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	H8S/2000
Core Size	32-Bit
Speed	24MHz
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	20K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 6x10b SAR; D/A 2x8b
Oscillator Type	Internal
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/df2215rute24wv

2.2 CPU Operating Modes

The H8S/2000 CPU has two operating modes: normal and advanced. Normal mode supports a maximum 64-kbyte address space. Advanced mode supports a maximum 16-Mbyte total address space. The mode is selected by the mode pins.

2.2.1 Normal Mode

The exception vector table and stack have the same structure as in the H8/300 CPU.

- Address Space

  A maximum address space of 64 kbytes can be accessed.
- Extended Registers (En)

  The extended registers (E0 to E7) can be used as 16-bit registers, or as the upper 16-bit segments of 32-bit registers. When En is used as a 16-bit register it can contain any value, even when the corresponding general register (Rn) is used as an address register. If the general register is referenced in the register indirect addressing mode with pre-decrement (@–Rn) or post-increment (@Rn+) and a carry or borrow occurs, however, the value in the corresponding extended register (En) will be affected.
- Instruction Set

  All instructions and addressing modes can be used. Only the lower 16 bits of effective addresses (EA) are valid.
- Exception Vector Table and Memory Indirect Branch Addresses

  In normal mode the top area starting at H'0000 is allocated to the exception vector table. One branch address is stored per 16 bits. The exception vector table in normal mode is shown in figure 2.1. For details of the exception vector table, see section 4, Exception Handling.

  The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instructions uses an 8-bit absolute address included in the instruction code to specify a memory operand that contains a branch address. In normal mode the operand is a 16-bit (word) operand, providing a 16-bit branch address. Branch addresses can be stored in the top area from H'0000 to H'00FF. Note that this area is also used for the exception vector table.
- Stack Structure

  When the program counter (PC) is pushed onto the stack in a subroutine call, and the PC, condition-code register (CCR), and extended control register (EXR) are pushed onto the stack in exception handling, they are stored as shown in figure 2.2. EXR is not pushed onto the stack in interrupt control mode 0. For details, see section 4, Exception Handling.

Note: Normal mode is not available in this LSI.

5.3 Register Descriptions

The interrupt controller has the following registers. For details on the system control register, refer to section 3.2.2, System Control Register (SYSCR).

- System control register (SYSCR)
- IRQ sense control register H (ISCRH)
- IRQ sense control register L (ISCRL)
- IRQ enable register (IER)
- IRQ status register (ISR)
- Interrupt priority register A (IPRA)
- Interrupt priority register B (IPRB)
- Interrupt priority register C (IPRC)
- Interrupt priority register D (IPRD)
- Interrupt priority register E (IPRE)
- Interrupt priority register F (IPRF)
- Interrupt priority register G (IPRG)
- Interrupt priority register I (IPRI)
- Interrupt priority register J (IPRJ)
- Interrupt priority register K (IPRK)
- Interrupt priority register M (IPRM)

5.6.2 Interrupt Control Mode 2

In interrupt control mode 2, mask control is done in eight levels for interrupt requests except for NMI by comparing the EXR interrupt mask level (I2 to I0 bits) in the CPU and the IPR setting.

Figure 5.5 shows a flowchart of the interrupt acceptance operation in this case.

1. If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
2. When interrupt requests are sent to the interrupt controller, the interrupt with the highest priority according to the interrupt priority levels set in IPR is selected, and lower-priority interrupt requests are held pending. If a number of interrupt requests with the same priority are generated at the same time, the interrupt request with the highest priority according to the priority system shown in table 5.2 is selected.
3. Next, the priority of the selected interrupt request is compared with the interrupt mask level set in EXR. An interrupt request with a priority no higher than the mask level set at that time is held pending, and only an interrupt request with a priority higher than the interrupt mask level is accepted.
4. When the CPU accepts an interrupt request, it starts interrupt exception handling after execution of the current instruction has been completed.
5. The PC, CCR, and EXR are saved to the stack area by interrupt exception handling. The PC saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.
6. The T bit in EXR is cleared to 0. The interrupt mask level is rewritten with the priority level of the accepted interrupt.

   If the accepted interrupt is NMI, the interrupt mask level is set to H'7.
7. The CPU generates a vector address for the accepted interrupt and starts execution of the interrupt handling routine at the address indicated by the contents of the vector address in the vector table.

