

CMOS 4-BIT SINGLE CHIP MICROCOMPUTER
E0C6S32 DEVELOPMENT TOOL MANUAL



NOTICE

No part of this material may be reproduced or duplicated in any form or by any means without the written permission of Seiko Epson. Seiko Epson reserves the right to make changes to this material without notice. Seiko Epson does not assume any liability of any kind arising out of any inaccuracies contained in this material or due to its application or use in any product or circuit and, further, there is no representation that this material is applicable to products requiring high level reliability, such as medical products. Moreover, no license to any intellectual property rights is granted by implication or otherwise, and there is no representation or warranty that anything made in accordance with this material will be free from any patent or copyright infringement of a third party. This material or portions thereof may contain technology or the subject relating to strategic products under the control of the Foreign Exchange and Foreign Trade Control Law of Japan and may require an export license from the Ministry of International Trade and Industry or other approval from another government agency. Please note that "E0C" is the new name for the old product "SMC". If "SMC" appears in other manuals understand that it now reads "E0C".

MS-DOS and EDLIN are registered trademarks of Microsoft Corporation, U.S.A.

PC-DOS, PC/AT, VGA, EGA and IBM are registered trademarks of International Business Machines Corporation, U.S.A.

NEC PC-9801 Series and NEC are registered trademarks of NEC Corporation.

All other product names mentioned herein are trademarks and/or registered trademarks of their respective owners.

E0C6S32 Development Tool Manual

PREFACE

This manual mainly explains the outline of the development support tool for the 4-bit Single Chip Micro-computer E0C6S32.

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.

<i>Development tools</i>	☞ E0C62 Family Development Tool Reference Manual EVA6S32 Manual ICE6200 Hardware Manual
<i>Development procedure</i>	☞ E0C62 Family Technical Guide
<i>Device (E0C6S32)</i>	☞ E0C6S32 Technical Manual
<i>Instructions</i>	☞ E0C6200/6200A Core CPU Manual

CONTENTS

1	<i>COMPOSITION OF DEVELOPMENT SUPPORT TOOL</i>	1
1.1	<i>Configuration of DEV6S32</i>	1
1.2	<i>Developmental Environment</i>	1
1.3	<i>Development Flow</i>	2
1.4	<i>Production of Execution Disk</i>	3
2	<i>DEVELOPMENT TOOL MANAGEMENT SYSTEM DMS6200</i>	4
2.1	<i>DMS6200 Outline</i>	4
2.2	<i>DMS6200 Quick Reference</i>	4
3	<i>CROSS ASSEMBLER ASM6S32</i>	5
3.1	<i>ASM6S32 Outline</i>	5
3.2	<i>E0C6S32 Restrictions</i>	5
3.3	<i>ASM6S32 Quick Reference</i>	6
4	<i>FUNCTION OPTION GENERATOR FOG6S32</i>	8
4.1	<i>FOG6S32 Outline</i>	8
4.2	<i>E0C6S32 Option List</i>	8
4.3	<i>Option Specifications and Selection Message</i>	10
4.4	<i>FOG6S32 Quick Reference</i>	17
4.5	<i>Sample File</i>	19
5	<i>SEGMENT OPTION GENERATOR SOG6S32</i>	22
5.1	<i>SOG6S32 Outline</i>	22
5.2	<i>Option List</i>	23
5.3	<i>Segment Ports Output Specifications</i>	24
5.4	<i>SOG6S32 Quick Reference</i>	25
5.5	<i>Sample Files</i>	27

6 ICE CONTROL SOFTWARE ICS6S32	29
6.1 ICS6S32 Outline	29
6.2 ICS6S32 Restrictions	29
6.3 ICS6S32 Quick Reference	31
7 MASK DATA CHECKER MDC6S32	34
7.1 MDC6S32 Outline	34
7.2 MDC6S32 Quick Reference	34
APPENDIX A. E0C6S32 INSTRUCTION Set	37
B. TROUBLESHOOTING	41
C. DEBUGGING FOR 1/2 DUTY DRIVE ON EVA6S32	43

1 COMPOSITION OF DEVELOPMENT SUPPORT TOOL

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

1.1 Configuration of DEV6S32

The below software are included in the product of the E0C6S32 development support tool DEV6S32.

1. Development Tool Management System DMS6200 Menu selection for each software / start-up software
2. Cross Assembler ASM6S32 Cross assembler for program preparation
3. Function Option Generator FOG6S32 Function option data preparation program
4. Segment Option Generator SOG6S32 Segment option data preparation program
5. ICE Control Software ICS6S32 ICE control program
6. Mask Data Checker MDC6S32 Mask data preparation program

1.2 Developmental Environment

The software product of the development support tool DEV6S32 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 E0C6S32, the above-mentioned host computer, editor, P-ROM writer, printer, etc. must be prepared by the user in addition to the development tool which is normally supported by Seiko Epson.

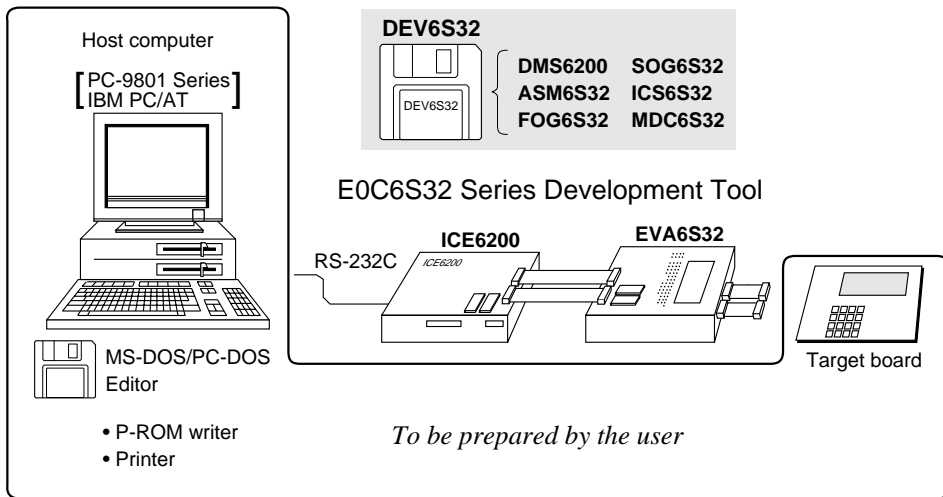


Fig. 1.2.1 System configuration

Note The DEV6S32 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 DEV6S32.

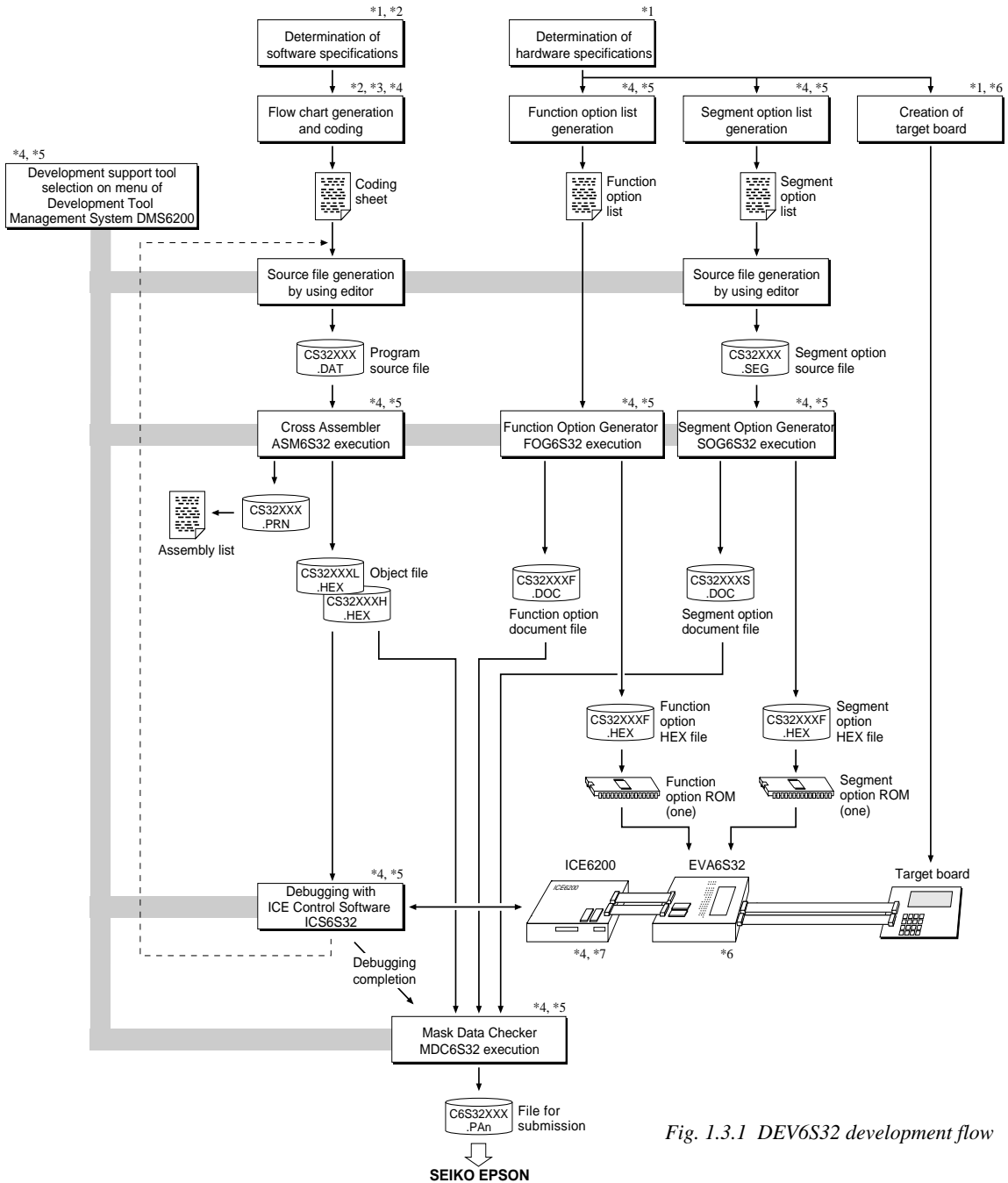


Fig. 1.3.1 DEV6S32 development flow

Concerning file names

All the input-output file name for the each development support tool commonly use "CS32XXX". In principle each file should be produced in this manner. Seiko Epson will designate the "XXX" for each customer.

Reference Manual

- *1 E0C6S32 Technical Hardware Manual
- *2 E0C6S32 Technical Software Manual
- *3 E0C6200/6200A Core CPU Manual
- *4 E0C62 Family Development Tool Reference Manual
- *5 E0C6S32 Development Tool Manual (this manual)
- *6 EVA6S32 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 DEV6S32 floppy disk.

The content of the files contained in the DEV6S32 floppy disk are shown below.

PC-DOS version	MS-DOS version	Contents
ASM6S32.EXE	ASM6S32.EXE	Cross Assembler execution file
DMS6200.EXE	DMS6200.EXE	Development Tool Management System execution file
FOG6S32.EXE	FOG6S32.EXE	Function Option Generator execution file
ICS6S32B.BAT	ICS6S32.BAT	ICE Control Software batch file
ICS6S32W.EXE	ICS6S32J.EXE	ICE Control Software execution file
ICS6S32P.PAR	ICS6S32P.PAR	ICE Control Software parameter file
MDC6S32.EXE	MDC6S32.EXE	Mask Data Checker execution file
SOG6S32.EXE	SOG6S32.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 subdirectory with an appropriate name (DEV6S32, etc.) then copy the content of the floppy disk into that subdirectory using the COPY command.

- Next make a CONFIG.SYS file using Editor or the like. When a CONFIG.SYS has previously been made using a hard disk system, check the setting of the FILES within it. (If there is none add it.) Set the number of files to be described in CONFIG.SYS at 10 or more, so that the Mask Data Checker MDC6S32 will handle many files.

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

- It is a good idea to copy the editor into the disk to be copied and the subdirectory so you can also select the editor from the DMS6200 menu.
- In "ICS6S32(B).BAT" the batch process is indicated such that the ICS6S32J(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.

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.

