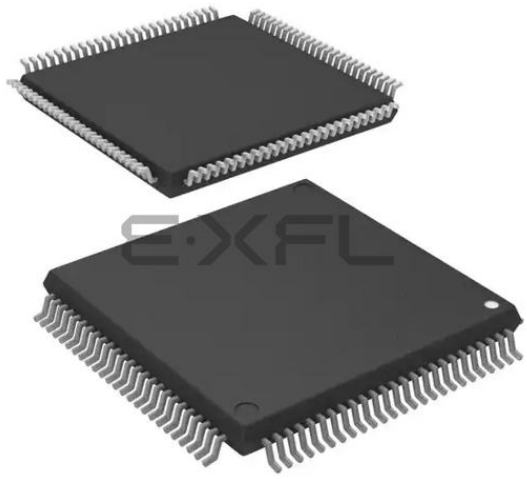


Welcome to E-XFL.COM

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	V850ES
Core Size	32-Bit Single-Core
Speed	48MHz
Connectivity	CSI, EBI/EMI, I²C, UART/USART, USB
Peripherals	DMA, LVD, PWM, WDT
Number of I/O	77
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	56K x 8
Voltage - Supply (Vcc/Vdd)	2.85V ~ 3.6V
Data Converters	A/D 12x10b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd70f3762gc-ueu-ax

2.3 Pin I/O Circuit Types, I/O Buffer Power Supplies and Connection of Unused Pins

Table 2-3. Pin I/O Circuit Types, I/O Buffer Power Supplies and Connection of Unused Pins (1/4)

Pin Name	Alternate Function	I/O Circuit Type	Recommended Connection	JG3-H	JH3-H
P00	INTP00	10-D	Input: Independently connect to EV_{DD} or V_{SS} via a resistor. Output: Leave open.	–	√
P01	INTP01			–	√
P02	NMI			√	√
P03	INTP02/ADTRG/UCLK			√	√
P04	INTP03			√	√
P05	INTP04			√	√
P10	ANO0	12-D	Input: Independently connect to AV_{REF1} or AV_{SS} via a resistor. Output: Leave open.	√	√
P11	ANO1			√	√
P20	TIAB03/KR2/TOAB03/RTP02	10-D	Input: Independently connect to EV_{DD} or V_{SS} via a resistor. Output: Leave open.	–	√
P21	SIF2/TIAB00/KR3/TOAB00/RTP03			–	√
P22	SOF2/KR4/RTP04			–	√
P23	$\overline{\text{SCKF2}}$/KR5/RTP05			–	√
P24	INTP05			–	√
P25	INTP06			–	√
P30	TXDC0/SOF4/INTP07	10-D	Input: Independently connect to EV_{DD} or V_{SS} via a resistor. Output: Leave open.	√	√
P31	RXDC0/SIF4/INTP08			√	√
P32	ASCKC0/$\overline{\text{SCKF4}}$/TIAA00/TOAA00			√	√
P33	TIAA01/TOAA01/RTCDIV/RTCCL			√	√
P34	TIAA10/TOAA10/TOAA1OFF/INTP09			√	√
P35	TIAA11/TOAA11/RTC1HZ			√	√
P36	TXDC3/SCL00/CTXD0[Note]/$\overline{\text{UDMARQ0}}$			√	√
P37	RXDC3/SDA00/CRXD0[Note]/$\overline{\text{UDMAAK0}}$			√	√
P40	SIF0/TXDC4/SDA01	10-D	Input: Independently connect to EV_{DD} or V_{SS} via a resistor. Output: Leave open.	√	√
P41	SOF0/RXDC4/SCL01			√	√
P42	$\overline{\text{SCKF0}}$/INTP10			√	√
P50	TIAB01/KR0/TOAB01/RTP00/$\overline{\text{UDMARQ1}}$	10-D	Input: Independently connect to EV_{DD} or V_{SS} via a resistor. Output: Leave open.	√	√
P51	TIAB02/KR1/TOAB02/RTP01/$\overline{\text{UDMAAK1}}$			√	√
P52	TIAB03/KR2/TOAB03/RTP02/DDI			√	–
P53	SIF2/TIAB00/KR3/TOAB00/RTP03/DDO			√	–
P54	SOF2/KR4/RTP04/DCK			√	–
P55	$\overline{\text{SCKF2}}$/KR5/RTP05/DMS			√	–
P56	INTP05/$\overline{\text{DRST}}$	10-N	Input: Independently connect to V_{SS} via a resistor. Fixing to V_{DD} level is prohibited. Output: Leave open. Internally pull-down after reset by $\overline{\text{RESET}}$ pin.	√	–

Note μPD70F3770, 70F3771 only

Remark JG3-H: V850ES/JG3-H, JH3-H: V850ES/JH3-H

(6) Port n function register (PFn)

The PFn register specifies normal output or N-ch open-drain output.

Each bit of this register corresponds to one pin of port n, and the output mode of the port pin can be specified in 1-bit units.

After reset: 00H R/W

	7	6	5	4	3	2	1	0
PFn	PFn7	PFn6	PFn5	PFn4	PFn3	PFn2	PFn1	PFn0

PFnm[Note]	Control of normal output/N-ch open-drain output
0	Normal output (CMOS output)
1	N-ch open-drain output

Note The PFnm bit of the PFn register is valid only when the PMnm bit of the PMn register is 0 (when the output mode is specified) in port mode (PMCnm bit = 0). When the PMnm bit is 1 (when the input mode is specified), the set value of the PFn register is invalid.

