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	Active
Core Processor	H8S/2000
Core Size	16-Bit
Speed	16MHz
Connectivity	SCI, SmartCard, USB
Peripherals	DMA, POR, PWM, WDT
Number of I/O	68
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 6x10b; D/A 2x8b
Oscillator Type	External
Operating Temperature	-20°C ~ 75°C (TA)
Mounting Type	Surface Mount
Package / Case	120-TQFP
Supplier Device Package	120-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/df2215te16v

Section 8 Data Transfer Controller (DTC)

Section 6 Bus Controller

Section 7 DMA Controller (DMAC)

Section 8 Data Transfer Controller (DTC)

Section 9 I/O Ports

1.5 Pin Functions

Type	Symbol	Pin No. TFP-120, TFP-120V	Pin No. BP-112, BP-112V	I/O	Function
Power Supply	VCC	10 75	E4 F11	Input	Power supply pins. Connect all these pins to the system power supply.
	VSS	12 61 73	E1 G10	Input	Ground pins. Connect all these pins to the system power supply (0 V).
	PLLVCC	64	H8	Input	Power supply pin for internal PLL oscillator. Connect this pin to the system power supply.
	PLLVSS	62	K10	Input	Ground pin for an on-chip PLL oscillator.
	PLLCAP	63	K11	Output	External capacitor pin for an on-chip PLL oscillator.
Clock	XTAL	74	F9	Input	For connection to a crystal resonator. For examples of crystal resonator connection and external clock input, see section 21, Clock Pulse Generator.
	EXTAL	76	F10	Input	For connection to a crystal resonator. (An external clock can be supplied from the EXTAL pin.) For examples of crystal resonator connection and external clock input, see section 21, Clock Pulse Generator.
	XTAL48	65	J10	Input	USB operating clock input pins. 48-MHz clock for USB communications is input. For examples of using an on-chip PLL, EXTAL48 must be fixed low and XTAL48 must be open.
	EXTAL48	66	J11	Input	
	ϕ	78	E11	Output	Supplies the system clock to external devices.
Operating Mode Control	MD2 MD1 MD0	77 68 67	F8 H10 H9	Input	Except for mode changing, be sure to fix the levels of the mode pins (MD2 to MD0) by pulling them down or pulling them up until the power turns off.

5.4 Interrupt Sources

5.4.1 External Interrupts

There are eight external interrupts: NMI, IRQ7, and IRQ5 to IRQ0. These interrupts can be used to restore this LSI from software standby mode. IRQ6 is an interrupt only for the on-chip USB. However, IRQ6 is functionally same as IRQ7 restore this LSI from software standby mode. IRQ6 is functionally same as IRQ7 and IRQ5 to IRQ0.

NMI Interrupt: NMI is the highest-priority interrupt, and is always accepted by the CPU regardless of the interrupt control mode or the status of the CPU interrupt mask bits. The NMIEG bit in SYSCR can be used to select whether an interrupt is requested at a rising edge or a falling edge on the NMI pin.

IRQ7 to IRQ0 Interrupts: Interrupts IRQ7 to IRQ0 are requested by an input signal at pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$. Interrupts IRQ7 to IRQ0 have the following features:

- Using ISCR, it is possible to select whether an interrupt is generated by a low level, falling edge, rising edge, or both edges, at pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$
- Enabling or disabling of interrupt requests IRQ7 to IRQ0 can be selected with IER.
- The interrupt priority level can be set with IPR.

— The status of interrupt requests IRQ7 to IRQ0 is indicated in ISR. ISR flags can be cleared to 0 by software.

A block diagram of IRQn interrupts is shown in figure 5.2.

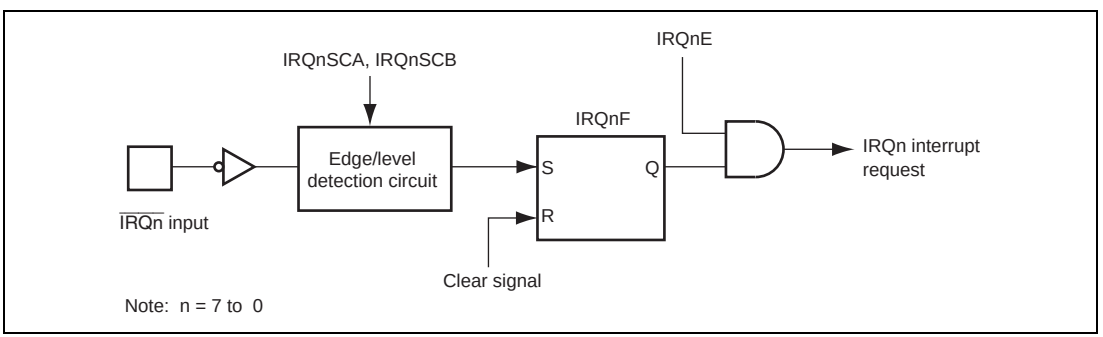


Figure 5.2 Block Diagram of IRQn Interrupts

The setting for IRQnF is shown in figure 5.3.

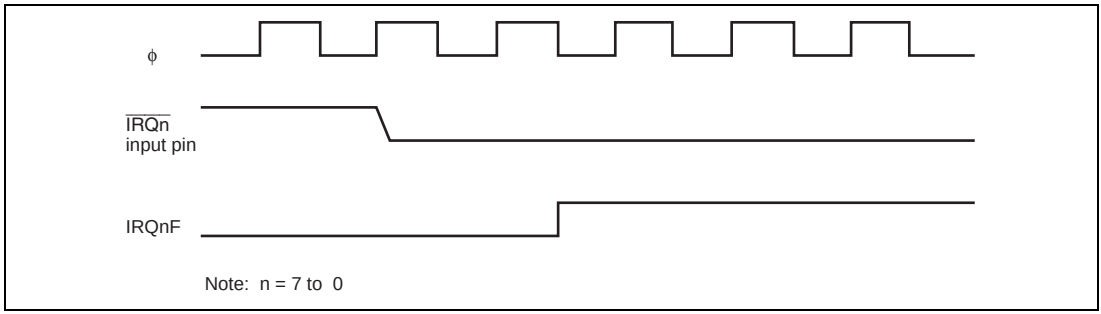


Figure 5.3 Set Timing for IRQnF

The detection of IRQn interrupts does not depend on whether the relevant pin has been set for input or output. However, when a pin is used as an external interrupt input pin, do not clear the corresponding DDR to 0; and use the pin as an I/O pin for another function. IRQnF interrupt request flag is set when the setting condition is satisfied, regardless of IER settings. Accordingly, refer to only necessary flags.