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 (DEV6S32), then insert the original disk into the A drive and execute the COPY command.

```
C>\MD DEV6S32 [↵]
```

```
C>\CD DEV6S32 [↵]
```

```
C\DEV6S32>COPY A:*. * [↵]
```

Example:

Setting of FILES (CONFIG.SYS)

```
C>\TYPE CONFIG.SYS [↵]
```

```
:
```

```
FILES=20
```

```
:
```

RS-232C Setting (PC-DOS version)

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

RS-232C Setting (MS-DOS version)

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

2 DEVELOPMENT TOOL MANAGEMENT SYSTEM DMS6200

2.1 DMS6200 Outline

The DMS6200 (Development Tool Management System) is a software which selects the DEV6S32 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.

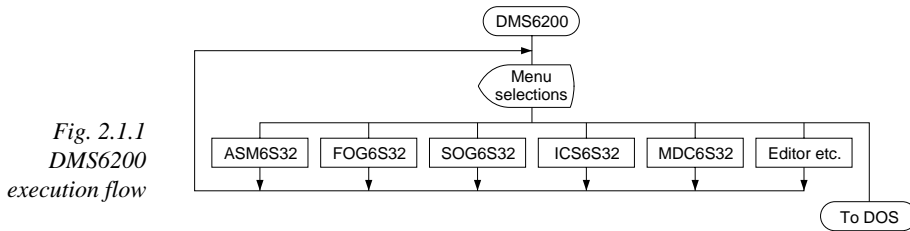


Fig. 2.1.1
DMS6200
execution flow

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

2.2 DMS6200 Quick Reference

■ Starting command

Execution file: **DMS6200.EXE**

Starting command: **DMS6200**

indicates the Return key.

■ Display examples

```

*** E0C6200 Development tool Management System. --- Ver 1.0 ***
EEEEEEEEEE PPPPPPPP SSSSSSS OOOOOOOO NNN NNN
EEEEEEEEEE PPPPPPPPPP SSS SSSS OOO OOO NNNN NNN
EEE PPP PPP SSS SSS OOO OOO NNNNN NNN
EEE PPP PPP SSS OOO OOO NNNNNN NNN
EEEEEEEEEE PPPPPPPPPP SSSSSSS 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 1990 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. 1990.
1) ASM6S32 .EXE
2) FOG6S32 .EXE
3) ICS6S32B.BAT
4) ICS6S32W.EXE
5) MDC6S32 .EXE
6) SOG6S32 .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. 1990.
1) CS32XXX .DAT
2) CS32XXX .PRN
3) CS32XXX .SEG
:
10) C6S32XXX.PAO
Input Number ? [ 1 ]
Edit > {ASM6S32 CS32XXX}
  
```

Source file selection screen

To starting ASM6S32, 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 ASM6S32

3.1 ASM6S32 Outline

The ASM6S32 cross assembler is an assembler program for generating the machine code used by the E0C6S32 4-bit, single-chip microcomputers. The Cross Assembler ASM6S32 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.

☞ *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.*

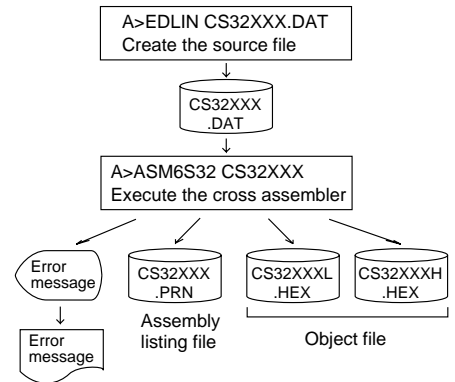


Fig. 3.1.1 ASM6S32 execution flow

3.2 E0C6S32 Restrictions

Note the following when generating a program by the E0C6S32:

■ ROM area

The capacity of the E0C6S32 ROM is 2K steps (0000H to 07FFH). Therefore, the specification range of the memory setting pseudo-instructions and PSET instruction is restricted.

■ RAM area

The capacity of the E0C6S32 RAM is 144 words (000H to 06FH, 080H to 09FH and 4 bits/word). Memory access is invalid when the unused area of the index register is specified.

■ Undefined codes

The following instructions have not been defined in the E0C6S32 instruction sets.

Memory configuration:

Bank: Only bank 0, Page: 8 pages (0 to 7H), 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

Example: LD X, 0A0H A0H is loaded into the IX register, but an unused area has been specified so that the memory accessible with the IX register (MX) is invalid.

LD Y, 0C7H C7H is loaded into the IY register, but an unused area has been specified so that the memory accessible with the IY register (MY) is invalid.

SLP			
PUSH	XP	PUSH	YP
POP	XP	POP	YP
LD	XP,r	LD	YP,r
LD	r,XP	LD	r,YP

3.3 ASM6S32 Quick Reference

■ Starting command and input/output files

_ indicates a blank.

indicates the Return key.

A parameter enclosed by [] can be omitted.

Execution file: ASM6S32.EXE

Starting command: **ASM6S32_ [drive-name:] source-file-name [.shp]_ [-N]**

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

Input file: CS32XXX.DAT (Source file)

Output file: CS32XXXL.HEX (Object file, low-order)
CS32XXXH.HEX (Object file, high-order)
CS32XXX.PRN (Assembly listing file)

■ Display example

```

*** E0C6S32 CROSS ASSEMBLER. --- Ver 2.00 ***
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 PPPPPPPPPP SSSS 000 000 NNN NNNNNN
EEE PPP SSS 000 000 NNN NNNNNN
EEE PPP SSS SSS 000 000 NNN NNNN
EEEEEEEEEE PPP SSSS SSS 000 000 NNN NNN
EEEEEEEEEE PPP SSSSSS 00000000 NNN NN

(C) COPYRIGHT 1989 SEIKO EPSON CORP.
SOURCE FILE NAME IS " CS32XXX.DAT "
THIS SOFTWARE MAKES NEXT FILES.
CS32XXXH.HEX ... HIGH BYTE OBJECT FILE.
CS32XXXL.HEX ... LOW BYTE OBJECT FILE.
CS32XXX .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 ASM6S32 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, ASM6S32 assembles the source file.

To suspend execution, press the "CTRL" and "C" keys together at stage (1) or (2).

■ Operators

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

■ Pseudo-instructions

Pseudo-instruction	Meaning	Example of Use
EQU (Equation)	To allocate data to label	ABC EQU 9 BCD EQU ABC+1
SET (Set)	To allocate data to label (data can be changed)	ABC SET 0001H ABC SET 0002H
DW (Define Word)	To define ROM data	ABC DW 'AB' BCD DW 0FFBH
ORG (Origin)	To define location counter	ORG 100H ORG 256
PAGE (Page)	To define boundary of page	PAGE 1H PAGE 3
SECTION (Section)	To define boundary of section	SECTION
END (End)	To terminate assembly	END
MACRO (Macro)	To define macro	CHECK MACRO DATA LOCAL LOOP
LOCAL (Local)	To make local specification of label during macro definition	LOOP CP MX , DATA JP NZ , LOOP ENDM
ENDM (End Macro)	To end macro definition	CHECK 1

■ 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)	<ul style="list-style-type: none"> The location counter value exceeded the upper limit of the program memory, or a location exceeding the upper limit was specified. A value greater than that which the number of significant digits of the operand will accommodate was specified.
! (Warning)	<ul style="list-style-type: none"> 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 FOG6S32

4.1 FOG6S32 Outline

With the 4-bit single-chip E0C6S32 microcomputers, the customer may select 10 hardware options. By modifying the mask patterns of the E0C6S32 according to the selected options, the system can be customized to meet the specifications of the target system.

The Function Option Generator FOG6S32 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 FOG6S32, the E0C6S32 mask pattern is automatically generated by a general purpose computer.

The HEX file for the evaluation board (EVA6S32) hardware option ROM is simultaneously generated with the data file.

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

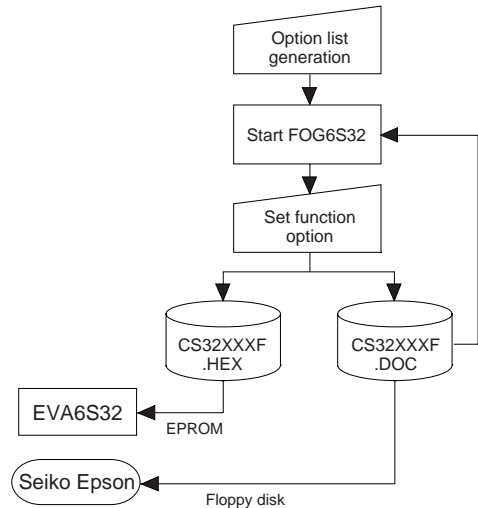


Fig. 4.1.1 FOG6S32 execution flow

4.2 E0C6S32 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. DEVICE TYPE

- 1. E0C6S32
- 2. E0C6SA32
- 3. E0C6SL32
- 4. E0C6SB32

2. OSC3 OSCILLATION CIRCUIT (only for E0C6SA32)

- 1. CR
- 2. Ceramic

3. MULTIPLE KEY ENTRY RESET

- KEY COMBINATION 1. Not Use
 - 2. Use K00, K01
 - 3. Use K00, K01, K02
 - 4. Use K00, K01, K02, K03
- TIME DETECTION 1. Not Use 2. Use

4. WATCH DOG TIMER

- 1. Not Use
- 2. Use

5. INTERRUPT NOISE REJECT

- K00-K03 1. Use 2. Not Use
- K10 1. Use 2. Not Use

6. INPUT K PORT WITH PULL DOWN RESISTOR

- K00 1. With Resistor 2. Gate Direct
- K01 1. With Resistor 2. Gate Direct
- K02 1. With Resistor 2. Gate Direct
- K03 1. With Resistor 2. Gate Direct
- K10 1. With Resistor 2. Gate Direct

7. R00–R03 SPECIFICATION

- R00 1. Complementary 2. Pch-OpenDrain
- R01 1. Complementary 2. Pch-OpenDrain
- R02 1. Complementary 2. Pch-OpenDrain
- R03 1. Complementary 2. Pch-OpenDrain

8. R10 SPECIFICATION

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

9. R11 SPECIFICATION

- OUTPUT SPECIFICATION 1. Complementary 2. Pch-OpenDrain

10.R12 SPECIFICATION

- OUTPUT SPECIFICATION 1. Complementary 2. Pch-OpenDrain
- OUTPUT TYPE 1. DC Output
- 2. FOUT 32768 (HZ)
- 3. FOUT 16384 (HZ)
- 4. FOUT 8192 (HZ)
- 5. FOUT 4096 (HZ)
- 6. FOUT 2048 (HZ)
- 7. FOUT 1024 (HZ)
- 8. FOUT 512 (HZ)
- 9. FOUT 256 (HZ)

11.R13 SPECIFICATION

- OUTPUT SPECIFICATION 1. Complementary 2. Pch-OpenDrain
- OUTPUT TYPE 1. DC Output
- 2. Buzzer Inverted Output (R13 control)
- 3. Buzzer Inverted Output (R10 control)

12. I/O PORT OUTPUT SPECIFICATION

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

13.EVENT COUNTER SPECIFICATION

- NOISE REJECT 1. Not Use 2. 2048 (HZ) 3. 256 (HZ)
- INPUT TERMINAL 1. K10 2. K03

14.LCD SPECIFICATION

- DUTY 1. 1/4 Duty 2. 1/3 Duty 3. 1/2 Duty
- LCD POWER SOURCE 1. Internal 2. Vss = VL1 3. Vss = VL2 4. Vss = VL3
- LCD BIAS 1. 1/3 Bias 2. 1/2 Bias
- LCD REGISTER SWITCHING 1. LCD ALL OFF/NORMAL

15.SEGMENT MEMORY ADDRESS

- 1. 40–6F (R/W) 2. C0–EF (W)

4.3 Option Specifications and Selection Message

Screen that can be selected as function options set on the E0C6S32 are shown below, and their specifications are also described.

1 Device type

```

*** OPTION NO.1 ***

--- DEVICE TYPE ---

      1. E0C6S32
      2. E0C6SA32
      3. E0C6SL32
      4. E0C6SB32

PLEASE SELECT NO. (1) ? 3 [ ]

      3. E0C6SL32  SELECTED
    
```

Select the chip specification. There are four models: E0C6S32 (3 V supply voltage), E0C6SL32, E0C6SB32 (1.5 V supply voltage, low-power specification) and E0C6SA32 (Twin Clock specification). When E0C6S32, E0C6SL32 or E0C6SB32 is selected, oscillation circuit OSC3 is fixed at CR oscillation. However, it can not be used.

2 OSC3 oscillation circuit