(2/2)

(b) V850ES/JH3-H

After reset: 00H R/W Address: FFFFF440H

	7	6	5	4	3	2	1	0
PMC0	0	0	PMC05	PMC04	PMC03	PMC02	PMC01	PMC00

PMC05	Specification of P05 pin operation mode
0	I/O port
1	INTP04 input

PMC04	Specification of P04 pin operation mode
0	I/O port
1	INTP03 input

PMC03	Specification of P03 pin operation mode
0	I/O port
1	INTP02 input/ADTRG input/UCLK input

PMC02	Specification of P02 pin operation mode
0	I/O port
1	NMI input

PMC01	Specification of P01 pin operation mode
0	I/O port
1	INTP01 input

PMC00	Specification of P00 pin operation mode
0	I/O port
1	INTP00 input

(4) Port 0 function control register (PFC0)

After reset: 00H R/W Address: FFFFF460H

	7	6	5	4	3	2	1	0
PFC0	0	0	0	0	PFC03	0	0	0

Remark For details of alternate function specification, see **4.3.1 (6) Port 0 alternate function specifications**.

(3) Port DL mode control register (PMCDL)

After reset: 0000H R/W Address: PMCDL FFFFF044H,
PMCDLL FFFFF044H, PMCDLH FFFFF045H

	15	14	13	12	11	10	9	8
PMCDL (PMCDLH)	PMCDL15	PMCDL14	PMCDL13	PMCDL12	PMCDL11	PMCDL10	PMCDL9	PMCDL8

	7	6	5	4	3	2	1	0
(PMCDLL)	PMCDL7	PMCDL6	PMCDL5	PMCDL4	PMCDL3	PMCDL2	PMCDL1	PMCDL0

PMCDLn	Specification of PDLn pin operation mode (n = 0 to 15)
0	I/O port
1	ADn I/O (address/data bus I/O)

Remarks 1. The PMCDL register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PMCDL register as the PMCDLH register and the lower 8 bits as the PMCDLL register, they can be read or written in 8-bit or 1-bit units.

2. To read/write bits 8 to 15 of the PMCDL register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PMCDLH register.

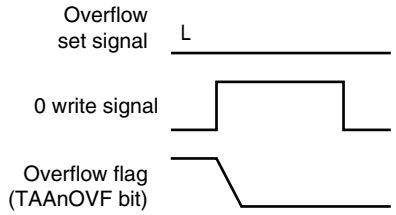
4.4 Port Register Settings When Alternate Function Is Used

Table 4-20 shows the port register settings when each port is used for an alternate function. When using a port pin as an alternate-function pin, refer to the description of each pin.

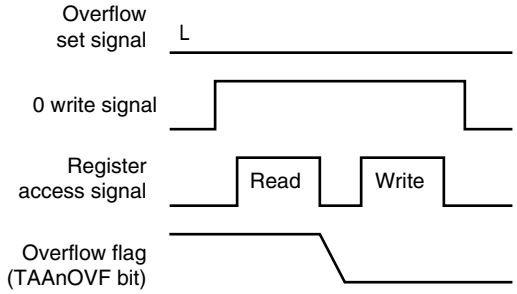
(e) Clearing overflow flag

The overflow flag can be cleared to 0 by clearing the TAAnOVF bit to 0 with the CLR instruction and by writing 8-bit data (bit 0 is 0) to the TAAnOPT0 register. To accurately detect an overflow, read the TAAnOVF bit when it is 1, and then clear the overflow flag by using a bit manipulation instruction.

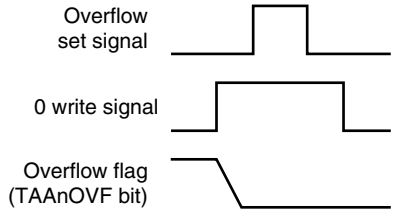
(i) Operation to write 0 (without conflict with setting)



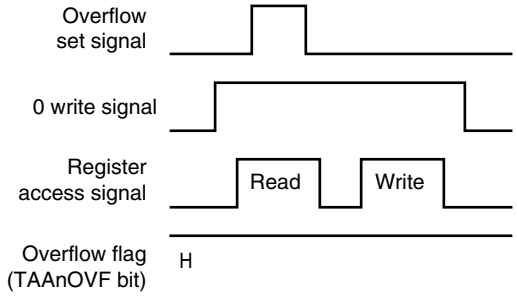
(iii) Operation to clear to 0 (without conflict with setting)



(ii) Operation to write 0 (conflict with setting)



(iv) Operation to clear to 0 (conflict with setting)



Remark n = 0 to 3, 5

To clear the overflow flag to 0, read the overflow flag to check if it is set to 1, and clear it with the CLR instruction. If 0 is written to the overflow flag without checking if the flag is 1, the set information of the overflow may be erased by writing 0 ((ii) in the above chart). Therefore, software may judge that no overflow has occurred even when an overflow has actually occurred.

