MF1134-01

EPSON

CMOS 4-BIT SINGLE CHIP MICROCOMPUTER **E0C6007**

DEVELOPMENT TOOL MANUAL

ENERGY SAVING EPSON

SEIKO EPSON CORPORATION

E0C6007 Development Tool Manual

PREFACE

This manual mainly explains the outline of the development support tool for the 4-bit Single Chip Microcomputers E0C6007.

Refer to the "E0C62 Family Development Tool Reference Manual" for the details (common to all models) of each development support tool. Manuals for hardware development tools are separate, so you should also refer to the below manuals.

Development tools	☞ E0C62 Family Development Tool Reference Manual EVA6007 Manual ICE6200 Hardware Manual
Development procedure	☞ E0C62 Family Technical Guide
Device (E0C6007)	☞ E0C6007 Technical Manual
Instructions	☞ E0C6200/6200A Core CPU Manual

CONTENTS

1 COMPOSITION OF DEVELOPMENT SUPPORT TOOL

Here we will explain the composition of the software for the development support tools, developmental enviroment and how to generate the execution disk.

1.1 Configuration of DEV6007

The below software are included in the product of the E0C6007 development support tool DEV6007.

1. Cross Assembler ASM6007 ..... Cross assembler for program preparation
2. Function Option Generator FOG6007 ..... Function option data preparation program
3. ICE Control Software ICS6007 ..... ICE control program
4. Mask Data Checker MDC6007 ..... Mask data preparation program

1.2 Developmental Environment

The software product of the development support tool DEV6007 operates on the following host systems:

- IBM PC/AT (at least PC-DOS Ver. 2.0)
- NEC PC-9801 Series (at least MS-DOS Ver. 3.1)

When developing the E0C6007, the above-mentioned host computer, editor, P-ROM writer, printer, etc. must be prepared by the user in addition to the development tool which is normally supported by Seiko Epson.

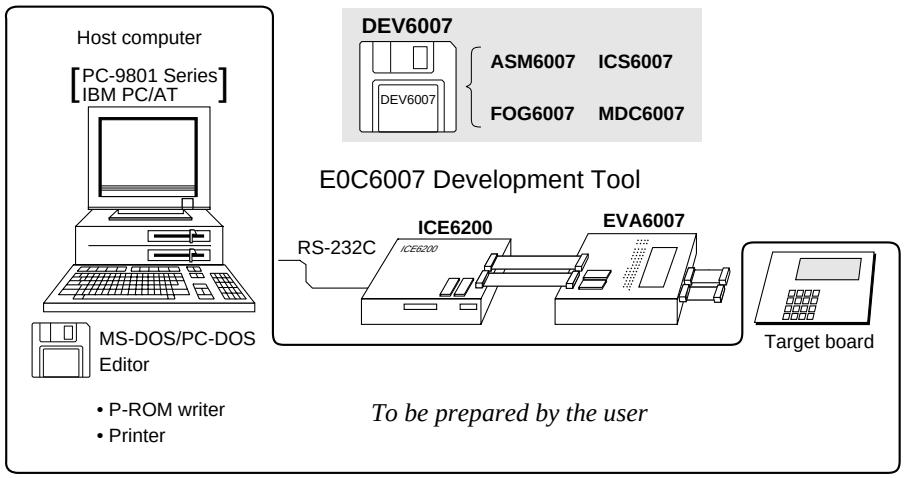

Fig. 1.2.1 System configuration

Note The DEV6007 system requires a host computer with a RAM capacity of about 140K bytes. Since the ICE6200 is connected to the host computer with a RS-232C serial interface, adapter board for asynchronous communication will be required depending on the host computer used.

1.3 Development Flow

Figure 1.3.1 shows the development flow through the DEV6007.

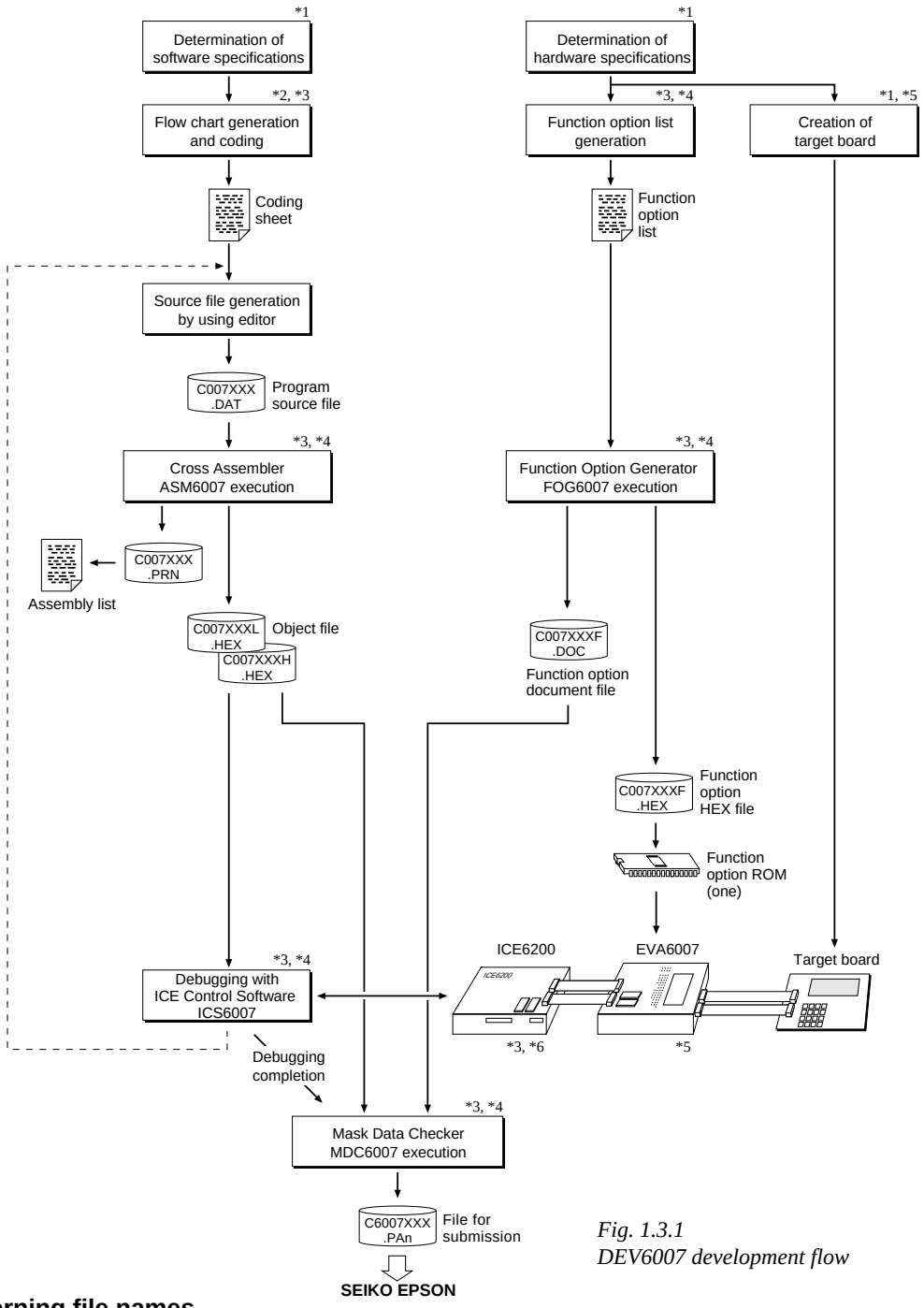

Fig. 1.3.1
DEV6007 development flow

Concerning file names

All the input-output file name for the each development support tool commonly use "C007XXX".
In principle each file should be produced in this manner. Seiko Epson will designate the "XXX" for each customer.

☞ Reference Manual

*1 E0C6007 Technical Hardware Manual
*2 E0C6200/6200A Core CPU Manual
*3 E0C62 Family Development Tool Reference Manual
*4 E0C6007 Development Tool Manual (this manual)
*5 EVA6007 Manual
*6 ICE6200 Hardware Manual

1.4 Production of Execution Disk

Execution files for each software development support tool and batch and parameter files for the ICE6200 are recorded in the DEV6007 floppy disk.
The content of the files contained in the DEV6007 floppy disk are shown below.

PC-DOS version	MS-DOS version	Contents
ASM6007.EXE	ASM6007.EXE	Cross Assembler execution file
FOG6007.EXE	FOG6007.EXE	Function Option Generator execution file
ICS6007B.BAT	ICS6007.BAT	ICE Control Software batch file
ICS6007W.EXE	ICS6007J.EXE	ICE Control Software execution file
ICS6007P.PAR	ICS6007P.PAR	ICE Control Software parameter file
MDC6007.EXE	MDC6007.EXE	Mask Data Checker execution file

- First copy the entire content of this disk using commands such as DISKCOPY then make the execution disk. Carefully conserve the original floppy disk for storage purposes.
  When copying into a hard disk, make a sub-directory with an appropriate name (DEV6007, etc.) then copy the content of the floppy disk into that subdirectory using the COPY command.

Example:

Copying into a floppy disk

Insert the original disk into the A drive and the formatted disk to be copied into B drive, then execute the DISKCOPY command.

```
A>DISKCOPY  A:  B:↵
```

Copying into a hard disk (C drive)

Make a subdirectory (example: DEV6007), then insert the original disk into the A drive and execute the COPY command.

```
C\>MD DEV6007↵
C\>CD DEV6007↵
C\DEV6007\>COPY A:*.*↵
```

- Next make a CONFIG.SYS file using Editor or the like. When a CONFIG.SYS has previously been made using a hard disk system, check the setting of the FILES within it. (If there is none add it.)
  Set the number of files to be described in CONFIG.SYS at 10 or more, so that the Mask Data Checker MDC6007 will handle many files.

Note *The driver for the RS-232C must be included in CONFIG.SYS by the host computer.*

Example:

Setting of FILES (CONFIG.SYS)

```
C\>TYPE  CONFIG.SYS↵
      :
FILES=20
      :
```

- In "ICS6007(B).BAT" the batch process is indicated such that the ICS6007J(W).EXE is executed after the execution of the command for the setting of the RS-232C communication parameters.
  The SPEED (MS-DOS) or MODE (PC-DOS) command is used for setting the RS-232C, so you should copy these commands into the disk to be executed or specify a disk or directory with the command using the PATH command.

RS-232C Setting (PC-DOS version)

```
MODE  COM1:  4800, n, 8, 1, p
```

RS-232C Setting (MS-DOS version)

```
SPEED  R0  9600  B8  PN  S1
```

2 CROSS ASSEMBLER ASM6007

2.1 ASM6007 Outline

The ASM6007 cross assembler is an assembler program for generating the machine code used by the E0C6007 4-bit, single-chip microcomputers. The Cross Assembler ASM6007 will assemble the program source files which have been input by the user's editor and will generate an object file in Intel-Hex format and assembly list file.
In this assembler, program modularization has been made possible through macro definition functions and programming independent of the ROM page structure has been made possible through the auto page set function. In addition, consideration has also been given to precise error checks for program capacity (ROM capacity) overflows, undefined codes and the like, and for debugging of such things as label tables for assembly list files and cross reference table supplements.

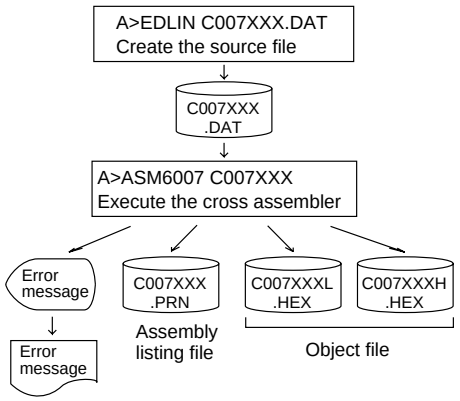

Fig. 2.1.1 ASM6007 execution flow

☞ *The format of the source file and its operating method are same as for the E0C62 Family. Refer to the "E0C62 Family Development Tool Reference Manual" for details.*

2.2 E0C6007 Restrictions

Note the following when generating a program by the E0C6007:

■ ROM area

The capacity of the E0C6007 ROM is 4K steps (0000H–0FFFH).
Therefore, the specification range of the memory setting pseudo-instructions and PSET instruction is restricted.

Memory configuration:

Bank: Only bank 0, Page: 16 pages (00H–0FH), each 256 steps

Significant specification range:

ORG	pseudo-instruction:	0000H–0FFFH
PAGE	pseudo-instruction:	00H–0FH
BANK	pseudo-instruction:	Only 0H
PSET	instruction:	00H–0FH

■ RAM area

The capacity of the E0C6007 RAM is 705 words (000H–1FFH, E00H–E4FH, E80H–ECFH, F00H–F7FH, 4 bits/word).
Memory access is invalid when the unused area of the index register is specified.

Example:

```
LD   A,03H
LD   XP,A
LD   X,80H
```

380H is loaded into the IX register, but an unused area has been specified so that the memory accessible with the IX register (MX) is invalid.

```
LD   B,0AH
LD   YP,B
LD   Y,80H
```

A80H is loaded into the IY register, but an unused area has been specified so that the memory accessible with the IY register (MY) is invalid.

■ Undefined codes

The SLP instruction has not been defined in the E0C6007 instruction sets.

2.3 ASM6007 Quick Reference

■ Starting command and input/output files

_ indicates a blank.
↵ indicates the Return key.
A parameter enclosed by [ ] can be omitted.

Execution file: ASM6007.EXE

Starting command: **ASM6007_ [drive-name:] source-file-name [.shp] _ [-N]↵**

Option:

.shp		Specifies the file I/O drives.
	s	Specifies the drive from which the source file is to be input. (A–P, @)
	h	Specifies the drive to which the object file is to be output. (A–P, @, Z)
	p	Specifies the drive to which the assembly listing file is to be output. (A–P, @, Z) @: Current drive, Z: File is not generated
-N		The code (FFH) in the undefined area of program memory is not created.

