MF733-03

EPSON

CMOS 4-BIT SINGLE CHIP MICROCOMPUTER **E0C6256**

DEVELOPMENT TOOL MANUAL

ENERGY SAVING EPSON

SEIKO EPSON CORPORATION

E0C6256 Development Tool Manual

PREFACE

This manual mainly explains the outline of the development support tool for the 4-bit Single Chip Microcomputer E0C6256.

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 EVA6256 Manual ICE62R (ICE6200) Hardware Manual
Development procedure	☞ E0C62 Family Technical Guide
Device (E0C6256)	☞ E0C6256 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 envilonment and how to generate the execution disk.

1.1 Configuration of DEV6256

The below software are included in the product of the E0C6256 development support tool DEV6256.

1. Development Tool Management System DMS6200 ..... Menu selection for each software / start-up software
2. Cross Assembler ASM6256 ..... Cross assembler for program preparation
3. Function Option Generator FOG6256 ..... Function option data preparation program
4. Segment Option Generator SOG6256 ..... Segment option data preparation program
5. ICE Control Software ICS6256 ..... ICE control program
6. Mask Data Checker MDC6256 ..... Mask data preparation program

1.2 Developmental Environment

The software product of the development support tool DEV6256 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 E0C6256, 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.

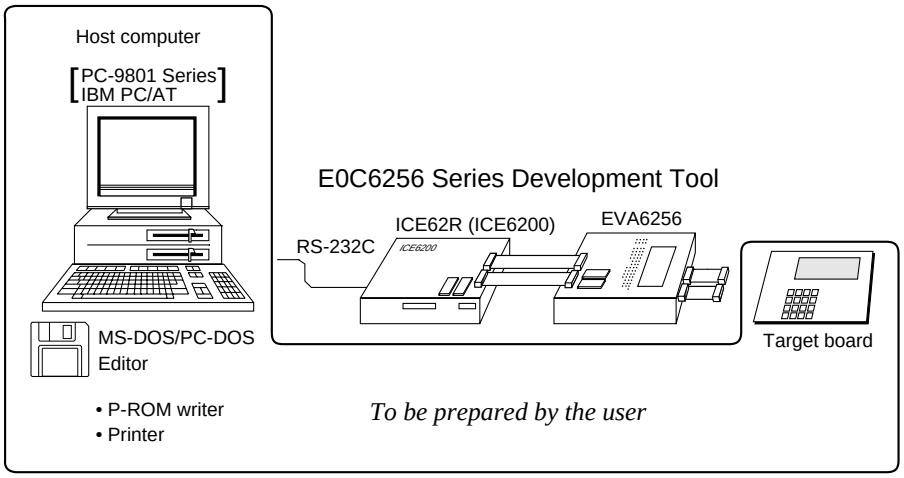


Fig. 1.2.1 System configuration

Note The DEV6256 system requires a host computer with a RAM capacity of about 140K bytes. Since the ICE62R (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 DEV6256.

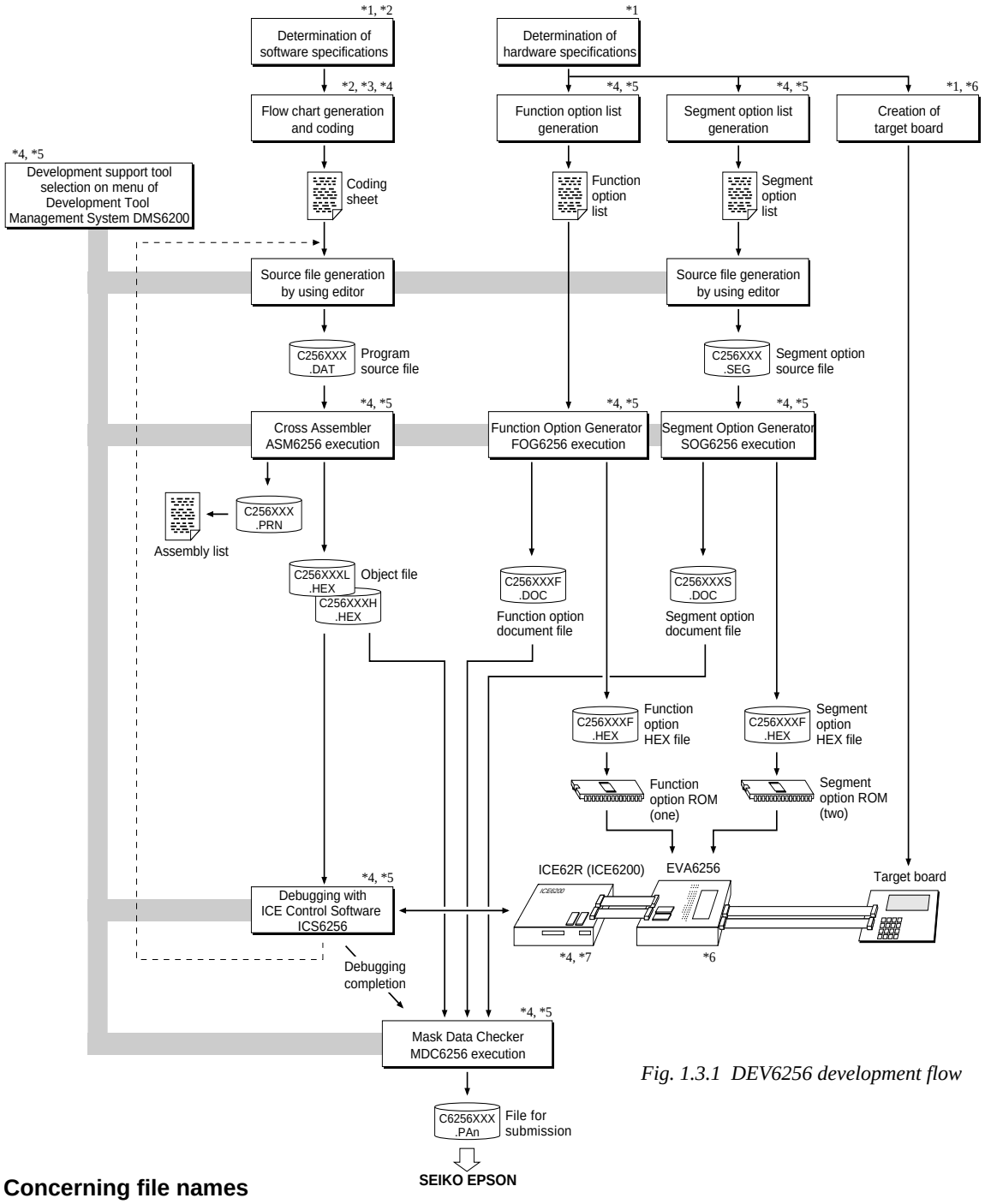


Fig. 1.3.1 DEV6256 development flow

Concerning file names

All the input-output file name for the each development support tool commonly use "C256XXX". In principle each file should be produced in this manner. Seiko Epson will designate the "XXX" for each customer.

☞ Reference Manual

*1 E0C6256 Technical Hardware Manual
*2 E0C6256 Technical Software Manual
*3 E0C6200/6200A Core CPU Manual
*4 E0C62 Family Development Tool Reference Manual
*5 E0C6256 Development Tool Manual (this manual)
*6 EVA6256 Manual
*7 ICE62R (ICE6200) Hardware Manual

1.4 Production of Execution Disk

Execution files for each software development support tool and batch and parameter files for the ICE62R (ICE6200) are recorded in the DEV6256 floppy disk.
The content of the files contained in the DEV6256 floppy disk are shown below.

PC-DOS version	MS-DOS version	Contents
ASM6256.EXE	ASM6256.EXE	Cross Assembler execution file
DMS6200.EXE	DMS6200.EXE	Development Tool Management System execution file
FOG6256.EXE	FOG6256.EXE	Function Option Generator execution file
ICS6256B.BAT	ICS6256.BAT	ICE Control Software batch file
ICS6256W.EXE	ICS6256J.EXE	ICE Control Software execution file
ICS6256P.PAR	ICS6256P.PAR	ICE Control Software parameter file
MDC6256.EXE	MDC6256.EXE	Mask Data Checker execution file
SOG6256.EXE	SOG6256.EXE	Segment Option Generator 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 (DEV6256, etc.) then copy the content of the floppy disk into that subdirectory using the COPY command.

- 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 MDC6256 will handle many files.

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

- It is a good idea to copy the editor into the disk to be copied and the subdirectory so you can also select the editor from the DMS6200 menu.

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 (DEV6256), then insert the original disk into the A drive and execute the COPY command.

```
C\>MD DEV6256↵
C\>CD DEV6256↵
C\DEV6256\>COPY A:*.*↵
```

Example:

Setting of FILES (CONFIG.SYS)

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

- In "ICS6256(B).BAT" the batch process is indicated such that the ICS6256J(W).EXE is executed after the execution of the command for the setting of the RS-232C communication parameters. When first executing the ICE Control Software after resetting the host computer, select then activate this batch file from the DMS6200 menu.
  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
```

Note *The DMS6200 prepares a menu from files that are in the current directory. Consequently, be sure to arrange the above mentioned files in the same disk or the same directory.*

2 DEVELOPMENT TOOL MANAGEMENT SYSTEM DMS6200

2.1 DMS6200 Outline

The DMS6200 (Development Tool Management System) is a software which selects the DEV6256 software development support tool and the program such as an editor in menu form and starts it.
In this way the various software frequently executed during debugging can be effectively activated.

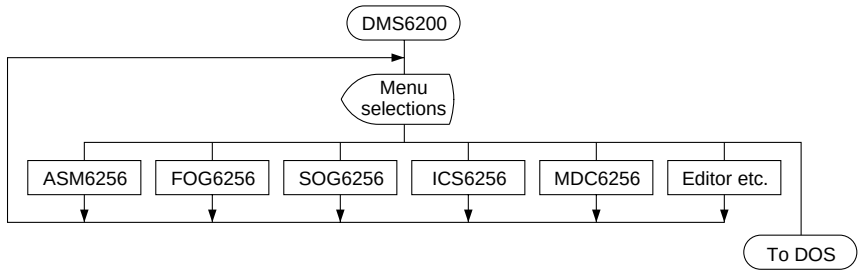


Fig. 2.1.1 DMS6200 execution flow

Refer to the "E0C62 Family Development Tool Reference Manual" for detailes of the operation.

2.2 DMS6200 Quick Reference

■ Starting command

Execution file: **DMS6200.EXE** ***Starting command:*** **DMS6200**↵ ↵ *indicates the Return key.*

■ Display examples

```
   *** E0C6200 Development tool Management System. --- Ver 1.0 ***

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.

                          STRIKE ANY KEY.
```

Start message
When DMS6200 is started, the following 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.

```
      DMS6200 Version 1.0     Copyright(C) SEIKO EPSON CORP. 1991.

1) ASM6256 .EXE
2) FOG6256 .EXE
3) ICS6256B.BAT
4) ICS6256W.EXE
5) MDC6256 .EXE
6) SOG6256 .EXE

Input Number ? [1  ]
```

Menu screen (PC-DOS Version)
A list of all executable files will appear on this menu screen.
Input the number of the development support tool you wish to start and then press the "RETURN" key. To return to DOS at this point, press the "ESC" key.

```
      DMS6200 Version 1.0     Copyright(C) SEIKO EPSON CORP. 1991.