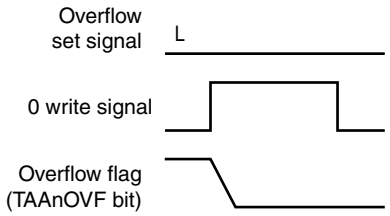
If execution of the CLR instruction conflicts with occurrence of an overflow when the overflow flag is cleared to 0 with the CLR instruction, the overflow flag remains set (1) even after execution of the clear instruction.

(2) Operation timing in pulse width measurement mode

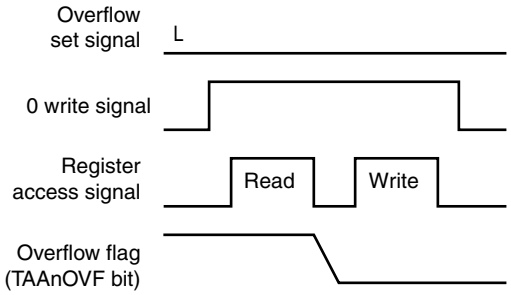
(a) Clearing overflow flag

The overflow flag can be cleared to 0 by clearing the TAAnOVF bit to 0 with the CLR instruction and by writing 8-bit data (bit 0 is 0) to the TAAnOPT0 register. To accurately detect an overflow, read the TAAnOVF bit when it is 1, and then clear the overflow flag by using a bit manipulation instruction.

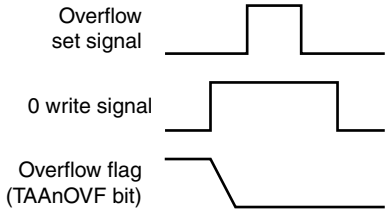
(i) Operation to write 0 (without conflict with setting)



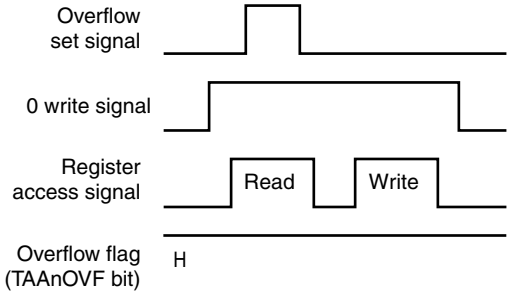
(iii) Operation to clear to 0 (without conflict with setting)



(ii) Operation to write 0 (conflict with setting)



(iv) Operation to clear to 0 (conflict with setting)



Remark n = 0 to 3, 5

To clear the overflow flag to 0, read the overflow flag to check if it is set to 1, and clear it with the CLR instruction. If 0 is written to the overflow flag without checking if the flag is 1, the set information of the overflow may be erased by writing 0 ((ii) in the above chart). Therefore, software may judge that no overflow has occurred even when an overflow has actually occurred.

If execution of the CLR instruction conflicts with occurrence of an overflow when the overflow flag is cleared to 0 with the CLR instruction, the overflow flag remains set (1) even after execution of the clear instruction.

8.5 Operation

TABn can perform the following operations.

Operation	TABnCTL1.TABnEST Bit (Software Trigger Bit)	TIABn0 Pin (External Trigger Input)	Capture/Compare Register Setting	Compare Register Write
Interval timer mode	Invalid	Invalid	Compare only	Anytime write
External event count mode[Note 1]	Invalid	Invalid	Compare only	Anytime write
External trigger pulse output mode[Note 2]	Valid	Valid	Compare only	Batch write
One-shot pulse output mode[Note 2]	Valid	Valid	Compare only	Anytime write
PWM output mode	Invalid	Invalid	Compare only	Batch write
Free-running timer mode	Invalid	Invalid	Switching enabled	Anytime write
Pulse width measurement mode[Note 2]	Invalid	Invalid	Capture only	Not applicable
Triangular wave PWM mode	Invalid	Invalid	Compare only	Batch write

Notes 1. To use the external event count mode, specify that the valid edge of the TIABn0 pin capture trigger input is not detected (by clearing the TABnIOC1.TABnIS1 and TABnIOC1.TABnIS0 bits to “00”).

2. When using the external trigger pulse output mode, one-shot pulse output mode, and pulse width measurement mode, select the internal clock as the count clock (by clearing the TABnCTL1.TABnEEE bit to 0).

Remark n = 0, 1

(2) TAB1 option register 2 (TAB1OPT2)

The TAB1OPT2 register is an 8-bit register that controls the timer Q option function.

This register can be rewritten when the TAB1CTL0.TAB1CE bit is 1. However, rewriting the TAB1DTM bit is prohibited when the TAB1CE bit is 1. The same value can be rewritten.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: FFFFF581H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
TAB1OPT2	TAB1RDE	TAB1DTM	TAB1ATM3	TAB1ATM2	TAB1AT3	TAB1AT2	TAB1AT1	TAB1AT0

TAB1RDE	Transfer culling enable
0	Do not cull transfer (transfer timing is generated every time at crest and valley).
1	Cull transfer at the same interval as interrupt culling set by the TAB1OPT1 register.

TAB1DTM	Dead-time counter operation mode selection (m = 1 to 3)
0	The dead-time counter counts up normally and, if TOAB1m output of TAB1 is at a narrow interval (TOAB1m output width < dead-time width), the dead-time counter is cleared and counts up again.
1	The dead-time counter counts up normally and, if TOAB1m output of TAB1 is at a narrow interval (TOAB1m output width < dead-time width), the dead-time counter counts down and the dead-time control width is automatically narrowed.
Rewriting the TAB1DTM bit is disabled during timer operation. If it is rewritten by mistake, stop the timer operation by clearing the TAB1CE bit to 0, and re-set the TAB1DTM bit.	