- WCRL

Bit	Bit Name	Initial Value	R/W	Description
7	W31	1	R/W	Area 3 Wait Control 1 and 0
6	W30	1	R/W	These bits select the number of program wait states when area 3 in external space is accessed while the AST3 bit in ASTCR is set to 1. 00: Program wait not inserted when external space area 3 is accessed 01: 1 program wait state inserted when external space area 3 is accessed 10: 2 program wait states inserted when external space area 3 is accessed 11: 3 program wait states inserted when external space area 3 is accessed
5	W21	1	R/W	Area 2 Wait Control 1 and 0
4	W20	1	R/W	These bits select the number of program wait states when area 2 in external space is accessed while the AST2 bit in ASTCR is set to 1. 00: Program wait not inserted when external space area 2 is accessed 01: 1 program wait state inserted when external space area 2 is accessed 10: 2 program wait states inserted when external space area 2 is accessed 11: 3 program wait states inserted when external space area 2 is accessed
3	W11	1	R/W	Area 1 Wait Control 1 and 0
2	W10	1	R/W	These bits select the number of program wait states when area 1 in external space is accessed while the AST1 bit in ASTCR is set to 1. 00: Program wait not inserted when external space area 1 is accessed 01: 1 program wait state inserted when external space area 1 is accessed 10: 2 program wait states inserted when external space area 1 is accessed 11: 3 program wait states inserted when external space area 1 is accessed

Bit	Bit Name	Initial Value	R/W	Description
1	W01	1	R/W	Area 0 Wait Control 1 and 0
0	W00	1	R/W	These bits select the number of program wait states when area 0 in external space is accessed while the AST0 bit in ASTCR is set to 1. 00: Program wait not inserted when external space area 0 is accessed 01: 1 program wait state inserted when external space area 0 is accessed 10: 2 program wait states inserted when external space area 0 is accessed 11: 3 program wait states inserted when external space area 0 is accessed

Write after Read: If an external write occurs after an external read while the ICIS0 bit in BCRH is set to 1, an idle cycle is inserted at the start of the write cycle.

Figure 6.23 shows an example of the operation in this case. In this example, bus cycle A is a read cycle from ROM with a long output floating time, and bus cycle B is a CPU write cycle. In (a), an idle cycle is not inserted, and a collision occurs in cycle B between the read data from ROM and the CPU write data. In (b), an idle cycle is inserted, and a data collision is prevented.

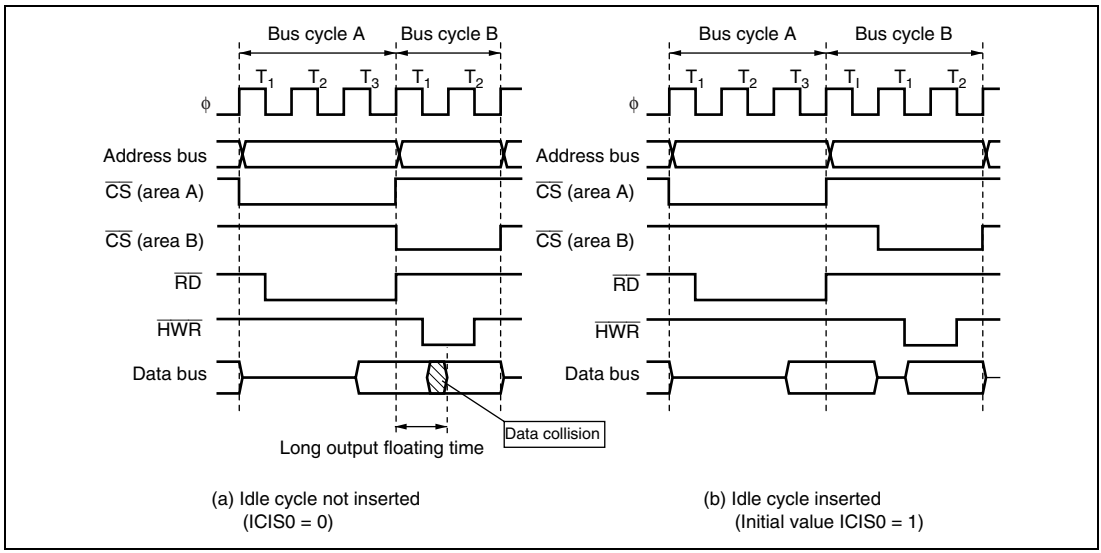

Figure 6.23 Example of Idle Cycle Operation (2)

10.3 Register Descriptions

The TPU has the following registers.

- Timer control register_0 (TCR_0)
- Timer mode register_0 (TMDR_0)
- Timer I/O control register H_0 (TIORH_0)
- Timer I/O control register L_0 (TIORL_0)
- Timer interrupt enable register_0 (TIER_0)
- Timer status register_0 (TSR_0)
- Timer counter_0 (TCNT_0)
- Timer general register A_0 (TGRA_0)
- Timer general register B_0 (TGRB_0)
- Timer general register C_0 (TGRC_0)
- Timer general register D_0 (TGRD_0)
- Timer control register_1 (TCR_1)
- Timer mode register_1 (TMDR_1)
- Timer I/O control register _1 (TIOR_1)
- Timer interrupt enable register_1 (TIER_1)
- Timer status register_1 (TSR_1)
- Timer counter_1 (TCNT_1)
- Timer general register A_1 (TGRA_1)
- Timer general register B_1 (TGRB_1)
- Timer control register_2 (TCR_2)
- Timer mode register_2 (TMDR_2)
- Timer I/O control register_2 (TIOR_2)
- Timer interrupt enable register_2 (TIER_2)
- Timer status register_2 (TSR_2)
- Timer counter_2 (TCNT_2)
- Timer general register A_2 (TGRA_2)
- Timer general register B_2 (TGRB_2)

Common Registers

- Timer start register (TSTR)
- Timer synchro register (TSYR)

Timing for Counter Clearing by Compare Match/Input Capture: Figure 10.34 shows the timing when counter clearing by compare match occurrence is specified, and figure 10.35 shows the timing when counter clearing by input capture occurrence is specified.