Input file: C007XXX.DAT (Source file)

Output file: C007XXXL.HEX (Object file, low-order)
C007XXXH.HEX (Object file, high-order)
C007XXX.PRN (Assembly listing file)

■ Display example (In case of ASM6007)

```
          *** E0C6007 CROSS ASSEMBLER. --- Ver 1.00 ***

EEEEEEEEEE   PPPPPPPP        SSSSSSS       OOOOOOOO      NNN      NNN
EEEEEEEEEE   PPPPPPPPPP     SSS   SSSS    OOO    OOO    NNNN     NNN
EEE          PPP     PPP   SSS     SSS   OOO      OOO   NNNNN    NNN
EEE          PPP     PPP    SSS          OOO      OOO   NNNNNN   NNN
EEEEEEEEEE   PPPPPPPPPP      SSSSSS      OOO      OOO   NNN NNN NNN
EEEEEEEEEE   PPPPPPPP           SSSS     OOO      OOO   NNN  NNNNNN
EEE          PPP                 SSS    OOO      OOO   NNN   NNNNN
EEE          PPP           SSS     SSS   OOO      OOO   NNN    NNNN
EEEEEEEEEE   PPP           SSSS   SSS     OOO    OOO    NNN     NNN
EEEEEEEEEE   PPP             SSSSSSS       OOOOOOOO      NNN      NN

             (C) COPYRIGHT 1991 SEIKO EPSON CORP.

          SOURCE FILE NAME IS " C007XXX.DAT "

          THIS SOFTWARE MAKES NEXT FILES.

             C007XXXH.HEX  ...  HIGH BYTE OBJECT FILE.
             C007XXXL.HEX  ...  LOW BYTE OBJECT FILE.
             C007XXX .PRN  ...  ASSEMBLY LIST FILE.

DO YOU NEED AUTO PAGE SET? (Y/N) Y                              ...(1)

DO YOU NEED CROSS REFERENCE TABLE? (Y/N) Y                      ...(2)
```

When ASM6007 is started, the start-up message is displayed.
At (1), select whether or not the auto-page-set function will be used.

Use ..... Y↵
Not use ..... N↵

If the assembly listing file output is specified, message (2) is displayed. At this stage, cross-reference table generation may be selected.

Generating ..... Y↵
Not generating ..... N↵

When the above operation is completed, ASM6007 assembles the source file.
To suspend execution, press the "CTRL" and "C" keys together at stage (1) or (2).

■ Operators

Arithmetic operators		Logical operators	
+a	Monadic positive	a_AND_b	Logical product
-a	Monadic negative	a_OR_b	Logical sum
a+b	Addition	a_XOR_b	Exclusive logical sum
a-b	Subtraction	NOT_a	Logical negation
a*b	Multiplication	**Relational operators**	
a/b	Division	a_EQ_b	True when a is equal to b
a_MOD_b	Remainder of a/b	a_NE_b	True when a is not equal to b
a_SHL_b	Shifts a b bits to the left	a_LT_b	True when a is less than b
a_SHR_b	Shifts a b bits to the right	a_LE_b	True when a is less than or equal to b
HIGH_a	Separates the high-order eight bits from a	a_GT_b	True when a is greater than b
LOW_a	Separates the low-order eight bits from a	a_GE_b	True when a is greater than or equal to b

■ Pseudo-instructions

Pseudo-instruction	Meaning	Example of use
EQU (Equation)	To allocate data to label	`ABC EQU 9` `BCD EQU ABC+1`
SET (Set)	To allocate data to label (data can be changed)	`ABC SET 0001H` `ABC SET 0002H`
DW (Define Word)	To define ROM data	`ABC DW 'AB'` `BCD DW 0FFBH`
ORG (Origin)	To define location counter	`ORG 100H` `ORG 256`
PAGE (Page)	To define boundary of page	`PAGE 1H` `PAGE 3`
SECTION (Section)	To define boundary of section	`SECTION`
END (End)	To terminate assembly	`END`

MACRO (Macro)	To define macro	`CHECK MACRO DATA`
		`LOCAL LOOP`
		`LOOP CP MX,DATA`
		`JP NZ,LOOP`
		`ENDM`
		
		`CHECK 1`

LOCAL (Local)	To make local specification of label during macro definition	
ENDM (End Macro)	To end macro definition	

■ Error messages

Error message	Explanation
S (Syntax Error)	An unrecoverable syntax error was encountered.
U (Undefined Error)	The label or symbol of the operand has not been defined.
M (Missing Label)	The label field has been omitted.
O (Operand Error)	A syntax error was encountered in the operand, or the operand could not be evaluated.
P (Phase Error)	The same label or symbol was defined more than once.
R (Range Error)	• A statement exceeded a page boundary although its location was not specified. • The location counter value exceeded the upper limit of the program memory, or a location exceeding the upper limit was specified. • A value greater than that which the number of significant digits of the operand will accommodate was specified.
! (Warning)	• Memory areas overlapped because of a "PAGE" or "ORG" pseudo-instruction or both.
FILE NAME ERROR	The source file name was longer than 8 characters.
FILE NOT PRESENT	The specified source file was not found.
DIRECTORY FULL	No space was left in the directory of the specified disk.
FATAL DISK WRITE ERROR	The file could not be written to the disk.
LABEL TABLE OVERFLOW	The number of defined labels and symbols exceeded the label table capacity (4000).
CROSS REFERENCE TABLE OVERFLOW	The label/symbol reference count exceeded the cross-reference table capacity (only when the cross-reference table is generated).

3 FUNCTION OPTION GENERATOR FOG6007

3.1 FOG6007 Outline

With the 4-bit single-chip E0C6007 microcomputers, the customer may select the hardware options. By modifying the mask patterns of the E0C6007 according to the selected options, the system can be customized to meet the specifications of the target system.
The Function Option Generator FOG6007 is a software tool for generating data files used to generate mask patterns. It enables the customer to interactively select and specify pertinent items for each hardware option. From the data file created with FOG6007, the E0C6007 mask pattern is automatically generated by a general purpose computer.
The HEX file for the evaluation board (EVA6007) hardware option ROM is simultaneously generated with the data file.

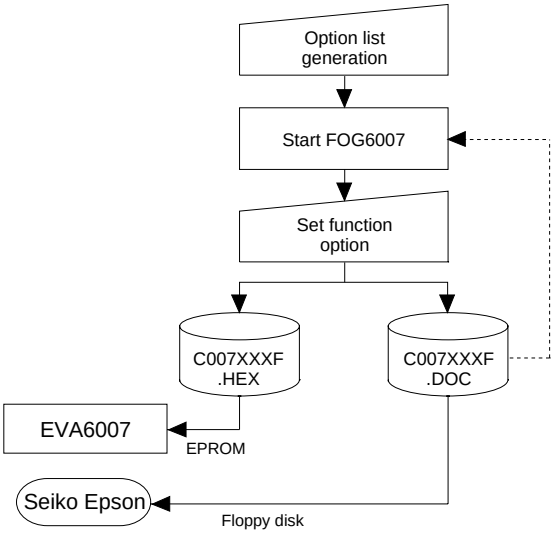

Fig. 3.1.1 FOG6007 execution flow

☞ *The operating method is same as for the E0C62 Family. Refer to the "E0C62 Family Development Tool Reference Manual" for details.*

3.2 E0C6007 Option List

Multiple specifications are available in each option item as indicated in the Option List. Using "3.3 Option Specifications and Selection Message" as reference, select the specifications that meet the target system. Be sure to record the specifications for unused ports too, according to the instructions provided.

1. OSC1 OSCILLATOR

☐ 1. Crystal ☐ 2. CR

2. OSC3 OSCILLATOR

☐ 1. CR ☐ 2. Ceramic

3. MULTIPLE KEY ENTRY RESET

☐ 1. Not Use
☐ 2. Use K00, K01
☐ 3. Use K00, K01, K02
☐ 4. Use K00, K01, K02, K03

4. INPUT PORTS PULL UP RESISTOR

- K00 ..... ☐ 1. With Resistor ☐ 2. Gate Direct
- K01 ..... ☐ 1. With Resistor ☐ 2. Gate Direct
- K02 ..... ☐ 1. With Resistor ☐ 2. Gate Direct
- K03 ..... ☐ 1. With Resistor ☐ 2. Gate Direct

5. OUTPUT PORT OUTPUT SPECIFICATION

- R32 ..... ☐ 1. Complementary ☐ 2. Nch-OpenDrain

6. OUTPUT PORT OUTPUT SPECIFICATION

- R33 ..... ☐ 1. Complementary ☐ 2. Nch-OpenDrain

7. R40 OUTPUT PORT SPECIFICATION

- OUTPUT SPECIFICATION ..... ☐ 1. Complementary ☐ 2. Nch-OpenDrain
- OUTPUT TYPE ..... ☐ 1. DC Output
  ☐ 2. /FOUT

8. R41 OUTPUT PORT SPECIFICATION

- OUTPUT SPECIFICATION ..... ☐ 1. Complementary ☐ 2. Nch-OpenDrain

9. R42 OUTPUT PORT SPECIFICATION

- OUTPUT SPECIFICATION ..... ☐ 1. Complementary ☐ 2. Nch-OpenDrain
- OUTPUT TYPE ..... ☐ 1. DC Output
  ☐ 2. FOUT
  ☐ 3. Buzzer Inverted Output (R43 Control)

10. R43 OUTPUT PORT SPECIFICATION

- OUTPUT SPECIFICATION ..... ☐ 1. Complementary ☐ 2. Nch-OpenDrain
- OUTPUT TYPE ..... ☐ 1. DC Output ☐ 2. Buzzer Output

11. FOUT FREQUENCY

☐ 1. Don't Use
☐ 2. FOUT (OSC3)
☐ 3. FOUT (32k)
☐ 4. FOUT (16k)
☐ 5. FOUT (8k)
☐ 6. FOUT (4k)
☐ 7. FOUT (2k)
☐ 8. FOUT (1k)
☐ 9. FOUT (512)
☐ 10. FOUT (256)

12. I/O PORTS OUTPUT SPECIFICATION

- P00–P03 ..... ☐ 1. Complementary ☐ 2. Nch-OpenDrain

13. LCD POWER VOLTAGE MODE

☐ 1. Internal Voltage ☐ 2. External Voltage

3.3 E0C6007 Option Specifications and Selection Message

Screen that can be selected as function options set on the E0C6007 are shown below, and their specifications are also described.

1 OSC1 oscillator

```
*** OPTION  NO.1 ***
--- OSC1 OSCILLATOR ---
                          1. CRYSTAL
                          2. CR
PLEASE SELECT NO.(1) ? 1↵
                          1. CRYSTAL  SELECTED
```

Select oscillation circuit that uses OSC1 and OSC2.
To minimize external components, CR oscillation circuit would be suitable; to obtain a stable oscillation frequency, crystal oscillation circuit would be suitable. When CR oscillation circuit is selected, only resistor is needed as external components since capacities are built-in.
On the other hand, when crystal oscillation circuit is selected, crystal oscillator, trimmer capacity and Rfx (feedback resistor) are needed as external components. Although when crystal oscillation circuit is selected, it is fixed at 32.768 kHz, when CR oscillation circuit is selected, frequency may be modified to a certain extent depending on the resistance of external components.

2 OSC3 oscillator

```
*** OPTION  NO.2 ***
--- OSC3 OSCILLATOR ---
                          1. CR
                          2. CERAMIC
PLEASE SELECT NO.(1) ? 1↵
                          1. CR  SELECTED
```

Select oscillation circuit that uses OSC3 and OSC4.
To minimize external components, CR oscillation circuit would be suitable; to obtain a stable oscillation frequency, ceramic oscillation circuit would be suitable. When CR oscillation circuit is selected, only resistor is needed as external components since capacities are built-in.
On the other hand, when ceramic oscillation circuit is selected, ceramic oscillator, feedback resistor, gate capacity and drain capacity are needed as external components.
Select CR oscillation circuit for unused OSC3 system clock.

3 Multiple key entry reset

```
*** OPTION  NO.3 ***
--- MULTIPLE KEY ENTRY RESET ---
                          1. NOT USE
                          2. USE K00,K01
                          3. USE K00,K01,K02
                          4. USE K00,K01,K02,K03
PLEASE SELECT NO.(1) ? 2↵
                          2. USE K00,K01  SELECTED
```

The reset function is set when K00 through K03 are entered.
When "Not Use" is selected, the reset function is not activated even if K00 through K03 are entered.
When "Use K00, K01" is selected, the system is reset immediately the K00 and K01 inputs go low at the same time. Similarly, the system is reset as soon as the K00 through K02 inputs or the K00 through K03 inputs go low.
The system is reset when a low signal is input for more than 2 seconds.
The system reset circuit is shown in Figure 3.3.1.

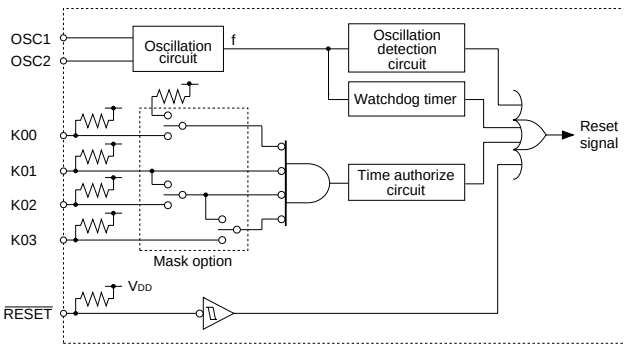

Fig. 3.3.1 System reset circuit

4 Input ports pull up resistor

```
*** OPTION  NO.4 ***

--- INPUT PORTS PULL UP RESISTOR ---

        K00             1. WITH RESISTOR
                        2. GATE DIRECT

PLEASE SELECT NO.(1) ? 2↵

        K01             1. WITH RESISTOR
                        2. GATE DIRECT

PLEASE SELECT NO.(1) ? 2↵

        K02             1. WITH RESISTOR
                        2. GATE DIRECT

PLEASE SELECT NO.(1) ? 2↵

        K03             1. WITH RESISTOR
                        2. GATE DIRECT

PLEASE SELECT NO.(1) ? 2↵

        K00             2. GATE DIRECT  SELECTED
        K01             2. GATE DIRECT  SELECTED
        K02             2. GATE DIRECT  SELECTED
        K03             2. GATE DIRECT  SELECTED
```

Select whether input ports (K00–K03) will each be supplemented with pull up resistors or not. When "Gate Direct" is selected, see to it that entry floating state does not occur. Select "With Resistor" pull up resistor for unused ports.
Moreover, the input port status is changed from low level (VSS) to high (VDD) with pull up resistors, a delay of approximately 500 µsec in waveform rise time will occur depending on the pull up resistor and entry load time constant. Because of this, when input reading is to be conducted, ensure the appropriate wait time with the program.
The configuration of the pull up resistor circuit is shown in Figure 3.3.2.

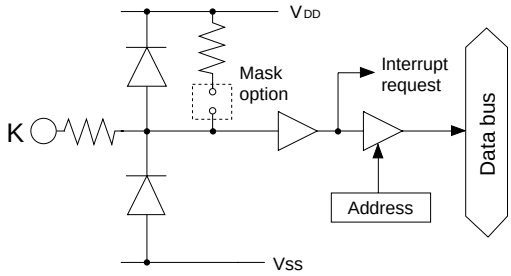

Fig. 3.3.2
Configuration of pull up resistor circuit

5 Output port output specification

```
*** OPTION  NO.5 ***

--- OUTPUT PORT OUTPUT SPECIFICATION ---

        R32             1. COMPLEMENTARY
                        2. NCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

        R32             1. COMPLEMENTARY  SELECTED
```

Select the output specification for the output port (R32).
Either complementary output or Nch open drain output may be selected.
When output port is to be used on key matrix configuration, select Nch open drain output.
For unused output ports, select complementary output.
The output circuit configuration is shown in Figure 3.3.3.