Cautions 1. When using interrupt culling (the TAB1OPT1.TAB1ID4 to TAB1OPT1.TAB1ID0 bits are set to other than 00000), be sure to set the TAB1RDE bit to 1.

This means that interrupts and transfers are generated at the same timing. Interrupts and transfers, cannot be set separately. If interrupts and transfers are set separately (TAB1RDE bit = 0), transfers are not performed normally.

2. When generating a dead-time period, set the TAB1DTC register to 1 or higher.

Note, when the operation is stopped (TAB1CTL0.TAB1CE bit = 0), a dead-time period is not generated, so the output levels of the TOAB1T1 to TOAB1T3 and TOAB1B1 to TOAB1B3 pins are in their default states. Therefore, for the protection of the system, take measures such as making the TOAB1T1 to TOAB1T3 and TOAB1B1 to TOAB1B3 pins go into a high-impedance state before stopping operation, or setting the output levels of the pins before switching port modes.

When a dead-time period is not needed, set the TAB1DTC register to 0.

(3) Intermittent batch rewrite mode (transfer culling mode)

This mode is set by clearing the TAB1OPT0.TAB1CMS bit to 0 and setting the TAB1OPT2.TAB1RDE bit to 1.

In this mode, the values written to each compare register are transferred to the internal buffer register all at once after the culled transfer timing and compared with the counter value. The transfer timing is the timing at which an interrupt is generated (INTTAB1CC0, INTTAB1OV) by interrupt culling.

For details of the interrupt culling function, see **11.4.3 Interrupt culling function**.

(a) Rewriting procedure

If data is written to the TAB1CCR1 register, the data of the TAB1CCR0 to TAB1CCR3, TAB1OPT1, TAA4CCR0, and TAA4CCR1 registers are transferred all at once to the internal buffer register at the next transfer timing. Therefore, write to the TAB1CCR1 register last. Writing to the register is prohibited after the TAB1CCR1 register has been written until the transfer timing is generated (until the INTTAB1OV or INTTAB1CC0 interrupt occurs). The operation procedure is as follows.

<1> Rewrite the TAB1CCR0, TAB1CCR2, TAB1CCR3, TAB1OPT1, TAA4CCR0, and TAA4CCR1 registers.
Do not rewrite registers that do not have to be rewritten.

<2> Rewrite the TAB1CCR1 register.
Rewrite the same value to the register even when it is not necessary to rewrite the TAB1CCR1 register.

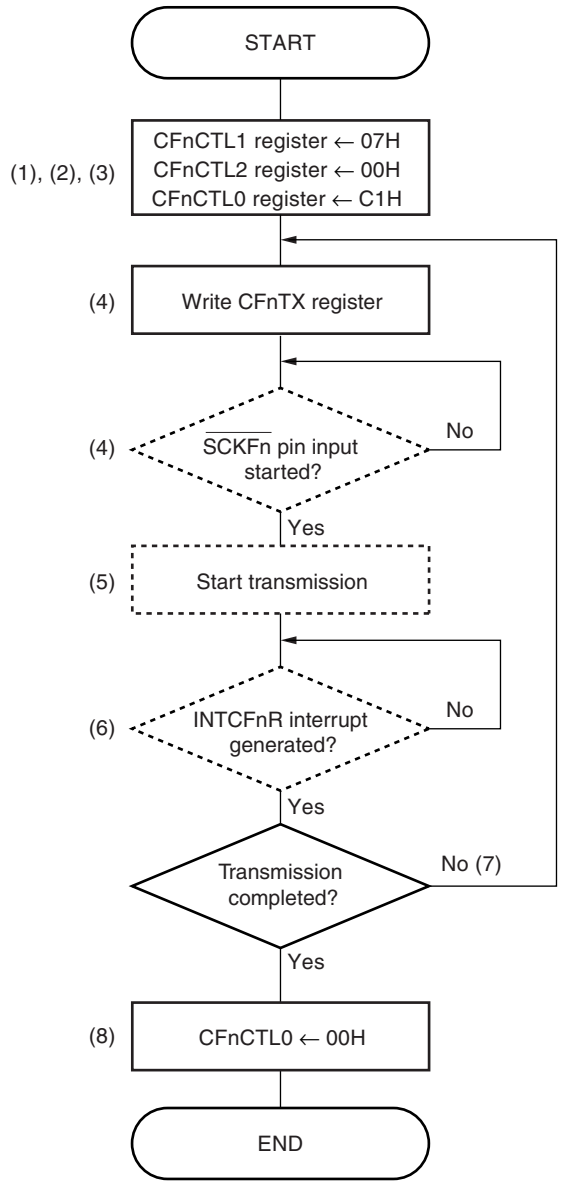
<3> Hold the next rewriting pending until the transfer timing is generated.
Perform the next rewrite after the INTTAB1OV or INTTAB1CC0 interrupt has occurred.

<4> Return to <1>.

