RTE-V850E/MA1-CB

USER'S MANUAL (Rev. 1.05)

Midas lab

REVISION HISTORY

Date	Revision	Chapter	Explanation of revision
May 12, 1999	0.50		Preliminary version: Supported by experimental board
July 19, 1999	1.00		Official version: Supported by mass-produced model
August 13, 1999	1.01	11.3	Addition of resources of RTE-MB-A
January 17, 2000	1.02	7.2.1	Reviced RFS1 Value: 0x0024/0016 -> 0x8024/8016
February 3, 2000	1.03	7.2.1	Reviced RFS1 Value: 0x8024/8016 -> 0x8017/800F
Jun 11, 2001	1.04	7.4.9	Add Word accessing to G-Bus function
Oct 25, 2001	1.05	7.2.1	Reviced ASC comments: all 0 addr wait -> all 1 addr wait

CONTENTS

1. INTRODUCTION 1
1.1. NUMERIC NOTATION 1

2. FUNCTIONS 2

3. MAJOR FEATURES 3

4. BASIC SPECIFICATIONS 3

5. BOARD CONFIGURATION 4
5.1. RESET SWITCH (SW_RESET) 4
5.2. POWER CONNECTOR (JPOWER) 4
5.3. SWITCH 1 (SW1) 4
5.4. SWITCH 2 (SW2) 5
5.5. SWITCH 3 (SW3) 5
5.6. SWITCH 4 (SW4) 6
5.7. SWITCHES 5 TO 8 (SW5 TO SW8) 7
5.8. 7SEG-LED, xxx-LED 8
5.9. TEST PINS FOR ROM EMULATOR (JROM-EM1) 8
5.10. CLOCK SOCKET (OSC1) 8
5.11. CRYSTAL SOCKET (JP1) 9
5.12. AVDD/AVREF SELECTOR JUMPER (JP2) 9
5.13. ROM SOCKETS 9
5.14. SELF-WRITING POWER CONNECTOR (JVPP) 9
5.15. FLASH WRITING CONNECTOR (JFLASH) The JFLASH connector is used to write data to the CPU's internal flash ROM by using a flash programmer device. To use JFLASH, SW4 must be set (see Section 5.6). 10
5.16. SERIAL CONNECTOR (JSIO1, JSIO2) 11
5.17. JGBUS CONNECTOR (JGBUS) 11
5.18. CPU CONNECTOR (JCPU1, JCPU2) 12

6. CONNECTION WITH THE HOST PC 14
6.1. RS-232C CONNECTION 14

7. HARDWARE REFERENCES 15
7.1. MEMORY AND I/O MAP 15
7.2. RECOMMENDED SETTINGS 17
7.2.1. MEMC Registers 17
7.3. MEMORY RESOURCES 18
7.3.1. SDRAM (CS1: 0800000 to 3FFFFFF) 18
7.3.2. SRAM (CS2: 0400000 to 07FFFFF) 18
7.3.3. UV-EPROM (CS0: 0000000 to 03FFFFF) 18
7.4. I/O MAP 19
7.4.1. I/O List 19
7.4.2. SW1 Read Port (SW1 7800000H [Read Only]) 19
7.4.3. SW2 Read Port (SW2 7801000H [Read Only]) 19

7.4.4. 7-Segment LED Display Data Output Port (7SEG-LED 7802000H [Write Only])20
7.4.5. Time-Over Ready LED Clear Pulse (TOVRDY_LED_CLRPLS 7803000H [Write Only])....20
7.4.6. Interrupt Controller (PIC: 7804000H to 7804020H [Read/Write]) 21
7.4.7. UART (TL16C550C: 7807000H to 7807070H) 22
7.4.8. TIC (μPD71054 7808000H to 7808030FH) 22
7.4.9. GBUS ACCESS CONTROL (7809000H [Read/Write]) 23

8. SOFTWARE 24
8.1. INITIALIZATION....24
8.2. SUCCESSIVE ACCESSES TO μPD71054 24
8.3. LIBRARIES....24
8.4. EXAMPLE OF USING TIMERS 25

9. DEVELOPMENT OF APPLICATIONS USING MASKABLE INTERRUPTS 26
9.1. INTERRUPT VECTOR 26
9.2. GENERAL RESTRICTIONS/NOTES 28
9.3. REWRITING THE ALTERNATE VECTOR AREA DURING DOWNLOADING....28
9.4. RESTRICTIONS/NOTES ON BREAKPOINTS....29

10. CPU PIN CONNECTION....30
10.1. PIN CONNECTION LIST....30
10.2. RESET- 31
10.3. PCM0/WAIT0-....32
10.4. PCM5/SELFREF 32
10.5. P20/NMI, P1/INTP000 33
10.6. P2/INTP001, P11/INTP010, P12/INTP011, P21/INTP020....33
10.7. P40/SO0/TXD0 34
10.8. P41/SI0/RXD0....34
10.9. P42/SCK0-, P43/SO1/TXD1, P44/SI1/TXD0, P45/SCK1- 34
10.10. P4/INTP100/DMARQ0-..P7/INTP103/DMARQ3-....35
10.11. PBD0/DMAAK0-..PBD3/DMAAK3-....35
10.12. P24/INTP110/TC0-..P27/INTP113/TC3-....36
10.13. OTHER SIGNALS 36

11. SPECIFIC GBUS SPECIFICATIONS....37
11.1. GENERAL....37
11.2. BUS CYCLE 38
11.3. CHIP SELECT....39

12. APPENDIX A MULTI MONITOR 40
12.1. BOARD SETTING 40
12.1.1. RTE for Win 32 Installation....40
12.1.2. SW1 Setting....40
12.1.3. Connection of Board....40
12.2. Multi MONITOR....41
12.2.1. 7-Segment LED on Startup....41
12.2.2. ROM Monitor Work RAM....41
12.2.3. Monitor Interrupt 41
12.2.4. _INIT_SP Setting....41

12.2.5. Timer Interrupt 41
12.2.6. Initializing Hardware 41
12.2.7. Special Instruction 42
12.3. RTE COMMANDS 42
12.3.1. HELP(?) 42
12.3.2. INIT 42
12.3.3. VER 42
12.3.4. SFR Command 42

13. APPENDIX B PARTNER MONITOR 43
13.1. BOARD SETTING 43
13.1.1. SW1 Setting 43
13.1.2. Connection of Board 43
13.2. PARTNER MONITOR 44
13.2.1. 7-Segment LED on Startup 44
13.2.2. ROM Monitor Work RAM 44
13.2.3. Monitor Interrupt 44
13.2.4. SP Setting 44
13.2.5. Initializing Hardware 44
13.2.6. Special Instruction 44

14. APPENDIX C GBUS COMMON SPECIFICATIONS 45
14.1. TERMINOLOGY 45
14.1.1. CPU Board and Motherboard 45
14.1.2. Bus Cycle and Micro Cycle 45
14.2. SIGNALS 45
14.3. PIN ASSIGNMENTS 50
14.4. PROCESSING OF UNUSED PINS 51
14.5. ALLOCATING GCS-[7:0] 51
14.6. BUS CYCLE 52
14.6.1. Single Cycle 52
14.6.2. Burst Cycle 52
14.6.3. GWAITI- 53
14.6.4. GBTERM- 54
14.7. TIMING55
14.7.1. Setup Time 55
14.7.2. Delay Time 55

1. INTRODUCTION

The RTE-V850E/MA1-CB is a CPU board that is designed to evaluate the NEC V850E/MA1 RISC processor.

The board features a V850E/MA1 capable of operating at a maximum speed of 50 MHz, memory, serial interface, and bus connector for expansion. As the memories, a high-speed SRAM and high-capacity SDRAM are provided as standard. The SDRAM is controlled by using the internal memory controller of the V850E/MA1.

These functions enable the RTE-V850E/MA1-CB to be used for a wide variety of applications including processor performance evaluation and application program development at the initial stage, and to also be used as an engine for demonstration and simulation.

The GHS Multi or NEC PARTNER source-level debugger can be used as a development software tool with the RTE-V850E/MA1-CB. The type of monitor to be stored in ROM depends on the debugger type.

In ROM, the monitor specified at the time of purchase is stored. Even when neither of the debuggers is purchased together with the RTE-V850E/MA1-CB, they can be purchased at anytime subsequently.

1.1. NUMERIC NOTATION

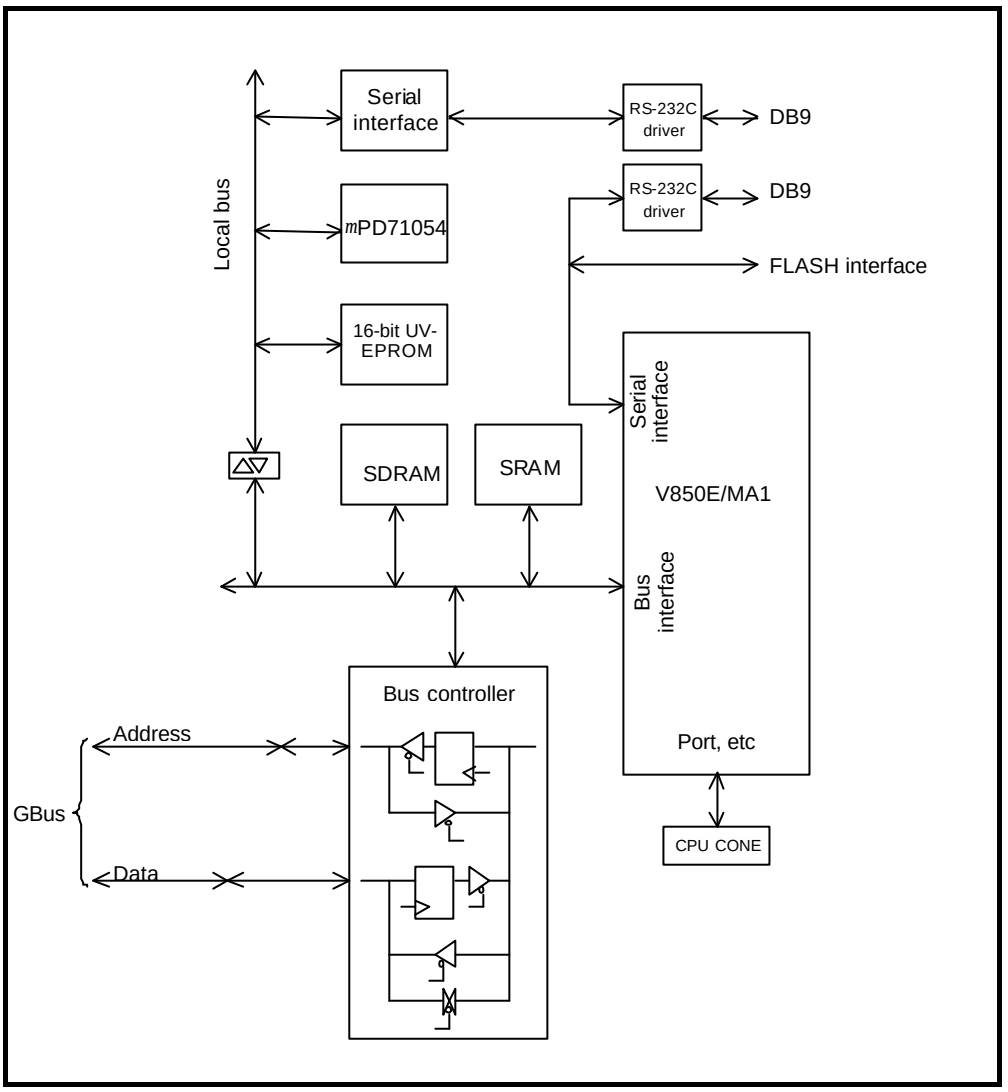
This manual represents numbers according to the notation described in the following table. Hexadecimal and binary numbers may be hyphenated at every four digits, if they are difficult to read because of many digits being in each number.

Number	Notation rule	Example
Decimal number	Only numerals are indicated.	"10" represents number 10 in decimal.
Hexadecimal number	A number is suffixed with letter H.	"10H" represents number 16 in decimal.
Binary number	A number is suffixed with letter B.	"10B" represents number 2 in decimal.

Number Notation Rules

2. FUNCTIONS

The overview of each function block of the RTE-V850E/MA1-CB is shown below.



RTE-V850E/MA1-CB Block Diagram

"Local bus" is a bus that buffers the CPU bus and is synchronized with the CPU. "Gbus" is independent of the CPU and is fixed at 33 MHz.

3. MAJOR FEATURES

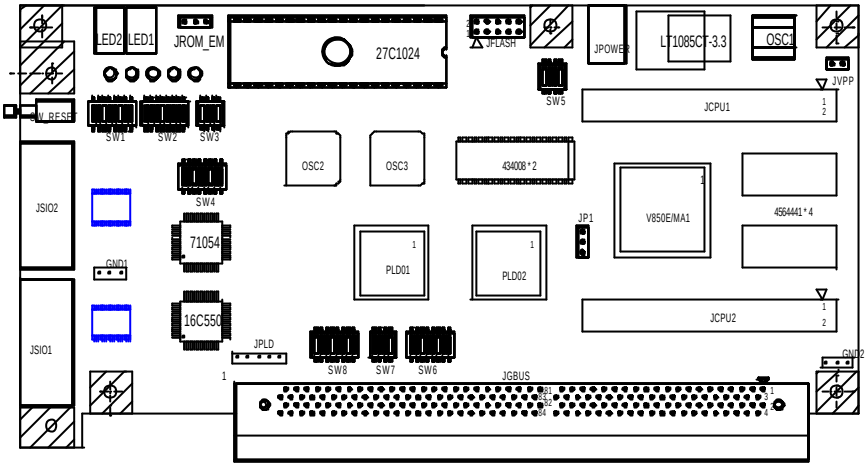
- Two types of monitor ROM are provided: one is used for the Green Hills Multi and the other for the NEC PARTNER.
- Real-time execution and evaluation at a high-level language level using Multi or PARTNER.
- A ROM emulator can be connected.
- 1M byte of high-speed SRAM and 32M bytes of SDRAM are provided as standard.
- Two serial interfaces are provided. (One channel uses an external controller and the other channel uses the internal controller of the CPU. The serial channel of an external controller is used as the monitor.)
- Three timer channels are provided. (One channel is used for the monitor.)

4. BASIC SPECIFICATIONS

Processor	V850E/MA1	
CPU clock	50 MHz	
Bus clock	50 MHz	
Power supply	+5 V, 2 A (max.)	
Memory		
EPROM	128 KB	64 K × 16 bits (40-pin DIP) × 1 (512K bytes max.)
SRAM	1 MB	512 K × 8 bits × 2
SDRAM	32 MB	4 M × 4 bits × 4 banks × 4
I/O		
Serial (2 ch)	Internal CPU (UART), DB9 connector Equivalent to NS16550, DB9 connector	
Timer	μPD71054, 500-ns resolution	
I/O port	LED (7-segment) display/switch input	
Others		
CPU connector	Connector with all function pins of the V850E/MA1 connected	
GBUS connector	RTE-CB standard 32-bit interface (4G bytes, 32-bit bus, correspond to DMA)	
FLASH interface	Interface for connecting FLASH Writer	
Reset switch	Push type	

5. BOARD CONFIGURATION

The physical layout of the major components on the RTE-V850E/MA1-CB board is shown below. This chapter explains each component.



RTE-V850E/MA1-CB Components Layout

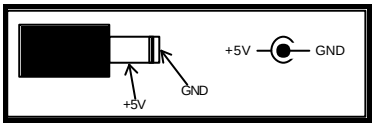
5.1. RESET SWITCH (SW_RESET)

SW_RESET is a reset switch for the entire board. Pressing this switch causes all the circuits including the CPU to be reset.

