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	SH2A-FPU
Core Size	32-Bit Single-Core
Speed	216MHz
Connectivity	CANbus, EBI/EMI, I²C, IEBus, SCI, SIO, SPI, USB
Peripherals	DMA, POR, PWM, WDT
Number of I/O	94
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	1.3M x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LFQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/r5s726b1p216fp-vz

Items	Specification
CD-ROM decoder	• Support of five formats: Mode 0, mode 1, mode 2, mode 2 form 1, and mode 2 form 2 • Sync codes detection and protection (Protection: When a sync code is not detected, it is automatically inserted.) • Descrambling • ECC correction — P, Q, PQ, and QP correction — PQ or QP correction can be repeated up to three times • EDC check Performed before and after ECC • Mode and form are automatically detected • Link sectors are automatically detected • Buffering data control Buffering CD-ROM data including Sync code is transferred in specified format, after the data is descrambled, corrected by ECC, and checked by EDC.
USB 2.0 host/function module	• Conforms to the Universal Serial Bus Specification Revision 2.0 • 12-Mbps transfer rates provided (host mode, function mode) • On-chip 2-Kbyte RAM as communication buffers
Sampling rate converter	• Data format: 32-bit stereo (16 bits each to L/R), 16-bit monaural • Input sampling rate: 8/11.025/12/16/22.05/24/32/44.1/48kHz • Output sampling rate: 32/44.1/48 kHz, 8/16 kHz (When input sampling rate is 44.1 KHz)
SD host interface	• SD memory I/O card interface (1-/4-bits SD bus) • Error check function: CRC7 (command), CRC16 (data) • Interrupt requests — Card access interrupt — SDIO access interrupt — Card detect interrupt • DMA transfer requests — SD_BUF write — SD_BUF read • Card detect function, write protect supported

2.2 Data Formats

2.2.1 Data Format in Registers

Register operands are always longwords (32 bits). If the size of memory operand is a byte (8 bits) or a word (16 bits), it is changed into a longword by expanding the sign-part when loaded into a register.

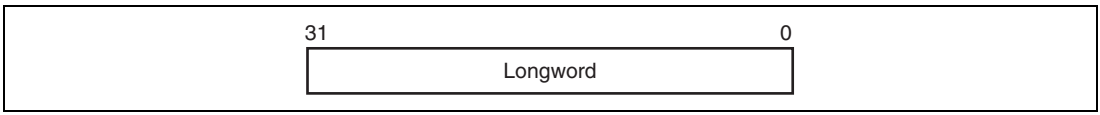


Figure 2.4 Data Format in Registers

2.2.2 Data Formats in Memory

Memory data formats are classified into bytes, words, and longwords. Memory can be accessed in 8-bit bytes, 16-bit words, or 32-bit longwords. A memory operand of fewer than 32 bits is stored in a register in sign-extended or zero-extended form.

A word operand should be accessed at a word boundary (an even address of multiple of two bytes: address 2n), and a longword operand at a longword boundary (an even address of multiple of four bytes: address 4n). Otherwise, an address error will occur. A byte operand can be accessed at any address.

Only big-endian byte order can be selected for the data format.

Data formats in memory are shown in figure 2.5.

