EPSON

CMOS 4-bit Single Chip Microcomputer

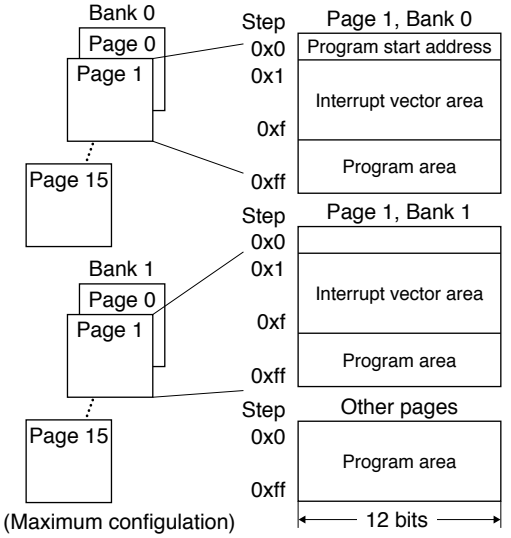
E0C62 Family Assembler Package

Quick Reference for Development

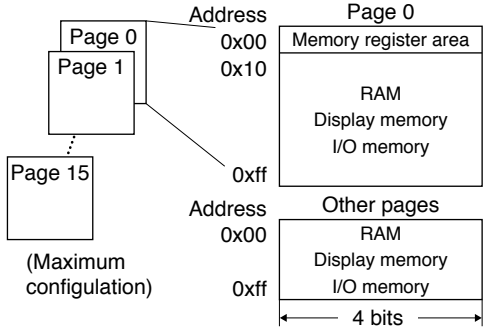
Memory Map

E0C6200 Core CPU

Program Memory Map



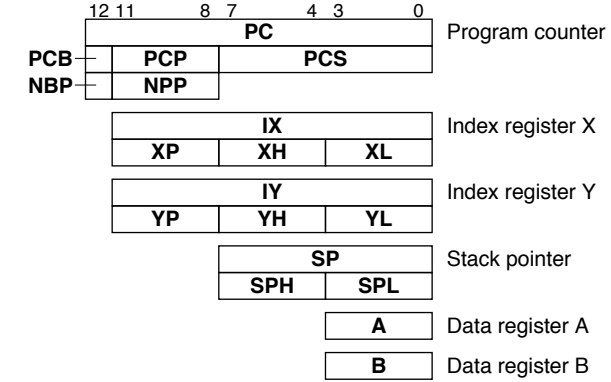
Data Memory Map



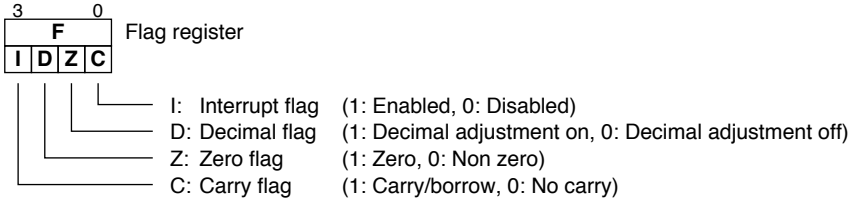
Registers

E0C6200 Core CPU

Registers



Flags



Symbols in the Instruction List

Registers/Register Data

A: Data register A or its contents (4 bits)
B: Data register B or its contents (4 bits)
X: Register XHL or its contents (8 low-order bits of the IX register)
XP: Register XP or its contents (4 high-order bits of the IX register)
XH: Register XH or its contents (4 high-order bits of the XHL register)
XL: Register XL or its contents (4 low-order bits of the XHL register)
Y: Register YHL or its contents (8 low-order bits of the IY register)
YP: Register YP or its contents (4 high-order bits of the IY register)
YH: Register YH or its contents (4 high-order bits of the YHL register)
YL: Register YL or its contents (4 low-order bits of the YHL register)
F: Flag register F or its contents (4 bits)
SP: Stack pointer SP or its contents (8 bits)
SPH: Stack pointer SPH or its contents (4 high-order bits of the stack pointer)
SPL: Stack pointer SPL or its contents (4 low-order bits of the stack pointer)
NBP: New bank pointer NBP or its contents (1 bit)
NPP: New page pointer NPP or its contents (4 bits)
PCB: Program counter bank PCB or its contents (1 bit)
PCP: Program counter page PCP or its contents (4 bits)
PCS: Program counter step PCS or its contents (8 bits)
PCSH: 4 high-order bits of PCS (4 bits)
PCSL: 4 low-order bits of PCS (4 bits)

Memory/Addresses/Memory Data

MX, M(X): Data memory addressed by IX or the contents of the specified memory
MY, M(Y): Data memory addressed by IY or the contents of the specified memory
Mn, M(n): Data memory addressed by n (n = 0 to 0xf) or the contents of the specified memory
M(SP): Stack addressed by SP or the contents of the stack address

Immediate Data

p: 5-bit immediate data or a label (0x0–0x1f)
s: 8-bit immediate data or a label (0x0–0xff)
l, x, y: 8-bit immediate data (0x0–0xff)
i: 4-bit immediate data (0x0–0xf)
n: 4-bit address for specifying Mn (0x0–0xf)
r, q: 2-bit immediate data for specifying a register or a data memory

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

Functions

←: Indicates that the right item is loaded or set to the left item.
+: Addition
-: Subtraction
&: AND
I: OR
^: XOR
!: NOT

Flags

Z: Zero flag
C: Carry flag
I: Interrupt flag
D: Decimal flag
–: Not changed
↔: Set (1), reset (0) or not changed
1: Set (1)
0: Reset (0)
★: Indicates that the instruction performs a decimal operation if the D flag is set.

Clk

Indicates the number of execution cycles.

Clasiffication	Mnemonic		Code												Flags				Clk	Function
	Opcode	Operand	MSB											LSB	I	D	Z	C		
Branch instructions	PSET	p	1	1	1	0	0	1	0			p			–	–	–	–	5	NPB←p[4], NPP←p[3:0]
	JP	s	0	0	0	0						s			–	–	–	–	5	PCB←NBP, PCP←NPP, PCS←s
		C, s	0	0	1	0						s			–	–	–	–	5	PCB←NBP, PCP←NPP, PCS←s, if C=1
		NC, s	0	0	1	1						s			–	–	–	–	5	PCB←NBP, PCP←NPP, PCS←s, if C=0
		Z, s	0	1	1	0						s			–	–	–	–	5	PCB←NBP, PCP←NPP, PCS←s, if Z=1
		NZ, s	0	1	1	1						s			–	–	–	–	5	PCB←NBP, PCP←NPP, PCS←s, 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						s			–	–	–	–	7	M(SP-1)←PCP, M(SP-2)←PCSH, M(SP-3)←PCSL+1, SP←SP-3, PCP←NPP, PCS←s
	CALZ	s	0	1	0	1						s			–	–	–	–	7	M(SP-1)←PCP, M(SP-2)←PCSH, M(SP-3)←PCSL+1, SP←SP-3, PCP←0, PCS←s
	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	I	0	0	0	1						I			–	–	–	–	12	PCSL←M(SP), PCSH←M(SP+1), PCP←M(SP+2), SP←SP+3, M(X)←I[3:0], M(X+1)←I[7:4], X←X+2
System control instructions	NOP5		1	1	1	1	1	1	1	1	1	0	1	1	–	–	–	–	5	No operation (5 clock cycles)
	NOP7		1	1	1	1	1	1	1	1	1	1	1	1	–	–	–	–	7	No operation (7 clock cycles)
	HALT		1	1	1	1	1	1	1	1	1	0	0	0	–	–	–	–	5	Halt (stop CPU)
	SLP		1	1	1	1	1	1	1	1	1	0	0	1	–	–	–	–	5	Sleep (stop CPU and oscillation)
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						x			–	–	–	–	5	XH←x[7:4], XL←x[3:0]
		Y, y	1	0	0	0						y			–	–	–	–	5	YH←y[7:4], YL←y[3:0]
		XP, r	1	1	1	0	1	0	0	0	0	0		r	–	–	–	–	5	XP←r
		XH, r	1	1	1	0	1	0	0	0	0	1		r	–	–	–	–	5	XH←r
		XL, r	1	1	1	0	1	0	0	0	1	0		r	–	–	–	–	5	XL←r
		YP, r	1	1	1	0	1	0	0	1	0	0		r	–	–	–	–	5	YP←r
		YH, r	1	1	1	0	1	0	0	1	0	1		r	–	–	–	–	5	YH←r
		YL, r	1	1	1	0	1	0	0	1	1	0		r	–	–	–	–	5	YL←r
		r, XP	1	1	1	0	1	0	1	0	0	0		r	–	–	–	–	5	r←XP
		r, XH	1	1	1	0	1	0	1	0	0	1		r	–	–	–	–	5	r←XH
		r, XL	1	1	1	0	1	0	1	0	1	0		r	–	–	–	–	5	r←XL
		r, YP	1	1	1	0	1	0	1	1	0	0		r	–	–	–	–	5	r←YP
		r, YH	1	1	1	0	1	0	1	1	0	1		r	–	–	–	–	5	r←YH
		r, YL	1	1	1	0	1	0	1	1	1	0		r	–	–	–	–	5	r←YL
	ADC	XH, i	1	0	1	0	0	0	0	0			i		–	–	↔	↔	7	XH←XH+i+C
		XL, i	1	0	1	0	0	0	0	1			i		–	–	↔	↔	7	XL←XL+i+C
		YH, i	1	0	1	0	0	0	1	0			i		–	–	↔	↔	7	YH←YH+i+C
		YL, i	1	0	1	0	0	0	1	1			i		–	–	↔	↔	7	YL←YL+i+C

