

CMOS 4-BIT SINGLE CHIP MICROCOMPUTER
E0C6282 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".

E0C6282 Development Tool Manual

PREFACE

This manual mainly explains the outline of the development support tool for the 4-bit Single Chip Micro-computer E0C6282.

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 EVA6282 Manual ICE6200 Hardware Manual |
| <i>Development procedure</i> | ☞ E0C62 Family Technical Guide |
| <i>Device (E0C6282)</i> | ☞ E0C6282 Technical Manual |
| <i>Instructions</i> | ☞ E0C6200/6200A Core CPU Manual |

CONTENTS

| | |
|---|-----------|
| 1 COMPOSITION OF DEVELOPMENT SUPPORT TOOL | 1 |
| 1.1 Configuration of DEV6282 | 1 |
| 1.2 Developmental Environment | 1 |
| 1.3 Development Flow | 2 |
| 1.4 Production of Execution Disk | 3 |
| 2 DEVELOPMENT TOOL MANAGEMENT SYSTEM DMS6200 | 4 |
| 2.1 DMS6200 Outline | 4 |
| 2.2 DMS6200 Quick Reference | 4 |
| 3 CROSS ASSEMBLER ASM6282 | 5 |
| 3.1 ASM6282 Outline | 5 |
| 3.2 E0C6282 Restrictions | 5 |
| 3.3 ASM6282 Quick Reference | 6 |
| 4 MELODY ASSEMBLER MLA6282 | 8 |
| 4.1 MLA6282 Outline | 8 |
| 4.2 E0C6282 Restrictions | 8 |
| 4.3 MLA6282 Quick Reference | 9 |
| 5 FUNCTION OPTION GENERATOR FOG6282 | 11 |
| 5.1 FOG6282 Outline | 11 |
| 5.2 E0C6282 Option List | 11 |
| 5.3 Option Specifications and Selection Message | 13 |
| 5.4 FOG6282 Quick Reference | 20 |
| 5.5 Sample File | 22 |

| | | |
|-----------------|---|-----------|
| 6 | SEGMENT OPTION GENERATOR SOG6282 | 24 |
| 6.1 | SOG6282 Outline | 24 |
| 6.2 | Option List | 24 |
| 6.3 | Segment Ports Output Specifications | 25 |
| 6.4 | SOG6282 Quick Reference | 26 |
| 6.5 | Sample Files | 28 |
| 7 | ICE CONTROL SOFTWARE ICS6282 | 32 |
| 7.1 | ICS6282 Outline | 32 |
| 7.2 | ICS6282 Restrictions | 32 |
| 7.3 | ICS6282 Quick Reference | 34 |
| 8 | MASK DATA CHECKER MDC6282 | 37 |
| 8.1 | MDC6282 Outline | 37 |
| 8.2 | MDC6282 Quick Reference | 37 |
| APPENDIX | A. E0C6282 INSTRUCTION SET | 40 |
| | B. E0C6282 RAM MAP | 44 |
| | C. E0C6282 I/O MEMORY MAP | 46 |
| | D. TROUBLESHOOTING | 48 |
| | E. CAUTION | 50 |

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 DEV6282

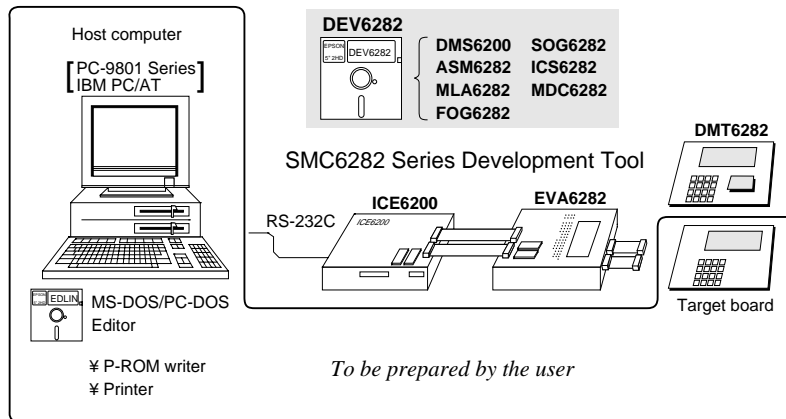
The below software are included in the product of the E0C6282 development support tool DEV6282.

1. Development Tool Management System DMS6200 Menu selection for each software / start-up software
2. Cross Assembler ASM6282 Cross assembler for program preparation
3. Melody Assembler MLA6282 Melody data preparation program
4. Function Option Generator FOG6282 Function option data preparation program
5. Segment Option Generator SOG6282 Segment option data preparation program
6. ICE Control Software ICS6282 ICE control program
7. Mask Data Checker MDC6282 Mask data preparation program

1.2 Developmental Environment

The software product of the development support tool DEV6282 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 E0C6282, 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 DEV6282 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.

☞ Demonstration Tool DMT6282 (under development)

This demonstration tool is intended for users who are currently planning applications using the E0C6282 Series to better understand the E0C6282 as well as to evaluate its functions. Target board may be created by using the DMT6282.

1.3 Development Flow

Figure 1.3.1 shows the development flow through the DEV6282.

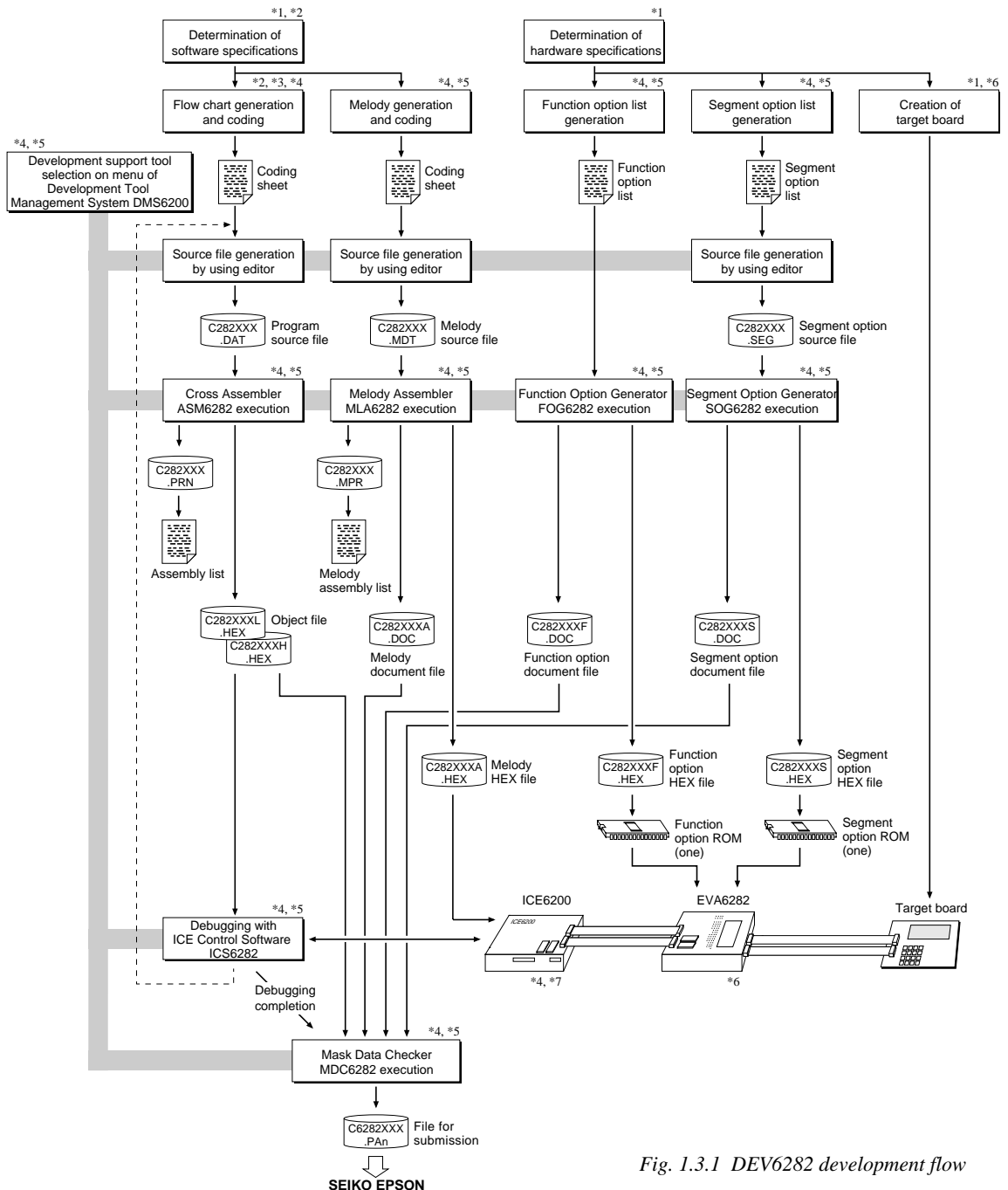


Fig. 1.3.1 DEV6282 development flow

Concerning file names

All the input-output file name for the each development support tool commonly use "C282XXX". In principle each file should be produced in this manner. Seiko Epson will designate the "XXX" for each customer.

Reference Manual

- *1 E0C6282 Technical Hardware Manual
- *2 E0C6282 Technical Software Manual
- *3 E0C6200/6200A Core CPU Manual
- *4 E0C62 Family Development Tool Reference Manual
- *5 E0C6282 Development Tool Manual (this manual)
- *6 EVA6282 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 DEV6282 floppy disk.

The content of the files contained in the DEV6282 floppy disk are shown below.

| PC-DOS version | MS-DOS version | Contents |
|----------------|----------------|---|
| ASM6282.EXE | ASM6282.EXE | Cross Assembler execution file |
| DMS6200.EXE | DMS6200.EXE | Development Tool Management System execution file |
| FOG6282.EXE | FOG6282.EXE | Function Option Generator execution file |
| ICS6282B.BAT | ICS6282.BAT | ICE Control Software batch file |
| ICS6282W.EXE | ICS6282J.EXE | ICE Control Software execution file |
| ICS6282P.PAR | ICS6282P.PAR | ICE Control Software parameter file |
| MDC6282.EXE | MDC6282.EXE | Mask Data Checker execution file |
| MLA6282.EXE | MLA6282.EXE | Melody Assembler execution file |
| SOG6282.EXE | SOG6282.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 (DEV6282, 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 MDC6282 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 "ICS6282(B).BAT" the batch process is indicated such that the ICS6282J(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 (DEV6282), then insert the original disk into the A drive and execute the COPY command.

```
C>MD DEV6282 [ ]
```

```
C>CD DEV6282 [ ]
```

```
C\DEV6282>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)

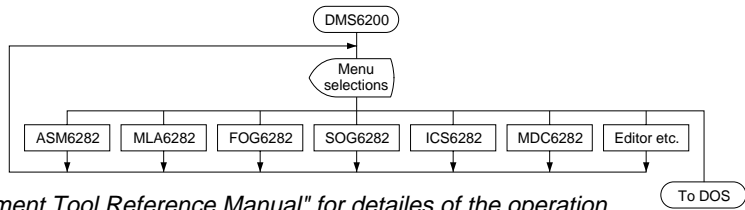
```
SPEED R0 9600 B8 PN S1
```

2 DEVELOPMENT TOOL MANAGEMENT SYSTEM DMS6200

2.1 DMS6200 Outline

The DMS6200 (Development Tool Management System) is a software which selects the DEV6282 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 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 1991 SEIKO EPSON CORP.

STRIKE ANY KEY.
  
```

Start message

When DMS6200 is started, the following message is displayed. For "STRIKE ANY KEY.", press any key to advance the program execution. To suspend execution, press the "CTRL" and "C" keys together: the sequence returns to the DOS command level.

```

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

1) ASM6282 .EXE
2) FOG6282 .EXE
3) ICS6282B.BAT
4) ICS6282W.EXE
5) MDC6282 .EXE
6) MLA6282 .EXE
7) SOG6282 .EXE

Input Number ? [ 1 ]
  
```

Menu screen (PC-DOS Version)

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

```

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

1) C282XXX .DAT
2) C282XXX .MDT
3) C282XXX .MPR
:
14) C6282XXX.PA0

Input Number ? [ 1 ]

Edit > [ASM6282 C282XXX ]
  
```

Source file selection screen

To starting ASM6282 or MLA6282, 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 ASM6282

3.1 ASM6282 Outline

The ASM6282 cross assembler is an assembler program for generating the machine code used by the E0C6282 4-bit, single-chip microcomputers. The Cross Assembler ASM6282 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.*

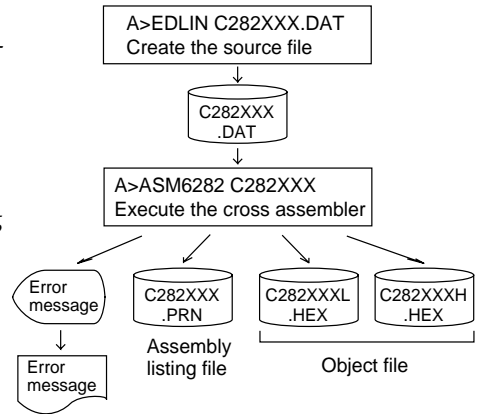


Fig. 3.1.1 ASM6282 execution flow

3.2 E0C6282 Restrictions

Note the following when generating a program by the E0C6282:

■ ROM area

The capacity of the E0C6282 ROM is 2K steps (0000H to 07FFH).

Therefore, the specification range of the memory setting pseudo-instructions and PSET instruction is restricted.

Memory configuration:

Bank: Only bank 0, Page: 8 pages (0 to 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 |

■ RAM area

The capacity of the E0C6282 RAM is 250 words (000H to 0F4H, 0F6H, and 0F9H to 0FCH, 4 bits/word). Memory access is invalid when the unused area of the index register is specified.

Example:

| | | |
|----|---------|---|
| LD | X, 0F5H | F5H 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, 0FDH | FDH is loaded into the IY register, but an unused area has been specified so that the memory accessible with the IY register (MY) is invalid. |

■ Undefined codes

The following instructions have not been defined in the E0C6282 instruction sets.