18.6.4 Single transfer mode (slave mode, transmission mode)

MSB first (CFnCTL0.CFnDIR bit = 0), communication type 1 (CFnCTL1.CFnCKP and CFnCTL1.CFnDAP bits = 00), communication clock (f_{CCLK}) = external clock ($\overline{SCKFn}$) (CFnCTL1.CFnCKS2 to CFnCTL1.CFnCKS0 bits = 111), transfer data length = 8 bits (CFnCTL2.CFnCL3 to CFnCTL2.CFnCL0 bits = 0000)

(1) Operation flow



Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0 to 4

19.1.2 UARTC4, CSIF0, and I²C01 mode switching

In the V850ES/JG3-H and V850ES/JH3-H, UARTC4, CSIF0, and I²C01 share the same pins and therefore cannot be used simultaneously. Switching among UARTC4, CSIF0, and I²C01 must be set in advance, using the PMC4, PFC4, and PFCE4 registers.

Caution The transmit/receive operation of UARTC4, CSIF0, and I²C01 is not guaranteed if these functions are switched during transmission or reception. Be sure to disable the one that is not used.

Figure 19-2. UARTC4, CSIF0, and I²C01 Mode Switch Settings

After reset: 00H R/W Address: FFFFF448H

	7	6	5	4	3	2	1	0
PMC4	0	0	0	0	0	PMC42	PMC41	PMC40

After reset: 00H R/W Address: FFFFF468H

	7	6	5	4	3	2	1	0
PFC4	0	0	0	0	0	PFC42	PFC41	PFC40

After reset: 00H R/W Address: FFFFF708H

	7	6	5	4	3	2	1	0
PFCE4	0	0	0	0	0	0	PFCE41	PFCE40

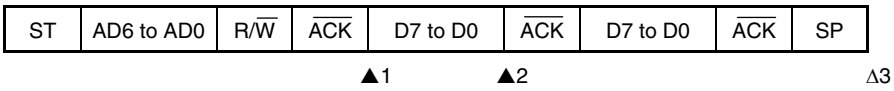
PMC4n	PFC4n	PFCE4n	Operation mode
0	×	×	Port I/O mode
1	0	0	CSIF0 mode
1	0	1	I²C01 mode
1	1	0	UARTC4 mode

Remarks 1. n = 0, 1

2. × = don't care

(3) When arbitration loss occurs during data transfer

<1> When IICCn.WTIMn bit = 0



▲1: IICSn register = 10001110B

▲2: IICSn register = 01000000B (Example: When ALDn bit is read during interrupt servicing)

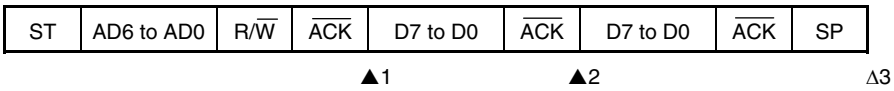
Δ 3: IICSn register = 00000001B

Remarks 1. ▲: Always generated

Δ: Generated only when SPIEn bit = 1

2. n = 0 to 2

<2> When WTIMn bit = 1



▲1: IICSn register = 10001110B

▲2: IICSn register = 01000100B (Example: When ALDn bit is read during interrupt servicing)

Δ 3: IICSn register = 00000001B

Remarks 1. ▲: Always generated

Δ: Generated only when SPIEn bit = 1

2. n = 0 to 2

(9) CAN0 module error counter register (C0ERC)

The C0ERC register indicates the count value of the transmission/reception error counter.

After reset: 0000H R Address: 03FEC054H

	15	14	13	12	11	10	9	8
C0ERC	REPS	REC6	REC5	REC4	REC3	REC2	REC1	REC0
	7	6	5	4	3	2	1	0
	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0

REPS	Reception error passive status bit
0	The value of the reception error counter is not error passive (< 128)
1	The value of the reception error counter is in the error passive range (≥ 128)

REC6 to REC0	Reception error counter bit
0 to 127	Number of reception errors. These bits reflect the status of the reception error counter. The number of errors is defined by the CAN protocol.

Remark The REC6 to REC0 bits of the reception error counter are invalid in the reception error passive status (C0INFO.RECS1, C0INFO.RECS0 bit = 11B).

TEC7 to TEC0	Transmission error counter bit
0 to 255	Number of transmission errors. These bits reflect the status of the transmission error counter. The number of errors is defined by the CAN protocol.

Remark The TEC7 to TEC0 bits of the transmission error counter are invalid in the bus-off status (C0INFO.BOFF bit = 1).

(2/2)

(a) Read

TSLOCK	Time stamp lock function enable bit
0	Time stamp lock function stopped. The TSOUT signal toggles each time the selected time stamp capture event occurs.
1	Time stamp lock function enabled. The TSOUT signal toggled each time the selected time stamp capture event occurred. However, the TSOUT output signal is locked when a data frame has been correctly received to message buffer 0[Note].

Note The TSEN bit is automatically cleared to 0.

TSSEL	Time stamp capture event selection bit
0	The time stamp capture event is SOF.
1	The time stamp capture event is the last bit of EOF.

TSEN	TSOUT operation setting bit
0	TSOUT toggle operation is disabled.
1	TSOUT toggle operation is enabled.

Remark The TSOUT signal is output from the CAN controller to a timer. For details, refer to **CHAPTER 7 16-BIT TIMER/EVENT COUNTER A (TAA)**.

(b) Write

Set TSLOCK	Clear TSLOCK	Setting of TSLOCK bit
0	1	TSLOCK bit is cleared to 0.
1	0	TSLOCK bit is set to 1.
Other than above		TSLOCK bit is not changed.

