MF798-02

EPSON

CMOS 4-BIT SINGLE CHIP MICROCOMPUTER

# E0C6292 DEVELOPMENT TOOL MANUAL

ENERGY SAVING EPSON

SEIKO EPSON CORPORATION

# E0C6292 Development Tool Manual

## *PREFACE*

This manual mainly explains the outline of the development support tool for the 4-bit Single Chip Microcomputer E0C6292.

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<br>EVA621C Manual<br>ICE6200 Hardware Manual |
| *Development procedure* | ☞ E0C62 Family Technical Guide |
| *Device (E0C6292)* | ☞ E0C6292 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 DEV6292

The below software are included in the product of the E0C6292 development support tool DEV6292.

1. Development Tool Management System DMS6200 ..... Menu selection for each software / start-up software
2. Cross Assembler ASM6292 ..... Cross assembler for program preparation
3. Function Option Generator FOG6292 ..... Function option data preparation program
4. Segment Option Generator SOG6292 ..... Segment option data preparation program
5. ICE Control Software ICS6292 ..... ICE control program
6. Mask Data Checker MDC6292 ..... Mask data preparation program

## 1.2 Developmental Environment

The software product of the development support tool DEV6292 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 E0C6292, 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.

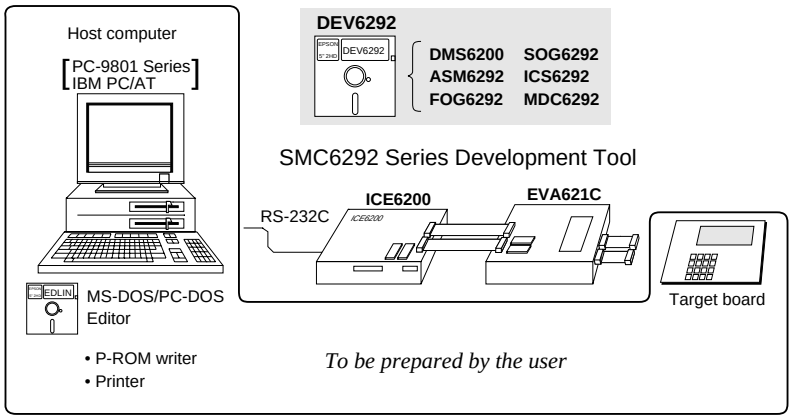



*Fig. 1.2.1 System configuration*

The EVA621C can be used as the EVA board for the E0C6292 by modifying the function option HEX data generated from the FOG6292. Therefore, if correct function option HEX data for the E0C6292 is not loaded in the EVA621C, the EVA621C does not identify as E0C6292 functions and does not operate correctly. Especially when the EVA621C is used for both E0C6292 and E0C621C, be aware of the function option HEX data to be loaded.

*Note The DEV6292 system requires a host computer with a RAM capacity of about 140K bytes. Since the ICE6200 is connected to the host computer with a RS-232C serial interface, adapter board for asynchronous communication will be required depending on the host computer used.*

## *1.3 Development Flow*

Figure 1.3.1 shows the development flow through the DEV6292.

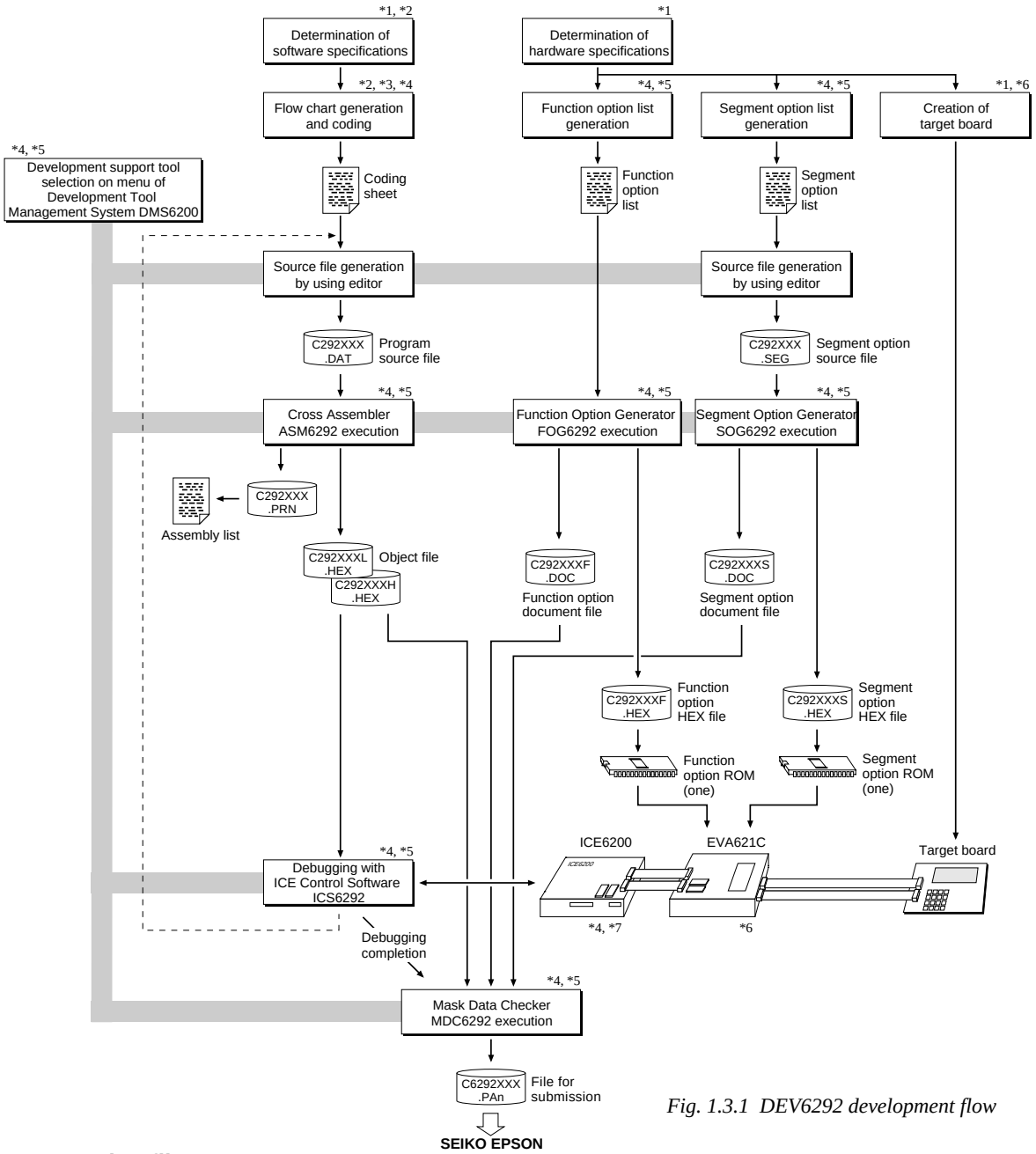



*Fig. 1.3.1 DEV6292 development flow*

### Concerning file names

All the input-output file name for the each development support tool commonly use "C292XXX". In principle each file should be produced in this manner. Seiko Epson will designate the "XXX" for each customer.

### ☞ Reference Manual

*1 E0C6292 Technical Hardware Manual
*2 E0C6292 Technical Software Manual
*3 E0C6200/6200A Core CPU Manual
*4 E0C62 Family Development Tool Reference Manual
*5 E0C6292 Development Tool Manual (this manual)
*6 EVA621C Manual
*7 ICE6200 Hardware Manual

## 1.4 Production of Execution Disk

Execution files for each software development support tool and batch and parameter files for the ICE6200 are recorded in the DEV6292 floppy disk.
The content of the files contained in the DEV6292 floppy disk are shown below.

| PC-DOS version | MS-DOS version | Contents |
|---|---|---|
| ASM6292.EXE | ASM6292.EXE | Cross Assembler execution file |
| DMS6200.EXE | DMS6200.EXE | Development Tool Management System execution file |
| FOG6292.EXE | FOG6292.EXE | Function Option Generator execution file |
| ICS6292B.BAT | ICS6292.BAT | ICE Control Software batch file |
| ICS6292W.EXE | ICS6292J.EXE | ICE Control Software execution file |
| ICS6292P.PAR | ICS6292P.PAR | ICE Control Software parameter file |
| MDC6292.EXE | MDC6292.EXE | Mask Data Checker execution file |
| SOG6292.EXE | SOG6292.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 (DEV6292, etc.) then copy the content of the floppy disk into that subdirectory using the COPY command.

Example:

***Copying into a floppy disk***

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

```
A>DISKCOPY  A:  B:↵
```

***Copying into a hard disk (C drive)***

Make a subdirectory (DEV6292), then insert the original disk into the A drive and execute the COPY command.

```
C\>MD DEV6292↵
C\>CD DEV6292↵
C\DEV6292\>COPY A:*.*↵
```

- Next make a CONFIG.SYS file using Editor or the like.
  When a CONFIG.SYS has previously been made using a hard disk system, check the setting of the FILES within it. (If there is none add it.)
  Set the number of files to be described in CONFIG.SYS at 10 or more, so that the Mask Data Checker MDC6292 will handle many files.

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

Example:

***Setting of FILES (CONFIG.SYS)***

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

- 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.

- In "ICS6292(B).BAT" the batch process is indicated such that the ICS6292J(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 DEV6292 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.

