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	Not For New Designs
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/df2215ute16v

Section 14 Boundary Scan Function

Section 15 Universal Serial Bus Interface (USB)

Section 16 A/D Converter

Section 17 D/A Converter

Section 19 Flash Memory (F-ZTAT Version)

2.4.4 Condition-Code Register (CCR)

This 8-bit register contains internal CPU status information, including an interrupt mask bit (I) and half-carry (H), negative (N), zero (Z), overflow (V), and carry (C) flags.

Operations can be performed on the CCR bits by the LDC, STC, ANDC, ORC, and XORC instructions. The N, Z, V, and C flags are used as branching conditions for conditional branch (Bcc) instructions.

Bit	Bit Name	Initial Value	R/W	Description
7	I	1	R/W	Interrupt Mask Bit Masks interrupts other than NMI when set to 1. NMI is accepted regardless of the I bit setting. The I bit is set to 1 by hardware at the start of an exception-handling sequence. For details, refer to section 5, Interrupt Controller.
6	UI	undefined	R/W	User Bit or Interrupt Mask Bit Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions. This bit cannot be used as an interrupt mask bit in this LSI.
5	H	undefined	R/W	Half-Carry Flag When the ADD.B, ADDX.B, SUB.B, SUBX.B, CMP.B or NEG.B instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 3, and cleared to 0 otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 11, and cleared to 0 otherwise. When the ADD.L, SUB.L, CMP.L, or NEG.L instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 27, and cleared to 0 otherwise.
4	U	undefined	R/W	User Bit Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions.
3	N	undefined	R/W	Negative Flag Stores the value of the most significant bit of data as a sign bit.
2	Z	undefined	R/W	Zero Flag Set to 1 to indicate zero data, and cleared to 0 to indicate non-zero data.

2.7.1 Register Direct—Rn

The register field of the instruction specifies an 8-, 16-, or 32-bit general register containing the operand. R0H to R7H and R0L to R7L can be specified as 8-bit registers. R0 to R7 and E0 to E7 can be specified as 16-bit registers. ER0 to ER7 can be specified as 32-bit registers.

2.7.2 Register Indirect—@ERn

The register field of the instruction code specifies an address register (ERn) which contains the address of the operand on memory. If the address is a program instruction address, the lower 24 bits are valid and the upper 8 bits are all assumed to be 0 (H'00).

2.7.3 Register Indirect with Displacement—@(d:16, ERn) or @(d:32, ERn)

A 16-bit or 32-bit displacement contained in the instruction is added to an address register (ERn) specified by the register field of the instruction, and the sum gives the address of a memory operand. A 16-bit displacement is sign-extended when added.

2.7.4 Register Indirect with Post-Increment or Pre-Decrement—@ERn+ or @-ERn

Register indirect with post-increment—@ERn+: The register field of the instruction code specifies an address register (ERn) which contains the address of a memory operand. After the operand is accessed, 1, 2, or 4 is added to the address register contents and the sum is stored in the address register. The value added is 1 for byte access, 2 for word transfer instruction, or 4 for longword transfer instruction. For word or longword transfer instruction, the register value should be even.

Register indirect with pre-decrement—@-ERn: The value 1, 2, or 4 is subtracted from an address register (ERn) specified by the register field in the instruction code, and the result becomes the address of a memory operand. The result is also stored in the address register. The value subtracted is 1 for byte access, 2 for word transfer instruction, or 4 for longword transfer instruction. For word or longword transfer instruction, the register value should be even.

2.7.5 Absolute Address—@aa:8, @aa:16, @aa:24, or @aa:32

The instruction code contains the absolute address of a memory operand. The absolute address may be 8 bits long (@aa:8), 16 bits long (@aa:16), 24 bits long (@aa:24), or 32 bits long (@aa:32). Table 2.12 indicates the accessible absolute address ranges.

To access data, the absolute address should be 8 bits (@aa:8), 16 bits (@aa:16), or 32 bits (@aa:32) long. For an 8-bit absolute address, the upper 24 bits are all assumed to be 1 (H'FFFF).