5.4.2 Internal Interrupts

The sources for internal interrupts from on-chip peripheral modules have the following features:

- For each on-chip peripheral module there are flags that indicate the interrupt request status, and enable bits that select enabling or disabling of these interrupts. If both of these are set to 1 for a particular interrupt source, an interrupt request is issued to the interrupt controller.
- The interrupt priority level can be set by means of IPR.
- The DMAC or DTC can be activated by a TPU, SCI, or other interrupt request.
- When the DMAC or DTC is activated by an interrupt request, it is not affected by the interrupt control mode or CPU interrupt mask bit.

5.5 Interrupt Exception Handling Vector Table

Table 5.2 shows interrupt exception handling sources, vector addresses, and interrupt priorities.

For default priorities, the lower the vector number, the higher the priority. Priorities among modules can be set by means of the IPR. Modules set at the same priority will conform to their default priorities. Priorities within a module are fixed.

6.5.2 On-Chip Peripheral Module Access Timing

The on-chip peripheral modules are accessed in two states except on-chip USB. The data bus is either 8 bits or 16 bits wide, depending on the particular internal I/O register being accessed. Figure 6.6 shows the access timing for the on-chip peripheral modules. Figure 6.7 shows the pin states.

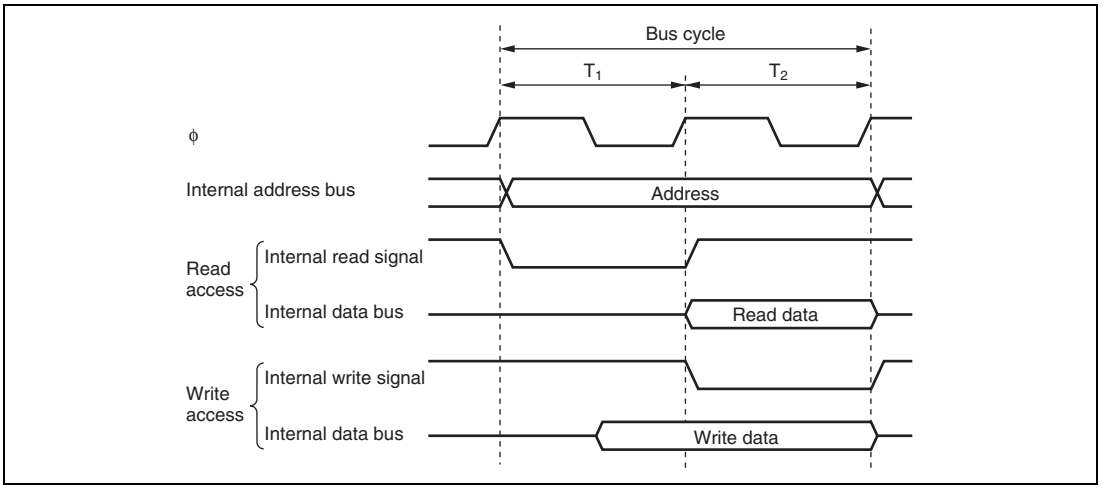


Figure 6.6 On-Chip Peripheral Module Access Cycle

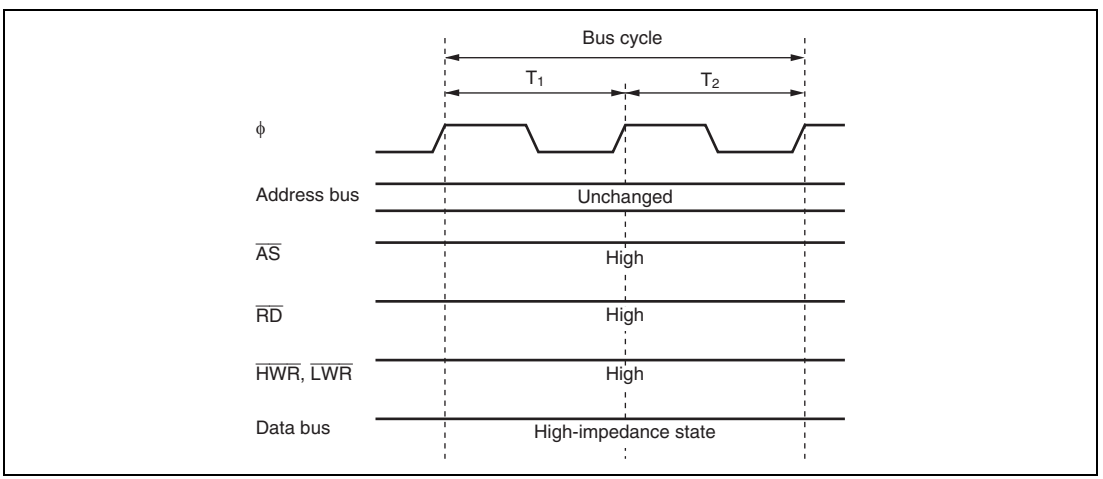
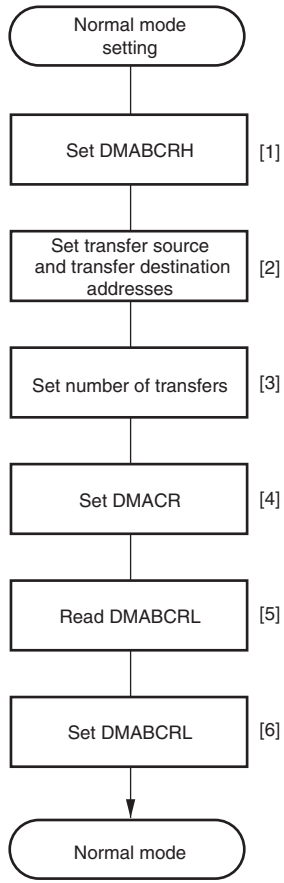


Figure 6.7 Pin States during On-Chip Peripheral Module Access

Figure 7.10 shows an example of the setting procedure for normal mode.