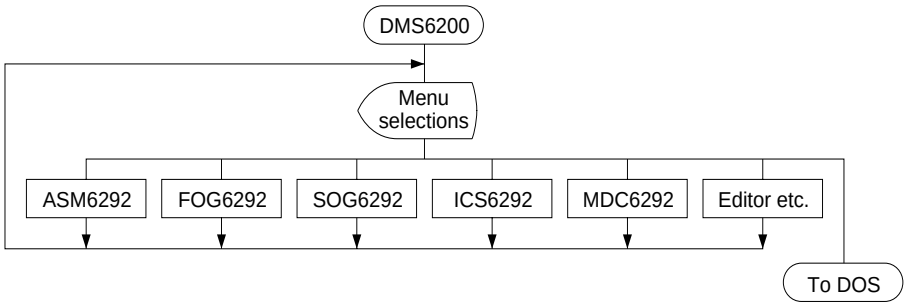



*Fig. 2.1.1 DMS6200 execution flow*

*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       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.

                          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) ASM6292 .EXE
2) FOG6292 .EXE
3) ICS6292B.BAT
4) ICS6292W.EXE
5) MDC6292 .EXE
6) SOG6292 .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) C292XXX .DAT
 2) C292XXX .PRN
 3) C292XXX .SEG
 :     :
10) C6292XXX.PA0

 Input Number ? [1  ]

   Edit > [ASM6292 C292XXX                                        ]
```

***Source file selection screen***
To starting ASM6292, 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 ASM6292

## 3.1 ASM6292 Outline

The ASM6292 cross assembler is an assembler program for generating the machine code used by the E0C6292 4-bit, single-chip microcomputers. The Cross Assembler ASM6292 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.

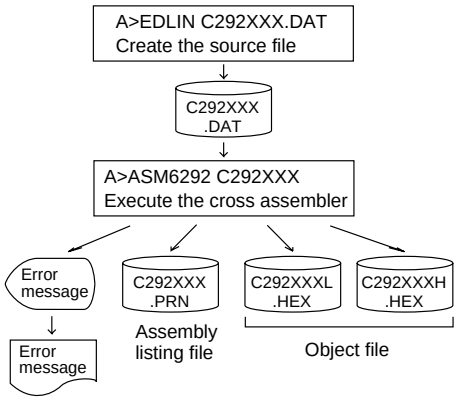



*Fig. 3.1.1 ASM6292 execution flow*

☞ *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 E0C6292 Restrictions

Note the following when generating a program by the E0C6292:

### ■ ROM area

The capacity of the E0C6292 ROM is 2K steps (0000H to 07FFH).
Therefore, the specification range of the memory setting pseudo-instructions and PSET instruction is restricted.

***Memory configuration:***

Bank: Only bank 0, Page: 8 pages (0 to 07H), each 256 steps

***Significant specification range:***

| | | |
|---|---|---|
| ORG | pseudo-instruction: | 0000H to 07FFH |
| PAGE | pseudo-instruction: | 00H to 07H |
| BANK | pseudo-instruction: | Only 0H |
| PSET | instruction: | 00H to 07H |

### ■ RAM area

The capacity of the E0C6292 RAM is 193 words (000H to 1FFH, 4 bits/word).
However, note the following points when programming.

(1) The following addresses become unused area. Memory access is invalid when the unused area is specified.
- 82H–90H, 93H–9FH, A1H–AFH, B1H, B3H, B5H, B7H–BFH, C0H, C2H–C5H, C7H, CBH–CFH, D7H–DFH, E0H–E7H, EBH–EFH, F2H, F5H–F7H, FAH and FDH–FFH in page 0 to 1
- 100H–15FH

(2) Since RAM is set for up to 1 page, only the subordinate 1 bit of the page section of the index register which specifies address is effective. (The 3 superordinate bits are ignored.)

***Example:***

```
LD    A,2
LD    XP,A
LD    X,90H
```

090H 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.

### ■ Undefined codes

The SLP instruction has not been defined in the E0C6292 instruction sets.

## 3.3 ASM6292 Quick Reference

### ■ Starting command and input/output files

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

***Execution file:*** ASM6292.EXE

***Starting command:*** **ASM6292_ [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)<br>@: Current drive, Z: File is not generated |
| -N | The code (FFH) in the undefined area of program memory is not created. |

***Input file:*** C292XXX.DAT (Source file)

***Output file:***
C292XXXL.HEX (Object file, low-order)
C292XXXH.HEX (Object file, high-order)
C292XXX.PRN (Assembly listing file)

### ■ Display example

```
          *** E0C6292 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 1993 SEIKO EPSON CORP.

          SOURCE FILE NAME IS " C292XXX.DAT "

          THIS SOFTWARE MAKES NEXT FILES.

             C292XXXH.HEX  ...  HIGH BYTE OBJECT FILE.
             C292XXXL.HEX  ...  LOW BYTE OBJECT FILE.
             C292XXX .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 ASM6292 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, ASM6292 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`<br>`BCD EQU ABC+1` |
| SET (Set) | To allocate data to label (data can be changed) | `ABC SET 0001H`<br>`ABC SET 0002H` |
| DW (Define Word) | To define ROM data | `ABC DW 'AB'`<br>`BCD DW 0FFBH` |
| ORG (Origin) | To define location counter | `ORG 100H`<br>`ORG 256` |
| PAGE (Page) | To define boundary of page | `PAGE 1H`<br>`PAGE 3` |
| SECTION (Section) | To define boundary of section | `SECTION` |
| END (End) | To terminate assembly | `END` |
| MACRO (Macro) | To define macro | `CHECK MACRO DATA`<br>`LOCAL LOOP`<br>`LOOP CP MX,DATA`<br>`JP NZ,LOOP`<br>`ENDM`<br><br>`CHECK 1` |
| LOCAL (Local) | To make local specification of label during macro definition | |
| ENDM (End Macro) | To end macro definition | |

## ■ Error messages

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

# 4 FUNCTION OPTION GENERATOR FOG6292

## 4.1 FOG6292 Outline

With the 4-bit single-chip E0C6292 microcomputers, the customer may select five hardware options. By modifying the mask patterns of the E0C6292 according to the selected options, the system can be customized to meet the specifications of the target system.
The Function Option Generator FOG6292 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 FOG6292, the E0C6292 mask pattern is automatically generated by a general purpose computer.
The HEX file for the evaluation board (EVA621C) hardware option ROM is simultaneously generated with the data file.

(The EVA621C can be used as the EVA board for the E0C6292 by modifying the function option HEX data generated from the FOG6292.)

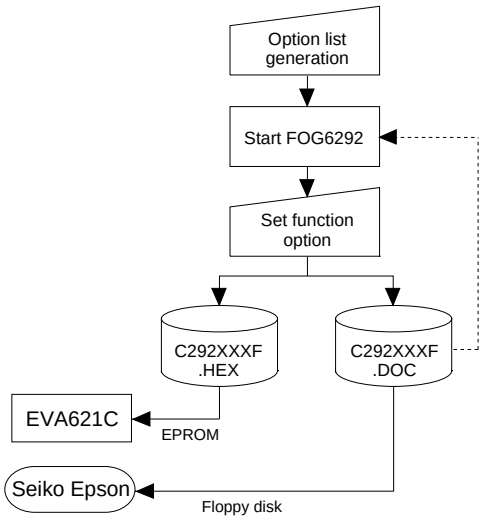



*Fig. 4.1.1 FOG6292 execution flow*

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

## 4.2 E0C6292 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 OSCILLATOR**

☐ 1. CR
☐ 2. Ceramic

**2. INTERRUPT (K00–K03)**

- K00 ..... ☐ 1. Use ☐ 2. Not Use
- K01 ..... ☐ 1. Use ☐ 2. Not Use
- K02 ..... ☐ 1. Use ☐ 2. Not Use
- K03 ..... ☐ 1. Use ☐ 2. Not Use

**3. INPUT PORT PULL-UP RESISTOR**

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

**4. R02 OUTPUT SPECIFICATION**

- R02 ..... ☐ 1. D.C.
  ☐ 2. FOUT

**FOUT OR SIF FREQUENCY**

- FREQ. ..... ☐ 1. 1/64 OSC1
  ☐ 2. 1/8 OSC1
  ☐ 3. OSC1
  ☐ 4. OSC3

## 5. R03 OUTPUT SPECIFICATION

- R03 ..... ☐ 1. D.C.
  ☐ 2. BZ 1/16 OSC1
  ☐ 3. BZ 1/8 OSC1

## 6. OUTPUT PORT OUTPUT SPECIFICATION

- R00 ..... ☐ 1. Complementary ☐ 2. Nch Open Drain
- R01 ..... ☐ 1. Complementary ☐ 2. Nch Open Drain
- R02 ..... ☐ 1. Complementary ☐ 2. Nch Open Drain
- R03 ..... ☐ 1. Complementary ☐ 2. Nch Open Drain

## 7. I/O PORT PULL-UP RESISTOR

- P00 ..... ☐ 1. With Resistor ☐ 2. Gate Direct
- P01 ..... ☐ 1. With Resistor ☐ 2. Gate Direct
- P02 ..... ☐ 1. With Resistor ☐ 2. Gate Direct
- P03 ..... ☐ 1. With Resistor ☐ 2. Gate Direct
- P10 ..... ☐ 1. With Resistor ☐ 2. Gate Direct
- P11 ..... ☐ 1. With Resistor ☐ 2. Gate Direct
- P12 ..... ☐ 1. With Resistor ☐ 2. Gate Direct
- P13 ..... ☐ 1. With Resistor ☐ 2. Gate Direct

## 8. SVD VOLTAGE

☐ 1. 2.4[V]
☐ 2. 2.5[V]
☐ 3. 2.6[V]
☐ 4. 2.7[V]

## 9. LCD DRIVE DUTY

☐ 1. 1/2
☐ 2. 1/3
☐ 3. 1/4

## 4.3 Option Specifications and Selection Message

Screen that can be selected as function options set on the E0C6292 are shown below, and their specifications are also described.