 1) C256XXX .DAT
 2) C256XXX .PRN
 3) C256XXX .SEG
  :     :
10) C6256XXX.PA0

 Input Number ? [1  ]

   Edit > [ASM6256 C256XXX                                        ]
```

Source file selection screen
To starting ASM6256, select the source file on this screen. When the source file is selected by number, the edit line enclosed in [ ] will appear; enter the option parameter if necessary. Press the "RETURN" key when input is completed. When starting, press the "RETURN" key twice particularly for the support tools which do not require source files. To return to DOS at this point, press the "ESC" key.

3 CROSS ASSEMBLER ASM6256

3.1 ASM6256 Outline

The ASM6256 cross assembler is an assembler program for generating the machine code used by the E0C6256 4-bit, single-chip microcomputers. The Cross Assembler ASM6256 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.

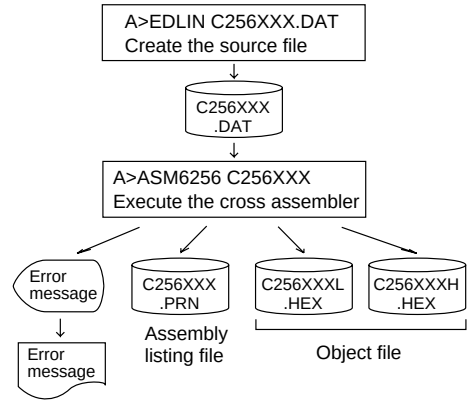


Fig. 3.1.1 ASM6256 execution flow

☞ *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.*

3.2 E0C6256 Restrictions

Note the following when generating a program by the E0C6256:

■ ROM area

The capacity of the E0C6256 ROM is 6K steps (0000H to 17FFH).
Therefore, the specification range of the memory setting pseudo-instructions and PSET instruction is restricted.

Memory configuration:

Bank 0: 16 pages (0 to 0FH), each 256 steps
Bank 1: 8 pages (0 to 07H)

Significant specification range:

ORG	pseudo-instruction:	0000H to 17FFH
BANK	pseudo-instruction:	0H and 1H
PAGE	pseudo-instruction:	Bank 0 = 00H to 0FH, Bank 1 = 00H to 07H
PSET	instruction:	Bank 0 = 00H to 0FH, Bank 1 = 00H to 07H

■ RAM area

The capacity of the E0C6256 RAM is 795 words (000H to 5FEH, 4 bits/word). (715 words when the display memory is allocated within page 1) However, note the following points when programming.

(1) The following addresses become unused area. Memory access is invalid when the unused area is specified.
- 85H–8FH, 93H, 97H–9FH, A4H–AFH, B3H, B7H–BFH, C7H, CFH, D5H–D7H, EDH–EFH, F7H and FFH in each page
- 500H–52FH
- 530H–57FH (when 130H–17FH is selected for the display memory)

(2) The display memory area can be assigned to 130H–17FH or 530H–57FH by software.
When 130H–17FH is selected: R/W, when 530H–57FH is selected: W only
If 130H–17FH is selected, RAM (80 words) is used as the display memory area.

(3) Only the subordinate 3 bits of the page section (XP, YP) of the index register which specifies address is effective. (The 1 superordinate bit is ignored.)

Example:

```
LD   A,5
LD   XP,A
LD   X,00H
```

500H 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.

3.3 ASM6256 Quick Reference

■ Starting command and input/output files

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

Execution file: ASM6256.EXE

Starting command: **ASM6256_ [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: C256XXX.DAT (Source file)

Output file: C256XXXL.HEX (Object file, low-order)
C256XXXH.HEX (Object file, high-order)
C256XXX.PRN (Assembly listing file)

■ Display example

```
          *** E0C6256 CROSS ASSEMBLER. --- Ver 2.00 ***

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 " C256XXX.DAT "

         THIS SOFTWARE MAKES NEXT FILES.

            C256XXXH.HEX  ...  HIGH BYTE OBJECT FILE.
            C256XXXL.HEX  ...  LOW BYTE OBJECT FILE.
            C256XXX .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 ASM6256 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, ASM6256 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)	• 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. • A statement exceeded a page boundary although its location was not specified.
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).

4 FUNCTION OPTION GENERATOR FOG6256

4.1 FOG6256 Outline

With the 4-bit single-chip E0C6256 microcomputers, the customer may select five hardware options. By modifying the mask patterns of the E0C6256 according to the selected options, the system can be customized to meet the specifications of the target system.
The Function Option Generator FOG6256 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 FOG6256, the E0C6256 mask pattern is automatically generated by a general purpose computer.
The HEX file for the evaluation board (EVA6256) hardware option ROM is simultaneously generated with the data file.

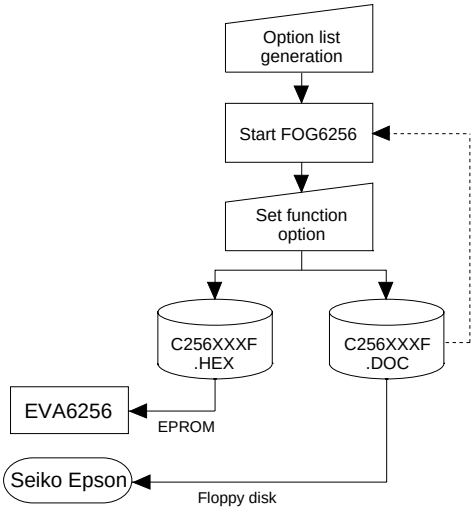


Fig. 4.1.1 FOG6256 execution flow

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

4.2 E0C6256 Option List

Multiple specifications are available in each option item as indicated in the Option List. Using "4.3 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. OSC3 SYSTEM CLOCK

☐ 1. Not Use
☐ 2. Use <CR>
☐ 3. Use <Ceramic>

2. MULTIPLE KEY ENTRY RESET (COMBINATION)

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

3. MULTIPLE KEY ENTRY RESET (TIME AUTHORIZE)

☐ 1. Not Use ☐ 2. Use

4. WATCHDOG TIMER

☐ 1. Not Use ☐ 2. Use

5. SVD CRITERIA VOLTAGE

☐ 1. 1.5V Version ☐ 2. 3.0V Version

6. INPUT PORT PULL DOWN 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
- K10 ..... ☐ 1. With Resistor ☐ 2. Gate Direct
- K11 ..... ☐ 1. With Resistor ☐ 2. Gate Direct
- K12 ..... ☐ 1. With Resistor ☐ 2. Gate Direct
- K13 ..... ☐ 1. With Resistor ☐ 2. Gate Direct

7. OUTPUT PORT OUTPUT SPECIFICATION

- R00 ..... ☐ 1. Complementary ☐ 2. Pch-OpenDrain
- R01 ..... ☐ 1. Complementary ☐ 2. Pch-OpenDrain
- R02 ..... ☐ 1. Complementary ☐ 2. Pch-OpenDrain
- R03 ..... ☐ 1. Complementary ☐ 2. Pch-OpenDrain
- R10 ..... ☐ 1. Complementary ☐ 2. Pch-OpenDrain
- R11 ..... ☐ 1. Complementary ☐ 2. Pch-OpenDrain
- R12 ..... ☐ 1. Complementary ☐ 2. Pch-OpenDrain
- R13 ..... ☐ 1. Complementary ☐ 2. Pch-OpenDrain

8. I/O PORT OUTPUT SPECIFICATION

- P00 ..... ☐ 1. Complementary ☐ 2. Pch-OpenDrain
- P01 ..... ☐ 1. Complementary ☐ 2. Pch-OpenDrain
- P02 ..... ☐ 1. Complementary ☐ 2. Pch-OpenDrain
- P03 ..... ☐ 1. Complementary ☐ 2. Pch-OpenDrain
- P10 ..... ☐ 1. Complementary ☐ 2. Pch-OpenDrain
- P11 ..... ☐ 1. Complementary ☐ 2. Pch-OpenDrain
- P12 ..... ☐ 1. Complementary ☐ 2. Pch-OpenDrain
- P13 ..... ☐ 1. Complementary ☐ 2. Pch-OpenDrain

4.3 Option Specifications and Selection Message

Screen that can be selected as function options set on the E0C6256 are shown below, and their specifications are also described.

1 OSC3 system clock

```
*** OPTION  NO.1 ***

--- OSC3 SYSTEM CLOCK ---
                          1. NOT USE
                          2. USE <CR>
                          3. USE <CERAMIC>

PLEASE SELECT NO.(1) ? 1↵

                          1. NOT USE  SELECTED
```

Select whether the OSC3 system clock will be used or not. If you use OSC3 system clock, either CR oscillation circuit or ceramic oscillation circuit can be selected. 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 resistors are needed as external components since capacities are built-in. On the other hand, when ceramic oscillation circuit is selected, ceramic oscillator, gate capacity and drain capacity are needed as external components. Although when ceramic oscillation circuit is selected, it is fixed at 1 MHz, when CR oscillation circuit is selected, frequency may be modified to a certain extent depending on the resistance of external components.

2 Multiple key entry reset (Combination)

```
*** OPTION  NO.2 ***

--- MULTIPLE KEY ENTRY RESET (COMBINATION) ---
                          1. NOT USE
                          2. USE <K00,K01,K02,K03>
                          3. USE <K00,K01,K02>
                          4. USE <K00,K01>

PLEASE SELECT NO.(1) ? 1↵

                          1. NOT USE  SELECTED
```

Select reset function by K00–K03 inputs.
When "Not Use" is selected, the reset function is not activated even if K00 through K03 are entered. If you use this reset function, select combination of K00–K03 inputs.
When "Use <K00, K01>" is set, the system is reset immediately the K00 and K01 inputs go high at the same time. The initial reset is done, even when a key entry including a combination of selected input ports is made.
The multiple key entry reset circuit is shown in Figure 4.3.1.

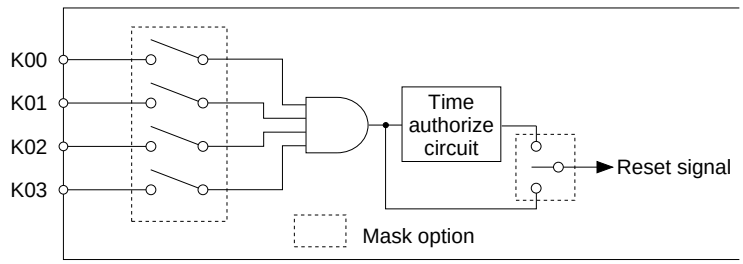


Fig. 4.3.1
Multiple key entry reset circuit

3 Multiple key entry reset (Time authorize)

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

--- MULTIPLE KEY ENTRY RESET (TIME AUTHORIZE) ---
                          1. NOT USE
                          2. USE

PLEASE SELECT NO.(1) ? 1↵

                          1. NOT USE  SELECTED
```

Select wtether the time authorize circuit will be used or not for the multiple key entry reset function.
When "Use" is selected, a simultaneous high input time is authorized. The initial reset when a signal is input for more than 1 to 2 sec. If the time authorize circuit is not used, the initial reset when a high signal is input for more than 6 msec.

* If "Not Use" is set for the OPTION NO.2 (combination), the time authorize selection is required.

4 Watchdog timer reset

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

--- WATCHDOG TIMER RESET ---
                          1. NOT USE
                          2. USE

PLEASE SELECT NO.(1) ? 2↵

                          2. USE  SELECTED
```

Select whether the watchdog timer built-in to detect CPU runaways will be used or not.
When the watchdog timer is not reset by the program within 3 to 4 second cycles, the CPU is initially reset.

5 SVD criteria voltage

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

--- SVD CRITERIA VOLTAGE ---
                          1. 1.5V VERSION
                          2. 3.0V VERSION

PLEASE SELECT NO.(1) ? 2↵

                          2. 3.0V VERSION  SELECTED
```

Select criteria voltage for SVD.
Either 1.5 V version or 3.0 V version can be selected.
Select it to match the specifications, such as batteries, to be used.

1.5 V version: 1.35/1.20/1.05 V programmable
3.0 V version: 2.60/2.45/2.30 V programmable

6 Input port pull down resistor

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

--- INPUT PORT PULL DOWN SPECIFICATION ---
        K00             1. WITH RESISTOR
                        2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1↲

        K01             1. WITH RESISTOR
                        2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1↲

        K02             1. WITH RESISTOR
                        2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1↲

        K03             1. WITH RESISTOR
                        2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1↲

        K10             1. WITH RESISTOR
                        2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1↲

        K11             1. WITH RESISTOR
                        2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1↲

        K12             1. WITH RESISTOR
                        2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1↲

        K13             1. WITH RESISTOR
                        2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1↲

        K00             1. WITH RESISTOR  SELECTED
        K01             1. WITH RESISTOR  SELECTED
        K02             1. WITH RESISTOR  SELECTED
        K03             1. WITH RESISTOR  SELECTED
        K10             1. WITH RESISTOR  SELECTED
        K11             1. WITH RESISTOR  SELECTED
        K12             1. WITH RESISTOR  SELECTED
        K13             1. WITH RESISTOR  SELECTED
```

Select whether input ports (K00–K03 and K10–K13) will each be supplemented with pull down resistors or not.
When "Gate Direct" is selected, see to it that entry floating state does not occur.
Moreover, the input port status is changed from high level (V_{DD}) to low (V_{SS}) with pull down resistors, a delay in waveform rise time will occur depending on the pull down resistor and entry load time constant. Because of this, when input reading is to be conducted, ensure the appropriate wait time with the program.
Select "With Resistor" pull down resistor for unused ports.
The configuration of the pull down resistor circuit is shown in Figure 4.3.2.