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

3.3 ASM6282 Quick Reference

■ Starting command and input/output files

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

Execution file: ASM6282.EXE

Starting command: **ASM6282_ [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: C282XXX.DAT (Source file)

Output file: C282XXXL.HEX (Object file, low-order)
 C282XXXH.HEX (Object file, high-order)
 C282XXX.PRN (Assembly listing file)

■ Display example

```

*** E0C6282 CROSS ASSEMBLER. --- Ver 2.00 ***

EEEEEEEEEE PFFFFFFFFP SSSSSSS 00000000 NNN NNN
EEEEEEEEEE PFFFFFFFFP 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 PFFFFFFFFP SSSSSS 000 000 NNN NNNNN
EEEEEEEEEE PFFFFFFFFP SSSS 000 000 NNN NNNNN
EEE PPP SSS SSS 000 000 NNN NNNNN
EEE PPP SSS SSS 000 000 NNN NNNNN
EEEEEEEEEE PPP SSSS SSS 000 000 NNN NNN
EEEEEEEEEE PPP SSSSSS 00000000 NNN NN

(C) COPYRIGHT 1991 SEIKO EPSON CORP.

SOURCE FILE NAME IS " C282XXX.DAT "

THIS SOFTWARE MAKES NEXT FILES.

C282XXXH.HEX ... HIGH BYTE OBJECT FILE.
C282XXXL.HEX ... LOW BYTE OBJECT FILE.
C282XXX.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 ASM6282 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, ASM6282 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 | 7 |
| 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"> • A statement exceeded a page boundary although its location was not specified. • The location counter value exceeded the upper limit of the program memory, or a location exceeding the upper limit was specified. • A value greater than that which the number of significant digits of the operand will accommodate was specified. |
| ! (Warning) | Memory areas overlapped because of a "PAGE" or "ORG" pseudo-instruction or both. |
| FILE NAME ERROR | The source file name was longer than or equal to 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 MELODY ASSEMBLER MLA6282

4.1 MLA6282 Outline

The Melody Assembler MLA6282 is an assembler that outputs melody ROM data of the 4-bit single-chip microcomputers E0C6282 Series. The Melody Assembler MLA6282 assembles the source file which has been input by the user's editor and generates the object file in Intel-Hex format, assembly list file and document file.

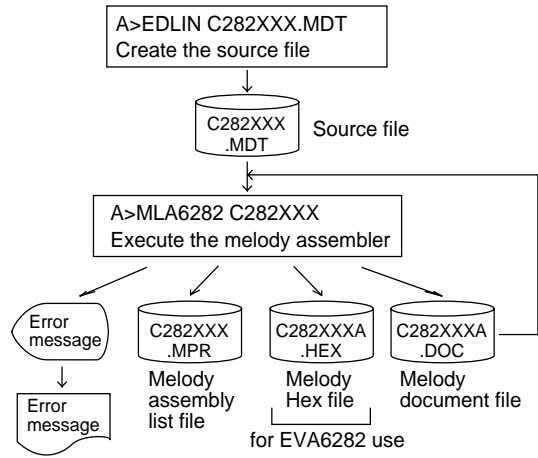


Fig. 4.1.1 MLA6282 execution flow

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

4.2 E0C6282 Restrictions

Note the following when generating a program by the E0C6282:

■ Melody ROM capacity

The capacity of the E0C6282 melody ROM is 128 words (00H to 7FH). Therefore, the number of melodies and the specification range of the ORG pseudo-instruction are limited to 00H-7FH. Specifying data above this range causes an error.

■ Scale ROM capacity

The capacity of the E0C6282 scale ROM is 32 words. Up to 32 scales (rest are included) can be specified. Specifying data above this range causes an error. C3-C6# or C4-C7# (with multiplier circuit) may be selected from among 31 scales.

■ Melody data

A melody data of the E0C6282 is composed of 10 bits / word.

| D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|-------------|-----------|----|----|--------------------|----|----|----|----|----------|
| Attack data | Note data | | | Scale address data | | | | | End data |

■ Melody HEX file

Set all unused ROM areas to FFH when writing the HEX file into the EPROM and installing on EVA6282. When loaded with ICE, the file format is checked, and an error results when it is not following format.

- Main ROM high-order data = 00H-03H
- Main ROM low-order data = 00H-FFH
- When 128 bytes are not used, the space is filled out with FFH.

4.3 MLA6282 Quick Reference

■ Starting command and input/output files

Execution file: MLA6282.EXE

_ indicates a blank.

☐ indicates the Return key.

A parameter enclosed by [] can be omitted.

Starting command: **MLA6282_ [drive-name:] source-file-name [.shp]_ [-H]☐**

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 HEX and document files are 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
 -H Activation of program to convert melody document file to melody HEX file.

Input file: C282XXX.MDT (Source file)
 C282XXXA.DOC (Melody document file, when -H option use)

Output file: C282XXXA.HEX (Melody HEX file)
 C282XXXA.DOC (Melody document file)
 C282XXX.MPR (Assembly listing file)

■ Display example

```

*** E0C6282 MELODY ASSEMBLER. - Ver 3.10 ***
EEEEEEEEEE P P P P P P P P S S S S S S S S 0 0 0 0 0 0 0 0 N N N N N N N N
EEEEEEEEEE P P P P P P P P P P S S S S S S S S 0 0 0 0 0 0 N N N N N N N N
EEE PPP P P P P P S S S S S S S S 0 0 0 0 0 0 N N N N N N N N
EEE P P P P P P P P S S S S S S S S 0 0 0 0 0 0 N N N N N N N N
EEEEEEEEEE P P P P P P P P P P S S S S S S S S 0 0 0 0 0 0 N N N N N N N N
EEEEEEEEEE P P P P P P P P S S S S S S S S 0 0 0 0 0 0 N N N N N N N N
EEE PPP P P P P P S S S S S S S S 0 0 0 0 0 0 N N N N N N N N
EEE P P P P P P P P S S S S S S S S 0 0 0 0 0 0 N N N N N N N N
EEEEEEEEEE P P P P P P P P S S S S S S S S 0 0 0 0 0 0 N N N N N N N N
EEEEEEEEEE P P P P P P P P S S S S S S S S 0 0 0 0 0 0 N N N N N N N N

(C) COPYRIGHT 1991 SEIKO EPSON CORP.

SOURCE FILE NAME IS " C282XXX.MDT "

THIS SOFTWARE MAKES NEXT FILES.

C282XXXA.HEX ... MELODY HEX FILE.
C282XXXA.DOC ... MELODY DOCUMENT FILE.
C282XXX.MPR ... MELODY ASSEMBLY FILE.

STRIKE ANY KEY

```

```

*** E0C6282 MELODY ASSEMBLER. - Ver 3.10 ***
EEEEEEEEEE P P P P P P P P S S S S S S S S 0 0 0 0 0 0 0 0 N N N N N N N N
EEEEEEEEEE P P P P P P P P P P S S S S S S S S 0 0 0 0 0 0 N N N N N N N N
EEE PPP P P P P P P S S S S S S S S 0 0 0 0 0 0 N N N N N N N N
EEE P P P P P P P P S S S S S S S S 0 0 0 0 0 0 N N N N N N N N
EEEEEEEEEE P P P P P P P P P P S S S S S S S S 0 0 0 0 0 0 N N N N N N N N
EEEEEEEEEE P P P P P P P P S S S S S S S S 0 0 0 0 0 0 N N N N N N N N
EEE PPP P P P P P S S S S S S S S 0 0 0 0 0 0 N N N N N N N N
EEE P P P P P P P P S S S S S S S S 0 0 0 0 0 0 N N N N N N N N
EEEEEEEEEE P P P P P P P P S S S S S S S S 0 0 0 0 0 0 N N N N N N N N
EEEEEEEEEE P P P P P P P P S S S S S S S S 0 0 0 0 0 0 N N N N N N N N

(C) COPYRIGHT 1991 SEIKO EPSON CORP.

SOURCE FILE NAME IS " C282XXX.DOC "

THIS SOFTWARE MAKES NEXT FILES.

C282XXXA.HEX ... MELODY HEX FILE.

STRIKE ANY KEY

```

Basic assembly (-H option not use)

With the message "STRIKE ANY KEY", the program is requesting key input for confirmation.

The program will proceed when any key is pressed.

To cancel the program, press the "CTRL" and "C" keys together. This will return you to the DOS command level.

-H option use

With the message "STRIKE ANY KEY", the program is requesting key input for confirmation. Check the source filename and option that you have input.

The program will proceed when any key is pressed.

To cancel the program, press the "CTRL" and "C" keys together. This will return you to the DOS command level.

■ Source format

Basic format

| | | | | |
|----------|--------|---------|-----------|-------------|
| <attack> | <note> | <scale> | <end bit> | <;comment> |
| 1 | 1 | C3 | | |
| : | : | : | | |
| 0 | 6 | A4# | 1 | ;1st Melody |

Attack 1: Attack output is performed. Or melody first word.
 0: Attack output is not performed.

Note 1-8 (When the "RR" is specified in <scale>, the rest may be selected.)

| No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|------|---|---|---|---|---|---|---|---|
| Note | | | | | | | | |

| No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|------|---|---|---|---|---|---|---|---|
| Rest | | | | | | | | |

Scale C3-C6# Maximum 31 types of scale can be selected.
 \$00-\$FD (for direct input)
 RR (for specifying the rest)

End bit 1: End of melody
 0 or omit: The others

| Scale | Scale data | | | | | | | | Hex. | Scale | Scale data | | | | | | | | Hex. |
|-------|------------|----|----|----|----|----|----|----|------|-------|------------|----|----|----|----|----|----|----|------|
| | S7 | S6 | S5 | S4 | S3 | S2 | S1 | S0 | | | S7 | S6 | S5 | S4 | S3 | S2 | S1 | S0 | |
| C3 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 04 | G4 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | B1 |
| C3# | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 12 | G4# | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | B5 |
| D3 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 20 | A4 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | B8 |
| D3# | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 2F | A4# | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | BC |
| E3 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 3B | B4 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | CD |
| F3 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 44 | C5 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | C4 |
| F3# | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 51 | C5# | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | C8 |
| G3 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 5B | D5 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | CD |
| G3# | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 65 | D5# | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | CE |
| A3 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 6C | E5 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | D3 |
| A3# | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 74 | F5 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | D4 |
| B3 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 7C | F5# | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | D9 |
| C4 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 84 | G5 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | DB |
| C4# | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 8D | G5# | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | DC |
| D4 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 92 | A5 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | DE |
| D4# | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 98 | A5# | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0E |
| E4 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 9E | B5 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | E2 |
| F4 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | A4 | C6 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | E4 |
| F4# | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | AB | C6# | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | E6 |

■ Pseudo-instructions

| Pseudo-instructions | Meaning | Example of use | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------------------------------|--|--------------------------|------|------|-----|------|-----|------|---|---|--|----|----|------|------|----|------|----|------|-----|---|---|----|----|----|----|----|----|--|----|------|----|----|-----|-----|-----|-----|--|
| ORG (Origin) | To define location counter | ORG 10H ORG 15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| .TEMPC0 = n (Tempo) .TEMPC1 = n | To select tempo (n = 0-15) | .TEMPC0=1 .TEMPC1=12 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | <table border="1"> <tr><td>n =</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr> <tr><td></td><td>30</td><td>32</td><td>34.3</td><td>36.9</td><td>40</td><td>43.6</td><td>48</td><td>53.3</td></tr> </table> <table border="1"> <tr><td>n =</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr> <tr><td></td><td>60</td><td>68.6</td><td>80</td><td>96</td><td>120</td><td>160</td><td>240</td><td>480</td></tr> </table> | n = | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | 30 | 32 | 34.3 | 36.9 | 40 | 43.6 | 48 | 53.3 | n = | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | | 60 | 68.6 | 80 | 96 | 120 | 160 | 240 | 480 | |
| n = | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 30 | 32 | 34.3 | 36.9 | 40 | 43.6 | 48 | 53.3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| n = | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 60 | 68.6 | 80 | 96 | 120 | 160 | 240 | 480 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| .OCTAVE=m (Octave) | To set scale range (m = 32 or 64) m = 32: C3-C6# m = 64: C4-C7# | .OCTAVE=32 .OCTAVE=64 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

■ Error messages

| Error message | Explanation |
|------------------------|--|
| S (Syntax Error) | Major syntax error <i>Error in scale field</i> Exceeded scale range: C3-C6# <i>Error in note field</i> Exceeded note range: 1-8 <i>Error in attack field</i> Number other than 0 or 1 was input. <i>Error in end bit field</i> Number other than 0 or 1 was input. |
| O (Scale ROM Overflow) | The definition exceeded the scale ROM capacity. |
| R (Range Error) | The value of the location counter exceeded the upper limit of the melody ROM capacity. Otherwise, the specified location exceeded the upper limit. |
| OPTION COMMAND MISSING | Options cannot be set. |
| FILE NAME ERROR | The source filename has eight or more characters. |
| FILE NOT PRESENT | The specified source file is not there. |
| DIRECTORY FULL | No more room in the directory of the specified disk. |
| FATAL DISK WRITE ERROR | The file cannot be written to the disk. |

5 FUNCTION OPTION GENERATOR FOG6282

5.1 FOG6282 Outline

With the 4-bit single-chip E0C6282 microcomputers, the customer may select 13 hardware options. By modifying the mask patterns of the E0C6282 according to the selected options, the system can be customized to meet the specifications of the target system.

The Function Option Generator FOG6282 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 FOG6282, the E0C6282 mask pattern is automatically generated by a general purpose computer. The HEX file for the evaluation board (EVA6282) 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.

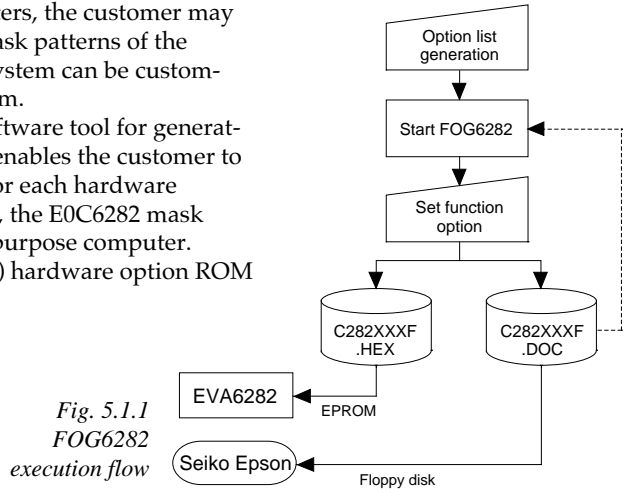


Fig. 5.1.1
FOG6282
execution flow

5.2 E0C6282 Option List

Multiple specifications are available in each option item as indicated in the Option List. Using "5.3 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. E0C6282(3.0V)
- 2. E0C62L82(1.5V)
- 3. E0C62A82(3.0V / TWIN CLOCK)

2. HEAVY LOAD PROTECTION FUNCTION FOR E0C6282/62A82

- 1. Not Use (Selection item of the heavy load protection circuit for E0C6282/62A82. Select "Not Use" when E0C62L82 is selected.)
- 2. Use

3. MULTIPLE KEY ENTRY RESET

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

4. INTERRUPT NOISE REJECTOR

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

5. INPUT PORT PULL DOWN RESISTOR

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

6. OUTPUT PORT OUTPUT SPECIFICATION (R00–R03)

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

7. R10 SPECIFICATION

- 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]
 10. FOUT OSC3 (Only E0C62A82)
- OUTPUT SPECIFICATION 1. Complementary 2. Pch-OpenDrain

8. R11 SPECIFICATION

- OUTPUT SPECIFICATION 1. Complementary 2. Pch-OpenDrain

9. MO, R12 SPECIFICATION

- OUTPUT TYPE 1. MO: Normal high, R12: D.C. output
 2. MO: Normal high, R12: Melody inverted output
 3. MO: Normal high, R12: Envelope (Nch-OpenDrain)
 4. MO: Normal low, R12: D.C. output
 5. MO: Normal low, R12: Melody inverted output
 6. MO: Normal low, R12: Envelope (Pch-OpenDrain)
- OUTPUT SPECIFICATION (When D.C. is selected for R12)
..... 1. Complementary 2. Pch-OpenDrain

10. I/O PORT 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

11. LCD COMMON DUTY

- 1. 1/4 Duty
- 2. 1/8 Duty

12. OSC3 SYSTEM CLOCK

- 1. Not Use
 - 2. Ceramic
 - 3. CR
- (Only E0C62A82. Select "Not Use" when E0C6282/62L82 is selected.)

13. OSC1 SYSTEM CLOCK

- 1. Crystal
 - 2. CR
- (Only E0C6282/62L82. Select "Crystal" when E0C62A82 is selected.)

5.3 Option Specifications and Selection Message

Screen that can be selected as function options set on the E0C6282 are shown below, and their specifications are also described.

1 Device type