[1] Set each bit in DMABCRH.
- Set the FAE bit to 1 to select full address mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address in MARA, and the transfer destination address in MARB.

[3] Set the number of transfers in ETCRA.

[4] Set each bit in DMACRA and DMACRB.
- Set the transfer data size with the DTSZ bit.
- Specify whether MARA is to be incremented, decremented, or fixed, with the SAID and SAIDE bits.
- Clear the BLKE bit to 0 to select normal mode.
- Specify whether MARB is to be incremented, decremented, or fixed, with the DAID and DAIDE bits.
- Select the activation source with bits DTF3 to DTF0.

[5] Read the DTE = 0 and DTME = 0 in DMABCRL.

[6] Set each bit in DMABCRL.
- Specify enabling or desabling of transfer end interrupts with the DTIE bit.
- Set both the DTME bit and the DTE bit to 1 to enable transfer.

Figure 7.10 Example of Normal Mode Setting Procedure

Figure 7.11 illustrates operation in block transfer mode when MARB is designated as a block area.

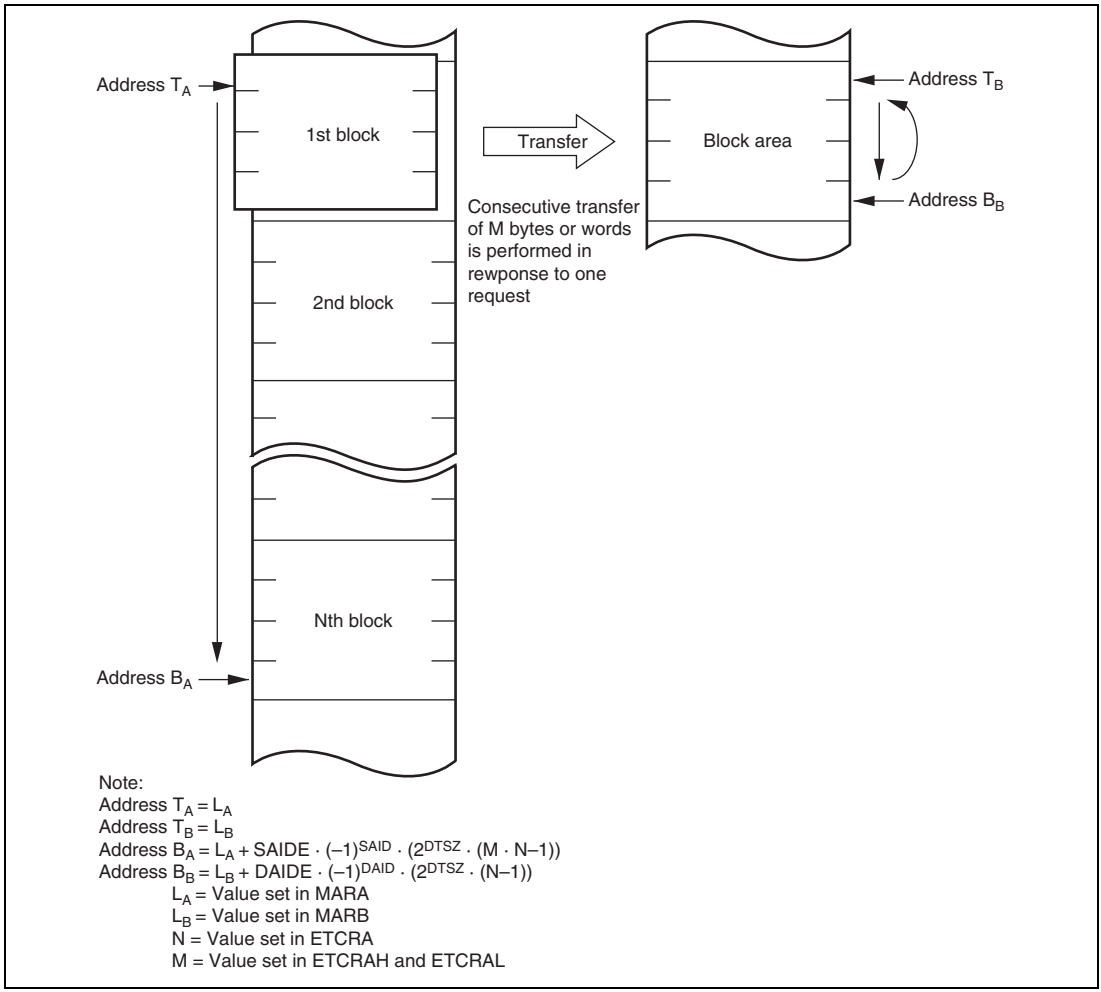


Figure 7.11 Operation in Block Transfer Mode (BLKDIR = 0)

8.5.1 Normal Mode

In normal mode, one operation transfers one byte or one word of data.

From 1 to 65,536 transfers can be specified. Once the specified number of transfers have ended, a CPU interrupt can be requested.

Table 8.4 shows the register information in normal mode, and figure 8.6 shows the memory mapping in normal mode.