Bit	Bit Name	Initial Value	R/W	Description
9	IRQ4SCB	0	R/W	IRQ4 Sense Control B
8	IRQ4SCA	0	R/W	IRQ4 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ4}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ4}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ4}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ4}}$ input
7	IRQ3SCB	0	R/W	IRQ3 Sense Control B
6	IRQ3SCA	0	R/W	IRQ3 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ3}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ3}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ3}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ3}}$ input
5	IRQ2SCB	0	R/W	IRQ2 Sense Control B
4	IRQ2SCA	0	R/W	IRQ2 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ2}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ2}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ2}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ2}}$ input
3	IRQ1SCB	0	R/W	IRQ1 Sense Control B
2	IRQ1SCA	0	R/W	IRQ1 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ1}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ1}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ1}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ1}}$ input

5.7.2 Instructions that Disable Interrupts

Instructions that disable interrupts are LDC, ANDC, ORC, and XORC. After any of these instructions is executed, all interrupts including NMI are disabled and the next instruction is always executed. When the I bit is set by one of these instructions, the new value becomes valid two states after execution of the instruction ends.

5.7.3 Times when Interrupts Are Disabled

There are times when interrupt acceptance is disabled by the interrupt controller.

The interrupt controller disables interrupt acceptance for a 3-state period after the CPU has updated the mask level with an LDC, ANDC, ORC, or XORC instruction.

5.7.4 Interrupts during Execution of EEPMOV Instruction

Interrupt operation differs between the EEPMOV.B instruction and the EEPMOV.W instruction.

With the EEPMOV.B instruction, an interrupt request (including NMI) issued during the transfer is not accepted until the move is completed.

With the EEPMOV.W instruction, if an interrupt request is issued during the transfer, interrupt exception handling starts at a break in the transfer cycle. The PC value saved on the stack in this case is the address of the next instruction.

Therefore, if an interrupt is generated during execution of an EEPMOV.W instruction, the following coding should be used.

```
L1: EEPMOV.W
    MOV.W  R4,R4
    BNE    L1
```

5.7.5 IRQ Interrupt

During clock operation, IRQ input is accepted in synchronization with the clock.

In software standby mode, non-synchronous input is accepted.

For details of the input conditions, see the Control Signal Timing description in the Electrical Characteristics section for the product in question.

8.2 Register Descriptions

DTC has the following registers.

- DTC mode register A (MRA)
- DTC mode register B (MRB)
- DTC source address register (SAR)
- DTC destination address register (DAR)
- DTC transfer count register A (CRA)
- DTC transfer count register B (CRB)

These six registers cannot be directly accessed from the CPU. When activated, the DTC reads a set of register information that is stored in an on-chip RAM to the corresponding DTC registers and transfers data. After the data transfer, it writes a set of updated register information back to the RAM.

- DTC enable registers (DTCERA to DTCERF)
- DTC vector register (DTVECR)

8.2.2 DTC Mode Register B (MRB)

MRB selects the DTC operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	CHNE	Undefined	—	DTC Chain Transfer Enable This bit specifies a chain transfer. For details, refer to section 8.5.4, Chain Transfer. In data transfer with CHNE set to 1, determination of the end of the specified number of transfers, clearing of the interrupt source flag, and clearing of DTCER, are not performed. 0: DTC data transfer completed (waiting for start) 1: DTC chain transfer (reads new register information and transfers data)
6	DISEL	Undefined	—	DTC Interrupt Select This bit specifies whether CPU interrupt is disabled or enabled after a data transfer. 0: Interrupt request is issued to the CPU when the specified data transfer is completed. 1: DTC issues interrupt request to the CPU in every data transfer (DTC does not clear the interrupt request flag that is a cause of the activation).
5 to 0	—	Undefined	—	Reserved These bits have no effect on DTC operation, and the write value should always be 0.

8.2.3 DTC Source Address Register (SAR)

SAR is a 24-bit register that designates the source address of data to be transferred by the DTC. For word-size transfer, specify an even source address.

8.2.4 DTC Destination Address Register (DAR)

DAR is a 24-bit register that designates the destination address of data to be transferred by the DTC. For word-size transfer, specify an even destination address.

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.

9.11 Port F

Port F is an 8-bit I/O port that also has external interrupt input ($\overline{IRQ2}$, $\overline{IRQ3}$), bus control sign I/O, system clock output. The port F has the following registers.