6 Output port output specification

```
*** OPTION  NO.6 ***

--- OUTPUT PORT OUTPUT SPECIFICATION ---

        R33             1. COMPLEMENTARY
                        2. NCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

        R33             1. COMPLEMENTARY  SELECTED
```

Select the output specification for the output port (R33).
Either complementary output or Nch open drain output may be selected.
When output port is to be used on key matrix configuration, select Nch open drain output.
For unused output ports, select complementary output.
The output circuit configuration is shown in Figure 3.3.3.

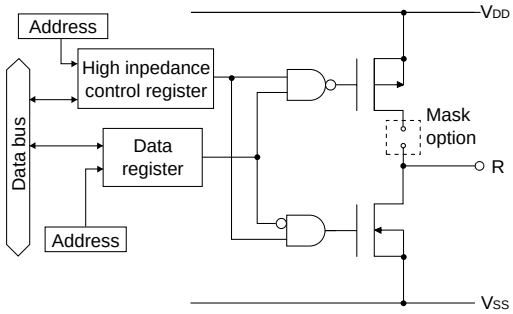

Fig. 3.3.3
Configuration of output circuit

7 R40 output port specification

```
*** OPTION  NO.7 ***

--- R40 OUTPUT PORT SPECIFICATION ---

    OUTPUT SPECIFICATION 1. COMPLEMENTARY
                         2. NCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

    OUTPUT TYPE          1. DC
                         2. /FOUT

PLEASE SELECT NO.(1) ? 1↵

    OUTPUT SPECIFICATION 1. COMPLEMENTARY  SELECTED
    OUTPUT TYPE          1. DC  SELECTED
```

Select the specification for R40 terminal.
Either complementary output or Nch open drain output may be selected for the output specification.
For the output type, either DC output or /FOUT output may be selected.
The output circuit configuration of R4 port is shown in Figure 3.3.4.

When DC output is selected

When DC output is selected, R40 becomes a regular output port. By writing "1" on R40 register, the R40 terminal output goes high (V_{DD}), and goes low (V_{SS}) by writing "0".
The output waveform is shown in Figure 3.3.5.

When /FOUT output is selected

When /FOUT output is selected, signal with frequency selected from among nine types, ranging from 256 Hz to 32768 Hz and OSC3 may be output from R40 terminal.
In this case, by writing "0" on R40 register, 50% duty and V_{DD}–V_{SS} amplitude square wave is output from R40 terminal at the specified frequency. Writing "1" will cause the R40 terminal to go high (V_{DD}).
The /FOUT output is normally used to supply clock to other devices but since hazard occurs when R40 register is re-written, great caution must be observed when using it.
Moreover, FOUT output may be selected in the same manner through the R42 terminal mask option. Note, however, that FOUT signal becomes antiphase to /FOUT signal.
The output waveform is shown in Figure 3.3.6.

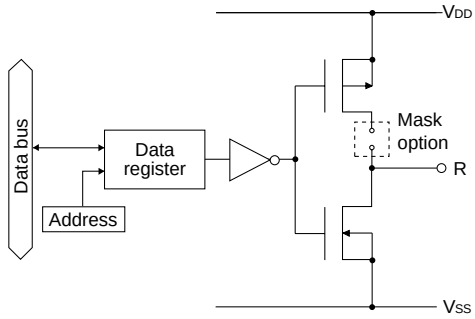

Fig. 3.3.4 Output circuit configuration of R4 port

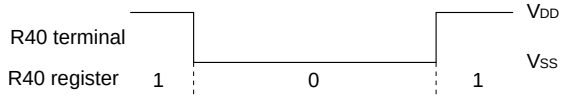

Fig. 3.3.5 Output waveform at DC output selection

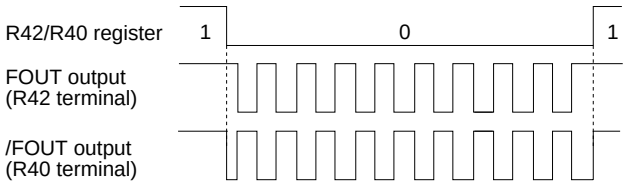

Fig. 3.3.6 FOUT signal and /FOUT signal output waveforms

8 R41 output port specification

```
*** OPTION  NO.8 ***

--- R41 OUTPUT PORT SPECIFICATION ---

    OUTPUT SPECIFICATION 1. COMPLEMENTARY
                         2. NCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

    OUTPUT SPECIFICATION 1. COMPLEMENTARY  SELECTED
```

Select the specification for R41 terminal.
Either complementary output or Nch open drain output may be selected for the output specification.
The circuit configuration is the same as that of the R40 output port (Figure 3.3.4).

9 R42 output port specification

```
*** OPTION  NO.9 ***

--- R42 OUTPUT PORT SPECIFICATION ---

    OUTPUT SPECIFICATION 1. COMPLEMENTARY
                         2. NCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

    OUTPUT TYPE          1. DC
                         2. FOUT
                         3. /BUZZER (R43 CONT)

PLEASE SELECT NO.(1) ? 2↵

    OUTPUT SPECIFICATION 1. COMPLEMENTARY  SELECTED
    OUTPUT TYPE          2. FOUT  SELECTED
```

Select the specification for R42 terminal.
Either complementary output or Nch open drain output may be selected for the output specification.
Any one of the following may be selected for the output type: DC output, /BUZZER output or /FOUT output.
The circuit configuration is the same as that of the R40 output port (Figure 3.3.4).

When DC output is selected, R42 becomes a regular output port. By writing "1" on R42 register, the R42 terminal output goes high (V_{DD}), and goes low (V_{SS}) by writing "0".

When BUZZER output is selected with the R43 mask option, /BUZZER output becomes a buzzer inverted signal for direct driving of the piezoelectric buzzer together with the BUZZER output.
The control of the buzzer signal output is accomplished through R43 register even if R42 has been set to /BUZZER.
Because of this, R42 cannot be set to /BUZZER output when R43 is set to DC output.
Refer to Figure 3.3.7 for the /BUZZER output waveform.

When FOUT output is selected, signal with frequency selected from among nine types, ranging from 256 Hz to 32768 Hz and OSC3 may be output from R42 terminal.
FOUT output is the same as that of /FOUT signal. Note, however, that /FOUT signal becomes antiphase to FOUT signal.

10 R43 output port specification

```
*** OPTION  NO.10 ***

--- R43 OUTPUT PORT SPECIFICATION ---

    OUTPUT SPECIFICATION 1. COMPLEMENTARY
                         2. NCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

    OUTPUT TYPE          1. DC
                         2. BUZZER

PLEASE SELECT NO.(1) ? 2↵

    OUTPUT SPECIFICATION 1. COMPLEMENTARY  SELECTED
    OUTPUT TYPE          2. BUZZER  SELECTED
```

* When "R42 OUTPUT TYPE" is set to /BUZZER", "DC" option may not be selected.

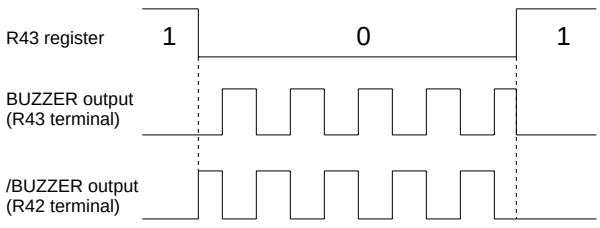

Fig. 3.3.7 Buzzer signal output waveform

Select the specification for R43 terminal.
Either complementary output or Nch open drain output may be selected for the output specification.
For the output type, either DC output or BUZZER output may be selected.
The circuit configuration is the same as that of the R40 output port (Figure 3.3.4).

When DC output is selected, R43 becomes a regular output port. By writing "1" on R43 register, the R43 terminal output goes high (V_{DD}), and goes low (V_{SS}) by writing "0".

When BUZZER output is selected, by writing "0" on R43 register, buzzer signal is output from the R43 terminal, low (V_{SS}) is output by writing "1".
When /BUZZER output (buzzer inverted output) is selected for R42 together with R43, it can be directly driven together with piezoelectric buzzer.
The control of the /BUZZER output is also accomplished through R43 register.
The BUZZER signal and /BUZZER signal output waveforms are shown in Figure 3.3.7.

11 FOUT frequency

```
*** OPTION  NO.11 ***

--- FOUT FREQUENCY ---

                  1. DON'T USE
                  2. FOUT (OSC3)
                  3. FOUT (32K)
                  4. FOUT (16K)
                  5. FOUT ( 8K)
                  6. FOUT ( 4K)
                  7. FOUT ( 2K)
                  8. FOUT ( 1K)
                  9. FOUT (512)
                  10.FOUT (256)

PLEASE SELECT NO.(1) ? 1↵

                  1. DON'T USE  SELECTED
```

When FOUT output is selected, signal with frequency selected from among nine types, ranging from 256 Hz to 32768 Hz and OSC3 may be output.

12 I/O ports output specification

```
*** OPTION  NO.12 ***

--- P0X OUTPUT SPECIFICATION ---

    P00-P03       1. COMPLEMENTARY
                  2. NCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

    P00-P03       1. COMPLEMENTARY  SELECTED
```

Select the output specification to be used during I/O ports (P00–P03) output mode selection.
Either complementary output or Nch open drain output may be selected.
The circuit configuration of the output driver is the same as that of output ports (Figure 3.3.3).
Select complementary output for unused ports.

The I/O port circuit configuration is shown in Figure 3.3.8.

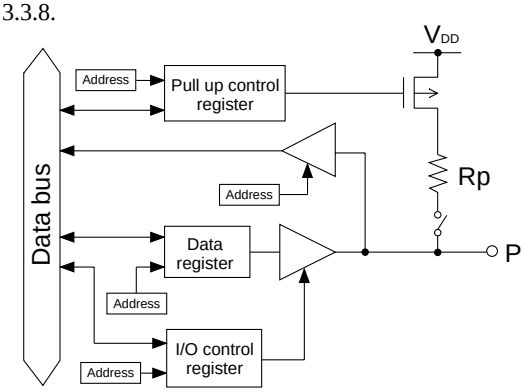

Fig. 3.3.8 Circuit configuration of I/O port

13 LCD power voltage mode

```
*** OPTION  NO.13 ***

--- LCD POWER VOLTAGE MODE ---

                  1. INTERNAL VOLTAGE
                  2. EXTERNAL VOLTAGE

PLEASE SELECT NO.(1) ? 1↵

                  1. INTERNAL VOLTAGE  SELECTED
```

Select the power mode for the LCD driver.
When internal voltage is selected, the LCD power source built in the E0C6007 is used and driving is fixed at 1/4 bias. In this case, the LCD contrast may be adjusted through the software.
When external voltage is selected, the LCD driving voltage is externally supplied. Aside from 1/4 bias, driving may also be set at 1/5 bias. Adjustment of the LCD contrast in this case requires setting up of external components.

3.4 FOG6007 Quick Reference

■ Starting command and input/output files

Execution file: FOG6007.EXE

↵ *indicates the Return key.*

Starting command: **FOG6007↵**

Input file: C007XXXF.DOC (Function option document file, when modifying)

Output file: C007XXXF.DOC (Function option document file)
C007XXXF.HEX (Function option HEX file)

■ Display example

```
     ***  E0C6007 FUNCTION OPTION GENERATOR. --- Ver 1.11  ***

EEEEEEEEEE   PPPPPPPP         SSSSSSS       OOOOOOOO     NNN      NNN
EEEEEEEEEE   PPPPPPPPPP     SSS    SSSS    OOO    OOO    NNNN     NNN
EEE          PPP     PPP   SSS     SSS   OOO      OOO   NNNNN    NNN
EEE          PPP     PPP    SSS          OOO      OOO   NNNNNN   NNN
EEEEEEEEEE   PPPPPPPPPP      SSSSSS      OOO      OOO   NNN NNN  NNN
EEEEEEEEEE   PPPPPPPP           SSSS     OOO      OOO   NNN  NNNNNN
EEE          PPP                SSS     OOO      OOO   NNN   NNNNN
EEE          PPP            SSS     SSS   OOO      OOO   NNN    NNNN
EEEEEEEEEE   PPP            SSSS    SSS    OOO    OOO    NNN     NNN
EEEEEEEEEE   PPP              SSSSSSS       OOOOOOOO     NNN      NN

              (C) COPYRIGHT 1991  SEIKO EPSON CORP.

         THIS SOFTWARE MAKES NEXT FILES.

            C007XXXF.HEX  ...  FUNCTION OPTION HEX FILE.
            C007XXXF.DOC  ...  FUNCTION OPTION DOCUMENT FILE.

                         STRIKE ANY KEY.
```

Start-up message
When FOG6007 is started, the start-up message is displayed.
For "STRIKE ANY KEY.", press any key to advance the program execution.