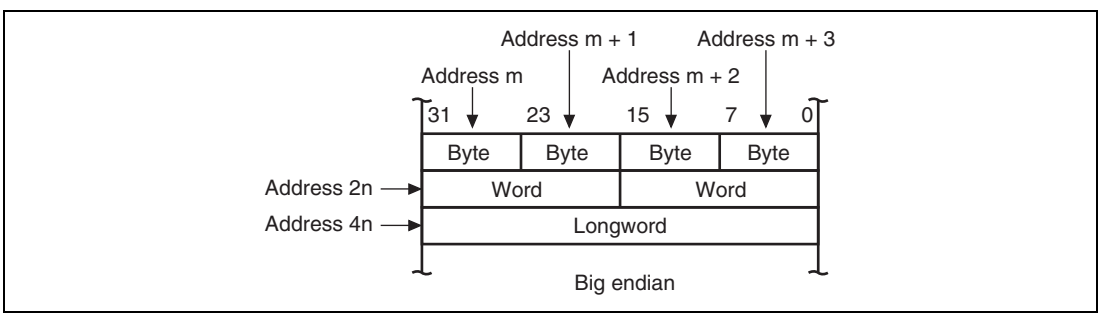


Figure 2.5 Data Formats in Memory

6.6.2 Trap Instructions

When a TRAPA instruction is executed, trap instruction exception handling starts. The CPU operates as follows:

1. The exception service routine start address which corresponds to the vector number specified in the TRAPA instruction is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the TRAPA instruction.
4. After jumping to the exception service routine start address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

6.6.3 Slot Illegal Instructions

An instruction placed immediately after a delayed branch instruction is called the “instruction placed in a delay slot”. When the instruction placed in the delay slot is undefined code (including FPU instructions and FPU-related CPU instructions in FPU module standby state), an instruction that rewrites the PC, a 32-bit instruction, an RESBANK instruction, a DIVS instruction, or a DIVU instruction, slot illegal exception handling starts when such kind of instruction is decoded. When the FPU has entered a module standby state, the floating point operation instruction and FPU-related CPU instructions are handled as undefined codes. If these instructions are placed in a delay slot and then decoded, a slot illegal instruction exception handling starts.

The CPU operates as follows:

1. The exception service routine start address is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the jump address of the delayed branch instruction immediately before the undefined code, the instruction that rewrites the PC, the 32-bit instruction, the RESBANK instruction, the DIVS instruction, or the DIVU instruction.
4. After jumping to the exception service routine start address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

12.3.13 Timer Start Register (TSTR)

TSTR is an 8-bit readable/writable register that selects operation/stoppage of TCNT for channels 0 to 4.

When setting the operating mode in TMDR or setting the count clock in TCR, first stop the TCNT counter.

Bit:	7	6	5	4	3	2	1	0
	CST4	CST3	-	-	-	CST2	CST1	CST0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	CST4	0	R/W	Counter Start 4 and 3
6	CST3	0	R/W	These bits select operation or stoppage for TCNT. If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the TIOC pin output compare output level is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value. 0: TCNT_4 and TCNT_3 count operation is stopped 1: TCNT_4 and TCNT_3 performs count operation
5 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	CST2	0	R/W	Counter Start 2 to 0
1	CST1	0	R/W	These bits select operation or stoppage for TCNT. If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the TIOC pin output compare output level is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value. 0: TCNT_2 to TCNT_0 count operation is stopped 1: TCNT_2 to TCNT_0 performs count operation
0	CST0	0	R/W	