### 1 OSC3 oscillator

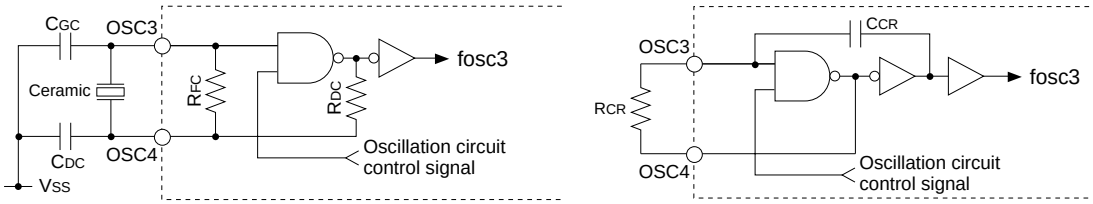
```
*** OPTION  NO.1 ***

--- OSC3 OSCILLATOR ---
                        1. CR
                        2. CERAMIC

PLEASE SELECT NO.(1) ? 1↵

                        1. CR  SELECTED
```

Select the OSC3 system clock to be used for the E0C6292. The OSC3 oscillation circuit generates the system clock when operating the CPU in high-speed. Either ceramic oscillation or CR oscillation can be selected for the oscillation circuit that uses the OSC3 and OSC4 terminals. When ceramic oscillation is selected, ceramic oscillator, gate capacitor and drain capacitor should be connected between the OSC3 and the OSC4 terminals. When CR oscillation is selected, resistor only need be connected between the terminals.
The configuration of the OSC3 oscillation circuit is shown in Figure 4.3.1.




(1) Ceramic oscillation circuit (2) CR oscillation circuit

*Fig. 4.3.1 OSC3 oscillation circuit*

### 2 Interrupt (K00–K03)

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

--- INTERRUPT (K00-KO3) ---
        K00             1. USE
                        2. NOT USE

PLEASE SELECT NO.(1) ? 1↵

        K01             1. USE
                        2. NOT USE

PLEASE SELECT NO.(1) ? 1↵

        K02             1. USE
                        2. NOT USE

PLEASE SELECT NO.(1) ? 1↵

        K03             1. USE
                        2. NOT USE

PLEASE SELECT NO.(1) ? 1↵

        K00             1. USE  SELECTED
        K01             1. USE  SELECTED
        K02             1. USE  SELECTED
        K03             1. USE  SELECTED
```

Select whether the input port interrupt function that generates interrupt according to the input status will be used or not. It should be selected for each bit of K00–K03.
When "NOT USE" is selected, the interrupt is masked even if the input signal to the input port changes to any other status.
The configuration of the input interrupt circuit is shown in Figure 4.3.2.

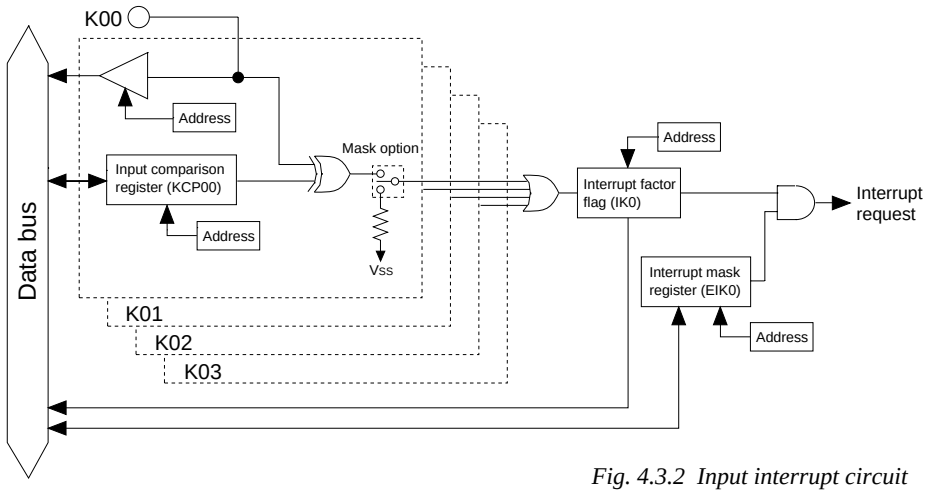



*Fig. 4.3.2 Input interrupt circuit*

## *3 Input port pull-up resistor*

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

--- INPUT PORT PULL-UP RESISTOR ---
        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↵

        K00             1. WITH RESISTOR  SELECTED
        K01             1. WITH RESISTOR  SELECTED
        K02             1. WITH RESISTOR  SELECTED
        K03             1. WITH RESISTOR  SELECTED
```

Select whether each input port (K00–K03) will be supplemented with a pull-up resistor or not.
When "GATE DIRECT" is selected, see to it that an entry floating state does not occur.
Moreover, when "WITH RESISTOR" has been selected and when input ports are changed from low level (VSS) to high level (VDD) with pull-up resistor, the rise of the waveform is delayed on account of the time constant of the pull-up resistor and input gate capacitance. Hence, when fetching input ports, be sure to wait an appropriate amount of time with the program.
For unused ports, select "WITH RESISTOR".
The configuration of the pull-up resistor circuit is shown in Figure 4.3.3.

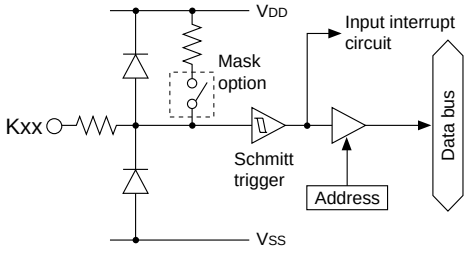



*Fig. 4.3.3 Configuration of pull-up resistor circuit*

## *4 R02 output specification*

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

--- R02 OUTPUT SPEC. ---
        R02             1. D.C.
                        2. FOUT

PLEASE SELECT NO.(1) ? 1↲

--- FOUT OR SIF FREQ. ---
        FREQ.           1. 1/64 OSC1
                        2.  1/8 OSC1
                        3.      OSC1
                        4.      OSC3

PLEASE SELECT NO.(1) ? 1↲

        R02             1. D.C.  SELECTED
        FREQ.           1. 1/64 OSC1  SELECTED
```

For the output specification of the R02 port, select either "DC" output or "FOUT" output.
When "DC" output is selected, the R02 port functions the same as the R00 and R01 ports.
When "FOUT" output is selected, the clock is output from the R02 port with the frequency selected from the following selection items.
"FOUT" output is controlled using the R02 register (A0H•D2); low level is output when the register is "0" and the "FOUT" signal is output when the register is "1".
In this option, frequency for "FOUT" or "SIF" should be selected. When "FOUT" output is selected for the R02 output specification, the selected frequency is used for the "FOUT" signal. The selected frequency is also used for the SIF (serial interface) clock to send/receive data when the SIF is in the master mode.
Because the selected frequency is used for the SIF even when "DC" output is selected for the R02 output specification, a frequency must be selected.
The configuration of the R02 port is shown in Figure 4.3.4.

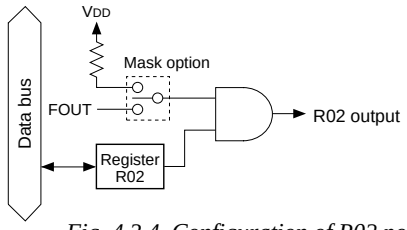



*Fig. 4.3.4 Configuration of R02 port*

## *5 R03 output specification*

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

--- R03 OUTPUT SPEC. ---
        R03             1. D.C.
                        2. BZ 1/16 OSC1
                        3. BZ  1/8 OSC1

PLEASE SELECT NO.(1) ? 1↲

        R03             1. D.C.  SELECTED
```

Select the output specification of the R03 port from the "DC" output, "BZ 1/16 OSC1" or "BZ 1/8 OSC1".
When "DC" output is selected, the R03 port functions the same as the R00 and R01 ports.
When "BZ 1/16 OSC1" or "BZ 1/8 OSC1" is selected, the buzzer drive waveform with the selected frequency is output from the R03 port.
"BZ" output is controlled using the R03 register (A0H•D3); low level is output when the register is "0" and the "BZ" signal is output when the register is "1".
The configuration of the R02 port is shown in Figure 4.3.5.