To suspend execution, press the "CTRL" and "C" keys together: the sequence returns to the DOS command level.

```
*** E0C6007 USER'S OPTION SETTING. --- Ver 1.11 ***

CURRENT DATE IS  92/02/14
PLEASE INPUT NEW DATE  :   92/02/17↵
```

Date input
Enter the 2-digit year, month, and day of the month by delimiting them with a slash ("/"). When not modifying the date, press the RETURN key "↵" to continue.

```
*** OPERATION SELECT MENU ***

       1. INPUT NEW FILE
       2. EDIT FILE
       3. RETURN TO DOS

PLEASE SELECT NO.?
```

Operation selection menu
Enter a number from 1 to 3 to select a subsequent operation.
1. To set new function options.
2. To modify the document file.
3. To terminate FOG6007.

```
*** OPERATION SELECT MENU ***

       1. INPUT NEW FILE
       2. EDIT FILE
       3. RETURN TO DOS

PLEASE SELECT NO.? 1↵
PLEASE INPUT FILE NAME? C0070A0↵                                    ..(1)
PLEASE INPUT USER'S NAME? SEIKO EPSON CORP.↵                        ..(2)
PLEASE INPUT ANY COMMENT
(ONE LINE IS 50 CHR)? TOKYO DESIGN CENTER↵                          ..(3)
                    ? 421-8 HINO HINO-SHI TOKYO 191-8501 JAPAN↵
                    ? TEL 042-587-5816↵
                    ? FAX 042-587-5624↵
                    ? ↵
```

Setting new function options
Select "1" on the operation selection menu.
(1) Enter the file name.
(2) Enter the customer's company name.
(3) Enter any comment.
(Within 50 characters x 10 lines)
Next, start function option setting from option No. 1.

```
PLEASE INPUT FILE NAME? C0070A0↵
EXISTS OVERWRITE(Y/N)? N↵
PLEASE INPUT FILE NAME? C0070B0↵
PLEASE INPUT USER'S NAME?
```

In case a function option document file with the same name as the file name specified in the current drive exists, the user is asked whether overwrition is desired. Enter "Y" or "N" accordingly.

```
*** OPERATION SELECT MENU ***

        1. INPUT NEW FILE
        2. EDIT FILE
        3. RETURN TO DOS

PLEASE SELECT NO.? 2↵

*** SOURCE FILE(S) ***

C0070A0          C0070B0          C0070C0                  ..(1)

PLEASE INPUT FILE NAME? C0070A0↵                          ..(2)
PLEASE INPUT USER'S NAME?↵                                ..(3)
PLEASE INPUT ANY COMMENT
(ONE LINE IS 50 CHR)?↵                                    ..(4)
PLEASE INPUT EDIT NO.? 4↵                                 ..(5)
            :
(Modifying function option settings)
            :
PLEASE INPUT EDIT NO.? E↵
```

In step (1), if no modifiable source exists, the following message is displayed and the sequence returns to the operation selection menu.

```
*** SOURCE FILE(S) ***

FUNCTION OPTION DOCUMENT FILE IS NOT FOUND.
```

In step (2), if the function option document file is not in the current drive, the following message is displayed, prompting entry of other file name.

```
PLEASE INPUT FILE NAME? C0070N0↵
FUNCTION OPTION DOCUMENT FILE IS NOT FOUND.
PLEASE INPUT FILE NAME?
```

```
BAD FUNCTION OPTION DOCUMENT FILE
```

```
*** OPTION  NO.3 ***

— MULTIPLE KEY ENTRY RESET —

                      1. Not Use
                      2. Use  K00,K01
                      3. Use  K00,K01,K02
                      4. Use  K00,K01,K02,K03

PLEASE SELECT  NO.(1) ? 2↵

                      2. Use  K00,K01  SELECTED
```

```
END OF OPTION SETTING.
DO YOU MAKE HEX FILE (Y/N) ? Y↵                           ..(1)

*** OPTION EPROM SELECT MENU ***

        1. 27C64
        2. 27C128
        3. 27C256
        4. 27C512

PLEASE SELECT NO.? 2↵                                     ..(2)

        2. 27C128   SELECTED

MAKING FILE(S) IS COMPLETED.

*** OPERATION SELECT MENU ***

        1. INPUT NEW FILE
        2. EDIT FILE
        3. RETURN TO DOS

PLEASE SELECT NO.?
```

Modifying function option settings

Select "2" on the operation selection menu.

(1) Will display the files on the current drive.
(2) Enter the file name.
(3) Enter the customer's company name.
(4) Enter any comment.
    Previously entered data can be used by pressing the RETURN key "↵" at (3) and (4).
(5) Enter the number of the function option to be modified. When selection of one option is complete, the system prompts entry of another function option number. Repeat selection until all options to be modified are selected.
    Enter "E↵" to end option setting. Then, move to the confirmation procedure for HEX file generation.

In cace of the file format specified by step (2) is not corrent (such as document file of other model), the message will be displayed and the FOG6007 program will be terminated.

Option selection

The selections for each option correspond one to one to the option list. Enter the selection number. The value in parentheses ( ) indicates the default value, and is set when only the RETURN key "↵" is pressed.
In return, the confirmation is displayed.
When you wish to modify previously set function options in the new setting process, enter "B↵" to return 1 step back to the previous function option setting operation.

EPROM selection

When setting function options setting is completed, the following message is output to ask the operator whether to generate the HEX file.

(1) When debugging the program with EVA6007, HEX file is needed, so enter "Y↵". If "N↵" is entered, no HEX file is generated and only document file is generated.
(2) For the option ROM selection menu displayed when "Y↵" is entered in Step (1), select the EPROM to be used for setting EVA6007 options.

When a series of operations are complete, the sequence returns to the operation selection menu.

3.5 Sample File

■ Example of function option document file

```
* E0C6007 FUNCTION OPTION DOCUMENT  V 1.11
*
* FILE NAME    TESTF.DOC
* USER'S NAME
* INPUT DATE   1998/07/03
*
*
*
* OPTION NO.1
* < OSC1 OSCILLATOR >
*                           CRYSTAL  ---------------------  SELECTED
 OPT0101 01
 OPT0102 01
*
* OPTION NO.2
* < OSC3 OSCILLATOR >
*                           CR  --------------------------  SELECTED
 OPT0201 01
*
* OPTION NO.3
* < MULTIPLE KEY ENTRY RESET >
*                           NOT USE  ---------------------  SELECTED
 OPT0301 01
*
* OPTION NO.4
* < INPUT PORTS PULL UP RESISTOR >
*     K00                   WITH RESISTOR  ---------------  SELECTED
*     K01                   WITH RESISTOR  ---------------  SELECTED
*     K02                   WITH RESISTOR  ---------------  SELECTED
*     K03                   WITH RESISTOR  ---------------  SELECTED
 OPT0401 01
 OPT0402 01
 OPT0403 01
 OPT0404 01
*
* OPTION NO.5
* < R32 OUTPUT PORT OUTPUT SPECIFICATION >
*     R32                   COMPLEMENTARY  ---------------  SELECTED
 OPT0501 01
*
* OPTION NO.6
* < R33 OUTPUT PORT OUTPUT SPECIFICATION >
*     R33                   COMPLEMENTARY  ---------------  SELECTED
 OPT0601 01
*
* OPTION NO.7
* < R40 OUTPUT PORT SPECIFICATION >
*     OUTPUT SPECIFICATION  COMPLEMENTARY  ---------------  SELECTED
*     OUTPUT TYPE           DC  --------------------------  SELECTED
 OPT0701 01
 OPT0702 01
*
* OPTION NO.8
* < R41 OUTPUT PORT SPECIFICATION >
*                           COMPLEMENTARY  ---------------  SELECTED
 OPT0801 01
*
* OPTION NO.9
* < R42 OUTPUT PORT SPECIFICATION >
*     OUTPUT SPECIFICATION  COMPLEMENTARY  ---------------  SELECTED
```

```
*     OUTPUT TYPE            DC  -------------------------  SELECTED
 OPT0901 01
 OPT0902 01
*
* OPTION NO.10
* < R43 OUTPUT PORT SPECIFICATION >
*     OUTPUT SPECIFICATION   COMPLEMENTARY  --------------  SELECTED
*     OUTPUT TYPE            DC  -------------------------  SELECTED
 OPT1001 01
 OPT1002 01
*
* OPTION NO.11
* < FOUT FREQUENCY >
*                            DON'T USE  ------------------  SELECTED
 OPT1101 01
*
* OPTION NO.12
* < P0X OUTPUT SPECIFICATION >
*                            COMPLEMENTARY  --------------  SELECTED
 OPT1201 01
*
* OPTION NO.13
* < LCD POWER VOLTAGE MODE >
*                            INTERNAL VOLTAGE  ------------  SELECTED
 OPT1301 01
*
*
*
* SEIKO EPSON'S AREA
*
*
*
* OPTION NO.14
 OPT1401 01
*
* OPTION NO.15
 OPT1501 02
*
* OPTION NO.16
 OPT1601 01
\\END
```

Note End mark "¥¥END" may be used instead of "\\END" depending on the PC used. (The code of \ and ¥ is 5CH.)

4 ICE CONTROL SOFTWARE ICS6007

4.1 ICS6007 Outline

The In-circuit Emulator ICE6200 connects the target board produced by the user via the EVA6007 and performs real time target system evaluation and debugging by passing through the RS-232C from the host computer and controlling it. The operation on the host computer side and ICE6200 control is done through the ICE Control Software ICS6007.
The ICS6007 has a set of numerous and highly functional emulation commands which provide sophisticated break function, on-the-fly data display, history display, etc., and so perform a higher level of debugging.

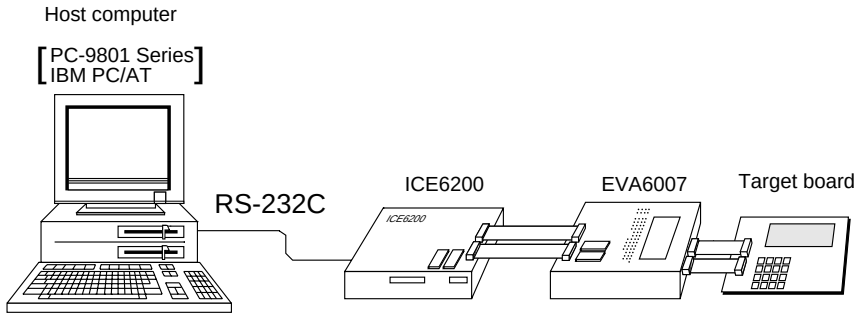

Fig. 4.1.1 Debugging system using ICE6200

☞ *The functions of the ICE6200 and commands are same as for the E0C62 Family. Refer to the "E0C62 Family Development Tool Reference Manual" for details.*

4.2 ICS6007 Restrictions

Take the following precautions when using the ICS6007.

■ ROM Area

The ROM area is limited to a maximum address of 0FFFH. Assigning data above the 0FFFH address causes an error.

■ RAM Area

The RAM area is limited to a maximum address of F7FH. However, as the following addresses are in the unused area, designation of this area with the ICE commands produces an error.

Unused area: 200H–DFFH, E50H–E7FH, ED0H–EFFH, F03H, F05H–F0FH, F13H, F15H–F1FH, F28H–F3FH, F41H–F52H, F55H–F5FH, F61H–F6FH, F73H, F7AH, F7CH

(Refer to the "E0C6007 Technical Manual" for details.)

■ Undefined Code

The SLP instruction is not specified for the E0C6007 and so cannot be used.

■ OPTLD Command

In the ICS6007, OPTLD command cannot be used.

4.3 ICS6007 Quick Reference

■ Starting command and input/output files

Execution file:	ICS6007.BAT (ICS6007J.EXE)	. . . for MS-DOS
	ICS6007B.BAT (ICS6007W.EXE)	. . . for PC-DOS
Starting command:	**ICS6007 (ICS6007J)↵**	. . . for MS-DOS
	ICS6007B (ICS6007W)↵	. . . for PC-DOS
Input file:	C007XXXL.HEX (Object file, low-order)	
	C007XXXH.HEX (Object file, high-order)	
	C007XXXD.HEX (Data RAM file)	
	C007XXXC.HEX (Control file)	
Output file:	C007XXXL.HEX (Object file, low-order)	
	C007XXXH.HEX (Object file, high-order)	
	C007XXXD.HEX (Data RAM file)	
	C007XXXC.HEX (Control file)	

↵ *indicates the Return key.*

■ Display example

```
          *** E0C6007 ICE CONTROL SOFTWARE. --- Ver 3.01 ***

EEEEEEEEEE   PPPPPPPP        SSSSSSS       OOOOOOOO      NNN      NNN
EEEEEEEEEE   PPPPPPPPPP     SSS    SSSS    OOO    OOO    NNNN     NNN
EEE          PPP     PPP   SSS      SSS   OOO      OOO   NNNNN    NNN
EEE          PPP     PPP    SSS           OOO      OOO   NNNNNN   NNN
EEEEEEEEEE   PPPPPPPPPP      SSSSSS       OOO      OOO   NNN NNN NNN
EEEEEEEEEE   PPPPPPPP           SSSS     OOO      OOO   NNN  NNNNNN
EEE          PPP                SSS     OOO      OOO   NNN   NNNNN
EEE          PPP            SSS   SSS    OOO      OOO   NNN    NNNN
EEEEEEEEEE   PPP            SSSS   SSS    OOO    OOO    NNN     NNN
EEEEEEEEEE   PPP              SSSSSSS       OOOOOOOO     NNN      NN

             (C) COPYRIGHT 1991 SEIKO EPSON CORP.

* ICE POWER ON RESET *
* DIAGNOSTIC TEST OK *
#
```

Start-up message

When ICS6007 is started, the start-up message is displayed, and a self-test is automatically performed. ICS6007 commands are awaited when the program is properly loaded and the # mark is displayed.

Debugging can be done by entering command after the # mark.

The ICS6007 program is terminated by entering the Q (Quit) command.

Note *Confirm that the cables connected properly, then operate the ICS6007.*

■ Error messages

Error message	Meaning	Recover procedure
* COMMUNICATION ERROR OR ICE NOT READY *	ICE6200 is disconnected or power is OFF.	Switch OFF the host power supply, connect cable, and reapply power. Or switch ON power to ICE6200.
* TARGET DOWN (1) *	Evaluation board is disconnected. (Check at power ON)	Switch OFF power to ICE, and connect the evaluation board. Then, apply power to ICE6200.
* TARGET DOWN (2) *	Evaluation board is disconnected. (Check at command execution)	Switch OFF power to ICE, and connect the evaluation board. Then, apply power to ICE6200.
* UNDEFINED PROGRAM CODE EXIST *	Undefined code is detected in the program loaded from ROM or FD.	Convert ROM and FD data with the cross assembler, then restart the ICE6200.
* COMMAND ERROR *	A miss occurs by command input.	Reenter the proper command.
(No response after power on)	The ICE-to-HOST cable is disconnected on the host side.	Switch OFF the host power supply, connect cable, and reapply power.