5.2. POWER CONNECTOR (JPOWER)

The power supplied to the JPOWER connector should be one rated as listed below.

Voltage:	+5 V
Current:	Maximum of 2 A
Mating connector:	Type A (5.5 mm in diameter)
Polarity:	



Note the polarity when attaching the power connector.
To supply power from the JGBUS connector, do not connect a power source to JPOWER.

5.3. SWITCH 1 (SW1)

SW1 is a general-purpose input port switch. The setting status can be read from an input port (see Section 7.4.2). When the port is read, a switch being set to OFF represents 1, while its being set to ON represents 0. When the monitor ROM is used, all SW1 switches except some are already set. Set this switch for assignment with the monitor ROM by referring to the following sections and in accordance with your environment:

When using Multi, see Section 12.1.2.

When using PARTNER, see Section 13.1.1.

5.4. SWITCH 2 (SW2)

SW2 selects an operation of the board by hardware. The setting of the switch can be read from an input port (see Section 7.4.3).

No.	Signal name	Factory setting	Function
1	FBOOT	OFF	Specifies resources to be allocated to the CS0 space. OFF: The on-board UV-EPROM is allocated to the CS0 space. ON: GCS1- space of GBUS is allocated to the CS0 space (see Section 7.1).
2	TEST	OFF	Set this signal to OFF.
3	BCLK_LOW	OFF	Selects frequency of oscillator mounted on OSC1. Depending on the value set, the monitor ROM changes the number of ROM and SRAM wait cycles. In addition, the number of I/O wait cycles is changed by hardware. OFF: Bus clock exceeds 33 MHz. ON: Bus clock is kept at 33 MHz or less.
4	Not used	OFF	
5	NMI/INT	OFF	Specifies interrupt to be used by the monitor. OFF: NMI0 ON: INTP000
6	Not used	OFF	Not used. Must be set to OFF.
7	Not used	OFF	Not used. Must be set to OFF.
8	Not used	OFF	Not used. Must be set to OFF.

SW2-5 is read only to set the interrupt controller by the monitor.

5.5. SWITCH 3 (SW3)

SW3 selects the type of ROM inserted in the ROM socket and performs setting related to banks.

No.	Signal name	Factory setting	Function
1	ROM_TYPE0	OFF	Selects the type of ROM. [ROM_TYPE1, ROM_TYPE0] [OFF , OFF]: When monitor ROM is used [OFF , ON]: When 27C4096 is used [ON , OFF]: When 27C2048 is used [ON , ON]: When 27C1024 is used
2	ROM_TYPE1	OFF	
3	BANK_DIS	OFF	Specifies whether the upper and lower halves (banks) of ROM are separated. Be sure to set this signal to OFF when monitor is used. OFF: Upper and lower halves of ROM are separated. ON: Upper and lower halves of ROM are used as a contiguous area.
4	BANK_LOW	OFF	Specifies whether either the upper or lower half of the ROM is valid when the ROM is used in bank mode. OFF: Selects lower half. ON: Selects upper half.

[Caution] To use the monitor ROM, do not change the factory setting.

5.6. SWITCH 4 (SW4)

SW4 specifies the mode of a CPU pin. When a signal of this switch is set to OFF, the corresponding CPU pin is 1; when it is set to ON, the pin is 0.

No.	CPU pin name	Factory setting	Function
1	MODE0	ON	Directly connected to the MODE0 pin of the CPU.
2	MODE1	ON	Directly connected to the MODE1 pin of the CPU.
3	CKSEL	ON	Directly connected to the CKSEL pin of the CPU.
4	FLASH-	OFF	ON when the programmer is connected to JFLASH; otherwise, OFF.

<<Cautions>>

1. Use MODE[1..0] under the following operating conditions.

MODE1	MODE0	Mode	Operating conditions
ON	ON	ROM-less mode 0 (16 bits)	Monitor can be used (factory-set condition).
ON	OFF	ROM-less mode 1 (8 bits)	Setting prohibited
OFF	ON	Single-chip mode 0 (0 address), or FLASH program mode	Monitor cannot be used.
OFF	OFF	Single-chip mode 1 (1M addresses), or FLASH program mode	Monitor can be used.

2. To set SW4-4 to ON, set SW4-2 to OFF (FLASH program mode).

5.7. SWITCHES 5 TO 8 (SW5 TO SW8)

SW5 to SW8 physically cuts the board's signal lines connected to CPU pins. All the switches are factory-set to ON (connected). Set a switch to OFF only when the corresponding signal line is used for an external source, but only if the internally used resources are not necessary.

Remark The following tables show the CPU pins and final internal resource names.

[SW5]

No.	CPU pin name	Factory setting	Internally used resource
1	P40/SO0/TXD0	ON	SIO2-TXD
2	P41/SI0/RXD0	ON	SIO2-RXD
3	P42/SCK0-	ON	SIO2-RTS-
4	P43/SO1/TXD1	ON	SIO2-CTS-
5	P44/SI0/RXD1	ON	SIO2-DSR-
6	P45/SCK1-	ON	SIO2-DTR-
7	Not used	OFF	
8	Not used	OFF	

[SW6]

No.	CPU pin name	Factory setting	Internally used resource
1	P04/INTP100/DMARQ0-	ON	DMARQ0- of GBUS
2	P05/INTP101/DMARQ1-	ON	DMARQ1- of GBUS
3	P06/INTP102/DMARQ2-	ON	DMARQ2- of GBUS
4	P07/INTP103/DMARQ3-	ON	DMARQ3- of GBUS
5	PBD0/DMAAK0-	ON	DMAAK0- of GBUS
6	PBD1/DMAAK1-	ON	DMAAK1- of GBUS
7	PBD2/DMAAK2-	ON	DMAAK2- of GBUS
8	PBD3/DMAAK3-	ON	DMAAK3- of GBUS

[SW7]

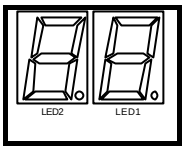
No.	CPU pin name	Factory setting	Internally used resource
1	P24/INTP110/TC0-	ON	Pin 129 of GBUS
2	P25/INTP111/TC1-	ON	Pin 130 of GBUS
3	P26/INTP112/TC2-	ON	Pin 131 of GBUS
4	P27/INTP113/TC3-	ON	Pin 132 of GBUS

[SW8]

No.	CPU pin name	Factory setting	Internally used resource
1	P02/INTP001/TI001	ON	GINT1- of GBUS
2	P11/INTP010/TI010	ON	GINT2- of GBUS
3	P12/INTP011/TI011	ON	GINT3- of GBUS
4	P21/INTP020/TI020	ON	OUT1- of TIC
5-8	Not used	OFF	Not used

5.8. 7SEG-LED, xxx-LED

The LEDs are used to indicate statuses, as listed below. The two 7-segment LEDs are used by the monitor at startup. After that, they can be used for any user application.



LED	Description
POWER	Lights when power is supplied to the RTE-V850E/MA1-CB board.
TOVRDY	Lights when time-over ready occurs, and does not go off until cleared by software (see Section 7.4.5).
CS0	Lights when CS0 space is accessed.
CS1	Lights when CS1 space is accessed.
CS2	Lights when CS2 space is accessed.
CS3	Lights when CS3 space is accessed.

Board LED Status

5.9. TEST PINS FOR ROM EMULATOR (JROM-EM1)

Test pins (JROM-EM1s) are used to connect a ROM emulator. They accept control signals listed below. The following table lists the signal names and functions.

Signal name	Input/ output	Function
RESET- (1)	Input	When a low level is supplied to this test pin, the CPU is reset. A reset request signal from the ROM emulator is connected to the test pin. The test pin is pulled up with 1 kΩ.
NMI- (2)	Input	When a low level is supplied to this test pin, an NMI signal is given to the CPU. (See Section 10.5.) An NMI request signal from the ROM emulator is connected to the test pin. The test pin is pulled up with 1 kΩ.
GND (3)	---	This test pin is at a ground level. The ground level of the ROM emulator is connected to the test pin.

JROM-EM1 Pin Functions

5.10. CLOCK SOCKET (OSC1)

An oscillator for generating the clock signal to be supplied to the CPU is mounted in the OSC1 socket. OSC1 is converted to the 3.3-V level, and is connected to the CPUCLK pin of the CPU.
Accepts DIP 8-pin (half-type) oscillators.

When you have to cut an oscillator pin for convenience, be careful not to cut it too short, or otherwise the frame (housing) of the oscillator may touch a pin in the socket, resulting in a short-circuit occurring.

5.11. CRYSTAL SOCKET (JP1)

JP1 has two functions: it is used to select the clock supplied to the CPU and it also acts as the connector for the crystal oscillator.

To use OSC1 as CPU clock

Short-circuit JP1 pins 1 and 2. In this case, do not mount the crystal.

To mount crystal on JP1 and use the CPU oscillation circuit

Mount the crystal between pins 1 and 3 on JP1. Do not short-circuit pins 1 and 2.

5.12. AVDD/AVREF SELECTOR JUMPER (JP2)

The JP2 jumper is used to select the voltage for AD (AVDD/AVREF) to be supplied to the CPU.

To supply voltage from board: Factory-set condition

Jumper pins 1 and 2 of JP1. +3.3 V will be supplied.

To supply voltage from external source (JCPU)

Jumper pins 2 and 3 of JP1. Supply the voltage from pin JCPU2-78.

5.13. ROM SOCKETS

The RTE-V850E/MA1-CB has ROM sockets to hold 40-pin ROM chips to provide standard 128K bytes (64K × 16 bits). When the ROM chips used here are to be replaced, their type should be 27C1024, 27C2048, or 27C4096, and the access time should be 120 ns or less.

5.14. SELF-WRITING POWER CONNECTOR (JVPP)

To execute self-writing of the CPU's internal flash ROM, a power supply capable of supplying 7.8 V is necessary. The JVPP connector is used to connect such a power supply.

To execute self-writing, connect a 10 to 12 V power supply to this connector and output a low level to pin P22 of the CPU. This supplies 7.8 V to the VPP pin of the CPU, thus enabling self-writing.

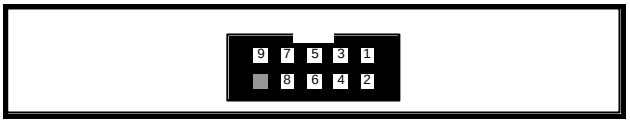
The pin configuration of JVPP is as shown below.

JVPP pin No.	Name	Input/output	Function
1	10-12V	Input	Inputs voltage of 10 to 12 V.
2	GND	Input	Connected to power supply GND.

5.15. FLASH WRITING CONNECTOR (JFLASH)

The JFLASH connector is used to write data to the CPU's internal flash ROM by using a flash programmer device.

To use JFLASH, SW4 must be set (see Section 5.6).



JFLASH Pin Arrangement

JFLASH pin No.	Signal name	Input/output	Remark
1	SO0	Output	Synchronous serial data output (CMOS level)
2	SI0	Input	Synchronous serial data input (CMOS level)
3	SCK0-	Input	Synchronous serial clock input (CMOS level)
4	RESET-	Input	Reset input
5	VPP	Input	VPP input
6	+3.3V	Output	CPU core power level output
7	+3.3V	Output	CPU-I/O power level output
8	GND	–	Ground
9	NC	–	No connection
(10)	NC	–	No connection (without pin)

JFLASH Connector Signals

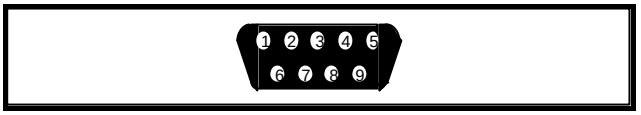
5.16. SERIAL CONNECTOR (JSIO1, JSIO2)

The JSIO1 connector is used for the RS-232C interface that is controlled by the serial controller (TL16C550CPT) on the board.

The JSIO2 connector is used for the RS-232C interface that is controlled by the built-in serial controller of the CPU.

JSIO1 and JSIO2 are 9-pin D-SUB RS-232C connectors (male) like that commonly used on the PC/AT. All signals on both of these connectors are converted to the RS-232C level. The figure and table below indicate the pin and signal arrangement of these connectors.

For the signals to be connected to the host, the table indicates two modes of wiring on the host: one for a 9-pin D-SUB connector, and the other for a 25-pin D-SUB connector. (Regular cross-cable wiring is used for these connections.)



Pin Arrangement of JSIO1 and JSIO2 (Male)

JSIO1 pin No.	Signal name	Input/ output	Connector pin number on the host side	
			D-SUB9	D-SUB25
1	DCD	Input		
2	RxD(RD)	Input	3	2
3	TxD(SD)	Output	2	3
4	DTR(DR)	Output	1, 6	6, 8
5	GND		5	7
6	DSR(ER)	Input	4	20
7	RTS(RS)	Output	8	5
8	CTS(CS)	Input	7	4
9	RI	Input		

JSIO1 Connector Signals

JSIO2 pin No.	CPU pin	Signal name	Input/ output	Connector pin number on the host side	
				D-SUB9	D-SUB25
1[Note]	--	DCD	Input		
2	P41	RxD (RD)	Input	3	2
3	P40	TxD (SD)	Output	2	3
4	P45	DTR (DR)	Output	1, 6	6, 8
5		GND		5	7
6	P44	DSR (ER)	Input	4	20
7	P42	RTS (RS)	Output	8	5
8	P43	CTS (CS)	Input	7	4
9[Note]	--	RI	Input		

JSIO2 Connector Signals

Note JSIO2 pins 1 and 9 are not used on board.

5.17. JGBUS CONNECTOR (JGBUS)

This is a 32-bit bus connector for expansion. For details, see Chapters 11 and 14.

5.18. CPU CONNECTOR (JCPU1, JCPU2)

The CPU connector signals are connected directly to the V850E/MA1. Many signals are used on the board. So, be careful when extracting signals from the JCPU.

JCPU pin No.	Signal name	JCPU pin No.	Signal name
1	+3.3V	2	GND
3	PDL15/D15	4	PDL14/D14
5	PDL13/D13	6	PDL12/D12
7	PDL11/D11	8	PDL10/D10
9	PDL9/D9	10	PDL8/D8
11	PDL7/D7	12	PDL6/D6
13	PDL5/D5	14	PDL4/D4
15	PDL3/D3	16	PDL2/D2
17	PDL1/D1	18	PDL0/D0
19	NC.	20	NC.
21	+3.3V	22	GND
23	P07/INTP103/DMARQ3-	24	P06/INTP102/DMARQ2-
25	P05/INTP101/DMARQ1-	26	P04/INTP100/DMARQ0-
27	P03/TO00	28	P02/INTP001/TI001
29	P01/INTP000/TI000	30	P00/PWM0
31	PBD3/DMAAK3-	32	PBD2/DMAAK2-
33	PBD1/DMAAK1-	34	PBD0/DMAAK0-
35	P13/TO01	36	P12/INTP011/TI011
37	P11/INTP010/TI010	38	P10/PWM1
39	NC.	40	NC.
41	+3.3V	42	GND
43	P27/INTP113/TC3-	44	P26/INTP112/TC2-
45	P25/INTP111/TC1-	46	P24/INTP110/TC0-
47	P23/TO02	48	P22/INTP021/TI021
49	P21/INTP020/TI020	50	P20/NMI
51	P37/INTP123/ADTRG	52	P36/INTP122
53	P35/INTP121	54	P34/INTP120/RXD2
55	P33/INTP133/TXD2	56	P32/INTP132/SCK2-
57	P31/INTP131/SI2	58	P30/INTP130/SO2
59	NC.	60	NC.
61	+5V	62	GND
63	RESET-	64	NC.
65	P45/SCK1-	66	P44/SI1/RXD1
67	P43/SO1/TXD1	68	P42/SCK0-
69	P41/SI0/RXD0	70	P40/SO0/TXD0
71	NC.	72	NC.
73	NC.	74	NC.
75	NC.	76	NC.
77	NC.	78	RESET_REQ-**Note**
79	+5V	80	GND

JCPU1 Connector

Note RESET_REQ is a reset request signal line from JCPC (low-active).