```

*** OPTION NO.2 ***

--- OSC3 OSCILLATION CIRCUIT ---

      1. CR OSCILLATION CIRCUIT
      2. CERAMIC OSCILLATION CIRCUIT

PLEASE SELECT NO. (1) ? 1 [ ]

      1. CR OSCILLATION CIRCUIT  SELECTED
    
```

* The above selection is only possible with 6SA32.

Select oscillation circuit that uses OSC3 and OSC4. To minimize external components, CR oscillation circuit would be suitable; to obtain a stable oscillation frequency, ceramic oscillation circuit would be suitable. When CR oscillation circuit is selected, only resistor is needed as external component since capacitie is built-in. On the other hand, when ceramic oscillation circuit is selected, ceramic oscillator, gate capacity and drain capacity are needed as external components. Although when ceramic oscillation circuit is selected, it is fixed at 1 MHz, when CR oscillation circuit is selected, frequency may be modified to a certain extent depending on the resistance of external component.

3 Multiple key entry reset

```

*** OPTION NO.3 ***

--- MULTIPLE KEY ENTRY RESET ---

KEY COMBINATION      1. NOT USE
                    2. USE K00,K01
                    3. USE K00,K01,K02
                    4. USE K00,K01,K02,K03

PLEASE SELECT NO. (1) ? 2 [ ]

TIME DETECTION CIRCUIT  1. NOT USE
                       2. USE

PLEASE SELECT NO. (1) ? 2 [ ]

KEY COMBINATION      2. USE K00,K01  SELECTED
TIME DETECTION CIRCUIT  2. USE      SELECTED
    
```

The reset function and time detection circuit are set when K00 through K03 are entered.

- **Key combination**
When "Not Use" is set for the combination, the reset function is not activated even if K00 through K03 are entered. When "Use K00, K01" is set, the system is reset immediately the K00 and K01 inputs go high at the same time. Similarly, the system is reset as soon as the K00 through K02 inputs or the K00 through K03 inputs go high.

- **Time detection circuit**
When "Use" is set for the time detection circuit, a simultaneous high input time is authorized. The system is reset when a signal is input for more than 1 to 3 sec.

If the time detection circuit is not used, the system is reset when a high signal is input for more than 6 msec.

- * Even if "Not Use" is set for the combination, the time detection selection is required.

The system reset circuit is shown in Figure 4.3.1.

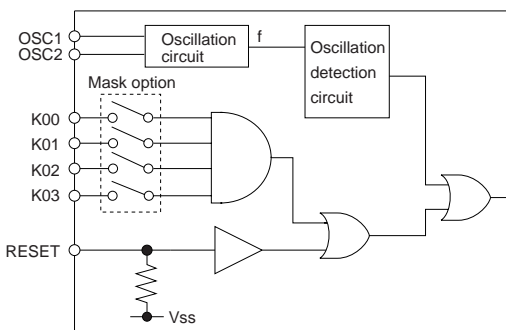


Fig. 4.3.1 System reset circuit

4 Watchdog timer

```

*** OPTION NO.4 ***

--- WATCH DOG TIMER ---

                                1. NOT USE
                                2. USE

PLEASE SELECT NO.(1) ? 1 

                                1. NOT USE  SELECTED

```

Select whether the watchdog timer built-in to detect CPU runaways will be used or not.

When the watchdog timer is not reset by the program within 3 to 4 second cycles, the CPU is initially reset.

5 Interrupt noise reject

```

*** OPTION NO.5 ***

--- INTERRUPT NOISE REJECT ---

K00-K03                        1. USE
                                2. NOT USE

PLEASE SELECT NO.(1) ? 2 

K10                             1. USE
                                2. NOT USE

PLEASE SELECT NO.(1) ? 2 

K00-K03                        2. NOT USE  SELECTED
K10                             2. NOT USE  SELECTED

```

Select whether noise rejector will be supplemented to the input interruptor of K00–K03 and K10.

When "Use" is selected, the entry signal will pass the noise rejector, and occurrence of interrupt errors due to noise or chattering can be avoided. Note, however, that because the noise rejector performs entry signal sampling at 4 kHz, "Not Use" should be selected when high speed response is required.

6 Input ports pull down resistor

```

*** OPTION NO.6 ***

--- INPUT K PORT WITH PULL DOWN RESISTOR ---

K00                            1. WITH RESISTOR
                                2. GATE DIRECT

PLEASE SELECT NO.(1) ? 2 

K01                            1. WITH RESISTOR
                                2. GATE DIRECT

PLEASE SELECT NO.(1) ? 2 

K02                            1. WITH RESISTOR
                                2. GATE DIRECT

PLEASE SELECT NO.(1) ? 2 

K03                            1. WITH RESISTOR
                                2. GATE DIRECT

PLEASE SELECT NO.(1) ? 2 

K10                            1. WITH RESISTOR
                                2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1 

K00                            2. GATE DIRECT  SELECTED
K01                            2. GATE DIRECT  SELECTED
K02                            2. GATE DIRECT  SELECTED
K03                            2. GATE DIRECT  SELECTED
K10                            2. GATE DIRECT  SELECTED

```

Select whether input ports (K00–K03 and K10) will each be supplemented with pull down resistors or not. When "Gate Direct" is selected, see to it that entry floating state does not occur. Select "With Resistor" pull down resistor for unused ports.

Moreover, the input port status is changed from high level (VDD) to low (VSS) with pull down resistors, a delay of approximately 1 msec in waveform rise time will occur depending on the pull down resistor and entry load time constant. Because of this, when input reading is to be conducted, ensure the appropriate wait time with the program.

The configuration of the pull down resistor circuit is shown in Figure 4.3.2.

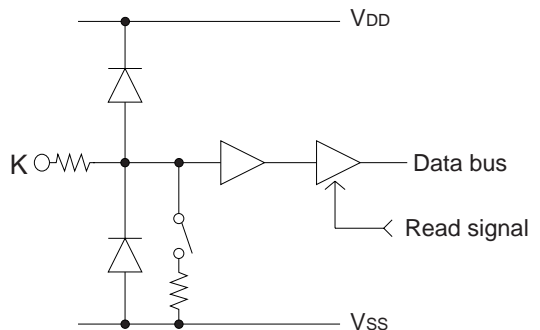


Fig. 4.3.2 Configuration of pull down resistor

7 R00–R03 specification

```

*** OPTION NO.7 ***
--- R00-R03 SPEC. ---

R00 OUTPUT SPEC.    1. COMPLEMENTARY
                   2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1☐

R01 OUTPUT SPEC.    1. COMPLEMENTARY
                   2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1☐

R02 OUTPUT SPEC.    1. COMPLEMENTARY
                   2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1☐

R03 OUTPUT SPEC.    1. COMPLEMENTARY
                   2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1☐

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

Select the output specification for the output ports (R00–R03).

Either complementary output or Pch open drain output may be selected.

When output port is to be used on key matrix configuration, select Pch open drain output.

For unused output ports, select complementary output.

The output circuit configuration is shown in Figure 4.3.3.

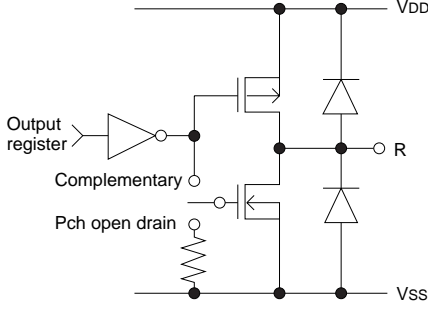


Fig. 4.3.3 Configuration of output circuit

8 R10 specification

```

*** OPTION NO.8 ***
--- R10 SPEC. ---

R10 OUTPUT SPEC.    1. COMPLEMENTARY
                   2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1☐

R10 OUTPUT TYPE     1. D.C.
                   2. BUZZER

PLEASE SELECT NO.(1) ? 2☐

R10 OUTPUT SPEC.    1. COMPLEMENTARY  SELECTED
R10 OUTPUT TYPE     2. BUZZER         SELECTED
    
```

Select the output specification and the output type for the R10 terminal.

- **Output specification**
Either complementary output or Pch open drain output may be selected.

- **Output type**
When DC output is selected, R10 becomes a regular output port. When buzzer output is selected, by writing "1" to the R10 register, buzzer drive (oscillation output) signal is output from the R10 terminal.

* When DC output is selected, the R13 terminal output type (see Option 11, "R13 specification") selection is limited to DC output only.

The circuit configuration is the same as that of output ports (R00–R03 shown in Figure 4.3.3).

Refer to Figure 4.3.6 for buzzer output waveform.

9 R11 specification

```

*** OPTION NO.9 ***
--- R11 SPEC. ---

R11 OUTPUT SPEC.    1. COMPLEMENTARY
                   2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1☐

R11 OUTPUT SPEC.    1. COMPLEMENTARY  SELECTED
    
```

Select the output specification for the R11 terminal. Either complementary output or Pch open drain output may be selected.

The circuit configuration is the same as that of output ports (R00–R03 shown in Figure 4.3.3).

10 R12 specification

```

*** OPTION NO.10 ***
--- R12 SPEC. ---
R12 OUTPUT SPEC.  1. COMPLEMENTARY
                  2. PCH-OPENDRAIN
PLEASE SELECT NO.(1) ? 1
R12 OUTPUT TYPE  1. D.C.
                  2. FOUT 32768 (HZ)
                  3. FOUT 16384 (HZ)
                  4. FOUT 8192 (HZ)
                  5. FOUT 4096 (HZ)
                  6. FOUT 2048 (HZ)
                  7. FOUT 1024 (HZ)
                  8. FOUT 512 (HZ)
                  9. FOUT 256 (HZ)
PLEASE SELECT NO.(1) ? 1
R12 OUTPUT SPEC.  1. COMPLEMENTARY  SELECTED
R12 OUTPUT TYPE  1. D.C.  SELECTED
    
```



Fig. 4.3.4 Output waveform at R12 DC output selection

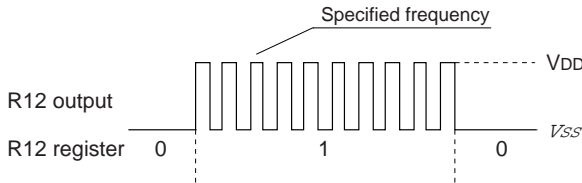


Fig. 4.3.5 Output waveform at R12 FOUT output selection

Select the output specification and the output type for the R12 terminal.

- **Output specification**
Either complementary output or Pch open drain output may be selected.

- **Output type**
Either DC output or FOUT output may be selected.
When DC output is selected, R12 becomes a regular output port.
When R12 register is set to "1", the R12 terminal output goes high (VDD), and goes low (VSS) when set to "0".
Output waveform is shown in Figure 4.3.4.

When FOUT is selected, a clock with a set frequency can be output from the R12 terminal. When FOUT bit (R12 register) is set to "1", 50% duty and VDD-VSS amplitude square wave is generated at the specified frequency. When set to "0", the FOUT terminal goes low (VSS). A FOUT frequency may be selected from among 8 types, ranging from 256 Hz to 32,768 Hz. FOUT output is normally utilized to provide clock to other devices but since hazard occurs at the square wave breaks, great caution must be observed when using it.
Output waveform is shown in Figure 4.3.5.

11 R13 specification

```

*** OPTION NO.11 ***
--- R13 SPEC. ---
R13 OUTPUT SPEC.  1. COMPLEMENTARY
                  2. PCH-OPENDRAIN
PLEASE SELECT NO.(1) ? 1
R13 OUTPUT TYPE  1. D.C.
                  2. NBZ CONTROLLED BY R13
                  3. NBZ CONTROLLED BY R10
PLEASE SELECT NO.(1) ? 1
R13 OUTPUT SPEC.  1. COMPLEMENTARY  SELECTED
R13 OUTPUT TYPE  1. D.C.  SELECTED
    
```

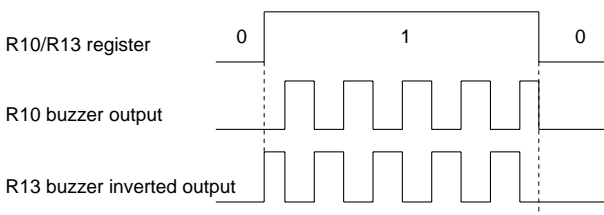


Fig. 4.3.6 Buzzer output waveform

Select the output specification and the output type for the R13 terminal.

- **Output specification**
Either complementary output or Pch open drain output may be selected.

