Renesas Electronics America Inc - M30291FATHP#U3AAG4 Datasheet

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	-
Core Size	-
Speed	-
Connectivity	-
Peripherals	-
Number of I/O	-
Program Memory Size	-
Program Memory Type	-
EEPROM Size	-
RAM Size	-
Voltage - Supply (Vcc/Vdd)	-
Data Converters	-
Oscillator Type	-
Operating Temperature	-
Mounting Type	-
Package / Case	-
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/m30291fathp-u3aag4

1.3 Product List

Tables 1.3 to 1.5 list the M16C/29 Group products and **Figure 1.3** shows the type numbers, memory sizes and packages. **Tables 1.6 to 1.8** list the product code of flash memory version for M16C/29 Group. **Figure 1.4 to Figure 1.6** show the marking diagram of flash memory version for M16C/29 Group.

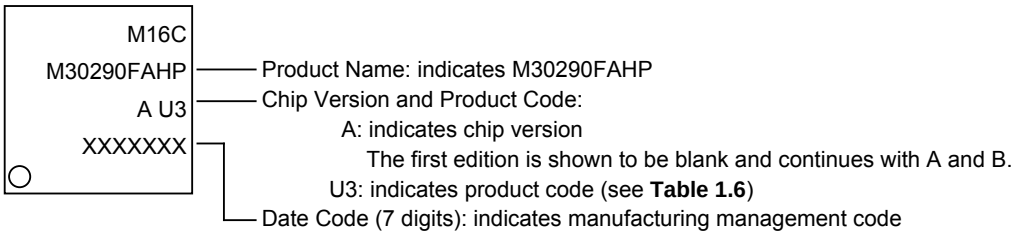
Table 1.3 Product List (1) -Normal Version **As of March, 2007**

Type Number	ROM Capacity	RAM Capacity	Package Type	Remarks	Product Code
M30290FAHP	96 K + 4 K	8 K	PLQP0080KB-A (80P6Q-A)	Flash Memory	U3, U5, U7, U9
M30290FCHP	128 K + 4 K	12 K			
M30291FAHP	96 K + 4 K	8 K	PLQP0064KB-A (64P6Q-A)		
M30291FCHP	128 K + 4 K	12 K			
M30290M8-XXXHP	64 K	4 K	PLQP0080KB-A (80P6Q-A)	Mask ROM	U3, U5
M30290MA-XXXHP	96 K	8 K			
M30290MC-XXXHP	128 K	12 K			
M30291M8-XXXHP	64 K	4 K	PLQP0064KB-A (64P6Q-A)		
M30291MA-XXXHP	96 K	8 K			
M30291MC-XXXHP	128 K	12 K			

Table 1.4 Product List (2) -T Version **As of March, 2007**

Type Number	ROM Capacity	RAM Capacity	Package Type	Remarks	Product Code
M30290FATHP	96 K + 4 K	8 K	PLQP0080KB-A (80P6Q-A)	Flash Memory	U3, U5, U7, U9
M30290FCTHP	128 K + 4 K	12 K			
M30291FATHP	96 K + 4 K	8 K	PLQP0064KB-A (64P6Q-A)		
M30291FCTHP	128 K + 4 K	12 K			
M30290M8T-XXXHP	64 K	4 K	PLQP0080KB-A (80P6Q-A)	Mask ROM	U0
M30290MAT-XXXHP	96 K	8 K			
M30290MCT-XXXHP	128 K	12 K			
M30291M8T-XXXHP	64 K	4 K	PLQP0064KB-A (64P6Q-A)		
M30291MAT-XXXHP	96 K	8 K			
M30291MCT-XXXHP	128 K	12 K			

(1) Flash Memory Version, PLQP0080KB-A (80P6Q-A), Normal-ver.



(2) Flash Memory Version, PLQP0064KB-A (64P6Q-A), Normal-ver.

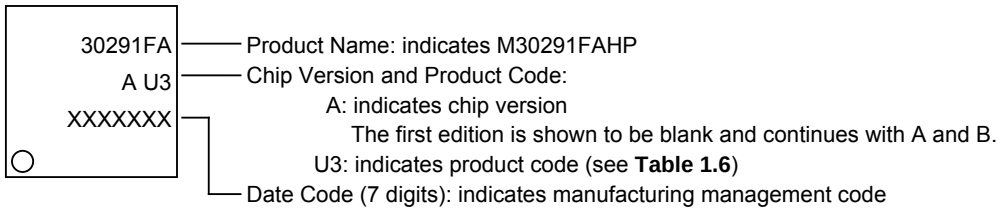
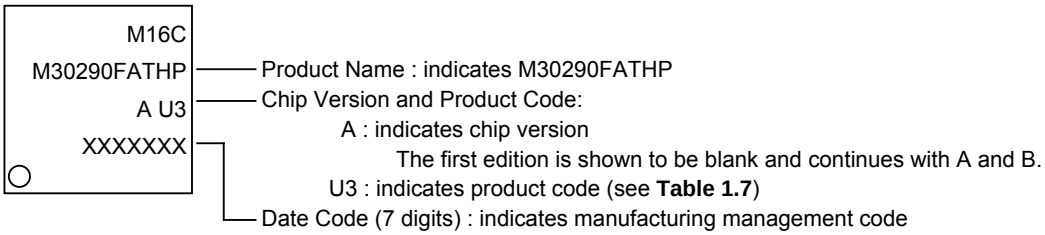


Figure 1.4 Marking Diagrams of Flash Memory Version - M16C/29 Group Normal-ver. (Top View)

(1) Flash Memory Version, PLQP0080KB-A (80P6Q-A), T-ver.



(2) Flash Memory Version, PLQP0064KB-A (64P6Q-A), T-ver.

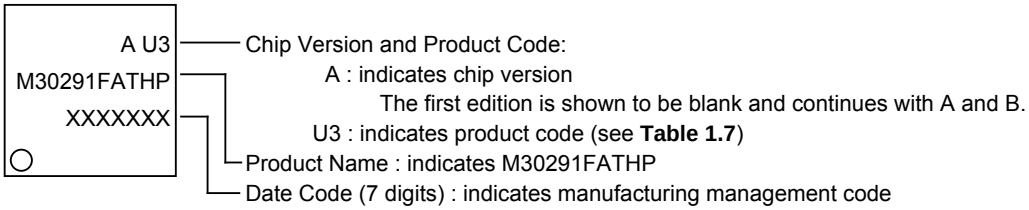


Figure 1.5 Marking Diagrams of Flash Memory Version - M16C/29 Group T-ver. (Top View)

Table 4.2 SFR Information (2)