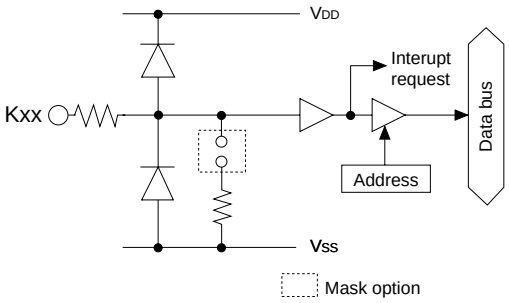


Fig. 4.3.2 Configuration of pull down resistor circuit

7 Output port output specification

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

--- OUTPUT PORT OUTPUT SPECIFICATION ---
        R00             1. COMPLEMENTARY
                        2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

        R01             1. COMPLEMENTARY
                        2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

        R02             1. COMPLEMENTARY
                        2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

        R03             1. COMPLEMENTARY
                        2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

        R10             1. COMPLEMENTARY
                        2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

        R11             1. COMPLEMENTARY
                        2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

        R12             1. COMPLEMENTARY
                        2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

        R13             1. COMPLEMENTARY
                        2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

        R00             1. COMPLEMENTARY  SELECTED
        R01             1. COMPLEMENTARY  SELECTED
        R02             1. COMPLEMENTARY  SELECTED
        R03             1. COMPLEMENTARY  SELECTED
        R10             1. COMPLEMENTARY  SELECTED
        R11             1. COMPLEMENTARY  SELECTED
        R12             1. COMPLEMENTARY  SELECTED
        R13             1. COMPLEMENTARY  SELECTED
```

Select the output specification of each output port (R00–R03 and R10–R13).
Either complementary output or Pch open drain output can be selected.
When output port is to be used on key matrix configuration, select Pch open drain output.
For unused output ports, select complementary output.
The output port circuit configuration is shown in Figure 4.3.3.

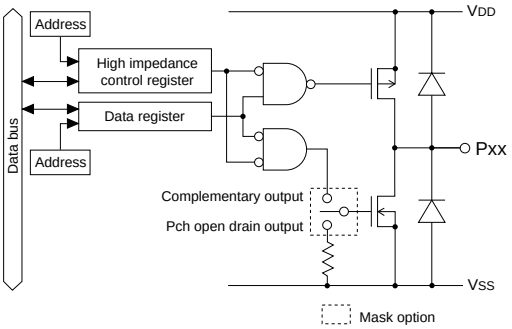


Fig. 4.3.3 Circuit configuration of output port

8 I/O port output specification

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

--- I/O PORT OUTPUT SPECIFICATION ---
        P00             1. COMPLEMENTARY
                        2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

        P01             1. COMPLEMENTARY
                        2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

        P02             1. COMPLEMENTARY
                        2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

        P03             1. COMPLEMENTARY
                        2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

        P10             1. COMPLEMENTARY
                        2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

        P11             1. COMPLEMENTARY
                        2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

        P12             1. COMPLEMENTARY
                        2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

        P13             1. COMPLEMENTARY
                        2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

        P00             1. COMPLEMENTARY  SELECTED
        P01             1. COMPLEMENTARY  SELECTED
        P02             1. COMPLEMENTARY  SELECTED
        P03             1. COMPLEMENTARY  SELECTED
        P10             1. COMPLEMENTARY  SELECTED
        P11             1. COMPLEMENTARY  SELECTED
        P12             1. COMPLEMENTARY  SELECTED
        P13             1. COMPLEMENTARY  SELECTED
```

Select the output specification of each I/O port (P00–P03 and P10–P13) when output mode is set. Either complementary output or Pch open drain output can be selected.
Select complementary output for unused ports.

The I/O ports can control the input/output direction according to the IOC00–IOC03 and IOC10–IOC13 registers (addresses B0H and B4H); at "1" and "0" settings, it is set to output port and input port, respectively.

When the serial interface function is selected, the output specification of the terminals SOUT, SCLK (during the master mode) and SRDY (during the slave mode) that is used as output in the input/output port of the serial interface is respectively selected by mask options of P11, P12 and P13. Select complementary output for the SIN (P10) output specification.
The output circuit configuration of an I/O port is shown in Figure 4.3.4.

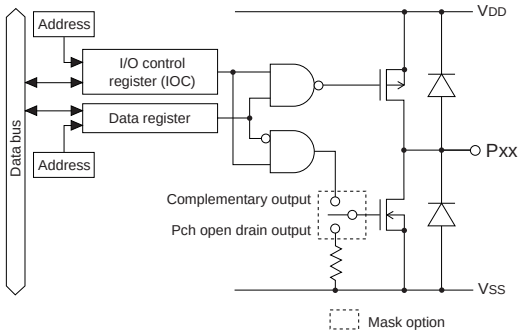


Fig. 4.3.4 Output circuit configuration of I/O port

4.4 FOG6256 Quick Reference

■ Starting command and input/output files

Execution file: FOG6256.EXE

Starting command: **FOG6256↵** ↵ *indicates the Return key.*

Input file: C256XXXF.DOC (Function option document file, when modifying)

Output file: C256XXXF.DOC (Function option document file)
C256XXXF.HEX (Function option HEX file)

■ Display example

```
     ***  E0C6256 FUNCTION OPTION GENERATOR. --- Ver 3.13  ***

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 1992  SEIKO EPSON CORP.

        THIS SOFTWARE MAKES NEXT FILES.

           C256XXXF.HEX  ...  FUNCTION OPTION HEX FILE.
           C256XXXF.DOC  ...  FUNCTION OPTION DOCUMENT FILE.

                        STRIKE ANY KEY.
```

Start-up message
When FOG6256 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.

```
*** E0C6256 USER'S OPTION SETTING. --- Ver 3.13 ***

CURRENT DATE IS  92/12/01
PLEASE INPUT NEW DATE  :   92/12/03↵
```

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 FOG6256.

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

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

PLEASE SELECT NO.? 1↵
PLEASE INPUT FILE NAME? C2560A0↵                              ..(1)
PLEASE INPUT USER'S NAME? SEIKO EPSON CORP.↵                  ..(2)
PLEASE INPUT ANY COMMENT
(ONE LINE IS 50 CHR)? ED MARKETING DEPARTMENT↵                ..(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? C2560A0↵
EXISTS OVERWRITE(Y/N)? N↵
PLEASE INPUT FILE NAME? C2560B0↵
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) ***

C2560A0         C2560B0         C2560C0                 ..(1)

PLEASE INPUT FILE NAME? C2560A0↵                        ..(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? C2560N0↵
FUNCTION OPTION DOCUMENT FILE IS NOT FOUND.
PLEASE INPUT FILE NAME?
```

In addition, if specified file format is different (such as document file for the other model), the following message is displayed and FOG6256 is terminated.

```
BAD FUNCTION OPTION DOCUMENT FILE.
```

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.

```
*** OPTION  NO.1 ***

--- OSC3 SYSTEM CLOCK ---
                            1. NOT USE
                            2. USE <CR>
                            3. USE <CERAMIC>

PLEASE SELECT  NO.(1) ? 1↵

                            1. NOT USE  SELECTED
```

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.

```
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.?
```

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 EVA6256, 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 EVA6256 options.

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

4.5 Sample File

■ Example of function option document file

```
* E0C6256 FUNCTION OPTION DOCUMENT  V 3.13
*
* FILE NAME    C2560A0F.DOC
* USER'S NAME  SEIKO EPSON CORP.
* INPUT DATE   92/12/01
*
* COMMENT      ED MARKETING DEPARTMENT
*              421-8 HINO HINO-SHI TOKYO 191-8501 JAPAN
*              TEL 042-587-5816
*              FAX 042-587-5624
*
*
* OPTION NO.1
* < OSC3 SYSTEM CLOCK >
*                            NOT USE  ----------------------  SELECTED
 OPT0101 01
*
* OPTION NO.2
* < MULTIPLE KEY ENTRY RESET (COMBINATION) >
*                            NOT USE  ----------------------  SELECTED
 OPT0201 01
*
* OPTION NO.3
* < MULTIPLE KEY ENTRY RESET (TIME AUTHORIZE) >
*                            NOT USE  ----------------------  SELECTED
 OPT0301 01
*
* OPTION NO.4
* < WATCHDOG TIMER RESET >
*                            USE  --------------------------  SELECTED
 OPT0401 02
*
* OPTION NO.5
* < SVD CRITERIA VOLTAGE >
*                            3.0V VERSION  -----------------  SELECTED
 OPT0501 02
*
* OPTION NO.6
* < INPUT PORT PULL DOWN RESISTOR >
*     K00                    WITH RESISTOR  ----------------  SELECTED
*     K01                    WITH RESISTOR  ----------------  SELECTED
*     K02                    WITH RESISTOR  ----------------  SELECTED
*     K03                    WITH RESISTOR  ----------------  SELECTED
*     K10                    WITH RESISTOR  ----------------  SELECTED
*     K11                    WITH RESISTOR  ----------------  SELECTED
*     K12                    WITH RESISTOR  ----------------  SELECTED
*     K13                    WITH RESISTOR  ----------------  SELECTED
 OPT0601 01
 OPT0602 01
 OPT0603 01
 OPT0604 01
 OPT0605 01
 OPT0606 01
 OPT0607 01
 OPT0608 01
*
```

```
* OPTION NO.7
* < OUTPUT PORT OUTPUT SPECIFICATION >
*     R00                    COMPLEMENTARY  -----------------  SELECTED
*     R01                    COMPLEMENTARY  -----------------  SELECTED
*     R02                    COMPLEMENTARY  -----------------  SELECTED
*     R03                    COMPLEMENTARY  -----------------  SELECTED
*     R10                    COMPLEMENTARY  -----------------  SELECTED
*     R11                    COMPLEMENTARY  -----------------  SELECTED
*     R12                    COMPLEMENTARY  -----------------  SELECTED
*     R13                    COMPLEMENTARY  -----------------  SELECTED
 OPT0701 01
 OPT0702 01
 OPT0703 01
 OPT0704 01
 OPT0705 01
 OPT0706 01
 OPT0707 01
 OPT0708 01
*
* OPTION NO.8
* < I/O PORT OUTPUT SPECIFICATION >
*     P00                    COMPLEMENTARY  -----------------  SELECTED
*     P01                    COMPLEMENTARY  -----------------  SELECTED
*     P02                    COMPLEMENTARY  -----------------  SELECTED
*     P03                    COMPLEMENTARY  -----------------  SELECTED
*     P10                    COMPLEMENTARY  -----------------  SELECTED
*     P11                    COMPLEMENTARY  -----------------  SELECTED
*     P12                    COMPLEMENTARY  -----------------  SELECTED
*     P13                    COMPLEMENTARY  -----------------  SELECTED
 OPT0801 01
 OPT0802 01
 OPT0803 01
 OPT0804 01
 OPT0805 01
 OPT0806 01
 OPT0807 01
 OPT0808 01
*
*
*
* SEIKO EPSON'S AREA
*
*
*
* OPTION NO.9
 OPT0901 01
*
* OPTION NO.10
 OPT1001 01
*
* OPTION NO.11
 OPT1101 01
*
* OPTION NO.12
 OPT1201 01
*
* OPTION NO.13
 OPT1301 01
\\END
```

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

5 SEGMENT OPTION GENERATOR SOG6256

5.1 SOG6256 Outline

With the 4-bit single-chip E0C6256 microcomputers, the customer may select the LCD segment options. By modifying the mask patterns of the E0C6256 according to the selected options, the system can be customized to meet the specifications of the target system.
The Segment Option Generator SOG6256 is a software tool for generating data file used to generate mask patterns. From the data file created with SOG6256, the E0C6256 mask pattern is automatically generated by a general purpose computer.
The HEX file for the evaluation board (EVA6256) segment option ROM is simultaneously generated with the data file.

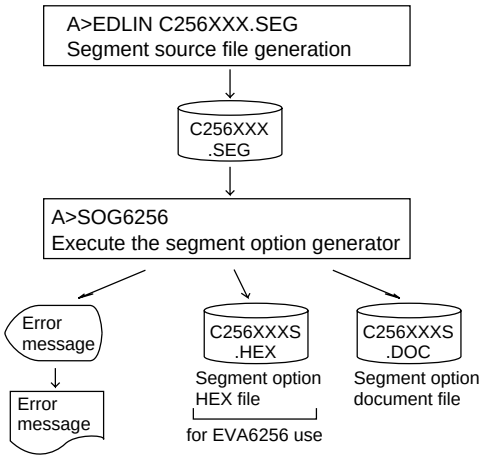


Fig. 5.1.1 SOG6256 execution flow

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

5.2 Option List

TERMINAL NAME	ADDRESS															OUTPUT SPECIFICATION	
	COM0			COM1			COM2			COM3			COM4				
	H	L	D	H	L	D	H	L	D	H	L	D	H	L	D		
SEG0																SEG output ☐	DC output ☐C ☐P
SEG1																SEG output ☐	DC output ☐C ☐P
SEG2																SEG output ☐	DC output ☐C ☐P
SEG3																SEG output ☐	DC output ☐C ☐P
SEG4																SEG output ☐	DC output ☐C ☐P
SEG5																SEG output ☐	DC output ☐C ☐P
SEG6																SEG output ☐	DC output ☐C ☐P
SEG7																SEG output ☐	DC output ☐C ☐P
SEG8																SEG output ☐	DC output ☐C ☐P
SEG9																SEG output ☐	DC output ☐C ☐P
SEG10																SEG output ☐	DC output ☐C ☐P
SEG11																SEG output ☐	DC output ☐C ☐P
SEG12																SEG output ☐	DC output ☐C ☐P
SEG13																SEG output ☐	DC output ☐C ☐P
SEG14																SEG output ☐	DC output ☐C ☐P
SEG15																SEG output ☐	DC output ☐C ☐P
SEG16																SEG output ☐	DC output ☐C ☐P
SEG17																SEG output ☐	DC output ☐C ☐P
SEG18																SEG output ☐	DC output ☐C ☐P
SEG19																SEG output ☐	DC output ☐C ☐P
SEG20																SEG output ☐	DC output ☐C ☐P
SEG21																SEG output ☐	DC output ☐C ☐P
SEG22																SEG output ☐	DC output ☐C ☐P
SEG23																SEG output ☐	DC output ☐C ☐P
SEG24																SEG output ☐	DC output ☐C ☐P
SEG25																SEG output ☐	DC output ☐C ☐P
SEG26																SEG output ☐	DC output ☐C ☐P
SEG27																SEG output ☐	DC output ☐C ☐P
SEG28																SEG output ☐	DC output ☐C ☐P
SEG29																SEG output ☐	DC output ☐C ☐P
SEG30																SEG output ☐	DC output ☐C ☐P
SEG31																SEG output ☐	DC output ☐C ☐P
SEG32																SEG output ☐	DC output ☐C ☐P
SEG33																SEG output ☐	DC output ☐C ☐P
SEG34																SEG output ☐	DC output ☐C ☐P
SEG35																SEG output ☐	DC output ☐C ☐P
SEG36																SEG output ☐	DC output ☐C ☐P
SEG37																SEG output ☐	DC output ☐C ☐P
SEG38																SEG output ☐	DC output ☐C ☐P
SEG39																SEG output ☐	DC output ☐C ☐P
SEG40																SEG output ☐	DC output ☐C ☐P
SEG41																SEG output ☐	DC output ☐C ☐P
SEG42																SEG output ☐	DC output ☐C ☐P
SEG43																SEG output ☐	DC output ☐C ☐P
SEG44																SEG output ☐	DC output ☐C ☐P
SEG45																SEG output ☐	DC output ☐C ☐P
SEG46																SEG output ☐	DC output ☐C ☐P
SEG47																SEG output ☐	DC output ☐C ☐P
SEG48																SEG output ☐	DC output ☐C ☐P
SEG49																SEG output ☐	DC output ☐C ☐P
SEG50																SEG output ☐	DC output ☐C ☐P
SEG51																SEG output ☐	DC output ☐C ☐P
SEG52																SEG output ☐	DC output ☐C ☐P
SEG53																SEG output ☐	DC output ☐C ☐P
SEG54																SEG output ☐	DC output ☐C ☐P
SEG55																SEG output ☐	DC output ☐C ☐P
SEG56																SEG output ☐	DC output ☐C ☐P
SEG57																SEG output ☐	DC output ☐C ☐P
SEG58																SEG output ☐	DC output ☐C ☐P
SEG59																SEG output ☐	DC output ☐C ☐P

Legend: <ADDRESS> H: High order address, L: Low order address D: Data bit <OUTPUT SPECIFICATION> C: Complementary output P: Pch open drain output

Note 1. Even if there are unused areas, set "---" (hyphens) such that there are no blank columns.
2. When DC output is selected, the display memory of the COM0 column becomes effective.

5.3 Segment Ports Output Specifications

For the output specification of the segment output ports SEG0–SEG59, segment output and DC output can be selected in unit of one terminal. When used for liquid crystal panel drives, select segment output; when used as regular output port, select DC output. When DC output is selected, either complementary output or Pch open drain output may further be selected.
However, for segment output ports that will not be used, select segment output.
Refer to the "E0C62 Family Development Tool Reference Manual" for the segment option source file creation.

■ When segment output is selected

The segment output port has a segment decoder built-in, and the data bit of the optional address in the display memory area (130H–17FH or 530H–57FH) can be allocated to the optional segment. With this, up to 300 segments (240, 180 or 120 segments when 1/4, 1/3 or 1/2 duty is selected, respectively) of liquid crystal panel could be driven.
The display memory may be allocated only one segment and multiple setting is not possible.
The allocated segment displays when the bit for this display memory is set to "1", and goes out when bit is set to "0".
Segment allocation is set to H for high address (3–7), to L for low address (0–F), and to D for data bit (0–3) and are recorded in their respective column in the option list. For segment ports that will not be used, write "---" (hyphen) in the H, L, and D columns of COM0–COM4.

Examples

- When 1/5 duty is selected