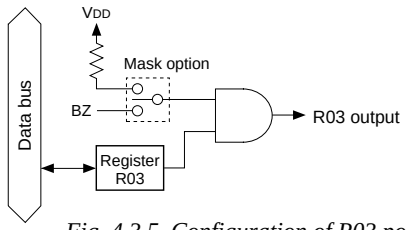



*Fig. 4.3.5 Configuration of R03 port*

## 6 Output port output specification

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

--- OUTPUT PORT OUTPUT SPEC. ---
        R00             1. COMPLEMENTARY
                        2. NCH OPEN DRAIN

PLEASE SELECT NO.(1) ? 1↵

        R01             1. COMPLEMENTARY
                        2. NCH OPEN DRAIN

PLEASE SELECT NO.(1) ? 1↵

        R02             1. COMPLEMENTARY
                        2. NCH OPEN DRAIN

PLEASE SELECT NO.(1) ? 1↵

        R03             1. COMPLEMENTARY
                        2. NCH OPEN DRAIN

PLEASE SELECT NO.(1) ? 1↵

        R00             1. COMPLEMENTARY  SELECTED
        R01             1. COMPLEMENTARY  SELECTED
        R02             1. COMPLEMENTARY  SELECTED
        R03             1. COMPLEMENTARY  SELECTED
```

Select the output specification of each output port (R00–R03).
Either "COMPLEMENTARY" output or "Nch OPEN DRAIN" output can be selected.
When the output port is used as a common output for key matrix, select "Nch OPEN DRAIN" output.
For unused output ports, select "COMPLEMENTARY" output.
The configuration of the output circuit is shown in Figure 4.3.6.

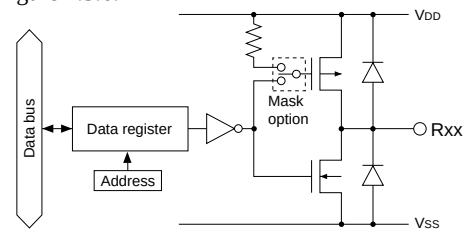



*Fig. 4.3.6 Configuration of output circuit*

## 7 I/O port pull-up resistor

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

--- I/O PORT PULL-UP RESISTOR ---
        P00             1. WITH RESISTOR
                        2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1↵

        P01             1. WITH RESISTOR
                        2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1↵

        P02             1. WITH RESISTOR
                        2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1↵

        P03             1. WITH RESISTOR
                        2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1↵

        P10             1. WITH RESISTOR
                        2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1↵

        P11             1. WITH RESISTOR
                        2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1↵

        P12             1. WITH RESISTOR
                        2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1↵

        P13             1. WITH RESISTOR
                        2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1↵

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

Select whether each I/O port (P00–P03, P10–P13) will be supplemented with a pull-up resistor that functions when the I/O port is in the input mode or not.
When "GATE DIRECT" is selected, see to it that an entry floating state does not occur.
Moreover, when "WITH RESISTOR" has been selected and when input terminals are changed from low level (VSS) to high level (VDD) with pull-up resistor, the rise of the waveform is delayed on account of the time constant of the pull-up resistor and input gate capacitance. Hence, when fetching input signals, be sure to wait an appropriate amount of time with the program.
For unused ports, select "WITH RESISTOR".
However, when the I/O port is in output mode, the port is not pulled up even if "WITH RESISTER" is selected.
The configuration of the I/O port is shown in Figure 4.3.7.

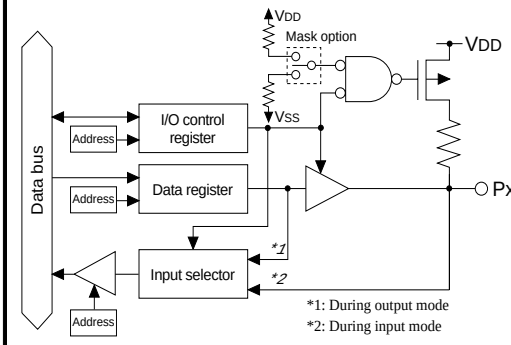



*Fig. 4.3.7 Configuration of I/O port*

## *8 SVD (Supply Voltage Detection circuit) voltage*

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

--- S.V.D VOLTAGE ---
                        1. 2.4[V]
                        2. 2.5[V]
                        3. 2.6[V]
                        4. 2.7[V]

PLEASE SELECT NO.(1) ? 1↲

                        1. 2.4[V]  SELECTED
```

Select criteria voltage for supply voltage detection.

## *9 LCD drive duty*

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

--- LCD DRIVE DUTY ---
                        1. 1/2
                        2. 1/3
                        3. 1/4

PLEASE SELECT NO.(1) ? 1↲

                        1. 1/2  SELECTED
```

Select drive duty for LCD panel from 1/2 duty, 1/3 duty or 1/4 duty.
The LCD panel is driven with the selected duty.

## 4.4 FOG6292 Quick Reference

### ■ Starting command and input/output files

***Execution file:*** FOG6292.EXE

***Starting command:*** **FOG6292**↵ ↵ *indicates the Return key.*

***Input file:*** C292XXXF.DOC (Function option document file, when modifying)

***Output file:*** C292XXXF.DOC (Function option document file)
C292XXXF.HEX (Function option HEX file)

### ■ Display example

```
     ***  E0C6292 FUNCTION OPTION GENERATOR. --- Ver 3.14  ***

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 1994  SEIKO EPSON CORP.

         THIS SOFTWARE MAKES NEXT FILES.

            C292XXXF.HEX  ...  FUNCTION OPTION HEX FILE.
            C292XXXF.DOC  ...  FUNCTION OPTION DOCUMENT FILE.

                        STRIKE ANY KEY.
```

***Start-up message***
When FOG6292 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.

```
*** E0C6292 USER'S OPTION SETTING. --- Ver 3.14 ***

CURRENT DATE IS  94/01/19
PLEASE INPUT NEW DATE  :   94/01/20↵
```

***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 FOG6292.

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

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

PLEASE SELECT NO.? 1↵
PLEASE INPUT FILE NAME? C2920A0↵                              ..(1)
PLEASE INPUT USER'S NAME? SEIKO EPSON CORP.↵                  ..(2)
PLEASE INPUT ANY COMMENT
(ONE LINE IS 50 CHR)? TOKYO DESIGN CENTER↵                    ..(3)
                    ? 421-8 HINO HINO-SHI TOKYO 191 JAPAN↵
                    ? TEL 0425-84-2551↵
                    ? FAX 0425-84-8512↵
                    ? ↵
```

***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? C2920A0↵
EXISTS OVERWRITE(Y/N)? N↵
PLEASE INPUT FILE NAME? C2920B0↵
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) ***

C2920A0         C2920B0         C2920C0                 ..(1)

PLEASE INPUT FILE NAME? C2920A0↵                        ..(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? C2920N0↵
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 FOG6292 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 OSCILLATOR ---
                            1. CR
                            2. CERAMIC

PLEASE SELECT  NO.(1) ? 1↵

                            1. CR  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 EVA621C, 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 EVA621C 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

```
* E0C6292 FUNCTION OPTION DOCUMENT  V 3.14
*
* FILE NAME    C292TS1F.DOC
* USER'S NAME  SEIKO EPSON CORP.
* INPUT DATE   1994/01/19
*
*
*
* OPTION NO.1
* < OSC3 OSCILLATOR >
*                            CR  ------------------  SELECTED
 OPT0101 01
*
* OPTION NO.2
* < INTERRUPT (K00-K03) >
*     K00                    USE  ------------------  SELECTED
*     K01                    USE  ------------------  SELECTED
*     K02                    USE  ------------------  SELECTED
*     K03                    USE  ------------------  SELECTED
 OPT0201 01
 OPT0202 01
 OPT0203 01
 OPT0204 01
*
* OPTION NO.3
* < INPUT PORT PULL-UP RESISTOR >
*     K00                    WITH RESISTOR  --------  SELECTED
*     K01                    WITH RESISTOR  --------  SELECTED
*     K02                    WITH RESISTOR  --------  SELECTED
*     K03                    WITH RESISTOR  --------  SELECTED
 OPT0301 01
 OPT0302 01
 OPT0303 01
 OPT0304 01
*
* OPTION NO.4
* < R02 OUTPUT SPEC. >
*     R02                    D.C.  ----------------  SELECTED
* < FOUT OR SIF FERQ. >
*     FERQ.                  1/64 OSC1  ------------  SELECTED
 OPT0401 01
 OPT0402 01
*
* OPTION NO.5
* < R03 OUTPUT SPEC. >
*     R03                    D.C.  ----------------  SELECTED
 OPT0501 01
*
* OPTION NO.6
* < OUTPUT PORT OUTPUT SPEC. >
*     R00                    COMPLEMENTARY  --------  SELECTED
*     R01                    COMPLEMENTARY  --------  SELECTED
*     R02                    COMPLEMENTARY  --------  SELECTED
*     R03                    COMPLEMENTARY  --------  SELECTED
```

```
 OPT0601 01
 OPT0602 01
 OPT0603 01
 OPT0604 01
*
* OPTION NO.7
* < I/O PORT OUTPUT PULL-UP RESISTOR >
*     P00                    WITH RESISTOR  --------  SELECTED
*     P01                    WITH RESISTOR  --------  SELECTED
*     P02                    WITH RESISTOR  --------  SELECTED
*     P03                    WITH RESISTOR  --------  SELECTED
*     P10                    WITH RESISTOR  --------  SELECTED
*     P11                    WITH RESISTOR  --------  SELECTED
*     P12                    WITH RESISTOR  --------  SELECTED
*     P13                    WITH RESISTOR  --------  SELECTED
 OPT0701 01
 OPT0702 01
 OPT0703 01
 OPT0704 01
 OPT0705 01
 OPT0706 01
 OPT0707 01
 OPT0708 01
*
* OPTION NO.8
* < S.V.D VOLTAGE >
*                            2.4[V]  ---------------  SELECTED
 OPT0801 01
*
* OPTION NO.9
* < LCD DRIVE DUTY >
*                            1/2  ------------------  SELECTED
 OPT0901 01
*
*
*
* SEIKO EPSON'S AREA
*
*
*
* OPTION NO.10
 OPT1001 01
*
* OPTION NO.11
 OPT1101 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 SOG6292

## 5.1 SOG6292 Outline

With the 4-bit single-chip E0C6292 microcomputers, the customer may select the LCD segment options. By modifying the mask patterns of the E0C6292 according to the selected options, the system can be customized to meet the specifications of the target system.
The Segment Option Generator SOG6292 is a software tool for generating data file used to generate mask patterns. From the data file created with SOG6292, the E0C6292 mask pattern is automatically generated by a general purpose computer.
The HEX file for the evaluation board (EVA621C) segment option ROM is simultaneously generated with the data file.

(The EVA621C can be used as the EVA board for the E0C6292 by modifying the function option HEX data generated from the FOG6292.)

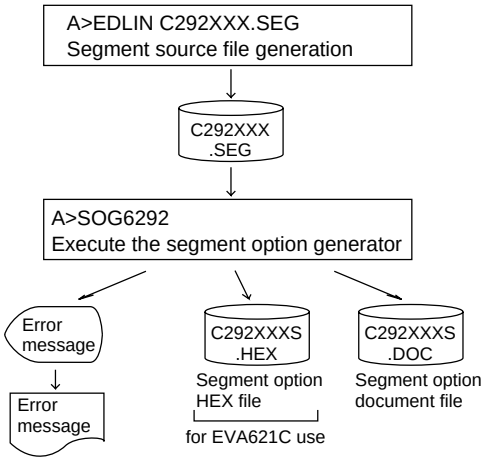



*Fig. 5.1.1 SOG6292 execution flow*

☞ *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 | | | |
| | H | L | D | H | L | D | H | L | D | H | L | D | |
| SEG0 | | | | | | | | | | | | | SEG output |
| SEG1 | | | | | | | | | | | | | DC output ☐ C ☐ N |
| SEG2 | | | | | | | | | | | | | SEG output |
| SEG3 | | | | | | | | | | | | | DC output ☐ C ☐ N |
| SEG4 | | | | | | | | | | | | | SEG output |
| SEG5 | | | | | | | | | | | | | DC output ☐ C ☐ N |
| SEG6 | | | | | | | | | | | | | SEG output |
| SEG7 | | | | | | | | | | | | | DC output ☐ C ☐ N |
| SEG8 | | | | | | | | | | | | | SEG output |
| SEG9 | | | | | | | | | | | | | DC output ☐ C ☐ N |
| SEG10 | | | | | | | | | | | | | SEG output |
| SEG11 | | | | | | | | | | | | | DC output ☐ C ☐ N |
| SEG12 | | | | | | | | | | | | | SEG output |
| SEG13 | | | | | | | | | | | | | DC output ☐ C ☐ N |
| SEG14 | | | | | | | | | | | | | SEG output |
| SEG15 | | | | | | | | | | | | | DC output ☐ C ☐ N |
| SEG16 | | | | | | | | | | | | | SEG output |
| SEG17 | | | | | | | | | | | | | DC output ☐ C ☐ N |
| SEG18 | | | | | | | | | | | | | SEG output |
| SEG19 | | | | | | | | | | | | | DC output ☐ C ☐ N |
| SEG20 | | | | | | | | | | | | | SEG output |
| SEG21 | | | | | | | | | | | | | DC output ☐ C ☐ N |

Legend: <ADDRESS>
H: High order address (6 or 7)
L: Low order address (0–F)
D: Data bit (0–3)

<OUTPUT SPECIFICATION>
C: Complementary output
N: Nch 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–SEG21, segment output and DC output can be selected in units of two terminals. 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 Nch 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 (160H–17FH) can be allocated to the optional segment. With this, up to 88 segments (66 or 44 segments when 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 (6 or 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–COM3.

Examples

- When 1/4 duty is selected