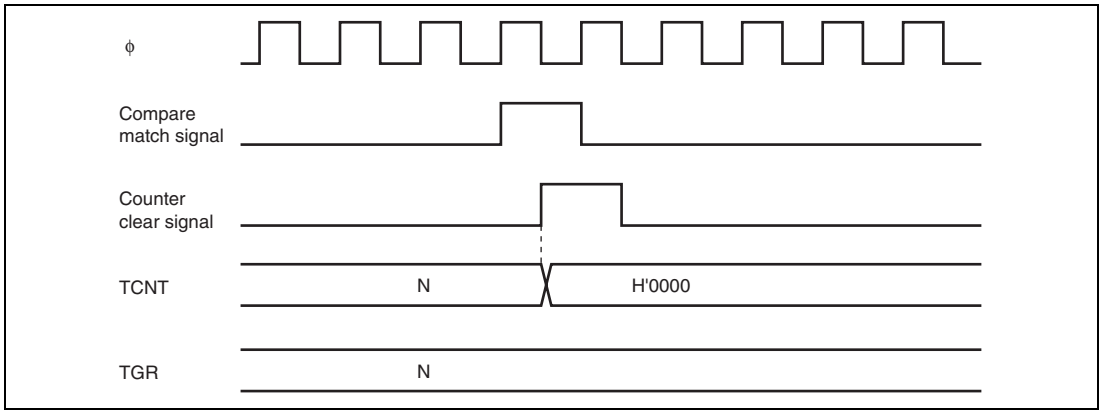

Figure 10.34 Counter Clear Timing (Compare Match)

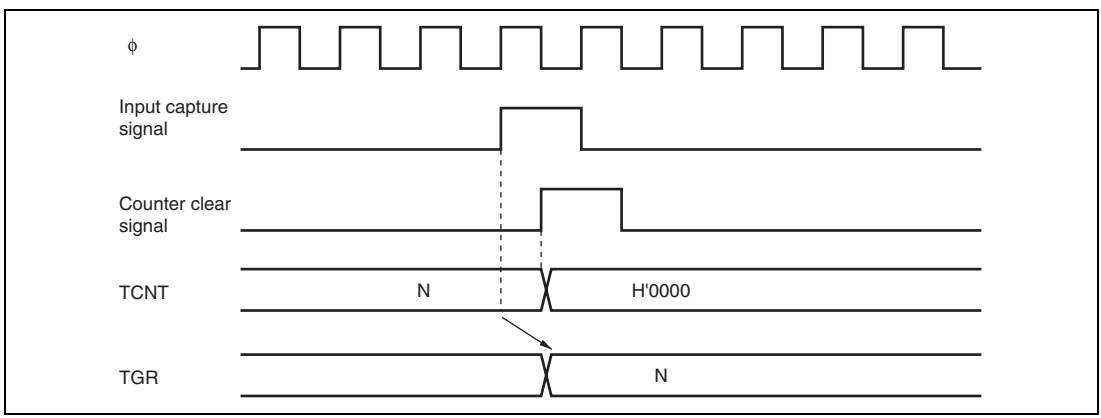

Figure 10.35 Counter Clear Timing (Input Capture)

11.3.2 Time Constant Registers A (TCORA)

The TCORA_0 and TCORA_1 registers are 8-bit readable/writable registers. TCORA_0 and TCORA_1 comprise a single 16-bit register so they can be accessed together by a word transfer instruction. The value in TCORA is continually compared with the value in TCNT. When a match is detected, the corresponding CMFA flag in TCSR is set to 1. Note, however, that comparison is disabled during the T_2 state of a TCOR write cycle. The timer output (TMO) can be freely controlled by these compare match signals and the settings of bits OS1 and OS0 in TCSR. TCORA_0 and TCORA_1 are each initialized to H'FF.

11.3.3 Time Constant Registers B (TCORB)

The TCORB_0 registers are 8-bit readable/writable registers. TCORB_0 and TCORB_1 comprise a single 16-bit register so they can be accessed together by a word transfer instruction. TCORB is continually compared with the value in TCNT. When a match is detected, the corresponding CMFB flag in TCSR is set to 1. Note, however, that comparison is disabled during the T_2 state of a TCOR write cycle. The timer output can be freely controlled by these compare match signals and the settings of output select bits OS3 and OS2 in TCSR. TCORB_0 and TCORB_1 are each initialized to H'FF.