```

*** OPTION NO.1 ***

--- DEVICE TYPE ---

    1. E0C6282 ( 3.0V )
    2. E0C62L82 ( 1.5V )
    3. E0C62A82 ( 3.0V/TWIN CLOCK )

PLEASE SELECT NO.(1) ? 1 

    1. E0C6282 ( 3.0V )  SELECTED
  
```

Select the chip specification.

E0C6282, E0C62L82 and E0C62A82 denote 3 V power source voltage specification, LOW POWER specification for 1.5 V power source voltage and TWIN CLOCK specification respectively.

When E0C6282 or E0C62L82 is selected, oscillation circuit OSC3 can not be selected.

When E0C62A82 is selected, oscillation circuit OSC1 is fixed at crystal oscillation.

2 Heavy load protection function for E0C6282/62A82

```

*** OPTION NO.2 ***

--- HEAVY LOAD PROTECTION ---

    1. NOT USE
    2. USE

PLEASE SELECT NO.(2) ? 2 

    2. USE  SELECTED
  
```

When E0C6282 or E0C62A82 is selected for "Device type", select whether the heavy load protection function will be used or not.

Select "NOT USE" when E0C62L82 is selected for "Device type".

In case of the E0C62L82, the heavy load protection function can be used regardless of this option selection.

3 Multiple key entry reset

```

*** OPTION NO.3 ***

--- MULTIPLE KEY ENTRY RESET ---

COMBINATION

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

PLEASE SELECT NO.(4) ? 4 

    4. USE K00,K01,K02,K03  SELECTED
  
```

The reset function is set when K00 through K03 are entered.

When "NOT USE" is selected, the reset function is not activated even if K00 through K03 are entered.

When "USE K00, K01" is selected, the system is reset immediately the K00 and K01 inputs go 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.

However, the system is reset when a high signal is input for more than a rule time (1–3 sec).

The system reset circuit is shown in Figure 5.3.1.

However, this function does not operate in the status where the oscillation circuit is reset by software and continues.

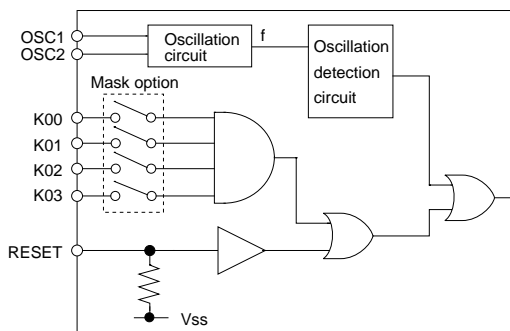


Fig. 5.3.1
System reset circuit

4 Interrupt noise rejector

```

*** OPTION NO.4 ***
--- INTERRUPT NOISE REJECTOR ---

      K00-K03          1. NOT USE
                      2. USE

PLEASE SELECT NO.(2) ? 2 [ ]

      K10              1. NOT USE
                      2. USE

PLEASE SELECT NO.(2) ? 2 [ ]

      K00-K03          2. USE   SELECTED
      K10              2. USE   SELECTED
    
```

Select whether noise rejector will be supplemented to the input interrupter 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.

5 Input ports pull down resistor

```

*** OPTION NO.5 ***
--- INPUT PORT PULL DOWN RESISTOR ---

      K00              1. WITH RESISTOR
                      2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1 [ ]

      K01              1. WITH RESISTOR
                      2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1 [ ]

      K02              1. WITH RESISTOR
                      2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1 [ ]

      K03              1. WITH RESISTOR
                      2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1 [ ]

      K10              1. WITH RESISTOR
                      2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1 [ ]

      K00              1. WITH RESISTOR   SELECTED
      K01              1. WITH RESISTOR   SELECTED
      K02              1. WITH RESISTOR   SELECTED
      K03              1. WITH RESISTOR   SELECTED
      K10              1. WITH RESISTOR   SELECTED
    
```

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 "H" level (VDD) to "L" (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 5.3.2.

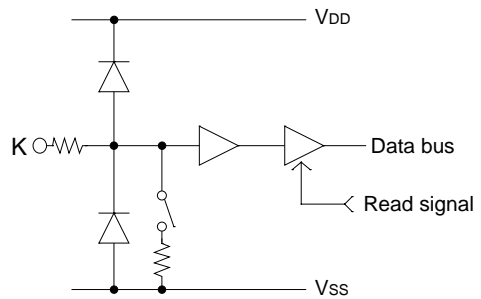


Fig. 5.3.2 Configuration of pull down resistor

6 Output port (R00–R03) output specification

```

*** OPTION NO.6 ***
--- OUTPUT PORT OUTPUT SPECIFICATION R00-R03 ---

      R00          1. COMPLEMENTARY
                  2. PCH-OPENDRAIN

PLEASE SELECT NO.(1)? 1 

      R01          1. COMPLEMENTARY
                  2. PCH-OPENDRAIN

PLEASE SELECT NO.(1)? 1 

      R02          1. COMPLEMENTARY
                  2. PCH-OPENDRAIN

PLEASE SELECT NO.(1)? 1 

      R03          1. COMPLEMENTARY
                  2. PCH-OPENDRAIN

PLEASE SELECT NO.(1)? 1 

      R00          1. COMPLEMENTARY   SELECTED
      R01          1. COMPLEMENTARY   SELECTED
      R02          1. COMPLEMENTARY   SELECTED
      R03          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 5.3.3.

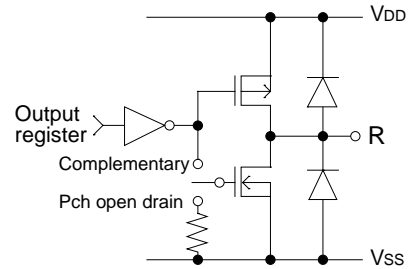


Fig. 5.3.3 Configuration of output circuit

7 R10 specification

```

*** OPTION NO.7 ***
--- R10 SPECIFICATION ---

      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]
                         10. FOUT  OSC3

PLEASE SELECT NO.(1) ? 1 

      OUTPUT SPECIFICATION  1. COMPLEMENTARY
                          2. PCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1 

      OUTPUT TYPE          1. D.C.   SELECTED
      OUTPUT SPECIFICATION  1. COMPLEMENTARY   SELECTED

```

"10. FOUT OSC3" is displayed only when the E0C62A82 is selected.

Select the output specification for R10 terminal. Either complementary output or Pch open drain output may be selected.

When DC output is selected, R10 becomes a regular output port. When FOUT is selected, clock with frequency selected from R10 terminal is generated by writing "1" to the R10 register.

- When DC output is selected
When R10 register (F4 address, D0 bit) is set to "1", the R10 terminal output goes high (VDD), and goes low (VSS) when set to "0".
Output waveform is shown in Figure 5.3.4.
- When FOUT output is selected
When FOUT bit (R10 register) is set to "1", 50% duty with VDD–VSS amplitude square wave is generated at the specified frequency. When set to "0", the FOUT terminal goes low (VSS).

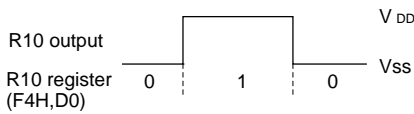


Fig. 5.3.4 Output waveform at DC output selection

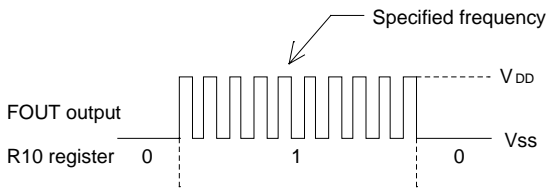


Fig. 5.3.5 Output waveform at R10 FOUT output selection

A FOUT frequency may be selected from among 9 types, ranging from 256 Hz to 32,768 Hz and OSC3. (OSC3 may be selected only when the E0C62A82 is selected for "Device type".

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 5.3.5.

8 R11 specification

```

*** OPTION NO.8 ***
--- R11 SPECIFICATION ---
      OUTPUT SPECIFICATION      1. COMPLEMENTARY
                                2. PCH-OPENDRAIN
PLEASE SELECT NO.(1) ? 1[ ]
      OUTPUT SPECIFICATION      1. COMPLEMENTARY      SELECTED
    
```

Select the output specification for 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 5.3.3).

9 MO, R12 specification

```

*** OPTION NO.9 ***
--- MO, R12 PORT OUTPUT SPECIFICATION ---
      OUTPUT TYPE                1. MO:NORMAL HIGH, R12:D.C.
                                2. MO:NORMAL HIGH, R12:/MO
                                3. MO:NORMAL HIGH, R12:ENV(NCH)
                                4. MO:NORMAL LOW, R12:D.C.
                                5. MO:NORMAL LOW, R12:/MO
                                6. MO:NORMAL LOW, R12:ENV(PCH)
PLEASE SELECT NO.(2) ? 1[ ]
      OUTPUT SPECIFICATION      1. COMPLEMENTARY
                                2. PCH-OPENDRAIN
PLEASE SELECT NO.(1) ? 1[ ]
      OUTPUT TYPE                1. MO:NORMAL HIGH, R12:D.C.      SELECTED
      OUTPUT SPECIFICATION      1. COMPLEMENTARY      SELECTED
    
```

The output specification may be selected only when the D.C. output is selected for R12 output type.

Select the specification for MO and R12 terminals.

Either normal high output or normal low output may be selected for the MO output type.

Any one of the following may be selected for the R12 output type: DC output, melody inverted output, or envelope.

When DC output is selected for the R12 output type, either complementary output or Pch open drain output may be selected for the output specification.

When melody inverted output (/MO) is selected, output specification is fixed to complementary output.

When "ENV" envelope is selected, an envelope can be added to the melody signal output from MO terminal.

The R12 terminal is used for that control. The output circuit configuration and output waveform of MO and R12 terminals by the option selection are shown in Figures 5.3.6–5.3.11.

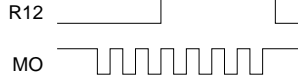
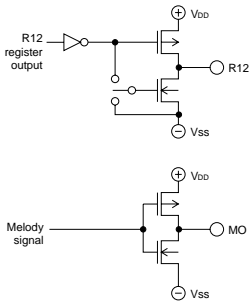


Fig. 5.3.6
Output circuit and output waveform
(MO: Normal high, R12: DC output)

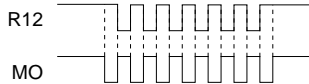
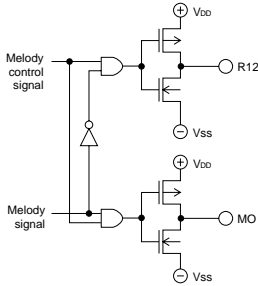


Fig. 5.3.7
Output circuit and output waveform
(MO: Normal high, R12: Melody inverted output)

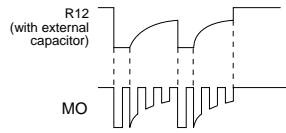
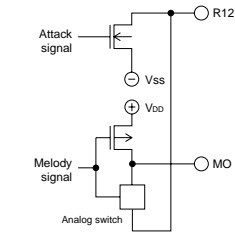


Fig. 5.3.8
Output circuit and output waveform
(MO: Normal high, R12: Envelope)

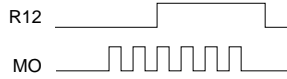
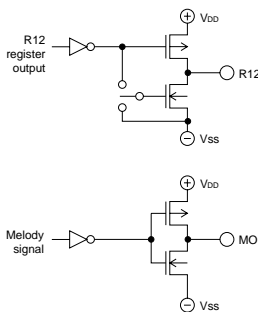


Fig. 5.3.9
Output circuit and output waveform
(MO: Normal low, R12: DC output)

(1) R12: DC output
(Melody output: Normal HIGH level)
Melody is output from the MO terminal and from the R12 terminal, data written on the "R12" register is output. The MO terminal is a complementary output terminal and goes high when melody is not played. Complementary output or Pch open drain output may be selected for the R12 terminal.

(2) R12: Melody inverted output
(Melody output: Normal HIGH level)
Using MO and R12 terminals, the piezo buzzer may be directly driven. During playing, inverted signal of the MO terminal is output from the R12 terminal. Both terminals go high when melody is not being played. The output configuration of both terminals becomes complementary.

(3) R12: With envelope function
(Melody output: Normal HIGH level)
Envelope function can be implemented by connecting an external capacitor to the R12 terminal. Melody is output from the MO terminal and the signal which will recharge the external capacitor will be output from the R12 terminal. The R12 electric potential will turn out supplying the negative electric potential of the MO terminal output and when the melody signal goes high, it will pass the analog switch and will be supplied to the MO terminal.

(4) R12: DC output
(Melody output: Normal LOW level)
Melody is output from the MO terminal and from the R12 terminal, data written on the "R12" register is output. The MO terminal is a complementary output terminal and goes low when melody is not played. Complementary output or Pch open drain output may be selected for the R12 terminal.

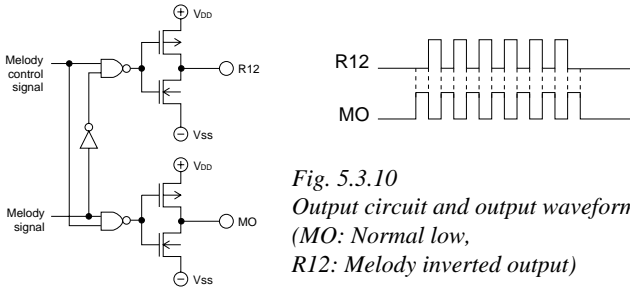


Fig. 5.3.10
Output circuit and output waveform
 (MO: Normal low,
 R12: Melody inverted output)

(5) R12: Melody inverted output (Melody output: Normal LOW level)
 Using MO and R12 terminals, the piezo buzzer may be directly driven. During playing, inverted signal of the MO terminal is output from the R12 terminal. Both terminals go low when melody is not being played. The output configuration of both terminals becomes complementary.