```
0  600  601  602  603  S
1  610  611  612  613  S
```

- When 1/3 duty is selected

```
0  600  601  602  ---  S
1  610  611  612  ---  S
```

### ■ When DC output is selected

The DC output can be selected in units of two terminals and up to 22 terminals may be allocated for DC output. Also, either complementary output or Nch open drain output is likewise selected in units of two terminals. 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–COM3 becomes ineffective. Write three hyphens ("---") in the COM1–COM3 columns in the option list.

Example

- When complementary output is set to SEG18 and SEG19, and Nch open drain output is set to SEG20 and SEG21.

```
18  720  ---  ---  ---  C
19  730  ---  ---  ---  C
20  740  ---  ---  ---  N
21  750  ---  ---  ---  N
```

## 5.4 SOG6292 Quick Reference

### ■ Starting command and input/output files

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

***Execution file:*** SOG6292.EXE

***Starting command:*** **SOG6292_ [-H]↵**

***Option:*** -H: Specifies the segment option document file for input file of SOG6292.

***Input file:*** C292XXX.SEG (Segment option source file)
C292XXXS.DOC (Segment option document file, when -H option use)

***Output file:*** C292XXXS.DOC (Segment option document file)
C292XXXS.HEX (Segment option HEX file)

### ■ Display example

```
     ***  E0C6292 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 1993  SEIKO EPSON CORP.

         SEGMENT OPTION SOURCE FILE NAME IS " C292XXX.SEG "

         THIS SOFTWARE MAKES NEXT FILES.

            C292XXXS.HEX  ...  SEGMENT OPTION HEX FILE.
            C292XXXS.DOC  ...  SEGMENT OPTION DOCUMENT FILE.

                        STRIKE ANY KEY.
```

***Start-up message***
When SOG6292 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.

```
*** E0C6292 USER'S OPTION SETTING. --- Ver 3.21 ***

CURRENT DATE IS  94/01/19
PLEASE INPUT NEW DATE  :   94/01/20↵
```

***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) ***

C2920A0         C2920B0         C2920C0                          ..(1)

PLEASE INPUT SEGMENT OPTION FILE NAME? C2920A0↵                  ..(2)
PLEASE INPUT USER'S NAME? SEIKO EPSON CORP.↵                     ..(3)
PLEASE INPUT ANY COMMENT
(ONE LINE IS 50 CHR)? TOKYO DESIGN CENTER↵                       ..(4)
                    ? 421-8 HINO HINO-SHI TOKYO 191 JAPAN↵
                    ? TEL 0425-84-2551↵
                    ? FAX 0425-84-8512↵
                    ? ↵
```

***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
```

```
PLEASE INPUT SEGMENT OPTION SOURCE FILE NAME? C2920N0↵
SEGMENT OPTION SOURCE FILE IS NOT FOUND.      ..(7) -H option not use

PLEASE INPUT SEGMENT OPTION DOCUMENT FILE NAME? C2920N0↵
SEGMENT OPTION DOCUMENT FILE IS NOT FOUND.    ..(8) -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.
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 EVA621C, 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 EVA621C options.

When a series of operations are complete, the SOG6292 generates files. If no error is committed while setting segment options, "MAKING FILE IS COMPLETED" will be displayed and the SOG6292 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.<br>Though DC output has been selected for output specification, data are described in COM1–COM3. |

## 5.5 Sample Files

### ■ Example of segment option source file

```
;C292TS1.SEG, VER3.21
;EVA621C LCD SEGMENT DECODE TABLE
;
   0  600  601  602  603  S
   1  610  611  612  613  S
   2  620  621  622  623  S
   3  630  631  632  633  S
   4  640  641  642  643  S
   5  650  651  652  653  S
   6  660  661  662  663  S
   7  670  671  672  673  S
   8  680  681  682  683  S
   9  690  691  692  693  S
  10  6A0  6A1  6A2  6A3  S
  11  6B0  6B1  6B2  6B3  S
  12  6C0  6C1  6C2  6C3  S
  13  6D0  6D1  6D2  6D3  S
  14  6E0  6E1  6E2  6E3  S
  15  6F0  6F1  6F2  6F3  S
  16  700  701  702  703  S
  17  710  711  712  713  S
  18  720  721  722  723  S
  19  730  731  732  733  S
  20  740  741  742  743  S
  21  750  751  752  753  S
```