JCPU pin No.	Signal name	JCPU pin No.	Signal name
1	+3.3V	2	GND
3	PAL0/A0	4	PAL1/A1
5	PAL2/A2	6	PAL3/A3
7	PAL4/A4	8	PAL5/A5
9	PAL6/A6	10	PAL7/A7
11	PAL8/A8	12	PAL9/A9
13	PAL10/A10	14	PAL11/A11
15	PAL12/A12	16	PAL13/A13
17	PAL14/A14	18	PAL15/A15
19	PAH0/A16	20	PAH1/A17
21	+3.3V	22	GND
23	PAH2/A18	24	PAH3/A19
25	PAH4/A20	26	PAH5/A21
27	PAH6/A22	28	PAH7/A23
29	PAH8/A24	30	PAH9/A25
31	NC.	32	NC.
33	NC.	34	NC.
35	PCD0/SDCKE	36	PCD1/SDCLK
37	PCD2/LBE-/SDCAS-	38	PCD3/UBE-/SDRAS-
39	PCS0/CS0-	40	PCS1/CS1-/RAS1-
41	+3.3V	42	GND
43	PCS2/CS2-/IOWR-	44	PCS3/CS3-/RAS1-
45	PCS4/CS4-/RAS4-	46	PCS5/CS5-/IORD-
47	PCS6/CS6-/RAS6-	48	PCS7/CS7-
49	PCT0/LCAS-/LWR-/LDQM	50	PCT1/UCAS-/UWR-/UDQM
51	PCT4/RD-	52	PCT5/WE-
53	PCT6/OE-	54	PCT7/BCYST-
55	PCM0/WAIT-	56	PCM1/CLKOUT/BUSCLK
57	PCM2/HLDAK-	58	PCM3/HLDRQ-
59	PCM4/REFRQ-	60	PCM5/SELFREF
61	+5V	62	GND
63	NC.	64	NC.
65	NC.	66	P50/TI030/INTP030
67	P51/TI031/INTP031	68	P52/TO03
69	P70/ANI0	70	P71/ANI1
71	P72/ANI2	72	P73/ANI3
73	P74/ANI4	74	P75/ANI5
75	P76/ANI6	76	P77/ANI7
77	AVSS (GND)	78	AVDD/AVREF (+3.3V)
79	+5V	80	GND

JCPU2 Connector

6. CONNECTION WITH THE HOST PC

6.1. RS-232C CONNECTION

Serially connect the host machine using the monitor ROM by means of the following procedure:

<1> Get an optional RS-232C cable and a power supply.

<2> Set and check the setting of the switches on the board. Specify a baud rate by using SW1 (see Sections 12.1.2 and 13.1.1).

<3> Connect the JSIO1 connector and host machine with the RS-232C cable, and supply power to the JPOWER connector. Confirm that the POWER-LED on the board lights and that the 7-segment LED indicating that the monitor has started lights.

If the LED does not light, turn off the power immediately, and check the connection.

<4> Start the debugger on the host machine, and connect it via the RS-232C interface. If an error occurs, confirm the connection of the serial cable and the setting of the switches (especially, baud rate). For the method and procedure of starting the debugger, see the debugger manual.

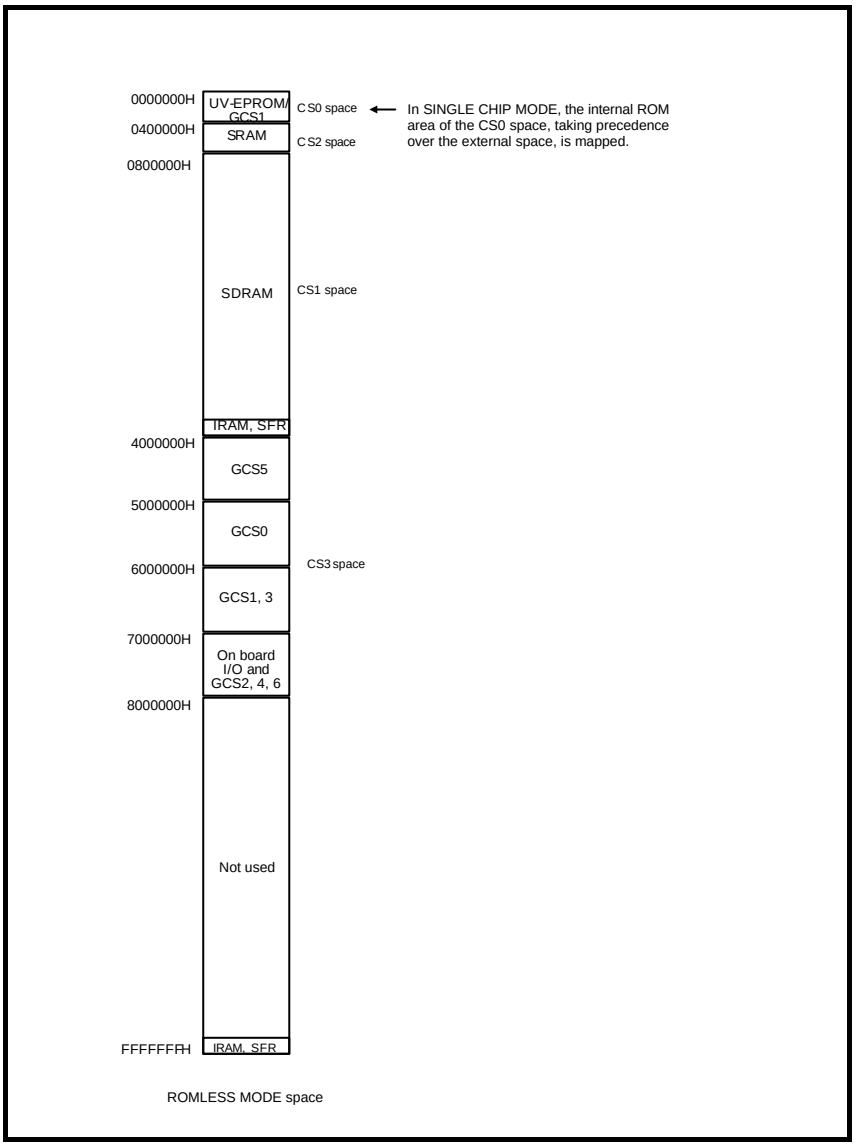
Place the board on an insulating material. If a conductive material touches the board while power is supplied to the board, the board may malfunction.

7. HARDWARE REFERENCES

This chapter describes the hardware of the RTE-V850E/MA1-CB.

7.1. MEMORY AND I/O MAP

The figure below shows the memory and I/O mapping on the board.



Memory I/O Map

CS0 space (UV-EPROM, GCS1-): 0000000 to 03FFFFF (4M bytes)

UV-EPROM is allocated to the CS0 space, or the CS0 space is reserved as a space for GCS-1 of GBUS. On-board UV-EPROM is allocated to the space when SW2-1 (FBOOT) is OFF. When SW2-1 (FBOOT) is ON, it is reserved as a space for GCS1- of GBUS.

The GCS1 space can be accessed from the CS3:6000000 to 6FFFFFF spaces. By locating a flash ROM in the GCS1 space of GBUS, therefore, the flash ROM can be rewritten by using the monitor ROM of UV-EPROM. Thereafter, a program can be booted from the flash ROM.

For GBUS, see Chapters 11 and 14.

In single-chip mode, the internal ROM space cannot be accessed from an external source.

CS2 space (SRAM): 0400000 to 07FFFFF (4M bytes)

SRAM is allocated to the CS2 space. The actual capacity is 1M byte. An image is generated every 1M byte within this space. When the monitor is used, part of this space is reserved for the monitor work area (see Sections 12.2.2 and 13.2.2).

CS1 space (SDRAM): 0800000 to 3FFFFFF (56M bytes)

SDRAM is allocated to the CS1 space. The actual capacity is 32M bytes. An image is generated every 32M bytes within this space. Access to the internal resources for an internal RAM or SFR area takes precedence over access to SDRAM.

CS3 space (GCS5-): 4000000 to 4FFFFFF (16M bytes)

The CS3 space is reserved as a space for GCS5- of GBUS. For GBUS, see Chapters 11 and 14.

CS3 space (GCS0-): 5000000 to 5FFFFFF (16M bytes)

The CS3 space is reserved as a space for GCS0- of GBUS. For GBUS, see Chapters 11 and 14.

CS3 space (GCS1-, GCS3-): 6000000 to 6FFFFFF (16M bytes)

The CS3 space is reserved as an area for GCS1- and GCS3- of GBUS. For GBUS, see Chapters 11 and 14.

CS3 space (I/O, GCS2-, GCS4-, GCS6-): 7000000 to 7FFFFFF (16M bytes)

The CS3 space is used as an I/O space. There are reserved spaces for the board's I/O and GCS2-, GCS4-, and GCS6- of GBUS. For the I/O map, see Section 7.4.1. For GBUS, see Chapters 11 and 14.

CS4 to CS7 spaces (not used): 8000000 to FFFFFFF (128M bytes)

The CS4 to CS7 spaces are not used on the board.

7.2. RECOMMENDED SETTINGS

This section explains the recommended setting values of each register related to memory and I/O resource access.

7.2.1. MEMC Registers

Make the following register settings for the system bus. Note that the setting for some registers differs depending on the setting of SW2-3 (BCLK_LOW).

Register name	Address	Setting	Remarks
CKC	0xFFFF822	0x07	DCLK: 00, TBCS: 00, CESEL: 00, CKDIV: 111
BCT0	0xFFFF480	0x88B8	CS0, 2, 3: SRAM/IO, CS1: SDARM
BCT1	0xFFFF482	0x8888	Same as the CPU initial value
DWC0 (BCLK_LOW = OFF) DWC0 (BCLK_LOW = ON)	0xFFFF484	0x1111 0x0000	CS0-3: 1 wait CS0-3: 0 wait
DWC1	0xFFFF486	0x7777	CS4-7,: 7 waits: Same as the CPU initial value
VSWC (BCLK_LOW = OFF) VSWC (BCLK_LOW = ON)	0xFFFF06E	0x14 0x12	VPB: BCLK >= 33 MHz = 0x14 < 33 MHz = 0x12
BCC	0xFFFF488	0xFFC0	CS0-2: 0 clk, CS3-7: 3clks
ASC	0xFFFF48A	0x5555	All 1 addr wait
BCP	0xFFFF48C	0x00	Normal bus cycle
CSC0	0xFFFF060	0xFCF3	(Chip Select Control Register0)
CSC1	0xFFFF062	0x2C11	(Chip Select Control Register1): Same as the CPU initial value
BSC	0xFFFF066	0x5555	All 16-bit: Same as the CPU initial value
BEC	0xFFFF068	0x0000	All little endian: Same as the CPU initial value
SCR1	0xFFFF4A4	0x2096	SDRAM (LTM = 2, BCW = 2, SSO = 16, RAW = 12, SAW = 10)
RFS1 (BCLK_LOW = OFF) RFS1 (BCLK_LOW = ON)	0xFFFF4A6	0x8017 0x800F	50 MHz: 15.4 μs 33 MHz: 15.5 μs

[Caution] For the setting procedure of SDRAM-related registers (SCR1 and SFR1), refer to the CPU manual.

7.3. MEMORY RESOURCES

RTE-V850E/MA1-CB has SDRAM, SRAM, and UV-EPROM as on-board memory resources. As part of the memory space, the GBUS chip select space is reserved.
This section explains these resources and memory devices.

7.3.1. SDRAM (CS1: 0800000 to 3FFFFFF)

Four SDRAM devices (μPD4564441G5), each consisting of 4M words × 4 bits × 4 banks, are provided as SDRAM. The total capacity is 32M bytes.

7.3.2. SRAM (CS2: 0400000 to 07FFFFF)

Two high-speed SRAM devices, each having a capacity of 512K words × 8 bits and a speed of 15 ns, are provided as SRAM. The total SRAM capacity, therefore, is 1M byte. If the bus clock exceeds 33 MHz, insert one wait cycle by using the wait controller of the CPU. At a clock frequency up to 33 MHz, it can be accessed without a wait cycle. Because the high-order bits of the address lines are not decoded, an image appears every 1M byte. The second-half 32 KB of the SRAM is used by the monitor as a work area, and cannot be used by a user program (see Sections 12.2.2 and 13.2.2).

7.3.3. UV-EPROM (CS0: 0000000 to 03FFFFF)

A ROM of 128K bytes (64K words × 16 bits), 256K bytes (128K words × 16 bits), or 512K bytes (256K words × 16 bits) with an access time of 120 ns or less can be mounted as UV-EPROM. The type of ROM to be mounted and conditions are specified with SW3 (see Section 5.5). Because the high-order bits of the address lines are not decoded, an image appears for each ROM capacity.
Depending on the setting of SW2-3 (BCLK_LOW), the number of ROM wait cycles is changed forcibly by hardware as shown below.

SW2-3 (BCLK_LOW): OFF = 7 wait cycles
SW2-3 (BCLK_LOW): ON = 5 wait cycles

7.4. I/O MAP

On-board I/O for the RTE-V850E/MA1-CB includes a serial controller (TL16C550CPT), timer (μPD71054), LEDs, and switches. Also the chip select space of GBUS is reserved as part of the I/O space.

This section explains mapping of the on-board I/O and I/O devices.

7.4.1. I/O List

The table below lists the I/O areas and functions. The number of wait cycles changes depending on the setting of SW2-3 (BCLK_LOW).

Address	Use	Number of wait cycles	
		BCLK_LOW OFF	BCLK_LOW ON
7800000	SW1	10	7
7801000	SW2	10	7
7802000	7SEGLED	10	7
7803000	TOVRDY_LED_CLRPLS	10	7
7804000 to 7804020	PIC	10	7
7807000 to 7807070	UART (TL16C550C)	10	7
7808000 to 7808030	TIC (μPD71054)	10	7
7809000	GBUS BYTE ACCESS CONTROL	10	7

7.4.2. SW1 Read Port (SW1 7800000H [Read Only])

This port is used to read the status of SW1. The table below indicates the data format.

Physical address	Data bus								Setting
	D7	D6	D5	D4	D3	D2	D1	D0	
7800000H input	SW1 -8	SW1 -7	SW1 -6	SW1 -5	SW1 -4	SW1 -3	SW1 -2	SW1 -1	0 = ON 1 = OFF

SW1-1 corresponds to switch 1 of SW1, while SW1-8 corresponds to switch 8 of SW1. When a bit of the corresponding switch is set to ON, 0 is read. When it is set to OFF, 1 is read. SW1 is used to set the operation of the monitor. For how to set this switch, see Sections 12.1.2 and 13.1.1.

7.4.3. SW2 Read Port (SW2 7801000H [Read Only])

This port is used to read the status of SW2. The data format of this port is shown in the table below.

Physical address	Data bus								Setting
	D7	D6	D5	D4	D3	D2	D1	D0	
7801000H input	SW2 -8	SW2 -7	SW2 -6	SW2 -5	SW2 -4	SW2 -3	SW2 -2	SW2 -1	0 = ON 1 = OFF

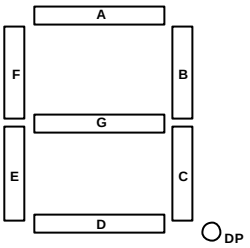
SW2-1 corresponds to bit 1 of SW2, and SW2-8 corresponds to bit 8 of SW2. When a bit of the corresponding switch is set to ON, 0 is read; when it is set to OFF, 1 is read. SW2 is used to switch the hardware operation. For the function of each switch, see Section 5.4.