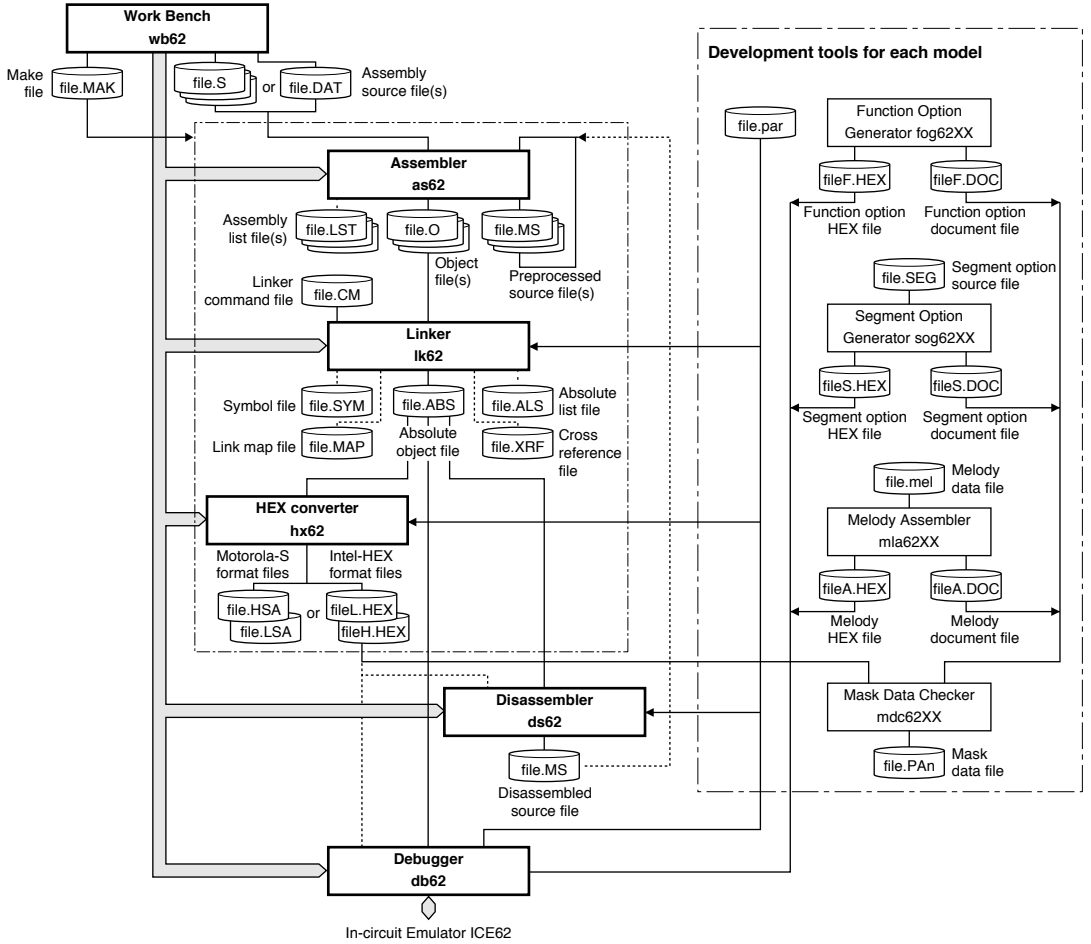
Remarks

Clasiffication	Mnemonic		Code												Flags				Clk	Function
	Opcode	Operand	MSB											LSB	I	D	Z	C		
Index operation instructions	CP	XH, i	1	0	1	0	0	1	0	0	i				–	–	↔	↔	7	XH-i
		XL, i	1	0	1	0	0	1	0	1	i				–	–	↔	↔	7	XL-i
		YH, i	1	0	1	0	0	1	1	0	i				–	–	↔	↔	7	YH-i
		YL, i	1	0	1	0	0	1	1	1	i				–	–	↔	↔	7	YL-i
Data transfer instructions	LD	r, i	1	1	1	0	0	0	r		i				–	–	–	–	5	r←i
		r, q	1	1	1	0	1	1	0	0	r		q		–	–	–	–	5	r←q
		A, Mn	1	1	1	1	1	0	1	0	n				–	–	–	–	5	A←M(n)
		B, Mn	1	1	1	1	1	0	1	1	n				–	–	–	–	5	B←M(n)
		Mn, A	1	1	1	1	1	0	0	0	n				–	–	–	–	5	M(n)←A
		Mn, B	1	1	1	1	1	0	0	1	n				–	–	–	–	5	M(n)←B
	LDPX	MX, i	1	1	1	0	0	1	1	0	i				–	–	–	–	5	M(X)←i, X←X+1
		r, q	1	1	1	0	1	1	1	0	r		q		–	–	–	–	5	r←q, X←X+1
	LDPY	MY, i	1	1	1	0	0	1	1	1	i				–	–	–	–	5	M(Y)←i, Y←Y+1
		r, q	1	1	1	0	1	1	1	1	r		q		–	–	–	–	5	r←q, Y←Y+1
	LBPX	MX, I	1	0	0	1	I								–	–	–	–	5	M(X)←I[3:0], M(X+1)←I[7:4], X←X+2
Flag operation instructions	SET	F, i	1	1	1	1	0	1	0	0	i				1	1	1	1	7	F←F I i
	RST	F, i	1	1	1	1	0	1	0	1	i				0	0	0	0	7	F←F & i
	SCF		1	1	1	1	0	1	0	0	0	0	0	1	–	–	–	1	7	C←1
	RCF		1	1	1	1	0	1	0	1	1	1	1	0	–	–	–	0	7	C←0
	SZF		1	1	1	1	0	1	0	0	0	0	1	0	–	–	1	–	7	Z←1
	RZF		1	1	1	1	0	1	0	1	1	1	0	1	–	–	0	–	7	Z←0
	SDF		1	1	1	1	0	1	0	0	0	1	0	0	–	1	–	–	7	D←1 (Decimal adjustment ON)
	RDF		1	1	1	1	0	1	0	1	1	0	1	1	–	0	–	–	7	D←0 (Decimal adjustment OFF)
	EI		1	1	1	1	0	1	0	0	1	0	0	0	1	–	–	–	7	I←1 (Enable interrupt)
	DI		1	1	1	1	0	1	0	1	0	1	1	1	0	–	–	–	7	I←0 (Disable 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	r		–	–	–	–	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	r		–	–	–	–	5	r←M(SP), SP←SP+1