### ■ Example of segment option source file

```
* E0C6292 SEGMENT OPTION DOCUMENT  V 3.21
*
* FILE NAME    C292TS1S.DOC
* USER'S NAME  SEIKO EPSON CORP.
* INPUT DATE   94/01/19
*
*
* OPTION NO.12
*
* < LCD SEGMENT DECODE TABLE >
*
* SEG COM0 COM1 COM2 COM3 SPEC
*
   0  600  601  602  603  S
   1  610  611  612  613  S
   2  620  621  622  623  S
   3  630  631  632  633  S
   4  640  641  642  643  S
   5  650  651  652  653  S
   6  660  661  662  663  S
   7  670  671  672  673  S
   8  680  681  682  683  S
   9  690  691  692  693  S
  10  6A0  6A1  6A2  6A3  S
  11  6B0  6B1  6B2  6B3  S
  12  6C0  6C1  6C2  6C3  S
  13  6D0  6D1  6D2  6D3  S
  14  6E0  6E1  6E2  6E3  S
  15  6F0  6F1  6F2  6F3  S
  16  700  701  702  703  S
  17  710  711  712  713  S
  18  720  721  722  723  S
  19  730  731  732  733  S
  20  740  741  742  743  S
  21  750  751  752  753  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 ICS6292

## 6.1 ICS6292 Outline

The In-circuit Emulator ICE6200 connects the target board produced by the user via the EVA621C and performs real time target system evaluation and debugging by passing through the RS-232C from the host computer and controlling it. The operation on the host computer side and ICE6200 control is done through the ICE Control Software ICS6292.
The ICS6292 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.

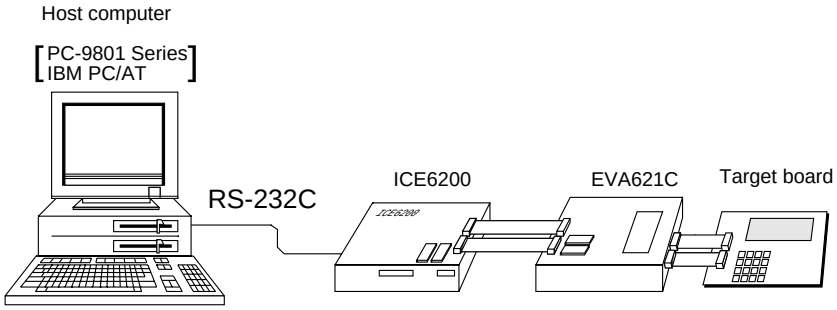



*Fig. 6.1.1 Debugging system using ICE6200*

(The EVA621C can be used as the EVA board for the E0C6292 by modifying the function option HEX data generated from the FOG6292.)

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

## 6.2 ICS6292 Restrictions

Take the following precautions when using the ICS6292.

### ■ ROM Area

The ROM area is limited to a maximum address of 07FFH. Assigning data above the 07FFH address causes an error.

### ■ RAM Area

The RAM area is limited to a maximum address of 1FFH. However, as the following addresses are in the unused area, designation of this area with the ICE commands produces an error.

- 82H–90H, 93H–9FH, A1H–AFH, B1H, B3H, B5H, B7H–BFH, C0H, C2H–C5H, C7H, CBH–CFH, D7H–DFH, E0H–E7H, EBH–EFH, F2H, F5H–F7H, FAH and FDH–FFH in page 0 to 1
- 100H–15FH

(Refer to the "E0C6292 Technical Manual" for details.)

### ■ Undefined Code

The SLP instruction is not specified for the E0C6292 and so cannot be used.

### ■ OPTLD Command

In the ICS6292, OPTLD command can be used.
This command is used to load HEX files (function option and segment option data for LCD) in the EVA621C memory with the ICE6200.

Load of function option data: `#OPTLD,1,C292XXX↵`
Load of segment option data: `#OPTLD,2,C292XXX↵`

## 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 EVA621C 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 EVA621C segment option data memory.
It is HEX file output by the segment option generator and has intel HEX format.
Loading time of a segment option HEX data is dependent on the OSC3 system clock. It takes about 5 to 10 minutes.

**Examples**

`#OPTLD,1,C292XXX↵` .... C292XXXF.HEX file is loaded in the function option data memory.

`#OPTLD,2,C292XXX↵` .... C292XXXS.HEX file is loaded in the segment option data memory.

## 6.3 ICS6292 Quick Reference

### ■ Starting command and input/output files

↵ *indicates the Return key.*

***Execution file:***
ICS6292.BAT (ICS6292J.EXE) . . . for MS-DOS
ICS6292B.BAT (ICS6292W.EXE) . . . for PC-DOS

***Starting command:***
**ICS6292 (ICS6292J)↵** . . . for MS-DOS
**ICS6292B (ICS6292W)↵** . . . for PC-DOS

***Input file:***
C292XXXL.HEX (Object file, low-order)
C292XXXH.HEX (Object file, high-order)
C292XXXD.HEX (Data RAM file)
C292XXXC.HEX (Control file)
C292XXXF.HEX (Function option HEX file)
C292XXXS.HEX (Segment option HEX file)

***Output file:***
C292XXXL.HEX (Object file, low-order)
C292XXXH.HEX (Object file, high-order)
C292XXXD.HEX (Data RAM file)
C292XXXC.HEX (Control file)

### ■ Display example

```
          *** E0C6292 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   PPPPPPPPP      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 ICS6292 is started, the start-up message is displayed, and a self-test is automatically performed. ICS6292 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 ICS6292 program is terminated by entering the Q (Quit) command.

*Note* *Confirm that the cables connected properly, then operate the ICS6292.*

### ■ Error messages

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

## ■ ICE6200 commands

| Item No. | Function | Command Format | Outline of Operation |
|---|---|---|---|
| 1 | Assemble | #A,a ↵ | Assemble command mnemonic code and store at address "a" |
| 2 | Disassemble | #L,a1,a2 ↵ | Contents of addresses a1 to a2 are disassembled and displayed |
| 3 | Dump | #DP,a1,a2 ↵ | Contents of program area a1 to a2 are displayed |
| | | #DD,a1,a2 ↵ | Content of data area a1 to a2 are displayed |
| 4 | Fill | #FP,a1,a2,d ↵ | Data d is set in addresses a1 to a2 (program area) |
| | | #FD,a1,a2,d ↵ | Data d is set in addresses a1 to a2 (data area) |
| 5 | Set Run Mode | #G,a ↵ | Program is executed from the "a" address |
| | | #TIM ↵ | Execution time and step counter selection |
| | | #OTF ↵ | On-the-fly display selection |
| 6 | Trace | #T,a,n ↵ | Executes program while displaying results of step instruction from "a" address |
| | | #U,a,n ↵ | Displays only the final step of #T,a,n |
| 7 | Break | #BA,a ↵ | Sets Break at program address "a" |
| | | #BAR,a ↵ | Breakpoint is canceled |
| | | #BD ↵ | Break condition is set for data RAM |
| | | #BDR ↵ | Breakpoint is canceled |
| | | #BR ↵ | Break condition is set for EVA621C CPU internal registers |
| | | #BRR ↵ | Breakpoint is canceled |
| | | #BM ↵ | Combined break conditions set for program data RAM address and registers |
| | | #BMR ↵ | Cancel combined break conditions for program data ROM address and registers |
| | | #BRES ↵ | All break conditions canceled |
| | | #BC ↵ | Break condition displayed |
| | | #BE ↵ | Enter break enable mode |
| | | #BSYN ↵ | Enter break disable mode |
| | | #BT ↵ | Set break stop/trace modes |
| | | #BRKSEL,REM ↵ | Set BA condition clear/remain modes |
| 8 | Move | #MP,a1,a2,a3 ↵ | Contents of program area addresses a1 to a2 are moved to addresses a3 and after |
| | | #MD,a1,a2,a3 ↵ | Contents of data area addresses a1 to a2 are moved to addresses a3 and after |
| 9 | Data Set | #SP,a ↵ | Data from program area address "a" are written to memory |
| | | #SD,a ↵ | Data from data area address "a" are written to memory |
| 10 | Change CPU Internal Registers | #DR ↵ | Display EVA621C CPU internal registers |
| | | #SR ↵ | Set EVA621C CPU internal registers |
| | | #I ↵ | Reset EVA621C 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 ↵<br>#HSR,a ↵ | Retrieves and indicates the history information which wrote or read the data area address "a" |
| 12 | File | #RF,file ↵ | Move program file to memory |
| | | #RFD,file ↵ | Move data file to memory |
| | | #VF,file ↵ | Compare program file and contents of memory |
| | | #VFD,file ↵ | Compare data file and contents of memory |
| | | #WF,file ↵ | Save contents of memory to program file |
| | | #WFD,file ↵ | Save contents of memory to data file |
| | | #CL,file ↵ | Load ICE6200 set condition from file |
| | | #CS,file ↵ | Save ICE6200 set condition to file |
| | | #OPTLD,1,file ↵<br>#OPTLD,2,file ↵ | Load function option data from file<br>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 ICE6200 instruction |
| 17 | Self Diagnosis | #CHK ↵ | Report results of ICE6200 self diagnostic test |

↵ means press the RETURN key.

# 7 MASK DATA CHECKER MDC6292

## 7.1 MDC6292 Outline

The Mask Data Checker MDC6292 is a software tool which checks the program data (C292XXXH.HEX and C292XXXL.HEX) and option data (C292XXXF.DOC and C292XXXS.DOC) created by the user and creates the data file (C6292XXX.PAn) for generating mask patterns. The user must send the file generated through this software tool to Seiko Epson.

Moreover, MDC6292 has the capability to restore the generated data file (C6292XXX.PA0) to the original file format.

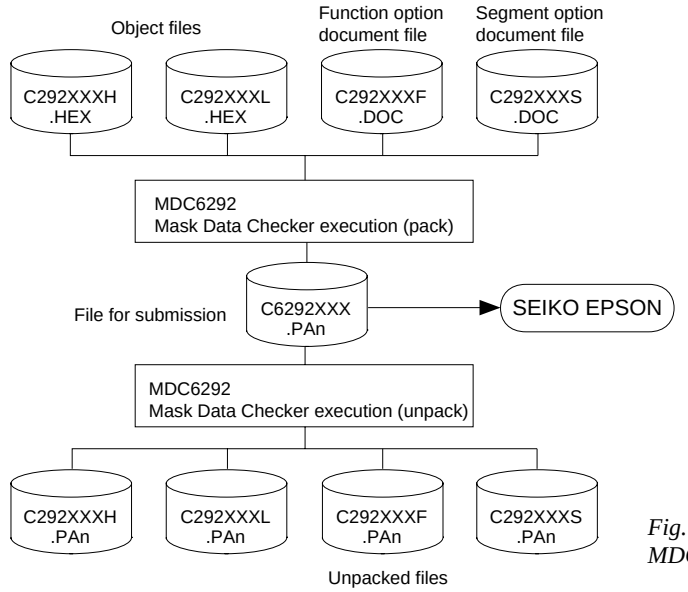



*Fig. 7.1.1*
*MDC6292 execution flow*

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

## 7.2 MDC6292 Quick Reference

### ■ Starting command and input/output files

***Execution file:*** MDC6292.EXE

***Starting command:*** **MDC6292**↵ ↵ *indicates the Return key.*

***Input file:***

| | | |
|---|---|---|
| C292XXXL.HEX | (Object file, low-order) | When packing |
| C292XXXH.HEX | (Object file, high-order) | |
| C292XXXF.DOC | (Function option document file) | |
| C292XXXS.DOC | (Segment option document file) | |
| C6292XXX.PAn | (Packed file) | When unpacking |

***Output file:***

| | | |
|---|---|---|
| C6292XXX.PAn | (Packed file) | When packing |
| C292XXXL.PAn | (Object file, low-order) | When unpacking |
| C292XXXH.PAn | (Object file, high-order) | |
| C292XXXF.PAn | (Function option document file) | |
| C292XXXS.PAn | (Segment option document file) | |

## ■ Display examples

```
          *** E0C6292 PACK / UNPACK PROGRAM Ver 2.001 ***

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 1993 SEIKO EPSON CORP.

                   --- OPERATION MENU ---

                          1. PACK
                          2. UNPACK

                   PLEASE SELECT NO.?
```

***Start-up message***

When MDC6292 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)

       C292XXXH.HEX --------+
                            |
       C292XXXL.HEX --------+
                            |-------- C292XXX.PAn (PACK FILE)
       C292XXXF.DOC --------+
                            |
       C292XXXS.DOC --------+

PLEASE INPUT PACK FILE NAME (C6292XXX.PAn) ?  C62920A0.PA0↵   ...(2)

       C2920A0H.HEX --------+
                            |
       C2920A0L.HEX --------+
                            |-------- C2920A0.PA0
       C2920A0F.DOC --------+
                            |
       C2920A0S.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 "C6292XXX.PA0" has been submitted, the pack file name should be entered as "C6292XXX.PA1".)