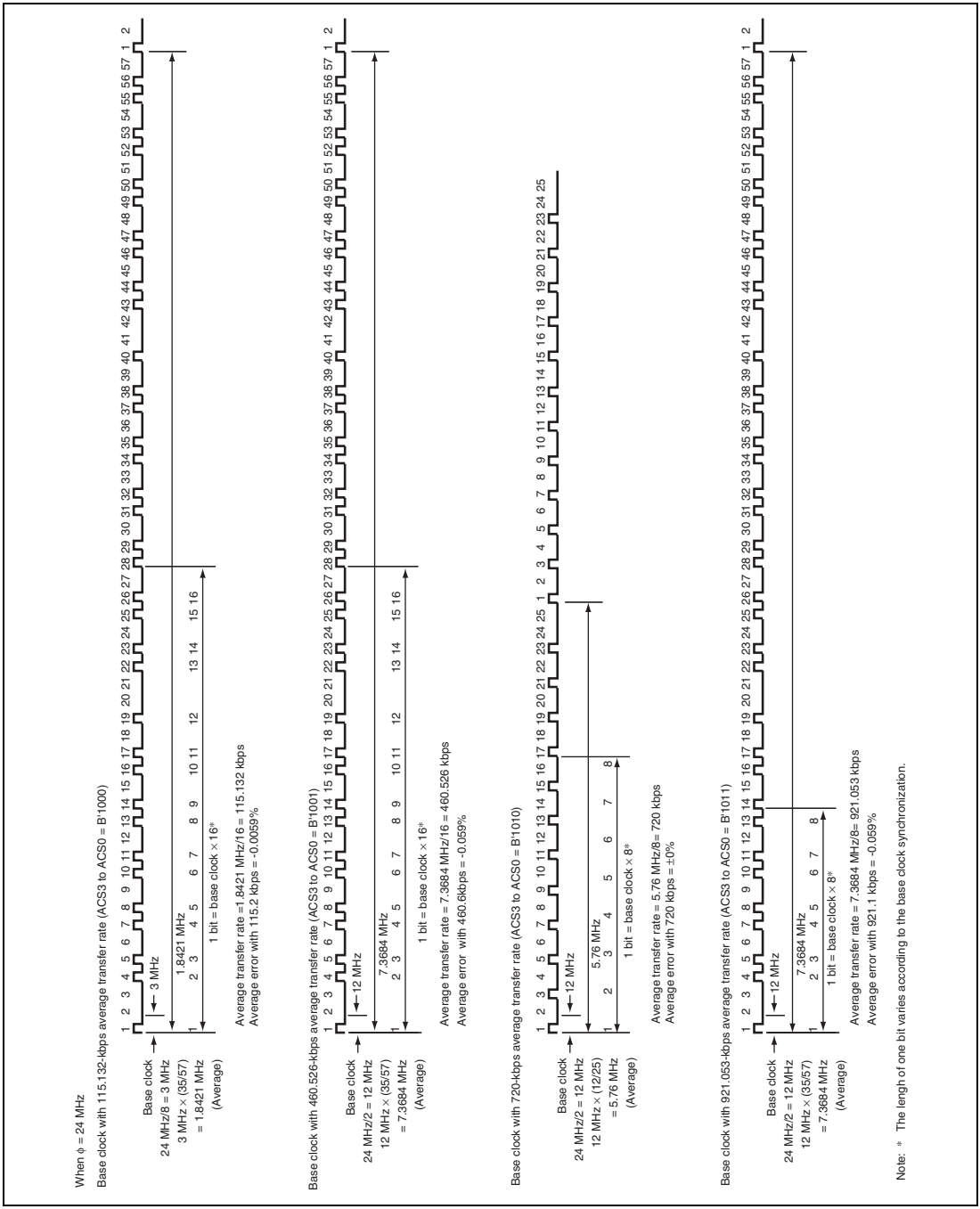

Figure 13.4 Examples of Base Clock when Average Transfer Rate Is Selected (2)