- Port F data direction register (PFDDR)
- Port F data register (PFDR)
- Port F register (PORTF)

9.11.1 Port F Data Direction Register (PFDDR)

The individual bits of PFDDR specify input or output for the pins of port F. Since this is a write-only register, bit manipulation instructions should not be used to write to it. For details, see section 2.9.4, Accessing Registers Containing Write-Only Bits.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7DDR	1/0*	W	Modes 4 to 6 Pin PF7 functions as the φ output pin when the corresponding PFDDR bit is set to 1, and as an input port when the bit is cleared to 0. The input/output direction specification in PFDDR is ignored for pins PF6 to PF3, which are automatically designated as bus control outputs. Pins PF2 to PF0 are made bus control input/output pins by bus controller settings. Otherwise, setting a PFDDR bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port. Mode 7 Setting a PFDDR bit to 1 makes the corresponding port F pin PF6 to PF0 an output port, or in the case of pin PF7, the φ output pin. Clearing the bit to 0 makes the pin an input port.
6	PF6DDR	0	W	
5	PF5DDR	0	W	
4	PF4DDR	0	W	
3	PF3DDR	0	W	
2	PF2DDR	0	W	
1	PF1DDR	0	W	
0	PF0DDR	0	W	

Note: * In modes 4 to 6, set to 1; in mode 7 cleared to 0.

9.13 Handling of Unused Pins

Unused input pins should be fixed high or low. Generally, the input pins of CMOS products are high-impedance. Leaving unused pins open can cause the generation of intermediate levels due to peripheral noise induction. This can result in shoot-through current inside the device and cause it to malfunction. Table 9.76 lists examples of ways to handle unused pins.

For the handling of dedicated boundary scan pins that are unused, see section 14.2, Pin Configuration, and section 14.5, Usage Notes. For the handling of dedicated USB pins that are unused, see section 15.9.14, Pin Processing when USB Not Used.

Table 9.76 Examples of Ways to Handle Unused Input Pins

Pin Name	Pin Handling Example
Port 1	Connect each pin to Vcc (pull-up) or to Vss (pull-down) via a resistor.
Port 3	
Port 4	Connect each pin to AVcc (pull-up) or to AVss (pull-down) via a resistor.
Port 7	Connect each pin to Vcc (pull-up) or to Vss (pull-down) via a resistor.
Port 9	Connect each pin to AVcc (pull-up) or to AVss (pull-down) via a resistor.
Port A	Connect each pin to Vcc (pull-up) or to Vss (pull-down) via a resistor.
Port B	
Port C	
Port D	
Port E	
Port F	
Port G	

3. Phase counting mode 3

Figure 10.28 shows an example of phase counting mode 3 operation, and table 10.22 summarizes the TCNT up/down-count conditions.

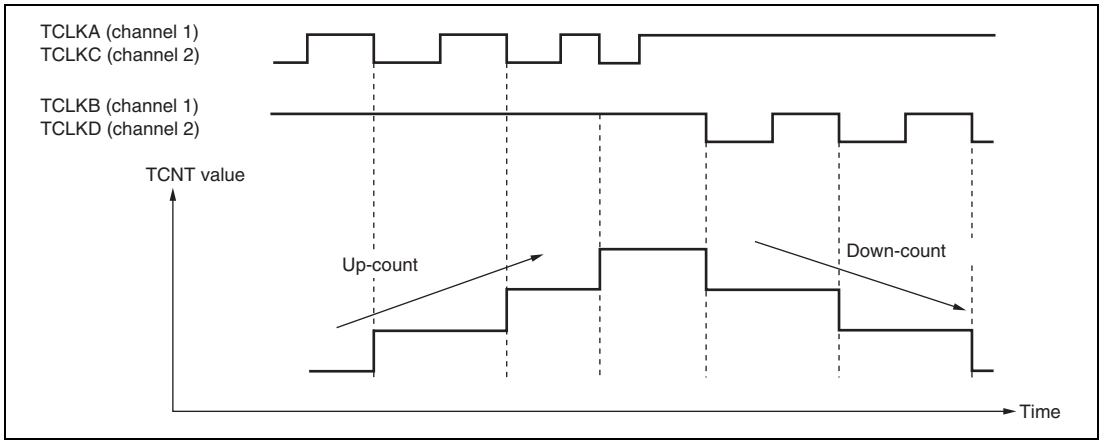

Figure 10.28 Example of Phase Counting Mode 3 Operation

Table 10.22 Up/Down-Count Conditions in Phase Counting Mode 3

TCLKA (Channel 1) / TCLKC (Channel 2)	TCLKB (Channel 1) / TCLKD (Channel 2)	Operation
High level	↑	Don't care
Low level	↓	
↑	Low level	
↓	High level	Up-count
High level	↓	Down-count
Low level	↑	Don't care
↑	High level	
↓	Low level	

Legend:
↑ : Rising edge
↓ : Falling edge

13.3.5 Serial Mode Register (SMR)

SMR is used to set the SCI's serial transfer format and select the baud rate generator clock source. Some bits in SMR have different functions in normal mode and smart card interface mode.