Address	Register	Symbol	After reset
0040₁₆			
0041₁₆	CAN0 wakeup interrupt control register	C01WKIC	XXXXX000₂
0042₁₆	CAN0 successful reception interrupt control register	C0RECIC	XXXXX000₂
0043₁₆	CAN0 successful transmission interrupt control register	C0TRMIC	XXXXX000₂
0044₁₆	INT3 interrupt control register	INT3IC	XX00X000₂
0045₁₆	ICOC 0 interrupt control register	ICOC0IC	XXXXX000₂
0046₁₆	ICOC 1 interrupt control register, I^2C bus interface interrupt control register 1	ICOC1IC,IICIC	XXXXX000₂
0047₁₆	ICOC base timer interrupt control register, SCL/SDA interrupt control register 2	BTIC,SCLDAIC	XXXXX000₂
0048₁₆	SI/O4 interrupt control register, INT5 interrupt control register	S4IC, INT5IC	XX00X000₂
0049₁₆	SI/O3 interrupt control register, INT4 interrupt control register	S3IC, INT4IC	XX00X000₂
004A₁₆	UART2 Bus collision detection interrupt control register	BCNIC	XXXXX000₂
004B₁₆	DMA0 interrupt control register	DM0IC	XXXXX000₂
004C₁₆	DMA1 interrupt control register	DM1IC	XXXXX000₂
004D₁₆	CAN0 error interrupt control register	C01ERRIC	XXXXX000₂
004E₁₆	A/D conversion interrupt control register, Key input interrupt control register (Note 2)	ADIC, KUPIC	XXXXX000₂
004F₁₆	UART2 transmit interrupt control register	S2TIC	XXXXX000₂
0050₁₆	UART2 receive interrupt control register	S2RIC	XXXXX000₂
0051₁₆	UART0 transmit interrupt control register	S0TIC	XXXXX000₂
0052₁₆	UART0 receive interrupt control register	S0RIC	XXXXX000₂
0053₁₆	UART1 transmit interrupt control register	S1TIC	XXXXX000₂
0054₁₆	UART1 receive interrupt control register	S1RIC	XXXXX000₂
0055₁₆	TimerA0 interrupt control register	TA0IC	XXXXX000₂
0056₁₆	TimerA1 interrupt control register	TA1IC	XXXXX000₂
0057₁₆	TimerA2 interrupt control register	TA2IC	XXXXX000₂
0058₁₆	TimerA3 interrupt control register	TA3IC	XXXXX000₂
0059₁₆	TimerA4 interrupt control register	TA4IC	XXXXX000₂
005A₁₆	TimerB0 interrupt control register	TB0IC	XXXXX000₂
005B₁₆	TimerB1 interrupt control register	TB1IC	XXXXX000₂
005C₁₆	TimerB2 interrupt control register	TB2IC	XXXXX000₂
005D₁₆	INT0 interrupt control register	INT0IC	XX00X000₂
005E₁₆	INT1 interrupt control register	INT1IC	XX00X000₂
005F₁₆	INT2 interrupt control register	INT2IC	XX00X000₂
0060₁₆	CAN0 message box 0: Identifier/DLC		XX₁₆
0061₁₆			XX₁₆
0062₁₆			XX₁₆
0063₁₆			XX₁₆
0064₁₆			XX₁₆
0065₁₆			XX₁₆
0066₁₆	CAN0 message box 0 : Data field		XX₁₆
0067₁₆			XX₁₆
0068₁₆			XX₁₆
0069₁₆			XX₁₆
006A₁₆			XX₁₆
006B₁₆			XX₁₆
006C₁₆			XX₁₆
006D₁₆			XX₁₆
006E₁₆	CAN0 message box 0 : Time stamp		XX₁₆
006F₁₆			XX₁₆
0070₁₆	CAN0 message box 1 : Identifier/DLC		XX₁₆
0071₁₆			XX₁₆
0072₁₆			XX₁₆
0073₁₆			XX₁₆
0074₁₆			XX₁₆
0075₁₆			XX₁₆
0076₁₆	CAN0 message box 1 : Data field		XX₁₆
0077₁₆			XX₁₆
0078₁₆			XX₁₆
0079₁₆			XX₁₆
007A₁₆			XX₁₆
007B₁₆			XX₁₆
007C₁₆			XX₁₆
007D₁₆			XX₁₆
007E₁₆	CAN0 message box 1 : Time stamp		XX₁₆
007F₁₆			XX₁₆

Note 1: The blank areas are reserved and cannot be used by users.
Note 2: A/D conversion interrupt control register is effective when the bit1(Interrupt source select register (address 35Eh IFSR2A) is set to "0". Key input interrupt control register is effective when the bit1 is set to "1".
X : Undefined

9.2 Interrupts and Interrupt Vector

One interrupt vector consists of 4 bytes. Set the start address of each interrupt routine in the respective interrupt vectors. When an interrupt request is accepted, the CPU branches to the address set in the corresponding interrupt vector. **Figure 9.2** shows the interrupt vector.

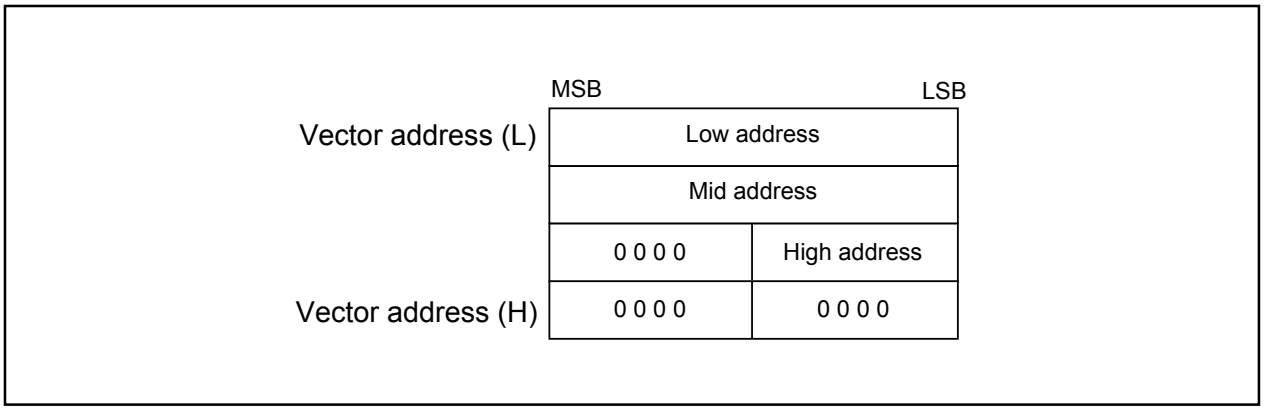


Figure 9.2 Interrupt Vector

9.2.1 Fixed Vector Tables

The fixed vector tables are allocated to the addresses from FFFDC16 to FFFFF16. **Table 9.1** lists the fixed vector tables. In the flash memory version of MCU, the vector addresses (H) of fixed vectors are used by the ID code check function. For details, refer to the section "flash memory rewrite disabling function".

Table 9.1 Fixed Vector Tables

Interrupt source	Vector table addresses Address (L) to address (H)	Remarks	Reference
Undefined instruction	FFFDC16 to FFFDF16	Interrupt on UND instruction	M16C/60, M16C/20 serise software maual
Overflow	FFFE016 to FFFE316	Interrupt on INTO instruction	
BRK instruction	FFFE416 to FFFE716	If the contents of address FFFE716 is FF16, program execution starts from the address shown by the vector in the relocatable vector table	
Address match	FFFE816 to FFFEB16		Address match interrupt
Single step (1)	FFFEC16 to FFFEF16		
Watchdog timer Oscillation stop and re-oscillation detection, low voltage detection	FFFF016 to FFFF316		Watchdog timer, clock generating circuit, voltage detection circuit
$\overline{\text{DBC}}$ (1)	FFFF416 to FFFF716		
$\overline{\text{NMI}}$	FFFF816 to FFFFB16		$\overline{\text{NMI}}$ interrupt
Reset(2)	FFFFC16 to FFFFF16		Reset

NOTE:

1. Do not normally use this interrupt because it is provided exclusively for use by development tools.

9.4.3 Saving Registers

In the interrupt sequence, the FLG register and PC are saved to the stack.

At this time, the 4 high-order bits of the PC and the 4 high-order (IPL) and 8 low-order bits of the FLG register, 16 bits in total, are saved to the stack first. Next, the 16 low-order bits of the PC are saved. **Figure 9.7** shows the stack status before and after an interrupt request is accepted.

The other necessary registers must be saved in a program at the beginning of the interrupt routine. Use the PUSHM instruction, and all registers except SP can be saved with a single instruction.