Remarks

Clasiffication	Mnemonic		Code												Flags				Clk	Function
	Opcode	Operand	MSB											LSB	I	D	Z	C		
Stack operation instructions	POP	XP	1	1	1	1	1	0	1	0	1	0	0		–	–	–	–	5	XP←M(SP), SP←SP+1
		XH	1	1	1	1	1	0	1	0	1	0	1		–	–	–	–	5	XH←M(SP), SP←SP+1
		XL	1	1	1	1	1	0	1	0	1	1	0		–	–	–	–	5	XL←M(SP), SP←SP+1
		YP	1	1	1	1	1	0	1	0	1	1	1		–	–	–	–	5	YP←M(SP), SP←SP+1
		YH	1	1	1	1	1	0	1	1	0	0	0		–	–	–	–	5	YH←M(SP), SP←SP+1
		YL	1	1	1	1	1	0	1	1	0	0	1		–	–	–	–	5	YL←M(SP), SP←SP+1
		F	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	0	0	0	r			–	–	–	–	5	SPH←r
		SPL, r	1	1	1	1	1	1	1	0	0	r			–	–	–	–	5	SPL←r
		r, SPH	1	1	1	1	1	1	0	0	1	r			–	–	–	–	5	r←SPH
		r, SPL	1	1	1	1	1	1	1	0	1	r			–	–	–	–	5	r←SPL
Arithmetic operation instructions	ADD	r, i	1	1	0	0	0	0	r			i			–	★	↔	↔	7	r←r+i
		r, q	1	0	1	0	1	0	0	0	r		q		–	★	↔	↔	7	r←r+q
	ADC	r, i	1	1	0	0	0	1	r			i			–	★	↔	↔	7	r←r+i+C
		r, q	1	0	1	0	1	0	0	1	r		q		–	★	↔	↔	7	r←r+q+C
	SUB	r, q	1	0	1	0	1	0	1	0	r		q		–	★	↔	↔	7	r←r-q
	SBC	r, i	1	1	0	1	0	1	r			i			–	★	↔	↔	7	r←r-i-C
		r, q	1	0	1	0	1	0	1	1	r		q		–	★	↔	↔	7	r←r-q-C
	AND	r, i	1	1	0	0	1	0	r			i			–	–	↔	–	7	r←r & i
		r, q	1	0	1	0	1	1	0	0	r		q		–	–	↔	–	7	r←r & q
	OR	r, i	1	1	0	0	1	1	r			i			–	–	↔	–	7	r←r l i
		r, q	1	0	1	0	1	1	0	1	r		q		–	–	↔	–	7	r←r l q
	XOR	r, i	1	1	0	1	0	0	r			i			–	–	↔	–	7	r←r ^ i
		r, q	1	0	1	0	1	1	1	0	r		q		–	–	↔	–	7	r←r ^ q
	CP	r, i	1	1	0	1	1	1	r			i			–	–	↔	↔	7	r-i
		r, q	1	1	1	1	0	0	0	0	r		q		–	–	↔	↔	7	r-q
	FAN	r, i	1	1	0	1	1	0	r			i			–	–	↔	–	7	r & i
		r, q	1	1	1	1	0	0	0	1	r		q		–	–	↔	–	7	r & q
	RLC	r	1	0	1	0	1	1	1	1	r		r		–	–	↔	↔	7	d3←d2, d2←d1, d1←d0, d0←C, C←d3
	RRC	r	1	1	1	0	1	0	0	0	1	1	r		–	–	↔	↔	5	d3←C, d2←d3, d1←d2, d0←d1, C←d0
	INC	Mn	1	1	1	1	0	1	1	0	n				–	–	↔	↔	7	M(n)←M(n)+1
	DEC	Mn	1	1	1	1	0	1	1	1	n				–	–	↔	↔	7	M(n)←M(n)-1
	ACPX	MX, r	1	1	1	1	0	0	1	0	1	0	r		–	★	↔	↔	7	M(X)←M(X)+r+C, X←X+1
	ACPY	MY, r	1	1	1	1	0	0	1	0	1	1	r		–	★	↔	↔	7	M(Y)←M(Y)+r+C, Y←Y+1
	SCPX	MX, r	1	1	1	1	0	0	1	1	1	0	r		–	★	↔	↔	7	M(X)←M(X)-r-C, X←X-1
	SCPY	MY, r	1	1	1	1	0	0	1	1	1	1	r		–	★	↔	↔	7	M(Y)←M(Y)-r-C, Y←Y-1
	NOT	r	1	1	0	1	0	0	r		1	1	1	1	–	–	↔	–	7	r←!r

Remarks



1. Programming

Create assembly source files using the work bench or an editor.

2. Assembly and Linking

2-1) Start up the work bench.
2-2) Create a project file, then insert source files into the project.
2-3) Execute the build process.
The work bench executes the assembler and linker sequentially to generate an executable object file.

3. Option Data Creation *

Create the option HEX/document files (function option, segment option, melody data) using the tools provided for each model.

4. Debugging

4-1) Start up the debugger from the work bench.
4-2) Load the executable object file and option HEX files, then debug the program using the debug commands.

5. Mask Data Creation *

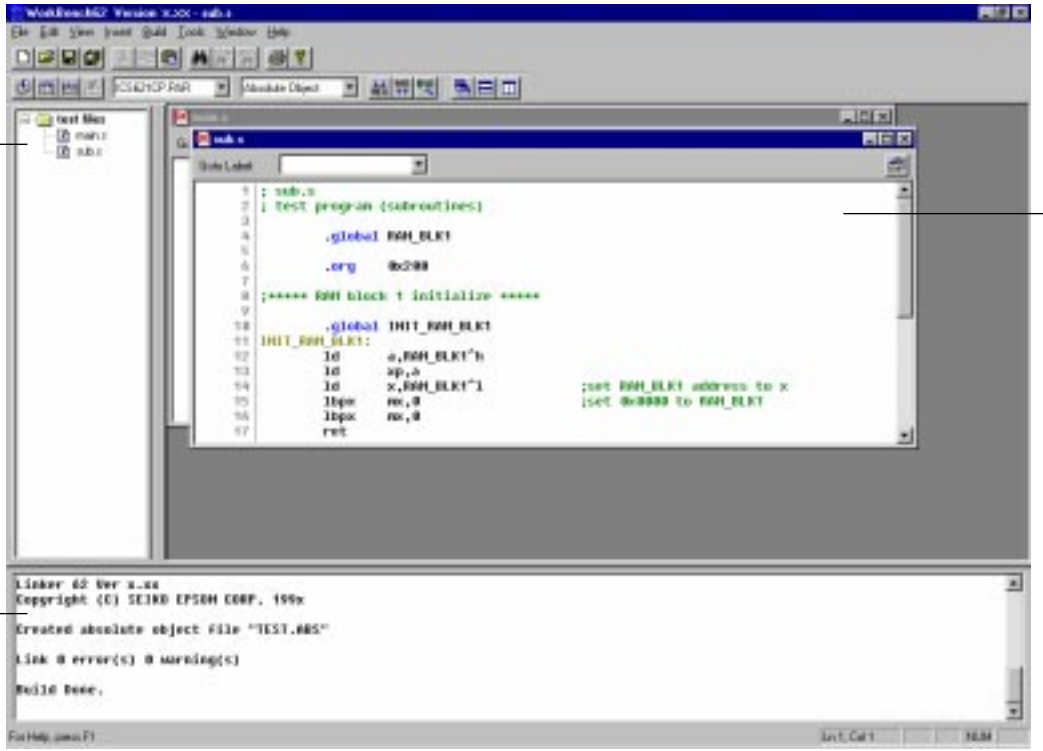
When the program development has been completed, create a mask data file.
5-1) Create the program HEX files using the HEX converter.
5-2) Convert the program and option document files into a mask data file using the mask data checker.
5-3) Submit the mask data file to Seiko Epson.