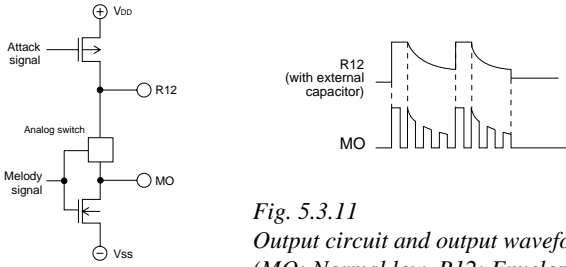


Fig. 5.3.11
Output circuit and output waveform
 (MO: Normal low, R12: Envelope)

(6) R12: With envelope function (Melody output: Normal LOW level)
 Envelope function can be implemented by connecting an external capacitor to the R12 terminal. Melody is output from the MO terminal and the signal which will recharge the external capacitor will be output from the R12 terminal. The R12 electric potential will turn out supplying the positive electric potential of the MO terminal output and when the melody signal goes low, it will pass the analog switch and will be supplied to the MO terminal.

10 I/O port specification

```

*** OPTION NO.10 ***
--- I/O PORT OUTPUT SPECIFICATION ---

      P00          1. COMPLEMENTARY
                  2. PCH-OPENDRAIN
PLEASE SELECT NO.(1) ? 1 [ ]

      P01          1. COMPLEMENTARY
                  2. PCH-OPENDRAIN
PLEASE SELECT NO.(1) ? 1 [ ]

      P02          1. COMPLEMENTARY
                  2. PCH-OPENDRAIN
PLEASE SELECT NO.(1) ? 1 [ ]

      P03          1. COMPLEMENTARY
                  2. PCH-OPENDRAIN
PLEASE SELECT NO.(1) ? 1 [ ]

      P00          1. COMPLEMENTARY   SELECTED
      P01          1. COMPLEMENTARY   SELECTED
      P02          1. COMPLEMENTARY   SELECTED
      P03          1. COMPLEMENTARY   SELECTED
    
```

Select the output specification to be used during I/O ports (P00–P03) 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 5.3.3).

Select complementary output for unused ports.

The I/O ports can control the input/output direction according to the IOC bit (FC address, 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.

11 LCD common duty

```

*** OPTION NO.11 ***
--- LCD COMMON DUTY ---

    1. 1/4 DUTY
    2. 1/8 DUTY

PLEASE SELECT NO.(2) ? 2 

    2. 1/8 DUTY   SELECTED

```

Select the common (drive) duty for the LCD segment. When 1/4 duty is selected, with 4 COM terminals and 42 SEG terminals, i.e., up to 168 segments may be driven; when 1/8 duty is selected, with 8 COM terminals and 38 SEG terminals, up to 304 segment drives will be possible.

12 OSC3 system clock

```

*** OPTION NO.12 ***
--- OSC3 SYSTEM CLOCK ---

    1. NOT USE
    2. CERAMIC
    3. CR

PLEASE SELECT NO.(1) ? 1 

    1. NOT USE   SELECTED

```

This option may be selected only when the E0C62A82 is selected.

Select oscillation circuit that uses OSC3 and OSC4 for the E0C62A82.

Select the "NOT USE" when the E0C6282/62L82 is selected for "Device type".

To minimize external components, CR oscillation circuit would be suitable; to obtain a stable oscillation frequency, ceramic oscillation circuit would be suitable.

When CR oscillation circuit is selected, only resistors are needed as external components since capacities are built-in. On the other hand, when ceramic oscillation circuit is selected, ceramic oscillator, gate capacity and drain capacity are needed as external components. Although when ceramic oscillation circuit is selected, it is fixed at 1 MHz, when CR oscillation circuit is selected, frequency may be modified to a certain extent depending on the resistance of external components.

13 OSC1 system clock

```

*** OPTION NO.13 ***
--- OSC1 SYSTEM CLOCK ---

    1. CRYSTAL
    2. CR

PLEASE SELECT NO.(1) ? 1 

    1. CRYSTAL   SELECTED

```

This option may not be selected when the E0C62A82 is selected.

Select oscillation circuit that uses OSC1 and OSC2 for the E0C6282/62L82.

Select the crystal oscillation when the E0C62A82 is selected for "Device type".

To minimize external components, CR oscillation circuit would be suitable; to obtain a stable oscillation frequency, crystal oscillation circuit would be suitable.

5.4 FOG6282 Quick Reference

■ Starting command and input/output files

Execution file: FOG6282.EXE

Starting command: **FOG6282** indicates the Return key.

Input file: C282XXXF.DOC (Function option document file, when modifying)

Output file: C282XXXF.DOC (Function option document file)
C282XXXF.HEX (Function option HEX file)

■ Display example

```

*** E0C6282 FUNCTION OPTION GENERATOR. --- Ver 3.10 ***
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 NNNNN
EEE PPP SSS 000 000 NNN NNNNN
EEE PPP SSS SSS 000 000 NNN NNNN
EEEEEEEEEE PPP SSSS SSS 000 000 NNN NNN
EEEEEEEEEE PPP SSSSSS 00000000 NNN NN

(C) COPYRIGHT 1991 SEIKO EPSON CORP.

THIS SOFTWARE MAKES NEXT FILES.

C282XXXF.HEX ... FUNCTION OPTION HEX FILE.
C282XXXF.DOC ... FUNCTION OPTION DOCUMENT FILE.

STRIKE ANY KEY.
    
```

Start-up message

When FOG6282 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.

```

*** E0C6282 USER'S OPTION SETTING. --- Ver 3.10 ***
CURRENT DATE IS 91/07/19
PLEASE INPUT NEW DATE : 91/07/22
    
```

Date input

Enter the 2-digit year, month, and day of the month by delimiting them with a slash ("/").

When not modifying the date, press the RETURN key "" to continue.

```

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

PLEASE SELECT NO.?
    
```

Operation selection menu

Enter a number from 1 to 3 to select a subsequent operation.

1. To set new function options.
2. To modify the document file.
3. To terminate FOG6282.

```

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

PLEASE SELECT NO.? 1
PLEASE INPUT FILE NAME? C2820A0 ..(1)
PLEASE INPUT USER'S NAME? SEIKO EPSON CORP. ..(2)
PLEASE INPUT ANY COMMENT
(ONE LINE IS 50 CHR)? TOKYO DESIGN CENTER ..(3)
? 421-8 HINO HINO-SHI TOKYO 191 JAPAN
? TEL 0425-84-2551
? FAX 0425-84-8512
? 
    
```

Setting new function options

Select "1" on the operation selection menu.

- (1) Enter the file name.
- (2) Enter the customer's company name.
- (3) Enter any comment.

(Within 50 characters x 10 lines)

Next, start function option setting from option No. 1.

```

PLEASE INPUT FILE NAME? C2820A0
EXISTS OVERWRITE(Y/N)? N
PLEASE INPUT FILE NAME? C2820B0
PLEASE INPUT USER'S NAME?
    
```

In case a function option document file with the same name as the file name specified in the current drive exists, the user is asked whether 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) ***

C2820A0      C2820B0      C2820C0      ..(1)

PLEASE INPUT FILE NAME? C2820A0□      ..(2)
PLEASE INPUT USER'S NAME? □      ..(3)
PLEASE INPUT ANY COMMENT
(OONE LINE IS 50 CHR)? □      ..(4)
PLEASE INPUT EDIT NO.? 4□      ..(5)
      :
(Modifying function option settings)
      :
PLEASE INPUT EDIT NO.? E□

```

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

```

*** SOURCE FILE(S) ***

FUNCTION OPTION DOCUMENT FILE IS NOT FOUND.

```

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

```

PLEASE INPUT FILE NAME? C2820N0□
FUNCTION OPTION DOCUMENT FILE IS NOT FOUND.
PLEASE INPUT FILE NAME?

```

```

*** OPTION NO.3 ***

- MULTIPLE KEY ENTRY RESET -

COMBINATION

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

PLEASE SELECT NO.(1) ? 2□

    2. Use K00,K01  SELECTED

```

```

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

*** OPTION EPROM SELECT MENU ***

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

PLEASE SELECT NO.? 2□      ..(2)

    2. 27C128  SELECTED

MAKING FILE(S) IS COMPLETED.

*** OPERATION SELECT MENU ***

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

PLEASE SELECT NO.?

```

Modifying function option settings

Select "2" on the operation selection menu.

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

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

- (5) Enter the number of the function option to be modified. When selection of one option is complete, the system prompts entry of another function option number. Repeat selection until all options to be modified are selected. Enter "E[□]" to end option setting. Then, move to the confirmation procedure for HEX file generation.

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 EVA6282, 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 EVA6282 options.

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

5.5 Sample File

■ Example of function option document file

```

* E0C6282 FUNCTION OPTION DOCUMENT V 3.10
*
* FILE NAME      C2820A0F.DOC
* USER'S NAME   SEIKO EPSON CORP.
* INPUT DATE    91/06/03
* COMMENT       TOKYO DESIGN CENTER
*               421-8 HINO HINO-SHI TOKYO 191 JAPAN
*               TEL 0425-84-2551
*               FAX 0425-84-8512
*
*
* OPTION NO.1
* < DEVICE TYPE >
*
*               E0C6282 ( 3.0V ) ----- SELECTED
OPT0101 01
OPT0102 01
*
* OPTION NO.2
* < HEAVY LOAD PROTECTION FUNCTION FOR E0C6282/62A82 >
*
*               USE ----- SELECTED
OPT0201 01
*
* OPTION NO.3
* < MULTIPLE KEY ENTRY RESET >
* COMBINATION   USE K00,K01,K02,K03 ----- SELECTED
OPT0301 04
*
* OPTION NO.4
* < INTERRUPT NOISE REJECTOR >
* K00-K03      USE ----- SELECTED
* K10          USE ----- SELECTED
OPT0401 01
OPT0402 01
*
* OPTION NO.5
* < INPUT PORT PULL DOWN RESISTOR >
* K00          WITH RESISTOR ----- SELECTED
* K01          WITH RESISTOR ----- SELECTED
* K02          WITH RESISTOR ----- SELECTED
* K03          WITH RESISTOR ----- SELECTED
* K10          WITH RESISTOR ----- SELECTED
OPT0501 01
OPT0502 01
OPT0503 01
OPT0504 01
OPT0505 01
*
* OPTION NO.6
* < OUTPUT PORT SPECIFICATION R00-R03 >
* R00          COMPLEMENTARY ----- SELECTED
* R01          COMPLEMENTARY ----- SELECTED
* R02          COMPLEMENTARY ----- SELECTED
* R03          COMPLEMENTARY ----- SELECTED
OPT0601 01
OPT0602 01
OPT0603 01
OPT0604 01
*
* OPTION NO.7
* < R10 SPECIFICATION >
* OUTPUT TYPE  D.C. ----- SELECTED
* OUTPUT SPECIFICATION COMPLEMENTARY ----- SELECTED
OPT0701 01
OPT0702 08

```