■ ICE6200 commands

Item No.	Function	Command Format	Outline of Operation
1	Assemble	#A,a ↵	Assemble command mnemonic code and store at address "a"
2	Disassemble	#L,a1,a2 ↵	Contents of addresses a1 to a2 are disassembled and displayed
3	Dump	#DP,a1,a2 ↵	Contents of program area a1 to a2 are displayed
		#DD,a1,a2 ↵	Content of data area a1 to a2 are displayed
4	Fill	#FP,a1,a2,d ↵	Data d is set in addresses a1 to a2 (program area)
		#FD,a1,a2,d ↵	Data d is set in addresses a1 to a2 (data area)
5	Set Run Mode	#G,a ↵	Program is executed from the "a" address
		#TIM ↵	Execution time and step counter selection
		#OTF ↵	On-the-fly display selection
6	Trace	#T,a,n ↵	Executes program while displaying results of step instruction from "a" address
		#U,a,n ↵	Displays only the final step of #T,a,n
7	Break	#BA,a ↵	Sets Break at program address "a"
		#BAR,a ↵	Breakpoint is canceled
		#BD ↵	Break condition is set for data RAM
		#BDR ↵	Breakpoint is canceled
		#BR ↵	Break condition is set for EVA6007 CPU internal registers
		#BRR ↵	Breakpoint is canceled
		#BM ↵	Combined break conditions set for program data RAM address and registers
		#BMR ↵	Cancel combined break conditions for program data ROM address and registers
		#BRES ↵	All break conditions canceled
		#BC ↵	Break condition displayed
		#BE ↵	Enter break enable mode
		#BSYN ↵	Enter break disable mode
		#BT ↵	Set break stop/trace modes
		#BRKSEL,REM ↵	Set BA condition clear/remain modes
8	Move	#MP,a1,a2,a3 ↵	Contents of program area addresses a1 to a2 are moved to addresses a3 and after
		#MD,a1,a2,a3 ↵	Contents of data area addresses a1 to a2 are moved to addresses a3 and after
9	Data Set	#SP,a ↵	Data from program area address "a" are written to memory
		#SD,a ↵	Data from data area address "a" are written to memory
10	Change CPU Internal Registers	#DR ↵	Display EVA6007 CPU internal registers
		#SR ↵	Set EVA6007 CPU internal registers
		#I ↵	Reset EVA6007 CPU
		#DXY ↵	Display X, Y, MX and MY
		#SXY ↵	Set data for X and Y display and MX, MY

Item No.	Function	Command Format	Outline of Operation
11	History	#H,p1,p2 ↵	Display history data for pointer 1 and pointer 2
		#HB ↵	Display upstream history data
		#HG ↵	Display 21 line history data
		#HP ↵	Display history pointer
		#HPS,a ↵	Set history pointer
		#HC,S/C/E ↵	Sets up the history information acquisition before (S), before/after (C) and after (E)
		#HA,a1,a2 ↵	Sets up the history information acquisition from program area a1 to a2
		#HAR,a1,a2 ↵	Sets up the prohibition of the history information acquisition from program area a1 to a2
		#HAD ↵	Indicates history acquisition program area
		#HS,a ↵	Retrieves and indicates the history information which executed a program address "a"
		#HSW,a ↵ #HSR,a ↵	Retrieves and indicates the history information which wrote or read the data area address "a"
12	File	#RF,file ↵	Move program file to memory
		#RFD,file ↵	Move data file to memory
		#VF,file ↵	Compare program file and contents of memory
		#VFD,file ↵	Compare data file and contents of memory
		#WF,file ↵	Save contents of memory to program file
		#WFD,file ↵	Save contents of memory to data file
		#CL,file ↵	Load ICE6200 set condition from file
		#CS,file ↵	Save ICE6200 set condition to file
13	Coverage	#CVD ↵	Indicates coverage information
		#CVR ↵	Clears coverage information
14	ROM Access	#RP ↵	Move contents of ROM to program memory
		#VP ↵	Compare contents of ROM with contents of program memory
		#ROM ↵	Set ROM type
15	Terminate ICE	#Q ↵	Terminate ICE and return to operating system control
16	Command Display	#HELP ↵	Display ICE6200 instruction
17	Self Diagnosis	#CHK ↵	Report results of ICE6200 self diagnostic test

↵ means press the RETURN key.

5 MASK DATA CHECKER MDC6007

5.1 MDC6007 Outline

The Mask Data Checker MDC6007 is a software tool which checks the program data (C007XXXH.HEX and C007XXXL.HEX) and option data (C007XXXF.DOC) created by the user and creates the data file (C6007XXX.PAn) for generating mask patterns. The user must send the file generated through this software tool to Seiko Epson.

Moreover, MDC6007 has the capability to restore the generated data file (C6007XXX.PA0) to the original file format.

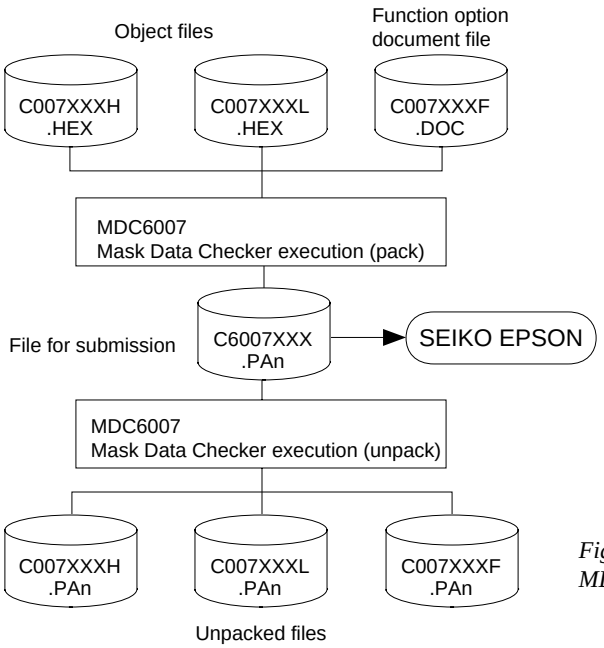

Fig. 5.1.1
MDC6007 execution flow

☞ *The operating method is same as for the E0C62 Family. Refer to the "E0C62 Family Development Tool Reference Manual" for details.*

5.2 MDC6007 Quick Reference

■ Starting command and input/output files

Execution file: MDC6007.EXE

Starting command: **MDC6007**↵ ↵ *indicates the Return key.*

Input file:	C007XXXL.HEX	(Object file, low-order)	When packing
	C007XXXH.HEX	(Object file, high-order)	
	C007XXXF.DOC	(Function option document file)	
	C6007XXX.PAn	(Packed file)	When unpacking
Output file:	C6007XXX.PAn	(Packed file)	When packing
	C007XXXL.PAn	(Object file, low-order)	When unpacking
	C007XXXH.PAn	(Object file, high-order)	
	C007XXXF.PAn	(Function option document file)	

■ Display examples

```
          *** E0C6007 PACK / UNPACK PROGRAM Ver 1.10 ***

EEEEEEEEEE   PPPPPPPP        SSSSSSS       OOOOOOOO      NNN      NNN
EEEEEEEEEE   PPPPPPPPPP     SSS   SSSS    OOO    OOO    NNNN     NNN
EEE          PPP     PPP   SSS     SSS   OOO      OOO   NNNNN    NNN
EEE          PPP     PPP    SSS          OOO      OOO   NNNNNN   NNN
EEEEEEEEEE   PPPPPPPPPP      SSSSSS      OOO      OOO   NNN NNN NNN
EEEEEEEEEE   PPPPPPPP           SSSS     OOO      OOO   NNN  NNNNNN
EEE          PPP                 SSS    OOO      OOO   NNN   NNNNN
EEE          PPP           SSS     SSS   OOO      OOO   NNN    NNNN
EEEEEEEEEE   PPP           SSSS   SSS     OOO    OOO    NNN     NNN
EEEEEEEEEE   PPP             SSSSSSS       OOOOOOOO      NNN      NN

             (C) COPYRIGHT 1991 SEIKO EPSON CORP.

                   --- OPERATION MENU ---

                          1. PACK
                          2. UNPACK

                   PLEASE SELECT NO.?
```

Start-up message

When MDC6007 is started, the start-up message and operation menu are displayed. Here, the user is prompted to select operation options.

```
                   --- OPERATION MENU ---

                          1. PACK
                          2. UNPACK

                   PLEASE SELECT NO.? 1↵                          ...(1)

       C007XXXH.HEX --------+
                            |
       C007XXXL.HEX --------+-------- C007XXX.PAn (PACK FILE)
                            |
       C007XXXF.DOC --------+

 PLEASE INPUT PACK FILE NAME (C6007XXX.PAn) ?  C60070A0.PA0↵   ...(2)

       C0070A0H.HEX --------+
                            |
       C0070A0L.HEX --------+-------- C0070A0.PA0
                            |
       C0070A0F.DOC --------+
```

Packing of data

(1) Select "1. PACK" in the operation menu.
(2) Enter the file name.
After submitting the data to Seiko Epson and there is a need to re-submit the data, increase the numeric value of "n" by one when the input is made.
(Example: When re-submitting data after "C6007XXX.PA0" has been submitted, the pack file name should be entered as "C6007XXX.PA1".)

With this, the mask file (C6007XXX.PAn) is generated, and the MDC6007 program will be terminated.
Submit this file to Seiko Epson.

Note Don't use the data generated with the -N option of the Cross Assembler (ASM6007) as program data. If the program data generated with the -N option of the Cross Assembler is packed, following message is displayed.

```
HEX DATA ERROR : DATA (NO FFh)
```

```
                   --- OPERATION MENU ---

                          1. PACK
                          2. UNPACK

                   PLEASE SELECT NO.? 2↵                          ...(1)

 PLEASE INPUT PACKED FILE NAME (C6007XXX.PAn) ?  C60070A0.PA0↵...(2)

                            +-------- C0070A0H.PA0
                            |
       C60070A0.PA0 --------+-------- C0070A0L.PA0
                            |
                            +-------- C0070A0F.PA0
```

Unpacking of data

(1) Select "2. UNPACK" in the operation menu.
(2) Enter the packed file name.

With this, the mask data file (C6007XXX.PAn) is restored to the original file format, and the MDC6007 program will be terminated.

Since the extension of the file name remains as "PAn", it must be renamed back to its original form ("HEX" and "DOC") in order to re-debug or modify the restored file.

■ Error messages

Program data error

Error Message	Explanation
1. HEX DATA ERROR : NOT COLON.	There is no colon.
2. HEX DATA ERROR : DATA LENGTH. (NOT 00-20h)	The data length of 1 line is not in the 00–20H range.
3. HEX DATA ERROR : ADDRESS.	The address is beyond the valid range of the program ROM.
4. HEX DATA ERROR : RECORD TYPE. (NOT 00)	The record type of 1 line is not 00.
5. HEX DATA ERROR : DATA. (NOT 00-FFh)	The data is not in the range between 00H and 0FFH.
6. HEX DATA ERROR : TOO MANY DATA IN ONE LINE.	There are too many data in 1 line.
7. HEX DATA ERROR : CHECK SUM.	The checksum is not correct.
8. HEX DATA ERROR : END MARK.	The end mark is not : 00000001FF.
9. HEX DATA ERROR : DUPLICATE.	There is duplicate definition of data in the same address.
10. HEX DATA ERROR : DATA (NO FFh)	There is an undefined field in the HEX data.

Function option data error

Error Message	Explanation
1. OPTION DATA ERROR : START MARK.	The start mark is not "\OPTION". (during unpacking) *
2. OPTION DATA ERROR : OPTION NUMBER.	The option number is not correct.
3. OPTION DATA ERROR : SELECT NUMBER.	The option selection number is not correct.
4. OPTION DATA ERROR : END MARK.	The end mark is not "\\END" (packing) or "\END" (unpacking).*

File error

Error Message	Explanation
1. <File_name> FILE IS NOT FOUND.	The file is not found or the file number set in CONFIG.SYS is less than 10.
2. PACK FILE NAME (File_name) ERROR.	The packed input format for the file name is wrong.
3. PACKED FILE NAME (File_name) ERROR.	The unpacked input format for the file name is wrong.
4. VERSION NUMBER ERROR : X.DOC	FOG6007 different from the version No. has been used.

System error

Error Message	Explanation
1. DIRECTORY FULL.	The directory is full.
2. DISK WRITE ERROR.	Writing on the disk is failed.

** \ sometimes appears as ¥, depending on the personal computer being used.*

APPENDIX A. E0C6007 INSTRUCTION SET

Classification	Mnemonic	Operand	Operation Code												Flag				Clock	Operation
			B	A	9	8	7	6	5	4	3	2	1	0	I	D	Z	C		
Branch instructions	PSET	p	1	1	1	0	0	1	0	p4	p3	p2	p1	p0					5	NBP ← p4, NPP ← p3~p0
	JP	s	0	0	0	0	s7	s6	s5	s4	s3	s2	s1	s0					5	PCB ← NBP, PCP ← NPP, PCS ← s7~s0
		C, s	0	0	1	0	s7	s6	s5	s4	s3	s2	s1	s0					5	PCB ← NBP, PCP ← NPP, PCS ← s7~s0 if C=1
		NC, s	0	0	1	1	s7	s6	s5	s4	s3	s2	s1	s0					5	PCB ← NBP, PCP ← NPP, PCS ← s7~s0 if C=0
		Z, s	0	1	1	0	s7	s6	s5	s4	s3	s2	s1	s0					5	PCB ← NBP, PCP ← NPP, PCS ← s7~s0 if Z=1
		NZ, s	0	1	1	1	s7	s6	s5	s4	s3	s2	s1	s0					5	PCB ← NBP, PCP ← NPP, PCS ← s7~s0 if Z=0
	JPBA		1	1	1	1	1	1	1	0	1	0	0	0					5	PCB ← NBP, PCP ← NPP, PCSH ← B, PCSL ← A
	CALL	s	0	1	0	0	s7	s6	s5	s4	s3	s2	s1	s0					7	M(SP-1) ← PCP, M(SP-2) ← PCSH, M(SP-3) ← PCSL+1 SP ← SP-3, PCP ← NPP, PCS ← s7~s0
	CALZ	s	0	1	0	1	s7	s6	s5	s4	s3	s2	s1	s0					7	M(SP-1) ← PCP, M(SP-2) ← PCSH, M(SP-3) ← PCSL+1 SP ← SP-3, PCP ← 0, PCS ← s7~s0
	RET		1	1	1	1	1	1	0	1	1	1	1	1					7	PCSL ← M(SP), PCSH ← M(SP+1), PCP ← M(SP+2) SP ← SP+3
	RETS		1	1	1	1	1	1	0	1	1	1	1	0					12	PCSL ← M(SP), PCSH ← M(SP+1), PCP ← M(SP+2) SP ← SP+3, PC ← PC+1
	RETD	*l*	0	0	0	1	*l*7	*l*6	*l*5	*l*4	*l*3	*l*2	*l*1	*l*0					12	PCSL ← M(SP), PCSH ← M(SP+1), PCP ← M(SP+2) SP ← SP+3, M(X) ← *l*3~*l*0, M(X+1) ← *l*7~*l*4, X ← X+2
System control instructions	NOP5		1	1	1	1	1	1	1	1	1	0	1	1					5	No operation (5 clock cycles)
	NOP7		1	1	1	1	1	1	1	1	1	1	1	1					7	No operation (7 clock cycles)
	HALT		1	1	1	1	1	1	1	1	1	0	0	0					5	Halt (stop clock)
Index operation instructions	INC	X	1	1	1	0	1	1	1	0	0	0	0	0					5	X ← X+1
		Y	1	1	1	0	1	1	1	1	0	0	0	0					5	Y ← Y+1
	LD	X, x	1	0	1	1	x7	x6	x5	x4	x3	x2	x1	x0					5	XH ← x7~x4, XL ← x3~x0
		Y, y	1	0	0	0	y7	y6	y5	y4	y3	y2	y1	y0					5	YH ← y7~y4, YL ← y3~y0
		XP, r	1	1	1	0	1	0	0	0	0	0	r1	r0					5	XP ← r
		XH, r	1	1	1	0	1	0	0	0	0	1	r1	r0					5	XH ← r
		XL, r	1	1	1	0	1	0	0	0	1	0	r1	r0					5	XL ← r
		YP, r	1	1	1	0	1	0	0	1	0	0	r1	r0					5	YP ← r
		YH, r	1	1	1	0	1	0	0	1	0	1	r1	r0					5	YH ← r
		YL, r	1	1	1	0	1	0	0	1	1	0	r1	r0					5	YL ← r
		r, XP	1	1	1	0	1	0	1	0	0	0	r1	r0					5	r ← XP
		r, XH	1	1	1	0	1	0	1	0	0	1	r1	r0					5	r ← XH
		r, XL	1	1	1	0	1	0	1	0	1	0	r1	r0					5	r ← XL
		r, YP	1	1	1	0	1	0	1	1	0	0	r1	r0					5	r ← YP
		r, YH	1	1	1	0	1	0	1	1	0	1	r1	r0					5	r ← YH
		r, YL	1	1	1	0	1	0	1	1	1	0	r1	r0					5	r ← YL
	ADC	XH, i	1	0	1	0	0	0	0	0	i3	i2	i1	i0			↕	↕	7	XH ← XH+i3~i0+C
		XL, i	1	0	1	0	0	0	0	1	i3	i2	i1	i0			↕	↕	7	XL ← XL+i3~i0+C
		YH, i	1	0	1	0	0	0	1	0	i3	i2	i1	i0			↕	↕	7	YH ← YH+i3~i0+C
		YL, i	1	0	1	0	0	0	1	1	i3	i2	i1	i0			↕	↕	7	YL ← YL+i3~i0+C