Note:
The part indicated as "Development tools for each model" and "*" (Steps 3 and 5) are not covered in the E0C62 Family Assembler Package, and the tools used for development depend on the model. For details, refer to the tool manual associated with each specific model.

Outline

The work bench provides an integrated development environment with Windows GUI. Creating/editing source files, selecting files and major startup options, and the startup of each tool can be made with simple Windows operations.

Windows



[Project] window

This window shows the currently opened work space folder and lists all the source files in the project, with a structure similar to Windows Explorer. Double-clicking a source file icon opens the source file in the [Edit] window.

[Edit] window

This window is used for editing a source file. A standard text file can also be displayed in this window. Two or more windows can be opened in the edit window area.

[Output] window

This window displays the messages delivered from the executed tools in a build or assemble process. Double-clicking a syntax error message with a source line number displayed in this window activates or opens the [Edit] window of the corresponding source so that the source line in which the error has occurred can be viewed.

Tool bars

[Standard] tool bar

[New] button
Creates a new document (source, header or project).

[Open] button
Opens a document (source, header or project).

[Save] button
Saves the document in the active [Edit] window to the file. The file will be overwritten.

[Save All] button
Saves the documents of all [Edit] windows and the project information to the respective files.

[Cut] button
Cuts the selected text in the [Edit] window to the clipboard.

[Copy] button
Copies the selected text in the [Edit] window to the clipboard.

[Paste] button
Pastes the text copied on the clipboard to the current cursor position in the [Edit] window.

[Find] button
Finds the specified word in the active [Edit] window.

[Find Next] button
Finds next target word towards the end of the file.

[Find Previous] button
Finds next target word towards the beginning of the file.

[Print] button
Prints the document in the active [Edit] window.

[Help] button
Displays the help window.

[Build] tool bar

[Assemble] button
Assembles the assembly source in the active [Edit] window.

[Build] button
Builds the currently opened project using a general make process.

[Rebuild All] button
Rebuilds the currently opened project.

[Stop Build] button
Stops the build process being executed.

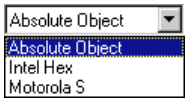
[Build] tool bar

[HEX Convert] button
Invokes the HEX converter.

[Disassemble] button
Invokes the disassembler.

[Debug] button
Invokes the debugger with the specified ICE parameter file.

[ICE Parameter] pull-down list box
Selects the ICE parameter file for the model being developed. In this box, all the ICE parameter files that exist in the "Dev62" directory are listed.



[Output Format] pull-down list box
Selects an executable object file format.
The build process will generate an executable object in the format selected here.

[Window] tool bar

[Cascade] button
Cascades the opened [Edit] windows.

[Tile Horizontally] button
Tiles the opened [Edit] window horizontally.

[Tile Vertically] button
Tiles the opened [Edit] window vertically.

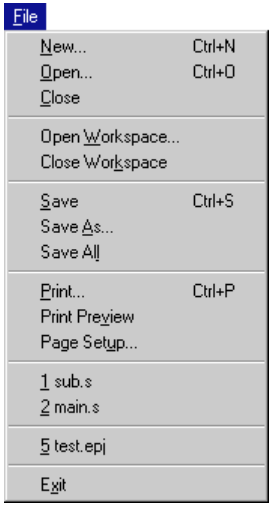
Controls on [Edit] window

[Insert Into project] button
Inserts the source file being edited into the current opened project.

[Goto Label] pull-down list box
Goes to the selected label position.

Menus

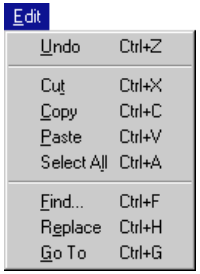
[File] menu



The file names listed in this menu are recently used source and project files.
Selecting one opens the file.

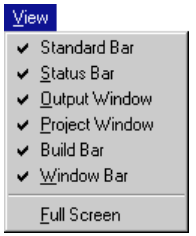
New... ([Ctrl]+[N])
Creates a new document (source, header or project).
Open... ([Ctrl]+[O])
Opens a document (source, header or project).
Close
Closes the active [Edit] window.
Open Workspace...
Opens a project.
Close Workspace
Closes the currently opened project.
Save ([Ctrl]+[S])
Saves the document in the active [Edit] window to the file.
Save As...
Saves the document in the active [Edit] window with another file name.
Save All
Saves the documents of all [Edit] windows and the project information to the respective files.
Print... ([Ctrl]+[P])
Prints the document in the active [Edit] window.
Print Preview
Displays a print image of the document in the active [Edit] window.
Page Setup...
Displays a dialog box for selecting paper and printer.
Exit
Terminates the work bench.

[Edit] menu



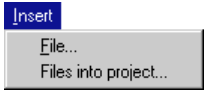
Undo ([Ctrl]+[Z])
Undoes the previous executed operation in the [Edit] window.
Cut ([Ctrl]+[X])
Cuts the selected text in the [Edit] window to the clipboard.
Copy ([Ctrl]+[C])
Copies the selected text in the [Edit] window to the clipboard.
Paste ([Ctrl]+[V])
Pastes the text copied to the the [Edit] window.
Select All ([Ctrl]+[A])
Selects all text in the active [Edit] window.
Find... ([Ctrl]+[F])
Finds the specified word in the active [Edit] window.
Replace ([Ctrl]+[H])
Replaces the specified words in the active [Edit] window.
Go To ([Ctrl]+[G])
Jumps to the specified line or label in the active [Edit] window.

[View] menu



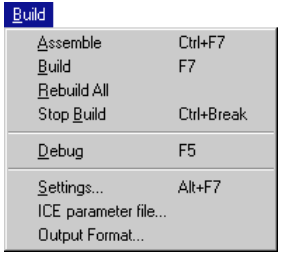
Standard Bar
Shows or hides the standard toolbar.
Status Bar
Shows or hides the status bar.
Output Window
Opens or closes the [Output] window.
Project Window
Opens or closes the [Project] window.
Build Bar
Shows or hides the build toolbar.
Window Bar
Shows or hides the window toolbar.
Full Screen
Maximizes the [Edit] window area to the full screen size.

[Insert] menu



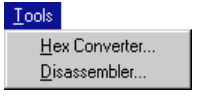
File...
Inserts the specified file to the text in the [Edit] window.
Files into project...
Adds the specified source file in the currently opened project.

[Build] menu



Assemble ([Ctrl]+[F7])
Assembles the assembly source in the active [Edit] window.
Build ([F7])
Builds the currently opened project using a general make process.
Rebuild All
Rebuilds the currently opened project.
Stop Build ([Ctrl]+[Break])
Stops the build process being executed.
Debug ([F5])
Invokes the debugger with the specified ICE parameter file.
Settings... ([Alt]+[F7])
Displays a dialog box for selecting tool options.
ICE parameter file...
Displays a dialog box for selecting an ICE parameter file.
Output Format...
Displays a dialog box for selecting an executable object file format.