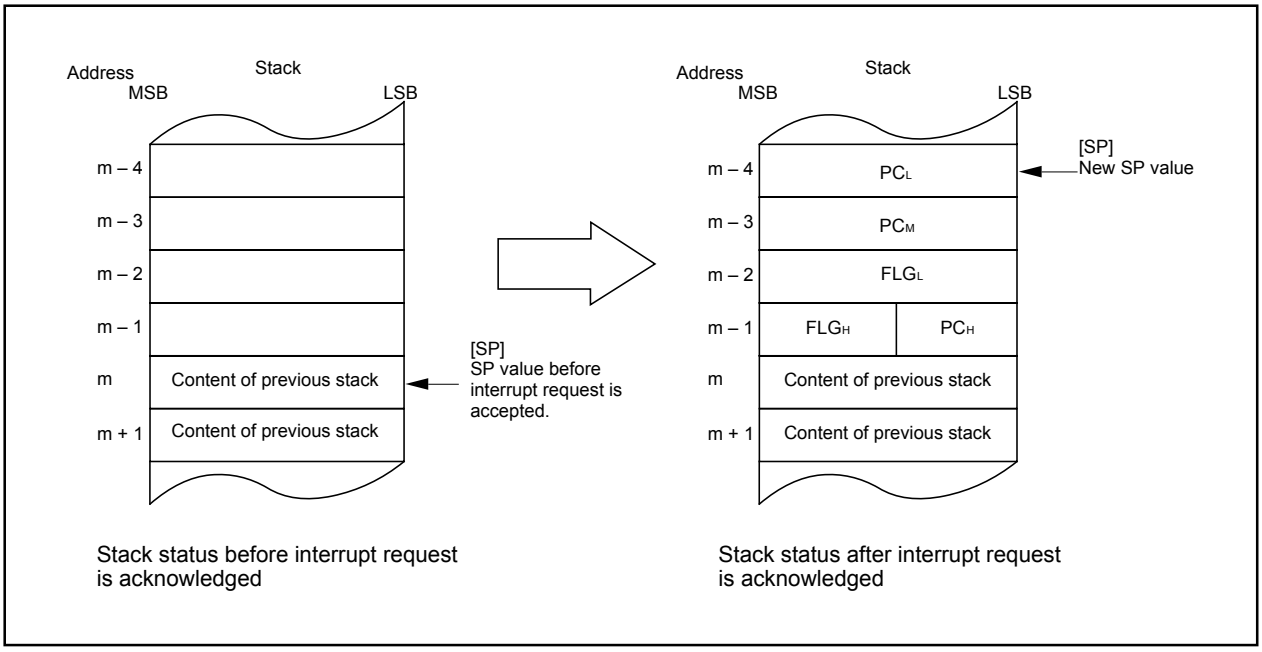


Figure 9.7 Stack Status Before and After Acceptance of Interrupt Request

Table 11.1 DMAC Specifications

Item		Specification
No. of channels		2 (cycle steal method)
Transfer memory space		• From any address in the 1M bytes space to a fixed address • From a fixed address to any address in the 1M bytes space • From a fixed address to a fixed address
Maximum No. of bytes transferred		128K bytes (with 16-bit transfers) or 64K bytes (with 8-bit transfers)
DMA request factors (1, 2)		Falling edge of $\overline{INT0}$ or $\overline{INT1}$ Both edge of $\overline{INT0}$ or $\overline{INT1}$ Timer A0 to timer A4 interrupt requests Timer B0 to timer B2 interrupt requests UART0 transfer, UART0 reception interrupt requests UART1 transfer, UART1 reception interrupt requests UART2 transfer, UART2 reception interrupt requests SI/O3, SI/O4 interrupt requests A/D conversion interrupt requests Timer S(IC/OC) requests Software triggers
Channel priority		DMA0 > DMA1 (DMA0 takes precedence)
Transfer unit		8 bits or 16 bits
Transfer address direction		forward or fixed (The source and destination addresses cannot both be in the forward direction)
Transfer mode	Single transfer	Transfer is completed when the DMAi transfer counter (i = 0,1) underflows after reaching the terminal count
	Repeat transfer	When the DMAi transfer counter underflows, it is reloaded with the value of the DMAi transfer counter reload register and a DMA transfer is con tinued with it
DMA interrupt request generation timing		When the DMAi transfer counter underflowed
DMA startup the		Data transfer is initiated each time a DMA request is generated when DMAE bit in the DMAiCON register = 1 (enabled)
DMA shutdown	Single transfer	• When the DMAE bit is set to 0 (disabled) • After the DMAi transfer counter underflows
	Repeat transfer	When the DMAE bit is set to 0 (disabled)
Reload timing for forward address pointer and transfer counter		When a data transfer is started after setting the DMAE bit to 1 (enabled), the forward address pointer is reloaded with the value of the SARi or the DARi pointer whichever is specified to be in the forward direction and the DMAi transfer counter is reloaded with the value of the DMAi transfer counter reload register

NOTES:

1. DMA transfer is not effective to any interrupt. DMA transfer is affected neither by the I flag nor by the interrupt control register.
2. The selectable causes of DMA requests differ with each channel.
3. Make sure that no DMAC-related registers (addresses 0020_{16} to $003F_{16}$) are accessed by the DMAC.

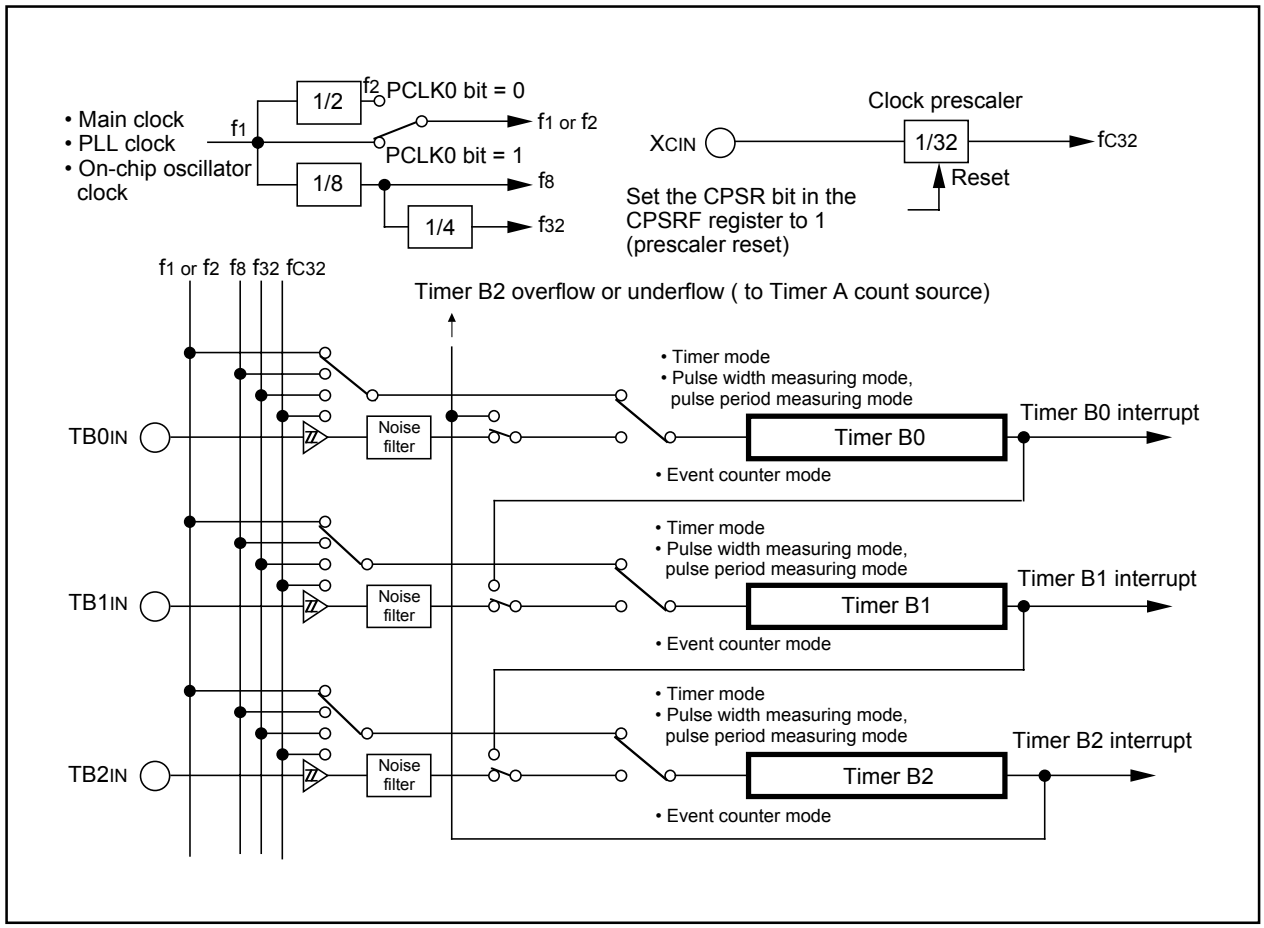


Figure 12.2. Timer B Configuration

Table 14.11 Registers to Be Used and Settings in I^2C bus mode (1) (Continued)