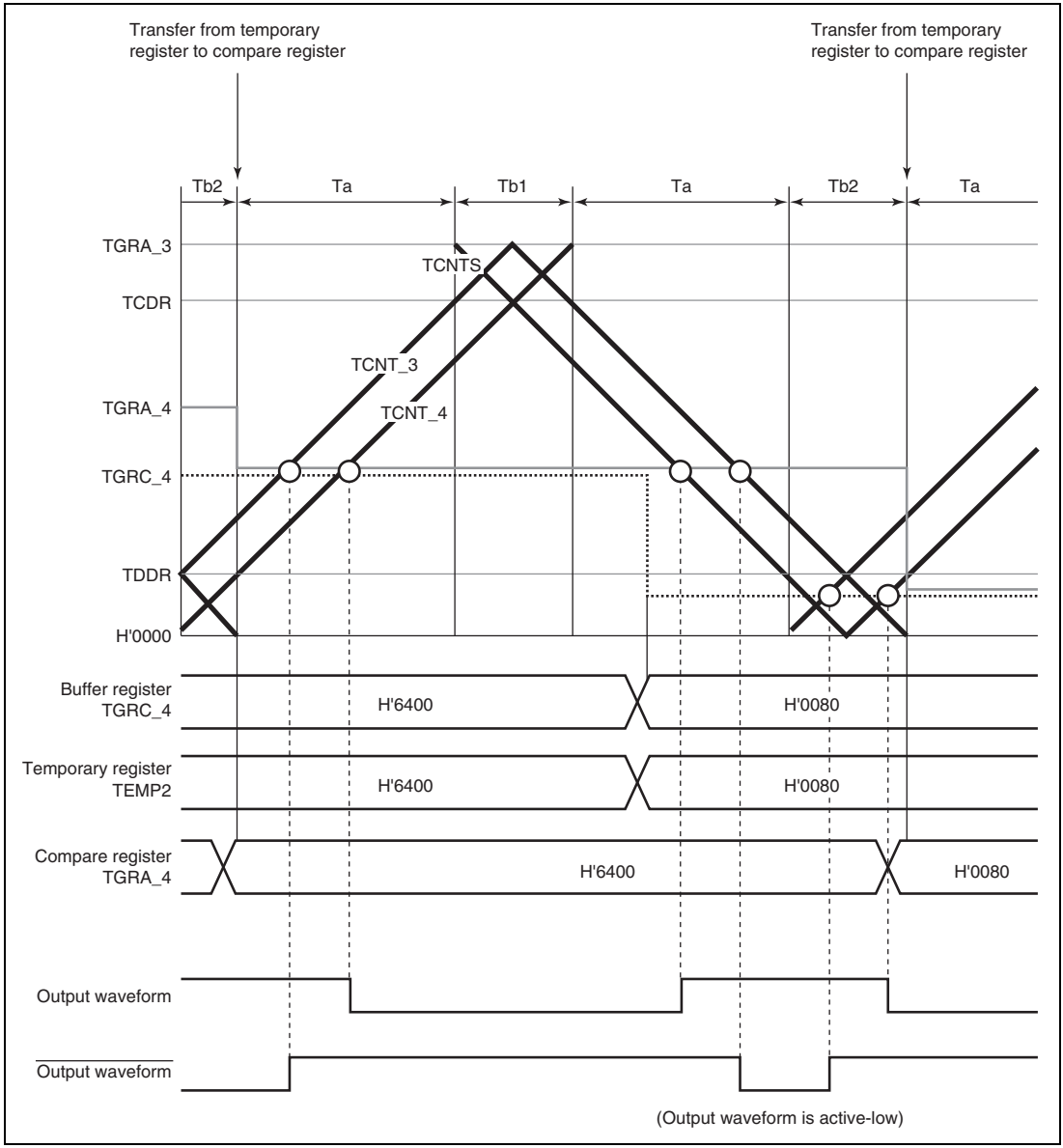


Figure 12.40 Example of Complementary PWM Mode Operation

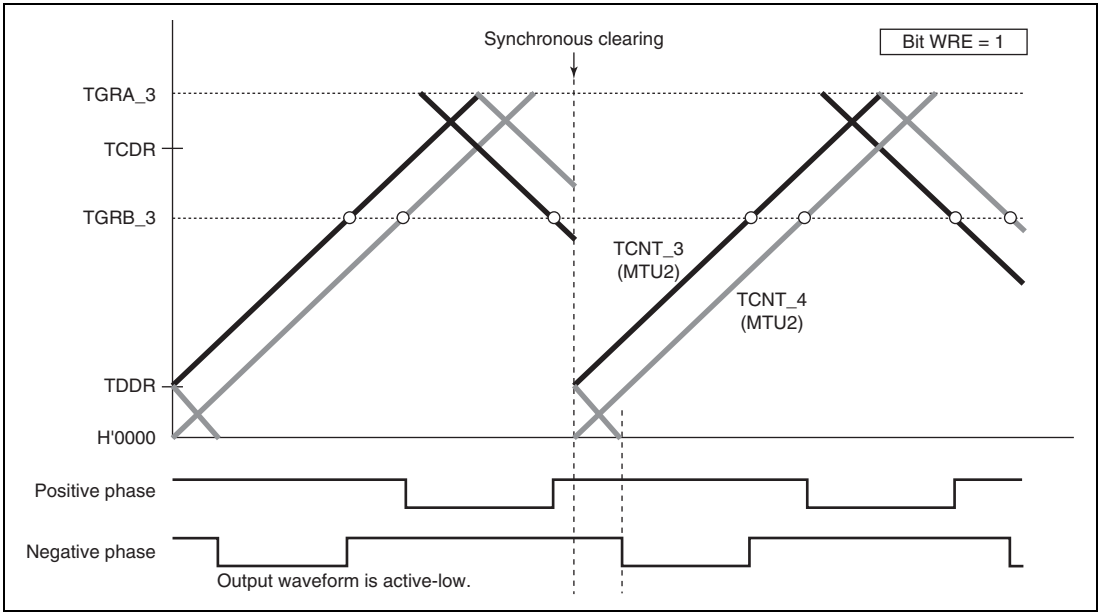


Figure 12.60 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 12.56; Bit WRE of TWCR is 1)

15.3.4 Hour Counter (RHRCNT)

RHRCNT is used for setting/counting in the BCD-coded hour section. The count operation is performed by a carry for each 1 hour of the minute counter.

The assignable range is from 00 through 23 (practically in BCD), otherwise operation errors occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.

Bit:	7	6	5	4	3	2	1	0
	-	-	10 hours		1 hour			
Initial value:	0	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	10 hours	Undefined	R/W	Counting Ten's Position of Hours Counts on 0 to 2 for ten's position of hours.
3 to 0	1 hour	Undefined	R/W	Counting One's Position of Hours Counts on 0 to 9 once per hour. When a carry is generated, 1 is added to the ten's position.

In serial transmission, this module operates as described below.

1. When data is written into the transmit FIFO data register (SCFTDR), the data is transferred from SCFTDR to the transmit shift register (SCTSR). Confirm that the TDFE flag in the serial status register (SCFSR) is set to 1 before writing transmit data to SCFTDR. The number of data bytes that can be written is (16 – transmit trigger setting).
2. When data is transferred from SCFTDR to SCTSR and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCFTDR. When the number of transmit data bytes in SCFTDR falls below the transmit trigger number set in the FIFO control register (SCFCR), the TDFE flag is set. If the TIE bit in the serial control register (SCSR) is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.

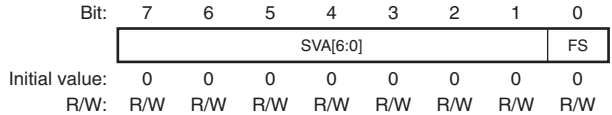
   The serial transmit data is sent from the TxD pin in the following order.

   A. Start bit: One-bit 0 is output.
   B. Transmit data: 8-bit or 7-bit data is output in LSB-first order.
   C. Parity bit: One parity bit (even or odd parity) is output. (A format in which a parity bit is not output can also be selected.)
   D. Stop bit(s): One or two 1 bits (stop bits) are output.
   E. Mark state: 1 is output continuously until the start bit that starts the next transmission is sent.
3. The SCFTDR transmit data is checked at the timing for sending the stop bit. If data is present, the data is transferred from SCFTDR to SCTSR, the stop bit is sent, and then serial transmission of the next frame is started.