- **Output type**
Either DC output or NBZ output (buzzer inverted output) may be selected.
When DC output is selected, R13 becomes a regular output port.
When "NBZ " is selected, inverted waveform of R10 buzzer output is generated from the R13 terminal. R13 and R10 control bits become buzzer inverted output when "1" is written to R13 and R10 registers, respectively.

* The buzzer inverted output may not be selected when the output type R10 terminal (see Option 8, "R10 specification") is not set to buzzer. Moreover, at this point, when the output type of R10 terminal is reselected after selecting buzzer inverted output, the output type of R10 is fixed at buzzer output.

Buzzer output waveform is shown in Figure 4.3.6.

12 I/O port specification

```

*** OPTION NO.12 ***
--- I/O PORT OUTPUT SPEC. ---

      P00          1. COMPLEMENTARY
                  2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1☐

      P01          1. COMPLEMENTARY
                  2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1☐

      P02          1. COMPLEMENTARY
                  2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1☐

      P03          1. COMPLEMENTARY
                  2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1☐

      P10          1. COMPLEMENTARY
                  2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1☐

      P11          1. COMPLEMENTARY
                  2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1☐

      P12          1. COMPLEMENTARY
                  2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1☐

      P13          1. COMPLEMENTARY
                  2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1☐

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

Select the output specification to be used during I/O ports (P00–P03 and P10–P13) output mode selection. Either complementary output or Pch open drain output may be selected.

The circuit configuration of the output driver is the same as that of output ports (R00–R03 shown in Figure 4.3.3).

Select complementary output for unused ports.

The I/O ports can control the input/output direction according to the IOC bit (7E address, D0 bit, and FE, D0 bit); at "1" and "0" settings, it is set to output port and input port, respectively.

The pull down resistor of this port is turned on by the read signal and is normally turned off to minimize leak current.

Because of this, when the port is set for input, take care that a floating state does not occur in the terminal.

The I/O port circuit configuration is shown in Figure 4.3.7.

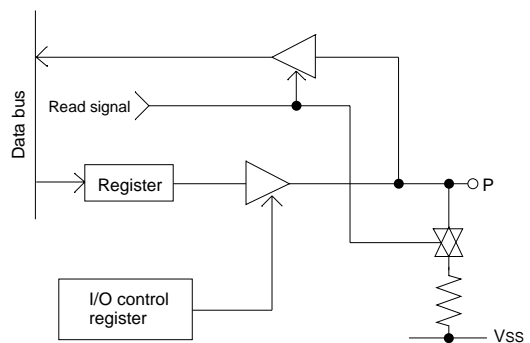


Fig. 4.3.7 Circuit configuration of I/O port

13 Event counter specification

```

*** OPTION NO.13 ***
--- EVENT COUNTER SPEC. ---

      NOISE REJECT 1. NOT USE
                  2. 2048 (HZ)
                  3. 256 (HZ)

PLEASE SELECT NO.(1) ? 1☐

      K10 OR K03  1. K10
                  2. K03

PLEASE SELECT NO.(1) ? 2☐

                  1. NOT USE  SELECTED
                  2. K03     SELECTED
    
```

Select the noise rejector and the input terminal for the event counter.

- **Noise reject**

The event counter is equipped with built-in noise rejector to prevent operational errors caused by noise and chattering in the K10 (K03) terminal. The noise rejector is bypassed when "Not Use" is selected. When using the noise rejector, either 2048 Hz or 256 Hz can be selected as the sampling frequency. Select a frequency suitable for the signal to be input.

- **Input terminal**

Either K10 or K03 may be selected. The phase of the K10 input signal is different from K03.

14 LCD specification

```

*** OPTION NO.14 ***

--- LCD SPEC. ---

      DUTY                1. 1/4 DUTY
                        2. 1/3 DUTY
                        3. 1/2 DUTY

PLEASE SELECT NO.(1) ? 3 [ ]

      LCD POWER SOURCE    1. INTERNAL
                        2. VL1=VSS (INTERNAL CONNECTION)
                        3. VL2=VSS (INTERNAL CONNECTION)
                        4. VL3=VSS (INTERNAL CONNECTION)

PLEASE SELECT NO.(1) ? 2 [ ]

      LCD BIAS            1. 1/3 BIAS
                        2. 1/2 BIAS

PLEASE SELECT NO.(1) ? 2 [ ]

      LCD REGISTER SWITCHING  1. LCD ALL OFF/NORMAL

PLEASE SELECT NO.(1) ? 1 [ ]

      DUTY                3. 1/2 DUTY  SELECTED
LCD POWER SOURCE          2. VL1=VSS (INTERNAL CONNECTION)  SELECTED
LCD BIAS                  2. 1/2 BIAS  SELECTED
LCD REGISTER SWITCHING    1. LCD ALL OFF  SELECTED
    
```

Select the specification (duty, LCD power source, LCD bias, LCD register function) for LCD drive circuit.

• **Duty**

Select the drive duty.

When 1/2 duty is selected, up to 76 segments of LCD panel can be driven with 2 COM terminals and 38 SEG terminals. When 1/3 duty is selected, up to 114 segments can be driven with 3 COM terminals, and when 1/4 duty is selected, up to 152 segments with 4 COM terminals.

Table 4.3.1 Common duty selection standard

Number of segments	Common duty
1-76	1/2
77-114	1/3
115-152	1/4

When 1/2 duty is selected, the COM0 and COM1 terminals are effective for COM output and the COM2 and COM3 terminals always output an off signal. When 1/3 duty is selected, the COM0 to COM2 terminals are effective and the COM3 terminal always outputs an off signal. Refer to Table 4.3.1 for common duty selection.

Figures 4.3.8 and 4.3.9 show the drive waveforms of 1/3 bias driving and 1/2 bias driving, respectively.

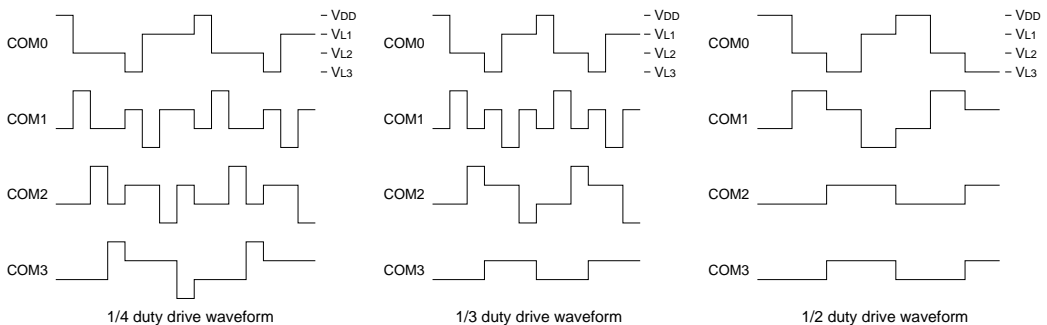


Fig. 4.3.8 Drive waveform from COM terminals (1/3 bias)

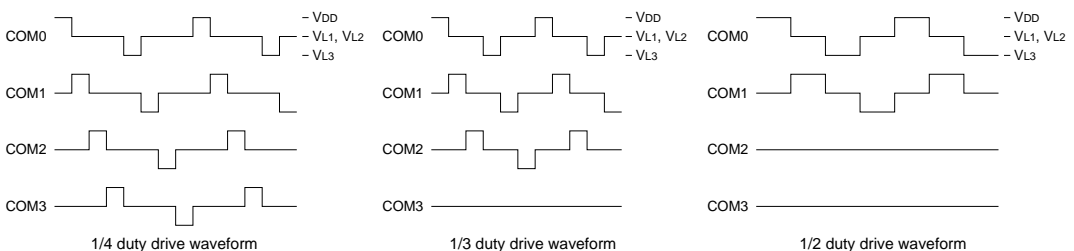


Fig. 4.3.9 Drive waveform from COM terminals (1/2 bias)

• 1/3 bias

	Internal	VL1 = VSS	VL2 = VSS	VL3 = VSS
E0C6S32	3.0 V LCD	×	×	3.0 V LCD
E0C6SA32	3.0 V LCD	×	4.5 V LCD	3.0 V LCD
E0C6SL32	3.0 V LCD	×	×	×
E0C6SB32	3.0 V LCD	×	×	×

• 1/2 bias

	Internal	VL1 = VSS	VL2 = VSS	VL3 = VSS
E0C6S32	×	×	×	3.0 V LCD
E0C6SA32	×	×	×	3.0 V LCD
E0C6SL32	×	3.0 V LCD	×	×

Combinations that are marked with an "X" cannot be selected.

• LCD power source

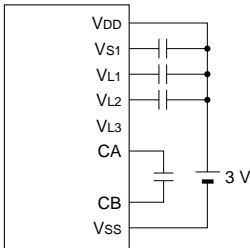
For the LCD power source, either "internal" (to generate internally) or "external" (to supply from outside of the IC) can be selected.

The tables on the left show the voltage of the LCD panel which can be used according to the selection of LCD voltage and model.

Figure 4.3.10 shows the external elements when the LCD system voltage regulator is not used.

• When VL3 = VSS is selected

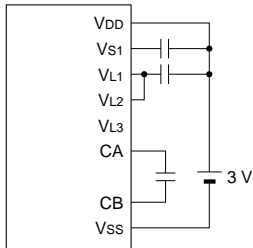
E0C6S32/6SA32
3 V LCD panel
1/4, 1/3, 1/2 duty, 1/3 bias



Note: VL3 is shorted to VSS inside the IC.

• When VL3 = VSS is selected

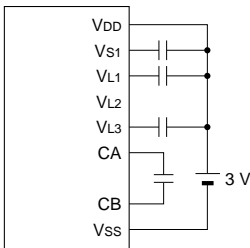
E0C6S32/6SA32
3 V LCD panel
1/4, 1/3, 1/2 duty, 1/2 bias



Note: VL3 is shorted to VSS inside the IC.

• When VL2 = VSS is selected

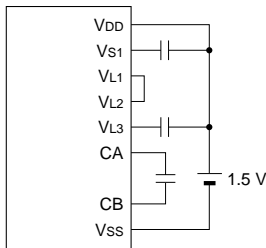
E0C6SA32
4.5 V LCD panel
1/4, 1/3, 1/2 duty, 1/3 bias



Note: VL2 is shorted to VSS inside the IC.

• When VL1 = VSS is selected

E0C6SL32
3 V LCD panel
1/4, 1/3, 1/2 duty, 1/2 bias



Note: VL1 is shorted to VSS inside the IC.

• LCD bias

Select the LCD drive bias.

For the LCD drive bias, either 1/3 bias (drives LCD with 4 levels, VDD, VL1, VL2 and VL3) or 1/2 bias (drives LCD with 3 levels, VDD, VL1 = VL2 and VL3) can be selected.

By selecting 1/2 bias, external elements can be minimized. However, it is limited when the LCD system voltage regulator is not used. Furthermore, when 1/2 bias is selected, be sure to short between the VL1 terminal and the VL2 terminal outside the IC. (See Figure 4.3.10.)

* The EVA board cannot output a 1/2 bias waveform. Be aware that the waveform is fixed at 1/3 bias.

• LCD register function (LCD drive switching)

Only LCD All OFF is selectable.

All the LCD segments go off without changing the display memory by writing "0" to the CSDC register. It is possible to go on by writing "1" to the CSDC register.

Fig. 4.3.10 External elements when LCD system voltage regulator is not used

15 Segment memory address

```

*** OPTION NO.15 ***

--- SEGMENT MEMORY ADDRESS ---

                1. 40 - 6F
                2. C0 - EF

PLEASE SELECT NO.(1) ? 2 [ ]

                2. C0 - EF  SELECTED
    
```

Select the segment memory area.

When "40-6F" is selected for the segment memory area, it is possible to read and write from/to this area because a RAM is assigned to this area.

When "C0-EF" is selected, this segment memory area becomes a write-only area.

4.4 FOG6S32 Quick Reference

■ Starting command and input/output files

Execution file: FOG6S32.EXE

Starting command: **FOG6S32** *indicates the Return key.*

Input file: CS32XXXF.DOC (Function option document file, when modifying)

Output file: CS32XXXF.DOC (Function option document file)
CS32XXXF.HEX (Function option HEX file)

■ Display example

```

*** EOC6S32 FUNCTION OPTION GENERATOR. --- Ver 3.13 ***
EEEEEEEEEE PPPPPPPP SSSSSSS 0000000 NNN NNN
EEEEEEEEEE PPPPPPPP SSS SSSS 000 000 NNNN NNN
EEE PPP PPP SSS SSS 000 000 NNNNN NNN
EEE PPP PPP SSS SSS 000 000 NNNNNN NNN
EEEEEEEEEE PPPPPPPP SSSSSS 000 000 NNN NNN NNN
EEEEEEEEEE PPPPPPPP SSSS 000 000 NNN NNNNN
EEE PPP SSS SSS 000 000 NNN NNNNN
EEE PPP SSS SSS 000 000 NNN NNNN
EEEEEEEEEE PPP SSS SSS 000 000 NNN NNN
EEEEEEEEEE PPP SSSSSS 0000000 NNN NN

(C) COPYRIGHT 1994 SEIKO EPSON CORP.