Register	Bit	Function	
		Master	Slave
U2TB	0 to 7	Set transmission data	Set transmission data
U2RB(1)	0 to 7	Reception data can be read	Reception data can be read
	8	ACK or NACK is set in this bit	ACK or NACK is set in this bit
	ABT	Arbitration lost detection flag	Invalid
	OER	Overrun error flag	Overrun error flag
U2BRG	0 to 7	Set bit rate	Invalid
U2MR(1)	SMD2 to SMD0	Set to 010_2	Set to 010_2
	CKDIR	Set to 0	Set to 1
	IOPOL	Set to 0	Set to 0
U2C0	CLK1, CLK0	Select the count source for the U2BRG register	Invalid
	CRS	Invalid because CRD = 1	Invalid because CRD = 1
	TXEPT	Transmit buffer empty flag	Transmit buffer empty flag
	CRD	Set to 1	Set to 1
	NCH	Set to 1	Set to 1
	CKPOL	Set to 0	Set to 0
	UFORM	Set to 1	Set to 1
U2C1	TE	Set this bit to 1 to enable transmission	Set this bit to 1 to enable transmission
	TI	Transmit buffer empty flag	Transmit buffer empty flag
	RE	Set this bit to 1 to enable reception	Set this bit to 1 to enable reception
	RI	Reception complete flag	Reception complete flag
	U2IRS	Invalid	Invalid
	U2RRM, U2LCH, U2ERE	Set to 0	Set to 0
U2SMR	IICM	Set to 1	Set to 1
	ABC	Select the timing at which arbitration-lost is detected	Invalid
	BBS	Bus busy flag	Bus busy flag
	3 to 7	Set to 0	Set to 0
U2SMR2	IICM2	Refer to **Table 14.13**	Refer to **Table 14.13**
	CSC	Set this bit to 1 to enable clock synchronization	Set to 0
	SWC	Set this bit to 1 to have SCL_2 output fixed to L at the falling edge of the 9th bit of clock	Set this bit to 1 to have SCL_2 output fixed to “L” at the falling edge of the 9^{th} bit of clock
	ALS	Set this bit to 1 to have SDA_2 output stopped when arbitration-lost is detected	Set to 0
	STAC	Set to 0	Set this bit to 1 to initialize UART2 at start condition detection
	SWC2	Set this bit to 1 to have SCL_2 output forcibly pulled low	Set this bit to 1 to have SCL_2 output forcibly pulled low
	SDHI	Set this bit to 1 to disable SDA_2 output	Set this bit to 1 to disable SDA_2 output
	7	Set to 0	Set to 0
U2SMR3	0, 2, 4 and NODC	Set to 0	Set to 0
	CKPH	Refer to **Table 14.13**	Refer to **Table 14.13**
	DL2 to DL0	Set the amount of SDA_2 digital delay	Set the amount of SDA_2 digital delay

NOTE:

1. Not all bits in the register are described above. Set those bits to 0 when writing to the registers in I^2C bus mode.

Table 14.12 Registers to Be Used and Settings in I^2C bus Mode (2) (Continued)

Register	Bit	Function	
		Master	Slave
U2SMR4	STAREQ	Set this bit to 1 to generate start condition	Set to 0
	RSTAREQ	Set this bit to 1 to generate restart condition	Set to 0
	STPREQ	Set this bit to 1 to generate stop condition	Set to 0
	STSPSEL	Set this bit to 1 to output each condition	Set to 0
	ACKD	Select ACK or NACK	Select ACK or NACK
	ACKC	Set this bit to 1 to output ACK data	Set this bit to 1 to output ACK data
	SCLHI	Set this bit to 1 to have SCL_2 output stopped when stop condition is detected	Set to 0
	SWC9	Set to 0	Set this bit to 1 to set the SCL_2 to "L" hold at the falling edge of the 9th bit of clock

NOTE:

1: Not all bits in the register are described above. Set those bits to 0 when writing to the registers in I^2C bus mode.

14.1.5 Special Mode 3 (IEBus mode)(UART2)

In this mode, one bit in the IEBus is approximated with one byte of UART mode waveform.

Table 14.17 lists the registers used in IEBus mode and the register values set. **Figure 14.30** shows the functions of bus collision detect function related bits.

If the TxD2 pin output level and RxD2 pin input level do not match, a UART2 bus collision detect interrupt request is generated.

Table 14.17 Registers to Be Used and Settings in IEBus Mode

Register	Bit	Function
U2TB	0 to 8	Set transmission data
U2RB(1)	0 to 8	Reception data can be read
	OER,FER,PER,SUM	Error flag
U2BRG	0 to 7	Set bit rate
U2MR	SMD2 to SMD0	Set to 110_2
	CKDIR	Select the internal clock or external clock
	STPS	Set to 0
	PRY	Invalid because PRYE is set to 0
	PRYE	Set to 0
	IOPOL	Select the TxD/RxD input/output polarity
U2C0	CLK1, CLK0	Select the count source for the U2BRG register
	CRS	Invalid because CRDis set to 1
	TXEPT	Transmit register empty flag
	CRD	Set to 1
	NCH	Select TxD2 pin output mode
	CKPOL	Set to 0
	UFORM	Set to 0
U2C1	TE	Set this bit to 1 to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to 1 to enable reception
	RI	Reception complete flag
	U2IRS	Select the source of UART2 transmit interrupt
	U2RRM, U2LCH, U2ERE	Set to 0
U2SMR	0 to 3, 7	Set to 0
	ABSCS	Select the sampling timing at which to detect a bus collision
	ACSE	Set this bit to 1 to use the auto clear function of transmit enable bit
	SSS	Select the transmit start condition
U2SMR2	0 to 7	Set to 0
U2SMR3	0 to 7	Set to 0
U2SMR4	0 to 7	Set to 0

NOTE:

1. Not all register bits are described above. Set those bits to 0 when writing to the registers in IEBus mode.

Figure 14.32 shows the example of connecting the SIM interface. Connect TxD2 and RxD2 and apply pull-up.