Bit	Bit Name	Initial Value	R/W	Description
25, 24	ADB[1:0]	00	R/W	Address Bit Size Sets the address size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
23, 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21, 20	OPDB[1:0]	00	R/W	Option Data Bit Size Sets the option data size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	SPIDB[1:0]	00	R/W	Transfer Data Bit Size Sets the transfer data size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	CDE	1	R/W	Command Enable Sets the command to be output. 0: Command output disabled 1: Command output enabled

19.3.6 Slave Address Register (SAR)

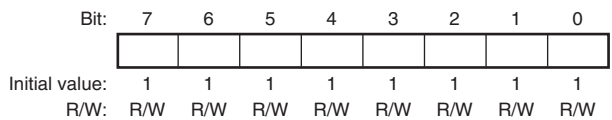
SAR is an 8-bit readable/writable register that selects the communications format and sets the slave address. In slave mode with the I²C bus format, if the upper seven bits of SAR match the upper seven bits of the first frame received after a start condition, this module operates as the slave device.



Bit	Bit Name	Initial Value	R/W	Description
7 to 1	SVA[6:0]	0000000	R/W	Slave Address These bits set a unique address in these bits, differing form the addresses of other slave devices connected to the I²C bus.
0	FS	0	R/W	Format Select 0: I²C bus format is selected 1: Clocked synchronous serial format is selected

19.3.7 I²C Bus Transmit Data Register (ICDRT)

ICDRT is an 8-bit readable/writable register that stores the transmit data. When ICDRT detects the space in the shift register (ICDRS), it transfers the transmit data which is written in ICDRT to ICDRS and starts transferring data. If the next transfer data is written to ICDRT while transferring data of ICDRS, continuous transfer is possible.



Bit	Bit Name	Initial Value	R/W	Description
9	RFFULE	0	R/W	Receive FIFO Full Enable 0: Disables interrupts due to receive FIFO full 1: Enables interrupts due to receive FIFO full
8	RDREQE	0	R/W	Receive FIFO Transfer Request Enable 0: Disables interrupts/DMA transfer requests due to receive FIFO transfer requests 1: Enables interrupts/DMA transfer requests due to receive FIFO transfer requests
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	FSERRE	0	R/W	Frame Synchronization Error Enable 0: Disables interrupts due to frame synchronization error 1: Enables interrupts due to frame synchronization error
3	TFOVFE	0	R/W	Transmit FIFO Overflow Enable 0: Disables interrupts due to transmit FIFO overflow 1: Enables interrupts due to transmit FIFO overflow
2	TFUDFE	0	R/W	Transmit FIFO Underflow Enable 0: Disables interrupts due to transmit FIFO underflow 1: Enables interrupts due to transmit FIFO underflow
1	RFUDFE	0	R/W	Receive FIFO Underflow Enable 0: Disables interrupts due to receive FIFO underflow 1: Enables interrupts due to receive FIFO underflow
0	RFOVFE	0	R/W	Receive FIFO Overflow Enable 0: Disables interrupts due to receive FIFO overflow 1: Enables interrupts due to receive FIFO overflow

Please note that for Mailbox 30 TTW is fixed to '01', Offset to '00' and rep_factor to ‘0’. The following tables report the combinations for the rep_factor and the offset.

Rep_factor	Description
3'b000	Every basic cycle (initial value)
3'b001	Every two basic cycle
3'b010	Every four basic cycle
3'b011	Every eight basic cycle
3'b100	Every sixteen basic cycle
3'b101	Every thirty two basic cycle
3'b110	Every sixty four basic cycle (once in system matrix)
3'b111	Reserved

The Offset Field determines the first cycle in which a Time Triggered Mailbox may start transmitting its Message.

Offset	Description
6'b000000	Initial Offset = 1st Basic Cycle (initial value)
6'b000001	Initial Offset = 2nd Basic Cycles
6'b000010	Initial Offset = 3rd Basic Cycles
6'b000011	Initial Offset = 4th Basic Cycles
6'b000100	Initial Offset = 5th Basic Cycles
…	
…	
6'b111110	Initial Offset = 63rd Basic Cycles
6'b111111	Initial Offset = 64th Basic Cycles

The following relation must be maintained:

$$Cycle\_Count\_Maximum + 1 >= Repeat\_Factor > Offset$$

$$Cycle\_Count\_Maximum = 2^{CMAX} - 1$$

$$Repeat\_Factor = 2^{rep\_factor}$$

Bit 5 — Error Passive Status Bit (GSR5): Indicates whether the CAN Interface is in Error Passive or not. This bit will be set high as soon as this module enters the Error Passive state and is cleared when the module enters again the Error Active state (this means the GSR5 will stay high during Error Passive and during Bus Off). Consequently to find out the correct state both GSR5 and GSR0 must be considered.