With this, the mask file (C6292XXX.PAn) is generated, and the MDC6292 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 (ASM6292) 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 (C6292XXX.PAn) ?  C62920A0.PA0↵...(2)

                            +-------- C2920A0H.PA0
                            |
                            +-------- C2920A0L.PA0
       C62920A0.PA0 --------|
                            +-------- C2920A0F.PA0
                            |
                            +-------- C2920A0S.PA0
```

***Unpacking of data***

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

(2) Enter the packed file name.

With this, the mask data file (C6292XXX.PAn) is restored to the original file format, and the MDC6292 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 | FOG6292, SOG6292 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. E0C6292 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<br>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<br>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)<br>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)<br>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)<br>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, *I* | 1 | 0 | 0 | 1 | *I*7 | *I*6 | *I*5 | *I*4 | *I*3 | *I*2 | *I*1 | *I*0 | | | | | 5 | M(X)←*I*3~*I*0, M(X+1)←*I*7~*I*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

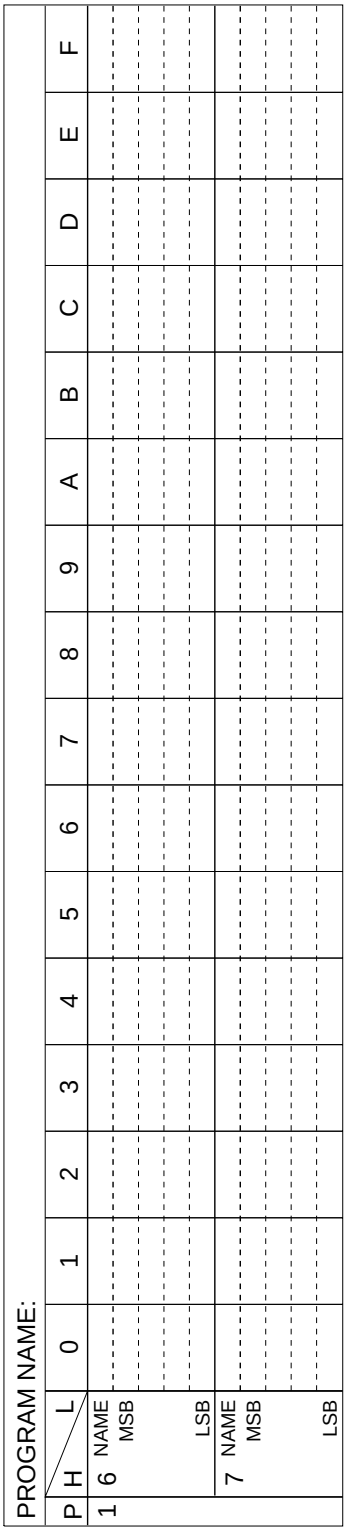
# APPENDIX B. E0C6292 RAM MAP

*RAM map - 1 (000H–07FH)*

PROGRAM NAME:

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

*Display memory map (160H–17FH)*



PROGRAM NAME:

| P | H \ L | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 01 | 8 | NAME | | | | | | | | | | | | | | | | |
| | | MSB | 0 | 0 | | | | | | | | | | | | | | |
| | | | 0 | 0 | | | | | | | | | | | | | | |
| | | | SVDDT | CLKCHG | | | | | | | | | | | | | | |
| | | LSB | SVDON | OSCC | | | | | | | | | | | | | | |
| | 9 | NAME | | | | | | | | | | | | | | | | |
| | | MSB | | K03 | KCP03 | | | | | | | | | | | | | |
| | | | | K02 | KCP02 | | | | | | | | | | | | | |
| | | | | K01 | KCP01 | | | | | | | | | | | | | |
| | | LSB | | K00 | KCP00 | | | | | | | | | | | | | |
| | A | NAME | | | | | | | | | | | | | | | | |
| | | MSB | R03 | | | | | | | | | | | | | | | |
| | | | R02 | | | | | | | | | | | | | | | |
| | | | R01 | | | | | | | | | | | | | | | |
| | | LSB | R00 | | | | | | | | | | | | | | | |
| | B | NAME | | | | | | | | | | | | | | | | |
| | | MSB | IOC03 | | P03 | | IOC13 | | P13 | | | | | | | | | |
| | | | IOC02 | | P02 | | IOC12 | | P12 | | | | | | | | | |
| | | | IOC01 | | P01 | | IOC11 | | P11 | | | | | | | | | |
| | | LSB | IOC00 | | P00 | | IOC10 | | P10 | | | | | | | | | |
| | C | NAME | | | | | | | | | | | | | | | | |
| | | MSB | | 0 | | | | | 0 | | 0 | TM3 | TM7 | | | | | |
| | | | | 0 | | | | | 0 | | 0 | TM2 | TM6 | | | | | |
| | | | | LOFF | | | | | 0 | | TMRUN | TM1 | TM5 | | | | | |
| | | LSB | | LPWR | | | | | WDRST | | TMRST | TM0 | TM4 | | | | | |
| | D | NAME | | | | | | | | | | | | | | | | |
| | | MSB | DBGOS | 0 | OVTBC | MC03 | MC07 | MC11 | MC15 | | | | | | | | | |
| | | | RFDBG | 0 | OVMC | MC02 | MC06 | MC10 | MC14 | | | | | | | | | |
| | | | RFCLK | SENS1 | RFRUN | MC01 | MC05 | MC09 | MC13 | | | | | | | | | |
| | | LSB | 0 | SENS0 | 0 | MC00 | MC04 | MC08 | MC12 | | | | | | | | | |
| | E | NAME | | | | | | | | | | | | | | | | |
| | | MSB | | | | | | | | | SCS0 | SD3 | SD7 | | | | | |
| | | | | | | | | | | | 0 | SD2 | SD6 | | | | | |
| | | | | | | | | | | | SCTRG | SD1 | SD5 | | | | | |
| | | LSB | | | | | | | | | ESIF | SD0 | SD4 | | | | | |
| | F | NAME | | | | | | | | | | | | | | | | |
| | | MSB | 0 | 0 | | 0 | 0 | | | | 0 | 0 | | 0 | 0 | | | |
| | | | 0 | 0 | | 0 | EIT2 | | | | 0 | 0 | | 0 | IT2 | | | |
| | | | 0 | 0 | | 0 | EIT1 | | | | 0 | 0 | | 0 | IT1 | | | |
| | | LSB | EISIF | EIRF | | EIK0 | EIT0 | | | | ISIF | IRF | | IK0 | IT0 | | | |

**Unused area**

# APPENDIX C. E0C6292 I/O MEMORY MAP

*I/O memory map - 1*

| Address *6 | Register D3 | D2 | D1 | D0 | Name | Init *1 | 1 | 0 | Comment |
|---|---|---|---|---|---|---|---|---|---|
| 80H | 0 | 0 | SVDDT | SVDON | 0 *5 | – *2 | | | Unused |
| | | | | | 0 *5 | – *2 | | | Unused |
| | R | | | R/W | SVDDT | 0 | Low | Normal | Supply voltage detaction data |
| | | | | | SVDON | 0 | On | Off | SVD circuit On/Off |
| 81H | 0 | 0 | CLKCHG | OSCC | 0 *5 | – *2 | | | Unused |
| | | | | | 0 *5 | – *2 | | | Unused |
| | R | | R/W | | CLKCHG | 0 | OSC3 | OSC1 | CPU clock switch |
| | | | | | OSCC | 0 | On | Off | OSC3 scillation On/Off |
| 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 | 1 | ↓ | ↑ | Input comparison register (K00–K03) |
| | | | | | KCP02 | 1 | ↓ | ↑ | |
| | R/W | | | | KCP01 | 1 | ↓ | ↑ | |
| | | | | | KCP00 | 1 | ↓ | ↑ | |
| A0H | R03 (BZ) | R02 (FOUT) | R01 | R00 | R03 | 0 | High | Low | Output port R03 |
| | | | | | (BZ) | 0 | On | Off | BZ On/Off (selected by mask option) |
| | | | | | R02 | 0 | High | Low | Output port R02 |
| | R/W | | | | (FOUT) | 0 | On | Off | FOUT On/Off (selected by mask option) |
| | | | | | R01 | 0 | High | Low | Output port R01 |
| | | | | | R00 | 0 | High | Low | Output port R00 |
| 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 | |
| B2H | P03 | P02 | P01 | P00 | P03 | 1 | High | Low | I/O port P00–P03 |
| | | | | | P02 | 1 | High | Low | |
| | R/W | | | | P01 | 1 | High | Low | |
| | | | | | P00 | 1 | High | Low | |
| B4H | IOC13 | IOC12 | IOC11 | IOC10 | IOC13 | 0 | Output | Input | I/O control register (P10–P13) |
| | | | | | IOC12 | 0 | Output | Input | |
| | R/W | | | | IOC11 | 0 | Output | Input | |
| | | | | | IOC10 | 0 | Output | Input | |
| B6H | P13 | P12 | P11 | P10 | P13 | 1 | High | Low | I/O port P10–P13 |
| | | | | | P12 | 1 | High | Low | |
| | R/W | | | | P11 | 1 | High | Low | |
| | | | | | P10 | 1 | High | Low | |

***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 Page switching in I/O memory is not necessary

*I/O memory map - 2*

| Address *6 | Register D3 | D2 | D1 | D0 | Name | Init *1 | 1 | 0 | Comment |
|---|---|---|---|---|---|---|---|---|---|
| C1H | 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 |
| C6H | 0 | 0 | 0 | WDRST | 0 *5 | – *2 | | | Unused |
| | | | | | 0 *5 | – *2 | | | Unused |
| | R | | | W | 0 *5 | – *2 | | | Unused |
| | | | | | WDRST*5 | Reset | Reset | – | Watchdog timer reset |
| C8H | 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 |
| C9H | 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) |
| CAH | 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) |
| D0H | DBGOS | RFDBG | RFCLK | 0 | DBGOS | 0 | Sensor | Ref.-R | R/F converter debugging element selection |
| | | | | | RFDBG | 0 | Debug | Normal | R/F converter debug mode selection |
| | R/W | | | R | RFCLK | 0 | OSC3 | OSC1 | R/F converter clock source selection |
| | | | | | 0 *5 | – *2 | | | Unused |
| D1H | 0 | 0 | SENS1 | SENS0 | 0 *5 | – *2 | | | Unused |
| | | | | | 0 *5 | – *2 | | | Unused |
| | R | | R/W | | SENS1 | 0 | | | R/F converter sensor selection |
| | | | | | SENS0 | 0 | | | 0=SEN0, 1=SEN1, 2&3=SEN2 |