```

OPT0703 01
*
* OPTION NO.8
* < R11 SPECIFICATION >
*   OUTPUT SPECIFICATION    COMPLEMENTARY  -----  SELECTED
OPT0801 01
*
* OPTION NO.9
* < MO, R12 PORT OUTPUT SPECIFICATION >
*   OUTPUT TYPE             MO:NORMAL HIGH, R12:/MO -----  SELECTED
OPT0901 02
*
* OPTION NO.10
* < I/O PORT OUTPUT SPECIFICATION >
*   P00                     COMPLEMENTARY  -----  SELECTED
*   P01                     COMPLEMENTARY  -----  SELECTED
*   P02                     COMPLEMENTARY  -----  SELECTED
*   P03                     COMPLEMENTARY  -----  SELECTED
OPT1001 01
OPT1002 01
OPT1003 01
OPT1004 01
*
* OPTION NO.11
* < LCD COMMON DUTY >
*                               1/8 DUTY  -----  SELECTED
OPT1101 02
*
* OPTION NO.12
* < OSC3 SYSTEM CLOCK >
*                               NOT USE  -----  SELECTED
OPT1201 01
*
* OPTION NO.13
* < OSC1 SYSTEM CLOCK >
*                               CRYSTAL  -----  SELECTED
OPT1301 01
*
*
* SEIKO EPSON'S AREA
*
*
* OPTION NO.14
OPT1401 01
OPT1402 01
OPT1403 01
OPT1404 01
*
* OPTION NO.15
OPT1501 01
OPT1502 01
OPT1503 01
OPT1504 01
*
* OPTION NO.16
OPT1601 01
*
* OPTION NO.17
OPT1701 01
OPT1702 01
*
* OPTION NO.18
OPT1801 01
OPT1802 01
\\END

```

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

6 SEGMENT OPTION GENERATOR SOG6282

6.1 SOG6282 Outline

With the 4-bit single-chip E0C6282 microcomputers, the customer may select the LCD segment options. By modifying the mask patterns of the E0C6282 according to the selected options, the system can be customized to meet the specifications of the target system.

The Segment Option Generator SOG6282 is a software tool for generating data file used to generate mask patterns. From the data file created with SOG6282, the E0C6282 mask pattern is automatically generated by a general purpose computer. The HEX file for the evaluation board (EVA6282) segment 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.

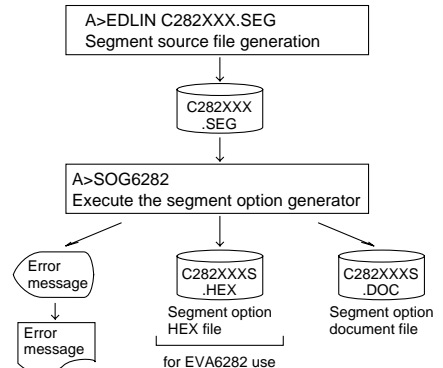


Fig. 6.1.1 SOG6282 execution flow

6.2 Option List

| Terminal name | Address | | | | | | | | | | | | | | | | | | | | | | | | Output specification | | |
|---------------|---------|---|---|------|---|---|------|---|---|------|---|---|------|---|---|------|---|---|------|---|---|------|---|---|----------------------|------------|---|
| | COM0 | | | COM1 | | | COM2 | | | COM3 | | | COM4 | | | COM5 | | | COM6 | | | COM7 | | | | | |
| | H | L | D | H | L | D | H | L | D | H | L | D | H | L | D | H | L | D | H | L | D | H | L | D | | | |
| SEG0 | | | | | | | | | | | | | | | | | | | | | | | | | | SEG output | <input type="checkbox"/> S |
| SEG1 | | | | | | | | | | | | | | | | | | | | | | | | | | DC output | <input type="checkbox"/> C <input type="checkbox"/> P |
| SEG2 | | | | | | | | | | | | | | | | | | | | | | | | | | SEG output | <input type="checkbox"/> S |
| SEG3 | | | | | | | | | | | | | | | | | | | | | | | | | | DC output | <input type="checkbox"/> C <input type="checkbox"/> P |
| SEG4 | | | | | | | | | | | | | | | | | | | | | | | | | | SEG output | <input type="checkbox"/> S |
| SEG5 | | | | | | | | | | | | | | | | | | | | | | | | | | DC output | <input type="checkbox"/> C <input type="checkbox"/> P |
| SEG6 | | | | | | | | | | | | | | | | | | | | | | | | | | SEG output | <input type="checkbox"/> S |
| SEG7 | | | | | | | | | | | | | | | | | | | | | | | | | | DC output | <input type="checkbox"/> C <input type="checkbox"/> P |
| SEG8 | | | | | | | | | | | | | | | | | | | | | | | | | | SEG output | <input type="checkbox"/> S |
| SEG9 | | | | | | | | | | | | | | | | | | | | | | | | | | DC output | <input type="checkbox"/> C <input type="checkbox"/> P |
| SEG10 | | | | | | | | | | | | | | | | | | | | | | | | | | SEG output | <input type="checkbox"/> S |
| SEG11 | | | | | | | | | | | | | | | | | | | | | | | | | | DC output | <input type="checkbox"/> C <input type="checkbox"/> P |
| SEG12 | | | | | | | | | | | | | | | | | | | | | | | | | | SEG output | <input type="checkbox"/> S |
| SEG13 | | | | | | | | | | | | | | | | | | | | | | | | | | DC output | <input type="checkbox"/> C <input type="checkbox"/> P |
| SEG14 | | | | | | | | | | | | | | | | | | | | | | | | | | SEG output | <input type="checkbox"/> S |
| SEG15 | | | | | | | | | | | | | | | | | | | | | | | | | | DC output | <input type="checkbox"/> C <input type="checkbox"/> P |
| SEG16 | | | | | | | | | | | | | | | | | | | | | | | | | | SEG output | <input type="checkbox"/> S |
| SEG17 | | | | | | | | | | | | | | | | | | | | | | | | | | DC output | <input type="checkbox"/> C <input type="checkbox"/> P |
| SEG18 | | | | | | | | | | | | | | | | | | | | | | | | | | SEG output | <input type="checkbox"/> S |
| SEG19 | | | | | | | | | | | | | | | | | | | | | | | | | | DC output | <input type="checkbox"/> C <input type="checkbox"/> P |
| SEG20 | | | | | | | | | | | | | | | | | | | | | | | | | | SEG output | <input type="checkbox"/> S |
| SEG21 | | | | | | | | | | | | | | | | | | | | | | | | | | DC output | <input type="checkbox"/> C <input type="checkbox"/> P |
| SEG22 | | | | | | | | | | | | | | | | | | | | | | | | | | SEG output | <input type="checkbox"/> S |
| SEG23 | | | | | | | | | | | | | | | | | | | | | | | | | | DC output | <input type="checkbox"/> C <input type="checkbox"/> P |
| SEG24 | | | | | | | | | | | | | | | | | | | | | | | | | | SEG output | <input type="checkbox"/> S |
| SEG25 | | | | | | | | | | | | | | | | | | | | | | | | | | DC output | <input type="checkbox"/> C <input type="checkbox"/> P |
| SEG26 | | | | | | | | | | | | | | | | | | | | | | | | | | SEG output | <input type="checkbox"/> S |
| SEG27 | | | | | | | | | | | | | | | | | | | | | | | | | | DC output | <input type="checkbox"/> C <input type="checkbox"/> P |
| SEG28 | | | | | | | | | | | | | | | | | | | | | | | | | | SEG output | <input type="checkbox"/> S |
| SEG29 | | | | | | | | | | | | | | | | | | | | | | | | | | DC output | <input type="checkbox"/> C <input type="checkbox"/> P |
| SEG30 | | | | | | | | | | | | | | | | | | | | | | | | | | SEG output | <input type="checkbox"/> S |
| SEG31 | | | | | | | | | | | | | | | | | | | | | | | | | | DC output | <input type="checkbox"/> C <input type="checkbox"/> P |
| SEG32 | | | | | | | | | | | | | | | | | | | | | | | | | | SEG output | <input type="checkbox"/> S |
| SEG33 | | | | | | | | | | | | | | | | | | | | | | | | | | DC output | <input type="checkbox"/> C <input type="checkbox"/> P |
| SEG34 | | | | | | | | | | | | | | | | | | | | | | | | | | SEG output | <input type="checkbox"/> S |
| SEG35 | | | | | | | | | | | | | | | | | | | | | | | | | | DC output | <input type="checkbox"/> C <input type="checkbox"/> P |
| SEG36 | | | | | | | | | | | | | | | | | | | | | | | | | | SEG output | <input type="checkbox"/> S |
| SEG37 | | | | | | | | | | | | | | | | | | | | | | | | | | DC output | <input type="checkbox"/> C <input type="checkbox"/> P |
| SEG38 | | | | | | | | | | | | | | | | | | | | | | | | | | SEG output | <input type="checkbox"/> S |
| SEG39 | | | | | | | | | | | | | | | | | | | | | | | | | | DC output | <input type="checkbox"/> C <input type="checkbox"/> P |
| SEG40 | | | | | | | | | | | | | | | | | | | | | | | | | | SEG output | <input type="checkbox"/> S |
| SEG41 | | | | | | | | | | | | | | | | | | | | | | | | | | DC output | <input type="checkbox"/> C <input type="checkbox"/> P |

Legend: <Address>
 H: High order address (9-D)
 L: Low order address (0-F)
 D: Data bit (0-3)

<Output specification>
 S: SEG output
 C: DC, complementary output
 P: DC, Pch open drain output

6.3 Segment Ports Output Specifications

For the output specification of the segment output ports, when selecting 1/4 duty, there are 42 segment terminals (SEG0–SEG41) and when selecting 1/8 duty, there are 38 segment terminals (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 (090H–0DFH) can be allocated to the optional segment. With this, up to 304 segments (168 segments when 1/4 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 (9–D), 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 that will not be used, write "---" (hyphen) in the H, L, and D columns.

Examples

- When 1/4 duty is selected

```
0 901 900 932 --- S
1 912 911 910 923 S
```

- When 1/8 duty is selected

```
0 901 900 932 903 B01 B00 B32 --- S
1 912 911 910 923 B12 B11 B10 B23 S
```

- When 1/8 duty has been selected, since the SEG38–SEG41 terminals become COM4–COM7 outputs, it is not necessary enter the allocation data.

■ When DC output is selected

The DC output can be selected in units of two terminals and up to 42 (when 1/4 duty is selected) or 38 (when 1/8 duty is selected) 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–COM7 becomes ineffective. Write three hyphens ("---") in the COM1–COM7 columns in the option list.

Example

- When complementary output is set to SEG24 and SEG25, and Pch open drain output is set to SEG26 and SEG27.

```
24 AE0 --- --- --- --- --- --- --- C
25 AF0 --- --- --- --- --- --- --- C
26 AE1 --- --- --- --- --- --- --- P
27 AF1 --- --- --- --- --- --- --- P
```

Note Only complementary output is enabled as the DC output of the SEG ports of EVA6282. Therefore, complementary output is enabled even if Pch open drain output is selected. Respond to it by adding external circuits as required.

6.4 SOG6282 Quick Reference

■ Starting command and input/output files

Execution file: SOG6282.EXE _ indicates a blank.
␣ indicates the Return key.
A parameter enclosed by [] can be omitted.

Starting command: **SOG6282_ [-H]␣**

Option: -H: Specifies the segment option document file for input file of SOG6282.

Input file: C282XXX.SEG (Segment option source file)
C282XXXS.DOC (Segment option document file, when -H option use)

Output file: C282XXXS.DOC (Segment option document file)
C282XXXS.HEX (Segment option HEX file)

■ Display example

```

*** E0C6282 SEGMENT OPTION GENERATOR. --- Ver 3.10 ***
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 SSS 000 000 NNNNN 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 NNNN
EEEEEEEEEE PPP SSSS SSS 000 000 NNN NNN
EEEEEEEEEE PPP SSSSSS 00000000 NNN NN

(C) COPYRIGHT 1991 SEIKO EPSON CORP.
SEGMENT OPTION SOURCE FILE NAME IS " C282XXX.SEG "
THIS SOFTWARE MAKES NEXT FILES.

C282XXXS.HEX ... SEGMENT OPTION HEX FILE.
C282XXXS.DOC ... SEGMENT OPTION DOCUMENT FILE.

STRIKE ANY KEY.
    
```

Start-up message

When SOG6282 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.

```

*** E0C6282 USER'S OPTION SETTING. --- Ver 3.10 ***
CURRENT DATE IS 91/07/19
PLEASE INPUT NEW DATE : 91/07/22␣
    
```

Date input

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

```

*** SOURCE FILE(S) ***
C2820A0 C2820B0 C2820C0 ..(1)
    
```

Input file selection

(1) Will display the files on the current drive.

```

*** SOURCE FILE(S) ***
SEGMENT OPTION SOURCE FILE IS NOT FOUND. ..(a) -H option not use
*** SOURCE FILE(S) ***
SEGMENT OPTION DOCUMENT FILE IS NOT FOUND. ..(b) -H option use
    
```

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

```

PLEASE INPUT SEGMENT OPTION FILE NAME? C2820A0␣ ..(2)
    
```

(2) Enter the file name.

```

PLEASE INPUT SEGMENT OPTION SOURCE FILE NAME? C2820N0␣
SEGMENT OPTION SOURCE FILE IS NOT FOUND. ..(c) -H option not use
PLEASE INPUT SEGMENT OPTION DOCUMENT FILE NAME? C2820A0␣
SEGMENT OPTION DOCUMENT FILE IS NOT FOUND. ..(d) -H option use
    
```

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

```
1/8 DUTY 38-SEG SELECT OK (Y/N)? Y  ..(3)
```

```
1/4 DUTY 42-SEG SELECT OK (Y/N)? ..(e) When the function option is set to 1/4 duty
```

```
C282XXXXF.DOC FILE IS NOT FOUND.
```

```
PLEASE INPUT USER'S NAME? SEIKO EPSON CORP.  ..(4)
PLEASE INPUT ANY COMMENT
(ONE LINE IS 50 CHR)? TOKYO DESIGN CENTER  ..(5)
? 421-8 HINO HINO-SHI TOKYO 191 JAPAN 
? TEL 0425-84-2551 
? FAX 0425-84-8512 
? 
```

(3) Refer to the file name input in number (2) + the "F.DOC" file from the current directory and indicate whether 1/4 or 1/8 duty has been selected.

If this duty is OK, enter a "Y" and if not enter an "N", then reset the function option using FOG6282.

When there is no file name input in number (2) + "F.DOC" file in the current directory, it terminates the program, displaying the message on the left.

(4) Enter the customer's company name.

(5) Enter any comment.
(Within 50 characters x 10 lines)

Then, move to the confirmation procedure for HEX file generation.

EPROM selection

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

```
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.
```

(1) When debugging the program with EVA6282, 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 EVA6282 options.

When a series of operations are complete, the SOG6282 generates files. If no error is committed while setting segment options, "MAKING FILE IS COMPLETED" will be displayed and the SOG6282 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. |

6.5 Sample Files

■ Example of segment option source file (when 1/8 is selected for LCD drive duty)

```

; C2820A0.SEG, VER.1.21
; EVA6282 LCD SEGMENT DECODE TABLE
;
0  901  900  932  903  B01  B00  B32  B03  S
1  912  911  910  923  B12  B11  B10  B23  S
2  913  920  921  922  B13  B20  B21  B22  S
3  933  902  930  931  B33  B02  B30  B31  S
4  941  940  972  943  B41  B40  B72  B43  S
5  952  951  950  963  B52  B51  B50  B63  S
6  953  960  961  962  B53  B60  B61  B62  S
7  973  942  970  971  B73  B42  B70  B71  S
8  981  980  9B2  983  B81  B80  BB2  B83  S
9  992  991  990  9A3  B92  B91  B90  BA3  S
10 993  9A0  9A1  9A2  B93  BA0  BA1  BA2  S
11 9B3  982  9B0  9B1  BB3  B82  BB0  BB1  S
12 9C1  9C0  9F2  9C3  BC1  BC0  BF2  BC3  S
13 9D2  9D1  9D0  9E3  BD2  BD1  BD0  BE3  S
14 9D3  9E0  9E1  9E2  BD3  BE0  BE1  BE2  S
15 9F3  9C2  9F0  9F1  BF3  BC2  BF0  BF1  S
16 A01  A00  A32  A03  C01  C00  C32  C03  S
17 A12  A11  A10  A23  C12  C11  C10  C23  S
18 A13  A20  A21  A22  C13  C20  C21  C22  S
19 A33  A02  A30  A31  C33  C02  C30  C31  S
20 A41  A40  A72  A43  C41  C40  C72  C43  S
21 A52  A51  A50  A63  C52  C51  C50  C63  S
22 A53  A60  A61  A62  C53  C60  C61  C62  S
23 A73  A42  A70  A71  C73  C42  C70  C71  S
24 A81  A80  AB2  A83  C81  C80  CB2  C83  S
25 A92  A91  A90  AA3  C92  C91  C90  CA3  S
26 A93  AA0  AA1  AA2  C93  CA0  CA1  CA2  S
27 AB3  A82  AB0  AB1  CB3  C82  CB0  CB1  S
28 AC1  AC0  AF2  AC3  CC1  CC0  CF2  CC3  S
29 AD2  AD1  AD0  AE3  CD2  CD1  CD0  CE3  S
30 AD3  AE0  AE1  AE2  CD3  CE0  CE1  CE2  S
31 AF3  AC2  AF0  AF1  CF3  CC2  CF0  CF1  S
32 D00  ---  ---  ---  ---  ---  ---  ---  C
33 D10  ---  ---  ---  ---  ---  ---  ---  C
34 D20  ---  ---  ---  ---  ---  ---  ---  C
35 D30  ---  ---  ---  ---  ---  ---  ---  C
36 D40  ---  ---  ---  ---  ---  ---  ---  C
37 D50  ---  ---  ---  ---  ---  ---  ---  C

```


■ Example of segment option source file (when 1/4 is selected for LCD drive duty)