Table 8.4 Register Information in Normal Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates source address
DTC destination address register	DAR	Designates destination address
DTC transfer count register A	CRA	Designates transfer count
DTC transfer count register B	CRB	Not used

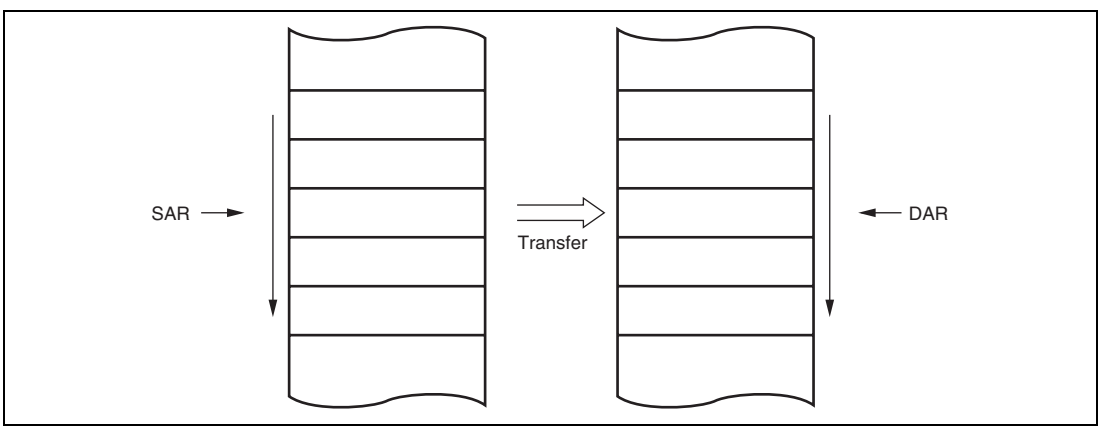


Figure 8.6 Memory Mapping in Normal Mode

8.7 Examples of Use of the DTC

8.7.1 Normal Mode

An example is shown in which the DTC is used to receive 128 bytes of data via the SCI.

1. Set MRA to fixed source address (SM1 = SM0 = 0), incrementing destination address (DM1 = 1, DM0 = 0), normal mode (MD1 = MD0 = 0), and byte size (Sz = 0). The DTS bit can have any value. Set MRB for one data transfer by one interrupt (CHNE = 0, DISEL = 0). Set the SCI RDR address in SAR, the start address of the RAM area where the data will be received in DAR, and 128 (H'0080) in CRA. CRB can be set to any value.
2. Set the start address of the register information at the DTC vector address.
3. Set the corresponding bit in DTCER to 1.
4. Set the SCI to the appropriate receive mode. Set the RIE bit in SCR to 1 to enable the reception complete (RXI) interrupt. Since the generation of a receive error during the SCI reception operation will disable subsequent reception, the CPU should be enabled to accept receive error interrupts.
5. Each time reception of one byte of data ends on the SCI, the RDRF flag in SSR is set to 1, an RXI interrupt is generated, and the DTC is activated. The receive data is transferred from RDR to RAM by the DTC. DAR is incremented and CRA is decremented. The RDRF flag is automatically cleared to 0.
6. When CRA becomes 0 after the 128 data transfers have ended, the RDRF flag is held at 1, the DTCE bit is cleared to 0, and an RXI interrupt request is sent to the CPU. The interrupt handling routine should perform wrap-up processing.

8.7.2 Software Activation

An example is shown in which the DTC is used to transfer a block of 128 bytes of data by means of software activation. The transfer source address is H'1000 and the destination address is H'2000. The vector number is H'60, so the vector address is H'04C0.

1. Set MRA to incrementing source address (SM1 = 1, SM0 = 0), incrementing destination address (DM1 = 1, DM0 = 0), block transfer mode (MD1 = 1, MD0 = 0), and byte size (Sz = 0). The DTS bit can have any value. Set MRB for one block transfer by one interrupt (CHNE = 0). Set the transfer source address (H'1000) in SAR, the destination address (H'2000) in DAR, and 128 (H'8080) in CRA. Set 1 (H'0001) in CRB.
2. Set the start address of the register information at the DTC vector address (H'04C0).
3. Check that the SWDTE bit in DTVECR is 0. Check that there is currently no transfer activated by software.

- When TGR is an output compare register

  When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register. This operation is illustrated in figure 10.16.

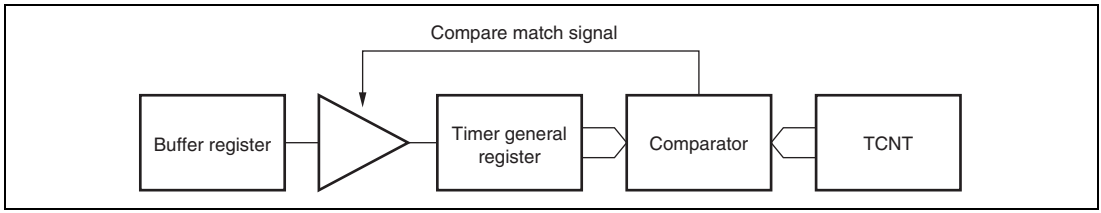


Figure 10.16 Compare Match Buffer Operation

- When TGR is an input capture register

  When input capture occurs, the value in TCNT is transferred to TGR and the value previously held in the timer general register is transferred to the buffer register. This operation is illustrated in figure 10.17.

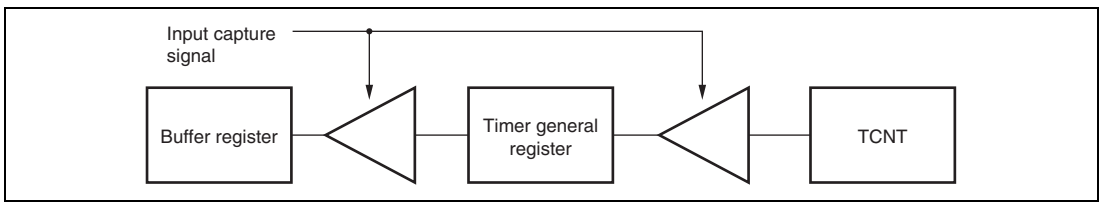


Figure 10.17 Input Capture Buffer Operation

Example of Buffer Operation Setting Procedure: Figure 10.18 shows an example of the buffer operation setting procedure.