Classification	Mnemonic	Operand	Operation Code												Flag				Clock	Operation
			B	A	9	8	7	6	5	4	3	2	1	0	I	D	Z	C		
Index operation instructions	CP	XH, i	1	0	1	0	0	1	0	0	i3	i2	i1	i0			↕	↕	7	XH-i3~i0
		XL, i	1	0	1	0	0	1	0	1	i3	i2	i1	i0			↕	↕	7	XL-i3~i0
		YH, i	1	0	1	0	0	1	1	0	i3	i2	i1	i0			↕	↕	7	YH-i3~i0
		YL, i	1	0	1	0	0	1	1	1	i3	i2	i1	i0			↕	↕	7	YL-i3~i0
Data transfer instructions	LD	r, i	1	1	1	0	0	0	r1	r0	i3	i2	i1	i0					5	r←i3~i0
		r, q	1	1	1	0	1	1	0	0	r1	r0	q1	q0					5	r←q
		A, Mn	1	1	1	1	1	0	1	0	n3	n2	n1	n0					5	A←M(n3~n0)
		B, Mn	1	1	1	1	1	0	1	1	n3	n2	n1	n0					5	B←M(n3~n0)
		Mn, A	1	1	1	1	1	0	0	0	n3	n2	n1	n0					5	M(n3~n0)←A
		Mn, B	1	1	1	1	1	0	0	1	n3	n2	n1	n0					5	M(n3~n0)←B
	LDPX	MX, i	1	1	1	0	0	1	1	0	i3	i2	i1	i0					5	M(X)←i3~i0, X←X+1
		r, q	1	1	1	0	1	1	1	0	r1	r0	q1	q0					5	r←q, X←X+1
	LDPY	MY, i	1	1	1	0	0	1	1	1	i3	i2	i1	i0					5	M(Y)←i3~i0, Y←Y+1
		r, q	1	1	1	0	1	1	1	1	r1	r0	q1	q0					5	r←q, Y←Y+1
	LBPX	MX, *l*	1	0	0	1	*l*7	*l*6	*l*5	*l*4	*l*3	*l*2	*l*1	*l*0					5	M(X)←*l*3~*l*0, M(X+1)←*l*7~*l*4, X←X+2
Flag operation instructions	SET	F, i	1	1	1	1	0	1	0	0	i3	i2	i1	i0	↑	↑	↑	↑	7	F←F∨i3~i0
	RST	F, i	1	1	1	1	0	1	0	1	i3	i2	i1	i0	↓	↓	↓	↓	7	F←F∧i3~i0
	SCF		1	1	1	1	0	1	0	0	0	0	0	1				↑	7	C←1
	RCF		1	1	1	1	0	1	0	1	1	1	1	0				↓	7	C←0
	SZF		1	1	1	1	0	1	0	0	0	0	1	0			↑		7	Z←1
	RZF		1	1	1	1	0	1	0	1	1	1	0	1			↓		7	Z←0
	SDF		1	1	1	1	0	1	0	0	0	1	0	0		↑			7	D←1 (Decimal Adjuster ON)
	RDF		1	1	1	1	0	1	0	1	1	0	1	1		↓			7	D←0 (Decimal Adjuster OFF)
	EI		1	1	1	1	0	1	0	0	1	0	0	0	↑				7	I ←1 (Enables Interrupt)
	DI		1	1	1	1	0	1	0	1	0	1	1	1	↓				7	I ←0 (Disables Interrupt)
Stack operation instructions	INC	SP	1	1	1	1	1	1	0	1	1	0	1	1					5	SP← SP+1
	DEC	SP	1	1	1	1	1	1	0	0	1	0	1	1					5	SP← SP-1
	PUSH	r	1	1	1	1	1	1	0	0	0	0	r1	r0					5	SP← SP-1, M(SP)←r
		XP	1	1	1	1	1	1	0	0	0	1	0	0					5	SP← SP-1, M(SP)←XP
		XH	1	1	1	1	1	1	0	0	0	1	0	1					5	SP← SP-1, M(SP)←XH
		XL	1	1	1	1	1	1	0	0	0	1	1	0					5	SP← SP-1, M(SP)←XL
		YP	1	1	1	1	1	1	0	0	0	1	1	1					5	SP← SP-1, M(SP)←YP
		YH	1	1	1	1	1	1	0	0	1	0	0	0					5	SP← SP-1, M(SP)←YH
		YL	1	1	1	1	1	1	0	0	1	0	0	1					5	SP← SP-1, M(SP)←YL
		F	1	1	1	1	1	1	0	0	1	0	1	0					5	SP← SP-1, M(SP)←F
	POP	r	1	1	1	1	1	1	0	1	0	0	r1	r0					5	r←M(SP), SP←SP+1
		XP	1	1	1	1	1	1	0	1	0	1	0	0					5	XP←M(SP), SP←SP+1
		XH	1	1	1	1	1	1	0	1	0	1	0	1					5	XH←M(SP), SP←SP+1
		XL	1	1	1	1	1	1	0	1	0	1	1	0					5	XL←M(SP), SP←SP+1
		YP	1	1	1	1	1	1	0	1	0	1	1	1					5	YP←M(SP), SP←SP+1

Classification	Mnemonic	Operand	Operation Code												Flag				Clock	Operation
			B	A	9	8	7	6	5	4	3	2	1	0	I	D	Z	C		
Stack operation instructions	POP	YH	1	1	1	1	1	1	0	1	1	0	0	0					5	YH← M(SP), SP ← SP+1
		YL	1	1	1	1	1	1	0	1	1	0	0	1					5	YL ← M(SP), SP ← SP+1
		F	1	1	1	1	1	1	0	1	1	0	1	0	↕	↕	↕	↕	5	F ← M(SP), SP ← SP+1
	LD	SPH, r	1	1	1	1	1	1	1	0	0	0	r1	r0					5	SPH← r
		SPL, r	1	1	1	1	1	1	1	1	0	0	r1	r0					5	SPL ← r
		r, SPH	1	1	1	1	1	1	1	0	0	1	r1	r0					5	r← SPH
		r, SPL	1	1	1	1	1	1	1	1	0	1	r1	r0					5	r← SPL
Arithmetic instructions	ADD	r, i	1	1	0	0	0	0	r1	r0	i3	i2	i1	i0	★	↕	↕		7	r← r+i3~i0
		r, q	1	0	1	0	1	0	0	0	r1	r0	q1	q0	★	↕	↕		7	r← r+q
	ADC	r, i	1	1	0	0	0	1	r1	r0	i3	i2	i1	i0	★	↕	↕		7	r← r+i3~i0+C
		r, q	1	0	1	0	1	0	0	1	r1	r0	q1	q0	★	↕	↕		7	r← r+q+C
	SUB	r, q	1	0	1	0	1	0	1	0	r1	r0	q1	q0	★	↕	↕		7	r← r-q
	SBC	r, i	1	1	0	1	0	1	r1	r0	i3	i2	i1	i0	★	↕	↕		7	r← r-i3~i0-C
		r, q	1	0	1	0	1	0	1	1	r1	r0	q1	q0	★	↕	↕		7	r← r-q-C
	AND	r, i	1	1	0	0	1	0	r1	r0	i3	i2	i1	i0		↕			7	r← r∧ i3~i0
		r, q	1	0	1	0	1	1	0	0	r1	r0	q1	q0		↕			7	r← r∧ q
	OR	r, i	1	1	0	0	1	1	r1	r0	i3	i2	i1	i0		↕			7	r← r∨i3~i0
		r, q	1	0	1	0	1	1	0	1	r1	r0	q1	q0		↕			7	r← r∨q
	XOR	r, i	1	1	0	1	0	0	r1	r0	i3	i2	i1	i0		↕			7	r← r∀i3~i0
		r, q	1	0	1	0	1	1	1	0	r1	r0	q1	q0		↕			7	r← r∀q
	CP	r, i	1	1	0	1	1	1	r1	r0	i3	i2	i1	i0		↕	↕		7	r-i3~i0
		r, q	1	1	1	1	0	0	0	0	r1	r0	q1	q0		↕	↕		7	r-q
	FAN	r, i	1	1	0	1	1	0	r1	r0	i3	i2	i1	i0		↕			7	r∧i3~i0
		r, q	1	1	1	1	0	0	0	1	r1	r0	q1	q0		↕			7	r∧q
	RLC	r	1	0	1	0	1	1	1	1	r1	r0	r1	r0		↕	↕		7	d3 ← d2, d2 ← d1, d1 ← d0, d0 ← C, C← d3
	RRC	r	1	1	1	0	1	0	0	0	1	1	r1	r0		↕	↕		5	d3 ← C, d2 ← d3, d1 ← d2, d0 ← d1, C← d0
	INC	Mn	1	1	1	1	0	1	1	0	n3	n2	n1	n0		↕	↕		7	M(n3~n0) ← M(n3~n0)+1
	DEC	Mn	1	1	1	1	0	1	1	1	n3	n2	n1	n0		↕	↕		7	M(n3~n0) ← M(n3~n0)-1
	ACPX	MX, r	1	1	1	1	0	0	1	0	1	0	r1	r0	★	↕	↕		7	M(X) ← M(X)+r+C, X ← X+1
	ACPY	MY, r	1	1	1	1	0	0	1	0	1	1	r1	r0	★	↕	↕		7	M(Y) ← M(Y)+r+C, Y ← Y+1
	SCPX	MX, r	1	1	1	1	0	0	1	1	1	0	r1	r0	★	↕	↕		7	M(X) ← M(X)-r-C, X← X+1
	SCPY	MY, r	1	1	1	1	0	0	1	1	1	1	r1	r0	★	↕	↕		7	M(Y) ← M(Y)-r-C, Y← Y+1
	NOT	r	1	1	0	1	0	0	r1	r0	1	1	1	1		↕			7	r← $\overline{r}$

Abbreviations used in the explanations have the following meanings.

Symbols associated with registers and memory

A A register
B B register
X XHL register
(low order eight bits of index register IX)
Y YHL register
(low order eight bits of index register IY)
XH XH register
(high order four bits of XHL register)
XL XL register
(low order four bits of XHL register)
YH YH register
(high order four bits of YHL register)
YL YL register
(low order four bits of YHL register)
XP XP register
(high order four bits of index register IX)
YP YP register
(high order four bits of index register IY)
SP Stack pointer SP
SPH High-order four bits of stack pointer SP
SPL Low-order four bits of stack pointer SP
MX, M(X) Data memory whose address is specified with index register IX
MY, M(Y) Data memory whose address is specified with index register IY
Mn, M(n) Data memory address 000H–00FH (address specified with immediate data n of 00H–0FH)
M(SP) Data memory whose address is specified with stack pointer SP
r, q Two-bit register code
r, q is two-bit immediate data; according to the contents of these bits, they indicate registers A, B, and MX and MY (data memory whose addresses are specified with index registers IX and IY)

r		q		Register specified
r1	r0	q1	q0	
0	0	0	0	A
0	1	0	1	B
1	0	1	0	MX
1	1	1	1	MY

Symbols associated with program counter

NBP New bank pointer
NPP New page pointer
PCB Program counter bank
PCP Program counter page
PCS Program counter step
PCSH Four high order bits of PCS
PCSL Four low order bits of PCS

Symbols associated with flags

F Flag register (I, D, Z, C)
C Carry flag
Z Zero flag
D Decimal flag
I Interrupt flag
↓ Flag reset
↑ Flag set
↕ Flag set or reset

Associated with immediate data

p Five-bit immediate data or label 00H–1FH
s Eight-bit immediate data or label 00H–0FFH
l Eight-bit immediate data 00H–0FFH
i Four-bit immediate data 00H–0FH

Associated with arithmetic and other operations

+ Add
- Subtract
∧ Logical AND
∨ Logical OR
∀ Exclusive-OR
★ Add-subtract instruction for decimal operation when the D flag is set

APPENDIX B. E0C6007 RAM MAP

RAM (000H–07FH)

PROGRAM NAME :

P	H	L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	NAME																
		MSB																
		LSB																
	1	NAME																
		MSB																
		LSB																
	2	NAME																
		MSB																
		LSB																
	3	NAME																
		MSB																
		LSB																
	4	NAME																
		MSB																
		LSB																
	5	NAME																
		MSB																
		LSB																
	6	NAME																
		MSB																
		LSB																
	7	NAME																
		MSB																
		LSB																

RAM (080H–0FFH)

PROGRAM NAME :																		
P	H	L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	8	NAME																
		MSB																
		LSB																
	9	NAME																
		MSB																
		LSB																
	A	NAME																
		MSB																
		LSB																
	B	NAME																
		MSB																
		LSB																
	C	NAME																
		MSB																
		LSB																
	D	NAME																
		MSB																
		LSB																
	E	NAME																
		MSB																
		LSB																
	F	NAME																
		MSB																
		LSB																

RAM (100H–17FH)

PROGRAM NAME :																		
P	H	L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	0	NAME																
		MSB																
		LSB																
	1	NAME																
		MSB																
		LSB																
	2	NAME																
		MSB																
		LSB																
	3	NAME																
		MSB																
		LSB																
	4	NAME																
		MSB																
		LSB																
	5	NAME																
		MSB																
		LSB																
	6	NAME																
		MSB																
		LSB																
	7	NAME																
		MSB																
		LSB																

RAM (180H–1FFH)

PROGRAM NAME :																		
P	H	L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	8	NAME																
		MSB																
		LSB																
	9	NAME																
		MSB																
		LSB																
	A	NAME																
		MSB																
		LSB																
	B	NAME																
		MSB																
		LSB																
	C	NAME																
		MSB																
		LSB																
	D	NAME																
		MSB																
		LSB																
	E	NAME																
		MSB																
		LSB																
	F	NAME																
		MSB																
		LSB																