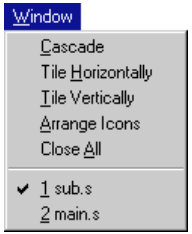
[Tools] menu



HEX Converter...
Invokes the HEX converter.
Disassembler...
Invokes the disassembler.

Menus

[Window] menu



This menu appears when an [Edit] window is opened.

Cascade
Cascades the opened [Edit] windows.

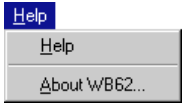
Tile Horizontally
Tiles the opened [Edit] window horizontally.

Tile Vertically
Tiles the opened [Edit] window vertically.

Arrange Icons
Arranges the minimized [Edit] window icons.

Close All
Closes all the [Edit] windows opened.

[Help] menu



Help
Displays the [Help] window.

About WB62...
Displays a dialog box showing the version of the work bench.

Error Messages

<filename> is changed by another editor. Reopen this file ?	The currently opened file is modified by another editor.
Cannot create file : <filename>	The file (linker command file, debugger command file, etc.) cannot be created.
Cannot find file : <filename>	The source file cannot be found.
Cannot find ICE parameter file	The ICE parameter file cannot be found.
Cannot open file : <filename>	The source file cannot be opened.
You cannot close workspace while a build is in progress. Select the Stop Build command before closing.	The project close command or work bench terminate command is specified while the build task is being processed.
Would you like to build it ?	The debugger invoke command is specified when the build task has not already been completed.

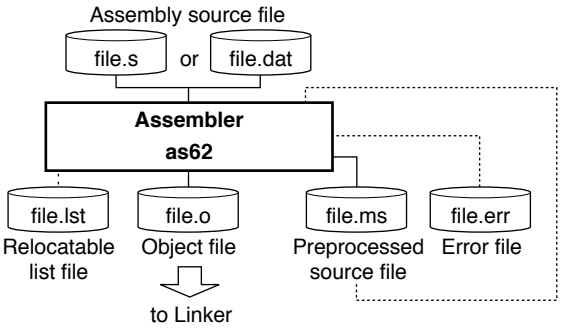
Short-Cut Key List

Ctrl + N	Creates a new document
Ctrl + O	Opens an existing document
Ctrl + F12	Opens an existing document
Ctrl + S	Saves the document
Ctrl + P	Print the active document
Ctrl + Shift + F12	Print the active document
Ctrl + Z	Undoes the last action
Alt + BackSpace	Undoes the last action
Ctrl + X	Cuts the selection and puts it on the clipboard
Shift + Delete	Cuts the selection and puts it on the clipboard
Ctrl + C	Copies the selection to the clipboard
Ctrl + Insert	Copies the selection to the clipboard
Ctrl + V	Inserts the clipboard contents at the insertion point
Shift + Insert	Inserts the clipboard contents at the insertion point
Ctrl + A	Selects the entire document
Ctrl + F	Finds the specified text
F3	Finds next
Shift + F3	Finds previous
Ctrl + H	Replaces the specified text with different text
Ctrl + G	Moves to the specified location
Ctrl + F7	Assembles the file
F7	Builds the project
Ctrl + Break	Stops the build
F5	Debugs the project
Alt + F7	Edits the project build and debug settings
Ctrl + Tab	Next MDI Window
Short-cut-key	Opens the popup menu
Shift + F10	Opens the popup menu

Outline

Converts the mnemonic of the source files into object codes (machine language) of the E0C62. The results are output in a relocatable object file. This assembler includes preprocessing functions such as macro definition/call, conditional assembly, and file-include functions.

Flowchart



Start-up Command Usage

```
Usage: as62 [options] <file name>
Options: -d <symbol>    Add preprocess definition
         -e             Output error log file (.ERR)
         -g             Add source debug information in object
         -l             Output relocatable list file (.LST)
         -o <file name> Specify output file name (.O or no extension)
File name: Source file name (.DAT, .S, or .MS)
```

Pseudo-instructions

#include	<file name>	Inserts other file in the source file.
#define	<define name> [<string>]	Defines a character string with a define name.
#macro	<macro name> [par] [,par] ...	Defines a statement string with a macro name.
	<statements>	Branch labels in a macro are specified with $$1 to $$n.
#endm		(par: Dummy parameters)
#ifdef	<name>	Conditional assembling
	<statements 1>	<name> defined: <statements 1> is assembled.
[#else		<name> undefined: <statements 2> is assembled.
	<statements 2>]	
#endif		
#ifndef	<name>	Conditional assembling
	<statements 1>	<name> undefined: <statements 1> is assembled.
[#else		<name> defined: <statements 2> is assembled.
	<statements 2>]	
#endif		
.code		Declares the start of a code section.
.bss		Declares the start of a bss section.
.org	<address>	Specifies an absolute address.
.page	<page number>	Specifies a page number.
.bank	<bank number>	Specifies a bank number.
.align	<alignment number>	Specifies alignment of a section.
.comm	<global symbol> <size>	Defines a global symbol and secures memory area in a bss section.
.lcomm	<local symbol> <size>	Defines a local symbol and secures memory area in a bss section.
.set	<symbol> <address>	Defines an absolute address for a symbol.
.global	<symbol>	Declares the symbol as global.
.codeword	<data>[<data> ... <data>]	Defines codes in the CODE section.
.list		Turns output ON(.list)/OFF(.nolist) in the assembly list
.nolist		file. (Effective only when the -l option is specified)
.stabs	"<file name>", FileName	Outputs source information for debugging.
.stabn	0, FileEnd	(Effective only when the -g option is specified)
.stabn	<line number>, LineInfo	

Operators

		Priority
+	Plus sign	1
-	Minus sign	1
^H	Acquires 8 high-order bits	2
^L	Acquires 8 low-order bits	2
~	Negation	2
( )	Parenthesis	(=3,)=11
*	Multiplication	4
/	Division	4
%	Residue	4
<<	Shifting to left	4
>>	Shifting to right	4
+	Addition	5
-	Subtraction	5
==	Equal (relational operator)	6
!=	Not equal (relational operator)	6
<	Less than (relational operator)	6
<=	Less than or equal (relational operator)	6
>	Greater than (relational operator)	6
>=	Greater than or equal (relational operator)	6
&	Bit AND	7
|	Bit OR	8
^	Bit XOR	8
&&	AND (relational operator)	9
||	OR (relational operator)	10

Numbers and symbols can be used as terms in expressions.
The expression is calculated as a signed 16-bit data.
Do not put any space or TAB between operator and number.

Error Messages

Cannot open <file kind> file <FILE NAME>	The specified file cannot be opened.
Cannot read <file kind> file <FILE NAME>	The specified file cannot be read.
Cannot write <file kind> file <FILE NAME>	Data cannot be written to the file.
Directory path length limit <directory path length limit> exceeded	The path name length has exceeded the limit.
File name length limit <file name length limit> exceeded	The file name length has exceeded the limit.
Division by zero	The divisor in the expression is 0.
Illegal macro label <label>	The internal branch label in macro definition is abnormal.
Illegal macro parameter <parameter>	The macro parameters are illegal.
Illegal syntax	The statement has a syntax error.
Line length limit <line length limit> exceeded	The number of characters in one line has exceeded the limit.
Macro parameter range <macro parameter range> exceeded	The number of macro parameters has exceeded the limit.
Memory mapping conflict	The address is duplicated.
Multiple statements on the same line	Two or more statements were described in one line.
Nesting level limit <nesting level limit> exceeded	Nesting of #include has exceeded the limit.
Number of macro labels limit <number of macro label limit> exceeded	The number of internal branch labels has exceeded the limit.
Out of memory	Cannot secure memory space.
Second definition of label <label>	The label is multiply defined.
Second definition of symbol <symbol>	The symbol is multiply defined.
Symbol name length limit <symbol name length limit> exceeded	The symbol name length has exceeded the limit.
Token length limit <token length limit> exceeded	The token length has exceeded the limit.
Unexpected character <name>	An invalid character has been used.
Unknown label <label>	Reference was made to an undefined label.
Unknown mnemonic <name>	A non-existing instruction was described.
Unknown register <name>	A non-existing register name was described.
Unknown symbol mask <name>	The symbol mask has a description error.
Unsupported directive <directive>	A non-existing pseudo-instruction was described.