7.4.4. 7-Segment LED Display Data Output Port (7SEG-LED 7802000H [Write Only])

This port sets the data to be displayed on the four 7-segment LED. The table below indicates the data format. When a bit is set to 0, the corresponding segment is turned on.

Logical address	Data bus								Setting
	D7/D15	D6/D14	D5/D13	D4/D12	D3/D11	D2/D10	D1/D9	D0/D8	
7802000H output	LED1 -DP	LED1 -G	LED1 -F	LED1 -E	LED1 -D	LED1 -C	LED1 -B	LED1 -A	0 = Turned on 1 = Turned off
7802001H output	LED2 -DP	LED2 -G	LED2 -F	LED2 -E	LED2 -D	LED2 -C	LED2 -B	LED2 -A	

The figure below illustrates the correspondence between the bits and the segments of the 7-segment LED.



7.4.5. Time-Over Ready LED Clear Pulse (TOVRDY_LED_CLRPLS 7803000H [Write Only])

If data is written to the port, the TOV_RDY LED, which lights when time-over ready occurs on the board, goes off, and the written data is ignored. Once the TOV_RDY LED is on, it does not go off until data is written to the port or the board is reset.

7.4.6. Interrupt Controller (PIC: 7804000H to 7804020H [Read/Write])

The PIC supports the Multi and PARTNER interrupts necessary for monitor program execution. The interrupts that can be connected are as follows:

1) Communication interrupt from RS-232C devices (UART, TL16C550C)
2) Timer interrupt request of TOUT0 of the timer (TIC, μPD71054)
3) Occurrence of time-over ready
4) GINT0 interrupt

Logical address	Register	Data bus							
		D7	D6	D5	D4	D3	D2	D1	D0
7804000H	PIC INT-MASK	x	x	x	x	IM3	IM2	IM1	IM0
7804010H	PIC INT-STATUS	x	x	x	x	IR3	IR2	IR1	IR0
7804020H	PIC INTEN	x	x	x	x	0	0	INTP000	INTEN

The INT-MASK register masks interrupts applied to its bits. When an INT_MASK bit is set to 1, the interrupt is enabled. When multiple bits are selected, each OR value activates an interrupt.

The INTR register is an interrupt status register, for which 1 is read whenever there is an interrupt request. This does not depend on the state of masking. To clear an edge interrupt request, the corresponding bit of this register must be set to 1.

The table below indicates the interrupt source assigned to each bit of IM[0..3] and IR[0..3].

PIC INT-MASK[], STATUS[]	Interrupt source	Request level
0	Timer 0 (mode 2)	Edge (rising)
1	Serial 0	Level (high)
2	Time-over	Level (high)
3	GINT0-	Level (low)

The INTEN register enables or disables all interrupts.

INTEN: Disables the interrupt to be used by the monitor by hardware. At this time, the interrupt request pin is low.

INTEN	NMI0/INTP000
0	Sets a mask. (Reset value)
1	Does not set a mask.

INTP000: Selects the interrupt to be used by the monitor.

INT0/NMI-	Interrupt for monitor
0	NMI0 is used. (Reset value)
1	INTP000 is used.

INTEN is used to create an edge for an interrupt request signal to the CPU each time an interrupt has been serviced when more than one interrupt occurs. If the CPU detects an interrupt reception using the edge, execute the processing that first sets the INTEN bit to 1 and then clears it to 0 at the last step of the interrupt handler. This allows a pending interrupt to be acknowledged.

[Caution] Do not change the contents of the PIC while the monitor is being used.

7.4.7. UART (TL16C550C: 7807000H to 7807070H)

The Texas Instruments TL16C550C LSI is used as the UART controller. The TL16C550C has an UART channel. It also has a 16-character FIFO buffer in the transmission/reception block of the UART, and a function for automatically controlling RTS/CTS flow. Therefore, an overrun error of communication can be suppressed by the minimum interrupt.

Each register of the TL16C550C is assigned as listed below. For an explanation of the function of each register, refer to the manual provided with the TL16C550C. (The manual for the TL16C550C is available from the TI&ME of the Texas Instruments home page (http://www.ti.com/).)

Address	Read	Write
7807000H	RBR/DLL	THR/DLL
7807010H	IER/DLM	IER/DLM
7807020H	IIR	FCR
7807030H	LCR	LCR
7807040H	MCR	MCR
7807050H	LSR	LSR
7807060H	MSR	MSR
7807070H	SCR	SCR

TL16C550C Register Arrangement

The XIN input of the TL16C550C is connected to the 16-MHz clock.

The UART interrupt can be input to NMI1 of the CPU via PIC of the CPU.

The UART is connected to the JSIO1 connector of the board. The UART is used to communicate with the host when the remote debugger is used.

TL16C550C is reset when the system is reset.

[Caution] Do not change the contents of the UART while the monitor is being used.

7.4.8. TIC (μPD71054 7808000H to 7808030FH)

The NEC μPD71054 is installed as a TIC. The μPD71054 is compatible with the Intel i8254. It has three timers/counters. These timers/counters are used to generate monitor timer interrupts.

Each register of the TIC is assigned as listed below.

Address	Read	Write
7808000H	COUNTER#0	COUNTER#0
7808010H	COUNTER#1	COUNTER#1
7808020H	COUNTER#2	COUNTER#2
7808030H	-----	Control Word

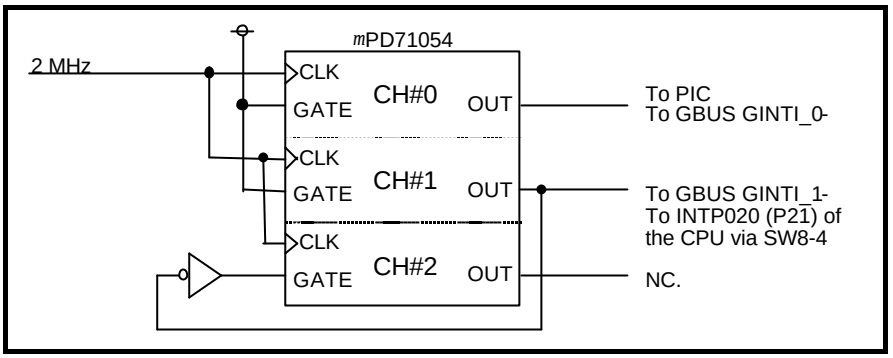
TIC Register Arrangement

The channels of the TIC are connected as shown in the figure below.

Channel 0 is used as the interval timer for the Multi ROM monitor program.

Channels 1 and 2 can be used by a user program as necessary.

Channel 2 is connected to channel 1 by means of a cascade connection.



Examples of modes

CH#0: Mode 2 (rate generator)
CH#1: Mode 2 (rate generator)
CH#2: Mode 0 (down counter)

[Caution] Do not change the contents of the CH0 while the monitor is being accessed.

7.4.9. GBUS ACCESS CONTROL (7809000H [Read/Write])

This register is used when accessing in access sizes other than half WORD (16-Bit) to GBUS.

Logical address	Register	Data bus							
		D7	D6	D5	D4	D3	D2	D1	D0
7809000H	GBUS ACC	x	x	x	x	x	x	Word	Byte

Byte: Set this bit to 1 to read GBUS in byte units. Usually, keep this bit set to 0. This is used to access GBUS in byte units. The read cycle of the V850E/MA1 is executed in half-word units, regardless of the access size of an instruction. Depending on the resources on the GBUS, however, some devices request an access only in byte units. At the time of access to such a device, please set this bit 1 and access in byte size from CPU.

Word: Set this bit to 1 to access GBUS in word(32-Bit) units. Usually, keep this bit set to 0. This is used to access GBUS in word units. The external bus of V850E/MA1 is 16-Bit. However, depending on the resources on the GBUS, some devices request an access only in word units. At the time of access to such a device, please set this bit 1 and access in word size from CPU.

[Caution]

Please do not set 1 as Bit of both Byte and Word.

After setting change should surely do IOWAIT() (dummy read of ROM area).

The function of Word-Bit is applied to what has the version of a board newer than 2.2.

8. SOFTWARE

This chapter describes the initialization of the hardware of the RTE-V850E/MA1-CB board, and explains how to use peripheral devices.

8.1. INITIALIZATION

To write a program that is booted from ROM without using the monitor, initialize the internal bus controller of the V850E/MA1 in the first routine of the program. For the values to be set for initialization, see Section 7.2.

8.2. SUCCESSIVE ACCESSES TO μPD71054

To successively access the μPD71054, access other spaces at least once between the first and second accesses to the μPD71054. These accesses ensure the recovery time of the μPD71054.

Recovery time is also assured by a dummy read of a resource (such as ROM) other than the μPD71054.

8.3. LIBRARIES

Libraries are required for programming using the C compiler for I/O accesses and other purposes. However, the methods of writing these libraries and passing their parameters are specific to the GHS. So, modifications may be required, for example, when another compiler is used.

```
/* I/O library */

/* GHS V800 compiler parameter passing */
/* arg0:r6, arg1:r7, arg2:r8, return:r10 */

inb(int addr)                       /* Byte (8 bits) input */
{
    __asm("  ld.b 0[r6], r10");
}

inh(int addr)                       /* Half-word (16 bits) input */
{
    __asm("  ld.h 0[r6], r10");
}

inw(int addr)                       /* Word (32 bits) input */
{
    __asm("  ld.w 0[r6], r10");
}

outb(int addr, int data)            /* Byte (8 bits) output */
{
    __asm("  st.b r7, 0[r6]");
}

outh(int addr, int data)            /* Half-word (16 bits) output */
{
    __asm("  st.h r7, 0[r6]");
}

outw(int addr, int data)            /* Word (32 bits) output */
{
    __asm("  st.w r7, 0[r6]");
}
```

8.4. EXAMPLE OF USING TIMERS

A sample time measurement is indicated below which uses timer 1 and timer 2 cascaded with each other by an external timer (μPD71054) on the board. Timer 1 is initialized as an interval counter (mode 2), and timer 2 is initialized as a down counter (mode 0). By determining the counter values before and after a routine whose execution time is to be measured, the execution time can be calculated. Note that both timers function as down counters. Note also that command recovery (ROM area dummy read) is required for successive accesses to the external timer.

```
/* Sample execution time measurement using timers */

#define TIMERCLK       2000000               /* 2 MHz */
#define INTERVAL       (TIMERCLK * 10 / 1000) /* 10 ms (1/100) */
#define IOWAIT()       (*(char *) 0x3D80050)  /* For I/O command recovery */

InitTimer() /* Timer initialization */
{
            outb(0x7808030, 0x74);           IOWAIT();  /* Timer 1 set to mode 2 */
            outb(0x7808010, INTERVAL);       IOWAIT();  /* Lower-digit count of timer 1 */
            outb(0x7808010, INTERVAL /256);  IOWAIT();  /* Higher-digit count of timer 1 */
            outb(0x7808030, 0xB0);           IOWAIT();  /* Timer 2 set to mode 0 */
            outb(0x7808020, 0xFF);           IOWAIT();  /* Lower-digit count of timer 2 */
            outb(0x7808020, 0xFF);           IOWAIT();  /* Higher-digit count of timer 2 */
            return 0;
}

LatchTimer()          /* Count latch */
{
            int count1, count2, counts;

            outb(0x7808030, 0xDC);           IOWAIT();  /* Timer 1/2 multiple latch */
            count1  = inb(0x7808010);        IOWAIT();
            count1 += inb(0x7808010) *256;   IOWAIT();  /* Count of timer 1 */
            count2  = inb(0x7808020);        IOWAIT();
            count2 += inb(0x7808020) *256;   IOWAIT();  /* Count of timer 2 */
            counts  =  INTERVAL * (0xFFFF - count2)
               + (INTERVAL - count1);
          return counts;
}

double total_time;

main()
{
            int start_count, stop_count;

            InitTimer();
            start_count = LatchTimer();       /* Start count value */
            func();
            stop_count = LatchTimer();        /* Stop count value */
            total_time = (double)(stop_count-start_count)
                        / (double)TIMERCLK;   /* Seconds */
            return 0;
}

#include <time.h>

func()                /* Time measurement routine */
{
          ....
}
```

9. DEVELOPMENT OF APPLICATIONS USING MASKABLE INTERRUPTS

This chapter describes the methods of developing an application on the RTE-V850E/MA1-CB by using a maskable interrupt, and related restrictions.

9.1. INTERRUPT VECTOR

The V850E/MA1 interrupt vector area of addresses 000000H to 0007FFH is fixed in the ROM, and cannot be rewritten. So, for the monitor, the following two vector areas are allocated in the SRAM:

Alternate vector area:

This vector area can be rewritten by the user program. It is used if a relative jump can be executed from the interrupt vector area. The branch instruction for the relative jump is placed in the vector area in this case.

Relay vector area:

This vector area is used by the monitor if a relative jump from the interrupt vector area cannot be executed. In this case, an instruction that saves the registers and a branch instruction for an absolute jump are placed in the interrupt vector area, and an instruction that restores the registers and a branch instruction for a relative jump to the alternate area are placed in this area.

Because the monitor can execute a relative jump from the interrupt vector area (addresses 0000000H to 00007FFH) to the alternate vector area, it branches to the alternate vector area via relay vector area.

Alternate vector area	Relay vector area
4F8000H to 4F87FFH	4F8800H to 4F8FFH

If, for example, an interrupt with exception code 0080H is generated, the CPU interrupt function causes a branch to address 000080H, where an instruction for causing a branch to 0080H (the offset of alternate vector area) is placed. It branches to offset address 0080H of the alternate vector area via that vector. By rewriting the alternate vector area in the destination, a branch to the user program interrupt handling routine can be caused when an interrupt is generated.

If an exception code 0080H interrupt occurs, therefore, write an instruction that branches execution to the specified interrupt processing at address 4F8080H.

The difference from an ordinary V850E/MA1 program is that a vector area is fixed in ROM, and no setting (rewriting) by a program is required. However, a program using the monitor on the RTE-V850E/MA1-CB must rewrite the vector area by the program to enable an interrupt.

A sample program for alternate vector rewriting is given below (when the relative address from the interrupt handling routine to an alternate vector area is within 22 bits).

```
void SetAJump(int addr, int jmpdest) /* ← Vector setting routine */
/* int addr;        address where we're storing the 'jr' */
/* int jmpdest;     address where the 'jr' jumps to */
{
   int offset;
   unsigned inst;
   unsigned int *p ;

   offset = jmpdest - addr;
   inst = 0x07800000 /* 'jr' opcode */ | (offset & 0x003fffff);
   *((UINT16 *)(addr    )) = (inst >> 16) & 0xffff ;
   *((UINT16 *)(addr + 2)) = (inst      ) & 0xffff ;
}
.....
void __interrupt IntEntry() /* ← Interrupt handling routine */
{
   .......
}
.....
main()
{
   ........
         SetAJump((int)(0x080 + 0x4f8000) ,(int)IntEntry) ;
         /*             ↑ Exception code of specified interrupt */
```

9.2. GENERAL RESTRICTIONS/NOTES

This section describes restrictions and notes relating to the debugging of an application using a maskable interrupt.