- Normal Serial Communication Interface Mode (When SMIF in SCMR is 0)

Bit	Bit Name	Initial Value	R/W	Description
7	C/$\overline{A}$	0	R/W	Communication Mode 0: Asynchronous mode 1: Clocked synchronous mode
6	CHR	0	R/W	Character Length (enabled only in asynchronous mode) 0: Selects 8 bits as the data length 1: Selects 7 bits as the data length. LSB-first is fixed and the MSB of TDR is not transmitted in transmission In clocked synchronous mode, a fixed data length of 8 bits is used.
5	PE	0	R/W	Parity Enable (enabled only in asynchronous mode) When this bit is set to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception. For a multiprocessor format, parity bit addition and checking are not performed regardless of the PE bit setting.
4	O/$\overline{E}$	0	R/W	Parity Mode (enabled only when the PE bit is 1 in asynchronous mode) 0: Selects even parity 1: Selects odd parity
3	STOP	0	R/W	Stop Bit Length (enabled only in asynchronous mode) Selects the stop bit length in transmission. 0: 1 stop bit 1: 2 stop bits In reception, only the first stop bit is checked. If the second stop bit is 0, it is treated as the start bit of the next transmit character.

Bit	Bit Name	Initial Value	R/W	Description
2	TEND	1	R	Transmit End This bit is set to 1 when no error signal has been sent back from the receiving end and the next transmit data is ready to be transferred to TDR. [Setting conditions] • When the TE bit in SCR is 0 and the ERS bit is also 0 • When the ESR bit is 0 and the TDRE bit is 1 after the specified interval following transmission of 1-byte data. The timing of bit setting differs according to the register setting as follows: When GM = 0 and BLK = 0, 2.5 etu after transmission starts When GM = 0 and BLK = 1, 1.0 etu after transmission starts When GM = 1 and BLK = 0, 1.5 etu after transmission starts When GM = 1 and BLK = 1, 1.0 etu after transmission starts [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or the DTC*[2] is activated by a TXI interrupt and transfers transmission data to TDR
1	MPB	0	R	Multiprocessor Bit This bit is not used in Smart Card interface mode.
0	MPBT	0	R/W	Multiprocessor Bit Transfer Write 0 to this bit in Smart Card interface mode.

Notes: 1. The write value should always be 0 to clear the flag.
2. The clearing conditions using the DTC are that DISEL bit be cleared to 0 and the transfer counter value be other than 0.
3. To clear the flag by the CPU on the H8S/2215R, H8S/2215T, and H8S/2215C, reread the flag after writing 0 to it.