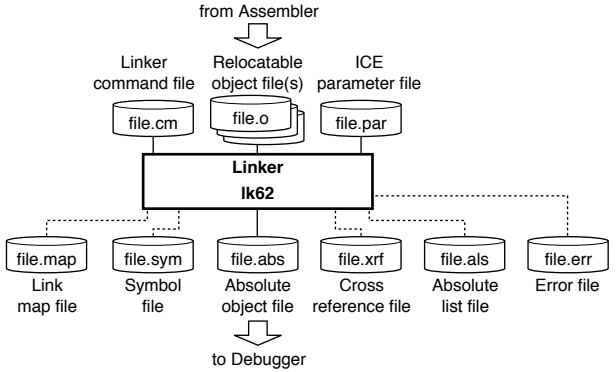
Warning Message

Second definition of define symbol <symbol>	The symbol is multiply defined by #define.
Section activation expected, use <.code/.bss>	There is no section definition.
Expression out of range	The result of the expression is out of the effective range.

Outline

Links the relocatable objects created by the assembler by fixing the memory locations, and creates executable absolute object codes. The linker also provides an auto PSET insertion/correction function allowing the programmer to create sources without having to know branch destination page numbers.

Flowchart



Start-up Command Usage

```
Usage: lk62 [options] <file names>
Options: -d                          Disable full PSET optimization
         -dr                         Disable PSET removal optimization
         -e                          Output error log file (.ERR)
         -g                          Add source debug information
         -l                          Output absolute list file (.ALS)
         -m                          Output map file (.MAP)
         -o <file name>              Output filename (.ABS or no extension)
         -or                         Optimize relocatable section location
         -s                          Output symbol file (.SYM)
         -x                          Output cross reference file (.XRF)
         -code <address>             Specify CODE start address
         -bss <address>              Specify BSS start address
         -rcode <file name>=<address> Specify CODE start address by file
         -rbss <file name>=<address>  Specify BSS  start address by file
         -defsym <symbol>=<address>   Define symbol address
File names: Relocatable obeject file names (.O)
            Command parameter file (.CM)
            ICE parameter file (.PAR)
```

Error Messages

Calling different bank at <address>	The call instruction calls a subroutine in another bank.
Cannot create <file kind> file <FILE NAME>	The file cannot be created.
Cannot open <file kind> file <FILE NAME>	The file cannot be opened.
Cannot read <file kind> file <FILE NAME>	The file cannot be read.
Cannot write <file kind> file <FILE NAME>	Data cannot be written to the file.
Illegal file name <FILE NAME>	The file name is incorrect.
Illegal file name <FILE NAME> specified with option <option>	The file name specified with the option is incorrect.
Illegal ICE parameter at line <line number> of <FILE NAME>	The ICE parameter file contains an illegal parameter setting.
Illegal object format <FILE NAME>	The input file is not an object file in IEEE-695 format.
Illegal option <option>	An illegal option is specified.
Memory mapping conflict at <Section type> section <address> - <address>	The address range of the section is duplicated.
No address specified with option <option>	Address is not specified with the option.
No code to locate	There is no valid code for mapping.
No debug information in <FILE NAME>	Debugging information is not included in the file.
No ICE parameter file specified	ICE parameter file is not specified.
No name and address specified with option <option>	Name and address are not specified with the option.
No object file specified	Object files to be linked are not specified.
Out of memory	Cannot secure memory space.
Page overflow at <Section type> section <address> - <address>	The section is across the page boundary.
Processor characteristics of object file <FILE NAME> mismatch	The object file is not matched to the specification in the ICE parameter file.
Second definition of Label <label> in <FILE NAME>	The label has already been defined.
Unavailable instruction code <instruction code> detected in <FILE NAME>	The object contains an instruction invalid for the model.
Unavailable memory mapped at <Section type> <address> - <address>	There is no valid memory space for allocating the section.
Unresolved external <label> in <FILE NAME>	Reference was made to an undefined symbol.

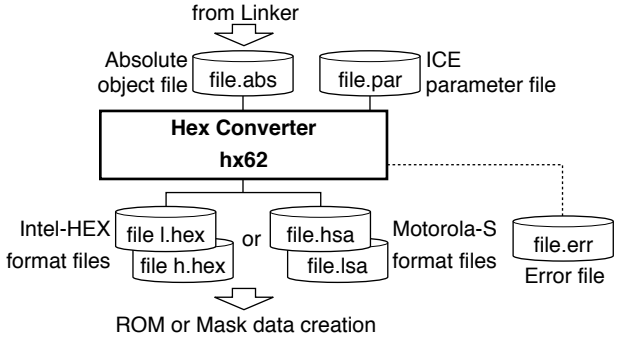
Warning Messages

Cannot open <file kind> file <FILE NAME>	The file cannot be opened.
No symbols found	Symbols cannot be found.

Outline

Converts an absolute object in IEEE-695 format output from the linker into ROM-image data in Intel-HEX format or Motorola-S format. This conversion is needed when making the ROM or when creating mask data using the development tools provided with each model.

Flowchart



Start-up Command Usage

```
Usage: hx62 [options] <file names>
Options: -b             Do not fill room with 0xff
         -e             Output error log file (HX62.ERR)
         -m             Use Motorola-S format
         -O <file name> Output file name (L/H.HEX, .L/HSA or no extension)
File name: Absolute object file (.ABS)
           ICE parameter file (.PAR)
```

Error Messages

Cannot create <file kind> file <FILE NAME>	The file cannot be created.
Cannot open <file kind> file <FILE NAME>	The file cannot be opened.
Cannot read <file kind> file <FILE NAME>	The file cannot be read.
Cannot write <file kind> file <FILE NAME>	Data cannot be written to the file.
Illegal file name <FILE NAME> specified with option <option>	The specified hex file name is incorrect.
Illegal ICE parameter at line <line number> of <FILE NAME>	The ICE parameter file contains an illegal parameter setting.
Illegal file name <FILE NAME>	The specified input file name is incorrect.
Illegal option <option>	An illegal option is specified.
Illegal absolute object format	The input file is not an object file in IEEE-695 format.
No ICE parameter file specified	ICE parameter file is not specified.
Out of memory	Cannot secure memory space.

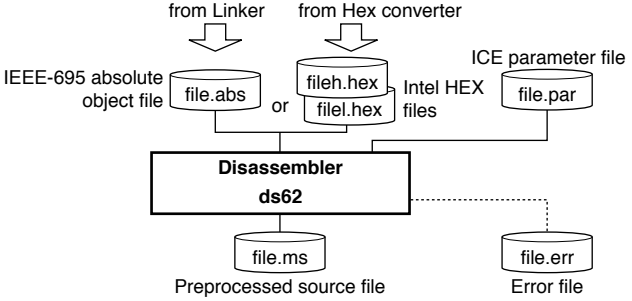
Warning Message

Input file name extension .XXX conflict	Two or more file names with the same extension have been specified. The last one is used.

Outline

Disassembles an absolute object file in IEEE-695 format or a hex file in Intel-HEX format, and restores it to a source format file. The restored source file can be processed in the assembler/linker/hex converter to obtain the same object or hex file.