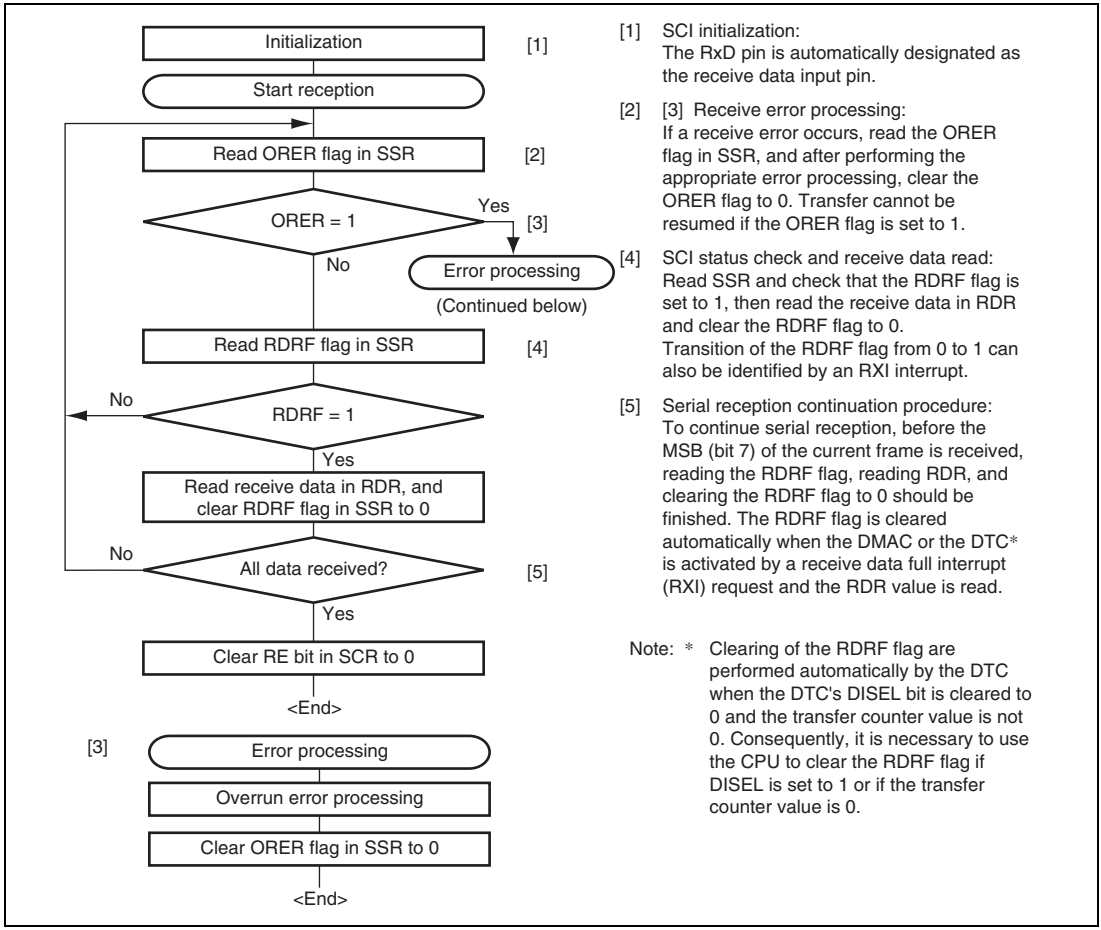

Figure 13.23 Sample Serial Reception Flowchart

13.6.5 Simultaneous Serial Data Transmission and Reception (Clocked Synchronous Mode)

Figure 13.24 shows a sample flowchart for simultaneous serial transmit and receive operations. The following procedure should be used for simultaneous serial data transmit and receive operations. To switch from transmit mode to simultaneous transmit and receive mode, after checking that the SCI has finished transmission and the TDRE and TEND flags are set to 1, clear TE to 0. Then simultaneously set TE and RE to 1 with a single instruction. To switch from receive mode to simultaneous transmit and receive mode, after checking that the SCI has finished reception, clear RE to 0. Then after checking that the RDRF and receive error flags (ORER, FER, and PER) are cleared to 0, simultaneously set TE and RE to 1 with a single instruction.

14.3.4 Boundary Scan Register (BSCANR)

BSCAN is a 217-bit shift register assigned on the pins to control input/output pins.

The I/O pins consists of three bits (IN, Control, OUT), input pins 1 bit (IN), and output pins 1 bit (OUT) of shift registers. The boundary scan test based on the JTAG standard can be performed by using instructions listed in table 14.2. Table 14.4 shows the correspondence between the LSI pins and boundary scan registers. (In table 14.4, Control indicates the high active pin. By specifying Control to high, the pin is driven by OUT.) Figure 14.2 shows the boundary scan register configuration example.

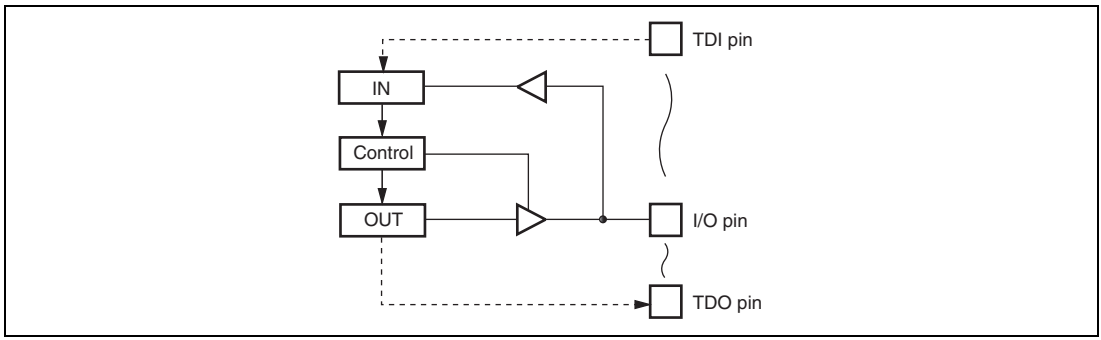

Figure 14.2 Boundary Scan Register Configuration

(2) Data Stage (Control-In)

The firmware first analyzes command data from the host in the setup stage, and determines the subsequent data stage direction. If the result of command data analysis is that the data stage is in-transfer, one packet of data to be sent to the host is written to the FIFO. If there is more data to be sent, this data is written to the FIFO after the data written first has been sent to the host (EP0iTS of UIFR0 is set to 1).