```
0  300  301  302  303  310  S
1  311  312  313  320  321  S
```

- When 1/3 duty is selected

```
0  300  301  302  ---  ---  S
1  311  312  313  ---  ---  S
```

■ When DC output is selected

The DC output can be selected for each terminal and up to 60 terminals may be allocated for DC output. Also, either complementary output or Pch open drain output is likewise selected for each terminal. When the bit for the selected display memory is set to "1", the segment output port goes high (V_{DD}), and goes low (V_{SS}) when set to "0". Segment allocation is the same as when segment output is selected but for the while the display memory allocated to COM1–COM4 becomes ineffective. Write three hyphens ("---") in the COM1–COM4 columns in the option list.

Example

- When complementary output is set to SEG56 and SEG57, and Pch open drain output is set to SEG58 and SEG59.

```
56  700  ---  ---  ---  ---  C
57  710  ---  ---  ---  ---  C
58  720  ---  ---  ---  ---  P
59  730  ---  ---  ---  ---  P
```

5.4 SOG6256 Quick Reference

■ Starting command and input/output files

Execution file: SOG6256.EXE

Starting command: **SOG6256_ [-H]↵**

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

Option: -H: Specifies the segment option document file for input file of SOG6256.

Input file: C256XXX.SEG (Segment option source file)
C256XXXS.DOC (Segment option document file, when -H option use)

Output file: C256XXXS.DOC (Segment option document file)
C256XXXS.HEX (Segment option HEX file)

■ Display example

```
     ***  E0C6256 SEGMENT OPTION GENERATOR. --- Ver 3.21  ***

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.

         SEGMENT OPTION SOURCE FILE NAME IS " C256XXX.SEG "

         THIS SOFTWARE MAKES NEXT FILES.

            C256XXXS.HEX  ...  SEGMENT OPTION HEX FILE.
            C256XXXS.DOC  ...  SEGMENT OPTION DOCUMENT FILE.

                        STRIKE ANY KEY.
```

Start-up message
When SOG6256 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.

```
*** E0C6256 USER'S OPTION SETTING. --- Ver 3.21 ***

CURRENT DATE IS  92/12/01
PLEASE INPUT NEW DATE  :   92/12/03↵
```

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.

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

C2560A0         C2560B0         C2560C0                          . . (1)

PLEASE INPUT SEGMENT OPTION FILE NAME? C2560A0↵                  . . (2)
PLEASE INPUT USER'S NAME? SEIKO EPSON CORP.↵                     . . (3)
PLEASE INPUT ANY COMMENT
(ONE LINE IS 50 CHR)? ED MARKETING DEPARTMENT↵                   . . (4)
                    ? 421-8 HINO HINO-SHI TOKYO 191-8501 JAPAN↵
                    ? TEL 042-587-5816↵
                    ? FAX 042-587-5624↵
                    ? ↵
```

Input file selection
(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.
(Within 50 characters x 10 lines)
Then, move to the confirmation procedure for HEX file generation.

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

SEGMENT OPTION SOURCE FILE IS NOT FOUND.          . . (5)  -H option not use

*** SOURCE FILE(S) ***

SEGMENT OPTION DOCUMENT FILE IS NOT FOUND.        . . (6)  -H option use
```

In step (1), if no modifiable source exists, an error message (5) or (6) will be displayed and the program will be terminated.

```
PLEASE INPUT SEGMENT OPTION SOURCE FILE NAME? C2560N0↵
SEGMENT OPTION SOURCE FILE IS NOT FOUND.          . . (7)  -H option not use

PLEASE INPUT SEGMENT OPTION DOCUMENT FILE NAME? C2560N0↵
SEGMENT OPTION DOCUMENT FILE IS NOT FOUND.        . . (8)  -H option use
```

In step (2), if the specified file name is not found in the current drive, an error message (7) or (8) is displayed, prompting entry of other file name.

```
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 IS COMPLETED.
```

EPROM selection

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

(1) When debugging the program with EVA6256, 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 EVA6256 options.

When a series of operations are complete, the SOG6256 generates files. If no error is committed while setting segment options, "MAKING FILE IS COMPLETED" will be displayed and the SOG6256 program will be terminated.

■ Error messages

Error message	Explanation
S (Syntax Error)	The data was written in an invalid format.
N (Segment No. Select Error)	The segment number outside the specificable range was specified.
R (RAM Address Select Error)	The segment memory address or data bit outside the specificable range was specified.
D (Duprication Error)	The same data (SEG port No., segment memory address, or data bit) was specified more then once.
Out Port Set Error	The output specifications were not set in units of two ports.

5.5 Sample Files

■ Example of segment option source file

```
; C2560A0.SEG, VER.3.21
; EVA6256 LCD SEGMENT DECODE TABLE
;
 0  300  301  302  303  310  S
 1  311  312  313  320  321  S
 2  322  323  330  331  332  S
 3  333  340  341  342  343  S
 4  350  351  352  353  360  S
 5  361  362  363  370  371  S
 6  372  373  380  381  382  S
 7  383  390  391  392  393  S
 8  3A0  3A1  3A2  3A3  3B0  S
 9  3B1  3B2  3B3  3C0  3C1  S
10  3C2  3C3  3D0  3D1  3D2  S
11  3D3  3E0  3E1  3E2  3E3  S
12  3F0  3F1  3F2  3F3  400  S
13  401  402  403  410  411  S
14  412  413  420  421  422  S
15  423  430  431  432  433  S
16  440  441  442  443  450  S
17  451  452  453  460  461  S
18  462  463  470  471  472  S
19  473  480  481  482  483  S
20  490  491  492  493  4A0  S
21  4A1  4A2  4A3  4B0  4B1  S
22  4B2  4B3  4C0  4C1  4C2  S
23  4C3  4D0  4D1  4D2  4D3  S
24  4E0  4E1  4E2  4E3  4F0  S
25  4F1  4F2  4F3  500  501  S
26  502  503  510  511  512  S
27  513  520  521  522  523  S
28  530  531  532  533  540  S
29  541  542  543  550  551  S
30  552  553  560  561  562  S
31  563  570  571  572  573  S
32  580  581  582  583  590  S
33  591  592  593  5A0  5A1  S
34  5A2  5A3  5B0  5B1  5B2  S
35  5B3  5C0  5C1  5C2  5C3  S
36  5D0  5D1  5D2  5D3  5E0  S
37  5E1  5E2  5E3  5F0  5F1  S
38  5F2  5F3  600  601  602  S
39  603  610  611  612  613  S
40  620  621  622  623  630  S
41  631  632  633  640  641  S
42  642  643  650  651  652  S
43  653  660  661  662  663  S
44  670  671  672  673  680  S
45  681  682  683  690  691  S
46  692  693  6A0  6A1  6A2  S
47  6A3  6B0  6B1  6B2  6B3  S
48  6C0  6C1  6C2  6C3  6D0  S
49  6D1  6D2  6D3  6E0  6E1  S
50  6E2  6E3  6F0  6F1  6F2  S
51  6F3  700  701  702  703  S
52  710  711  712  713  720  S
53  721  722  723  730  731  S
54  732  733  740  741  742  S
55  743  750  751  752  753  S
56  760  761  762  763  770  S
57  771  772  773  780  781  S
58  782  783  790  791  792  S
59  793  7A0  7A1  7A2  7A3  S
\\END
```

■ Example of segment option source file