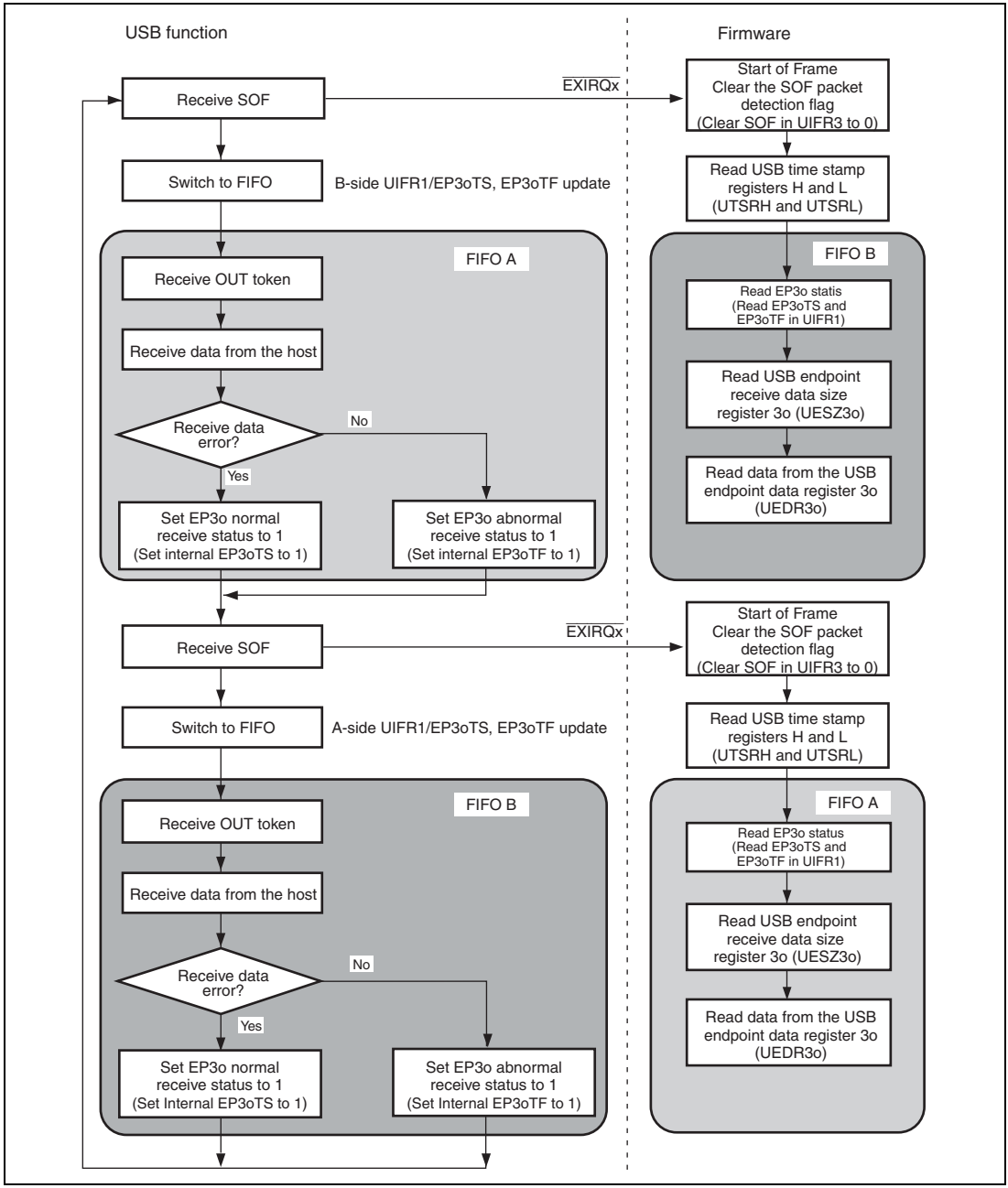

Figure 15.22 EP3o Isochronous-Out Transfer Operation

15.9.14 Pin Processing when USB Not Used

Pin processing should be performed as follows.

DrVCC = Vcc, DrVSS = 0 V, USD+ = USD- = USPND = open state, VBUS = $\overline{\text{UBPM}}$ = 0 V

15.9.15 Notes on Emulator Usage

Using the I/O register window function, or the like, to display UEDR0o, UEDR2o, UEDR3o, and UEDR4o can cause the EP0oFIFO, EP2oFIFO, EP3oFIFO, and EP4oFIFO read pointers to malfunction, preventing UEDR0o to UEDR4o and UESZ0o to UESZ4o from being read correctly. Therefore, UEDR0o to UEDR4o should not be displayed.

15.9.16 Notes on TR Interrupt

Note the following when using the transfer request interrupt (TR interrupt) for IN transfer to EP0i, EP2i, EP3i, EP4i, or EP5i.

The TR interrupt flag is set if the FIFO for the target EP has no data when the IN token is sent from the USB host. However, at the timing shown in figure 15.38, multiple TR interrupts occur successively. Take appropriate measures against malfunction in such a case.

Note: This module determines whether to return NAK if the FIFO of the target EP has no data when receiving the IN token, but the TR interrupt flag is set only after a NAK handshake is sent. If the next IN token is sent before PKTE of UTRGx is written to, the TR interrupt flag is set again.