The end of the data stage is identified when the host transmits an OUT token and the status stage is entered.

(5) Status Stage (Control-Out)

The control-out status stage starts with an IN token from the host. When an IN-token is received at the start of the status stage, there is not yet any data in the EP0iFIFO, and so an EP0i transfer request interrupt is generated. The application recognizes from this interrupt that the status stage has started. Next, in order to transmit 0-byte data to the host, 1 is written to the EP0i packet enable bit but no data is written to the EP0iFIFO. As a result, the next IN token causes 0-byte data to be transmitted to the host, and control transfer ends.

After the application has finished all processing relating to the data stage, 1 should be written to the EP0i packet enable bit.

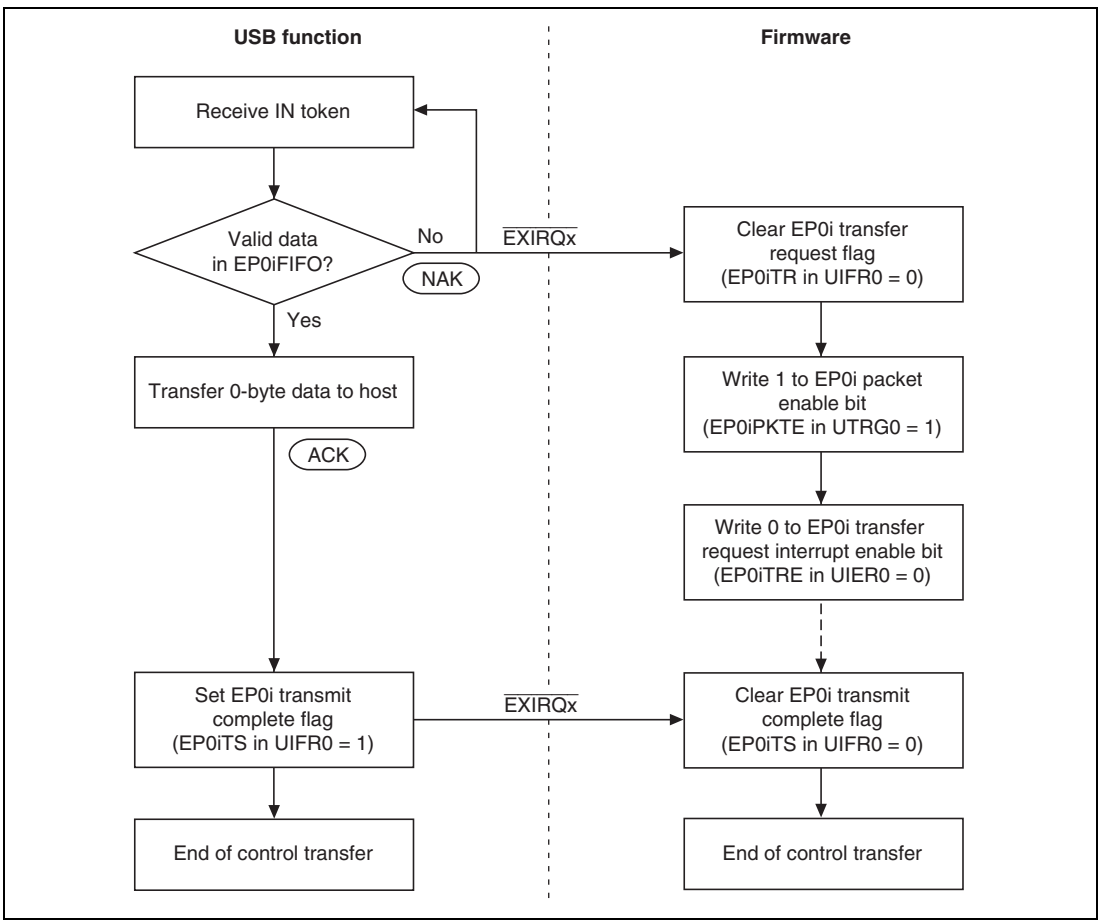

Figure 15.17 Status Stage Operation (Control-Out)

Table 22.1 LSI Internal States in Each Mode

Function		High-Speed	Medium-Speed	Sleep	Module Stop	Software Standby	Hardware Standby
System clock pulse generator		Functioning	Functioning	Functioning	Functioning	Halted	Halted
CPU	Instructions	Functioning	Medium-speed operation	Halted	High/ medium-speed operation	Halted (retained)	Halted (undefined)
	Registers			(retained)			
External interrupts	NMI	Functioning	Functioning	Functioning	Functioning	Functioning	Halted
	IRQ0 to 5, 7						
Peripheral functions	DMAC	Functioning	Medium-speed operation	Functioning	Halted (retained)	Halted (retained)	Halted (reset)
	DTC						
	I/O	Functioning	Functioning	Functioning	Functioning	Retained	High impedance
	TPU	Functioning	Functioning	Functioning	Halted (retained)	Halted (retained)	Halted (reset)
	TMR						
	WDT	Functioning	Functioning	Functioning	Functioning	Halted (retained)	Halted (reset)
	D/A	Functioning	Functioning	Functioning	Halted (retained)	Halted (retained)	Halted (reset)
	A/D	Functioning	Functioning	Functioning	Halted (reset)	Halted (reset)	Halted (reset)
	SCI						
	USB	Functioning	Function not guaranteed	Functioning	Halted (retained)	Halted (retained)	Halted (reset)
	USB operating clock oscillator				Halted	Halted	
	PLL circuit						
	RAM	Functioning	Functioning	Functioning	Functioning	Retained	Retained