1) If an interrupt is generated before alternate vector setting, or if an interrupt is generated with other than a valid alternate vector set, a break occurs at the point where the interrupt is generated. This is because the initial value of the alternate vector is an instruction for causing a branch to the break handling routine of the monitor.
2) If the relative address from an alternate vector area to the interrupt handling routine exceeds 22 bits, the contents of at least one register must be destroyed, or a branch relay point must be created, to cause a branch to the interrupt handling routine.
3) The alternate vector area can be rewritten by the program or when the program is downloaded (see Section 9.3). To rewrite these areas when the program is downloaded, however, rewrite only the interrupts that are used.
4) All peripherals, including interrupt-related peripherals, can be initialized only with the reset switch on the board. This means that if, after a program is executed, another program is loaded, the peripherals will still be in the statuses set by the previous program. So, use the procedure below when, for a program that uses a peripheral, another program is to be loaded and executed.
   (1) Disconnect the monitor.
   (2) For resetting the board, press the reset switch of the RTE-V850E/MA1-CB.
   (3) Connect the monitor.
   (4) Load and execute another program.
5) Before setting the EI (interrupt enable) state, set the DI (interrupt disable) state at the start of program execution, then set the peripherals and vectors.
6) To disable (DI) or enable (EI) an interrupt during a break with the I/O (register) manipulation function of the debugger, use the corresponding bit of the interrupt mask register (IMRn). If the interrupt control register (PICn or PnnICn) is manipulated with the I/O (register) manipulation function of the debugger during a break, do nothing to the interrupt control register, since the interrupt operation may not be performed correctly.

9.3. REWRITING THE ALTERNATE VECTOR AREA DURING DOWNLOADING

A vector can be rewritten in various ways while a program is being downloaded. This section shows an example in the Multi environment of GHS. The method used can be thought of as being similar to a program stored to ROM. Also see the program example shown above.

1) Defining program for rewriting the interrupt vector area (ASM language)
   Define a program consisting of only the branch instructions to be placed in the interrupt vector as shown below. For details of the coding, refer to the manual for the language processor.

```
      .section    "intvct", .text  /* Defined section name */
      .align    4
      .globl      _Int80
_Int80:
      jr _IntEntry                 /* jump to handler */
      nop
      nop
```

   Note, however, that the program cannot be defined in such a way that it exceeds the vector boundary for one interrupt.

2) Defining the section map

Define the section map that is used during linking as follows. An example of placing a program in the internal ROM area is shown below. For details of the coding, refer to the manual for the language processor.

```
{
      .intvct        0x80        :
      .text          0x1000      :
      .data          align(0x10) :
            .
            .
            .
}
```

First define the section of the program to be placed in the vector.

To use more than one interrupt, define one section if the vectors are contiguous (the interrupt vector boundaries must match). If the vectors are not contiguous, define a section for each interrupt, and specify all the sections in the section map.
In this way, a specific location for the alternate vector area can be rewritten when the program is downloaded, and no code is necessary to rewrite the interrupt vector.

9.4. RESTRICTIONS/NOTES ON BREAKPOINTS

Note that the following restrictions and notes must be observed when a breakpoint is set or executed (single step) in an interrupt handling routine.

1) During a break, all maskable interrupts are rejected.

2) The single step function sets a temporary breakpoint in the next instruction. As a result, even though the user program in the EI (enable interrupt) status is executed on a single-step basis, execution can branch to the interrupt handler for handling of an interrupt while one instruction is being executed. Be sure, therefore, to observe the points noted regarding breakpoints during single-step execution.

3) Exiting from the interrupt handling routine by single stepping is impossible. (Specifically, single stepping based on the last "}" of the interrupt handling routine is disabled.) Similarly, reti instruction single stepping is impossible. The Return function of the debugger does not support return from an interrupt handling routine to the original routine.

10. CPU PIN CONNECTION

This chapter explains the uses of the CPU pins in the RTE-V850E/MA1-CB.

10.1. PIN CONNECTION LIST

The table below lists the uses of the main CPU pins. Details are given in the subsequent sections.

Pin name	Use	Section reference
PDL0-15/D0-15	Used as the system data bus.	
PAL0-15/A0-15,PAH0-9/A16-25	Used as the system address bus.	
PCT4/RD- PCT7/BCYST-	Used as the system bus control signals.	
PCT0/LCAS-/LWR-/LDQM PCT1/UCAS-/UWR-/UDQM	Used by the system bus and SDRAM.	
PCT5/WE- PCT6/OE- PCD0/SDCKE PCD1/SDCLK PCD2/LBE-/SDCAS- PCD3/UBE-/SDRAS-	Used by SDRAM.	
PCS0/CS0- PCS1/CS1-/RAS1- PCS2/CS2- PCS3/CS3-/RAS3-	Used by system bus as CSx-.	
PCM0/WAIT-	Used as WAIT-. Pulled up at 1K.	10.3
PCM5/SELFREF	Functions as SELFREF. Pulled down at 47K.	10.4
P20/NMI P1/INTP000/TI000	Used as interrupt line.	10.5
P2/INTP001/TI001 P11/INTP010/TI010 P12/INTP011/TI011 P21/INTP020/TI020	Reserved as interrupt line (can be disconnected by SW8).	10.6
P40/SO0/TXD0	Reserved as SIO2-TXD (can be disconnected by SW5).	10.7
P41/SI0/RXD0	Reserved as SIO2-RXD (can be disconnected by SW5).	10.8
P42/SCK0- P43/SO1/TXD1 P44/SI1/RXD1 P45/SCK1-	Reserved as port to control SIO2 (can be disconnected by SW5).	10.9
P4/INTP100/DMARQ0- P5/INTP101/DMARQ1- P6/INTP102/DMARQ2- P7/INTP103/DMARQ3-	Reserved as DMARQ. Connected to GBUS-DMARQ0-3 (can be disconnected by SW6).	10.10
PBD0/DMAAK0- PBD1/DMAAK1- PBD2/DMAAK2- PBD3/DMAAK3-	Reserved as DMAAK. Connected to GBUS-DMAAK0-3 (can be disconnected by SW6).	10.11
P24/INTP110/TC0- P25/INTP111/TC1- P26/INTP112/TC2- P27/INTP113/TC3-	Connected to reserve pin of GBUS (can be disconnected with SW7).	10.12

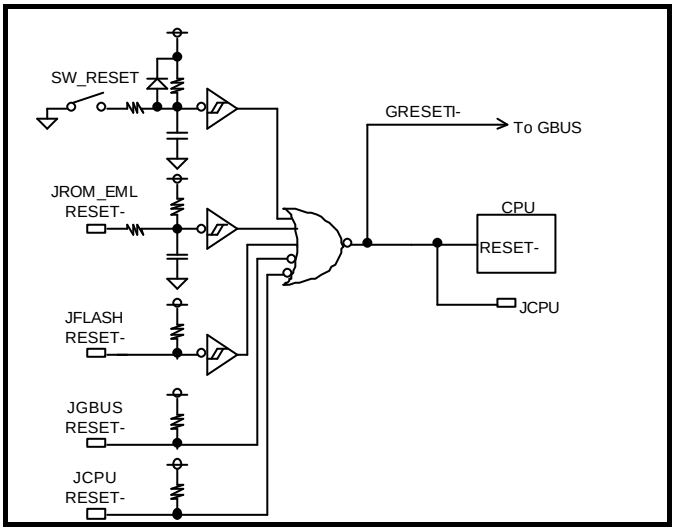
Pin name	Use	Section reference
P22/INTP021/TI021	Used as port to turn ON/OFF VPP for self-writing of flash.	
P70-77,ANI0-7	Not used. Pulled down at 470 kΩ.	10.13
Others	Not used. Pulled up at 47 kΩ.	10.13
RESET-	Inputs RESET.	10.2
MODE0-1,CKSEL	Can be set in any way by SW4.	5.6
VPP	Connected to JFLASH and on-board regulator.	

10.2. RESET-

The factors listed below trigger a CPU reset. These factors reset the CPU. They also system-reset the board.

- Power-on reset: Occurs when the power to the board is switched on.
- Reset request received from JROM_EM: Reset by input from the RESET- pin of the JROMEM connector. (See Section 5.9.)
- Reset by the SW_RESET: Generated by the reset switch (SW_RESET) on the rear panel. (See Section 5.1.)
- Reset from JGBUS: Reset signal from the board connected to JGBUS.
- Reset from JCPU: Reset signal from the board connected to JCPU.
- Reset from JFLASH: Reset signal from the flash programmer.

The figure below outlines the reset signal generation logic.



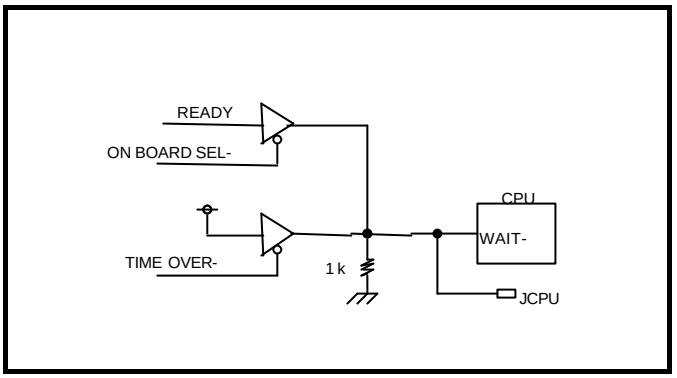
10.3. PCM0/WAIT0-

The PCM0/WAIT0- pin drives a READY signal if a resource in the board is accessed.

A time-over ready occurs if the bus cycle does not close after a specific time or if a space allocated to GBUS is accessed when the GMOTHER_DETECT- signal of GBUS is high (when a board is not connected to GBUS).

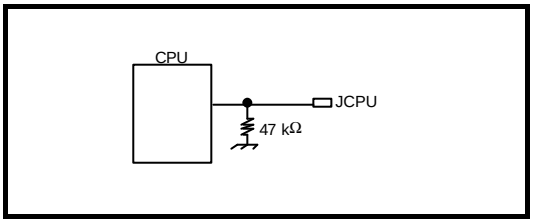
If time-over ready occurs, TOVER_LED on the board lights and an interrupt is issued to the PIC. TOVER_LED remains lit until a time-over ready LED clear pulse is generated by software or the board is reset (see Section 7.4.5).

The configuration of the READY drive block is shown below.



10.4. PCM5/SELFREF

The PCM5/SELFREF pin functions as a SELFREF pin when an external bus is used, and is pulled down at 47 kΩ on board.



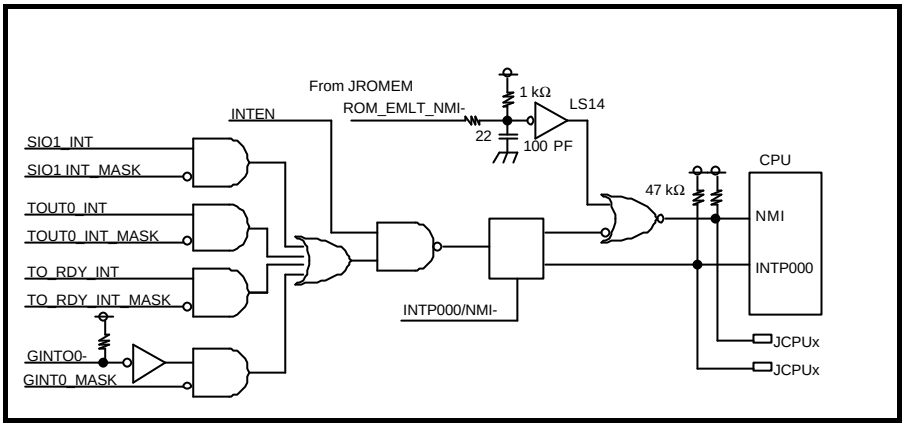
10.5. P20/NMI, P1/INTP000

P20/NMI and P1/INTP000 are used as interrupts.

NMI and INTP000 are interrupts to the monitor, and the following interrupt sources are combined by hardware via the PIC. For how to select an interrupt, see Section 7.4.6.

- UART_INT: Interrupt issued by UART of the TL16C550C (See Section 7.4.7.)
- TOUT0_INT: Interrupt issued by TOUT of CH#0 of the TIC (*m*PD71054) (See Section 7.4.8.)
- TO_RDY_INT: Interrupt resulting from time-over ready occurrence (See Section 10.3.)
- GINT0-: Interrupt from GINT0- of GBUS (See Chapter 14.)

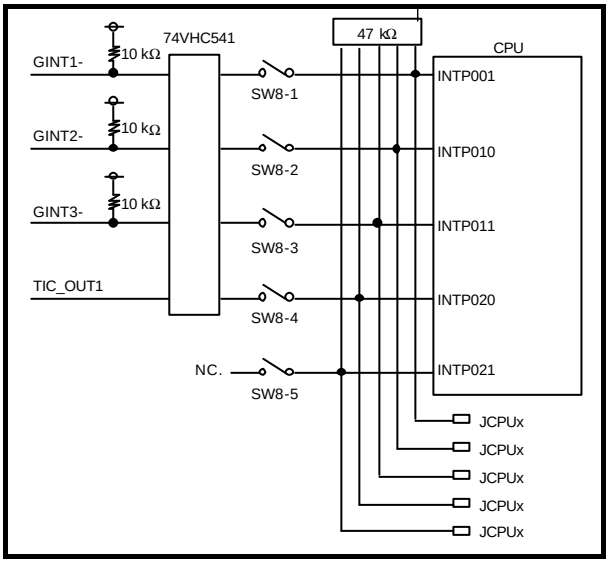
The concept of the NMI1 generation logic is shown below. The xxx_MASK signal in this figure indicates setting of the registers of the PIC (see Section 7.4.6).



10.6. P2/INTP001, P11/INTP010, P12/INTP011, P21/INTP020

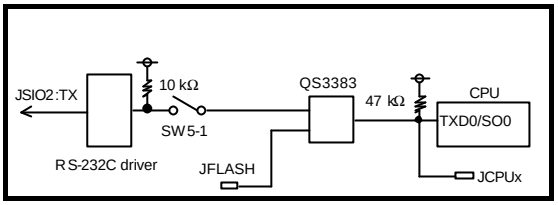
P2/INTP001, P11/INTP010, P12/INTP011, and P21/INTP020 are used as an interrupt, and connected to GBUS-INT0, 1, 2, and TIC_OUT1 (output of timer CH1) via a switch.

Connection of each pin is shown below.



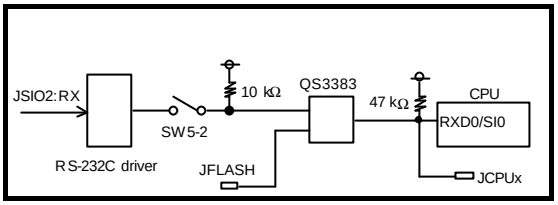
10.7. P40/SO0/TXD0

The P40/SO0/TXD0 pin is used for Tx of SIO2 via an RS-232C transmitter driver, as shown below.



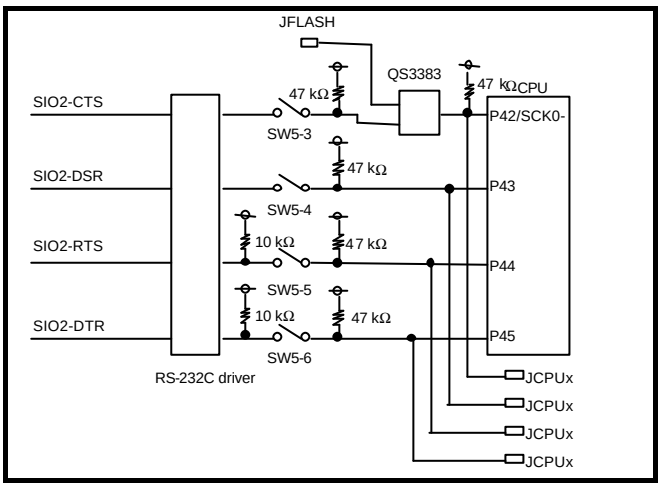
10.8. P41/SI0/RXD0

A signal that is Rx of SIO2 converted to TTL level by an RS-232C receiver driver is connected to the P41/SI0/RXD0 pin via a switch, as shown below.