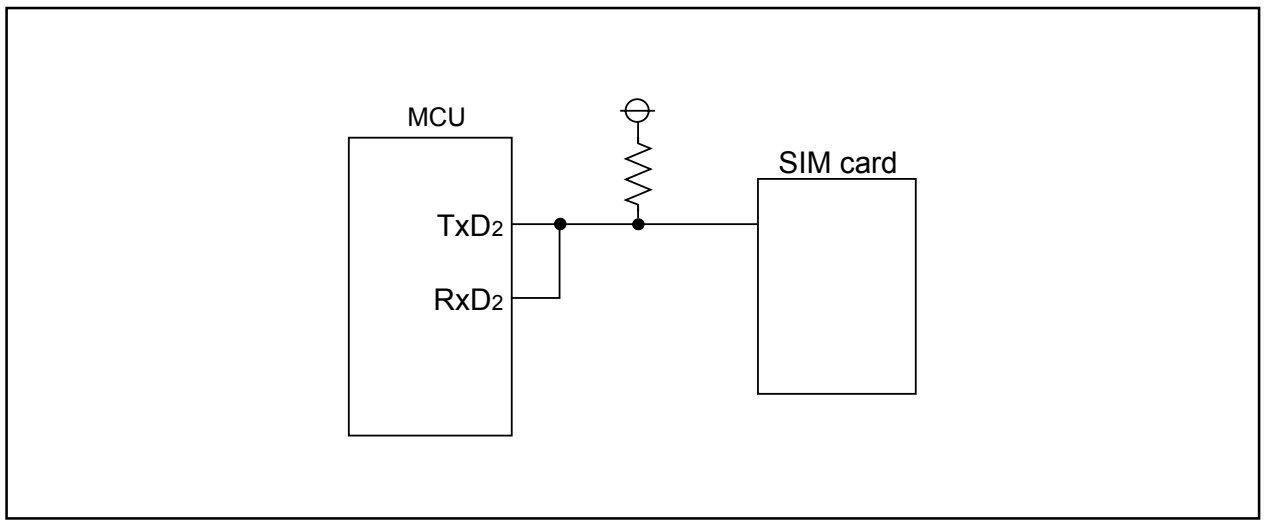


Figure 14.32 SIM Interface Connection

14.1.6.1 Parity Error Signal Output

The parity error signal is enabled by setting the U2ERE bit in theU2C1 register to 1.

• When receiving

The parity error signal is output when a parity error is detected while receiving data. This is achieved by pulling the TxD2 output low with the timing shown in **Figure 14.33**. If the R2RB register is read while outputting a parity error signal, the PER bit is cleared to 0 and at the same time the TxD2 output is returned high.

• When transmitting

A transmission-finished interrupt request is generated at the falling edge of the transfer clock pulse that immediately follows the stop bit. Therefore, whether a parity signal has been returned can be determined by reading the port that shares the RxD2 pin in a transmission-finished interrupt service routine.