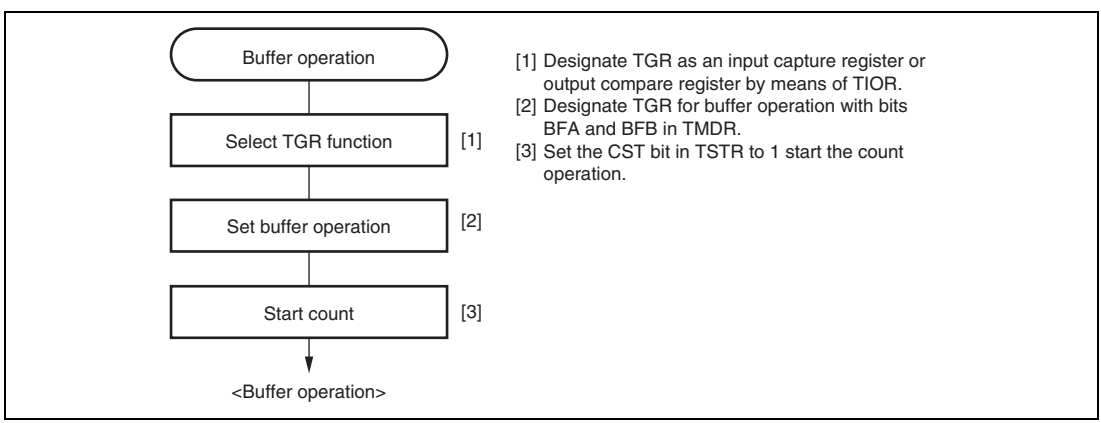


Figure 10.18 Example of Buffer Operation Setting Procedure

14.4 Boundary Scan Function Operation

14.4.1 TAP Controller

Figure 14.3 shows the TAP controller status transition diagram, based on the JTAG standard.

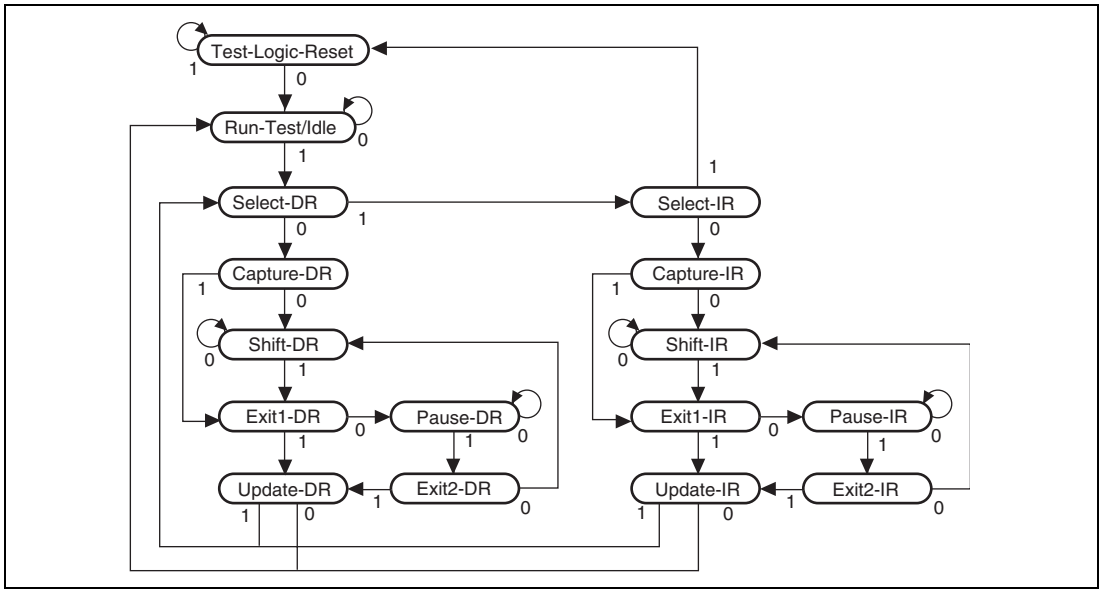


Figure 14.3 TAP Controller Status Transition

Note: The transition condition is the TMS value at the rising edge of TCK. The TDI value is sampled at the rising edge of the TCK and shifted at the falling edge of the TCK. The TDO value changes at the falling edge of the TCK. In addition, TDO is high-impedance state in a state other than Shift-DR or Shift-IR state. If $\overline{\text{TRST}}$ is 0, Test-Logic-Reset state is entered asynchronously with the TCK.

15.3.28 USB Interrupt Flag Register 1 (UIFR1) (Only in H8S/2215R, H8S/2215T and H8S/2215C)

UIFR1 is an interrupt flag register indicating the EP2i, EP2o, EP3i, and EP3o. If the corresponding bit is set to 1, the corresponding $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$ interrupt is requested from the CPU. EP2iTR and EP3iTR flags can cleared by writing 0 to them. Writing 1 to them is invalid and causes no operation. Consequently, to clear only a specific flag it is necessary to write 0 to the bit corresponding to the flag to be cleared and 1 to all the other bits. (To clear bit 5 only, write H'DF.) The bit-clear instruction is a read/modify/write instruction. There is a danger that the wrong bits may be cleared if a new flag is set between the read and write. Therefore, the bit-clear instruction should not be used to clear bits in this interrupt flag register. However, EP2iEMPTY, EP2oREDY, EP3oTS and EP3oTF are status bits, and cannot be cleared.