```

; C2820B0.SEG, VER.1.21
; EVA6282 LCD SEGMENT DECODE TABLE
;
0  900  901  902  903  S
1  920  921  922  923  S
2  940  941  942  943  S
3  960  961  962  963  S
4  980  981  982  983  S
5  9A0  9A1  9A2  9A3  S
6  9C0  9C1  9C2  9C3  S
7  9E0  9E1  9E2  9E3  S
8  A00  A01  A02  A03  S
9  A20  A21  A22  A23  S
10 A40  A41  A42  A43  S
11 A60  A61  A62  A63  S
12 A80  A81  A82  A83  S
13 AA0  AA1  AA2  AA3  S
14 AC0  AC1  AC2  AC3  S
15 AE0  AE1  AE2  AE3  S
16 B00  B01  B02  B03  S
17 B20  B21  B22  B23  S
18 B40  B41  B42  B43  S
11 B60  B61  B62  B63  S
20 B80  B81  B82  B83  S
21 BA0  BA1  BA2  BA3  S
22 BC0  BC1  BC2  BC3  S
23 BE0  BE1  BE2  BE3  S
24 C00  C01  C02  C03  S
25 C20  C21  C22  C23  S
26 C40  C41  C42  C43  S
27 C60  C61  C62  C63  S
28 C80  C81  C82  C83  S
29 CA0  CA1  CA2  CA3  S
30 CC0  CC1  CC2  CC3  S
31 CE0  CE1  CE2  CE3  S
32 D00  ---  ---  ---  C
33 D10  ---  ---  ---  C
34 D20  ---  ---  ---  C
35 D30  ---  ---  ---  C
36 D40  ---  ---  ---  C
37 D50  ---  ---  ---  C
38 DC0  DC1  DC2  DC3  S
39 DD0  DD1  DD2  DD3  S
40 DE0  DE1  DE2  DE3  S
41 DF0  DF1  DF2  DF3  S

```

■ Example of segment option source file (when 1/8 is selected for LCD drive duty)

```

* E0C6282 SEGMENT OPTION DOCUMENT V 3.10
*
* FILE NAME      C2820A0S.DOC
* USER'S NAME   SEIKO EPSON CORP.
* INPUT DATE    91/02/01
* COMMENT       TOKYO DESIGN CENTER
*               421-8 HINO HINO-SHI TOKYO 191 JAPAN
*               TEL 0425-84-2551
*               FAX 0425-84-8512
*
* OPTION NO.19
*
* < LCD SEGMENT DECODE TABLE >
*
* SEG COM0 COM1 COM2 COM3 COM4 COM5 COM6 COM7 SPEC
*
  0  901  900  932  903  B01  B00  B32  B03  S
  1  912  911  910  923  B12  B11  B10  B23  S
  2  913  920  921  922  B13  B20  B21  B22  S
  3  933  902  930  931  B33  B02  B30  B31  S
  4  941  940  972  943  B41  B40  B72  B43  S
  5  952  951  950  963  B52  B51  B50  B63  S
  6  953  960  961  962  B53  B60  B61  B62  S
  7  973  942  970  971  B73  B42  B70  B71  S
  8  981  980  9B2  983  B81  B80  BB2  B83  S
  9  992  991  990  9A3  B92  B91  B90  BA3  S
 10  993  9A0  9A1  9A2  B93  BA0  BA1  BA2  S
 11  9B3  982  9B0  9B1  BB3  B82  BB0  BB1  S
 12  9C1  9C0  9F2  9C3  BC1  BC0  BF2  BC3  S
 13  9D2  9D1  9D0  9E3  BD2  BD1  BD0  BE3  S
 14  9D3  9E0  9E1  9E2  BD3  BE0  BE1  BE2  S
 15  9F3  9C2  9F0  9F1  BF3  BC2  BF0  BF1  S
 16  A01  A00  A32  A03  C01  C00  C32  C03  S
 17  A12  A11  A10  A23  C12  C11  C10  C23  S
 18  A13  A20  A21  A22  C13  C20  C21  C22  S
 19  A33  A02  A30  A31  C33  C02  C30  C31  S
 20  A41  A40  A72  A43  C41  C40  C72  C43  S
 21  A52  A51  A50  A63  C52  C51  C50  C63  S
 22  A53  A60  A61  A62  C53  C60  C61  C62  S
 23  A73  A42  A70  A71  C73  C42  C70  C71  S
 24  A81  A80  AB2  A83  C81  C80  CB2  C83  S
 25  A92  A91  A90  AA3  C92  C91  C90  CA3  S
 26  A93  AA0  AA1  AA2  C93  CA0  CA1  CA2  S
 27  AB3  A82  AB0  AB1  CB3  C82  CB0  CB1  S
 28  AC1  AC0  AF2  AC3  CC1  CC0  CF2  CC3  S
 29  AD2  AD1  AD0  AE3  CD2  CD1  CD0  CE3  S
 30  AD3  AE0  AE1  AE2  CD3  CE0  CE1  CE2  S
 31  AF3  AC2  AF0  AF1  CF3  CC2  CF0  CF1  S
 32  D00  D01  D02  D03  D60  D61  D62  D63  C
 33  D10  D11  D12  D13  D70  D71  D72  D73  C
 34  D20  D21  D22  D23  D80  D81  D82  D83  C
 35  D30  D31  D32  D33  D90  D91  D92  D93  C
 36  D40  D41  D42  D43  DA0  DA1  DA2  DA3  C
 37  D50  D51  D52  D53  DB0  DB1  DB2  DB3  C
 38  DC0  DC1  DC2  DC3  DC0  DC1  DC2  DC3  M
 39  DD0  DD1  DD2  DD3  DD0  DD1  DD2  DD3  M
 40  DE0  DE1  DE2  DE3  DE0  DE1  DE2  DE3  M
 41  DF0  DF1  DF2  DF3  DF0  DF1  DF2  DF3  M
\\END

```

Note End mark "~~¥~~END" may be used instead of "\\END" depending on the PC used.
(The code of \ and ¥ is 5CH.)
An "M" appearing at the point of an output specification (SPEC) indicates it is a COM terminal (COM4–COM7).

■ Example of segment option source file (when 1/4 is selected for LCD drive duty)

```

* E0C6282 SEGMENT OPTION DOCUMENT V 3.10
*
* FILE NAME      C2820B0S.DOC
* USER'S NAME   SEIKO EPSON CORP.
* INPUT DATE    91/11/29
* COMMENT       TOKYO DESIGN CENTER
*               421-8 HINO HINO-SHI TOKYO 191 JAPAN
*               TEL 0425-84-2551
*               FAX 0425-84-8512
*
* OPTION NO.19
*
* < LCD SEGMENT DECODE TABLE >
*
* SEG COM0 COM1 COM2 COM3 COM4 COM5 COM6 COM7 SPEC
*
  0  900  901  902  903  910  911  912  913  S
  1  920  921  922  923  930  931  932  933  S
  2  940  941  942  943  950  951  952  953  S
  3  960  961  962  963  970  971  972  973  S
  4  980  981  982  983  990  991  992  993  S
  5  9A0  9A1  9A2  9A3  9B0  9B1  9B2  9B3  S
  6  9C0  9C1  9C2  9C3  9D0  9D1  9D2  9D3  S
  7  9E0  9E1  9E2  9E3  9F0  9F1  9F2  9F3  S
  8  A00  A01  A02  A03  A10  A11  A12  A13  S
  9  A20  A21  A22  A23  A30  A31  A32  A33  S
 10  A40  A41  A42  A43  A50  A51  A52  A53  S
 11  A60  A61  A62  A63  A70  A71  A72  A73  S
 12  A80  A81  A82  A83  A90  A91  A92  A93  S
 13  AA0  AA1  AA2  AA3  AB0  AB1  AB2  AB3  S
 14  AC0  AC1  AC2  AC3  AD0  AD1  AD2  AD3  S
 15  AE0  AE1  AE2  AE3  AF0  AF1  AF2  AF3  S
 16  B00  B01  B02  B03  B10  B11  B12  B13  S
 17  B20  B21  B22  B23  B30  B31  B32  B33  S
 18  B40  B41  B42  B43  B50  B51  B52  B53  S
 19  B60  B61  B62  B63  B70  B71  B72  B73  S
 20  B80  B81  B82  B83  B90  B91  B92  B93  S
 21  BA0  BA1  BA2  BA3  BB0  BB1  BB2  BB3  S
 22  BC0  BC1  BC2  BC3  BD0  BD1  BD2  BD3  S
 23  BE0  BE1  BE2  BE3  CF0  BF1  BF2  BF3  S
 24  C00  C01  C02  C03  C10  C11  C12  C13  S
 25  C20  C21  C22  C23  C30  C31  C32  C33  S
 26  C40  C41  C42  C43  C50  C51  C52  C53  S
 27  C60  C61  C62  C63  C70  C71  C72  C73  S
 28  C80  C81  C82  C83  C90  C91  C92  C93  S
 29  CA0  CA1  CA2  CA3  CB0  CB1  CB2  CB3  S
 30  CC0  CC1  CC2  CC3  CD0  CD1  CD2  CD3  S
 31  CE0  CE1  CE2  CE3  CF0  CF1  CF2  CF3  S
 32  D00  D01  D02  D03  D11  D12  D13  D21  C
 33  D10  D22  D23  D31  D32  D33  D41  D42  C
 34  D20  D43  D51  D52  D53  D60  D61  D62  C
 35  D30  D63  D70  D71  D72  D73  D80  D81  C
 36  D40  D82  D83  D90  D91  D92  D93  DA0  C
 37  D50  DA1  DA2  DA3  DB0  DB1  DB2  DB3  C
 38  DC0  DC1  DC2  DC3  DB3  DB3  DB3  DB3  S
 39  DD0  DD1  DD2  DD3  DB3  DB3  DB3  DB3  S
 40  DE0  DE1  DE2  DE3  DB3  DB3  DB3  DB3  S
 41  DF0  DF1  DF2  DF3  DB3  DB3  DB3  DB3  S
\\END

```

Note Data corresponding to COM4–COM7 is disregarded.

7 ICE CONTROL SOFTWARE ICS6282

7.1 ICS6282 Outline

The In-circuit Emulator ICE6200 connects the target board produced by the user via the EVA6282 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 ICS6282.

The ICS6282 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.

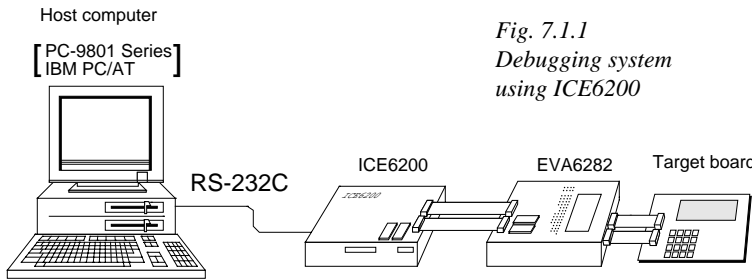


Fig. 7.1.1
Debugging system
using ICE6200

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.

7.2 ICS6282 Restrictions

Take the following precautions when using the ICS6282.

■ 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 0FCH. However, as the following addresses are in the unused area, designation of this area with the ICE commands produces an error.

Unused area: 0F5H, 0F7H to 0F8H

Memory 090H to 0DFH is display memory; 0E0H to 0FCH is I/O memory.
(Refer to the "E0C6282 Technical Manual" for details.)

■ Undefined Code

The instructions below are not specified for the E0C6282 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 ICS6282, OPTLD command can be used.

This command is used to load HEX files (melody, function option and segment option data for LCD) in the EVA6282 memory with the ICE6200. It is also necessary in order to play a melody using the ICE6200 and EVA6282.

Load of melody data: #OPTLD , 0 , C282XXXXA
 Load of function option data: #OPTLD , 1 , C282XXXXF
 Load of segment option data: #OPTLD , 2 , C282XXXXS

OPTLD *READ HEXA DATA FILE*

Format

```
#OPTLD, 0, <file name>
#OPTLD, 1, <file name>
#OPTLD, 2, <file name>
```

Function

- (1) Load melody HEX file in the EVA6282 melody data memory.
It is HEX file output by the melody assembler and has intel HEX format.
- (2) Load function option HEX file in the EVA6282 function option data memory.
It is HEX file output by the function option generator and has intel HEX format.
- (3) Load segment option HEX file in the EVA6282 segment option data memory.
It is HEX file output by the segment option generator and has intel HEX format.
Since it takes about 12 minutes to load segment option HEX data, when you want to load at high speed, execute this command by changing the EVA6282 operation clock from OSC1 to OSC3. (When OSC3 = 1 MHz, since it takes about 7 minutes to load segment option HEX data.)

Examples

```
#OPTLD, 0, C2XXYYY ..... C2XXYYA.HEX file is loaded in the melody data memory.
#OPTLD, 1, C2XXYYY ..... C2XXYYF.HEX file is loaded in the function option data memory.
#OPTLD, 2, C2XXYYY ..... C2XXYYYS.HEX file is loaded in the segment option data memory.

#SD, FC
FC 0:4 ..... The OSC3 oscillation is turned ON.
UNUSED AREA

#SD, FC
FC 4:C ..... Switching from OSC1 to OSC3.
```

7.3 ICS6282 Quick Reference

■ **Starting command and input/output files**

␣ indicates the Return key.

Execution file: ICS6282.BAT (ICS6282J.EXE) . . . for MS-DOS
 ICS6282B.BAT (ICS6282W.EXE) . . . for PC-DOS

Starting command: **ICS6282 (ICS6282J)**␣ . . . for MS-DOS
ICS6282B (ICS6282W)␣ . . . for PC-DOS

Input file: C282XXXL.HEX (Object file, low-order)
 C282XXXH.HEX (Object file, high-order)
 C282XXXD.HEX (Data RAM file)
 C282XXXC.HEX (Control file)
 C282XXXA.HEX (Melody HEX file)

Output file: C282XXXL.HEX (Object file, low-order)
 C282XXXH.HEX (Object file, high-order)
 C282XXXD.HEX (Data RAM file)
 C282XXXC.HEX (Control file)

■ **Display example**

```

*** E0C6282 ICE CONTROL SOFTWARE. --- Ver 3.01 ***
EEEEEEEEEE PPPPPPPP SSSSSSS 00000000 NNN NNN
EEEEEEEEEE PPPPPPPPP SSS SSSS 000 000 NNNN NNN
EEE PPP PPP SSS SSS 000 000 NNNNNN NNN
EEE PPP PPP SSS 000 000 NNNNNN NNN
EEEEEEEEEE PPPPPPPPP SSSSSSS 000 000 NNN NNN NNN
EEEEEEEEEE PPPPPPPPP SSSS 000 000 NNN NNNNNN
EEE PPP SSS SSS 000 000 NNN NNNNN
EEE PPP SSS SSS 000 000 NNN NNN
EEEEEEEEEE PPP SSS SSS 000 000 NNN NNN
EEEEEEEEEE PPP SSSSSSS 00000000 NNN NN
#
(C) COPYRIGHT 1991 SEIKO EPSON CORP.
* ICE POWER ON RESET *
* DIAGNOSTIC TEST OK *
    