Notes: “Halted (retained)” means that internal register values are retained. The internal state is “operation suspended.”

“Halted (reset)” means that internal register values and internal states are initialized.

In module stop mode, only modules for which a stop setting has been made are halted (reset or retained).

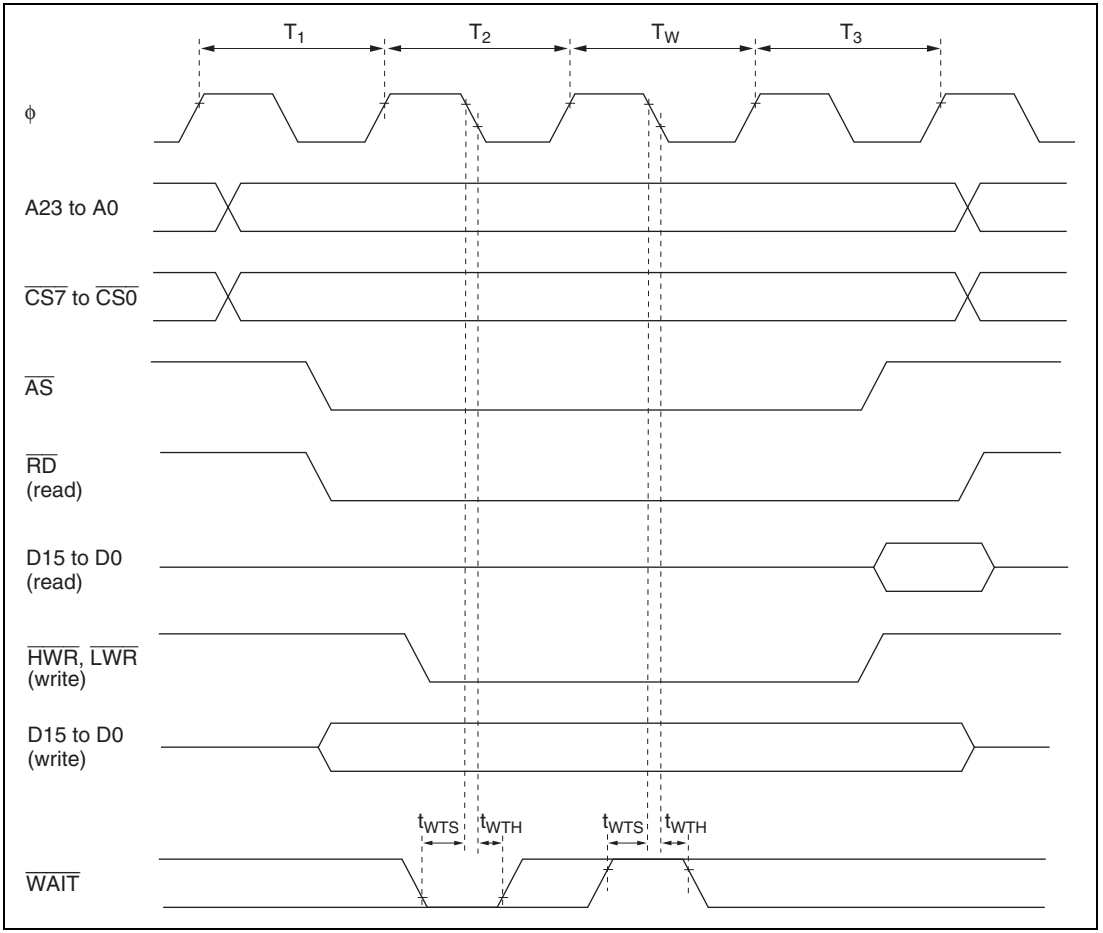

Figure 24.9 Basic Bus Timing (Three-State Access with One Wait State)

25.4.4 Timing of On-Chip Supporting Modules

Table 25.7 lists the timing of on-chip supporting modules.