Bit	Bit Name	Initial Value	R/W	Description
7	EP3oTF	0	R	EP3o Abnormal Receive Indicates the status of EP3o FIFO, which can be read after the next SOF packet has been received following the data transmission from the host. This flag is set to 1 if a PID error, CRC error, bit staff error, data size error, or Bad EOP occurs when the data is transferred from the host to the EP3o. This is a status bit and cannot be cleared. In addition, an interrupt cannot be requested by this flag.
6	EP3oTS	0	R	EP3o Normal Receive Indicates the status of EP3o FIFO, which can be read after the next SOF packet has been received following the data transmission from the host. This flag is set to 1 if data is normally transferred from the host to the EP3o. This is a status bit and cannot be cleared. In addition, an interrupt cannot be requested by this flag.
5	EP3iTF	0	R/(W)*	EP3i Abnormal Transfer Set to 1 if data to be written to the EP3i FIFO is lost because no IN token has been returned. This flag is set when the SOF packet that is two packets after the data write is received. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.
4	EP3iTR	0	R/(W)*	EP3i Transmit Request Set to 1 if there is no valid transmit data in the FIFO to be accessed by the UDC when an IN token is sent from the host to EP3i. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.

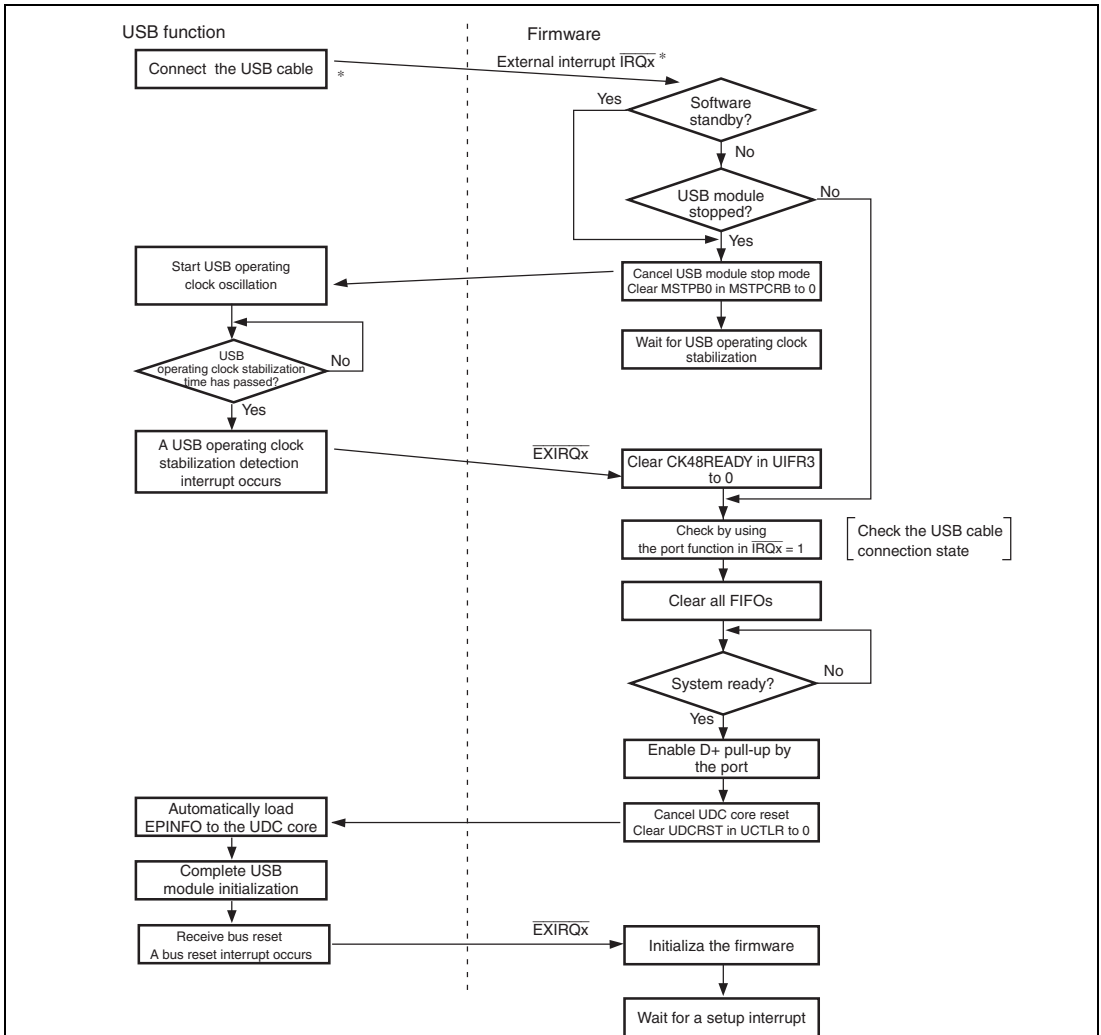
15.3.31 USB Interrupt Flag Register 3 (UIFR3)

UIFR3 is an interrupt flag register indicating the USB status. If the corresponding bit is set to 1, the corresponding $\overline{\text{EXIRQ0}}$, $\overline{\text{EXIRQ1}}$, or $\overline{\text{IRQ6}}$ interrupt is requested from the CPU. VBUSi, SPRSi, SETI, SETC, SOF, and CK48READY flags can be cleared by writing 0. Writing 1 to them is invalid and causes no operation. Consequently, to clear only a specific flag it is necessary to write 0 to the bit corresponding to the flag to be cleared and 1 to all the other bits. (To clear bit 5 only, write H'DF.) The bit-clear instruction is a read/modify/write instruction. There is a danger that the wrong bits may be cleared if a new flag is set between the read and write. Therefore, the bit-clear instruction should not be used to clear bits in this interrupt flag register. VBUSs and SPRSs are status flags and cannot be cleared.