Display memory (E00H–E4FH)

PROGRAM NAME :																		
P	H	L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
E	0	NAME																
		MSB	[3,0]	[7,0]	[3,1]	[7,1]	[3,2]	[7,2]	[3,3]	[7,3]	[3,4]	[7,4]	[3,5]	[7,5]	[3,6]	[7,6]	[3,7]	[7,7]
			[2,0]	[6,0]	[2,1]	[6,1]	[2,2]	[6,2]	[2,3]	[6,3]	[2,4]	[6,4]	[2,5]	[6,5]	[2,6]	[6,6]	[2,7]	[6,7]
			[1,0]	[5,0]	[1,1]	[5,1]	[1,2]	[5,2]	[1,3]	[5,3]	[1,4]	[5,4]	[1,5]	[5,5]	[1,6]	[5,6]	[1,7]	[5,7]
		LSB	[0,0]	[4,0]	[0,1]	[4,1]	[0,2]	[4,2]	[0,3]	[4,3]	[0,4]	[4,4]	[0,5]	[4,5]	[0,6]	[4,6]	[0,7]	[4,7]
	1	NAME																
		MSB	[3,8]	[7,8]	[3,9]	[7,9]	[3,10]	[7,10]	[3,11]	[7,11]	[3,12]	[7,12]	[3,13]	[7,13]	[3,14]	[7,14]	[3,15]	[7,15]
			[2,8]	[6,8]	[2,9]	[6,9]	[2,10]	[6,10]	[2,11]	[6,11]	[2,12]	[6,12]	[2,13]	[6,13]	[2,14]	[6,14]	[2,15]	[6,15]
			[1,8]	[5,8]	[1,9]	[5,9]	[1,10]	[5,10]	[1,11]	[5,11]	[1,12]	[5,12]	[1,13]	[5,13]	[1,14]	[5,14]	[1,15]	[5,15]
		LSB	[0,8]	[4,8]	[0,9]	[4,9]	[0,10]	[4,10]	[0,11]	[4,11]	[0,12]	[4,12]	[0,13]	[4,13]	[0,14]	[4,14]	[0,15]	[4,15]
	2	NAME																
		MSB	[3,16]	[7,16]	[3,17]	[7,17]	[3,18]	[7,18]	[3,19]	[7,19]	[3,20]	[7,20]	[3,21]	[7,21]	[3,22]	[7,22]	[3,23]	[7,23]
			[2,16]	[6,16]	[2,17]	[6,17]	[2,18]	[6,18]	[2,19]	[6,19]	[2,20]	[6,20]	[2,21]	[6,21]	[2,22]	[6,22]	[2,23]	[6,23]
			[1,16]	[5,16]	[1,17]	[5,17]	[1,18]	[5,18]	[1,19]	[5,19]	[1,20]	[5,20]	[1,21]	[5,21]	[1,22]	[5,22]	[1,23]	[5,23]
		LSB	[0,16]	[4,16]	[0,17]	[4,17]	[0,18]	[4,18]	[0,19]	[4,19]	[0,20]	[4,20]	[0,21]	[4,21]	[0,22]	[4,22]	[0,23]	[4,23]
	3	NAME																
		MSB	[3,24]	[7,24]	[3,25]	[7,25]	[3,26]	[7,26]	[3,27]	[7,27]	[3,28]	[7,28]	[3,29]	[7,29]	[3,30]	[7,30]	[3,31]	[7,31]
			[2,24]	[6,24]	[2,25]	[6,25]	[2,26]	[6,26]	[2,27]	[6,27]	[2,28]	[6,28]	[2,29]	[6,29]	[2,30]	[6,30]	[2,31]	[6,31]
			[1,24]	[5,24]	[1,25]	[5,25]	[1,26]	[5,26]	[1,27]	[5,27]	[1,28]	[5,28]	[1,29]	[5,29]	[1,30]	[5,30]	[1,31]	[5,31]
		LSB	[0,24]	[4,24]	[0,25]	[4,25]	[0,26]	[4,26]	[0,27]	[4,27]	[0,28]	[4,28]	[0,29]	[4,29]	[0,30]	[4,30]	[0,31]	[4,31]
	4	NAME																
		MSB	[3,32]	[7,32]	[3,33]	[7,33]	[3,34]	[7,34]	[3,35]	[7,35]	[3,36]	[7,36]	[3,37]	[7,37]	[3,38]	[7,38]	[3,39]	[7,39]
			[2,32]	[6,32]	[2,33]	[6,33]	[2,34]	[6,34]	[2,35]	[6,35]	[2,36]	[6,36]	[2,37]	[6,37]	[2,38]	[6,38]	[2,39]	[6,39]
			[1,32]	[5,32]	[1,33]	[5,33]	[1,34]	[5,34]	[1,35]	[5,35]	[1,36]	[5,36]	[1,37]	[5,37]	[1,38]	[5,38]	[1,39]	[5,39]
		LSB	[0,32]	[4,32]	[0,33]	[4,33]	[0,34]	[4,34]	[0,35]	[4,35]	[0,36]	[4,36]	[0,37]	[4,37]	[0,38]	[4,38]	[0,39]	[4,39]

Display memory (E80H–ECFH)

PROGRAM NAME :																		
P	H	L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
E	8	NAME																
		MSB	[11,0]	[15,0]	[11,1]	[15,1]	[11,2]	[15,2]	[11,3]	[15,3]	[11,4]	[15,4]	[11,5]	[15,5]	[11,6]	[15,6]	[11,7]	[15,7]
			[10,0]	[14,0]	[10,1]	[14,1]	[10,2]	[14,2]	[10,3]	[14,3]	[10,4]	[14,4]	[10,5]	[14,5]	[10,6]	[14,6]	[10,7]	[14,7]
			[9,0]	[13,0]	[9,1]	[13,1]	[9,2]	[13,2]	[9,3]	[13,3]	[9,4]	[13,4]	[9,5]	[13,5]	[9,6]	[13,6]	[9,7]	[13,7]
		LSB	[8,0]	[12,0]	[8,1]	[12,1]	[8,2]	[12,2]	[8,3]	[12,3]	[8,4]	[12,4]	[8,5]	[12,5]	[8,6]	[12,6]	[8,7]	[12,7]
	9	NAME																
		MSB	[11,8]	[15,8]	[11,9]	[15,9]	[11,10]	[15,10]	[11,11]	[15,11]	[11,12]	[15,12]	[11,13]	[15,13]	[11,14]	[15,14]	[11,15]	[15,15]
			[10,8]	[14,8]	[10,9]	[14,9]	[10,10]	[14,10]	[10,11]	[14,11]	[10,12]	[14,12]	[10,13]	[14,13]	[10,14]	[14,14]	[10,15]	[14,15]
			[9,8]	[13,8]	[9,9]	[13,9]	[9,10]	[13,10]	[9,11]	[13,11]	[9,12]	[13,12]	[9,13]	[13,13]	[9,14]	[13,14]	[9,15]	[13,15]
		LSB	[8,8]	[12,8]	[8,9]	[12,9]	[8,10]	[12,10]	[8,11]	[12,11]	[8,12]	[12,12]	[8,13]	[12,13]	[8,14]	[12,14]	[8,15]	[12,15]
	A	NAME																
		MSB	[11,16]	[15,16]	[11,17]	[15,17]	[11,18]	[15,18]	[11,19]	[15,19]	[11,20]	[15,20]	[11,21]	[15,21]	[11,22]	[15,22]	[11,23]	[15,23]
			[10,16]	[14,16]	[10,17]	[14,17]	[10,18]	[14,18]	[10,19]	[14,19]	[10,20]	[14,20]	[10,21]	[14,21]	[10,22]	[14,22]	[10,23]	[14,23]
			[9,16]	[13,16]	[9,17]	[13,17]	[9,18]	[13,18]	[9,19]	[13,19]	[9,20]	[13,20]	[9,21]	[13,21]	[9,22]	[13,22]	[9,23]	[13,23]
		LSB	[8,16]	[12,16]	[8,17]	[12,17]	[8,18]	[12,18]	[8,19]	[12,19]	[8,20]	[12,20]	[8,21]	[12,21]	[8,22]	[12,22]	[8,23]	[12,23]
	B	NAME																
		MSB	[11,24]	[15,24]	[11,25]	[15,25]	[11,26]	[15,26]	[11,27]	[15,27]	[11,28]	[15,28]	[11,29]	[15,29]	[11,30]	[15,30]	[11,31]	[15,31]
			[10,24]	[14,24]	[10,25]	[14,25]	[10,26]	[14,26]	[10,27]	[14,27]	[10,28]	[14,28]	[10,29]	[14,29]	[10,30]	[14,30]	[10,31]	[14,31]
			[9,24]	[13,24]	[9,25]	[13,25]	[9,26]	[13,26]	[9,27]	[13,27]	[9,28]	[13,28]	[9,29]	[13,29]	[9,30]	[13,30]	[9,31]	[13,31]
		LSB	[8,24]	[12,24]	[8,25]	[12,25]	[8,26]	[12,26]	[8,27]	[12,27]	[8,28]	[12,28]	[8,29]	[12,29]	[8,30]	[12,30]	[8,31]	[12,31]
	C	NAME																
		MSB	[11,32]	[15,32]	[11,33]	[15,33]	[11,34]	[15,34]	[11,35]	[15,35]	[11,36]	[15,36]	[11,37]	[15,37]	[11,38]	[15,38]	[11,39]	[15,39]
			[10,32]	[14,32]	[10,33]	[14,33]	[10,34]	[14,34]	[10,35]	[14,35]	[10,36]	[14,36]	[10,37]	[14,37]	[10,38]	[14,38]	[10,39]	[14,39]
			[9,32]	[13,32]	[9,33]	[13,33]	[9,34]	[13,34]	[9,35]	[13,35]	[9,36]	[13,36]	[9,37]	[13,37]	[9,38]	[13,38]	[9,39]	[13,39]
		LSB	[8,32]	[12,32]	[8,33]	[12,33]	[8,34]	[12,34]	[8,35]	[12,35]	[8,36]	[12,36]	[8,37]	[12,37]	[8,38]	[12,38]	[8,39]	[12,39]

I/O memory (F00H–F7FH)

PROGRAM NAME :																		
P	H	L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
F	0	NAME	ZTI	ZISW	ZIPT		ZIK0											
		MSB	IT1	0	0		0											
			IT2	0	0		0											
			IT8	ISW1	0		0											
		LSB	IT32	ISW0	IPT		IK0											
	1	NAME	ZEIT	ZEISW	ZEIPT		ZEIK0											
		MSB	EIT1	0	0		EIK03											
			EIT2	0	0		EIK02											
			EIT8	EISW1	0		EIK01											
		LSB	EIT32	EISW0	EIPT		EIK00											
	2	NAME	ZTML	ZTMH	ZSWL	ZSWH	ZPTL	ZPTH	ZRDL	ZRDH								
		MSB	TM3	TM7	SWL3	SWH3	PT3	PT7	RD3	RD7								
			TM2	TM6	SWL2	SWH2	PT2	PT6	RD2	RD6								
			TM1	TM5	SWL1	SWH1	PT1	PT5	RD1	RD5								
		LSB	TM0	TM4	SWL0	SWH0	PT0	PT4	RD0	RD4								
	3	NAME																
		MSB																
		LSB																
	4	NAME	ZK0															
		MSB	K03															
			K02															
			K01															
		LSB	K00															
	5	NAME				ZR3	ZR4											
		MSB				R33	R43											
						R32	R42											
						0	R41											
		LSB				0	R40											
	6	NAME	ZP0															
		MSB	P03															
			P02															
			P01															
		LSB	P00															
	7	NAME	ZOSC	ZLCD	ZLC		ZBZ	ZENV	ZTRST	ZSWR	ZPTR	ZPTC		ZHZR		ZIOC	ZPUP	ZLCD
		MSB	CLKCHG	ALOFF	LC3		SHOTPW	BZSHOT	0	0	0	PTCOUT		HZR3		0	0	0
			OSCC	ALON	LC2		BZFQ2	ENVRST	0	0	0	PTC2		0		0	0	0
			VSC1	LDUTY	LC1		BZFQ1	ENVRT	TMRST	SWRST	PTRST	PTC1		0		0	0	0
		LSB	VSC0	HLMOD	LC0		BZFQ0	ENVON	WDRST	SWRUN	PTRUN	PTC0		0		IOC0	PUP0	LCDOFF

APPENDIX C. E0C6007 I/O MEMORY MAP

I/O memory (F00H–F25H)

Address	Register D3	D2	D1	D0	Name	Init *1	1	0	Comment
F00H	IT1	IT2	IT8	IT32	IT1 *3	0	Yes	No	Interrupt factor flag (clock timer 1 Hz)
					IT2 *3	0	Yes	No	Interrupt factor flag (clock timer 2 Hz)
	R				IT8 *3	0	Yes	No	Interrupt factor flag (clock timer 8 Hz)
					IT32 *3	0	Yes	No	Interrupt factor flag (clock timer 32 Hz)
F01H	0	0	ISW1	ISW0	0 *4	– *2			
					0 *4	– *2			
	R				ISW1 *3	0	Yes	No	Interrupt factor flag (stopwatch 1 Hz)
					ISW0 *3	0	Yes	No	Interrupt factor flag (stopwatch 10 Hz)
F02H	0	0	0	IPT	0 *4	– *2			
					0 *4	– *2			
	R				0 *4	– *2			
					IPT *3	0	Yes	No	Interrupt factor flag (programmable timer)
F04H	0	0	0	IK0	0 *4	– *2			
					0 *4	– *2			
	R				0 *4	– *2			
					IK0 *3	0	Yes	No	Interrupt factor flag (K00–K03)
F10H	EIT1	EIT2	EIT8	EIT32	EIT1	0	Enable	Mask	Interrupt mask register (clock timer 1 Hz)
					EIT2	0	Enable	Mask	Interrupt mask register (clock timer 2 Hz)
	R/W				EIT8	0	Enable	Mask	Interrupt mask register (clock timer 8 Hz)
					EIT32	0	Enable	Mask	Interrupt mask register (clock timer 32 Hz)
F11H	0	0	EISW1	EISW0	0 *4	– *2			
					0 *4	– *2			
	R		R/W		EISW1	0	Enable	Mask	Interrupt mask register (stopwatch 1 Hz)
					EISW0	0	Enable	Mask	Interrupt mask register (stopwatch 10 Hz)
F12H	0	0	0	EIPT	0 *4	– *2			
					0 *4	– *2			
	R			R/W	0 *4	– *2			
					EIPT	0	Enable	Mask	Interrupt mask register (programmble timer)
F14H	EIK03	EIK02	EIK01	EIK00	EIK03	0	Enable	Mask	Interrupt mask register (K03)
					EIK02	0	Enable	Mask	Interrupt mask register (K02)
	R/W				EIK01	0	Enable	Mask	Interrupt mask register (K01)
					EIK00	0	Enable	Mask	Interrupt mask register (K00)
F20H	TM3	TM2	TM1	TM0	TM3	0			Clock timer data (16 Hz)
					TM2	0			Clock timer data (32 Hz)
	R				TM1	0			Clock timer data (64 Hz)
					TM0	0			Clock timer data (128 Hz)
F21H	TM7	TM6	TM5	TM4	TM7	0			Clock timer data (1 Hz)
					TM6	0			Clock timer data (2 Hz)
	R				TM5	0			Clock timer data (4 Hz)
					TM4	0			Clock timer data (8 Hz)
F22H	SWL3	SWL2	SWL1	SWL0	SWL3	0			MSB
					SWL2	0			Stopwatch timer
	R				SWL1	0			1/100 sec data (BCD)
					SWL0	0			LSB
F23H	SWH3	SWH2	SWH1	SWH0	SWH3	0			MSB
					SWH2	0			Stopwatch timer
	R				SWH1	0			1/10 sec data (BCD)
					SWH0	0			LSB
F24H	PT3	PT2	PT1	PT0	PT3	X *5			MSB
					PT2	X *5			Programmable timer data (low-order)
	R				PT1	X *5			
					PT0	X *5			LSB
F25H	PT7	PT6	PT5	PT4	PT7	X *5			MSB
					PT6	X *5			Programmable timer data (high-order)
	R				PT5	X *5			
					PT4	X *5			LSB