THIS SOFTWARE MAKES NEXT FILES.

CS32XXXF.HEX ... FUNCTION OPTION HEX FILE.
CS32XXXF.DOC ... FUNCTION OPTION DOCUMENT FILE.

STRIKE ANY KEY.
    
```

```

*** EOC6S32 USER'S OPTION SETTING. --- Ver 3.13 ***
CURRENT DATE IS 95/06/13
PLEASE INPUT NEW DATE : 95/06/15
    
```

```

*** OPERATION SELECT MENU ***
1. INPUT NEW FILE
2. EDIT FILE
3. RETURN TO DOS

PLEASE SELECT NO.?
    
```

```

*** OPERATION SELECT MENU ***
1. INPUT NEW FILE
2. EDIT FILE
3. RETURN TO DOS

PLEASE SELECT NO.? 1
PLEASE INPUT FILE NAME? CS320A0 ..(1)
PLEASE INPUT USER'S NAME? SEIKO EPSON CORP. ..(2)
PLEASE INPUT ANY COMMENT
(ONE LINE IS 50 CHR)? FUJIMI PLANT ..(3)
? 281 FUJIMI SUWA-GUN NAGANO-KEN 399-0200 JAPAN
? TEL 0266-61-1211
? FAX 0266-61-1273
?
    
```

```

PLEASE INPUT FILE NAME? CS320A0
EXISTS OVERWRITE(Y/N)? N
PLEASE INPUT FILE NAME? CS320B0
PLEASE INPUT USER'S NAME?
    
```

Start-up message

When FOG6S32 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.

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 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 FOG6S32.

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.

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 overwriting 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) ***

CS320A0      CS320B0      CS320C0      ..(1)

PLEASE INPUT FILE NAME? CS320A0
PLEASE INPUT USER'S NAME?
PLEASE INPUT ANY COMMENT
(ONE LINE IS 50 CHR)?
PLEASE INPUT EDIT NO.? 4
:
(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? CS320N0
FUNCTION OPTION DOCUMENT FILE IS NOT FOUND.
PLEASE INPUT FILE NAME?
    
```

```

*** OPTION NO.3 ***

--- MULTIPLE KEY ENTRY RESET ---

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

PLEASE SELECT NO.(1) ? 2

KEY COMBINATION      2. Use K00,K01  SELECTED
    
```

```

END OF OPTION SETTING.
DO YOU MAKE HEX FILE (Y/N) ? Y
*** OPTION EPROM SELECT MENU ***

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

PLEASE SELECT NO.? 2

    2. 27C128  SELECTED

MAKING FILE(S) IS COMPLETED.

*** OPERATION SELECT MENU ***

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

PLEASE SELECT NO.?
    
```

Modifying function option settings

Select "2" on the operation selection menu.

- (1) Will display the files on the current drive.
- (2) Enter the file name.
- (3) Enter the customer's company name.
- (4) Enter any comment.

Previously entered data can be used by pressing the RETURN key "" at (3) and (4).

- (5) Enter the number of the function option to be modified. When selection of one option is complete, the system prompts entry of another function option number. Repeat selection until all options to be modified are selected.

Enter "E" to end option setting. Then, move to the confirmation procedure for HEX file generation.

Option selection

The selections for each option correspond one to one to the option list. Enter the selection number.

The value in parentheses () indicates the default value, and is set when only the RETURN key "" is pressed.

In return, the confirmation is displayed.

When you wish to modify previously set function options in the new setting process, enter "B" to return 1 step back to the previous function option setting operation.

EPROM selection

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

- (1) When debugging the program with EVA6S32, 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 EVA6S32 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

```

* E0C6S32 FUNCTION OPTION DOCUMENT V 3.13
*
* FILE NAME      CS320A2F.DOC
* USER'S NAME    SEIKO EPSON CORP.
* INPUT DATE     1995/06/13
*
*
* OPTION NO.1
* < DEVICE TYPE >
*
*                               E0C6SL32  -----  SELECTED
OPT0101 03
*
* OPTION NO.2
* < OSC3 OSCILLATION CIRCUIT >
*
*                               CR OSCILLATION CIRCUIT  -----  SELECTED
OPT0201 01
*
* OPTION NO.3
* < MULTIPLE KEY ENTRY RESET >
*   KEY COMBINATION      USE  K00,K01  -----  SELECTED
*   TIME DETECTION CIRCUIT USE  -----  SELECTED
OPT0301 02
OPT0302 02
*
* OPTION NO.4
* < WATCH DOG TIMER >
*
*                               NOT USE  -----  SELECTED
OPT0401 01
*
* OPTION NO.5
* < INTERRUPT NOISE REJECTOR >
*   K00-K03              NOT USE  -----  SELECTED
*   K10                  NOT USE  -----  SELECTED
OPT0501 01
OPT0502 01
*
* OPTION NO.6
* < INPUT K PORT WITH PULL DOWN RESISTOR >
*   K00                  GATE DIRECT -----  SELECTED
*   K01                  GATE DIRECT -----  SELECTED
*   K02                  GATE DIRECT -----  SELECTED
*   K03                  GATE DIRECT -----  SELECTED
*   K10                  GATE DIRECT -----  SELECTED
OPT0601 02
OPT0602 02
OPT0603 02
OPT0604 02
OPT0605 02
*
* OPTION NO.7
* < R00-R03 SPEC. >
*   R00 OUTPUT SPEC.    COMPLEMENTARY -----  SELECTED
*   R01 OUTPUT SPEC.    COMPLEMENTARY -----  SELECTED
*   R02 OUTPUT SPEC.    COMPLEMENTARY -----  SELECTED
*   R03 OUTPUT SPEC.    COMPLEMENTARY -----  SELECTED
OPT0701 01
OPT0702 01
OPT0703 01
OPT0704 01

```

```

*
* OPTION NO.8
* < R10 SPEC. >
*   R10 OUTPUT SPEC.      COMPLEMENTARY ----- SELECTED
*   R10 OUTPUT TYPE       BZ OUTPUT ----- SELECTED
  OPT0801 01
  OPT0802 02
*
* OPTION NO.9
* < R11 SPEC. >
*   R11 OUTPUT SPEC.      COMPLEMENTARY ----- SELECTED
  OPT0901 01
*
* OPTION NO.10
* < R12 SPEC. >
*   R12 OUTPUT SPEC.      COMPLEMENTARY ----- SELECTED
*   R12 OUTPUT TYPE       FOUT 512 (HZ) ----- SELECTED
  OPT1001 01
  OPT1002 02
  OPT1003 07
*
* OPTION NO.11
* < R13 SPEC. >
*   R13 OUTPUT SPEC.      COMPLEMENTARY ----- SELECTED
*   R13 OUTPUT TYPE       NBZ CONTROLLED BY R13 ----- SELECTED
  OPT1101 01
  OPT1102 02
*
* OPTION NO.12
* < I/O PORT OUTPUT SPEC. >
*   P00                    COMPLEMENTARY ----- SELECTED
*   P01                    COMPLEMENTARY ----- SELECTED
*   P02                    COMPLEMENTARY ----- SELECTED
*   P03                    COMPLEMENTARY ----- SELECTED
*   P10                    COMPLEMENTARY ----- SELECTED
*   P11                    COMPLEMENTARY ----- SELECTED
*   P12                    COMPLEMENTARY ----- SELECTED
*   P13                    COMPLEMENTARY ----- SELECTED
  OPT1201 01
  OPT1202 01
  OPT1203 01
  OPT1204 01
  OPT1205 01
  OPT1206 01
  OPT1207 01
  OPT1208 01
*
* OPTION NO.13
* < EVENT COUNTER SPEC. >
*   NOISE REJECT           NOT USE ----- SELECTED
*   K10 OR K03             K03 ----- SELECTED
  OPT1301 01
  OPT1302 02
*
* OPTION NO.14
* < LCD SPEC. >
*   DUTY                   1/2 DUTY ----- SELECTED
*   LCD POWER SOURCE       VSS=VL1 INTERNAL CONNECTION - SELECTED
*   LCD BIAS               1/2 BIAS (3.0V LCD) ----- SELECTED
*   LCD REGISTER SWITCHING LCD ALL OFF ----- SELECTED
  OPT1401 03
  OPT1402 02
  OPT1403 02
  OPT1404 01

```



```

*
* OPTION NO.15
* < SEGMENT MEMORY ADDRESS >
*
*          C0-EF ----- SELECTED
OPT1501 02
*
*
* SEIKO EPSON'S AREA
*
*
* OPTION NO.16
OPT1601 01
OPT1602 01
OPT1603 01
OPT1604 01
OPT1605 01
OPT1606 01
OPT1607 01
OPT1608 01
*
* OPTION NO.17
OPT1701 01
OPT1702 01
OPT1703 01
OPT1704 01
OPT1705 01
OPT1706 01
OPT1707 01
OPT1708 01
*
* OPTION NO.19
OPT1901 01
*
* OPTION NO.20
OPT2001 01
OPT2002 01
*
* OPTION NO.21
OPT2101 01
OPT2102 01
*
* OPTION NO.22
OPT2201 01
*
* OPTION NO.23
OPT2301 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 SOG6S32

5.1 SOG6S32 Outline

With the 4-bit single-chip E0C6S32 microcomputers, the customer may select the LCD segment options. By modifying the mask patterns of the E0C6S32 according to the selected options, the system can be customized to meet the specifications of the target system.

The Segment Option Generator SOG6S32 is a software tool for generating data file used to generate mask patterns. From the data file created with SOG6S32, the E0C6S32 mask pattern is automatically generated by a general purpose computer. The HEX file for the evaluation board (EVA6S32) segment option ROM is simultaneously generated with the data file.

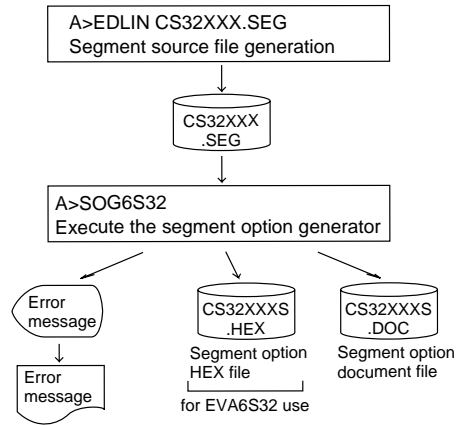


Fig. 5.1.1 SOG6S32 execution flow

☞ 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 <input type="checkbox"/> C <input type="checkbox"/> P
SEG2														SEG output
SEG3														DC output <input type="checkbox"/> C <input type="checkbox"/> P
SEG4														SEG output
SEG5														DC output <input type="checkbox"/> C <input type="checkbox"/> P
SEG6														SEG output
SEG7														DC output <input type="checkbox"/> C <input type="checkbox"/> P
SEG8														SEG output
SEG9														DC output <input type="checkbox"/> C <input type="checkbox"/> P
SEG10														SEG output
SEG11														DC output <input type="checkbox"/> C <input type="checkbox"/> P
SEG12														SEG output
SEG13														DC output <input type="checkbox"/> C <input type="checkbox"/> P
SEG14														SEG output
SEG15														DC output <input type="checkbox"/> C <input type="checkbox"/> P
SEG16														SEG output
SEG17														DC output <input type="checkbox"/> C <input type="checkbox"/> P
SEG18														SEG output
SEG19														DC output <input type="checkbox"/> C <input type="checkbox"/> P
SEG20														SEG output
SEG21														DC output <input type="checkbox"/> C <input type="checkbox"/> P
SEG22														SEG output
SEG23														DC output <input type="checkbox"/> C <input type="checkbox"/> P
SEG24														SEG output
SEG25														DC output <input type="checkbox"/> C <input type="checkbox"/> P
SEG26														SEG output
SEG27														DC output <input type="checkbox"/> C <input type="checkbox"/> P
SEG28														SEG output
SEG29														DC output <input type="checkbox"/> C <input type="checkbox"/> P
SEG30														SEG output
SEG31														DC output <input type="checkbox"/> C <input type="checkbox"/> P
SEG32														SEG output
SEG33														DC output <input type="checkbox"/> C <input type="checkbox"/> P
SEG34														SEG output
SEG35														DC output <input type="checkbox"/> C <input type="checkbox"/> P
SEG36														SEG output
SEG37														DC output <input type="checkbox"/> C <input type="checkbox"/> P
Legend:	<ADDRESS>												<OUTPUT SPECIFICATION>	
	H: High order address, L: Low order address												C: Complementary output	
	D: Data bit												P: Pch open drain output	

Note:

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

5.3 Segment Ports Output Specifications

For the output specification of the segment output ports SEG0–SEG37, 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 Pch open drain output may further be selected.

However, for segment output ports that will not be used, select segment output.

Refer to the "E0C62 Family Development Tool Reference Manual" for the segment option source file creation.

■ **When segment output is selected**

The segment output port has a segment decoder built-in, and the data bit of the optional address in the segment memory area (040H–06FH or 0C0H–0EFH) can be allocated to the optional segment. With this, up to 152 segments (114 segments when 1/3 duty is selected or 76 segments when 1/2 duty is selected) of liquid crystal panel could be driven.

The segment memory may be allocated only one segment and multiple setting is not possible.

The allocated segment displays when the bit for this segment memory is set to "1", and goes out when bit is set to "0".

Segment allocation is set to H for high address (4–6 or C–E), 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 C00 C01 C02 C03 S
1 C10 C11 C12 C13 S
```