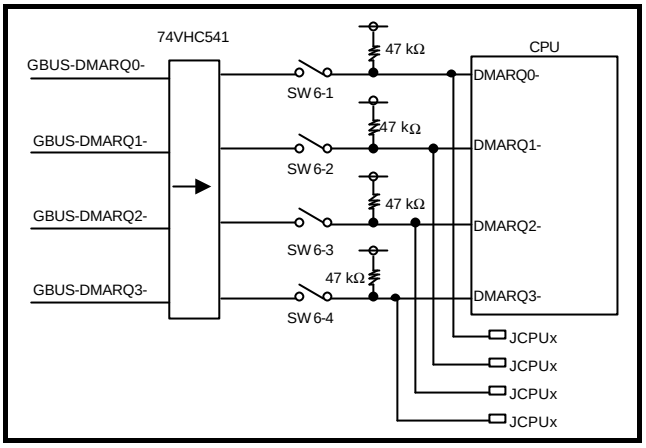
10.9. P42/SCK0-, P43/SO1/TXD1, P44/SI1/TXD0, P45/SCK1-

These pins are used for CTS, DTR, RTS, and DTR of SIO2 via an RS-232C transmitter/receiver driver, as shown below.



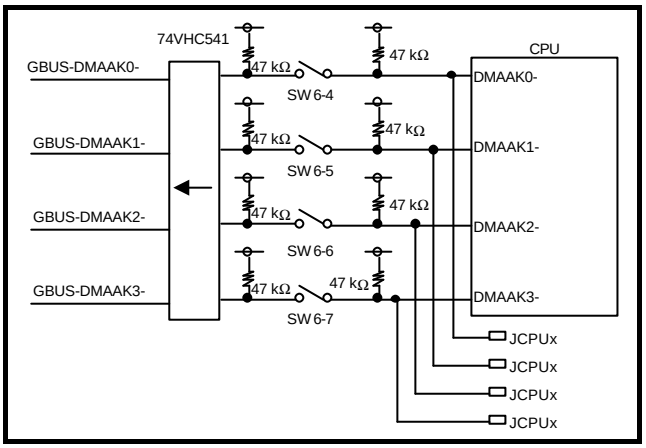
10.10. P4/INTP100/DMARQ0-..P7/INTP103/DMARQ3-

These signal pins connect the DMARQ request from GBUS via a switch, as shown below.



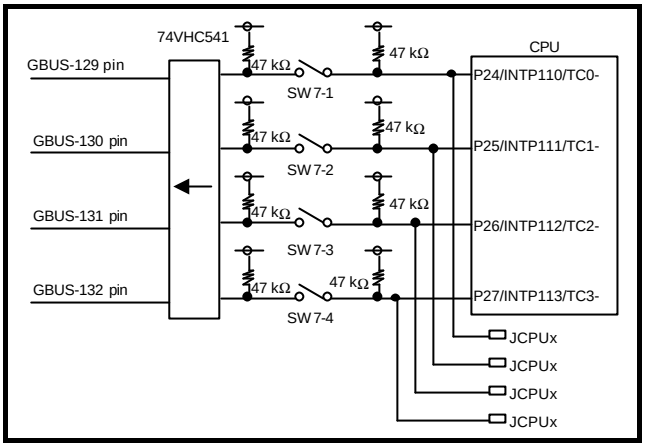
10.11. PBD0/DMAAK0-..PBD3/DMAAK3-

These signal pins connect the signal output by the CPU via switch to DMAAK of GBUS, as shown below.



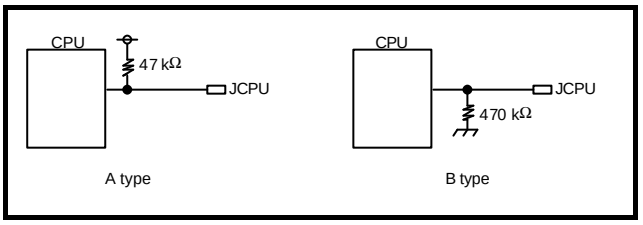
10.12. P24/INTP110/TC0-..P27/INTP113/TC3-

These signal pins invert the logic of a signal output by the CPU and are connected to a reserve pin of GBUS via a switch, as shown below.



10.13. OTHER SIGNALS

Board signals not used are connected to the JCPU connector as shown below.



11. SPECIFIC GBUS SPECIFICATIONS

This chapter explains how GBUS is used with the RTE-V850E/MA1-CB. For the general GBUS specifications, see Chapter 14.

11.1. GENERAL

The following table shows the GBUS signal lines used by the RTE-V850E/MA1-CB.

GBUS signal name	Function	See
GADDR[31:2]	Used as address lines. GADDR[26:31] are not connected. GADDR[25:24] are don't care.	
GDATA[31:0]	Used as data lines. In a read cycle, the signal that is latched on the rising edge of VBCLK is supplied to the CPU.	
GCS-[6:0]	Created by the board and output as chip select lines.	
GCLK	An asynchronous 33-MHz clock is connected to BUSCLK of the CPU.	
GRESETI-	Outputs the reset request generated by the board.	
GRESETO-	Used as reset request from GBUS.	
GADS-, GREADY-, GBLAST-, GW/R-	Used as bus control signals.	
GWAITI-	Not connected.	
GBTERM-	Not connected.	
GRD-, GWR-	RD- and WR- signals generated from the GBUS control signals are connected.	
GHOLD-, GHLDA-	Not connected.	
GBREQ-	Not connected.	
GDMARQ-[3:0]	Used as DMA request signal.	10.10
GDMAAK-[3:0]	Used as DMA acknowledge signal.	10.11
GINTO-[3:0]	Used as interrupt request signal.	10.6
GINTI-[1:0]	OUT0 and OUT1 of TIC (μPD71054) are connected to GINTI0- and GINTI1-.	10.6
GETC[7:0]	Not connected.	
GAHI_EN-	Not connected.	
GMOTHER_DETECT-	Used by time-over ready generation circuit.	
GUSE_DIRECT_ACC-	Not connected.	
GCLK_LOW-	Not connected.	
GLOCK-[1:0]	Not connected.	

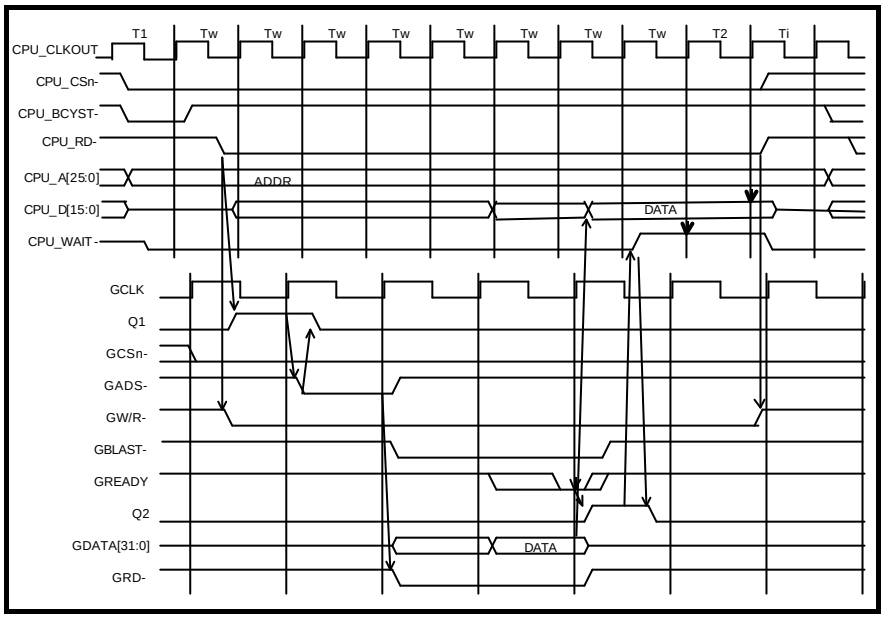
11.2. BUS CYCLE

A 33-MHz clock that is asynchronous with CLK of the CPU is connected to GCLK of GBUS. In addition, because GAHI_EN- is not connected, GADDR[26:31] are not connected. GADDR[24:25] are always [0, 0].

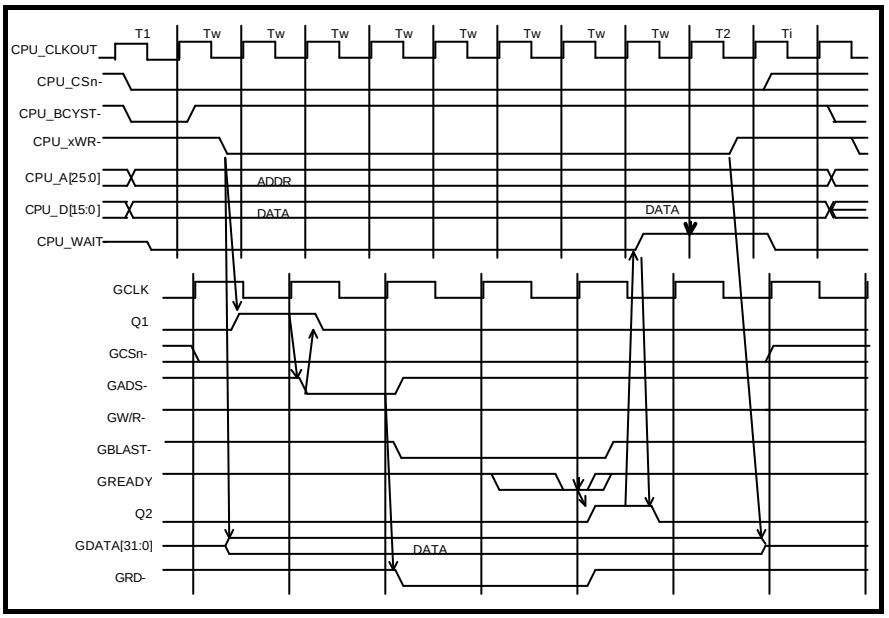
A read cycle from GBUS can be executed on GBUS without a wait cycle.

The CPU_xxx signal shown in the timing charts below is a CPU signal. The Gxxx signal is a GBUS signal.

The following chart shows a read cycle.



The following chart shows a write cycle.



11.3. CHIP SELECT

On the board, the following spaces are allocated to GBUS chip select. In all the following spaces, set the internal bus configuration register of the CPU to 16-bit data bus width and 0 or 1 wait cycle (see Section 7.2.1).

GBUS signal name	CPU address space	Physical address range	RTE-MB-A resources
GCS0-	Space of ADDR[25,24] = [0,1] of CS3 space	5000000 to 5FFFFFF	Common SRAM (2M)
GCS1-	Space of ADDR[25,24] = [1,0], ADDR[23] = 0 of CS3 space Space of CS0 if SW2-1 (FBOOT) is ON	6000000 to 67FFFFF 0000000 to 03FFFFF	Flash ROM (8M)
GCS2-	Space of ADDR[25,24] = [1,1], ADDR[23..19] = [10010] of CS3 space	7900000 to 797FFFF	I/O register
GCS3-	Space of ADDR[25,24] = [1,0], ADDR[23] = 1 of CS3 space	6800000 to 6FFFFFF	EXT-bus: memory space
GCS4-	Space of ADDR[25,24] = [1,1], ADR[23..21] = [101] of CS3 space	7A00000 to 7BFFFFF	EXT-Bus: I/O space
GCS5-	Space of ADDR[25,24] = [0,0] of CS3 space	4000000 to 4FFFFFF	PCI bus space
GCS6-	Space of ADDR[25,24] = [1,1], ADDR[23..19] = [10011] of CS3 space	7980000 to 798FFFF	PCI-Cont register

12. APPENDIX A Multi MONITOR

This chapter describes how to make the settings required to establish a connection between the Multi monitor stored in ROM and the Multi debugger on the host. It also provides notes on the use of the Multi monitor.

12.1. BOARD SETTING

12.1.1. RTE for Win 32 Installation

When the board is used with the Multi debugger, communication software called RTE for Win32 must be installed in the PC. Refer to the RTE for Win32 Installation Manual (supplied with this product) for installation and test methods.

12.1.2. SW1 Setting

SW1 is a switch for general-purpose input ports. For the Multi monitor in the factory-installed ROM, SW1 is used as shown below.

SW1	1	2	Baud rate
Setting	ON	ON	115,200 baud
	OFF	ON	38,400 baud
	ON	OFF	19,200 baud
	OFF	OFF	9,600 baud (Factory-set)

Baud Rate Setting

SW1	3	4	Profiler period
Setting	ON	ON	Timer is not used.
	OFF	ON	200 Hz 5 ms
	ON	OFF	100 Hz 10 ms
	OFF	OFF	60 Hz 16.67 ms (Factory-set)

Profiler Period Setting

SW1	8	Debugger mode
Setting	ON	Monitor is started in test mode.
	OFF	Normal use state (Factory-set)

Debug Mode Setting

SW1-5 to SW1-7 are not used with the Multi monitor.

If SW1-8 is set to ON, it takes some time to start the monitor. The monitor also lights the LED. Usually, keep this switch set to OFF.

12.1.3. Connection of Board

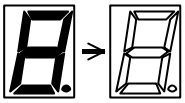
Connect the board to the PC serially, by referring to Chapter 6.

12.2. Multi MONITOR

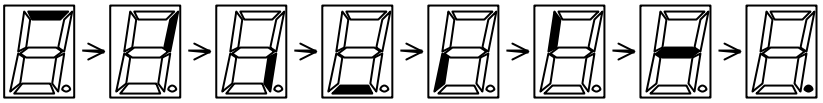
12.2.1. 7-Segment LED on Startup

The 7-segment LED of the ROM monitor for Multi operates as follows when power is supplied to the board (black indicates the segment that lights).

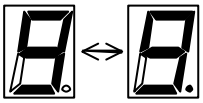
1) Check operation of 7-segment LED (See figure below.)
   If SW1-8 is OFF:



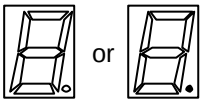
   If SW1-8 is ON:



2) Number counting by simple SRAM memory check
   * Not executed if SW1-8 is OFF.
3) Connection wait status (The dot does not blink if the profiler timer is stopped.)



4) Connection status (The status of the dot is retained on connection.)



12.2.2. ROM Monitor Work RAM

The ROM monitor uses the first 32K-byte area (4F8000H to 4FFFFFH) in the SRAM as work RAM. In other words, user programs are not allowed to use this area or its image area.

12.2.3. Monitor Interrupt

The interrupt selected by SW2-5 is used for monitor communication, the timer, and forced break.

12.2.4. _INIT_SP Setting

_INIT_SP (stack pointer initial value) is set to 4F7FF0H (immediately before monitor work RAM) by the monitor. (_INIT_SP can be changed in the Multi environment.) The monitor uses a 32-byte stack area set by the user program.

12.2.5. Timer Interrupt

If the timer interrupt is disabled, the profiler function of Multi cannot be used (for how to set the timer interrupt, see Section 12.1.2).

12.2.6. Initializing Hardware

The ROM monitor performs initialization so that the resources on the board can be directly accessed.

12.2.7. Special Instruction

The monitor uses the following instruction for the single step, breakpoint, and system call functions.

BRKTRAP instruction (0xnn40)

Do not use a code that may be interpreted as a break instruction in the user program.

12.3. RTE COMMANDS

When the monitor and server are connected, the TARGET window is opened. The RTE commands can be issued in this window. The following table lists the RTE commands.