Bit	Bit Name	Initial Value	R/W	Description
7	CK48READY	0	R/(W)*	USB Operating Clock (48 MHz) Stabilization Detection Set to 1 when the 48-MHz USB operating clock stabilization time has been automatically counted after USB module rest mode cancellation. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$. CK48READY can also operate in USB interface software reset state (the UIFRST bit of UCTLR is set to 1). Note that USB operating clock stabilization time differs according to the clock source, refer to the UCKS3 to UCKS0 bits of the UCTLR.
6	SOF	0	R/(W)*	Start of Frame Packet Detection Set to 1 if the SOF packet is detected. This flag can be used to start time stamp check, EP3i transmit data write, or EP3o receive data read timing in EP3 isochronous transfer. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.
5	SETC	0	R/(W)*	Set_Configuration Command Detection Set to 1 if the Set_Configuration command is detected. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.
4	SETI	0	R/(W)*	Set_Inferface Command Detection Set to 1 if the Set_Interface command is detected. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.

(2) USB Cable Connection (When USB Module Stop or Software Standby Is Used)

If the USB cable enters the connection state from disconnection state an application (self powered) where USB module stop or software standby mode is used, perform the operation as shown in figure 15.5.



Note: * A VBUS interrupt in the USB module cannot be detected in the software standby state or in the USB module stop state. Accordingly, in an application in which software standby or USB module stop is used in self-powered mode, a VBUS interrupt of the USB must be detected via the external interrupt pin IRQx. In this case, the IRQx pin must be specified as both-edge sensitive. When IRQx is used, a VBUS interrupt in the USB module need not to be used.

Figure 15.5 USB Cable Connection (When USB Module Stop or Software Standby Is Used)

Table 19.4 SCI Boot Mode Operation

Item	Host Operation	LSI Operation
		Branches to boot program at reset-start.
Bit rate adjustment	Continuously transmits data H'00 at specified bit rate.	Measures low-level period of receive data H'00.
		Calculates bit rate and sets it in BRR of SCI_2.
	Transmits data H'55 when data H'00 is received error-free.	Transmits data H'00 to host as adjustment end indication.
		Transmits data H'AA to host when data H'55 is received.
Transmits number of bytes (N) of programming control program	Transmits number of bytes (N) of programming control program to be transferred as 2-byte data (low-order byte following high-order byte)	Echobacks the 2-byte data received as verification data.
Transmits 1-byte of programming control program (repeated for N times)	Transmits 1-byte of programming control program	Echobacks received data to host and also transfers it to RAM
Flash memory erase		Checks flash memory data, erases all flash memory blocks in case of written data existing, and transmits data H'AA to host. (If erase could not be done, transmits data H'FF to host and aborts operation.)
Programming control program execution		Branches to programming control program transferred to on-chip RAM and starts execution.

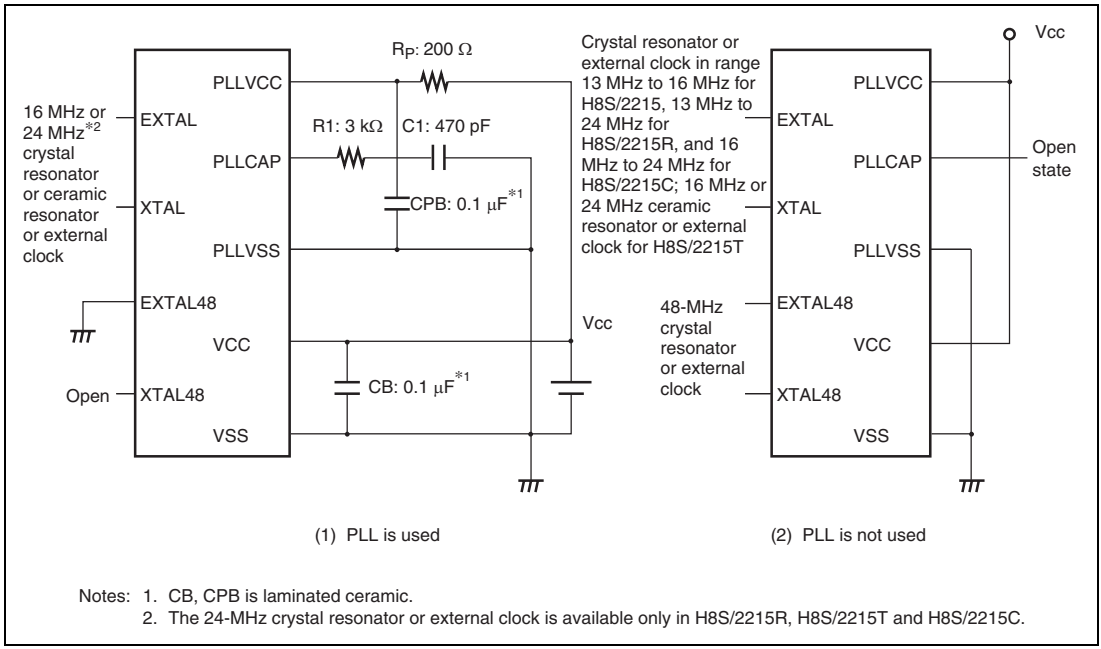
Table 19.5 System Clock Frequencies for Which Automatic Adjustment of LSI Bit Rate Is Possible

Host Bit Rate	System Clock Frequency Range of LSI
19,200 bps	HD64F2215: 13 to 16 MHz
9,600 bps	HD64F2215R: 13 to 24 MHz
4,800 bps	HD64F2215T: 16 MHz and 24 MHz

21.7 PLL Circuit for USB

The PLL circuit has the function of tripling or doubling* the 16- or 24-MHz* clock from the system oscillator to generate the 48-MHz USB operating clock. When the PLL circuit is used, set the UCKS3 to UCKS0 bits of UCTLR. For details, refer to section 15, Universal Serial Bus Interface (USB). When the PLL circuit is not used, connect the PLLVCC pin to VCC, and leave the PLLCAP pin open as shown in figure 21.11.

Note: * Available only in H8S/2215R, H8S/2215T and H8S/2215C.



Notes: 1. CB, CPB is laminated ceramic.
2. The 24-MHz crystal resonator or external clock is available only in H8S/2215R, H8S/2215T and H8S/2215C.