```
* E0C6256 SEGMENT OPTION DOCUMENT  V 3.21
*
* FILE NAME    C2560A0S.DOC
* USER'S NAME  SEIKO EPSON CORP.
* INPUT DATE   92/12/03
* COMMENT      ED MARKETING DEPARTMENT
*              421-8 HINO HINO-SHI TOKYO 191-8501 JAPAN
*              TEL 042-587-5816
*              FAX 042-587-5624
*
*
* OPTION NO.19
*
* < LCD SEGMENT DECODE TABLE >
*
* SEG COM0 COM1 COM2 COM3 COM4 SPEC
*
   0  300  301  302  303  310  S
   1  311  312  313  320  321  S
   2  322  323  330  331  332  S
   3  333  340  341  342  343  S
   4  350  351  352  353  360  S
   5  361  362  363  370  371  S
   6  372  373  380  381  382  S
   7  383  390  391  392  393  S
   8  3A0  3A1  3A2  3A3  3B0  S
   9  3B1  3B2  3B3  3C0  3C1  S
  10  3C2  3C3  3D0  3D1  3D2  S
  11  3D3  3E0  3E1  3E2  3E3  S
  12  3F0  3F1  3F2  3F3  400  S
  13  401  402  403  410  411  S
  14  412  413  420  421  422  S
  15  423  430  431  432  433  S
  16  440  441  442  443  450  S
  17  451  452  453  460  461  S
  18  462  463  470  471  472  S
  19  473  480  481  482  483  S
  20  490  491  492  493  4A0  S
  21  4A1  4A2  4A3  4B0  4B1  S
  22  4B2  4B3  4C0  4C1  4C2  S
  23  4C3  4D0  4D1  4D2  4D3  S
  24  4E0  4E1  4E2  4E3  4F0  S
  25  4F1  4F2  4F3  500  501  S
  26  502  503  510  511  512  S
  27  513  520  521  522  523  S
  28  530  531  532  533  540  S
  29  541  542  543  550  551  S
  30  552  553  560  561  562  S
  31  563  570  571  572  573  S
  32  580  581  582  583  590  S
  33  591  592  593  5A0  5A1  S
  34  5A2  5A3  5B0  5B1  5B2  S
  35  5B3  5C0  5C1  5C2  5C3  S
  36  5D0  5D1  5D2  5D3  5E0  S
  37  5E1  5E2  5E3  5F0  5F1  S
  38  5F2  5F3  600  601  602  S
  39  603  610  611  612  613  S
  40  620  621  622  623  630  S
  41  631  632  633  640  641  S
  42  642  643  650  651  652  S
  43  653  660  661  662  663  S
  44  670  671  672  673  680  S
  45  681  682  683  690  691  S
  46  692  693  6A0  6A1  6A2  S
  47  6A3  6B0  6B1  6B2  6B3  S
  48  6C0  6C1  6C2  6C3  6D0  S
  49  6D1  6D2  6D3  6E0  6E1  S
  50  6E2  6E3  6F0  6F1  6F2  S
  51  6F3  700  701  702  703  S
  52  710  711  712  713  720  S
  53  721  722  723  730  731  S
  54  732  733  740  741  742  S
  55  743  750  751  752  753  S
  56  760  761  762  763  770  S
  57  771  772  773  780  781  S
  58  782  783  790  791  792  S
  59  793  7A0  7A1  7A2  7A3  S
\\END
```

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

6 ICE CONTROL SOFTWARE ICS6256

6.1 ICS6256 Outline

The In-circuit Emulator ICE62R (ICE6200) connects the target board produced by the user via the EVA6256 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 ICE62R (ICE6200) control is done through the ICE Control Software ICS6256.
The ICS6256 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.

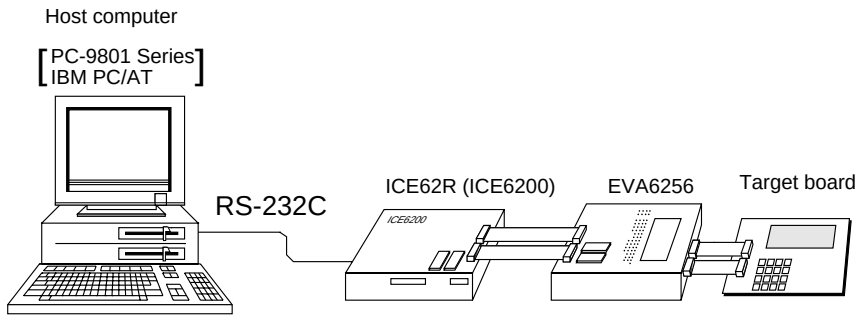


Fig. 6.1.1 Debugging system using ICE62R (ICE6200)

☞ *The functions of the ICE62R (ICE6200) and commands are same as for the E0C62 Family. Refer to the "E0C62 Family Development Tool Reference Manual" for details.*

6.2 ICS6256 Restrictions

Take the following precautions when using the ICS6256.

■ ROM Area

The ROM area is limited to a maximum address of 17FFH. Assigning data above the 17FFH address causes an error.

■ RAM Area

The RAM area is limited to a maximum address of 5FEH. However, as the following addresses are in the unused area, designation of this area with the ICE commands produces an error.

- 85H–8FH, 93H, 97H–9FH, A4H–AFH, B3H, B7H–BFH, C7H, CFH, D5H–D7H, EDH–EFH, F7H and FFH in each page
- 500H–52FH
- 530H–57FH (when 130H–17FH is selected for the display memory)

(Refer to the "E0C6256 Technical Manual" for details.)

■ OPTLD Command

In the ICS6256, OPTLD command can be used.
This command is used to load HEX files (function option and segment option data for LCD) in the EVA6256 memory with the ICE62R (ICE6200).

Load of function option data: `#OPTLD,1,C256XXX↵`
Load of segment option data: `#OPTLD,2,C256XXX↵`

OPTLD *READ HEXA DATA FILE*

Format

#OPTLD,1,<*file name*>↵ ...(1)
#OPTLD,2,<*file name*>↵ ...(2)

Function

(1) Load function option HEX file in the EVA6256 function option data memory.
It is HEX file output by the function option generator and has intel HEX format.

(2) Load segment option HEX file in the EVA6256 segment option data memory.
It is HEX file output by the segment option generator and has intel HEX format.
Since it takes about 11 minutes to load segment option HEX data, when you want to load at high speed, execute this command by changing the EVA6256 operation clock from OSC1 to OSC3. (When OSC3 = 1 MHz, since it takes about 6 minutes to load segment option HEX data.)

* Since function option HEX file cannot be loaded in OSC3 clock operation, you should not change the operation clock.

Examples

```
#OPTLD,1,C256XXX↵ ..... C256XXXF.HEX file is loaded in the function option data memory.
#OPTLD,2,C256XXX↵ ..... C256XXXS.HEX file is loaded in the segment option data memory.
#SD,DF↵
 DF |0:1↵              ..... The OSC3 oscillation is turned ON.
 E0 |0:/↵
#SD,DF↵
 DF |1:3↵              ..... Switching from OSC1 to OSC3.
 E0 |0:/↵
#I↵                    ..... The CPU is reset.
                              (Switches CPU clock to OSC1 when OSC3 oscillation is set.)
```

6.3 ICS6256 Quick Reference

■ Starting command and input/output files

↵ *indicates the Return key.*

Execution file:
ICS6256.BAT (ICS6256J.EXE) . . . for MS-DOS
ICS6256B.BAT (ICS6256W.EXE) . . . for PC-DOS

Starting command:
ICS6256 (ICS6256J)↵ . . . for MS-DOS
ICS6256B (ICS6256W)↵ . . . for PC-DOS

Input file:
C256XXXL.HEX (Object file, low-order)
C256XXXH.HEX (Object file, high-order)
C256XXXD.HEX (Data RAM file)
C256XXXC.HEX (Control file)
C256XXXF.HEX (Function option HEX file)
C256XXXS.HEX (Segment option HEX file)

Output file:
C256XXXL.HEX (Object file, low-order)
C256XXXH.HEX (Object file, high-order)
C256XXXD.HEX (Data RAM file)
C256XXXC.HEX (Control file)

■ Display example

```
          *** E0C6256 ICE CONTROL SOFTWARE. --- Ver 3.01 ***

 EEEEEEEEEE   PPPPPPPP        SSSSSSS       00000000     NNN     NNN
 EEEEEEEEEE   PPPPPPPPPP     SSS    SSSS    000    000    NNNN    NNN
 EEE          PPP     PPP   SSS     SSS   000      000   NNNNN   NNN
 EEE          PPP     PPP    SSS          000      000   NNNNNN  NNN
 EEEEEEEEEE   PPPPPPPPPP      SSSSSS      000      000   NNN NNN NNN
 EEEEEEEEEE   PPPPPPPP           SSSS     000      000   NNN  NNNNNN
 EEE          PPP                SSS    000      000   NNN   NNNNN
 EEE          PPP           SSS     SSS  000      000   NNN    NNNN
 EEEEEEEEEE   PPP           SSSS   SSS     000    000   NNN     NNN
 EEEEEEEEEE   PPP             SSSSSSS       00000000    NNN      NN

              (C) COPYRIGHT 1991 SEIKO EPSON CORP.

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

Start-up message
When ICS6256 is started, the start-up message is displayed, and a self-test is automatically performed. ICS6256 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 ICS6256 program is terminated by entering the Q (Quit) command.

Note *Confirm that the cables connected properly, then operate the ICS6256.*

■ Error messages

Error message	Meaning	Recover procedure
* COMMUNICATION ERROR OR ICE NOT READY *	ICE62R (ICE6200) is disconnected or power is OFF.	Switch OFF the host power supply, connect cable, and reapply power. Or switch ON power to ICE62R (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 ICE62R (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 ICE62R (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 ICE62R (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.

■ Command list

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 ↵ #BAR,a ↵	Sets Break at program address "a" Breakpoint is canceled
		#BD ↵ #BDR ↵	Break condition is set for data RAM Breakpoint is canceled
		#BR ↵ #BRR ↵	Break condition is set for EVA6256 CPU internal registers Breakpoint is canceled
		#BM ↵ #BMR ↵	Combined break conditions set for program data RAM address and registers 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 EVA6256 CPU internal registers
		#SR ↵	Set EVA6256 CPU internal registers
		#I ↵	Reset EVA6256 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 ICE62R (ICE6200) set condition from file
		#CS,file ↵	Save ICE62R (ICE6200) set condition to file
		#OPTLD,1,file ↵ #OPTLD,2,file ↵	Load function option data from file Load segment option data from 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 ICE62R (ICE6200) instruction
17	Self Diagnosis	#CHK ↵	Report results of ICE62R (ICE6200) self diagnostic test

↵ means press the RETURN key.

7 MASK DATA CHECKER MDC6256

7.1 MDC6256 Outline

The Mask Data Checker MDC6256 is a software tool which checks the program data (C256XXXH.HEX and C256XXXL.HEX) and option data (C256XXXF.DOC and C256XXXS.DOC) created by the user and creates the data file (C6256XXX.PAn) for generating mask patterns. The user must send the file generated through this software tool to Seiko Epson.

Moreover, MDC6256 has the capability to restore the generated data file (C6256XXX.PA0) to the original file format.

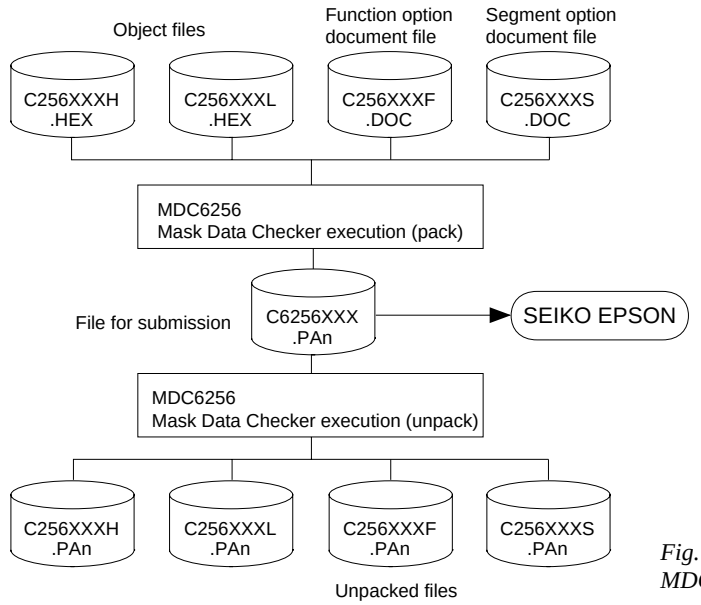


Fig. 7.1.1 MDC6256 execution flow

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

7.2 MDC6256 Quick Reference

■ Starting command and input/output files

Execution file: MDC6256.EXE

Starting command: **MDC6256**↵ ↵ *indicates the Return key.*

Input file:

C256XXXL.HEX	(Object file, low-order)	When packing
C256XXXH.HEX	(Object file, high-order)	
C256XXXF.DOC	(Function option document file)	
C256XXXS.DOC	(Segment option document file)	
C6256XXX.PAn	(Packed file)	When unpacking

Output file:

C6256XXX.PAn	(Packed file)	When packing
C256XXXL.PAn	(Object file, low-order)	When unpacking
C256XXXH.PAn	(Object file, high-order)	
C256XXXF.PAn	(Function option document file)	
C256XXXS.PAn	(Segment option document file)	

■ Display examples

```
          *** E0C6256 PACK / UNPACK PROGRAM Ver 2.000 ***

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 MDC6256 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)

       C256XXXH.HEX --------+
                            |
       C256XXXL.HEX --------+
                            |-------- C256XXX.PAn (PACK FILE)
       C256XXXF.DOC --------+
                            |
       C256XXXS.DOC --------+

PLEASE INPUT PACK FILE NAME (C6256XXX.PAn) ?  C62560A0.PA0↵    ...(2)

       C2560A0H.HEX --------+
                            |
       C2560A0L.HEX --------+
                            |-------- C2560A0.PA0
       C2560A0F.DOC --------+
                            |
       C2560A0S.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 "C6256XXX.PA0" has been submitted, the pack file name should be entered as "C6256XXX.PA1".)

With this, the mask file (C6256XXX.PAn) is generated, and the MDC6256 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 (ASM6256) 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 (C6256XXX.PAn) ?  C62560A0.PA0↵  ...(2)

                            +-------- C2560A0H.PA0
                            |
                            +-------- C2560A0L.PA0
       C62560A0.PA0 --------|
                            +-------- C2560A0F.PA0
                            |
                            +-------- C2560A0S.PA0
```

Unpacking of data

(1) Select "2. UNPACK" in the operation menu.

(2) Enter the packed file name.