Command	Description
HELP, ?	Displays help messages.
INIT	Initializes.
VER	Displays the version number.
SFR	Displays or sets the internal I/O.

RTE Commands

Some commands require parameters. All numeric parameters such as addresses and data are assumed to be hexadecimal numbers. The following numeric representations are invalid:

0x1234 1234H $1234

12.3.1. HELP(?)

<Format> HELP [command-name]

Displays a list of RTE commands and their formats. A question mark (?) can also be used in place of the character string HELP. If no command name is specified in the parameter part, the HELP command lists all usable commands.

<Example> HELP SFR

Displays help messages for the SFR command.

12.3.2. INIT

<Format> INIT

Initializes the RTE environment. Usually, this command should not be used.

12.3.3. VER

<Format> VER

Displays the version number of the current RTE environment.

12.3.4. SFR Command

<Format> SFR [register-name [=data]]

When a register name is specified with data omitted, the data read from the register is displayed. When a register name is specified, and data is specified after =, the data is written to the register. The size of data is automatically determined according to the valid size of the specified register. For details of the internal I/O registers, refer to the manual provided with the V850E/MA1 CPU.

<Example 1> SFR

A list of registers is displayed.

<Example 2> SFR IMR

The contents of the IMR register are displayed.

<Example 3> SFR IMR=55AA

Data 55AAH is written into the IMR register.

13. APPENDIX B PARTNER MONITOR

This chapter describes how to make the settings required to establish a connection between the PARTNER monitor stored in ROM and the PARTNER on the host. It also provides notes on the use of the PARTNER monitor.

13.1. BOARD SETTING

13.1.1. SW1 Setting

SW1 is a switch for general-purpose input ports. For the PARTNER monitor in the factory-installed ROM, SW1 is used as shown below.

SW1	1	2	Baud rate
Setting	ON	ON	115,200 baud
	OFF	ON	38,400 baud
	ON	OFF	19,200 baud
	OFF	OFF	9,600 baud (Factory-set)

Baud Rate Setting

SW1	3	4	Timer
Setting	ON	ON	Always use this switch in this status.

SW1	8	Debugger mode
Setting	ON	Monitor is started in test mode.
	OFF	Normal use state (Factory-set)

Debug Mode Setting

SW1-5 to SW1-7 are not used with the PARTNER monitor.

If SW1-8 is set to ON, it takes some time to start the monitor. The monitor also lights the LED. Usually, keep this switch set to OFF.

13.1.2. Connection of Board

Connect the board to the PC serially, by referring to Chapter 6.

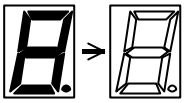
13.2. PARTNER MONITOR

13.2.1. 7-Segment LED on Startup

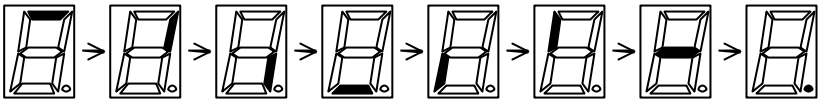
The 7-segment LED of the ROM monitor for PARTNER operates as follows when power is supplied to the board (black indicates the segment that lights).

1) Check operation of 7-segment LED (See figure below.)

   If SW1-8 is OFF:



   If SW1-8 is ON:

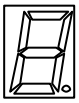


2) Number counting by simple RAM memory check

   * Not executed if SW1-8 is OFF.

3) Connection wait status

4) Connection status



13.2.2. ROM Monitor Work RAM

The ROM monitor uses the first 32K-byte area (4F8000H to 4FFFFFH) in the SRAM as work RAM. In other words, user programs are not allowed to use this area or its image area.

13.2.3. Monitor Interrupt

The interrupt selected by SW2-5 is used for monitor communication and forced break (ESC button).

13.2.4. SP Setting

The stack pointer initial value is set to 4F-7FF0H by the monitor. The monitor uses a 32-byte stack area set by the user program.

13.2.5. Initializing Hardware

The ROM monitor performs initialization so that the resources on the board can be directly accessed.

13.2.6. Special Instruction

The monitor uses the following instruction for the single step, breakpoint, and system call functions.

BRKTRAP instruction (0xnn40)

Do not use a code that may be interpreted as a break instruction in the user program.

14. APPENDIX C GBUS COMMON SPECIFICATIONS

This appendix explains the GBUS specifications that are not dependent on the type of board.

14.1. TERMINOLOGY

Terminology used in this appendix is explained below.

14.1.1. CPU Board and Motherboard

A board in the RTE-CB series is called a CPU board and a Midas lab board connected to GBUS of the CPU board is called a motherboard.

14.1.2. Bus Cycle and Micro Cycle

GBUS is a general bus that can be accessed in burst mode.

A bus cycle consists of a series of cycles, including a one in which a burst access occurs, that is completed (asserting of GADS- is necessary to mark the end of a bus cycle).

Bus cycles are classified into single cycles and a burst cycles. A single cycle is a bus cycle in which data transfer occurs only once. A burst cycle is a bus cycle in which data transfer occurs two or more times.

One cycle for each data transfer in a burst cycle is called a micro cycle.

14.2. SIGNALS

The GBUS signals are listed below. The input/output direction of each GBUS signal is indicated as viewed from the motherboard. Therefore, "input" means that a signal output from the CPU board is input to the motherboard (this also applies to signal names).

"Bidirectional" signals change direction depending on the status of the bus cycle.

"Input/output" signals also change direction depending on whether the bus master is the CPU board or motherboard. The direction written first is the signal direction when the CPU board is the bus master, and the direction written later is the signal direction when the motherboard is the bus master.

A GBUS signal is a +5-V TTL level signal. The motherboard is always little endian.

Signal name	Input/output	Function
GCLK	Input	• Synchronization clock of GBUS. The maximum frequency is 33.33 MHz, and the minimum frequency is 10.0 MHz. GBUS operates synchronized with the rising edge of this clock. • Since, on the motherboard, this clock is terminated at 330 Ω with respect to +5V and GND, the circuit on the CPU board must be able to drive this resistance. • If GCLK is less than 16.67 MHz, GCLK_LOW- goes low. In this way, the motherboard can adjust the number of wait cycles. • Because a PLL (Phase Lock Loop) zero delay buffer may be used, if the frequency of GCLK is changed, the motherboard must not be accessed for at least 1 ms after the frequency has been changed to allow the PLL to be locked.
GRESETI-	Input	• Reset signal of GBUS. If a reset occurs on the CPU board, this signal goes low. The motherboard is reset by this signal (the motherboard can also be reset for other causes on the motherboard).
GRESETO-	Output	• This signal goes low if the motherboard is reset. • The motherboard ORs the reset signal on the motherboard with GRESETI- as GRESETO-. Accordingly, the CPU board resets the circuits on the CPU board by ORing GRESETI- and GRESETO- (GRESETI- and GRESETO- are ORed because there is a possibility that the motherboard is not connected).

Signal name	Input/output	Function
GADDR[31:2]	Input/output	• Address signals of GBUS. These signals are driven by a valid value during a cycle. • GADDR[31] is ignored on the motherboard if the CPU is the bus master. • The low-order addresses A1 and A0 use a byte enable signal. • GADDR[31:26] from the CPU board can be treated as 0 by using the GAHI_EN- signal. • If the bus master is the motherboard and if GADDR[25] is 0, the resources on the motherboard are selected; if GADDR[25] is 1, the resources on the CPU board is selected.
GBEN-[3:0]	Input/output	• Byte enable signals of GBUS. These signals are always driven by a valid value during a cycle. • GBEN0-, GBEN1-, GBEN2-, and GBEN3- correspond to byte lanes GDATA[7:0], GDATA[15:8], GDATA[23:16], and GDATA[31:24], and the corresponding byte lane is valid if GBENx- is low.
GDATA[31:0]	Bidirectional	• Bus data signals of GBUS. • These signals are pulled up to 10 kΩ on the motherboard. • The direction of these signals is determined by GW/R-.
GADS-	Input/output	• Address strobe signal of GBUS. If this signal is sampled low on the rising edge of GCLK, the start of a bus cycle is indicated. • The motherboard ignores GADS- if none of the chip select signals (GCS-[7:0]) is active.
GREADY-	Output/input	• Ready signal of GBUS. If this signal is sampled low and GWAITI is sampled high on the rising edge of GCLK during a micro cycle, the end of the micro cycle is indicated. • Time-over ready when the CPU board accesses the motherboard is generated by the motherboard. The reason is to avoid collision with the GREADY- signal.
GWAITI-	Input	• Wait request signal. This signal is sampled on the rising edge of GCLK. • If the CPU board cannot support a cycle with a few wait cycles, the CPU board samples GWAITI- low at the sample timing of GREADY- so that the motherboard cannot handle GREADY- as a ready signal even though it is low at the time. Usually, this signal is used if the CPU board cannot support zero wait burst (see Section 14.6.3). • This signal is valid only in a cycle in which the CPU board is the bus master.
GBLAST-	Input/output	• Bus cycle completion notification signal. This signal is sampled on the rising edge of GCLK. • This signal is asserted low by the bus master when a micro cycle that completes the bus cycle starts. • The bus cycle is completed if the low level of GBLAST-, low level of GREADY-, and high level of GWAITI- are sampled on the rising edge of GCLK.
GBTERM-	Output/input	• Bus cycle completion request signal. This signal is sampled on the rising edge of GCLK. • If the accessed side requests completion of the bus cycle, the GREADY- and GBTERM- signals go low. If the bus master samples GBTERM- as low when it samples GREADY- as low, it must complete the bus cycle even though GBLAST- has not been asserted, and start the bus cycle again by asserting GADS- again. GBTERM- must be asserted at the same time as GREADY-. • This signal is used to complete the bus cycle if the accessed side does not support burst cycles or if a burst cycle exceeding the supported number of bursts is requested.

Signal name	Input/output	Function
GW/R-	Input/output	• Write/Read signal. This signal indicates the direction of the data bus. It is always driven by a valid value during the bus cycle. • This signal indicates the direction of the data bus for the bus master.
GCS-[7:0]	Input	• Chip select signals. These signals are always driven by a valid value during the bus cycle. • The CPU board makes the corresponding chip select signal active to specify the resources on the motherboard when the CPU board is the bus master. • Each chip select signal specifies the type of memory/I/O space and the width of the space (see Section 14.5).
GRD-	Input	• Read timing signal. This signal is asserted when the CPU board is the bus master. • This signal is not used by the motherboard. • If the CPU has an RD- command signal, that signal is usually connected.
GWR-	Input	• Write timing signal. This signal is asserted when the CPU board is the bus master. • This signal is not used by the motherboard. • If the CPU has a WR- command signal, that signal is usually connected.
GHOLD-	Output	• Bus hold request signal. • This signal is asserted low when the motherboard accesses the resources on the CPU board to acquire bus mastership. • If the GUSE_DIRECT_ACC- signal is high, the GHOLD- signal indicates to the CPU board that the motherboard has no resources that can be accessed. In this case, the CPU board does not have to support GHOLD-.
GHLDA-	Input	• Bus hold acknowledge signal. • This signal indicates that the CPU board releases bus mastership of GBUS to the motherboard. It is then asserted low. • The CPU board that asserts the GUSE_DIRECT_ACC- signal high can disconnect the GHLDA- signal.
GBREQ-	Input	• Bus mastership release request signal • When the motherboard has bus mastership from asserting GHLDA- low, the CPU board asserts GBREQ- low when it requires bus mastership. • If GBREQ- is asserted low and the motherboard is in bus cycle, GBLAST- must be asserted in the next micro cycle, the bus cycle must be completed in the next micro cycle, and GHOLD- must be deasserted. • GBREQ- is used to return bus mastership to the CPU board temporarily if the number of bursts in the bus cycle is large when the motherboard is the bus master, or if a bus cycle with a high priority such as a refresh cycle is pending on the CPU board.
GDMARQ-[3:0]	Output	• DMA request signals. Only two-cycle DMA is supported. Fly-by DMA is not supported. • These signals are asserted low if a DMA request is generated on the motherboard. • The CPU board must support all four DMA signals. The number of DMA signals that can be asserted at the same time and can be supported by the GDMAAK- signal depends on the CPU board. • The CPU board uses the DMAAK signal in preference to DMAAK-[3:2] if correspondence between all four GDMARQ- signals and GDMAAK- signals cannot be established.

Signal name	Input/output	Function
GDMAAK-[3:0]	Input	• DMA acknowledge signals. • These signals are asserted low to acknowledge DMA requests from the motherboard. • The CPU board uses the DMAAK signal in preference to DMAAK-[3:2] if correspondence between all four GDMARQ- signals and GDMAAK- signals cannot be established. • The motherboard is designed to operate even though there is no GDMAAK- signal.
GINTO-[3:0]	Output	• Interrupt request signals. • GINTO0- can be used as a level-sensitive signal. • Whether GINTO-[3:1] can be used as level-sensitive signals or edge-sensitive signals depends on the CPU board (since they may be directly connected to the CPU). The motherboard can support both level- and edge-sensitive signals. • Occurrence of an interrupt is indicated when these signals are low or on the falling edges of these signals.
GINTI-[1:0]	Input	• Interrupt request signals. • These interrupt signals are used to combine an interrupt on the CPU board with an interrupt on the other motherboard and return the combined signal to GINTO-[3:0]. • Usually, OUT0 and OUT1 of TIC (*m*PD71054) on the CPU board are connected. The motherboard can select the type of sensitivity and polarity of these interrupt signals.
GETC[7:0]	--	• CPU board dependent signals. • The contents of GETC[7:0], including the direction and contents of the signals, are determined by the CPU board. The CPU board uses these signals to exchange special signals with the motherboard.
GAHI_EN-	Input	• Upper address valid signal. • If this signal is low and if the CPU board is the bus master, the CPU board drives a valid value on GADDR[31:26]. If this signal is high, the CPU board does not drive a valid signal on GADDR[31:26], and the circuits on the motherboard perform processing with all of GADDR[31:26] low.
GMOTHER_ DETECT-	Output	• Motherboard detection signal. • This signal is pulled up on the CPU board, and is connected to GND on the motherboard. The CPU board uses this signal when it must determine if the motherboard is connected (for example, time-over ready generation circuit of the CPU board).
GUSE_ DIRECT_ACC-	Input	• If this signal is low, the CPU board has resources that can be accessed by the motherboard.
GCLK_LOW-	Input	• If this signal is low, the frequency of GCLK is 16.67 MHz or less. If it is high, the frequency of GCLK is 16.67 to 33.33 MHz. • The circuits on the motherboard use this signal to determine the number of wait cycles required for accessing the resources on the motherboard.
GBLOCK-[1:0]	Input	• Bus lock signals. These signals must be valid during a bus cycle and for bus cycles that must be locked. • If a bus lock signal is output by the CPU, the bus lock signal is connected to the motherboard using these pins. • The GBLOCK0- signal is valid for the GCS0- space. GBLOCK1- is valid for the GCS5- and GCS7- spaces.
+5V	Output	• Power supply. Supplies +5 V ±5% from the motherboard to the CPU board.

Signal name	Input/output	Function
+12V	Output	• Power supply. Supplies +12 V ±10% from the motherboard to the CPU board. However, if the CPU board does not require +12 V, the motherboard does not have to supply +12 V.

14.3. PIN ASSIGNMENTS

The following table shows the GBUS pin assignments. Reserve indicates a reserved pin. N/C indicates that a pin is not connected.