Set TSSEL	Clear TSSEL	Setting of TSSEL bit
0	1	TSSEL bit is cleared to 0.
1	0	TSSEL bit is set to 1.
Other than above		TSSEL bit is not changed.

Set TSEN	Clear TSEN	Setting of TSEN bit
0	1	TSEN bit is cleared to 0.
1	0	TSEN bit is set to 1.
Other than above		TSEN bit is not changed.

(29) UF0 FIFO clear 0 register (UF0FIC0)

This register clears each FIFO.

This register is write-only, in 8-bit units. If this register is read, 00H is read.

FW can clear the target FIFO by writing 1 to the corresponding bit of this register. The bit to which 1 has been written is automatically cleared to 0. Writing 0 to the bit is invalid.

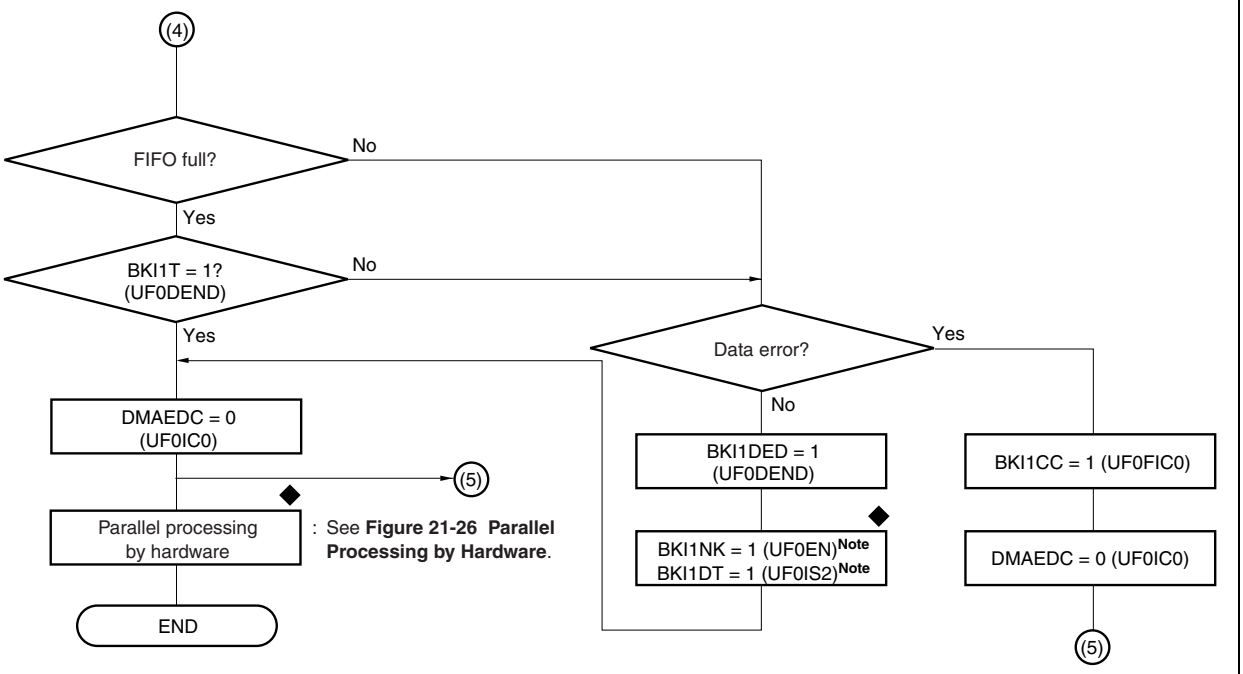
The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1, 3, 7) and the current setting of the interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0FIC0	BKI2SC	BKI2CC	BKI1SC	BKI1CC	ITR2C	ITR1C	EP0WC	EP0RC	00200060H	00H

Bit position	Bit name	Function
7, 5	BKInSC	These bits clear only the FIFO on the SIE side of the UF0BIn register (reset the counter). 1: Clear Writing these bits is invalid while an IN token for Endpoint m is being processed with the BKInNK bit set to 1. The BKInNK bit is automatically cleared to 0 by clearing the FIFO. Make sure that the FIFO on the CPU side is empty when these bits are used.
6, 4	BKInCC	These bits clear only the FIFO on the CPU side of the UF0BIn register (reset the counter). 1: Clear
2	ITR1C	These bits clear the UF0INT1 register (reset the counter). 1: Clear Writing these bits is invalid while an IN token for Endpoint 7 is being processed with the IT1NK bit set to 1. The IT1NK bit is automatically cleared to 0 by clearing the FIFO.
1	EP0WC	This bit clears the UF0E0W register (resets the counter). 1: Clear Writing this bit is invalid while an IN token for Endpoint0 is being processed with the EP0NKW bit set to 1. The EP0NKW bit is automatically cleared to 0 by clearing the FIFO.
0	EP0RC	This bit clears the UF0E0R register (resets the counter). 1: Clear When the EP0NKR bit is set to 1 (except when it has been set by FW), the EP0NKR bit is automatically cleared to 0 by clearing the FIFO.

Remark n = 1, 2
m = 1 where n = 1
m = 3 where n = 2

Figure 21-33 DMA Processing by Bulk Transfer (IN) (4/4)



Note The timing of the bit value changes depending on the status on the SIE side.