*1 Initial value following initial reset
*2 Not set in the circuit
*3 Reset (0) immediately after being read
*4 Always "0" when being read
*5 Undefined

I/O memory (F26H–F76H)

Address	Register D3	D2	D1	D0	Name	Init *1	1	0	Comment
F26H	RD3	RD2	RD1	RD0	RD3	X *5			MSB
					RD2	X *5			Programmable timer
	R/W				RD1	X *5			reload data (low-order)
					RD0	X *5			LSB
F27H	RD7	RD6	RD5	RD4	RD7	X *5			MSB
					RD6	X *5			Programmable timer
	R/W				RD5	X *5			reload data (high-order)
					RD4	X *5			LSB
F40H	K03	K02	K01	K00	K03	– *2	High	Low	Input port (K00–K03)
					K02	– *2	High	Low	
	R				K01	– *2	High	Low	
					K00	– *2	High	Low	
F53H	R33	R32	0	0	R33	X *5	High	Low	Output port (R33)
							Off	On	PTCLK output
	R/W		R		R32	X *5	High	Low	Output port (R32)
					0 *4	– *2			
					0 *4	– *2			
F54H	R43	R42	R41	R40	R43	1	High	Low	Output port (R43)
							Off	On	Buzzer output (BZ)
	R/W				R42	1	High	Low	Output port (R42)
							Off	On	Clock output (FOUT)
							*6	*6	[Buzzer inverted output (#BZ)]
					R41	1	High	Low	Output port (R41)
					R40	1	High	Low	Output port (R40)
							Off	On	Clock inverted output (#FOUT)
F60H	P03	P02	P01	P00	P03	X *5	High	Low	I/O port (P00–P03)
					P02	X *5	High	Low	
	R/W				P01	X *5	High	Low	
					P00	X *5	High	Low	
F70H	CLKCHG	OSCC	VSC1	VSC0	CLKCHG	0	OSC3	OSC1	CPU system clock switch
					OSCC	0	On	Off	OSC3 oscillation On/Off
	R/W				VSC1	0			CPU operating voltage switch
					VSC0	0			
F71H	ALOFF	ALON	LDUTY	HLMOD	ALOFF	1	All off	Normal	All LCD dots fade out control
					ALON	0	All on	Normal	All LCD dots displayed control
	R/W				LDUTY	0	1/8	1/16	LCD drive duty switch
					HLMOD	0	HLMOD	Normal	Heavy load protection mode
F72H	LC3	LC2	LC1	LC0	LC3	X *5			LCD contrast adjustment
					LC2	X *5			LC3–LC0 = 0 light
	R/W				LC1	X *5			:
					LC0	X *5			LC3–LC0 = 15 dark
F74H	SHOTPW	BZFQ2	BZFQ1	BZFQ0	SHOTPW	0	62.5 ms	31.25 ms	1-shot buzzer pulse width
					BZFQ2	0			Buzzer frequency selection
	R/W				BZFQ1	0			
					BZFQ0	0			
F75H	BZSHOT	ENVRST	ENVRT	ENVON	BZSHOT	0	Trigger	–	1-shot buzzer trigger (W)
							BUSY	READY	Status (R)
	W	W	R/W		ENVRST	RESET	Reset	–	Envelope reset
	R				ENVRT	0	1.0 sec	0.5 sec	Envelope cycle selection
					ENVON	0	On	Off	Envelope On/Off
F76H	0	0	TMRST	WDRST	0 *4	– *2			
					0 *4	– *2			
	R		W		TMRST *4	Reset	Reset	–	Clock timer reset
					WDRST	Reset	Reset	–	Watchdog timer reset

*1 Initial value following initial reset
*2 Not set in the circuit
*3 Reset (0) immediately after being read
*4 Always "0" when being read
*5 Undefined
*6 When selecting options enclosed in brackets [ ] as output option, the output register will function as register only and will not affect the individual outputs.

I/O memory (F77H–F7FH)

Address	Register								Comment
	D3	D2	D1	D0	Name	Init *1	1	0	
F77H	0	0	SWRST	SWRUN	0 *4	– *2			
					0 *4	– *2			
	R		W	R/W	SWRST *4	Reset	Reset	–	Stopwatch timer reset
					SWRUN	0	Run	Stop	Stopwatch timer Run/Stop
F78H	0	0	PTRST	PTRUN	0 *4	– *2			
					0 *4	– *2			
	R		W	R/W	PTRST *4	Reset	Reset	–	Programmable timer reset
					PTRUN	0	Run	Stop	Programmable timer Run/Stop
F79H	PTCOUT	PTC2	PTC1	PTC0	PTCOUT	0	On	Off	Programmable timer clock output
					PTC2	0			Programmable timer input clock selection
	R/W				PTC1	0			
					PTC0	0			
F7BH	HZR3	0	0	0	HZR3	0	Output	High-Z	R32–R33 output high-impedance control
					0 *4	– *2			
	R/W	R			0 *4	– *2			
					0 *4	– *2			
F7DH	0	0	0	IOC0	0 *4	– *2			
					0 *4	– *2			
	R			R/W	0 *4	– *2			
					IOC0	0	Output	Input	I/O control (P00–P03)
F7EH	0	0	0	PUP0	0 *4	– *2			
					0 *4	– *2			
	R			R/W	0 *4	– *2			
					PUP0	0	Off	On	I/O pull up resistor On/Off (P00–P03)
F7FH	0	0	0	LCDOFF	0 *4	– *2			
					0 *4	– *2			
	R			R/W	0 *4	– *2			
					LCDOFF	1	Normal	Off	LCD display control

*1 Initial value following initial reset
*2 Not set in the circuit
*3 Reset (0) immediately after being read
*4 Always "0" when being read
*5 Undefined

APPENDIX D. TROUBLESHOOTING

Tool	Problem	Remedy measures
ICE6200	Nothing appears on the screen, or nothing works, after activation.	Check the following and remedy if necessary: • Is the RS-232C cable connected correctly? • Is the RS-232C driver installed? • Is SPEED.COM or MODE.COM on the disk? • Is the execution file correct? MS-DOS ICS6007J.EXE PC-DOS ICS6007W.EXE • Is the DOS version correct? MS-DOS Ver. 3.1 or later PC-DOS Ver. 2.1 or later • Is the DIP switches that set the baud rate of the main ICE6200 unit set correctly? • Is the breaker of the ICE6200 set to ON?
	The ICE6200 breaker tripped immediately after activation.	Check the following and remedy if necessary: • Are connectors F1 and F5 connected to the EVA6007 correctly? • Is the target board power short-circuiting?
	<ILLEGAL VERSION ICE6200> appears on the screen immediately after activation.	The wrong version of ICE6200 is being used. Use the latest version.
	<ILLEGAL VERSION PARAMETER FILE> appears on the screen immediately after activation.	The wrong version of ICS6007P.PAR is being used. Use the latest version.
	Immediate values A (10) and B (11) cannot be entered correctly with the A command.	The A and B registers are reserved for the entry of A and B. Write 0A and 0B when entering A (10) and B (11). *Example:* `LD A, B` Data in the B register is loaded into the A register. `LD B, 0A` Immediate value A is loaded into the B register.
	<UNUSED AREA> is displayed by the SD command.	This message is output when the address following one in which data is written is unused. It does not indicates problem. Data is correctly set in areas other than the read-only area.
	You can not do a real-time run in break-trace mode.	Since the CPU stops temporarily when breaking conditions are met, executing in a real-time is not performed.
	Output from the EVA is impossible when data is written to the I/O memory for Buzzer and Fout output with the ICE command.	Output is possible only in the real-time run mode.

Tool	Problem	Remedy measures
ASM6007	An R error occurs although the final page is passed.	The cross assembler is designed to output "R error" every time the page is changed. Use a pseudo-instruction to set the memory, such as ORG or PAGE, to change the page. See "Memory setting pseudo-instructions" in the cross assembler manual.
MDC6007	Activation is impossible.	Check the following and remedy if necessary: • Is the number of files set at ten or more in OS environment file CONFIG.SYS?
EVA6007	The EVA6007 does not work when it is used independently.	Check the following and remedy if necessary: • Has the EPROM for F.HEX been replaced by the EPROM for the target? • Is the EPROM for F.HEX installed correctly? • Is the appropriate voltage being supplied? (5V DC, 3A, or more) • Are the program ROMs (H and L) installed correctly? • Is data written from address 4000H? (When the 27C256 is used as the program ROM) • Is the EN/DIS switch on the EVA6007 set to EN?
	Target segment does not light.	Check the following and remedy if necessary: • Has the VADJ VR inside the EVA6007 top cover been turned to a lower setting?

EPSON International Sales Operations

AMERICA

S-MOS SYSTEMS, INC.
150 River Oaks Parkway
San Jose, CA 95134, U.S.A.
Phone: +1-408-922-0200 Fax: +1-408-922-0238
Telex: 176079 SMOS SNJUD

S-MOS SYSTEMS, INC.
EASTERN AREA SALES AND TECHNOLOGY CENTER
301 Edgewater Place, Suite 120
Wakefield, MA 01880, U.S.A.
Phone: +1-617-246-3600 Fax: +1-617-246-5443

S-MOS SYSTEMS, INC.
SOUTH EASTERN AREA SALES AND TECHNOLOGY CENTER
4300 Six Forks Road, Suite 430
Raleigh, NC 27609, U.S.A.
Phone: +1-919-781-7667 Fax: +1-919-781-6778

S-MOS SYSTEMS, INC.
CENTRAL AREA SALES AND TECHNOLOGY CENTER
1450 E.American Lane, Suite 1550
Schaumburg, IL 60173, U.S.A.
Phone: +1-847-517-7667 Fax: +1-847-517-7601

EUROPE

- HEADQUARTERS -

EPSON EUROPE ELECTRONICS GmbH
Riesstrasse 15
80992 Muenchen, GERMANY
Phone: +49-(0)89-14005-0 Fax: +49-(0)89-14005-110

- GERMANY -

EPSON EUROPE ELECTRONICS GmbH
SALES OFFICE
Breidenbachstrasse 46
D-51373 Leverkusen, GERMANY
Phone: +49-(0)214-83070-0 Fax: +49-(0)214-83070-10

- UNITED KINGDOM -

EPSON EUROPE ELECTRONICS GmbH
UK BRANCH OFFICE
G6 Doncastle House, Doncastle Road
Bracknell, Berkshire RG12 8PE, ENGLAND
Phone: +44-(0)1344-381700 Fax: +44-(0)1344-381701

- FRANCE -

EPSON EUROPE ELECTRONICS GmbH
FRENCH BRANCH OFFICE
1 Avenue de l' Atlantique, LP 915 Les Conquerants
Z.A. de Courtaboeuf 2, F-91976 Les Ulis Cedex, FRANCE
Phone: +33-(0)1-64862350 Fax: +33-(0)1-64862355

ASIA

- HONG KONG, CHINA -

EPSON HONG KONG LTD.
20/F., Harbour Centre, 25 Harbour Road
Wanchai, HONG KONG
Phone: +852-2585-4600 Fax: +852-2827-4346
Telex: 65542 EPSCO HX

- CHINA -

SHANGHAI EPSON ELECTRONICS CO., LTD.
4F, Bldg., 27, No. 69, Gui Jing Road
Caohejing, Shanghai, CHINA
Phone: 21-6485-5552 Fax: 21-6485-0775

- TAIWAN, R.O.C. -

EPSON TAIWAN TECHNOLOGY & TRADING LTD.
10F, No. 287, Nanking East Road, Sec. 3
Taipei, TAIWAN, R.O.C.
Phone: 02-2717-7360 Fax: 02-2712-9164
Telex: 24444 EPSONTB

EPSON TAIWAN TECHNOLOGY & TRADING LTD.
HSINCHU OFFICE
13F-3, No. 295, Kuang-Fu Road, Sec. 2
HsinChu 300, TAIWAN, R.O.C.
Phone: 03-573-9900 Fax: 03-573-9169

- SINGAPORE -

EPSON SINGAPORE PTE., LTD.
No. 1 Temasek Avenue, #36-00
Millenia Tower, SINGAPORE 039192
Phone: +65-337-7911 Fax: +65-334-2716

- KOREA -

SEIKO EPSON CORPORATION
KOREA OFFICE
10F, KLI 63 Bldg., 60 Yoido-Dong
Youngdeungpo-Ku, Seoul, 150-010, KOREA
Phone: 02-784-6027 Fax: 02-767-3677

- JAPAN -

SEIKO EPSON CORPORATION
ELECTRONIC DEVICES MARKETING DIVISION

Electronic Device Marketing Department
IC Marketing & Engineering Group
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-(0)42-587-5816 Fax: +81-(0)42-587-5624

ED International Marketing Department I
(Europe & U.S.A.)
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-(0)42-587-5812 Fax: +81-(0)42-587-5564

ED International Marketing Department II (Asia)
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-(0)42-587-5814 Fax: +81-(0)42-587-5110

In pursuit of **“Saving” Technology**, Epson electronic devices.
Our lineup of semiconductors, liquid crystal displays and quartz devices
assists in creating the products of our customers’ dreams.
Epson IS energy savings.

EPSON

SEIKO EPSON CORPORATION
ELECTRONIC DEVICES MARKETING DIVISION

■ Electronic devices information on the Epson WWW server

http://www.epson.co.jp

Issue OCTOBER 1998, Printed in Japan Ⓜ A