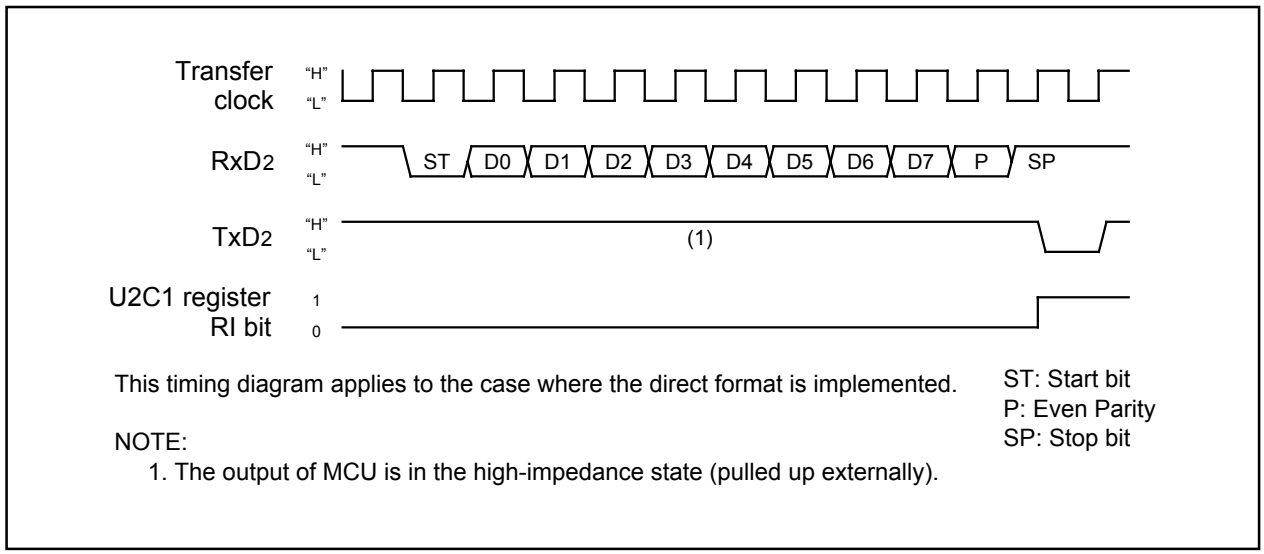


Figure 14.33 Parity Error Signal Output Timing

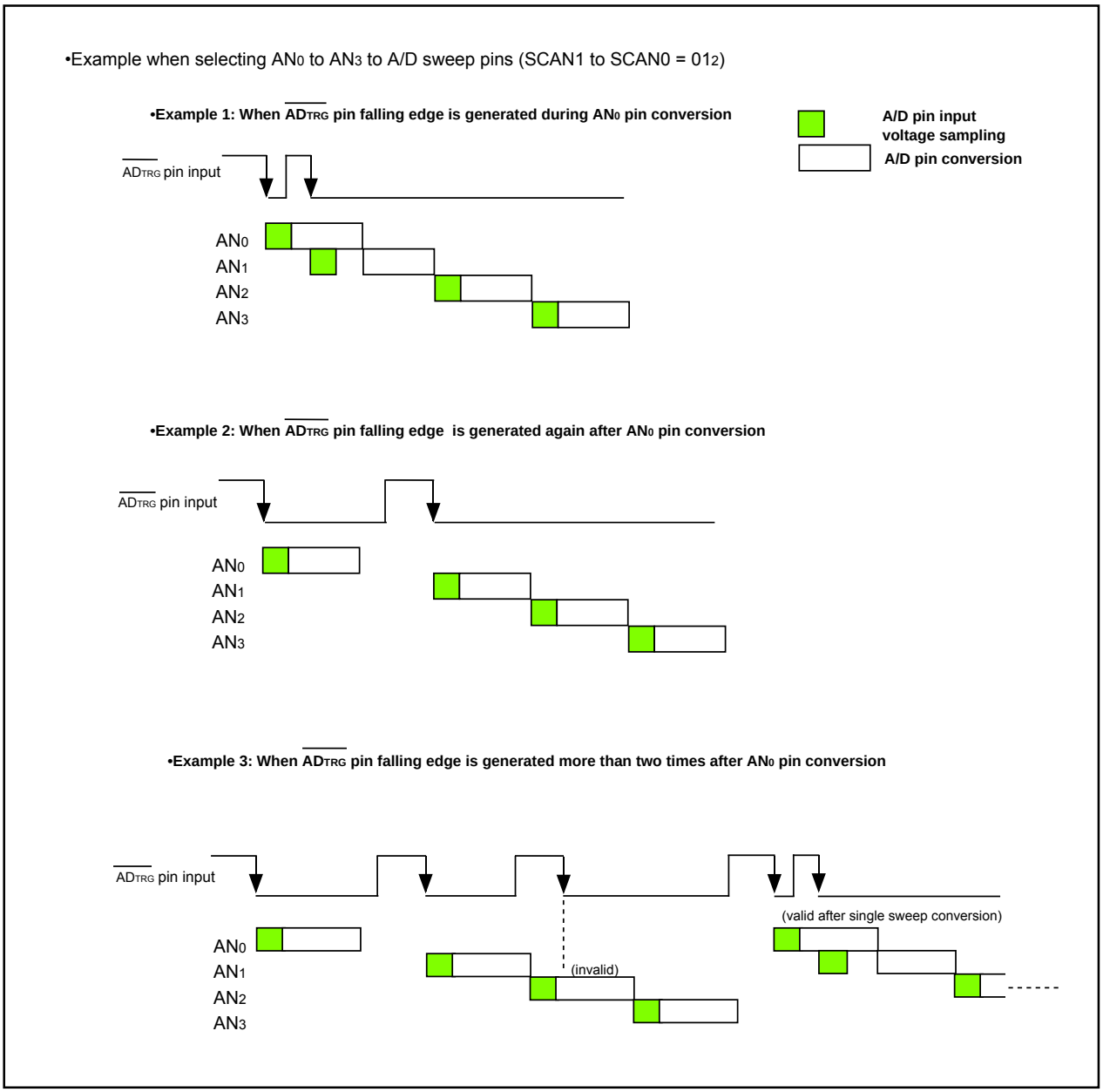


Figure 15.24 Operation Example in Delayed Trigger Mode1

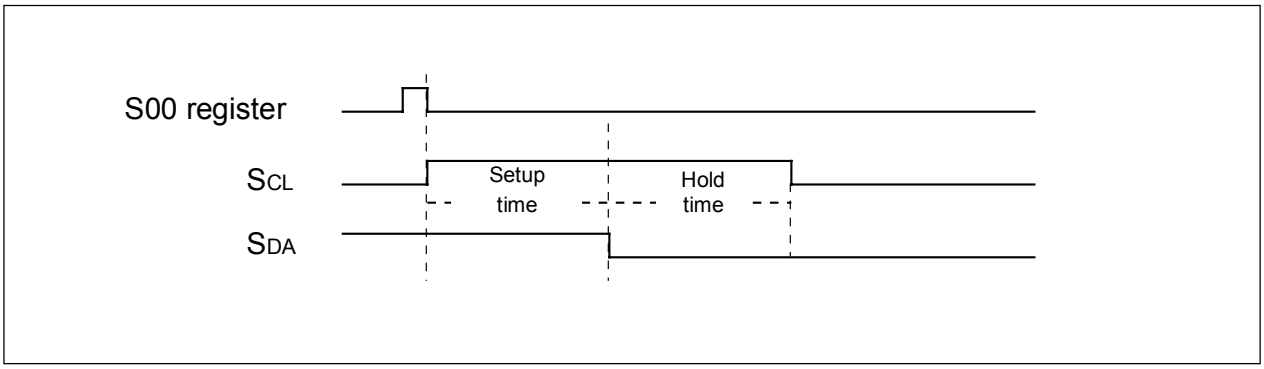


Figure 16.16 Start condition generation timing diagram

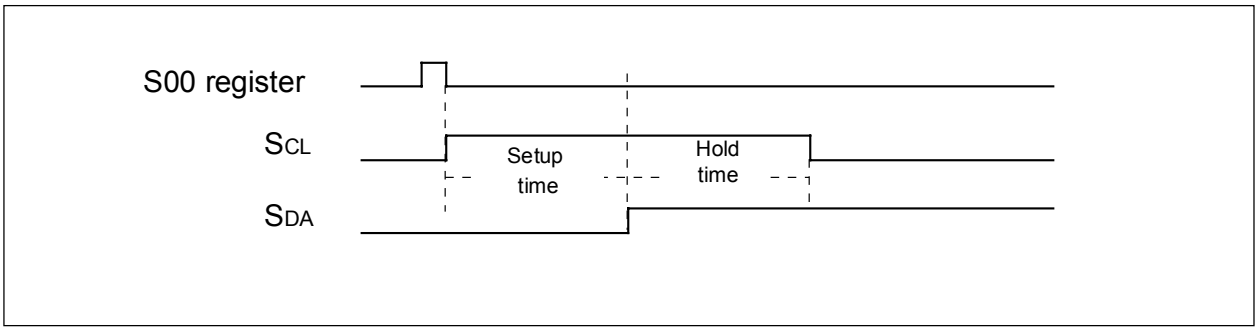


Figure 16.17 Stop condition generation timing diagram

Table 16.8 Start/Stop generation timing table

	Start/Stop Condition Generation Select Bit	Standard Clock Mode	High-speed Clock Mode
Setup time	0	5.0 µs (20 cycles)	2.5 µs (10 cycles)
	1	13.0 µs (52 cycles)	6.5 µs (26 cycles)
Hold time	0	5.0 µs (20 cycles)	2.5 µs (10 cycles)
	1	13.0 µs (52 cycles)	6.5 µs (26 cycles)

N OTE:
1. Actual time at the time of VIIC = 4MHz, The contents in () denote cycle numbers.

As mentioned above, when bits MST and TRX are set to 1, START condition or STOP condition mode is entered by writing 1 or 0 to the BB flag in the S10 register and writing 0 to the PIN bit and 4 low-order bits in the S10 register at the same time. Then SDAMM is left open in the START condition standby mode and SDAMM is set to low-level ("L") in the STOP condition standby mode. When the S00 register is set, the START/STOP conditions are generated. In order to set bits MST and TRX to 1 without generating the START/STOP conditions, write 1 to the 4 low-order bits simultaneously. **Table 16.9** lists functions along with the S10 register settings.

Table 16.9 S10 Register Settings and Functions

S10 Register Settings								Function
MST	TRX	BB	PIN	AL	AAS	AS0	LRB	
1	1	1	0	0	0	0	0	Setting up the START condition stand by in master transmit mode
1	1	0	0	0	0	0	0	Setting up the STOP condition stand by in master transmit mode
0/1	0/1	-	0	1	1	1	1	Setting up each communication mode (refer to **16.5 I²C status register**)

20.5.2 Flash Memory Control Register 1 (FMR1)

•FMR11 Bit

EW mode 1 is entered by setting the FMR11 bit to 1 (EW mode 1). The FMR11 bit is valid only when the FMR01 bit is set to 1.

•FMR16 Bit

The combined setting of bits FMR02 and FMR16 enables program and erase in the user ROM area. To set the FMR16 bit to 1, first set it to 0 and then 1. The FMR16 bit is valid only when the FMR01 bit is set to 1 (CPU rewrite mode enable).

•FMR17 Bit

If the FMR17 bit is set to 1 (with wait state), 1 wait state is inserted when blocks A and B are accessed, regardless of the content of the PM17 bit in the PM1 register. The PM17 bit setting is reflected to access other blocks and internal RAM, regardless of the FMR17 bit setting.

Set the FMR17 bit to 1 (with wait state) to rewrite more than 100 times (U7, U9).

Table 20.4 Protection using FMR16 and FMR02

FMR16	FMR02	Block A, Block B	Block 0, Block 1	other user block
0	0	write enabled	write disabled	write disabled
0	1	write enabled	write disabled	write disabled
1	0	write enabled	write disabled	write enabled
1	1	write enabled	write enabled	write enabled

20.5.3 Flash Memory Control Register 4 (FMR4)

•FMR40 Bit

The erase-suspend function is enabled when the FMR40 bit is set to 1 (enabled).

•FMR41 Bit

When the FMR41 bit is set to 1 by program during auto-erasing in EW mode 0, erase-suspend mode is entered. In EW mode 1, the FMR41 bit is automatically set to 1 (suspend request) to enter erase-suspend mode when an enabled interrupt request is generated. Set the FMR41 bit to 0 (erase restart) to restart an auto-erasing operation.

•FMR46 Bit

The FMR46 bit is set to 0 during auto-erasing. It is set to 1 in erase-suspend mode.

Do not access to flash memory when the FMR46 bit is set to 0.

VCC = 3V

Timing Requirements

(VCC = 3V, VSS = 0V, at Topr = – 20 to 85°C / – 40 to 85°C unless otherwise specified)

Table 21.27 Timer A Input (Counter Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	150		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	60		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	60		ns

Table 21.28 Timer A Input (Gating Input in Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	600		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	300		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	300		ns

Table 21.29 Timer A Input (External Trigger Input in One-shot Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	300		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	150		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	150		ns

Table 21.30 Timer A Input (External Trigger Input in Pulse Width Modulation Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(TAH)}$	TAiIN input HIGH pulse width	150		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	150		ns

Table 21.31 Timer A Input (Counter Increment/decrement Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(UP)}$	TAiOUT input cycle time	3000		ns
$t_{w(UPH)}$	TAiOUT input HIGH pulse width	1500		ns
$t_{w(UPL)}$	TAiOUT input LOW pulse width	1500		ns
$t_{su(UP-TIN)}$	TAiOUT input setup time	600		ns
$t_{h(TIN-UP)}$	TAiOUT input hold time	600		ns

Table 21.32 Timer A Input (Two-phase Pulse Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	2		µs
$t_{su(TAIN-TAOUT)}$	TAiOUT input setup time	500		ns
$t_{su(TAOUT-TAIN)}$	TAiIN input setup time	500		ns