- When 1/3 duty is selected

```
0 C00 C01 C02 --- S
1 C10 C11 C12 --- S
```

- When 1/2 duty is selected

```
0 C00 C01 --- --- S
1 C10 C11 --- --- S
```

■ **When DC output is selected**

The DC output can be selected in units of two terminals and up to 38 terminals may be allocated for DC output. Also, either complementary output or Pch open drain output is likewise selected in units of two terminals. When the bit for the selected segment memory is set to "1", the segment output port goes high (VDD), and goes low (VSS) when set to "0". Segment allocation is the same as when segment output is selected but for the while the segment 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 SEG34 and SEG35, and Pch open drain output is set to SEG36 and SEG37.

```
34 E00 --- --- --- C
35 E10 --- --- --- C
36 E20 --- --- --- P
37 E30 --- --- --- P
```

Note Only complementary output is enabled as the DC output of the SEG ports of EVA6S32. Therefore, complementary output is enabled even if Pch open drain output is selected. Respond to it by adding external circuits as required.

5.4 SOG6S32 Quick Reference

■ Starting command and input/output files

Execution file: SOG6S32.EXE

_ indicates a blank.

Starting command: SOG6S32_ [-H]

indicates the Return key.

A parameter enclosed by [] can be omitted.

Option: -H: Specifies the segment option document file for input file of SOG6S32.

Input file: CS32XXX.SEG (Segment option source file)
CS32XXXS.DOC (Segment option document file, when -H option use)

Output file: CS32XXXS.DOC (Segment option document file)
CS32XXXS.HEX (Segment option HEX file)

■ Display example

```

*** E0C6S32 SEGMENT OPTION GENERATOR. --- Ver 3.20 ***
EEEEEEEEEE PPPPPPPP SSSSSSS 00000000 NNN NNN
EEEEEEEEEE PPPPPPPPP SSS SSSS 000 000 NNNN NNN
EEE PPP PPP SSS SSS 000 000 NNNNN NNN
EEE PPP PPP SSS SSS 000 000 NNNNNN NNN
EEEEEEEEEE PPPPPPPPP SSSSSSS 000 000 NNN NNN NNN
EEEEEEEEEE PPPPPPPP SSSS 000 000 NNN NNNNNN
EEE PPP SSS SSS 000 000 NNN NNNNN
EEE PPP SSS SSS 000 000 NNN NNNNN
EEEEEEEEEE PPF SSSS SSS 000 000 NNN NNN
EEEEEEEEEE PPF SSSSSSS 00000000 NNN NN

(C) COPYRIGHT 1990 SEIKO EPSON CORP.

SEGMENT OPTION SOURCE FILE NAME IS " CS32XXX.SEG "

THIS SOFTWARE MAKES NEXT FILES.

CS32XXXS.HEX ... SEGMENT OPTION HEX FILE.
CS32XXXS.DOC ... SEGMENT OPTION DOCUMENT FILE.

STRIKE ANY KEY.

```

```

*** E0C6S32 USER'S OPTION SETTING. --- Ver 3.20 ***
CURRENT DATE IS 95/04/17
PLEASE INPUT NEW DATE : 95/04/19 

```

```

*** SOURCE FILE(S) ***
CS320A0 CS320B0 CS320C0 ..(1)
PLEASE INPUT SEGMENT OPTION FILE NAME? CS320A0  ..(2)
PLEASE INPUT USER'S NAME? SEIKO EPSON CORP.  ..(3)
PLEASE INPUT ANY COMMENT
(ONE LINE IS 50 CHR)? FUJIMI PLANT  ..(4)
? 281 FUJIMI SUWA-GUN NAGANO-KEN 399-0200 JAPAN 
? TEL 0266-61-1211 
? FAX 0266-61-1273 
? 

```

```

*** 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? CS320N0 
SEGMENT OPTION SOURCE FILE IS NOT FOUND. ..(7) -H option not use
PLEASE INPUT SEGMENT OPTION DOCUMENT FILE NAME? CS320N0 
SEGMENT OPTION DOCUMENT FILE IS NOT FOUND. ..(8) -H option use

```

Start-up message

When SOG6S32 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.

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.

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.

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 EVA6S32, 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 EVA6S32 options.

When a series of operations are complete, the SOG6S32 generates files. If no error is committed while setting segment options, "MAKING FILE IS COMPLETED" will be displayed and the SOG6S32 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 (Duplication Error)	The same data (SEG port No., segment memory address, or data bit) was specified more then once.
Out Port Set Error	The output specifications were not set in units of two ports.

5.5 Sample Files

■ Example of segment option source file

```

; CS320A2.SEG, VER.3.20
; EVA6S32 LCD SEGMENT DECODE TABLE
;
0   C01  C00  C32  E20  S
1   C12  C11  C10  C23  S
2   C13  C20  C21  C22  S
3   E00  C02  C30  C31  S
4   C41  C40  C72  E21  S
5   C52  C51  C50  C63  S
6   C53  C60  C61  C62  S
7   E01  C42  C70  C71  S
8   C81  C80  CB2  E22  S
9   C92  C91  C90  CA3  S
10  C93  CA0  CA1  CA2  S
11  E02  C82  CB0  CB1  S
12  CC1  CC0  CF2  E23  S
13  CD2  CD1  CD0  CE3  S
14  CD3  CE0  CE1  CE2  S
15  E03  CC2  CF0  CF1  S
16  D01  D00  D32  E30  S
17  D12  D11  D10  D23  S
18  D13  D20  D21  D22  S
19  E10  D02  D30  D31  S
20  D41  D40  D72  E31  S
21  D52  D51  D50  D63  S
22  D53  D60  D61  D62  S
23  E11  D42  D70  D71  S
24  D81  D80  DB2  E32  S
25  D92  D91  D90  DA3  S
26  D93  DA0  DA1  DA2  S
27  E12  D82  DB0  DB1  S
28  DC1  DC0  DF2  E33  S
29  DD2  DD1  DD0  DE3  S
30  DD3  DE0  DE1  DE2  S
31  E13  DC2  DF0  DF1  S
32  C03  C43  C83  CC3  S
33  D03  D43  D83  DC3  S
34  C33  C73  CB3  CF3  S
35  D33  D73  DB3  DF3  S
36  EE0  ---  ---  ---  C
37  EF0  ---  ---  ---  C

```

■ Example of segment option source file

```

* E0C6S32 SEGMENT OPTION DOCUMENT V 3.20
*
* FILE NAME      CS320A2S.DOC
* USER'S NAME    SEIKO EPSON CORP.
* INPUT DATE     95/04/17
*
*
* OPTION NO.16
*
* < LCD SEGMENT DECODE TABLE >
*
* SEG COM0 COM1 COM2 COM3 SPEC
*
  0 C01 C00 C32 E20 S
  1 C12 C11 C10 C23 S
  2 C13 C20 C21 C22 S
  3 E00 C02 C30 C31 S
  4 C41 C40 C72 E21 S
  5 C52 C51 C50 C63 S
  6 C53 C60 C61 C62 S
  7 E01 C42 C70 C71 S
  8 C81 C80 CB2 E22 S
  9 C92 C91 C90 CA3 S
 10 C93 CA0 CA1 CA2 S
 11 E02 C82 CB0 CB1 S
 12 CC1 CC0 CF2 E23 S
 13 CD2 CD1 CD0 CE3 S
 14 CD3 CE0 CE1 CE2 S
 15 E03 CC2 CF0 CF1 S
 16 D01 D00 D32 E30 S
 17 D12 D11 D10 D23 S
 18 D13 D20 D21 D22 S
 19 E10 D02 D30 D31 S
 20 D41 D40 D72 E31 S
 21 D52 D51 D50 D63 S
 22 D53 D60 D61 D62 S
 23 E11 D42 D70 D71 S
 24 D81 D80 DB2 E32 S
 25 D92 D91 D90 DA3 S
 26 D93 DA0 DA1 DA2 S
 27 E12 D82 DB0 DB1 S
 28 DC1 DC0 DF2 E33 S
 29 DD2 DD1 DD0 DE3 S
 30 DD3 DE0 DE1 DE2 S
 31 E13 DC2 DF0 DF1 S
 32 C03 C43 C83 CC3 S
 33 D03 D43 D83 DC3 S
 34 C33 C73 CB3 CF3 S
 35 D33 D73 DB3 DF3 S
 36 EE0 E40 E41 E42 C
 37 EF0 E43 E50 E51 C
\\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 ICS6S32

6.1 ICS6S32 Outline

The In-circuit Emulator ICE6200 connects the target board produced by the user via the EVA6S32 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 ICS6S32.

The ICS6S32 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.

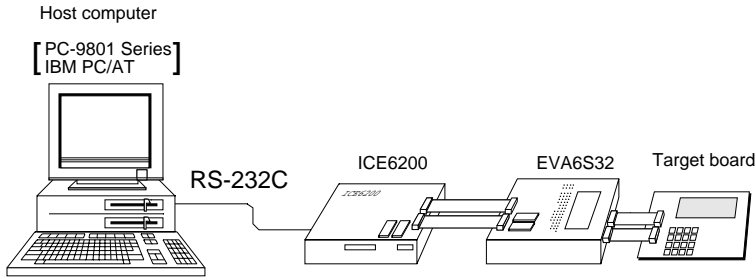


Fig. 6.1.1
Debugging system
using ICE6200

☞ The 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 ICS6S32 Restrictions

Take the following precautions when using the ICS6S32.

■ ROM Area

The ROM area is limited to a maximum address of 7FFH. Assigning data above the 7FFH address causes an error.

■ RAM Area

The RAM area is limited to a maximum address of 0FFH. However, as the following addresses are in the unused area, designation of this area with the ICE commands produces an error.

Unused area: 0A0H to 0EFH (When segment memory is assigned from 040H to 06FH)
0A0H to 0BFH (When segment memory is assigned from 0C0H to 0EFH)
0F0H to 0F5H, 0FFH (Area from 0F6H to 0FEH is I/O memory)

Refer to the "E0C6S32 Technical Manual" for details.

■ Undefined Code

The instructions below are not specified for the E0C6S32 and so cannot be used.

SLP

PUSH	XP	POP	XP	LD	XP,r	LD	r,XP
PUSH	YP	POP	YP	LD	YP,r	LD	r,YP

■ OPTLD Command

In the ICS6S32, OPTLD command can be used.

This command is used to load HEX files (function option data and segment option data for LCD) in the EVA6S32 memory with the ICE6200.

Load of function option data: #OPTLD, 1, CS32XXX□

Load of segment option data: #OPTLD, 2, CS32XXX□

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 EVA6S32 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 EVA6S32 segment option data memory.
 It is HEX file output by the segment option generator and has intel HEX format.

Examples #OPTLD, 1, CS32XXX CS32XXXF.HEX file is loaded in the function option data memory.
 #OPTLD, 2, CS32XXX CS32XXXS.HEX file is loaded in the segment option data memory.
 #SD, DF
 DF | 0: 1 The OSC3 oscillation is turned ON.
 E0 | 0: /
 #SD, DF
 DF | 1: 3 Switching from OSC1 to OSC3.
 E0 | 0: /
 #I The CPU is reset.
 (Switches CPU clock to OSC1 when OSC3 oscillation is set.)

6.3 ICS6S32 Quick Reference

Starting command and input/output files

␣ indicates the Return key.

Execution file: ICS6S32.BAT (ICS6S32J.EXE) ... for MS-DOS
ICS6S32B.BAT (ICS6S32W.EXE) ... for PC-DOS

Starting command: ICS6S32 (ICS6S32J)␣ ... for MS-DOS
ICS6S32B (ICS6S32W)␣ ... for PC-DOS

Input file: CS32XXXL.HEX (Object file, low-order)
CS32XXXH.HEX (Object file, high-order)
CS32XXXD.HEX (Data RAM file)
CS32XXXC.HEX (Control file)

Output file: CS32XXXL.HEX (Object file, low-order)
CS32XXXH.HEX (Object file, high-order)
CS32XXXD.HEX (Data RAM file)
CS32XXXC.HEX (Control file)

Display example