Flowchart



Start-up Command Usage

```
Usage: ds62 [options] <file names>
Options: -cl            Use lower case characters
         -cu            Use upper case characters
         -e             Output error log file (DS62.ERR)
         -o <file name> Output file name (.MS or no extension)
         -s <address>   Offset address (Default 0x0)
File names: Absolute object file (.ABS or L/H.HEX)
            ICE parameter file (.PAR)
```

Error Messages

Cannot create <file kind> file <FILE NAME>	The file cannot be created.
Cannot open <file kind> file <FILE NAME>	The file cannot be opened.
Cannot read <file kind> file <FILE NAME>	The file cannot be read.
Cannot write <file kind> file <FILE NAME>	Data cannot be written to the file.
HEX data size does not match ICE parameter	The size of the input HEX file does not match the ICE parameter.
Illegal file name <FILE NAME> specified with option <option>	The specified output source file name is incorrect.
Illegal ICE parameter at line <line number> of <FILE NAME>	The ICE parameter file contains an illegal parameter setting.
Illegal file name <FILE NAME>	The specified input file name is incorrect.
Illegal HEX data format	The input file is not an Intel-HEX format file.
Illegal offset address <offset address>	The specified address is invalid.
Illegal option <option>	An illegal option is specified.
No ICE parameter file specified	ICE parameter file is not specified.
Out of memory	Cannot secure memory space.

Warning Message

Input file name extension .XXX conflict	Two or more file names with the same extension have been specified. The last one is used.

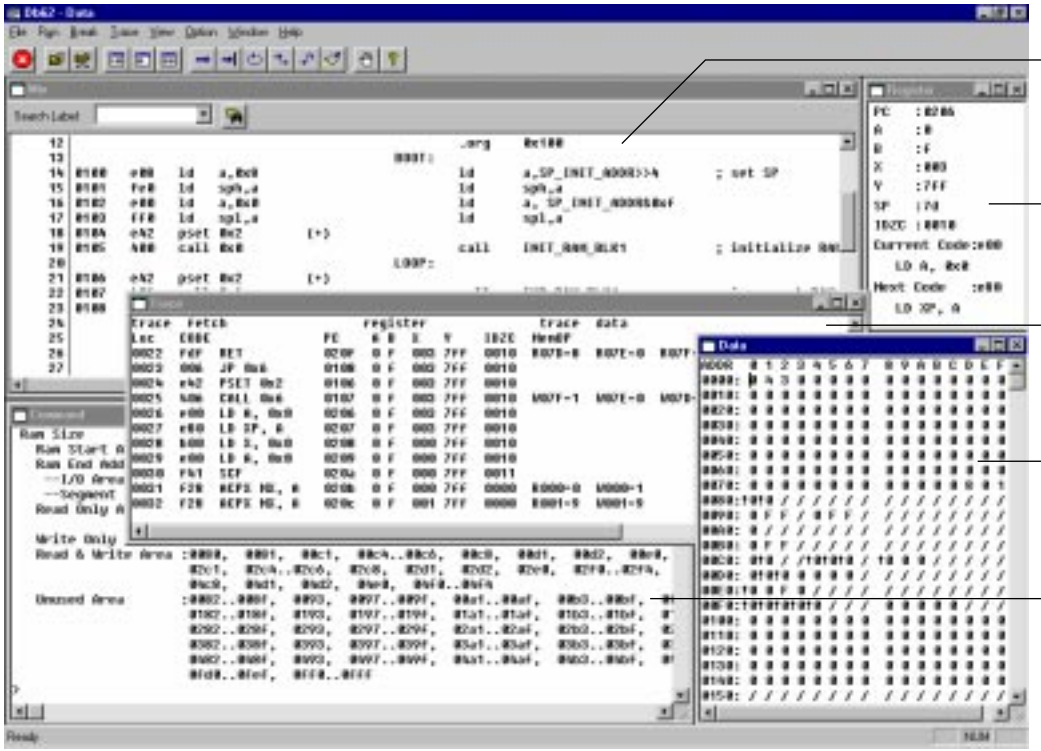
Outline

This software performs debugging by controlling the ICE62 hardware tool. Commands that are used frequently, such as break and step, are registered on the tool bar, minimizing the necessary keyboard operations. Moreover, sources, registers, and command execution results can be displayed in multiple windows, with resultant increased efficiency in the debugging tasks.

Start-up Command Usage

```
-Usage-
   db62^<parameter file name>^[startup option]
Options:
   command file:  ... specifies a command file
   -comX(X:1-4)   ... com port, default com1
   -b             ... baud rate, 1200, 4800, 9600, 19200(default)
```

Windows



[Source] window

Displays programs with unassemble codes, source codes or disassemble and source codes.

[Register] window

Displays register values and fetched codes.

[Trace] window

Displays traced data.

[Data] window

Displays the contents of the data memory.

[Command] window

Used to enter debug commands and display the execution results.

Buttons

Tool bar

[Key Break] button
Forcibly breaks execution of the target program.

[Load File] button
Reads an object file in the IEEE-695 format into the debugger.

[Load Option] button
Reads a program or optional HEX file in Intel-HEX format into the debugger.

[Source] button
Switches the display of the [Source] window to the source mode.

[Unassemble] button
Switches the display of the [Source] window to the unassemble mode.

[Mix] button
Switches the display of the [Source] window to the mix mode.

[Go] button
Executes the target program from the address indicated by the current PC.

[Go to Cursor] button
Executes the target program from the address indicated by the current PC to the cursor position in the [Source] window (the address of that line).

[Go from Reset] button
Resets the CPU and then executes the target program from the program start address (0x100).

[Step] button
Executes one instruction step at the address indicated by the current PC.

[Next] button
Executes one instruction step at the address indicated by the current PC.
The call and calz instructions and their subroutines are executed as one step.

[Reset] button
Resets the CPU.

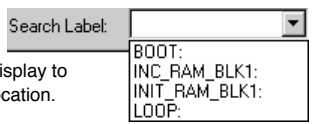
[Break] Button
Sets or clears a breakpoint at the address where the cursor is located in the [Source] window.

[Help] Button
Displays the help window.

Controls on [Source] window

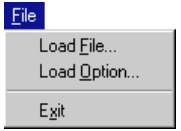
[Find] button
Searches the specified word and moves the source display to the found word location.

[Search Label] pull-down list box
Moves the source display to the selected label location.



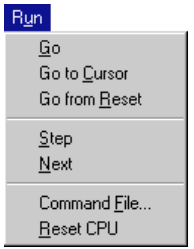
Menus

[File] menu



Load File...
Reads an object file in the IEEE-695 format into the debugger.
Load Option...
Reads a program or optional HEX file in Intel-HEX format into the debugger.
Exit
Terminates the debugger.

[Run] menu



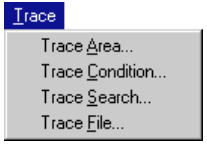
Go
Executes the target program from the address indicated by the current PC.
Go to Cursor
Executes the target program from the address indicated by the current PC to the cursor position in the [Source] window.
Go from Reset
This menu item resets the CPU and then executes the target program from the program start address (0x100).
Step
Executes one instruction step at the address indicated by the current PC.
Next
Executes one instruction step at the address indicated by the current PC.
The call and calz instructions and their subroutines are executed as one step.
Command File...
Reads a command file and executes the debug commands written in that file.
Reset CPU
Resets the CPU.

[Break] menu

Breakpoint Set...
Displays, sets or clears PC breakpoints.
Data Break...
Displays, sets or clears data break conditions.
Register Break...
Displays, sets or clears register break conditions.
Multiple Break...
Displays, sets or clears multiple break conditions.
Break All Clear
Clears all break conditions.