With this, the mask data file (C6256XXX.PAn) is restored to the original file format, and the MDC6256 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).*

Segment option data error

Error Message	Explanation
1. SEGMENT DATA ERROR : START MARK.	The start mark is not "\SEGMENT". (during unpacking) *
2. SEGMENT DATA ERROR : DATA.	The segment data is not correct.
3. SEGMENT DATA ERROR : SEGMENT NUMBER.	The SEG No. is not correct.
4. SEGMENT DATA ERROR : SPEC.	The output specification of the SEG terminal is not correct.
5. SEGMENT 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	FOG6256, SOG6256 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. E0C6256 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← $\bar{r}$

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

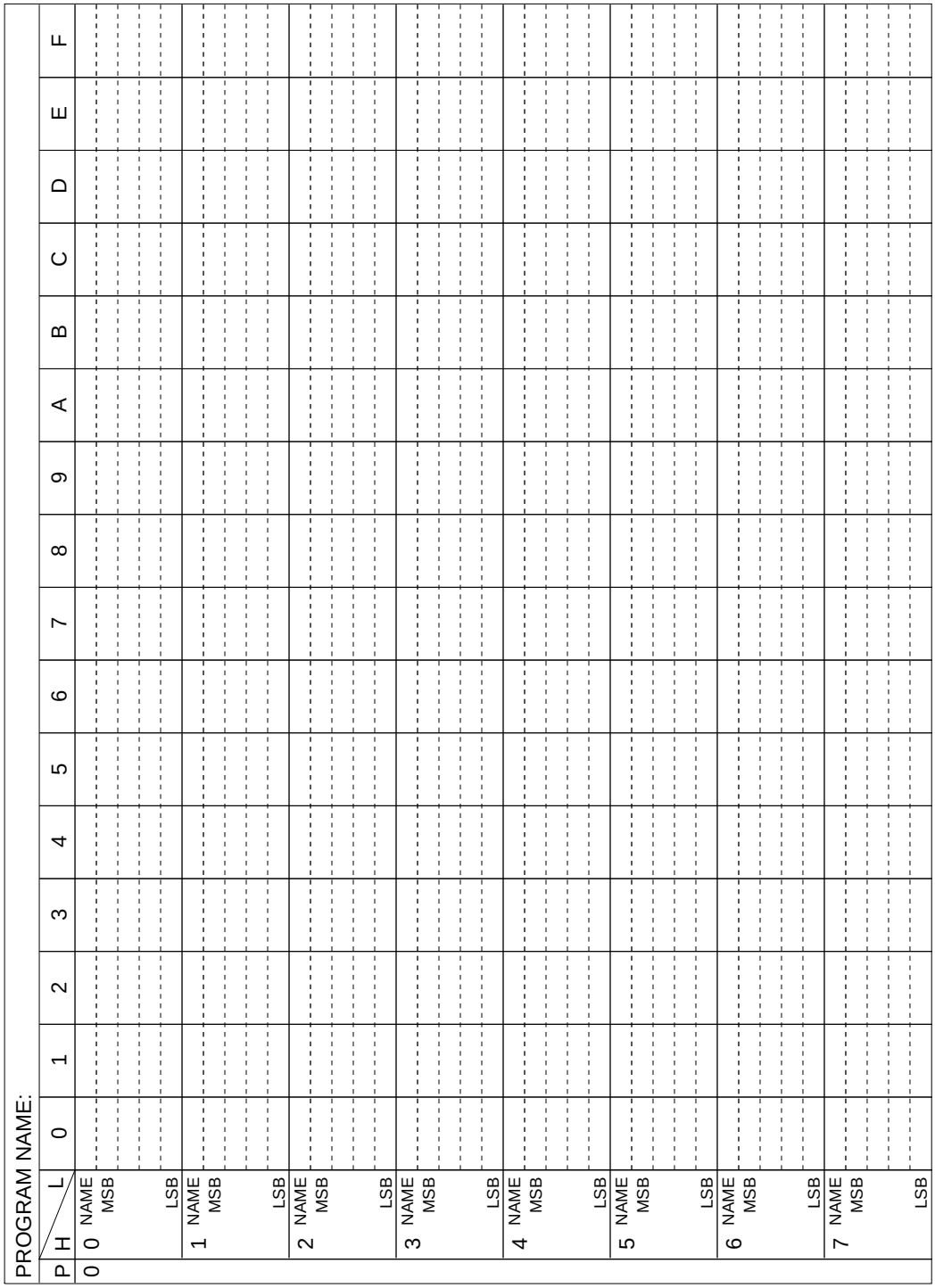
p Five-bit immediate data or label 00H–1FH
s Eight-bit immediate data or label 00H–0FFH
l Eight-bit immediate data 00H–0FFH
i Four-bit immediate data 00H–0FH

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. E0C6256 RAM MAP

RAM map - 1 (000H–07FH)

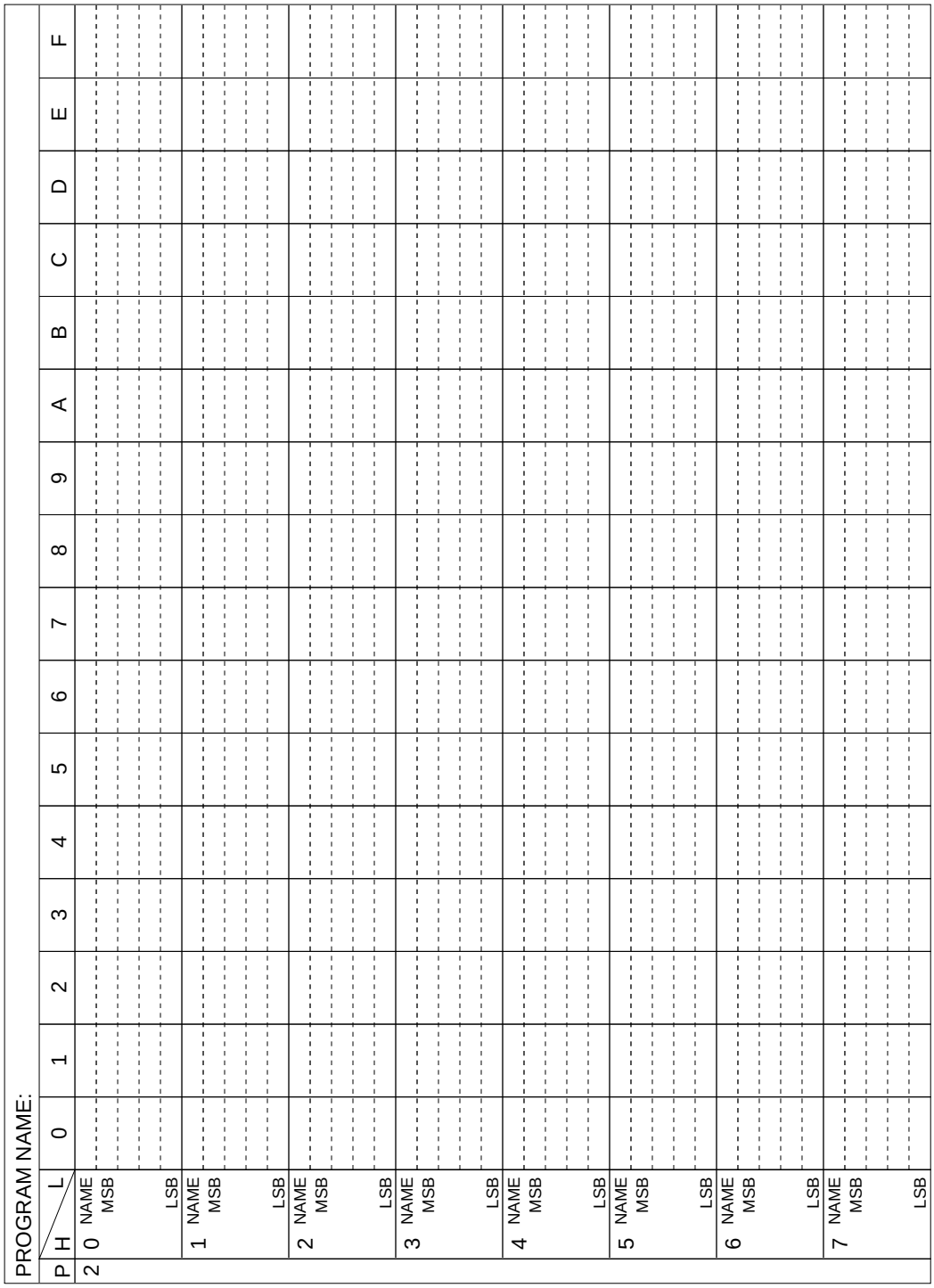


RAM map - 2 (100H–17FH)

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 map - 3 (200H–27FH)



RAM map - 4 (300H–37FH)

PROGRAM NAME:

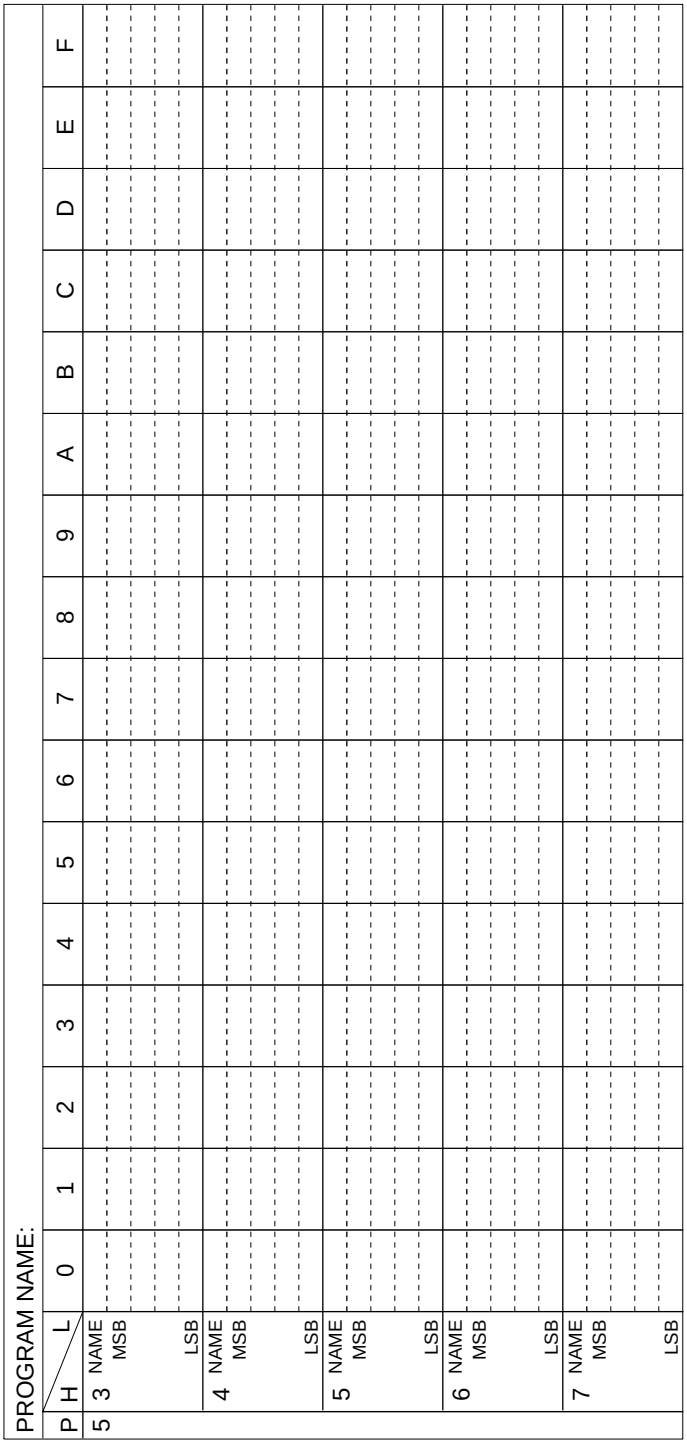
P	H \ L		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
3	0	NAME																
		MSB																
		LSB																
	1	NAME																
		MSB																
		LSB																
	2	NAME																
		MSB																
		LSB																
	3	NAME																
		MSB																
		LSB			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

RAM map - 5 (400H–47FH)

PROGRAM NAME:

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

Display memory map (530H–57FH, W only)



I/O memory map (80H–FEH)