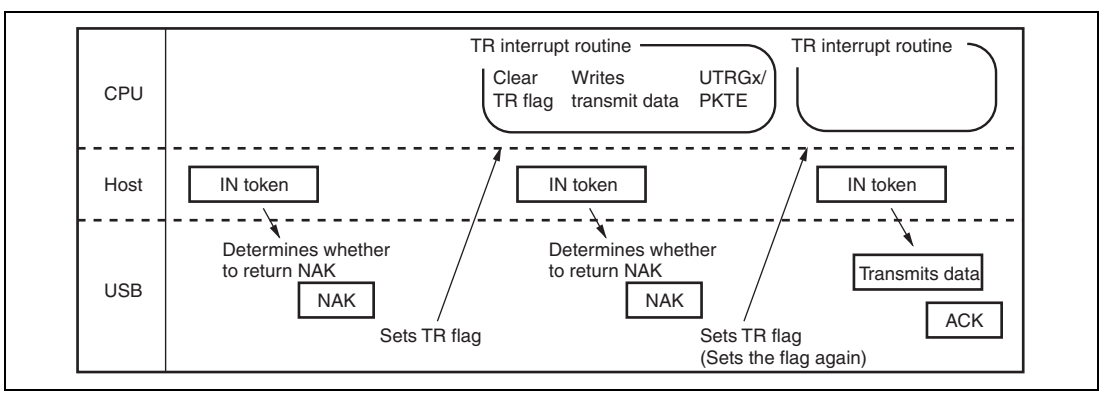

Figure 15.38 TR Interrupt Flag Set Timing

22.4.3 Setting Oscillation Stabilization Time after Clearing Software Standby Mode

Bits STS2 to STS0 in SBYCR should be set as described below.

- Using a Crystal Oscillator

  Set bits STS2 to STS0 so that the standby time is at least t_{OSC2} ms (the oscillation stabilization time).

  Table 22.3 shows the standby times for different operating frequencies and settings of bits STS2 to STS0.
- Using an External Clock

  Set bits STS2 to STS0 as any value. Usually, minimum value is recommended. A 16-state standby time cannot be used in the F-ZTAT version; a standby time of 2,048 states or longer should be used.

Table 22.3 Oscillation Stabilization Time Settings

STS2	STS1	STS0	Standby Time	24 MHz*2	20 MHz*1	16 MHz	13 MHz	10 MHz	8 MHz	6 MHz	4 MHz	2 MHz	Unit
0	0	0	8192 states	0.3	0.4	0.51	0.6	0.8	1.0	1.3	2.0	4.1	ms
		1	16384 states	0.7	0.8	1.0	1.3	1.6	2.0	2.7	4.1	8.2	
	1	0	32768 states	1.4	1.6	2.0	2.5	3.3	4.1	5.5	8.2	16.4	
		1	65536 states	2.7	3.3	4.1	5.0	6.6	8.2	10.9	16.4	32.8	
1	0	0	131072 states	5.5	6.6	8.2	10.1	13.1	16.4	21.8	32.8	65.5	
		1	262144 states	10.9	13.1	16.4	20.2	26.2	32.8	43.6	65.6	131.2	
	1	0	2048 states	0.09	0.1	0.13	0.16	0.2	0.3	0.3	0.5	1.0	
		1	16 states	0.7	0.8	1.0	1.2	1.6	2.0	1.7	4.0	8.0	µs

Notes: 1. Only in H8S/2215R and H8S/2215C.
2. Only in H8S/2215R, H8S/2215T and H8S/2215C.

: Recommended time setting (See the t_{OSC2} item in table 24.4 or table 25.4, for conditions)

24.3 DC Characteristics

Table 24.2 lists the DC characteristics. Table 24.3 lists the permissible output currents.