No.	Signal name	No.	Signal name	No.	Signal name	No.	Signal name
1	+12V	2	+12V	3	GND	4	+5V
5	GADDR2	6	GADDR3	7	GADDR4	8	GADDR5
9	GADDR6	10	GADDR7	11	GND	12	+5V
13	GADDR8	14	GADDR9	15	GADDR10	16	GADDR11
17	GADDR12	18	GADDR13	19	GADDR14	20	GADDR15
21	GND	22	+5V	23	GADDR16	24	GADDR17
25	GADDR18	26	GADDR19	27	GADDR20	28	GADDR21
29	GADDR22	30	GADDR23	31	GND	32	+5V
33	GADDR24	34	GADDR25	35	GADDR26	36	GADDR27
37	GADDR28	38	GADDR29	39	GADDR30	40	GADDR31
41	GND	42	+5V	43	GBEN3-	44	GBEN2-
45	GBEN1-	46	GBEN0-	47	GND	48	+5V
49	GDATA31	50	GDATA30	51	GDATA29	52	GDATA28
53	GDATA27	54	GDATA26	55	GDATA25	56	GDATA24
57	GND	58	+5V	59	GDATA23	60	GDATA22
61	GDATA21	62	GDATA20	63	GDATA19	64	GDATA18
65	GDATA17	66	GDATA16	67	GND	68	+5V
69	GDATA15	70	GDATA14	71	GDATA13	72	GDATA12
73	GDATA11	74	GDATA10	75	GDATA9	76	GDATA8
77	GND	78	+5V	79	GDATA7	80	GDATA6
81	GDATA5	82	GDATA4	83	GDATA3	84	GDATA2
85	GDATA1	86	GDATA0	87	GND	88	+5V
89	GND	90	GW/R-	91	GBTERM-	92	GREADY-
93	GRESETI-	94	GADS-	95	GBLAST-	96	GWAITI-
97	GND	98	GCLK	99	GND	100	+5V
101	GCS0-	102	GCS1-	103	GCS2-	104	GCS3-
105	GCS4-	106	GCS5-	107	GCS6-	108	GCS7-
109	Reserve	110	Reserve	111	Reserve	112	Reserve
113	GRD-	114	GWR-	115	GND	116	+5V
117	GHOLD-	118	GHLDA-	119	GBREQ-	120	N/C
121	GDMARQ0-	122	GDMARQ1-	123	GDMARQ2-	124	GDMARQ3-
125	GDMAAK0-	126	GDMAAK1-	127	GDMAAK2-	128	GDMAAK3-
129	Reserve	130	Reserve	131	Reserve	132	Reserve
133	GND	134	+5V	135	GINTO0-	136	GINTO1-
137	GINTO2-	138	GINTO3-	139	GINTI0-	140	GINTI1-
141	GETC0	142	GETC1	143	GETC2	144	GETC3
145	GETC4	146	GETC5	147	GETC6	148	GETC7
149	Reserve	150	Reserve	151	GAHI_EN-	152	GMOTHER_DETECT-
153	GND	154	+5V	155	GUSE_DIRECT_ACC-	156	GCLK_LOW-
157	GRESETO-	158	GBLOCK0-	159	GBLOCK1-	160	N/C
161	N/C	162	N/C	163	N/C	164	N/C
165	N/C	166	N/C	167	N/C	168	N/C
169	N/C	170	N/C	171	N/C	172	N/C
173	N/C	174	N/C	175	N/C	176	N/C
177	GND	178	+5V	179	+12V	180	+12V

The following connectors are used:

CPU board side connector → Kell 8817-180-170L
Motherboard side connector (straight) → Kell 8807-180-170S
Motherboard side connector (L angle) → Kell 8807-180-170L

14.4. PROCESSING OF UNUSED PINS

Signals that are not input to the GBUS motherboard are pulled up or down on the motherboard and can be unconnected on the CPU board. Signals that can be unconnected and the processing performed on the motherboard for those pins are shown below.

Signal name	Processing
GADDR[31:26]	• If GADDR[31:26] are not used, GADDR[31:26] can be unconnected by making the GAHI_EN- signal high or by disconnecting it. In this case, if the CPU is the bus master, all the bits of GADDR[31:26] are treated as 0 on the motherboard.
GWAITI-	• Pull-up processing is performed.
GBLAST-	• Pull-up processing is performed.
GBTERM-	• Pull-up processing is performed.
GCS-[7:0]	• Pull-up processing is performed.
GHLDA -	• Pull-up processing is performed.
GBREQ-	• Pull-up processing is performed.
GDMAAK-[3:0]	• Pull-up processing is performed.
GINTI-[1:0]	• Pull-up processing is performed.
GAHI_EN-	• Pull-up processing is performed.
GUSE_DIRECT_ACC-	• Pull-up processing is performed.
GCLK_LOW-	• Pull-up processing is performed.
GBLOCK-[1:0]	• Pull-up processing is performed.

14.5. ALLOCATING GCS-[7:0]

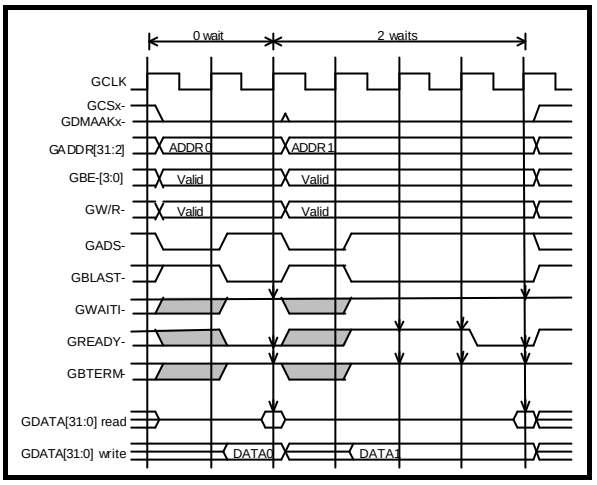
The following table shows the allocation of the chip select signals (GCS-[7:0]). All of the spaces can be accessed in a burst cycle. A space marked I/O under the heading "Recommended space" means that, if the CPU has an I/O space, it is recommended that the space be allocated as an I/O space. "Minimum range" indicates that the CPU board must allocate at least the indicated area for the corresponding chip select space. "Maximum range" indicates that, if the CPU board has an extra address range, addresses can be allocated for the indicated range.

Signal name	Recommended space	Minimum range	Maximum range	Remark
GCS0-	Memory	1M byte		Bus lock possible with GLOCK0-
GCS1-	Memory	2M bytes		Because a flash ROM is allocated to this space on the motherboard, the program must be able to be booted from this space, instead of from UV-EPROM on the CPU board, via a switch.
GCS2-	I/O	64K bytes		
GCS3-	Memory	64Kbytes	16M bytes	
GCS4-	I/O	64K bytes	16M bytes	
GCS5-	Memory	1M byte	2G bytes	Bus lock possible with GLOCK1-
GCS6-	I/O	512 bytes		
GCS7-	I/O	64K bytes	2G bytes	Bus lock possible with GLOCK1-

14.6. BUS CYCLE

14.6.1. Single Cycle

The following chart shows the single cycle when GBWAITI- and GBTERM- are always inactive and the CPU board is the bus master. If the motherboard is the bus master, the GCSx-, GDMAAK-, and GWAITI- signals are not used.

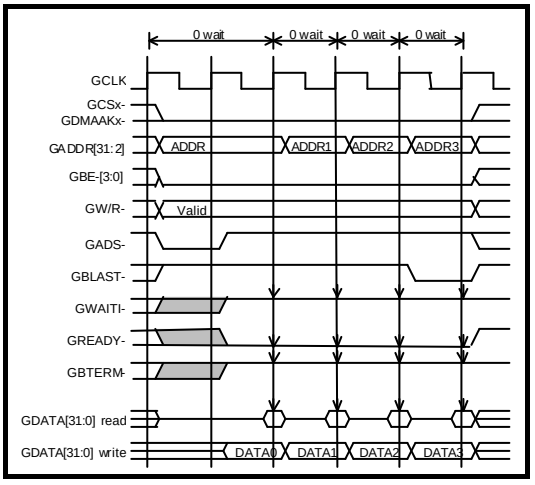


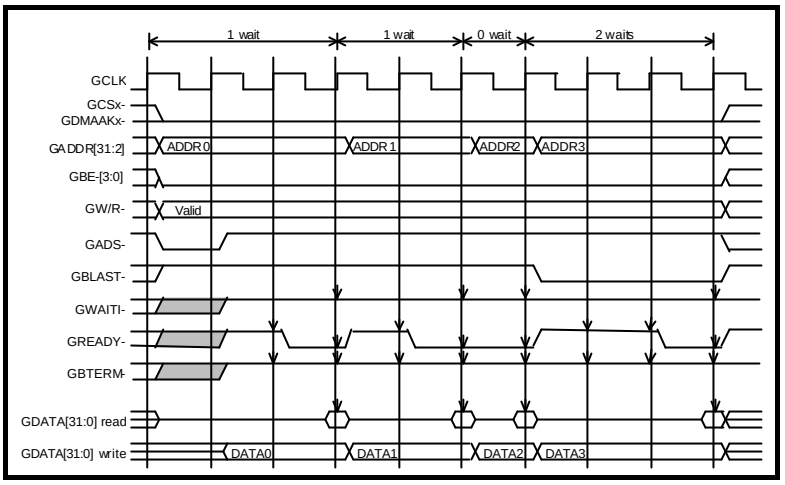
14.6.2. Burst Cycle

The following rules apply to a burst cycle:

- The addresses in the burst cycle can be in any sequence allowed by the GBUS specifications. However, the address sequence may be specified according to what is to be accessed.
- In a burst cycle, all of GBE-[3:0] must be active.
- The number of bursts (the number of micro cycles) is not limited. If the target of the access limits the number of bursts, use the GBTERM- signal (see Section 14.6.4) to request canceling of the burst.

The following charts show the burst cycle when GBWAITI- and GBTERM- are always inactive and the CPU board is the bus master. If the motherboard is the bus master, the GCSx-, GDMAAK-, and GWAITI- signals are not used.



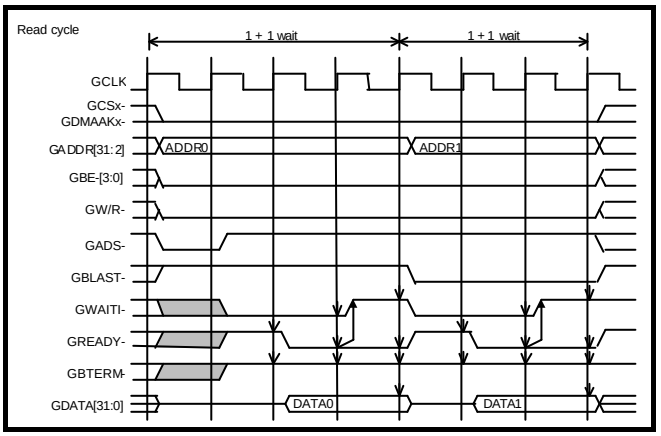


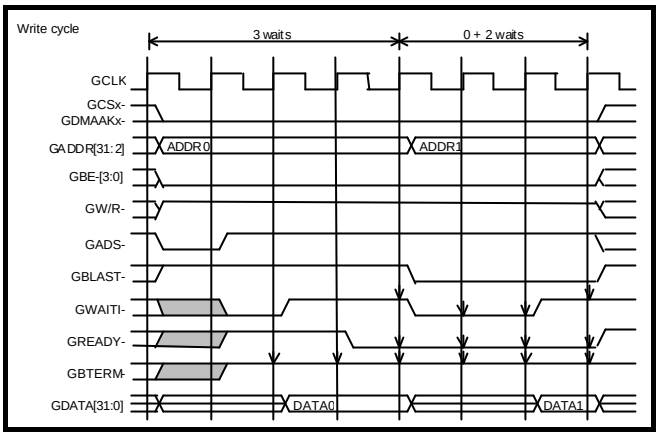
14.6.3. GWAITI-

The GWAITI- signal can be used as follows in a cycle in which the CPU board is the bus master:

- To delay sampling of data by a specific number of clocks because the data cannot be sampled in the read cycle.
- To hold the target of an access by the specific number of clocks because data for the next micro cycle is not ready immediately after completion of the first micro cycle in the burst cycle of a write cycle.

In other words, the roles of the read cycle and write cycle are switched, but GREADY- and GWAITI- serve as data transmission ready and data reception ready signals.
The following charts show that a wait cycle is inserted by the GWAITI- signal.



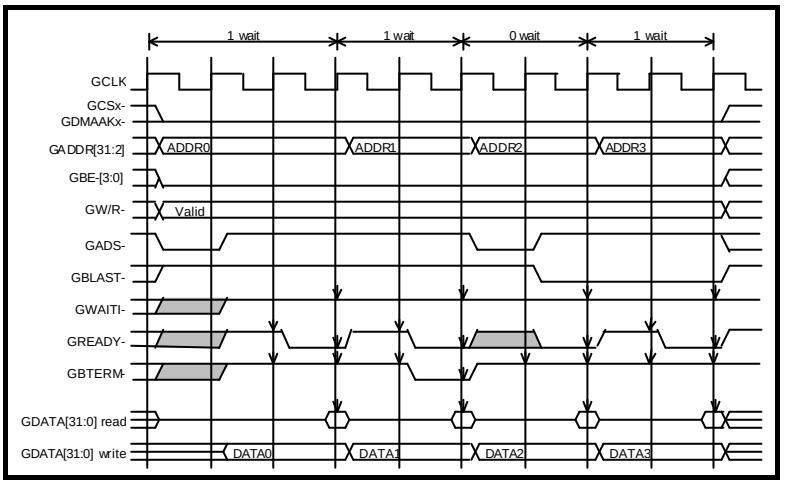


14.6.4. GBTERM-

If both the GBTERM- signal and GREADY- signal become active at the same time, the bus master completes the bus cycle after the current micro cycle ends, and then starts the burst cycle again by asserting GADS- active.

The GBTERM- signal is asserted active if the target of the access does not support burst cycles or accesses are made more than the supported number of bursts. Asserting the GBTERM- signal only without also asserting the GREADY- signal is not allowed.

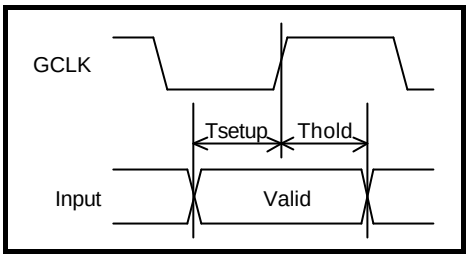
The following chart shows that the burst cycle is canceled by the GBTERM- signal.



14.7. TIMING

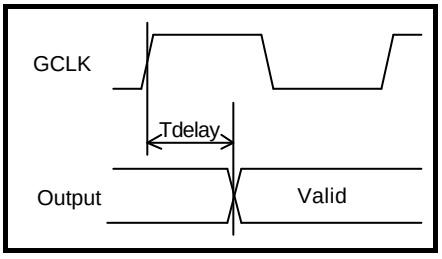
This chapter describes the timing of Midas lab's motherboard. The CPU board is designed to satisfy this timing.

14.7.1. Setup Time



Signal name	Tsetup Min. (ns)	Thold Min. (ns)
GADDR[31:2]	12	0
GBEN-[3:0]	8	0
GDATA[31:0]	7	0
GADS-	14	0
GREADY-	9	1
GWAITI-	14	0
GBLAST-	8	0
GBTERM-	8	1
GW/R-	10	0
GCS-[7:0]	14	0
GBREQ-	15	0
GDMAAK-[3:0]	6	0
GLOCK-[1:0]	12	0

14.7.2. Delay Time



Signal name	Tdelay Max. (ns)
GADDR[31:2]	21
GBEN-[3:0]	17
GDATA[31:0]	21
GADS-	15
GREADY-	15
GBLAST-	17
GBTERM-	16
GW/R-	15

- Memo -