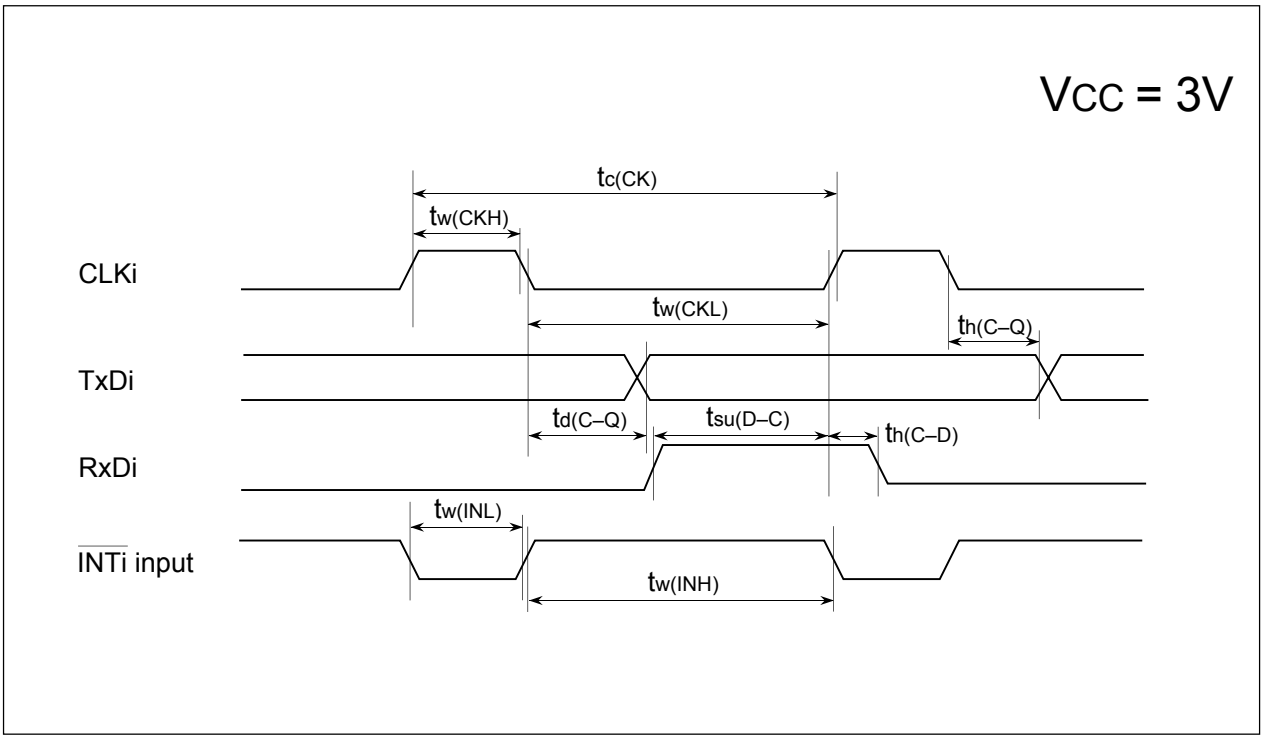


Figure 21.5 Timing Diagram (2)

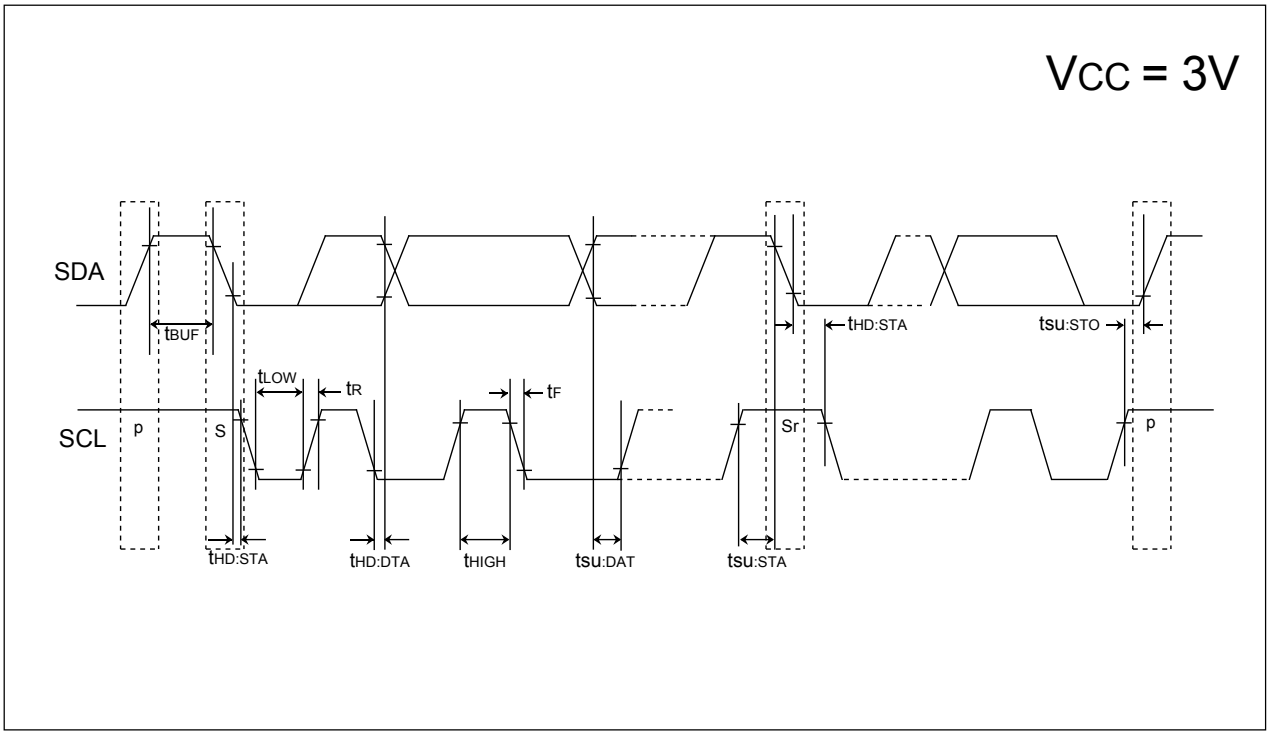


Figure 21.6 Timing Diagram (3)

V_{CC} = 5V

Table 21.47 Electrical Characteristics (2) (Note 1)

Symbol	Parameter	Measurement Condition			Standard			Unit
					Min.	Typ.	Max.	
I_{CC}	Power Supply Current (V_{CC}=4.2 to 5.5V)	Output pins are left open and other pins are connected to V_{SS}	Mask ROM	f(BCLK) = 20 MHz, main clock, no division		18	25	mA
				On-chip oscillation, $f_{2(ROC)}$ selected, f(BCLK) = 1 MHz		2		mA
			Flash memory	f(BCLK) = 20 MHz, main clock, no division		18	25	mA
				On-chip oscillation, $f_{2(ROC)}$ selected, f(BCLK) = 1 MHz		2		mA
			Flash memory program	f(BCLK) = 10 MHz, Vcc = 5.0 V		11		mA
			Flash memory erase	f(BCLK) = 10 MHz, Vcc = 5.0 V		11		mA
			Mask ROM	f(BCLK) = 32 kHz, In low-power consumption mode, Program running on ROM(3)		25		µA
				On-chip oscillation, $f_{2(ROC)}$ selected, f(BCLK) = 1 MHz, In wait mode		50		µA
			Flash memory	f(BCLK) = 32 kHz, In low-power consumption mode, Program running on RAM(3)		25		µA
				f(BCLK) = 32 kHz, In low-power consumption mode, Program running on flash memory(3)		450		µA
				On-chip oscillation, $f_{2(ROC)}$ selected, f(BCLK) = 1 MHz, In wait mode		50		µA
			Mask ROM, Flash memory	f(BCLK) = 32 kHz, In wait mode(2), Oscillation capacity high		8.5		µA
				f(BCLK) = 32 kHz, In wait mode(2), Oscillation capacity low		3		µA
				While clock stops, Topr = 25° C		0.8	3	µA

NOTES:

1. Referenced to V_{CC} = 4.2 to 5.5 V, V_{SS} = 0 V at Topr = -40 to 85 ° C, f(BCLK) = 20 MHz unless otherwise specified.
2. With one timer operates, using f_{C32}.
3. This indicates the memory in which the program to be executed exists.