Remark ♦: Processing by hardware

(4) DMA addressing control registers 0 to 3 (DADC0 to DADC3)

The DADC0 to DADC3 registers are 16-bit registers that control the DMA transfer mode for DMA channel n (n = 0 to 3).

These registers can be read or written in 16-bit units.

Reset sets these registers to 0000H.

After reset: 0000H R/W Address: DADC0 FFFFF0D0H, DADC1 FFFFF0D2H, DADC2 FFFFF0D4H, DADC3 FFFFF0D6H

DADCn (n = 0 to 3)

15	14	13	12	11	10	9	8
0	DS0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
SAD1	SAD0	DAD1	DAD0	0	0	0	0

DS0	Setting of transfer data size
0	8 bits
1	16 bits

SAD1	SAD0	Setting of count direction of the transfer source address
0	0	Increment
0	1	Decrement
1	0	Fixed
1	1	Setting prohibited

DAD1	DAD0	Setting of count direction of the destination address
0	0	Increment
0	1	Decrement
1	0	Fixed
1	1	Setting prohibited

Cautions 1. Be sure to clear bits 15, 13 to 8, and 3 to 0 of the DADCn register to 0.

2. Set the DADCn register at the following timing when DMA transfer is disabled (DCHCn.Enn bit = 0).

- **Period from after reset to start of first DMA transfer**
- **Period from after channel initialization by DCHCn.INITn bit to start of DMA transfer**
- **Period from after completion of DMA transfer (DCHCn.TCn bit = 1) to start of the next DMA transfer**

3. The DS0 bit specifies the size of the transfer data, and does not control bus sizing. If 8-bit data (DS0 bit = 0) is set, therefore, the lower data bus is not always used.

4. If the transfer data size is set to 16 bits (DS0 bit = 1), transfer cannot be started from an odd address. Transfer is always started from an address with the first bit of the lower address aligned to 0.

5. If DMA transfer is executed on an on-chip peripheral I/O register (as the transfer source or destination), be sure to specify the same transfer size as the register size. For example, to execute DMA transfer on an 8-bit register, be sure to specify 8-bit transfer.

22.9 DMA Transfer Start Factors

There are two types of DMA transfer start factors, as shown below.

(1) Request by software

If the STGn bit is set to 1 while the DCHCn.TCn bit = 1 and Enn bit = 1 (DMA transfer enabled), DMA transfer is started.

To request the next DMA transfer cycle immediately after that, confirm, by using the DBCn register, that the preceding DMA transfer cycle has been completed, and set the STGn bit to 1 again (n = 0 to 3).

TCn bit = 0, Enn bit = 1
↓
STGn bit = 1 … Starts the first DMA transfer.
↓
Confirm that the contents of the DBCn register have been updated.
STGn bit = 1 … Starts the second DMA transfer.
↓
:
↓
Generation of terminal count … Enn bit = 0, TCn bit = 1, and INTDMAn signal is generated.

(2) Request by on-chip peripheral I/O

If an interrupt request is generated from the on-chip peripheral I/O set by the DTFRn register when the DCHCn.TCn bit = 0 and Enn bit = 1 (DMA transfer enabled), DMA transfer is started.

Cautions 1. Two start factors (software trigger and hardware trigger) cannot be used for one DMA channel. If two start factors are simultaneously generated for one DMA channel, only one of them is valid. The start factor that is valid cannot be identified.

2. A new transfer request that is generated after the preceding DMA transfer request was generated or in the preceding DMA transfer cycle is ignored (cleared).

3. The transfer request interval of the same DMA channel varies depending on the setting of bus wait in the DMA transfer cycle, the start status of the other channels, or the external bus hold request. In particular, as described in Caution 2, a new transfer request that is generated for the same channel before the DMA transfer cycle or during the DMA transfer cycle is ignored. Therefore, the transfer request intervals for the same DMA channel must be sufficiently separated by the system. When the software trigger is used, completion of the DMA transfer cycle that was generated before can be checked by updating the DBCn register.