```

Start-up message

When ICS6282 is started, the start-up message is displayed, and a self-test is automatically performed. ICS6282 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 ICS6282 program is terminated by entering the Q (Quit) command.

Note Confirm that the cables connected properly, then operate the ICS6282.

■ **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 EVA6282CPU 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 EVA6282CPU internal registers |
| | | #SR [↵] | Set EVA6282CPU internal registers |
| | | #I [↵] | Reset EVA6282CPU |
| | | #DXY [↵] | Display X, Y, MX and MY |
| | | #SXY [↵] | Set data for X and Y display and MX, MY |

| Item No. | Function | Command Format | Outline of Operation |
|-----------------|------------------------------------|-----------------|---|
| 11 | History | #H,p1,p2 ↵ | Display history data for pointer 1 and pointer 2 |
| | | #HB ↵ | Display upstream history data |
| | | #HG ↵ | Display 21 line history data |
| | | #HP ↵ | Display history pointer |
| | | #HPS,a ↵ | Set history pointer |
| | | #HC,S/C/E ↵ | Sets up the history information acquisition before (S), before/after (C) and after (E) |
| | | #HA,a1,a2 ↵ | Sets up the history information acquisition from program area a1 to a2 |
| | | #HAR,a1,a2 ↵ | Sets up the prohibition of the history information acquisition from program area a1 to a2 |
| | | #HAD ↵ | Indicates history acquisition program area |
| | | #HS,a ↵ | Retrieves and indicates the history information which executed a program address "a" |
| | | #HSW,a ↵ | Retrieves and indicates the history information which wrote or read the data area address "a" |
| 12 | File | #RF,file ↵ | Move program file to memory |
| | | #RFD,file ↵ | Move data file to memory |
| | | #VF,file ↵ | Compare program file and contents of memory |
| | | #VFD,file ↵ | Compare data file and contents of memory |
| | | #WF,file ↵ | Save contents of memory to program file |
| | | #WFD,file ↵ | Save contents of memory to data file |
| | | #CL,file ↵ | Load ICE6200 set condition from file |
| | | #CS,file ↵ | Save ICE6200 set condition to file |
| | | #OPTLD,0,file ↵ | Load melody data from file |
| | | #OPTLD,1,file ↵ | Load function option data from file |
| #OPTLD,2,file ↵ | Load segment option data from file | | |
| 13 | Coverage | #CVD ↵ | Indicates coverage information |
| | | #CVR ↵ | Clears coverage information |
| 14 | ROM Access | #RP ↵ | Move contents of ROM to program memory |
| | | #VP ↵ | Compare contents of ROM with contents of program memory |
| | | #ROM ↵ | Set ROM type |
| 15 | Terminate ICE | #Q ↵ | Terminate ICE and return to operating system control |
| 16 | Command Display | #HELP ↵ | Display ICE6200 instruction |
| 17 | Self Diagnosis | #CHK ↵ | Report results of ICE6200 self diagnostic test |

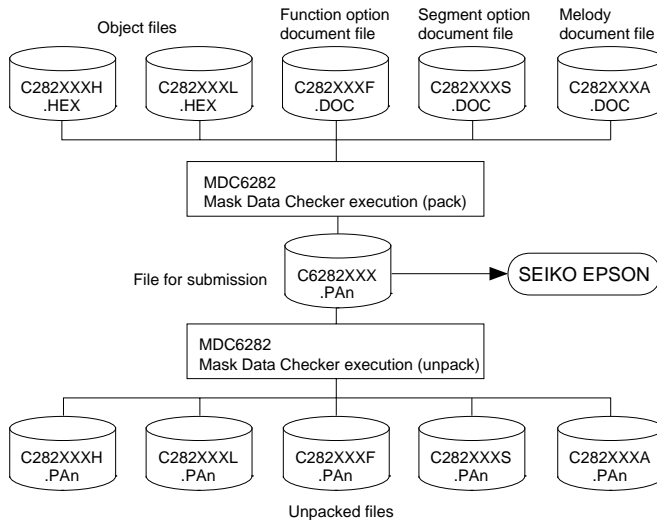
↵ means press the RETURN key.

8 MASK DATA CHECKER MDC6282

8.1 MDC6282 Outline

The Mask Data Checker MDC6282 is a software tool which checks the program data (C282XXXH.HEX and C282XXXL.HEX), option data (C282XXXF.DOC and C282XXXS.DOC), and melody data (C282XXXA.DOC) created by the user and creates the data file (C6282XXX.PAn) for generating mask patterns. The user must send the file generated through this software tool to Seiko Epson.

Moreover, MDC6282 has the capability to restore the generated data file (C6282XXX.PA0) to the original file format .



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

Fig. 8.1.1
MDC6282 execution flow

8.2 MDC6282 Quick Reference

■ Starting command and input/output files

Execution file: MDC6282.EXE

Starting command: **MDC6282**

indicates the Return key.

Input file:

- C282XXXL.HEX (Object file, low-order)
- C282XXXH.HEX (Object file, high-order)
- C282XXXA.DOC (Melody document file)
- C282XXXF.DOC (Function option document file)
- C282XXXS.DOC (Segment option document file)
- C6282XXX.PAn (Packed file)

When packing

When unpacking

Output file:

- C6282XXX.PAn (Packed file)
- C282XXXL.PAn (Object file, low-order)
- C282XXXH.PAn (Object file, high-order)
- C282XXXA.PAn (Melody document file)
- C282XXXF.PAn (Function option document file)
- C282XXXS.PAn (Segment option document file)

When packing

When unpacking

■ Display examples

```

*** E0C6282 PACK / UNPACK PROGRAM Ver 2.00 ***
EEEEEEEEEE PPPPPPPP SSSSSSS OOOOOOOO NNN NNN
EEEEEEEEEE PPPPPPPPPP SSS SSS 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 SSS SSS OOO OOO NNN NNN
EEEEEEEEEE PPP SSSSSS OOOOOOOO NNN NN
    
```

(C) COPYRIGHT 1991 SEIKO EPSON CORP.

--- OPERATION MENU ---

1. PACK
2. UNPACK

PLEASE SELECT NO.?

```

--- OPERATION MENU ---
1. PACK
2. UNPACK
PLEASE SELECT NO.? 1
    
```

C282XXXH.HEX -----+
C282XXXL.HEX -----+
C282XXXA.DOC -----+----- C282XXX.PAn (PACK FILE)
C282XXXF.DOC -----+
C282XXXS.DOC -----+
PLEASE INPUT PACK FILE NAME (C6282XXX.PAn) ? C62820A0.PA0 ... (2)

C2820A0H.HEX -----+
C2820A0L.HEX -----+
C2820A0A.DOC -----+----- C2820A0.PA0
C2820A0F.DOC -----+
C2820A0S.DOC -----+

Note Don't use the data generated with the -N option of the Cross Assembler (ASM6282) as program data. If the program data generated with the -N option of the Cross Assembler is packed, following message is displayed.

```

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

```

--- OPERATION MENU ---
1. PACK
2. UNPACK
PLEASE SELECT NO.? 2
    
```

PLEASE INPUT PACKED FILE NAME (C6282XXX.PAn) ? C62820A0.PA0 ... (2)

C62820A0.PA0 -----+----- C2820A0H.PA0
+----- C2820A0L.PA0
+----- C2820A0A.PA0
+----- C2820A0F.PA0
+----- C2820A0S.PA0

Start-up message

When MDC6282 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 "C6282XXX.PA0" has been submitted, the pack file name should be entered as "C6282XXX.PA1".)

With this, the mask file (C6282XXX.PAn) is generated, and the MDC6282 program will be terminated. Submit this file to Seiko Epson.

Unpacking of data

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

With this, the mask data file (C6282XXX.PAn) is restored to the original file format, and the MDC6282 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, melody and scale ROM. |
| 4. HEX DATA ERROR : RECORD TYPE. (NOT 00) | The record type of 1 line is not 00. |
| 5. HEX DATA ERROR : DATA. (NOT 00-FFh) | The data is not in the range between 00H and 0FFH. |
| 6. HEX DATA ERROR : TOO MANY DATA IN ONE LINE. | There are too many data in 1 line. |
| 7. HEX DATA ERROR : CHECK SUM. | The checksum is not correct. |
| 8. HEX DATA ERROR : END MARK. | The end mark is not : 00000001FF. |
| 9. HEX DATA ERROR : DUPLICATE. | There is duplicate definition of data in the same address. |
| 10. HEX DATA ERROR : DATA (NO FFh) | There is an undefined field in the HEX data. |

Function option data error

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

Segment option data error

| Error Message | Explanation |
|---|---|
| 1. SEGMENT DATA ERROR : START MARK. | The start mark is not "\SEGMENT". (during unpacking) * |
| 2. SEGMENT DATA ERROR : DATA. | The segment data is not correct. |
| 3. SEGMENT DATA ERROR : SEGMENT NUMBER. | The SEG No. is not correct. |
| 4. SEGMENT DATA ERROR : SPEC. | The output specification of the SEG terminal is not correct. |
| 5. SEGMENT DATA ERROR : END MARK. | The end mark is not "\\END" (packing) or "\END" (unpacking).* |

File error

| Error Message | Explanation |
|--|---|
| 1. <File_name> FILE IS NOT FOUND. | The file is not found or the file number set in CONFIG.SYS is less than 10. |
| 2. PACK FILE NAME (File_name) ERROR. | The packed input format for the file name is wrong. |
| 3. PACKED FILE NAME (File_name) ERROR. | The unpacked input format for the file name is wrong. |
| 4. VERSION NUMBER ERROR : X.DOC | A FOG6282, SOG6282, MLA6282 different from the version No. has been used. |

System error

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

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

APPENDIX A. E0C6282 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 |
| 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 | |
| | | XP, r* | 1 1 1 0 | 1 0 0 0 | 0 0 r1 r0 | | | | | | | | | | | | 5 | XP ← r | |
| | | XH, r | 1 1 1 0 | 1 0 0 0 | 0 1 r1 r0 | | | | | | | | | | | | 5 | XH ← r | |
| | | XL, r | 1 1 1 0 | 1 0 0 0 | 1 0 r1 r0 | | | | | | | | | | | | 5 | XL ← r | |
| | | YP, r* | 1 1 1 0 | 1 0 0 1 | 0 0 r1 r0 | | | | | | | | | | | | 5 | YP ← r | |
| | | YH, r | 1 1 1 0 | 1 0 0 1 | 0 1 r1 r0 | | | | | | | | | | | | 5 | YH ← r | |
| | | YL, r | 1 1 1 0 | 1 0 0 1 | 1 0 r1 r0 | | | | | | | | | | | | 5 | YL ← r | |
| | | r, XP* | 1 1 1 0 | 1 0 1 0 | 0 0 r1 r0 | | | | | | | | | | | | 5 | r ← XP | |
| | | r, XH | 1 1 1 0 | 1 0 1 0 | 0 1 r1 r0 | | | | | | | | | | | | 5 | r ← XH | |
| | | r, XL | 1 1 1 0 | 1 0 1 0 | 1 0 r1 r0 | | | | | | | | | | | | 5 | r ← XL | |
| | | r, YP* | 1 1 1 0 | 1 0 1 1 | 0 0 r1 r0 | | | | | | | | | | | | 5 | r ← YP | |
| | | r, YH | 1 1 1 0 | 1 0 1 1 | 0 1 r1 r0 | | | | | | | | | | | | 5 | r ← YH | |
| | | r, YL | 1 1 1 0 | 1 0 1 1 | 1 0 r1 r0 | | | | | | | | | | | | 5 | r ← YL | |
| | ADC | XH, i | 1 0 1 0 | 0 0 0 0 | i3 i2 i1 i0 | | | | | | | | | ↑ ↓ | 7 | XH ← XH+i3~i0+C | | | |
| | | XL, i | 1 0 1 0 | 0 0 0 1 | i3 i2 i1 i0 | | | | | | | | | ↑ ↓ | 7 | XL ← XL+i3~i0+C | | | |
| | | YH, i | 1 0 1 0 | 0 0 1 0 | i3 i2 i1 i0 | | | | | | | | | ↑ ↓ | 7 | YH ← YH+i3~i0+C | | | |
| | | YL, i | 1 0 1 0 | 0 0 1 1 | i3 i2 i1 i0 | | | | | | | | | ↑ ↓ | 7 | YL ← YL+i3~i0+C | | | |

"*" mean "not in E0C6282 Series".

| 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 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | | | | | 5 | M(X) ← 13~10, M(X+1) ← 17~14, X ← X+2 | |
| Flag operation instructions | SET | F, i | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | i3 | i2 | i1 | i0 | ↑ | ↑ | ↑ | ↑ | 7 | F ← FV i3~i0 |
| | RST | F, i | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | i3 | i2 | i1 | i0 | ↓ | ↓ | ↓ | ↓ | 7 | F ← F∧ i3~i0 |
| | SCF | | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | | | ↑ | | 7 | C ← 1 |
| | RCF | | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | | | ↓ | | 7 | C ← 0 |
| | SZF | | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | | | ↑ | | 7 | Z ← 1 |
| | RZF | | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | | | ↓ | | 7 | Z ← 0 |
| | SDF | | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | | | ↑ | | 7 | D ← 1 (Decimal Adjuster ON) |
| | RDF | | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | | | ↓ | | 7 | D ← 0 (Decimal Adjuster OFF) |
| | EI | | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | | | ↑ | | 7 | I ← 1 (Enables Interrupt) |
| DI | | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | | | ↓ | | 7 | I ← 0 (Disables Interrupt) | |
| Stack operation instructions | INC | SP | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | | | | | 5 | SP ← SP+1 |
| | DEC | SP | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | | | | | 5 | SP ← SP-1 |
| | PUSH | r | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | r1 | r0 | | | | | 5 | SP ← SP-1, M(SP) ← r |
| | | XP * | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | | | | | 5 | SP ← SP-1, M(SP) ← XP |
| | | XH | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | | | | | 5 | SP ← SP-1, M(SP) ← XH |
| | | XL | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | | | | | 5 | SP ← SP-1, M(SP) ← XL |
| | | YP * | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | | | | | 5 | SP ← SP-1, M(SP) ← YP |
| | | YH | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | | | | | 5 | SP ← SP-1, M(SP) ← YH |
| | | YL | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | | | | | 5 | SP ← SP-1, M(SP) ← YL |
| | | F | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | | | | | 5 | SP ← SP-1, M(SP) ← F |
| | POP | r | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | r1 | r0 | | | | | 5 | r ← M(SP), SP ← SP+1 |
| | | XP * | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | | | | | 5 | XP ← M(SP), SP ← SP+1 |
| | | XH | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | | | | | 5 | XH ← M(SP), SP ← SP+1 |
| | | XL | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | | | | | 5 | XL ← M(SP), SP ← SP+1 |
| YP * | | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | | | | | 5 | YP ← M(SP), SP ← SP+1 | |

"*" mean "not in E0C6282 Series".

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

Abbreviations used in the explanations have the following meanings.