| D2H | OVTBC | OVMC | RFRUN | 0 | OVTBC | 0 | Overflow | Non-over | Time base counter overflow flag |
| | | | | | OVMC | 0 | Overflow | Non-over | Measurement counter overflow flag |
| | R/W | | | R | RFRUN | 0 | Run | Stop | R/F converter Run/Stop |
| | | | | | 0 *5 | – *2 | | | Unused |
| D3H | MC03 | MC02 | MC01 | MC00 | MC03 | – *3 | | | Measurement counter MC00–MC03 |
| | | | | | MC02 | – *3 | | | |
| | R/W | | | | MC01 | – *3 | | | |
| | | | | | MC00 | – *3 | | | |
| D4H | MC07 | MC06 | MC05 | MC04 | MC07 | – *3 | | | Measurement counter MC04–MC07 |
| | | | | | MC06 | – *3 | | | |
| | R/W | | | | MC05 | – *3 | | | |
| | | | | | MC04 | – *3 | | | |
| D5H | MC11 | MC10 | MC09 | MC08 | MC11 | – *3 | | | Measurement counter MC08–MC11 |
| | | | | | MC10 | – *3 | | | |
| | R/W | | | | MC09 | – *3 | | | |
| | | | | | MC08 | – *3 | | | |
| D6H | MC15 | MC14 | MC13 | MC12 | MC15 | – *3 | | | Measurement counter MC12–MC15 |
| | | | | | MC14 | – *3 | | | |
| | R/W | | | | MC13 | – *3 | | | |
| | | | | | MC12 | – *3 | | | |

*I/O memory map - 3*

| address *6 | Register D3 | D2 | D1 | D0 | Name | Init *1 | 1 | 0 | Comment |
|---|---|---|---|---|---|---|---|---|---|
| E8H | SCS0 | 0 | SCTRG | ESIF | SCS0 | 0 | FOUT | Slave | Serial I/F clock mode selection |
| | | | | | 0 *5 | – *2 | | | Unused |
| | | | | | SCTRG(W) | – *2 | Trigger | – | Serial I/F clock trigger (writing) |
| | R/W | R | R/W | | SCTRG(R) | 0 | Run | Stop | Serial I/F clock status (reading) |
| | | | | | ESIF | 0 | SIF port | I/O port | P1 port function selection |
| E9H | SD3 | SD2 | SD1 | SD0 | SD3 | – *2 | | | MSB |
| | | | | | SD2 | – *2 | | | Serial I/F data (low-order 4 bits) |
| | R/W | | | | SD1 | – *2 | | | |
| | | | | | SD0 | – *2 | | | LSB |
| EAH | SD7 | SD6 | SD5 | SD4 | SD7 | – *2 | | | MSB |
| | | | | | SD6 | – *2 | | | Serial I/F data (high-order 4 bits) |
| | R/W | | | | SD5 | – *2 | | | |
| | | | | | SD4 | – *2 | | | LSB |
| F0H | 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 I/F) |
| F1H | 0 | 0 | 0 | EIRF | 0 *5 | – *2 | | | Unused |
| | | | | | 0 *5 | – *2 | | | Unused |
| | R | | | R/W | 0 *5 | – *2 | | | Unused |
| | | | | | EIRF | 0 | Enable | Mask | Interrupt mask register (R/F converter) |
| F3H | 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) |
| F4H | 0 | EIT2 | EIT1 | EIT0 | 0 *5 | – *2 | | | Unused |
| | | | | | EIT2 | 0 | Enable | Mask | Interrupt mask register (Clock timer 2 Hz) |
| | R | R/W | | | EIT1 | 0 | Enable | Mask | Interrupt mask register (Clock timer 8 Hz) |
| | | | | | EIT0 | 0 | Enable | Mask | Interrupt mask register (Clock timer 32 Hz) |
| F8H | 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 I/F) |
| F9H | 0 | 0 | 0 | IRF | 0 *5 | – *2 | | | Unused |
| | | | | | 0 *5 | – *2 | | | Unused |
| | R | | | | 0 *5 | – *2 | | | Unused |
| | | | | | IRF *4 | 0 | Yes | No | Interrupt factor flag (R/F converter) |
| FBH | 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) |
| FCH | 0 | IT2 | IT1 | IT0 | 0 *5 | – *2 | | | Unused |
| | | | | | IT2 *4 | 0 | Yes | No | Interrupt factor flag (Clock timer 2 Hz) |
| | R | | | | IT1 *4 | 0 | Yes | No | Interrupt factor flag (Clock timer 8 Hz) |
| | | | | | IT0 *4 | 0 | Yes | No | Interrupt factor flag (Clock timer 32 Hz) |

# APPENDIX D. TROUBLESHOOTING

| Tool | Problem | Remedy measures |
|---|---|---|
| ICE6200 | Nothing appears on the screen, or nothing works, after activation. | Check the following and remedy if necessary:<br>• Is the RS-232C cable connected correctly?<br>• Is the RS-232C driver installed?<br>• Is SPEED.COM or MODE.COM on the disk?<br>• Is the execution file correct?<br>MS-DOS ICS6292J.EXE<br>PC-DOS ICS6292W.EXE<br>• Is the DOS version correct?<br>MS-DOS Ver. 3.1 or later<br>PC-DOS Ver. 2.1 or later<br>• Is the DIP switches that set the baud rate of the main ICE6200 unit set correctly?<br>• Is the breaker of the ICE6200 set to ON? |
| | The ICE6200 breaker tripped immediately after activation. | Check the following and remedy if necessary:<br>• Are connectors F1 and F5 connected to the EVA621C correctly?<br>• Is the target board power short-circuiting? |
| | <ILLEGAL VERSION ICE6200> appears on the screen immediately after activation. | The wrong version of ICE6200 is being used. Use the latest version. |
| | <ILLEGAL VERSION PARAMETER FILE> appears on the screen immediately after activation. | The wrong version of ICS6292P.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).<br>*Example:* `LD A, B` Data in the B register is loaded into the A register.<br>`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. |
| SOG6292 | An R error occurs although the address is correctly set in the segment source file. | Check the following and remedy if necessary:<br>• Does the address symbol use capital letters? |

| Tool | Problem | Remedy measures |
|---|---|---|
| ASM6292 | 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. |
| MDC6292 | Activation is impossible. | Check the following and remedy if necessary:<br>• Is the number of files set at ten or more in OS environment file CONFIG.SYS? |
| EVA621C | The EVA621C does not work when it is used independently. | Check the following and remedy if necessary:<br>• Has the EPROM for F.HEX and S.HEX been replaced by the EPROM for the target?<br>• Is the EPROM for F.HEX and S.HEX installed correctly?<br>• Is the appropriate voltage being supplied? (5V DC, 3A, or more)<br>• Are the program ROMs (H and L) installed correctly?<br>• Is data written from address 4000H? (When the 27C256 is used as the program ROM) |
| | Target segment does not light. | Check the following and remedy if necessary:<br>• Is an EPROM with an access time of 250 ns or less being used for S.HEX.<br>• Has the VADJ VR inside the EVA621C top cover been turned to a lower setting? |

# EPSON International Sales Operations

## AMERICA

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

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

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

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

## EUROPE

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

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

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

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

## ASIA

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

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

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

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

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

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

- JAPAN -
**SEIKO EPSON CORPORATION**
**ELECTRONIC DEVICES MARKETING DIVISION**

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

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

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


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

EPSON

SEIKO EPSON CORPORATION
ELECTRONIC DEVICES MARKETING DIVISION

■ Electronic devices information on the Epson WWW server

http://www.epson.co.jp

Issue APRIL 1998 Printed in Japan Ⓜ A