<5> QB-V850ESJX3H[Note] In-circuit emulator	The in-circuit emulator serves to debug hardware and software when developing application systems using the V850ES/JG3-H or V850ES/JH3-H. It supports the integrated debugger ID850QB. This emulator should be used in combination with a power supply unit and emulation probe. Use the USB interface cable to connect this emulator to the host machine.
<3> USB interface cable	Cable to connect the host machine and the QB-V850ESJX3H.
<4> AC adapter	100 to 240 V can be supported by replacing the AC plug.
<9> QB-100GC-EA-04S QB-128GF-EA-01S QB-100GC-EA-05T QB-128GF-EA-02T Exchange adapter	Adapter to perform pin conversion. • QB-100GC-EA-04S: 100-pin plastic LQFP (GC-UEU type) • QB-128GF-EA-01S: 128-pin plastic LQFP (GF-GAT type) • QB-100GC-EA-05T: 100-pin plastic LQFP (GC-UEU type) • QB-128GF-EA-02T: 128-pin plastic LQFP (GC-GAT type)
<10> QB-100-CA-01S QB-128-CA-01S (S type only) Check pin adapter	Adapter used in waveform monitoring using the oscilloscope, etc. • QB-100-CA-01S: 100-pin plastic LQFP (GC-UEU type) • QB-128-CA-01S: 128-pin plastic LQFP (GF-GAT type)
<11> QB-100-SA-01S QB-144-SA-01S QB-100GC-YS-01T QB-128GF-YS-01T Space adapter	Adapter to adjust the height. • QB-100-SA-01S: 100-pin plastic LQFP (GC-UEU type) • QB-144-SA-01S: 128-pin plastic LQFP (GF-GAT type) • QB-100GC-YS-01T: 100-pin plastic LQFP (GC-UEU type) • QB-128GF-YS-01T: 128-pin plastic LQFP (GF-GAT type)
<12> QB-100GC-YQ-01T QB-128GF-YQ-01T (T type only) YQ connector	Conversion adapter to connect target connector and exchange adapter • QB-100GC-YQ-01T: 100-pin plastic LQFP (GC-UEU type) • QB-128GF-YQ-01T: 128-pin plastic LQFP (GF-GAT type)
<13> QB-100GC-MA-01S QB-128GF-MA-01S QB-100GC-HQ-01T QB-128GF-HQ-01T Mount adapter	Adapter to mount the V850ES/JG3-H or V850ES/JH3-H on a socket. • QB-100GC-MA-01S: 100-pin plastic LQFP (GC-UEU type) • QB-128GF-MA-01S: 128-pin plastic LQFP (GF-GAT type) • QB-100GC-HQ-01T: 100-pin plastic LQFP (GC-UEU type) • QB-128GF-HQ-01T: 128-pin plastic LQFP (GF-GAT type)
<14> QB-100GC-TC-01S QB-128GF-TC-01S QB-100GC-NQ-01T QB-128GF-NQ-01T Target connector	Connector to solder on the target system. • QB-100GC-TC-01S: 100-pin plastic LQFP (GC-UEU type) • QB-128GF-TC-01S: 128-pin plastic LQFP (GF-GAT type) • QB-100GC-NQ-01T: 100-pin plastic LQFP (GC-UEU type) • QB-128GF-NQ-01T: 128-pin plastic LQFP (GF-GAT type)

Note The QB-V850ESJX3H is supplied with a power supply unit, USB interface cable, and flash memory programmer (MINICUBE2). It is also supplied with integrated debugger ID850QB as control software.

Remark The numbers in the angle brackets correspond to the numbers in Figure A-2.

(6/37)

Symbol	Name	Unit	Page
C0MDATA110	CAN0 message data byte 1 register 10	CAN	969
C0MDATA111	CAN0 message data byte 1 register 11	CAN	969
C0MDATA112	CAN0 message data byte 1 register 12	CAN	969
C0MDATA113	CAN0 message data byte 1 register 13	CAN	969
C0MDATA114	CAN0 message data byte 1 register 14	CAN	969
C0MDATA115	CAN0 message data byte 1 register 15	CAN	969
C0MDATA116	CAN0 message data byte 1 register 16	CAN	969
C0MDATA117	CAN0 message data byte 1 register 17	CAN	969
C0MDATA118	CAN0 message data byte 1 register 18	CAN	969
C0MDATA119	CAN0 message data byte 1 register 19	CAN	969
C0MDATA120	CAN0 message data byte 1 register 20	CAN	969
C0MDATA121	CAN0 message data byte 1 register 21	CAN	969
C0MDATA122	CAN0 message data byte 1 register 22	CAN	969
C0MDATA123	CAN0 message data byte 1 register 23	CAN	969
C0MDATA124	CAN0 message data byte 1 register 24	CAN	969
C0MDATA125	CAN0 message data byte 1 register 25	CAN	969
C0MDATA126	CAN0 message data byte 1 register 26	CAN	969
C0MDATA127	CAN0 message data byte 1 register 27	CAN	969
C0MDATA128	CAN0 message data byte 1 register 28	CAN	969
C0MDATA129	CAN0 message data byte 1 register 29	CAN	969
C0MDATA130	CAN0 message data byte 1 register 30	CAN	969
C0MDATA131	CAN0 message data byte 1 register 31	CAN	969
C0MDATA200	CAN0 message data byte 2 register 00	CAN	969
C0MDATA201	CAN0 message data byte 2 register 01	CAN	969
C0MDATA202	CAN0 message data byte 2 register 02	CAN	969
C0MDATA203	CAN0 message data byte 2 register 03	CAN	969
C0MDATA204	CAN0 message data byte 2 register 04	CAN	969
C0MDATA205	CAN0 message data byte 2 register 05	CAN	969
C0MDATA206	CAN0 message data byte 2 register 06	CAN	969
C0MDATA207	CAN0 message data byte 2 register 07	CAN	969
C0MDATA208	CAN0 message data byte 2 register 08	CAN	969
C0MDATA209	CAN0 message data byte 2 register 09	CAN	969
C0MDATA210	CAN0 message data byte 2 register 10	CAN	969
C0MDATA211	CAN0 message data byte 2 register 11	CAN	969
C0MDATA212	CAN0 message data byte 2 register 12	CAN	969
C0MDATA213	CAN0 message data byte 2 register 13	CAN	969
C0MDATA214	CAN0 message data byte 2 register 14	CAN	969
C0MDATA215	CAN0 message data byte 2 register 15	CAN	969
C0MDATA216	CAN0 message data byte 2 register 16	CAN	969
C0MDATA217	CAN0 message data byte 2 register 17	CAN	969