Symbols associated with registers and memory

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

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

Symbols associated with program counter

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

Symbols associated with flags

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

Associated with immediate data

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

Associated with arithmetic and other operations

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

APPENDIX B. E0C6282 RAM MAP

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

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

APPENDIX C. E0C6282 I/O MEMORY MAP

| Address | Register | | | | Name | Init | 1 | 0 | Comment |
|---------|----------|-------|-------|-------|-------|------|---------|---|--|
| | D3 | D2 | D1 | D0 | | | | | |
| 0E0H | K03 | K02 | K01 | K00 | K03 | - | High | Low | Input port K03 |
| | R | | | | K02 | - | High | Low | Input port K02 |
| | R | | | | K01 | - | High | Low | Input port K01 |
| | R | | | | K00 | - | High | Low | Input port K00 |
| 0E1H | 0 | 0 | 0 | K10 | 0 | | | | |
| | R | | | | 0 | | | | |
| 0E2H | SWL3 | SWL2 | SWL1 | SWL0 | SWL3 | 0 | | | Stopwatch timer data 3 (1/100 sec) MSB |
| | R | | | | SWL2 | 0 | | | Stopwatch timer data 2 (1/100 sec) |
| | R | | | | SWL1 | 0 | | | Stopwatch timer data 1 (1/100 sec) |
| | R | | | | SWL0 | 0 | | | Stopwatch timer data 0 (1/100 sec) LSB |
| 0E3H | SWH3 | SWH2 | SWH1 | SWH0 | SWH3 | 0 | | | Stopwatch timer data 3 (1/10 sec) MSB |
| | R | | | | SWH2 | 0 | | | Stopwatch timer data 2 (1/10 sec) |
| | R | | | | SWH1 | 0 | | | Stopwatch timer data 1 (1/10 sec) |
| | R | | | | SWH0 | 0 | | | Stopwatch timer data 0 (1/10 sec) LSB |
| 0E4H | TM3 | TM2 | TM1 | TM0 | TM3 | - | High | Low | Timer data 3 (2 Hz) |
| | R | | | | TM2 | - | High | Low | Timer data 2 (4 Hz) |
| | R | | | | TM1 | - | High | Low | Timer data 1 (8 Hz) |
| | R | | | | TM0 | - | High | Low | Timer data 0 (16 Hz) |
| 0E5H | KCP03 | KCP02 | KCP01 | KCP00 | KCP03 | 0 | Falling | Rising | Input comparison register (K03) |
| | R/W | | | | KCP02 | 0 | Falling | Rising | Input comparison register (K02) |
| | R/W | | | | KCP01 | 0 | Falling | Rising | Input comparison register (K01) |
| | R/W | | | | KCP00 | 0 | Falling | Rising | Input comparison register (K00) |
| 0E6H | 0 | 0 | 0 | KCP10 | 0 | | | | |
| | R | | | | 0 | | | | |
| 0E7H | 0 | 0 | 0 | EIMEL | 0 | | | | |
| | R | | | | 0 | | | | |
| | R | | | | 0 | | | | |
| | R | | | | EIMEL | 0 | Enable | Mask | Interrupt mask register (melody) |
| 0E8H | EIK03 | EIK02 | EIK01 | EIK00 | EIK03 | 0 | Enable | Mask | Interrupt mask register (K03) |
| | R/W | | | | EIK02 | 0 | Enable | Mask | Interrupt mask register (K02) |
| | R/W | | | | EIK01 | 0 | Enable | Mask | Interrupt mask register (K01) |
| | R/W | | | | EIK00 | 0 | Enable | Mask | Interrupt mask register (K00) |
| 0E9H | 0 | 0 | 0 | EIK10 | 0 | | | | |
| | R | | | | 0 | | | | |
| 0EAH | 0 | 0 | EISW1 | EISW0 | 0 | | | | |
| | R | | R/W | | 0 | | | | |
| | R | | R/W | | EISW1 | 0 | Enable | Mask | Interrupt mask register (Stopwatch 1 Hz) |
| | R | | R/W | | EISW0 | 0 | Enable | Mask | Interrupt mask register (Stopwatch 10 Hz) |
| 0EBH | 0 | EIT2 | EIT8 | EIT32 | 0 | | | | |
| | R | R/W | | | EIT2 | 0 | Enable | Mask | Interrupt mask register (Clock timer 2 Hz) |
| | R | R/W | | | EIT8 | 0 | Enable | Mask | Interrupt mask register (Clock timer 8 Hz) |
| 0ECH | 0 | 0 | 0 | IMEL | 0 | | | | |
| | R | | | | 0 | | | | |
| | R | | | | 0 | | | | |
| | R | | | | IMEL | 0 | Yes | No | Interrupt factor flag (melody) |
| 0EDH | 0 | 0 | IK1 | IK0 | 0 | | | | |
| | R | | | | 0 | | | | |
| | R | | | | IK1 | 0 | Yes | No | Interrupt factor flag (K10) |
| 0EEH | 0 | 0 | ISW1 | ISW0 | 0 | | | | |
| | R | | | | 0 | | | | |
| | R | | | | ISW1 | 0 | Yes | No | Interrupt factor flag (Stopwatch 1 Hz) |
| | R | | | | ISW0 | 0 | Yes | No | Interrupt factor flag (Stopwatch 10 Hz) |
| 0EFH | 0 | IT2 | IT8 | IT32 | 0 | | | | |
| | R | | | | 0 | | | | |
| | R | | | | IT2 | 0 | Yes | No | Interrupt factor flag (Clock timer 2 Hz) |
| | R | | | | IT8 | 0 | Yes | No | Interrupt factor flag (Clock timer 8 Hz) |
| R | | | | IT32 | 0 | Yes | No | Interrupt factor flag (Clock timer 32 Hz) | |

| Address | Register | | | | Name | Init | 1 | 0 | Comment |
|---------|----------|-------|-------|-------|--------|--------|---------|---------------------------------------|-------------------------------------|
| | D3 | D2 | D1 | D0 | | | | | |
| 0F0H | MAD3 | MAD2 | MAD1 | MAD0 | MAD3 | 0 | High | Low | Melody ROM address AD3 |
| | R/W | | | | MAD2 | 0 | High | Low | Melody ROM address AD2 |
| | R/W | | | | MAD1 | 0 | High | Low | Melody ROM address AD1 |
| | R/W | | | | MAD0 | 0 | High | Low | Melody ROM address AD0 (LSB) |
| 0F1H | 0 | MAD6 | MAD5 | MAD4 | 0 | 0 | High | Low | Melody ROM address AD6 (MSB) |
| | R | R/W | | | MAD5 | 0 | High | Low | Melody ROM address AD5 |
| | R/W | | | | MAD4 | 0 | High | Low | Melody ROM address AD4 |
| 0F2H | CLKC1 | CLKC0 | TEMPC | MELC | CLKC1 | 0 | High | Low | Playing speed control |
| | R/W | | | | CLKC0 | 0 | High | Low | Playing speed control |
| | R/W | | | | TEMPC | 0 | High | Low | Tempo control |
| | R/W | | | | MELC | 0 | On | Off | Melody On/Off control |
| 0F3H | R03 | R02 | R01 | R00 | R03 | 0 | High | Low | Output port R03 |
| | R/W | | | | R02 | 0 | High | Low | Output port R02 |
| | R/W | | | | R01 | 0 | High | Low | Output port R01 |
| | R/W | | | | R00 | 0 | High | Low | Output port R00 |
| 0F4H | MELD | R12 | R11 | R10 | MELD | 0 | Disable | Enable | Melody output mask |
| | | MO | | MO | 0 | High | Low | Output port R12 | |
| | | ENV | | ENV | High-Z | - | - | - | Melody inverted output |
| | R/W | | | | ENV | High-Z | - | - | Envelope control |
| | R/W | | | | R11 | 0 | High | Low | Output port R11 |
| | R/W | | | | R10 | 0 | High | Low | Output port R10 |
| R/W | | | | FOUT | 0 | On | Off | Frequency output | |
| 0F6H | P03 | P02 | P01 | P00 | P03 | - | High | Low | I/O port P03 |
| | R/W | | | | P02 | - | High | Low | I/O port P02 |
| | R/W | | | | P01 | - | High | Low | I/O port P01 |
| | R/W | | | | P00 | - | High | Low | I/O port P00 |
| 0F9H | 0 | TMRST | SWRUN | SWRST | 0 | Reset | Reset | - | Clock timer reset |
| | R | W | R/W | W | SWRUN | 0 | Run | Stop | Stopwatch timer Run/Stop |
| | R/W | | | | SWRST | Reset | Reset | - | Stopwatch timer reset |
| 0FAH | HLMOD | 0 | SVDDT | SVDON | HLMOD | 0 | Heavy | Normal | Heavy load protection mode register |
| | R/W | | | | 0 | 0 | Low | Normal | Supply voltage detection data |
| | R/W | | | | SVDDT | 0 | Low | Normal | Supply voltage detection data |
| R/W | | | | SVDON | 0 | On | Off | Supply voltage detection On/Off | |
| 0FBH | CSDC | 0 | CMPDT | CMPON | CSDC | 0 | Static | Dynamic | LCD drive switch |
| | R/W | | | | 0 | 0 | Static | Dynamic | LCD drive switch |
| | R/W | | | | CMPDT | 1 | + > - | - > + | Comparator's voltage condition |
| R/W | | | | CMPON | 0 | On | Off | Comparator On/Off | |
| 0FCH | CLKCHG | OSCC | 0 | IOC | CLKCHG | 0 | OSC3 | OSC1 | CPU CLK switch |
| | R/W | | | | OSCC | 0 | On | Off | OSC3 oscillation On/Off |
| | R/W | | | | 0 | 0 | On | Off | OSC3 oscillation On/Off |
| R/W | | | | IOC | 0 | Out | In | I/O port P00–P03 input/output control | |

APPENDIX D. 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? <table style="margin-left: 40px; border: none;"> <tr> <td>MS-DOS</td> <td>ICS6282J.EXE</td> </tr> <tr> <td>PC-DOS</td> <td>ICS6282W.EXE</td> </tr> </table> • Is the DOS version correct? <table style="margin-left: 40px; border: none;"> <tr> <td>MS-DOS</td> <td>Ver. 3.1 or later</td> </tr> <tr> <td>PC-DOS</td> <td>Ver. 2.1 or later</td> </tr> </table> • 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? | MS-DOS | ICS6282J.EXE | PC-DOS | ICS6282W.EXE | MS-DOS | Ver. 3.1 or later | PC-DOS | Ver. 2.1 or later |
| | MS-DOS | ICS6282J.EXE | | | | | | | | |
| | PC-DOS | ICS6282W.EXE | | | | | | | | |
| | MS-DOS | Ver. 3.1 or later | | | | | | | | |
| | PC-DOS | Ver. 2.1 or later | | | | | | | | |
| | 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 EVA6282 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 ICS6282P.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> LD A, B Data in the B register is loaded into the A register.</p> <p style="margin-left: 100px;">LD B, 0A Immediate value A is loaded into the B register.</p> | | | | | | | | | |
| <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. | | | | | | | | | |
| SOG6282 | 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 |
|---------|--|---|
| ASM6282 | 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. |
| MDC6282 | 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? |
| MLA6282 | No melody is output. | Check the following and remedy if necessary: <ul style="list-style-type: none"> • Has the OPTLD command of the ICE6200 been executed? (When the ICE6200 is connected to the EVA6282) • Is the MELODY ROM installed? (When the EVA6282 is used independently) • Is the attack bit of the melody data set to "1"? |
| EVA6282 | The EVA6282 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, 3 A, 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 EVA6282 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 EVA6282 top cover been turned to a lower setting? |

APPENDIX E. CAUTION

USERS NOT UTILIZING THE MELODY FUNCTION → **See Item 1**

USERS NOT UTILIZING THE LCD SEGMENT → **See Item 2**

PLEASE BE SURE TO READ THE FOLLOWING:

1. To users not utilizing the melody function

Even if the melody function will not be utilized, create the source file listed below which is just for option setting instructions; operate the melody assembler (MLA6282) and produce the melody document file. Without the melody document file, error will be detected by the mask data checker (MDC6282) and mask data will not be produced.

Source file

```
.TEMPC0=5
.TEMPC1=8
.OCTAVE=32
```

- Melody document file corresponding to the above source file

```
:1000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF00
:1000100FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0
:1000200FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE0
:1000300FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFD0
:1000400FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC0
:1000500FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFB0
:1000600FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFA0
:1000700FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF90
:00000001FF
:1000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF00
:1000100FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0
:1000200FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE0
:1000300FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFD0
:1000400FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC0
:1000500FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFB0
:1000600FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFA0
:1000700FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF90
:00000001FF
:1000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF00
:1000100FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC42B
:00000001FF
*   E0C6282 MELODY OPTION DOCUMENT V.3.10
*
* OPTION NO.20
*   OCTAVE CIRCUIT
*   32KHZ ----- SELECTED
OPT2001 01
*
* OPTION NO.21
* < MELODY TEMPO SELECTION >
*   TEMPC0 TEMPO 5 ----- SELECTED
*   TEMPC1 TEMPO 8 ----- SELECTED
OPT2101 03
OPT2102 04
OPT2103 02
OPT2104 04
\\END
```

2. To users not utilizing the LCD segment

Even if the LCD segment will not be utilized, create the segment source file listed below; operate the segment option generator (SOG6282) and produce the segment document file.

Without the segment document file, error will be detected by the mask data checker (MDC6282) and mask data will not be produced.

Moreover, selects 1/4 duty for the LCD drive duty.

Source file

| | | | | | |
|----|-----|-----|-----|-----|---|
| 0 | --- | --- | --- | --- | S |
| 1 | --- | --- | --- | --- | S |
| 2 | --- | --- | --- | --- | S |
| 3 | --- | --- | --- | --- | S |
| 4 | --- | --- | --- | --- | S |
| 5 | --- | --- | --- | --- | S |
| 6 | --- | --- | --- | --- | S |
| 7 | --- | --- | --- | --- | S |
| 8 | --- | --- | --- | --- | S |
| 9 | --- | --- | --- | --- | S |
| 10 | --- | --- | --- | --- | S |
| 11 | --- | --- | --- | --- | S |
| 12 | --- | --- | --- | --- | S |
| 13 | --- | --- | --- | --- | S |
| 14 | --- | --- | --- | --- | S |
| 15 | --- | --- | --- | --- | S |
| 16 | --- | --- | --- | --- | S |
| 17 | --- | --- | --- | --- | S |
| 18 | --- | --- | --- | --- | S |
| 19 | --- | --- | --- | --- | S |
| 20 | --- | --- | --- | --- | S |
| 21 | --- | --- | --- | --- | S |
| 22 | --- | --- | --- | --- | S |
| 23 | --- | --- | --- | --- | S |
| 24 | --- | --- | --- | --- | S |
| 25 | --- | --- | --- | --- | S |
| 26 | --- | --- | --- | --- | S |
| 27 | --- | --- | --- | --- | S |
| 28 | --- | --- | --- | --- | S |
| 29 | --- | --- | --- | --- | S |
| 30 | --- | --- | --- | --- | S |
| 31 | --- | --- | --- | --- | S |
| 32 | --- | --- | --- | --- | S |
| 33 | --- | --- | --- | --- | S |
| 34 | --- | --- | --- | --- | S |
| 35 | --- | --- | --- | --- | S |
| 36 | --- | --- | --- | --- | S |
| 37 | --- | --- | --- | --- | S |
| 38 | --- | --- | --- | --- | S |
| 39 | --- | --- | --- | --- | S |
| 40 | --- | --- | --- | --- | S |
| 41 | --- | --- | --- | --- | S |

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