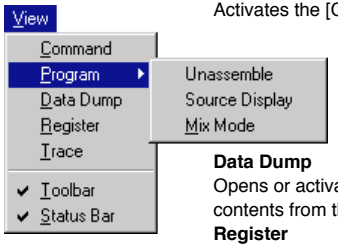
Menus

[Trace] menu



Trace Area...
Sets or clears program address ranges for tracing.
Trace Condition...
Sets a trace condition (Start, Middle, End).
Trace Search...
Searches trace information from the trace memory.
Trace File...
Saves the specified range of the trace information displayed in the [Trace] window to a file.

[View] menu

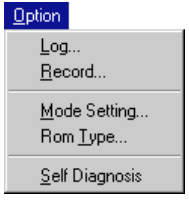


Command
Activates the [Command] window.

Program (Unassemble, Source Display, Mix Mode)
Opens or activates the [Source] window and displays the program from the current PC address in the display mode selected from the sub menu items.

Data Dump
Opens or activates the [Data] window and displays the data memory contents from the memory start address.
Register
Opens or activates the [Register] window and displays the current values of the registers.
Trace
Opens or activates the [Trace] window and displays the trace data sampled in the ICE trace memory.
Toolbar
Shows or hides the toolbar.
Status Bar
Shows or hides the status bar.

[Option] menu



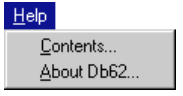
Log...
Starts or stops logging.
Record...
Starts or stops recording of commands executed.
Mode Setting...
Sets the on-the-fly display, break and execution counter modes.
Rom Type...
Specifies the program ROM type which is installed in the ICE ROM socket.
Self Diagnosis
Displays the results of the diagnostic test in the ICE62.

[Window] menu

Cascade
Cascades the opened windows.
Tile
Tiles the opened windows.

This menu shows the currently opened window names.
Selecting one activates the window.

[Help] menu



Contents...
Displays the contents of help topics.
About Db62...
Displays an About dialog box for the debugger.

Debug Commands

Program memory operation

as [<addr> <mnemonic>]	Assemble mnemonic
pe [<addr> <code1> [<code2> [..<code8>]]]	Input program code
pf [<addr1> <addr2> <code>]	Fill program area
pm [<addr1> <addr2> <addr3>]	Copy program memory

Data memory operation

dd [<addr1> [<addr2>]]	Dump data memory
de [<addr> <data1> [<data2> [..<data16>]]]	Input data
df [<addr1> <addr2> <data>]	Fill data area
dm [<addr1> <addr2> <addr3>]	Copy data area

Register operation

rd	Display register values
rs [<reg> <value> [<reg> <value>..]]	Modify register values

Program execution

g [<addr>]	Execute successively
gr	Reset CPU and execute successively
s [<step(D)>]	Step into
n [<step(D)>]	Step over

CPU reset

rst	Reset CPU

Break

bp [<addr1> [<addr2> [..<addr4>]]]	Set PC breakpoint
bpc [<addr1> [<addr2> [..<addr4>]]]	Clear PC breakpoint
bd [<addr> <data> {r|w|*}]	Set data break
bdc	Clear data break
br [<reg> <value> [<reg> <value>..]]	Set register break
brc	Clear register break
bm [{pc|addr|data|opt|a|b|f|x|y|} <value>..]	Set multiple break
bmc	Clear multiple break
bl	Display all break conditions
bac	Clear all break conditions
be	Set break enable mode
bsyn	Set break disable (synchronous) mode

Program display

u [<addr>]	Unassemble display
sc [<addr>]	Source display
m [<addr>]	Mix display

Symbol information

sy [{$<keyword>|#<keyword>}] [/a]	List symbols

Load file

lf [<file name>]	Load IEEE-695 format absolute object file
lo [<file name>]	Load Intel-HEX format file

ROM access

rp	Load program from ROM
vp	Verify the contents of ROM with program memory
rom [{64|128|256|512}]	Set ROM type

Trace

tc [{s|m|e}]	Set trace condition
ta [{all | <start1> <end1> [..<start4> <end4>]}]	Set trace area
tac [<start1> <end1> [..<start4> <end4>]]	Clear trace area
tp	Display current trace pointer
td [<num(D)>]	Display trace information
ts [{pc|dr|dw} <addr>]	Search trace information
tf [[<num1(D)> <num2(D)>] <file name>]	Save trace information into file

Others

cv [<addr1> [<addr2>]]	Display coverage information
cvc	Clear coverage information
com [<file name> [<interval(D)>]]	Load & execute command file
rec [<file name>]	Record commands to a command file
log [<file name>]	Turn log output on or off
ma	Display map information
otf	Turn on-the-fly display on or off
tim	Set time or step measurement mode
chk	Report results of ICE62 self diagnostic test
q	Quit debugger

The parameters with (D) should be specified with a decimal number. For other parameters, hexadecimal numbers can only be used.
A symbol can be used to specify an address as follows:
@<global symbol> or @<local symbol>@<source file name>

Debugger Messages

ICE errors

communication error	There is a problem in communication between Host and ICE.
ID not match	ICE protocol ID error
ROM sum check error	ROM sum error found during self diagnostic test.
RAM check error	RAM error found during self diagnostic test.
undefined code detected	Some undefined code is detected when loading file.

ICE status

break hit	A breakpoint is met when executing a program.
break switch pushed	Break switch is pressed.
halt	The status of ICE is halt.
key break	Key break is pressed.
reset switch target	Reset switch is pressed.
reset switch idle	Reset switch is idle.
target down	There is a problem in communication between the ICE and EVA board.
time out	The time waiting for a message from ICE is too long.

Command errors

No coverage address	There is no coverage information. (cv)
No trace data	There is no trace data in trace memory. (td, ts)
address beyond code range	The specified program memory address is out of range. (pe, pf, pm, sc, m, u, g, gr, bp, bm, ts, cv)
address beyond data range	The specified data memory address is out of range. (de, df, dm, bd, bm, ts)
can't open file	The file cannot be opened. (lf, lo)
data range (0 - 0xf)	The specified number is out of the data range. (de, df)
different chip type, can't load this file	A different ICE parameter is used in this file. (lf)
end address < start address	The start address is larger than the end address. (pf, pm, df, dm, bp, cv)
error file type (extension should be CMD)	The extension of the command file should be CMD. (com)
identifier (PC/ADDR/DATA/OPT/A/B/X/Y/F)	An illegal parameter has been specified for an item of the bm command. (bm)
illegal code	The input code is not available. (pe, pf)
illegal mnemonic	The input mnemonic is invalid for E0C62. (as)
invalid command	This is an invalid command. (All commands)
invalid data pattern	The input data pattern is invalid. (bd, br, bm)
invalid value	The input data, address or symbol is invalid. (All commands)
no high and low ROM	No ROM is installed in ICE. (rp)

Command errors

no high ROM	No high-order ROM is installed in ICE. (rp)
no low ROM	No low-order ROM is installed in ICE. (rp)
no mapping area	A no-map area is specified. (pm, dm)
no such symbol	There is no such symbol. (All symbol support commands)
number of parameter	The parameter number is incorrect. (All commands)
over max nesting level (5), can't open file	Nestling of the com command exceeds the limit. (com)
r/w option (r, w or *)	An illegal R/W option is specified. (bd, bm)
ROM program verify error	ROM program checks out different codes. (vp)
ROM type (64/128/256/512)	An illegal value is specified for the ROM type parameter of the rom command. (rom)
step range (0 - 65535)	The specified step count is out of range. (s, n)
symbol type error	The symbol type (CODE / BSS) is error. (All symbol support commands)
this chip not support this function	The chip with the used parameter file cannot support this option function. (lo)
undefined code detected	Undefined code is detected when loading file. (rp)
valid register name (PC/A/B/X/Y/F)	An invalid register name is specified. (br)

Command warning

read only address, can't write	This data address is read only, cannot be written to. (de)