Table 24.2 DC Characteristics

Conditions: 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, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)*1

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	$\overline{IRQ0}$ to $\overline{IRQ5}$, $\overline{IRQ7}$	V_T^-	$V_{CC} \times 0.2$	—	—	V	
		V_T^+	—	—	$V_{CC} \times 0.8$	V	
		$V_T^+ - V_T^-$	$V_{CC} \times 0.05$	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD2 to MD0, $\overline{TRST}$, TCK, TMS, TDI, VBUS, $\overline{UBPM}$, FWE*5	V_{IH}	$V_{CC} \times 0.9$	—	V_{CC} + 0.3	V	
	EXTAL, EXTAL48, Ports 1, 3, 7, and A to G		$V_{CC} \times 0.8$	—	V_{CC} + 0.3	V	
	Ports 4*6 and 9		$V_{CC} \times 0.8$	—	AV_{CC} + 0.3*6	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD2 to MD0, $\overline{TRST}$, TCK, TMS, TDI, VBUS, $\overline{UBPM}$, FWE*5	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	EXTAL, EXTAL48, NMI, Ports 1, 3, 4, 7, 9, and A to G		–0.3	—	$V_{CC} \times 0.2$	V	
Output high voltage	All output pins	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} = –200 µA
			V_{CC} – 1.0	—	—	V	I_{OH} = –1 mA

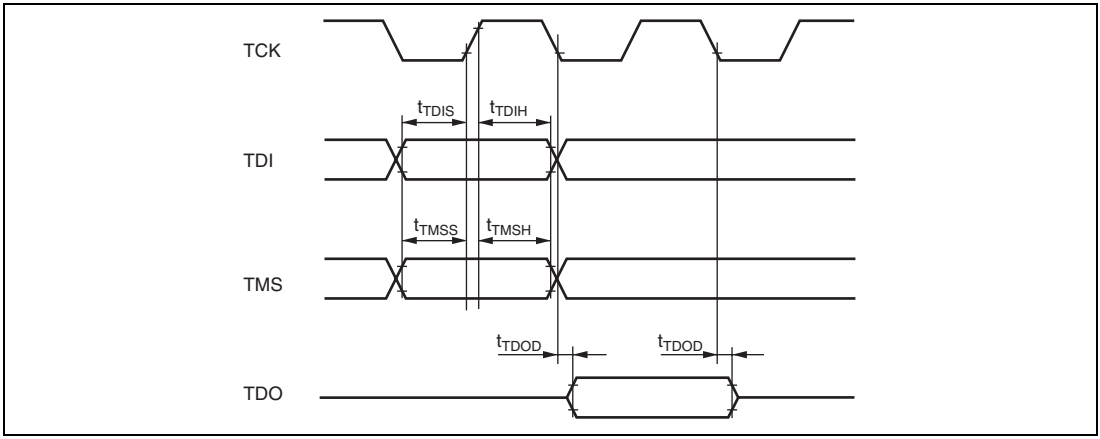

Figure 24.23 Boundary Scan Data Transmission Timing

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

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OH} = 0.4 mA
			—	—	0.4	V	I_{OL} = 0.8 mA
Input leakage current	$\overline{RES}$, $\overline{STBY}$, NMI, MD2 to MD0, FWE*5, VBUS,$\overline{UBPM}$	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 V to V_{CC} – 0.5 V
	Ports 4, 9	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 V to AV_{CC} – 0.5 V
3-state leak current (off status)	Ports 1, 3, 7, A to G	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 V to V_{CC} – 0.5 V
Input pull-up MOS current	Ports A to E	$-I_P$	10	—	300	µA	V_{in} = 0 V
Input capacity	$\overline{RES}$, NMI	C_{in}	—	—	30	pF	V_{in} = 0 V
	All input pins other than $\overline{RES}$ and NMI		—	—	15	pF	f = 1 MHz T_a = 25°C
Current dissipation*2	Normal operation (USB halts)	I_{CC}*3	—	23 (V_{CC} = 3.3 V)	40 (V_{CC} = 3.6 V)	mA	f = 16 MHz
			—	34 (V_{CC} = 3.3 V)	55 (V_{CC} = 3.6 V)	mA	f = 24 MHz
	Normal operation (USB operates)		—	28 (V_{CC} = 3.3 V)	50 (V_{CC} = 3.6 V)	mA	f = 16 MHz (PLL 3 multiplication)
			—	40 (V_{CC} = 3.3 V)	60 (V_{CC} = 3.6 V)	mA	f = 24 MHz (PLL 2 multiplication)
	Sleep mode		—	18 (V_{CC} = 3.3 V)	35 (V_{CC} = 3.6 V)	mA	f = 16 MHz (when USB and PLL are halted)
			—	26 (V_{CC} = 3.3 V)	45 (V_{CC} = 3.6 V)	mA	f = 24 MHz (when USB and PLL are halted)