Figure 21.11 Example of PLL Circuit

When designing the board, place the capacitor C1 and resistor R1 as close as possible to the PLLCAP pin. Other signal lines should be routed away from the oscillator circuit to prevent induction from interfering with correct oscillation. C1 must be grounded to PLLVSS. In addition, PLLVCC and PLLVSS must be separated from the VCC and VSS pins. Bypass capacitors CPB and CB must be connected between VCC and VSS and between PLVCC and PLVSS, respectively.

Register Name	Abbreviation	Number of Bits	Address	Data Bus Width	Number of Access States	Module
Timer general register B_0	TGRB_0	16	H'FF1A	16	2	TPU_0
Timer general register C_0	TGRC_0	16	H'FF1C	16	2	
Timer general register D_0	TGRD_0	16	H'FF1E	16	2	
Timer control register_1	TCR_1	8	H'FF20	16	2	TPU_1
Timer mode register_1	TMDR_1	8	H'FF21	16	2	
Timer I/O control register _1	TIOR_1	8	H'FF22	16	2	
Timer interrupt enable register _1	TIER_1	8	H'FF24	16	2	
Timer status register_1	TSR_1	8	H'FF25	16	2	
Timer counter_1	TCNT_1	16	H'FF26	16	2	
Timer general register A_1	TGRA_1	16	H'FF28	16	2	
Timer general register B_1	TGRB_1	16	H'FF2A	16	2	
Timer control register_2	TCR_2	8	H'FF30	16	2	TPU_2
Timer mode register_2	TMDR_2	8	H'FF31	16	2	
Timer I/O control register 2	TIOR_2	8	H'FF32	16	2	
Timer interrupt enable register 2	TIER_2	8	H'FF34	16	2	
Timer status register_2	TSR_2	8	H'FF35	16	2	
Timer counter_2	TCNT_2	16	H'FF36	16	2	
Timer general register A_2	TGRA_2	16	H'FF38	16	2	
Timer general register B_2	TGRB_2	16	H'FF3A	16	2	
DMA write enable register	DMAWER	8	H'FF60	8	2	DMAC
DMA control register 0A	DMACR0A	8	H'FF62	16	2	
DMA control register 0B	DMACR0B	8	H'FF63	16	2	
DMA control register 1A	DMACR1A	8	H'FF64	16	2	
DMA control register 1B	DMACR1B	8	H'FF65	16	2	
DMA band control register	DMABCR	16	H'FF66	16	2	
Timer control register_0	TCR_0	8	H'FF68	8	2	TMR_0
Timer control register_1	TCR_1	8	H'FF69	8	2	TMR_1
Timer control/status register_0	TCSR_0	8	H'FF6A	8	2	TMR_0
Timer control/status register_1	TCSR_1	8	H'FF6B	8	2	TMR_1
Time constant register A0	TCORA_0	8	H'FF6C	8	2	TMR_0
Time constant register A1	TCORA_1	8	H'FF6D	8	2	TMR_1
Time constant register B0	TCORB_0	8	H'FF6E	8	2	TMR_0

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TCR_0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_0
TMDR_0	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
TIORH_0	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_0	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
TIER_0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TSR_0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRA_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRB_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRC_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRD_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCR_1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_1
TMDR_1	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRA_1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRB_1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Register Name	Power-on Reset	Manual Reset	High-Speed	Medium-Speed	Sleep	Module Stop	Software Standby	Hardware Standby	Module
DMAWER	Initialized	Initialized	—	—	—	—	—	Initialized	DMAC
DMACR0A	Initialized	Initialized	—	—	—	—	—	Initialized	
DMACR0B	Initialized	Initialized	—	—	—	—	—	Initialized	
DMACR1A	Initialized	Initialized	—	—	—	—	—	Initialized	
DMACR1B	Initialized	Initialized	—	—	—	—	—	Initialized	
DMABCR	Initialized	Initialized	—	—	—	—	—	Initialized	
TCR_0	Initialized	Initialized	—	—	—	—	—	Initialized	TMR_0
TCR_1	Initialized	Initialized	—	—	—	—	—	Initialized	TMR_1
TCSR_0	Initialized	Initialized	—	—	—	—	—	Initialized	TMR_0
TCSR_1	Initialized	Initialized	—	—	—	—	—	Initialized	TMR_1
TCORA_0	Initialized	Initialized	—	—	—	—	—	Initialized	TMR_0
TCORA_1	Initialized	Initialized	—	—	—	—	—	Initialized	TMR_1
TCORB_0	Initialized	Initialized	—	—	—	—	—	Initialized	TMR_0
TCORB_1	Initialized	Initialized	—	—	—	—	—	Initialized	TMR_1
TCNT_0	Initialized	Initialized	—	—	—	—	—	Initialized	TMR_0
TCNT_1	Initialized	Initialized	—	—	—	—	—	Initialized	TMR_1
TCSR	Initialized	Initialized	—	—	—	—	—	Initialized	WDT
TCNT	Initialized	Initialized	—	—	—	—	—	Initialized	
RSTCSR	Initialized	Initialized	—	—	—	—	—	Initialized	
SMR_0	Initialized	Initialized	—	—	—	—	—	Initialized	SCI_0
BRR_0	Initialized	Initialized	—	—	—	—	—	Initialized	
SCR_0	Initialized	Initialized	—	—	—	—	—	Initialized	
TDR_0	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	
SSR_0	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	
RDR_0	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	
SCMR_0	Initialized	Initialized	—	—	—	—	—	Initialized	
SMR_1	Initialized	Initialized	—	—	—	—	—	Initialized	SCI_1
BRR_1	Initialized	Initialized	—	—	—	—	—	Initialized	
SCR_1	Initialized	Initialized	—	—	—	—	—	Initialized	
TDR_1	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	
SSR_1	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	
RDR_1	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	
SCMR_1	Initialized	Initialized	—	—	—	—	—	Initialized	