PROGRAM NAME:																	
P	H \ L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	8 NAME																
1	MSB	0	0	0	0	0											
		0	0	VSEL	CLKCHG	VLCHG2											
2		SVDS1	SVDDT	HLON	OSCC	VLCHG1											
	LSB	SVDS0	SVDON	DBON	VSCHG	VLCHG0											
3	9 NAME																
4	MSB	SIK03	K03	KCP03		SIK13	K13	KCP13									
		SIK02	K02	KCP02		SIK12	K12	KCP12									
5		SIK01	K01	KCP01		SIK11	K11	KCP11									
	LSB	SIK00	K00	KCP00		SIK10	K10	KCP10									
	A NAME																
	MSB	0	R03	0	R13												
		0	R02	0	R12												
		0	R01	0	R11												
	LSB	R0HIZ	R00	R1HIZ	R10												
	B NAME																
	MSB	IOC03	PUL03	P03		IOC13	PUL13	P13									
		IOC02	PUL02	P02		IOC12	PUL12	P12									
		IOC01	PUL01	P01		IOC11	PUL11	P11									
	LSB	IOC00	PUL00	P00		IOC10	PUL10	P10									
	C NAME																
	MSB	FOUTE	0	LDMS	0	0	TM3	TM7		0	0	PNRFS	RD3	RD7	PT3	PT7	
		0	0	CSDC	0	0	TM2	TM6		0	0	PTOE	RD2	RD6	PT2	PT6	
		FOFQ1	0	LDTY1	LOFF	TMRUN	TM1	TM5		PTPS1	PTPC1	PTRUN	RD1	RD5	PT1	PT5	
	LSB	FOFQ0	WDRST	LDTY0	LPWR	TMRST	TM0	TM4		PTPS0	PTPC0	PTRST	RD0	RD4	PT0	PT4	
	D NAME																
	MSB	EDIR	LCURF	SWL3	SWM3	SWH3				0	SDP	SD3	SD7	ENRTM	0	0	0
		DKM2	CRNWF	SWL2	SWM2	SWH2				0	SCPS	SD2	SD6	ENRST	BZSTP	BZFQ2	BDTY2
		DKM1	SWRUN	SWL1	SWM1	SWH1				SCTRG	SCS1	SD1	SD5	ENON	BZSHT	BZFQ1	BDTY1
	LSB	DKM0	SWRST	SWL0	SWM0	SWH0				ESIF	SCS0	SD0	SD4	BZE	SHTPW	BZFQ0	BDTY0
	E NAME																
	MSB	0	WRSEL	0	TC3	TC7	TC11	TC15	C3	C7	C11	C15	C19	0			
		APWR	CRSEL	0	TC2	TC6	TC10	TC14	C2	C6	C10	C14	C18	OVF2			
		SENSEL	ADCLK1	0	TC1	TC5	TC9	TC13	C1	C5	C9	C13	C17	OVF1			
	LSB	ADMODE	ADCLK0	CHSEL	TC0	TC4	TC8	TC12	C0	C4	C8	C12	C16	ADRUN			
	F NAME																
	MSB	0	0	0	0	0	EIRUN	EIT3		0	0	0	0	0	IRUN	IT3	
		0	0	0	0	0	EILAP	EIT2		0	0	0	0	0	ILAP	IT2	
		0	0	0	0	0	EISW1	EIT1		0	0	0	0	0	ISW1	IT1	
	LSB	EIAD	EIK1	EIK0	EISIF	EIPT	EISW0	EIT0		IAD	IK1	IK0	ISIF	IPT	ISW0	IT0	

Unused area

APPENDIX C. E0C6256 I/O MEMORY MAP

I/O memory map (80H–84H)

Address *7	Register D3	D2	D1	D0	Name	Init *1	1	0	Comment
80H	0	0	SVDS1	SVDS0	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R		R/W		SVDS1	0			SVD criteria voltage setting (1.5 V/3.0 V)
					SVDS0	0			0: 1.05/2.30, 1: 1.20/2.45, 2 & 3: 1.35/2.60 (V)
81H	0	0	SVDDT	SVDON	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R			R/W	SVDDT	0	Low	Normal	Supply voltage detection data
					SVDON	0	On	Off	SVD circuit On/Off
82H	0	VSEL	HLON	DBON	0 *5	– *2			Unused
					VSEL	0	V_{S2}	V_{SS}	Voltage regulator power source selection
	R	R/W			HLON	0	On	Off	Halver On/Off
					DBON	0	On	Off	Doubler On/Off
83H	0	CLKCHG	OSCC	VSCHG	0 *5	– *2			Unused
					CLKCHG	0	OSC3	OSC1	CPU clock selection
	R	R/W			OSCC	0	On	Off	OSC3 oscillation On/Off
					VSCHG	0	-2.1V	-1.05V	V_{S1} output voltage change
84H	0	VLCHG2	VLCHG1	VLCHG0	0 *5	– *2			Unused
					VLCHG2	0			V_{L1} output voltage change
	R	R/W			VLCHG1	0			0: 1.05, 1: 1.10, 2: 1.15, 3: 1.20,
					VLCHG0	0			4: 1.25, 5: 1.30, 6: 1.35, 7: 1.40 (V)

Remarks

*1 Initial value at the time of initial reset
*2 Not set in the circuit
*3 Undefined
*4 Reset (0) immediately after being read
*5 Constantly "0" when being read
*6 Refer to main manual
*7 Page switching in I/O memory is not necessary

I/O memory map (90H–96H)

Address *7	Register D3	D2	D1	D0	Name	Init *1	1	0	Comment
90H	SIK03	SIK02	SIK01	SIK00	SIK03	0	Enable	Disable	Interrupt selection register (K03)
					SIK02	0	Enable	Disable	Interrupt selection register (K02)
	R/W				SIK01	0	Enable	Disable	Interrupt selection register (K01)
					SIK00	0	Enable	Disable	Interrupt selection register (K00)
91H	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	
92H	KCP03	KCP02	KCP01	KCP00	KCP03	0	↓	↑	Input comparison register (K00–K03)
					KCP02	0	↓	↑	
	R/W				KCP01	0	↓	↑	
					KCP00	0	↓	↑	
94H	SIK13	SIK12	SIK11	SIK10	SIK13	0	Enable	Disable	Interrupt selection register (K13)
					SIK12	0	Enable	Disable	Interrupt selection register (K12)
	R/W				SIK11	0	Enable	Disable	Interrupt selection register (K11)
					SIK10	0	Enable	Disable	Interrupt selection register (K10)
95H	K13	K12	K11	K10	K13	– *2	High	Low	Input port (K10–K13)
					K12	– *2	High	Low	
	R				K11	– *2	High	Low	
					K10	– *2	High	Low	
96H	KCP13	KCP12	KCP11	KCP10	KCP13	0	↓	↑	Input comparison register (K10–K13)
					KCP12	0	↓	↑	
	R/W				KCP11	0	↓	↑	
					KCP10	0	↓	↑	

I/O memory map (A0H–A3H)

Address *7	Register D3	D2	D1	D0	Name	Init *1	1	0	Comment
A0H	0	0	0	R0HIZ	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R			R/W	0 *5	– *2			Unused
					R0HIZ	0	High-Z	Output	R0 output high-impedance control
A1H	R03	R02	R01	R00	R03	0	High	Low	Output port (R00–R03)
					R02	0	High	Low	
	R/W				R01	0	High	Low	
					R00	0	High	Low	
A2H	0	0	0	R1HIZ	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R			R/W	0 *5	– *2			Unused
					R1HIZ	0	High-Z	Output	R1 output high-impedance control
A3H	R13	R12	R11	R10	R13	0	High	Low	Output port (R13)
					R12	0	High	Low	Output port (R12)
		BZ	PTOVF	FOUT	BZ	0	Off	On	Buzzer output
	R/W				R11	0	High	Low	Output port (R11)
					PTOVF	0	Off	On	PTOVF output
					R10	0	High	Low	Output port (R10)
					FOUT	0	Off	On	FOUToutput

I/O memory map (B0H–B6H)

Address *7	Register D3	D2	D1	D0	Name	Init *1	1	0	Comment
B0H	IOC03	IOC02	IOC01	IOC00	IOC03	0	Output	Input	I/O control register (P00–P03)
					IOC02	0	Output	Input	
	R/W				IOC01	0	Output	Input	
					IOC00	0	Output	Input	
B1H	PUL03	PUL02	PUL01	PUL00	PUL03	1	On	Off	Pull down control register (P00–P03)
					PUL02	1	On	Off	
	R/W				PUL01	1	On	Off	
					PUL00	1	On	Off	
B2H	P03	P02	P01	P00	P03	– *2	High	Low	I/O port (P00–P03)
					P02	– *2	High	Low	
	R/W				P01	– *2	High	Low	
					P00	– *2	High	Low	
B4H	IOC13	IOC12	IOC11	IOC10	IOC13	0	Output	Input	I/O control register (P10–P13) (ESIF = 0)
					IOC12	0	Output	Input	
	R/W				IOC11	0	Output	Input	
					IOC10	0	Output	Input	
	When the serial I/F is used (ESIF = 1):				IOC13	0	Output	Input	Master mode: P13 I/O control register
	P10 = SIN (in), P11 = SOUT (out),				IOC13	0	1	0	Slave mode: General-purpose register
	P12 = SCLK (master: out, slave: in),				IOC12	0	1	0	General-purpose register
	P13 = SRDY (slave: out),				IOC11	0	1	0	
	P13 = I/O port (master: in/out)				IOC10	0	1	0	
B5H	PUL13	PUL12	PUL11	PUL10	PUL13	1	On	Off	Pull down control register (P10–P13) (ESIF = 0)
					PUL12	1	On	Off	
	R/W				PUL11	1	On	Off	
					PUL10	1	On	Off	
	When the serial I/F is used (ESIF = 1):				PUL13	1	On	Off	Master mode: P13 pull down control register
	P10 = SIN (in), P11 = SOUT (out),				PUL13	1	1	0	Slave mode: General-purpose register
	P12 = SCLK (master: out, slave: in),				PUL12	1	1	0	Master mode: General-purpose register
	P13 = SRDY (slave: out),				PUL12	1	On	Off	Slave mode: SCKL pull down control register
	P13 = I/O port (master: in/out)				PUL11	1	1	0	General-purpose register
					PUL10	1	On	Off	SIN pull down control register
B6H	P13	P12	P11	P10	P13	– *2	High	Low	I/O port (P10–P13) (ESIF = 0)
					P12	– *2	High	Low	
	R/W				P11	– *2	High	Low	
					P10	– *2	High	Low	
	When the serial I/F is used (ESIF = 1):				P13	– *2	High	Low	Master mode: I/O port P13
	P10 = SIN (in), P11 = SOUT (out),				P13	– *2	1	0	Slave mode: General-purpose register
	P12 = SCLK (master: out, slave: in),				P12	– *2	1	0	General-purpose register
	P13 = SRDY (slave: out),				P11	– *2	1	0	
	P13 = I/O port (master: in/out)				P10	– *2	1	0	

I/O memory map (C0H–CEH)

Address *7	Register D3	D2	D1	D0	Name	Init *1	1	0	Comment
C0H	FOUTE	0	FOFQ1	FOFQ0	FOUTE	0	Enable	Disable	FOUT output enable
					0 *5	– *2			Unused
	R/W	R	R/W		FOFQ1	0			FOUT frequency selection
					FOFQ0	0			0: 512 Hz, 1: 4096 Hz, 2: $fosc_1$, 3: $fosc_3$
C1H	0	0	0	WDRST	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R			W	0 *5	– *2			Unused
					WDRST *5	Reset	Reset	–	Wachdog timer reset
C2H	LDMS	CSDC	LDTY1	LDTY0	LDMS	0	1p(R/W)	5p(W)	LCD data memory area selection
					CSDC	0	Static	Dynamic	LCD drive switch
	R/W				LDTY1	0			LCD drive duty selection
					LDTY0	0			0: 1/2, 1: 1/3, 2: 1/4, 3: 1/5
C3H	0	0	LOFF	LPWR	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R		R/W		LOFF	0	All Off	Normal	LCD display control
					LPWR	0	On	Off	LCD power supply On/Off
C4H	0	0	TMRUN	TMRST	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R		R/W	W	TMRUN	0	Run	Stop	Clock timer Run/Stop
					TMRST *5	Reset	Reset	–	Clock timer reset
C5H	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)
C6H	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)
C8H	0	0	PTPS1	PTPS0	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R		R/W		PTPS1	0			Prog. timer prescaler selection
					PTPS0	0			0: 1/1, 1: 1/4, 2: 1/32, 3: 1/256
C9H	0	0	PTPC1	PTPC0	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R		R/W		PTPC1	0			Prog. timer prescaler clock source selection
					PTPC0	0			0: OSC1, 1: OSC3, 2: K02, 3: K02(NR)
CAH	PNRFS	PTOE	PTRUN	PTRST	PNRFS	0	1024Hz	256Hz	Noise rejector clock frequency selection
					PTOE	0	Enable	Disable	PTOVF output enable
	R/W			W	PTRUN	0	Run	Stop	Programmable timer Run/Stop
					PTRST *5	Rst (reload)	Rst (reload)	–	Programmable timer reset (reload)
CBH	RD3	RD2	RD1	RD0	RD3	0			MSB
					RD2	0			Programmable timer reload data
	R/W				RD1	0			(low-order 4 bits)
					RD0	0			LSB
CCH	RD7	RD6	RD5	RD4	RD7	0			MSB
					RD6	0			Programmable timer reload data
	R/W				RD5	0			(high-order 4 bits)
					RD4	0			LSB
CDH	PT3	PT2	PT1	PT0	PT3	0			MSB
					PT2	0			Programmable timer data
	R				PT1	0			(low-order 4 bits)
					PT0	0			LSB
CEH	PT7	PT6	PT5	PT4	PT7	0			MSB
					PT6	0			Programmable timer data
	R				PT5	0			(high-order 4 bits)
					PT4	0			LSB

I/O memory map (D0H–DFH)

