE File

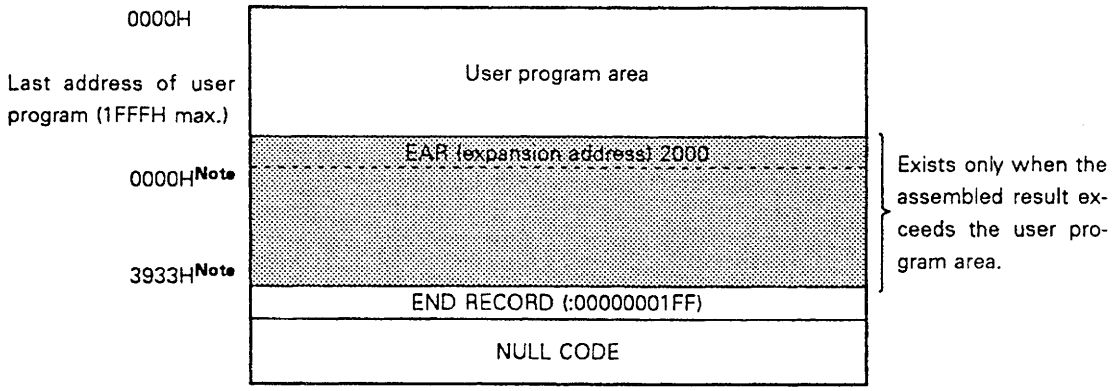
File output in the HEX format for the in-circuit emulator (IE-17K or IE-17K-ET or EMU-17K[Note]) by AS17K. When assembling using the μSxxxAS17062, the output format will be as shown in Figure 6-1.

Note Manufactured by I.C Co., Ltd.

The ICE file consists of two files. The first is the area for the program. It is divided into the user program area and patch area. The patch area exists only when a patch is created on the in-circuit emulator. The second file contains the in-circuit emulator operation environment information area, assemble environment information area, and SE board environment information area. These areas contain information for regulating in-circuit emulator operations.

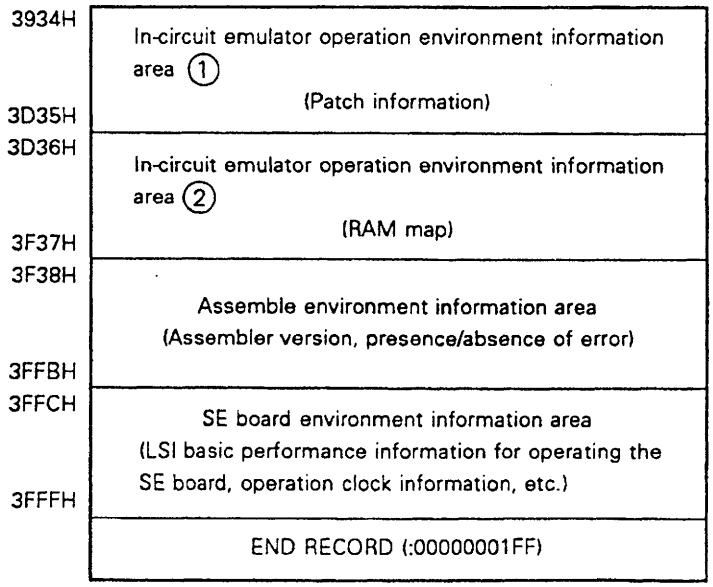
Figure 6-1. ICE File Format

1st file.... Program Area



Note With the in-circuit emulator, it is 8000H to B933H.

2nd file .. In-circuit emulator operating environment area + Assemble environment information area



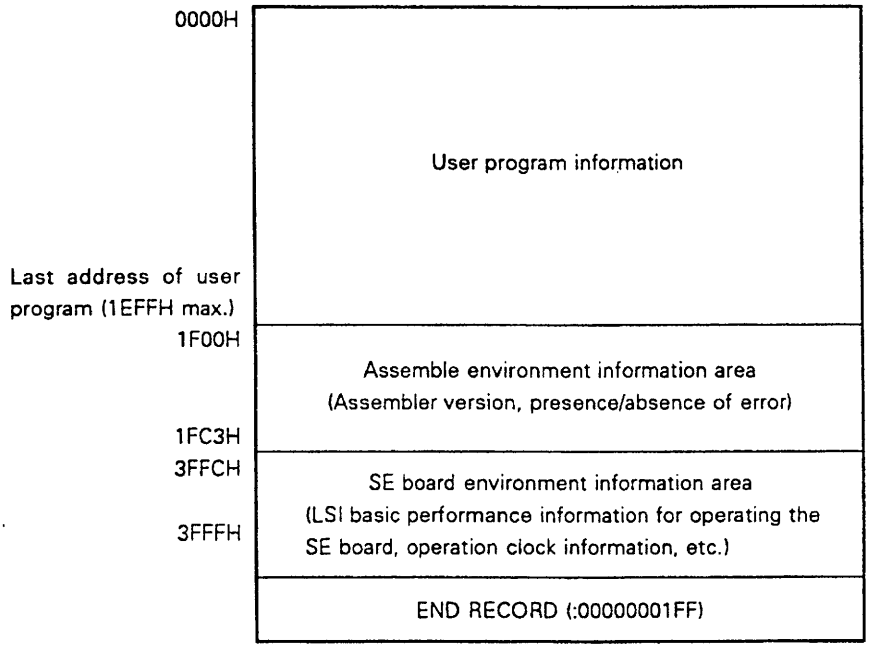
(3) PRO File

HEX data output by AS17K for the PROM and one time PROM products used for ordering masks or for evaluating SE board by itself. These are output during assembly when /PRO is specified at the assemble option.

The output format when assembling using the μSxxxxAS17062 is shown in Figure 6-2.

The PRO file consists of one file. It contains the user program area, assemble environment information area, and SE board environment information area.

Figure 6-2. PRO File Format



Remark The addresses 1FC4H to 3FFBH do not exist in the PRO file.

(4) Load Module File Comparison

Even when source files are not changed, the outputs of the assembler (assemble environment information area) may differ. This is because this assemble environment information area contains information such as the date the source file was created.

Table 6-1. Items with which assembler outputs may differ even when the source file has not been changed

Item	Address	
	ICE file	PRO file
Program name Character string specified with assemble option (/'PROG=') (Max. 32 bytes)	3F38H-3F57H	1F00H-1F1FH
SIMPLEHOST™ information	3FADH	1F75H
Presence/absence of errors or warnings	3FB0H	1F78H
Source file creation year, month, date, time, and minute[Note]	3FBEH-3FC7H	1F86H-1F8FH
Device file version	3FDDH	1FA5H
Assembler version	3FE1H	1FA9H

Note If the source file is divided into several modules, the latest year, month, date, time, and minute of the module will be written.

Caution Do not change only the load module file.
Change the load module file by changing the source file and assembling. If only the load module file is changed, its history will not match that of other files and this will result in bugs.