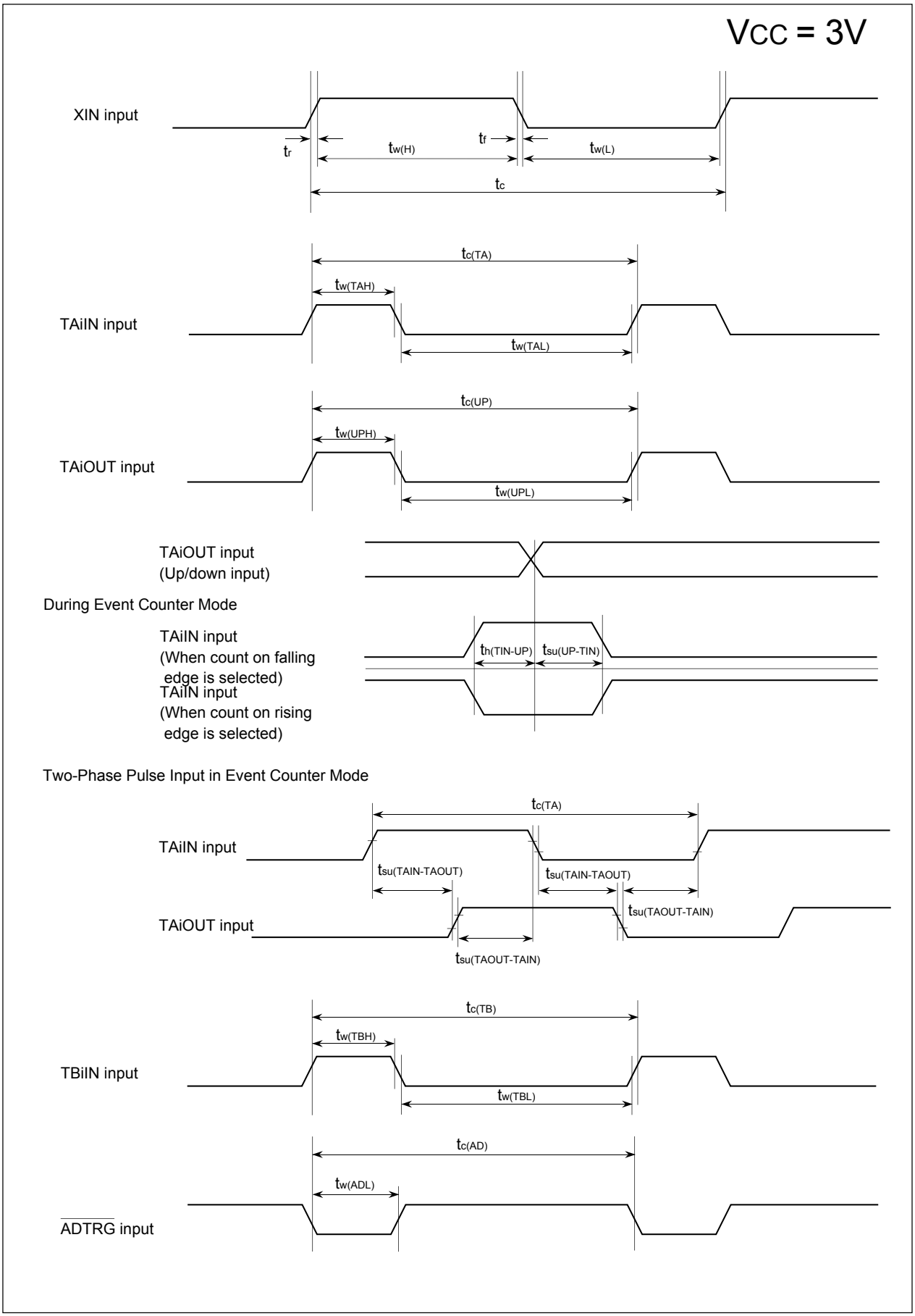


Figure 21.10 Timing Diagram (1)

22.2.2 Power Control

1. When exiting stop mode by hardware reset, the device will startup using the on-chip oscillator.

2. Set the MR0 bit in the TAiMR register(i=0 to 4) to 0 (pulse is not output) to use the timer A to exit stop mode.

3. When entering wait mode, insert a JMP.B instruction before a WAIT instruction. Do not excute any instructions which can generate a write to RAM between the JMP.B and WAIT instructions. Disable the DMA transfers, if a DMA transfer may occur between the JMP.B and WAIT instructions. After the WAIT instruction, insert at least 4 NOP instructions. When entering wait mode, the instruction queue reads ahead the instructions following WAIT, and depending on timing, some of these may execute before the MCU enters wait mode.

Program example when entering wait mode

```
Program Example:    JMP.B   L1    ; Insert JMP.B instruction before WAIT instruction
                L1:
                    FSET    I     ;
                    WAIT          ; Enter wait mode
                    NOP           ; More than 4 NOP instructions
                    NOP
                    NOP
                    NOP
```

4. When entering stop mode, insert a JMP.B instruction immediately after executing an instruction which sets the CM10 bit in the CM1 register to 1, and then insert at least 4 NOP instructions. When entering stop mode, the instruction queue reads ahead the instructions following the instruction which sets the CM10 bit to 1 (all clock stops), and, some of these may execute before the MCU enters stop mode or before the interrupt routine for returning from stop mode.

Program example when entering stop mode

```
Program Example:    FSET    I
                    BSET    CM10    ; Enter stop mode
                    JMP.B   L2      ; Insert JMP.B instruction
                L1:
                    NOP             ; More than 4 NOP instructions
                    NOP
                    NOP
                    NOP
```

Appendix 2.2 Difference between M16C/28 and M16C/29 Group (T-ver./V-ver.) (1)

Item	Description	M16C/28(T-ver./V-ver.)	M16C/29(T-ver./V-ver.)
Protection	Function of the PRC0 bit	Enable to set the CM0, CM1, CM2, POCR, PLC0 and PCLKR registers	Enable to set the CM0, CM1, CM2, POCR, PLC0, PCLKR and CCLKR registers
Interrupt	The IFSR20 bit setting in the IFSR2A register	Set to 1	Set to 0
	The b1 bit in the IFSR2A register	Not available (reserved bit)	Interrupt cause switching bit (0: A/D conversion, 1:key input)
	The b2 bit in the IFSR2A register	Not available (reserved bit)	Interrupt cause switching bit (0: CAN0 wake-up/ error)
	Interrupt cause in the Interrupt number 13	Key input interrupt	CAN0 error
	Interrupt cause in the Interrupt number 14	Key input interrupt	A/D, key input interrupt
CAN module	compatible to 2.0B	Not available (all related registers are reserved registers)	Available (1 channel)
Pin Function	2 pins (80-pin/85-pin package), 62 pins (64-pin package)	$P9_3$/$AN2_4$	$P9_3$/$AN2_4$/CTX
	3 pins (80-pin/85-pin package), 64 pins (64-pin package)	$P9_2$/$TB2_{IN}$	$P9_2$/$AN3_2$/$TB2_{IN}$/CRX

I: Input O: Output I/O: Input and output

NOTE:
1. Since the M16C/28 group uses the common emulator used in the M16C/29 group, all the functions are available for M16C/28. When evaluating M16C/28 group, do not access to the SFR which is not built-in the M16C/28 gorup. Refere to hardware manual for details and electrical characteristics.