Address *7	Register D3	D2	D1	D0	Name	Init *1	1	0	Comment
D0H	EDIR	DKM2	DKM1	DKM0	EDIR	0	Enable	Disable	Enable direct input function
					DKM2	0			Direct key mask factor 0: Non, 1: K02,
	R/W				DKM1	0			2: K02-03, 3: K02-03-10, 4: K10, 5: K10-11,
					DKM0	0			6: K10-11-12, 7: K10-11-12-13
D1H	LCURF	CRNWF	SWRUN	SWRST	LCURF	0	Reqest	No	Lap data carry-up request flag
					CRNWF	0	Renewal	No	Capture renewal flag
	R		R/W	W	SWRUN	0	Run	Stop	Stopwatch timer Run/Stop
					SWRST *5	Reset	Reset	–	Stopwatch timer reset
D2H	SWL3	SWL2	SWL1	SWL0	SWL3	0			MSB
					SWL2	0			Stopwatch timer data 1/1000 sec (BCD)
	R				SWL1	0			
					SWL0	0			LSB
D3H	SWM3	SWM2	SWM1	SWM0	SWM3	0			MSB
					SWM2	0			Stopwatch timer data 1/100 sec (BCD)
	R				SWM1	0			
					SWM0	0			LSB
D4H	SWH3	SWH2	SWH1	SWH0	SWH3	0			MSB
					SWH2	0			Stopwatch timer data 1/10 sec (BCD)
	R				SWH1	0			
					SWH0	0			LSB
D8H	0	0	SCTRG	ESIF	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R		R/W		SCTRG(W)	– *2	Trigger	–	Serial interface clock trigger (writing)
					SCTRG(R)	0	Run	Stop	Serial interface clock status (reading)
					ESIF	0	SIF port	I/O port	P1 port function selection
D9H	SDP	SCPS	SCS1	SCS0	SDP	0	LSB first	MSB first	Serial data input/output permutation
					SCPS	0			Serial interface clock phase selection
	R/W				SCS1	0			Serial interface clock mode selection
					SCS0	0			0: Slave, 1: PTOVF, 2: CLK/2, 3: CLK
DAH	SD3	SD2	SD1	SD0	SD3	– *3			MSB
					SD2	– *3			Serial interface data (low-order 4 bits)
	R/W				SD1	– *3			
					SD0	– *3			LSB
DBH	SD7	SD6	SD5	SD4	SD7	– *3			MSB
					SD6	– *3			Serial interface data (high-order 4 bits)
	R/W				SD5	– *3			
					SD4	– *3			LSB
DCH	ENRTM	ENRST	ENON	BZE	ENRTM	0	1sec	0.5sec	Envelope releasing time
					ENRST *5	Reset	Reset	–	Envelope reset
	R/W	W	R/W		ENON	0	On	Off	Envelope On/Off
					BZE	0	Enable	Disable	Buzzer output enable
DDH	0	BZSTP	BZSHT	SHTPW	0 *5	– *2			Unused
					BZSTP *5	– *2	Stop	–	1-shot buzzer stop
	R	W	R/W		BZSHT(W)	– *2	Trigger	–	1-shot buzzer trigger (writing)
					BZSHT(R)	0	Busy	Ready	1-shot buzzer status (reading)
					SHTPW	0	125msec	31.25msec	1-shot buzzer pulse width setting
DEH	0	BZFQ2	BZFQ1	BZFQ0	0 *5	– *2			Unused
					BZFQ2	0			Buzzer frequency selection
	R	R/W			BZFQ1	0			0: 4096.0, 1: 3276.8, 2: 2730.7, 3: 2340.6,
					BZFQ0	0			4: 2048.0, 5: 1638.4, 6: 1365.3, 7: 1170.3 (Hz)
DFH	0	BDTY2	BDTY1	BDTY0	0 *5	– *2			Unused
					BDTY2	0			Buzzer signal duty ratio selection *6
	R	R/W			BDTY1	0			
					BDTY0	0			

I/O memory map (E0H–ECH)

Address *7	Register D3	D2	D1	D0	Name	Init *1	1	0	Comment
E0H	0	APWR	SENSEL	ADMODE	0 *5	– *2			Unused
					APWR	0	On	Off	A/D converter power supply On/Off
	R	R/W			SENSEL	0	Sensor	Reference	Sensor/reference device selection
					ADMODE	0	Continuous	Normal	A/D converter operation mode selection
E1H	WRSEL	0	ADCLK1	ADCLK0	WRSEL	0	Humidity	Resistance	CH1 sensor selection (resistance/humidity)
					0 *5	– *2			Unused
	R/W	R	R/W		ADCLK1	0	OSC3	OSC1	CH1 clock selection
					ADCLK0	0	OSC3	OSC1	CH0 clock selection
E2H	0	0	0	CHSEL	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R			R/W	0 *5	– *2			Unused
					CHSEL	0	CH1	CH0	Channel 1/0 selection
E3H	TC3	TC2	TC1	TC0	TC3	– *3			Up/down counter data (TC0–TC3)
					TC2	– *3			
	R/W				TC1	– *3			
					TC0	– *3			LSB
E4H	TC7	TC6	TC5	TC4	TC7	– *3			Up/down counter data (TC4–TC7)
					TC6	– *3			
	R/W				TC5	– *3			
					TC4	– *3			
E5H	TC11	TC10	TC9	TC8	TC11	– *3			Up/down counter data (TC8–TC11)
					TC10	– *3			
	R/W				TC9	– *3			
					TC8	– *3			
E6H	TC15	TC14	TC13	TC12	TC15	– *3			MSB
					TC14	– *3			Up/down counter data (TC12–TC15)
	R/W				TC13	– *3			
					TC12	– *3			
E7H	C3	C2	C1	C0	C3	– *3			Up counter data (C0–C3)
					C2	– *3			
	R/W				C1	– *3			
					C0	– *3			LSB
E8H	C7	C6	C5	C4	C7	– *3			Up counter data (C4–C7)
					C6	– *3			
	R/W				C5	– *3			
					C4	– *3			
E9H	C11	C10	C9	C8	C11	– *3			Up counter data (C8–C11)
					C10	– *3			
	R/W				C9	– *3			
					C8	– *3			
EAH	C15	C14	C13	C12	C15	– *3			Up counter data (C12–C15)
					C14	– *3			
	R/W				C13	– *3			
					C12	– *3			
EBH	C19	C18	C17	C16	C19	– *3			MSB
					C18	– *3			Up counter data (C16–C19)
	R/W				C17	– *3			
					C16	– *3			
ECH	0	OVF2	OVF1	ADRUN	0 *5	– *2			Unused
					OVF2(R)	0	Yes	No	Up/down counter overflow flag
					OVF2(W)	Reset	Reset	–	Up/down counter overflow flag reset
	R	R/W			OVF1(R)	0	Yes	No	Up counter overflow flag
					OVF1(W)	Reset	Reset	–	Up counter overflow flag reset
					ADRUN	0	Start	Stop	A/D conversion Start/Stop

I/O memory map (F0H–FEH)

Address *7	Register D3	D2	D1	D0	Name	Init *1	1	0	Comment
F0H	0	0	0	EIAD	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R			R/W	0 *5	– *2			Unused
					EIAD	0	Enable	Mask	Interrupt mask register (A/D converter)
F1H	0	0	0	EIK1	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R			R/W	0 *5	– *2			Unused
					EIK1	0	Enable	Mask	Interrupt mask register (K10–K13)
F2H	0	0	0	EIK0	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R			R/W	0 *5	– *2			Unused
					EIK0	0	Enable	Mask	Interrupt mask register (K00–K03)
F3H	0	0	0	EISIF	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R			R/W	0 *5	– *2			Unused
					EISIF	0	Enable	Mask	Interrupt mask register (Serial interface)
F4H	0	0	0	EIPT	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R			R/W	0 *5	– *2			Unused
					EIPT	0	Enable	Mask	Interrupt mask register (Programmable timer)
F5H	EIRUN	EILAP	EISW1	EISW0	EIRUN	0	Enable	Mask	Interrupt mask register (Stopwatch direct RUN)
					EILAP	0	Enable	Mask	Interrupt mask register (Stopwatch direct LAP)
	R/W				EISW1	0	Enable	Mask	Interrupt mask register (Stopwatch 1 Hz)
					EISW0	0	Enable	Mask	Interrupt mask register (Stopwatch 10 Hz)
F6H	EIT3	EIT2	EIT1	EIT0	EIT3	0	Enable	Mask	Interrupt mask register (Clock timer 2 Hz)
					EIT2	0	Enable	Mask	Interrupt mask register (Clock timer 8 Hz)
	R/W				EIT1	0	Enable	Mask	Interrupt mask register (Clock timer 16 Hz)
					EIT0	0	Enable	Mask	Interrupt mask register (Clock timer 32 Hz)
F8H	0	0	0	IAD	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R				0 *5	– *2			Unused
					IAD *4	0	Yes	No	Interrupt factor flag (A/D converter)
F9H	0	0	0	IK1	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R				0 *5	– *2			Unused
					IK1 *4	0	Yes	No	Interrupt factor flag (K10–K13)
FAH	0	0	0	IK0	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R				0 *5	– *2			Unused
					IK0 *4	0	Yes	No	Interrupt factor flag (K00–K03)
FBH	0	0	0	ISIF	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R				0 *5	– *2			Unused
					ISIF *4	0	Yes	No	Interrupt factor flag (Serial interface)
FCH	0	0	0	IPT	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R				0 *5	– *2			Unused
					IPT *4	0	Yes	No	Interrupt factor flag (Programmable timer)
FDH	IRUN	ILAP	ISW1	ISW0	IRUN *4	0	Yes	No	Interrupt factor flag (Stopwatch direct RUN)
					ILAP *4	0	Yes	No	Interrupt factor flag (Stopwatch direct LAP)
	R				ISW1 *4	0	Yes	No	Interrupt factor flag (Stopwatch 1 Hz)
					ISW0 *4	0	Yes	No	Interrupt factor flag (Stopwatch 10 Hz)
FEH	IT3	IT2	IT1	IT0	IT3 *4	0	Yes	No	Interrupt factor flag (Clock timer 2 Hz)
					IT2 *4	0	Yes	No	Interrupt factor flag (Clock timer 8 Hz)
	R				IT1 *4	0	Yes	No	Interrupt factor flag (Clock timer 16 Hz)
					IT0 *4	0	Yes	No	Interrupt factor flag (Clock timer 32 Hz)

APPENDIX D. TROUBLESHOOTING

Tool	Problem	Remedy measures
ICE62R (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 ICS6256J.EXE PC-DOS ICS6256W.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 ICE62R (ICE6200) unit set correctly? • Is the breaker of the ICE62R (ICE6200) set to ON?
	The ICE6200 breaker tripped or the ICE62R fuse cut immediately after activation.	Check the following and remedy if necessary: • Are connectors F1 and F5 connected to the EVA6256 correctly? • Is the target board power short-circuiting?
	<ILLEGAL VERSION ICE6200> appears on the screen immediately after activation.	The wrong version of ICE is being used. Use the latest version.
	<ILLEGAL VERSION PARAMETER FILE> appears on the screen immediately after activation.	The wrong version of ICS6256P.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.
SOG6256	An R error occurs although the address is correctly set in the segment source file.	Check the following and remedy if necessary: • Does the address symbol use capital letters?

Tool	Problem	Remedy measures
ASM6256	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.
MDC6256	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?
EVA6256	The EVA6256 does not work when it is used independently.	Check the following and remedy if necessary: • Has the EPROM for F.HEX and S.HEX been replaced by the EPROM for the target? • Is the EPROM for F.HEX and S.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 EVA6256 set to EN?
	Target segment does not light.	Check the following and remedy if necessary: • Is an EPROM with an access time of 250 ns or less being used for S.HEX. • Has the VADJ VR inside the EVA6256 top cover been turned to a lower setting?

EPSON International Sales Operations

AMERICA

EPSON ELECTRONICS AMERICA, INC.

- HEADQUARTERS -
1960 E. Grand Avenue
El Segundo, CA 90245, U.S.A.
Phone: +1-310-955-5300 Fax: +1-310-955-5400

- SALES OFFICES -

West
150 River Oaks Parkway
San Jose, CA 95134, U.S.A.
Phone: +1-408-922-0200 Fax: +1-408-922-0238

Central
101 Virginia Street, Suite 290
Crystal Lake, IL 60014, U.S.A.
Phone: +1-815-455-7630 Fax: +1-815-455-7633

Northeast
301 Edgewater Place, Suite 120
Wakefield, MA 01880, U.S.A.
Phone: +1-781-246-3600 Fax: +1-781-246-5443

Southeast
3010 Royal Blvd. South, Suite 170
Alpharetta, GA 30005, U.S.A.
Phone: +1-877-EEA-0020 Fax: +1-770-777-2637

EUROPE

EPSON EUROPE ELECTRONICS GmbH

- HEADQUARTERS -
Riesstrasse 15
80992 Muenchen, GERMANY
Phone: +49-(0)89-14005-0 Fax: +49-(0)89-14005-110

- GERMANY -
SALES OFFICE
Altstadtstrasse 176
51379 Leverkusen, GERMANY
Phone: +49-(0)217-15045-0 Fax: +49-(0)217-15045-10

- UNITED KINGDOM -
UK BRANCH OFFICE
2.4 Doncastle House, Doncastle Road
Bracknell, Berkshire RG12 8PE, ENGLAND
Phone: +44-(0)1344-381700 Fax: +44-(0)1344-381701

- FRANCE -
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

- CHINA -
EPSON (CHINA) CO., LTD.
28F, Beijing Silver Tower 2# North RD DongSanHuan
ChaoYang District, Beijing, CHINA
Phone: 64106655 Fax: 64107320

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

- 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

- 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

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
50F, 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 Epson WWW server

http://www.epson.co.jp/device/

First issue SEPTEMBER 1993, Printed SEPTEMBER 1999 in Japan Ⓜ A