```

*** E0C6S32 ICE CONTROL SOFTWARE. --- Ver 3.01 ***
EEEEEEEEEE PPPPPPPP SSSSSSS 00000000 NNN NNN
EEEEEEEEEE PPPPPPPP SSS SSSS 000 000 NNNN NNN
EEE PPP PPP SSS SSS 000 000 NNNNN NNN
EEE PPP PPP SSS 000 000 NNNNNN NNN
EEEEEEEEEE PPPPPPPP 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 NNNNN
EEEEEEEEEE PPP SSS SSS 000 000 NNN NNN
EEEEEEEEEE PPP SSSSSS 00000000 NNN NN

(C) COPYRIGHT 1991 SEIKO EPSON CORP.

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

```

Start-up message

When ICS6S32 is started, the start-up message is displayed, and a self-test is automatically performed. ICS6S32 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 ICS6S32 program is terminated by entering the Q (Quit) command.

Note Confirm that the cables connected properly, then operate the ICS6S32.

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 EVA6S32 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
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 EVA6S32 CPU internal registers
		#SR [↵]	Set EVA6S32 CPU internal registers
		#I [↵]	Reset EVA6S32 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 <input type="checkbox"/>	Display history data for pointer 1 and pointer 2
		#HB <input type="checkbox"/>	Display upstream history data
		#HG <input type="checkbox"/>	Display 21 line history data
		#HP <input type="checkbox"/>	Display history pointer
		#HPS,a <input type="checkbox"/>	Set history pointer
		#HC,S/C/E <input type="checkbox"/>	Sets up the history information acquisition before (S), before/after (C) and after (E)
		#HA,a1,a2 <input type="checkbox"/>	Sets up the history information acquisition from program area a1 to a2
		#HAR,a1,a2 <input type="checkbox"/>	Sets up the prohibition of the history information acquisition from program area a1 to a2
		#HAD <input type="checkbox"/>	Indicates history acquisition program area
		#HS,a <input type="checkbox"/>	Retrieves and indicates the history information which executed a program address "a"
		#HSW,a <input type="checkbox"/> #HSR,a <input type="checkbox"/>	Retrieves and indicates the history information which wrote or read the data area address "a"
12	File	#RF,file <input type="checkbox"/>	Move program file to memory
		#RFD,file <input type="checkbox"/>	Move data file to memory
		#VF,file <input type="checkbox"/>	Compare program file and contents of memory
		#VFD,file <input type="checkbox"/>	Compare data file and contents of memory
		#WF,file <input type="checkbox"/>	Save contents of memory to program file
		#WFD,file <input type="checkbox"/>	Save contents of memory to data file
		#CL,file <input type="checkbox"/>	Load ICE6200 set condition from file
		#CS,file <input type="checkbox"/>	Save ICE6200 set condition to file
		#OPTLD,1,file <input type="checkbox"/> #OPTLD,2,file <input type="checkbox"/>	Load function option data from file Load segment option data from file
13	Coverage	#CVD <input type="checkbox"/>	Indicates coverage information
		#CVR <input type="checkbox"/>	Clears coverage information
14	ROM Access	#RP <input type="checkbox"/>	Move contents of ROM to program memory
		#VP <input type="checkbox"/>	Compare contents of ROM with contents of program memory
		#ROM <input type="checkbox"/>	Set ROM type
15	Terminate ICE	#Q <input type="checkbox"/>	Terminate ICE and return to operating system control
16	Command Display	#HELP <input type="checkbox"/>	Display ICE6200 instruction
17	Self Diagnosis	#CHK <input type="checkbox"/>	Report results of ICE6200 self diagnostic test

means press the RETURN key.

7 MASK DATA CHECKER MDC6S32

7.1 MDC6S32 Outline

The Mask Data Checker MDC6S32 is a software tool which checks the program data (CS32XXXH.HEX and CS32XXXL.HEX) and option data (CS32XXXF.DOC and CS32XXXS.DOC) created by the user and creates the data file (C6S32XXX.PAn) for generating mask patterns. The user must send the file generated through this software tool to Seiko Epson.

Moreover, MDC6S32 has the capability to restore the generated data file (C6S32XXX.PA0) to the original file format.

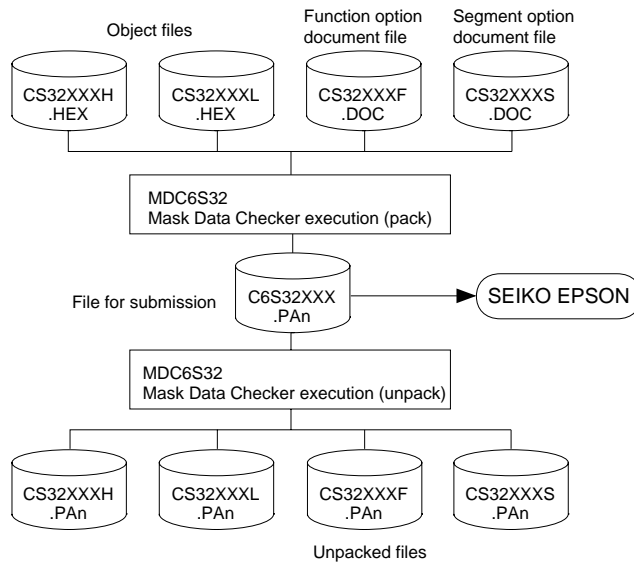


Fig. 7.1.1
MDC6S32 execution flow

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

7.2 MDC6S32 Quick Reference

■ Starting command and input/output files

Execution file: MDC6S32.EXE

Starting command: **MDC6S32** indicates the Return key.

Input file:	CS32XXXL.HEX (Object file, low-order)] When packing
	CS32XXXH.HEX (Object file, high-order)	
	CS32XXXF.DOC (Function option document file)	
	CS32XXXS.DOC (Segment option document file)	
	C6S32XXX.PAn (Packed file)	

Output file:	C6S32XXX.PAn (Packed file)] When packing
	CS32XXXL.PAn (Object file, low-order)] When unpacking
	CS32XXXH.PAn (Object file, high-order)	
	CS32XXXF.PAn (Function option document file)	
	CS32XXXS.PAn (Segment option document file)	

■ Display examples

```

*** E0C6S32 PACK / UNPACK PROGRAM Ver 1.00 ***

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

(C) COPYRIGHT 1990 SEIKO EPSON CORP.

--- OPERATION MENU ---

      1. PACK
      2. UNPACK

PLEASE SELECT NO. ?

```

```

--- OPERATION MENU ---

      1. PACK
      2. UNPACK

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

CS32XXXH.HEX -----+
CS32XXXL.HEX -----+
CS32XXXF.DOC -----+----- C6S32XXX.PAn (PACK FILE)
CS32XXXS.DOC -----+

PLEASE INPUT PACK FILE NAME (C6S32XXX.PAn) ? C6S320A0.PA0  ... (2)

CS320A0H.HEX -----+
CS320A0L.HEX -----+
CS320A0F.DOC -----+----- C6S320A0.PA0
CS320A0S.DOC -----+

```

Start-up message

When MDC6S32 is started, the start-up message and operation menu are displayed. Here, the user is prompted to select operation options.

Packing of data

- (1) Select "1" 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 "C6S32XXX.PA0" has been submitted, the pack file name should be entered as "C6S32XXX.PA1".)

With this, the mask file (C6S32XXX.PAn) is generated, and the MDC6S32 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 (ASM6S32) as program data. If the program data generated with the -N option of the Cross Assembler is packed, undefined program area is filled with FFH code. In this case, following message is displayed.

```
WARNING: FILLED <file_name> FILE WITH FFH.
```

```

--- OPERATION MENU ---

      1. PACK
      2. UNPACK

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

PLEASE INPUT PACKED FILE NAME (C6S32XXX.PAn) ? C6S320A0.PA0  ... (2)