Table 25.7 Timing of On-Chip Supporting Modules

Condition A: V_{CC} = $PLLV_{CC}$ = DrV_{CC} = 2.7 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V, V_{ref} = 2.7 V to AV_{CC}, V_{SS} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 13 MHz to 16 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Condition B: V_{CC} = $PLLV_{CC}$ = DrV_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 13 MHz to 24 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

			Condition A		Condition B			Test
Item		**Symbol**	**Min.**	**Max.**	**Min.**	**Max.**	**Unit**	**Conditions**
I/O port	Output data delay time	t_{PWD}	—	60	—	40	ns	Figure 25.12
	Input data setup time	t_{PRS}	50	—	30	—		
	Input data hold time	t_{PRH}	50	—	30	—		
TPU	Timer output delay time	t_{TOCD}	—	60	—	40	ns	Figure 25.13
	Timer input setup time	t_{TICS}	40	—	30	—		
	Timer clock input setup time	t_{TCKS}	40	—	30	—	ns	Figure 25.14
	Timer clock pulse width Single edge	t_{TCKWH}	1.5	—	1.5	—	t_{cyc}	
	Timer clock pulse width Both edges	t_{TCKWL}	2.5	—	2.5	—		
TMR	Timer output delay time	t_{TMOD}	—	60	—	41	ns	Figure 25.15
	Timer reset input setup time	t_{TMRS}	50	—	29	—	ns	Figure 25.17
	Timer clock input setup time	t_{TMCS}	50	—	29	—	ns	Figure 25.16

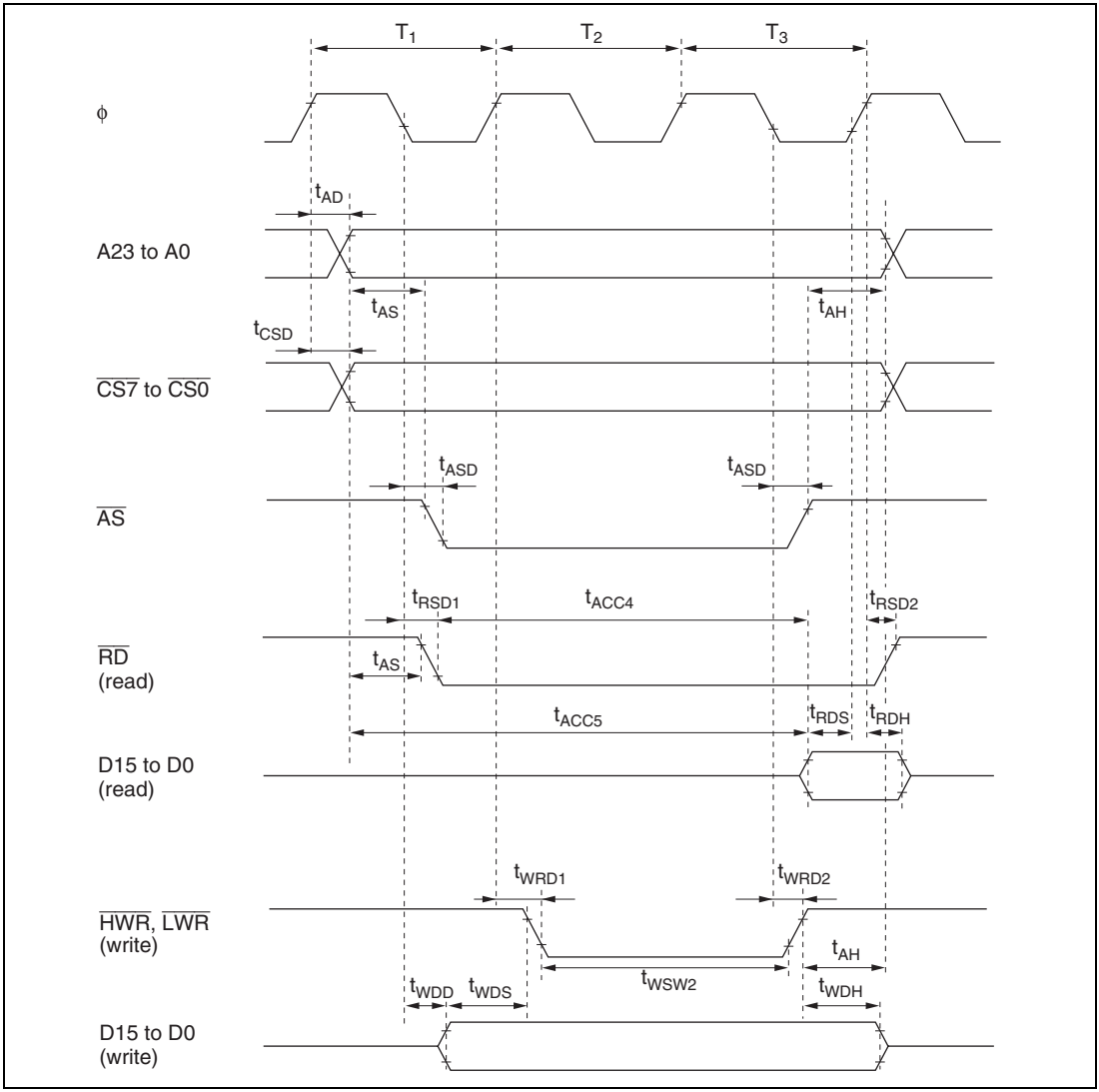

Figure 26.8 Basic Bus Timing (Three-State Access)