      +----- CS320A0H.PA0
      |----- CS320A0L.PA0
C6S320A0.PA0 -----+----- CS320A0F.PA0
      |----- CS320A0S.PA0
      +-----

```

Unpacking of data

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

With this, the mask data file (C6S32XXX.PAn) is restored to the original file format, and the MDC6S32 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.

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.

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. E0C6S32 INSTRUCTION SET

Classification	Mnemonic	Operand	Operation Code						Flag			Clock	Operation							
			B	A	9	8	7	6	5	4	3			2	1	0	I	D	Z	C
Branch instructions	PSET	p	1	1	1	0	0	1	0	p4	p3	p2	p1	p0					5	NBP ← p4, NPP ← p3~p0
	JP	s	0	0	0	0	s7	s6	s5	s4	s3	s2	s1	s0					5	PCB ← NBP, PCP ← NPP, PCS ← s7~s0
		C, s	0	0	1	0	s7	s6	s5	s4	s3	s2	s1	s0					5	PCB ← NBP, PCP ← NPP, PCS ← s7~s0 if C=1
		NC, s	0	0	1	1	s7	s6	s5	s4	s3	s2	s1	s0					5	PCB ← NBP, PCP ← NPP, PCS ← s7~s0 if C=0
		Z, s	0	1	1	0	s7	s6	s5	s4	s3	s2	s1	s0					5	PCB ← NBP, PCP ← NPP, PCS ← s7~s0 if Z=1
		NZ, s	0	1	1	1	s7	s6	s5	s4	s3	s2	s1	s0					5	PCB ← NBP, PCP ← NPP, PCS ← s7~s0 if Z=0
	JPBA		1	1	1	1	1	1	1	0	1	0	0	0					5	PCB ← NBP, PCP ← NPP, PCSH ← B, PCSL ← A
	CALL	s	0	1	0	0	s7	s6	s5	s4	s3	s2	s1	s0					7	M(SP-1) ← PCP, M(SP-2) ← PCSH, M(SP-3) ← PCSL+1 SP ← SP-3, PCP ← NPP, PCS ← s7~s0
	CALZ	s	0	1	0	1	s7	s6	s5	s4	s3	s2	s1	s0					7	M(SP-1) ← PCP, M(SP-2) ← PCSH, M(SP-3) ← PCSL+1 SP ← SP-3, PCP ← 0, PCS ← s7~s0
	RET		1	1	1	1	1	1	0	1	1	1	1	1					7	PCSL ← M(SP), PCSH ← M(SP+1), PCP ← M(SP+2) SP ← SP+3
RETS		1	1	1	1	1	1	0	1	1	1	1	0					12	PCSL ← M(SP), PCSH ← M(SP+1), PCP ← M(SP+2) SP ← SP+3, PC ← PC+1	
RETD	l	0	0	0	1	l7	l6	l5	l4	l3	l2	l1	l0					12	PCSL ← M(SP), PCSH ← M(SP+1), PCP ← M(SP+2) SP ← SP+3, M(X) ← l3~l0, M(X+1) ← l7~l4, 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
		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
		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, 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, YH	1	1	1	0	1	0	1	1	0	1	r1	r0					5	r ← YH
		r, YL	1	1	1	0	1	0	1	1	1	0	r1	r0					5	r ← YL
	ADC	XH, i	1	0	1	0	0	0	0	0	i3	i2	i1	i0				↑ ↓	7	XH ← XH+i3~i0+C
		XL, i	1	0	1	0	0	0	0	1	i3	i2	i1	i0				↓ ↓	7	XL ← XL+i3~i0+C
YH, i		1	0	1	0	0	0	1	0	i3	i2	i1	i0				↑ ↓	7	YH ← YH+i3~i0+C	
YL, i		1	0	1	0	0	0	1	1	i3	i2	i1	i0				↓ ↓	7	YL ← YL+i3~i0+C	

Classification	Mnemonic	Operand	Operation Code						Flag			Clock	Operation							
			B	A	9	8	7	6	5	4	3			2	1	0	I	D	Z	C
Index operation instructions	CP	XH, i	1	0	1	0	0	1	0	0	i3	i2	i1	i0	↑	↓	↑	↓	7	XH-i3~i0
		XL, i	1	0	1	0	0	1	0	1	i3	i2	i1	i0	↑	↓	↑	↓	7	XL-i3~i0
		YH, i	1	0	1	0	0	1	1	0	i3	i2	i1	i0	↑	↓	↑	↓	7	YH-i3~i0
		YL, i	1	0	1	0	0	1	1	1	i3	i2	i1	i0	↑	↓	↑	↓	7	YL-i3~i0
Data transfer instructions	LD	r, i	1	1	1	0	0	0	r1	r0	i3	i2	i1	i0					5	r ← i3~i0
		r, q	1	1	1	0	1	1	0	0	r1	r0	q1	q0					5	r ← q
		A, Mn	1	1	1	1	1	0	1	0	n3	n2	n1	n0					5	A ← M(n3~n0)
		B, Mn	1	1	1	1	1	0	1	1	n3	n2	n1	n0					5	B ← M(n3~n0)
		Mn, A	1	1	1	1	1	0	0	0	n3	n2	n1	n0					5	M(n3~n0) ← A
		Mn, B	1	1	1	1	1	0	0	1	n3	n2	n1	n0					5	M(n3~n0) ← B
	LDPX	MX, i	1	1	1	0	0	1	1	0	i3	i2	i1	i0					5	M(X) ← i3~i0, X ← X+1
		r, q	1	1	1	0	1	1	1	0	r1	r0	q1	q0					5	r ← q, X ← X+1
	LDPY	MY, i	1	1	1	0	0	1	1	1	i3	i2	i1	i0					5	M(Y) ← i3~i0, Y ← Y+1
		r, q	1	1	1	0	1	1	1	1	r1	r0	q1	q0					5	r ← q, Y ← Y+1
LBPX	MX, l	1	0	0	1	l7	l6	l5	l4	l3	l2	l1	l0					5	M(X) ← l3~l0, M(X+1) ← l7~l4, 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
		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
		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
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	

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	$YH \leftarrow M(SP), SP \leftarrow SP+1$
		YL	1	1	1	1	1	1	0	1	1	0	0	1					5	$YL \leftarrow M(SP), SP \leftarrow SP+1$
		F	1	1	1	1	1	1	0	1	1	0	1	0	$\uparrow \downarrow \uparrow \downarrow \uparrow \downarrow$				5	$F \leftarrow M(SP), SP \leftarrow SP+1$
	LD	SPH, r	1	1	1	1	1	1	1	0	0	0	r1	r0					5	$SPH \leftarrow r$
		SPL, r	1	1	1	1	1	1	1	1	0	0	r1	r0					5	$SPL \leftarrow r$
		r, SPH	1	1	1	1	1	1	1	0	0	1	r1	r0					5	$r \leftarrow SPH$
		r, SPL	1	1	1	1	1	1	1	1	0	1	r1	r0					5	$r \leftarrow SPL$
Arithmetic instructions	ADD	r, i	1	1	0	0	0	0	r1	r0	i3	i2	i1	i0	$\star \uparrow \downarrow$				7	$r \leftarrow r+i3-i0$
		r, q	1	0	1	0	1	0	0	0	r1	r0	q1	q0	$\star \uparrow \downarrow$				7	$r \leftarrow r+q$
	ADC	r, i	1	1	0	0	0	1	r1	r0	i3	i2	i1	i0	$\star \uparrow \downarrow$				7	$r \leftarrow r+i3-i0+C$
		r, q	1	0	1	0	1	0	0	1	r1	r0	q1	q0	$\star \uparrow \downarrow$				7	$r \leftarrow r+q+C$
	SUB	r, q	1	0	1	0	1	0	1	0	r1	r0	q1	q0	$\star \uparrow \downarrow$				7	$r \leftarrow r-q$
		SBC	r, i	1	1	0	1	0	1	r1	r0	i3	i2	i1	i0	$\star \uparrow \downarrow$				7
	r, q		1	0	1	0	1	0	1	1	r1	r0	q1	q0	$\star \uparrow \downarrow$				7	$r \leftarrow r-q-C$
	AND	r, i	1	1	0	0	1	0	r1	r0	i3	i2	i1	i0	\uparrow				7	$r \leftarrow r \wedge i3-i0$
		r, q	1	0	1	0	1	1	0	0	r1	r0	q1	q0	\uparrow				7	$r \leftarrow r \wedge q$
	OR	r, i	1	1	0	0	1	1	r1	r0	i3	i2	i1	i0	\uparrow				7	$r \leftarrow r \vee i3-i0$
		r, q	1	0	1	0	1	1	0	1	r1	r0	q1	q0	\uparrow				7	$r \leftarrow r \vee q$
	XOR	r, i	1	1	0	1	0	0	r1	r0	i3	i2	i1	i0	\uparrow				7	$r \leftarrow r \vee i3-i0$
		r, q	1	0	1	0	1	1	1	0	r1	r0	q1	q0	\uparrow				7	$r \leftarrow r \vee q$
	CP	r, i	1	1	0	1	1	1	r1	r0	i3	i2	i1	i0	$\uparrow \downarrow$				7	$r-i3-i0$
		r, q	1	1	1	1	0	0	0	0	r1	r0	q1	q0	$\uparrow \downarrow$				7	$r-q$
	FAN	r, i	1	1	0	1	1	0	r1	r0	i3	i2	i1	i0	\uparrow				7	$r \wedge i3-i0$
		r, q	1	1	1	1	0	0	0	1	r1	r0	q1	q0	\uparrow				7	$r \wedge q$
	RLC	r	1	0	1	0	1	1	1	1	r1	r0	r1	r0	$\uparrow \downarrow$				7	$d3 \leftarrow d2, d2 \leftarrow d1, d1 \leftarrow d0, d0 \leftarrow C, C \leftarrow d3$
	RRC	r	1	1	1	0	1	0	0	0	1	1	r1	r0	$\uparrow \downarrow$				5	$d3 \leftarrow C, d2 \leftarrow d3, d1 \leftarrow d2, d0 \leftarrow d1, C \leftarrow d0$
	INC	Mn	1	1	1	1	0	1	1	0	n3	n2	n1	n0	$\uparrow \downarrow$				7	$M(n3-n0) \leftarrow M(n3-n0)+1$
	DEC	Mn	1	1	1	1	0	1	1	1	n3	n2	n1	n0	$\uparrow \downarrow$				7	$M(n3-n0) \leftarrow M(n3-n0)-1$
	ACPX	MX, r	1	1	1	1	0	0	1	0	1	0	r1	r0	$\star \uparrow \downarrow$				7	$M(X) \leftarrow M(X)+r+C, X \leftarrow X+1$
	ACPY	MY, r	1	1	1	1	0	0	1	0	1	1	r1	r0	$\star \uparrow \downarrow$				7	$M(Y) \leftarrow M(Y)+r+C, Y \leftarrow Y+1$
	SCPX	MX, r	1	1	1	1	0	0	1	1	1	0	r1	r0	$\star \uparrow \downarrow$				7	$M(X) \leftarrow M(X)-r-C, X \leftarrow X+1$
	SCPY	MY, r	1	1	1	1	0	0	1	1	1	1	r1	r0	$\star \uparrow \downarrow$				7	$M(Y) \leftarrow M(Y)-r-C, Y \leftarrow Y+1$
	NOT	r	1	1	0	1	0	0	r1	r0	1	1	1	1	\uparrow				7	$r \leftarrow \bar{r}$

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)
- 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. TROUBLESHOOTING

Tool	Problem	Remedy measures
ICE6200	Nothing appears on the screen, or nothing works, after activation.	<p>Check the following and remedy if necessary:</p> <ul style="list-style-type: none"> • Is the RS-232C cable connected correctly? • Is the RS-232C driver installed? • Is SPEED.COM or MODE.COM on the disk? • Is the execution file correct? <ul style="list-style-type: none"> MS-DOS ICS6S32J.EXE PC-DOS ICS6S32W.EXE • Is the DOS version correct? <ul style="list-style-type: none"> MS-DOS Ver. 3.1 or later PC-DOS Ver. 2.1 or later • Is the DIP switches that set the baud rate of the main ICE6200 unit set correctly? • Is the breaker of the ICE6200 set to ON?
	The ICE6200 breaker tripped immediately after activation.	<p>Check the following and remedy if necessary:</p> <ul style="list-style-type: none"> • Are connectors F1 and F5 connected to the EVA6S32 correctly? • Is the target board power short-circuiting?
	<ILLEGAL VERSION ICE6200> appears on the screen immediately after activation.	The wrong version of ICE6200 is being used. Use the latest version.
	<ILLEGAL VERSION PARAMETER FILE> appears on the screen immediately after activation.	The wrong version of ICS6S32P.PAR is being used. Use the latest version.
	Immediate values A (10) and B (11) cannot be entered correctly with the A command.	<p>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).</p> <p><i>Example:</i></p> <pre>LD A, B Data in the B register is loaded into the A register. LD B, 0A Immediate value A is loaded into the B register.</pre>
	<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 indicate a 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.
SOG6S32	An R error occurs although the address is correctly set in the segment source file.	<p>Check the following and remedy if necessary:</p> <ul style="list-style-type: none"> • Does the address symbol use capital letters? • Are the output ports set for every two terminals?

Tool	Problem	Remedy measures
ASM6S32	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.
MDC6S32	Activation is impossible.	Check the following and remedy if necessary: <ul style="list-style-type: none"> • Is the number of files set at ten or more in OS environment file CONFIG.SYS?
EVA6S32	The EVA6S32 does not work when it is used independently.	Check the following and remedy if necessary: <ul style="list-style-type: none"> • Has the EPROM for F.HEX and S.HEX been replaced by the EPROM for the target? • Is the EPROM for F.HEX and S.HEX installed correctly? • Is the appropriate voltage being supplied? (5V DC, 3A, or more) • Are the program ROMs (H and L) installed correctly? • Is data written from address 4000H? (When the 27C256 is used as the program ROM) • Is the EN/DIS switch on the EVA6S32 set to EN?
	Target segment does not light.	Check the following and remedy if necessary: <ul style="list-style-type: none"> • Is an EPROM with an access time of 250 ns or less being used for S.HEX. • Has the VADJ VR inside the EVA6S32 top cover been turned to a lower setting?

APPENDIX C. DEBUGGING FOR 1/2 DUTY DRIVE ON EVA6S32

The EVA6S32 is used to debug E0C6S32 systems. However, the EVA6S32 cannot output 1/2 duty signal to drive LCD. When debugging a E0C6S32 system that uses a 1/2 duty drive, it should be done with the following procedure.

1. Set 1/3 duty function option using the FOG6S32.
2. Assign the segments using the SOG6S32.
Since only the COM0 and COM1 signals are used for 1/2 duty drive, assign the RAM addresses to be used to COM0 and COM1.

```
;< LCD SEGMENT DECODE TABLE >
;SEG  COM0  COM1  COM2  COM3  SPEC
  0    C00   C01   ---   ---   S
  1    C02   C03   ---   ---   S
  2    C10   C11   ---   ---   S
  3    C12   C13   ---   ---   S
  4    C20   C21   ---   ---   S
  :
  :
  :
36    DA0   DB0   ---   ---   S
37    DC0   DC1   ---   ---   S
```

↑
RAM addresses
that are used for 1/2 duty drive

3. Debug with the EVA6S32 and check whether the segment assignment is correct or not.
Be aware that the LCD drive waveform is different and the LCD corresponding to COM2 goes on.

Creating Mask Data

1. Edit the function option data (option No. 14) using the FOG6S32 to change the 1/3 duty setting to the 1/2 duty setting.
2. It is unnecessary to change the segment assignment by the SOG6S32.

Note: The EVA6S32 does not support the 1/2 bias LCD drive method. The LCD drive waveform is a 1/3 bias waveform.

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