Bit 5: GSR5	Description
0	This module is not in Error Passive or in Bus Off status (Initial value) [Reset condition] This module is in Error Active state
1	This module is in Error Passive (if GSR0 = 0) or Bus Off (if GSR0 = 1) [Setting condition] When TEC ≥ 128 or REC ≥ 128 or if Error Passive Test Mode is selected

Bit 4 — Halt/Sleep Status Bit (GSR4): Indicates whether the CAN engine is in the halt/sleep state or not. Please note that the clearing time of this flag is not the same as the setting time of IRR12.

Please note that this flag reflects the status of the CAN engine and not of the full this module IP. This module exits sleep mode and can be accessed once MCR5 is cleared. The CAN engine exits sleep mode only after two additional transmission clocks on the CAN Bus.

Bit 4: GSR4	Description
0	This module is not in the Halt state or Sleep state (Initial value)
1	Halt mode (if MCR1 = 1) or Sleep mode (if MCR5 = 1) [Setting condition] If MCR1 is set and the CAN bus is either in intermission or idle or MCR5 is set and this module is in the halt mode or this module is moving to Bus Off when MCR14 and MCR6 are both set

Bit 3 — Reset Status Bit (GSR3): Indicates whether this module is in the reset state or not.

Bit 3: GSR3	Description
0	This module is not in the reset state
1	Reset state (Initial value) [Setting condition] After an internal reset of this module (due to SW or HW reset)

22.5 Interrupt Sources

Table 22.2 lists this module interrupt sources. These sources can be masked. Masking is implemented using the mailbox interrupt mask registers (MBIMR) and interrupt mask register (IMR). For details on the interrupt vector of each interrupt source, see section 7, Interrupt Controller.

Table 22.2 Interrupt Sources

Interrupt	Description	Interrupt Flag	DMAC Activation
ERSn*[1]	Error Passive Mode (TEC ≥ 128 or REC ≥ 128)	IRR5	Not possible
	Bus Off (TEC ≥ 256)/Bus Off recovery	IRR6	
	Error warning (TEC ≥ 96)	IRR3	
	Error warning (REC ≥ 96)	IRR4	
OVRn*[1]	Reset/halt/CAN sleep transition	IRR0	
	Overload frame transmission	IRR7	
	Unread message overwrite (overrun)	IRR9	
	Start of new system matrix	IRR10	
	TCMR2 compare match	IRR11	
	Bus activity while in sleep mode	IRR12	
	Timer overrun/Next_is_Gap reception/message error	IRR13	
	TCMR0 compare match	IRR14	
	TCMR1 compare match	IRR15	
RM0n*[1]*[2], RM1n*[1]*[2]	Data frame reception	IRR1*[3]	Possible*[4]
	Remote frame reception	IRR2*[3]	
SLEn*[1]	Message transmission/transmission disabled (slot empty)	IRR8	Not possible

Notes: 1. n = 0, 1

2. RM0 is an interrupt generated by the remote request pending flag for mailbox 0 (RFPR0[0]) or the data frame receive flag for mailbox 0 (RXPR0[0]). RM1 is an interrupt generated by the remote request pending flag for mailbox n (RFPR0[n]) or the data frame receive flag for mailbox n (RXPR0[n]) (n = 1 to 31).
3. IRR1 is a data frame received interrupt flag for mailboxes 0 to 31, and IRR2 is a remote frame request interrupt flag for mailboxes 0 to 31.
4. The direct memory access controller is activated only by an RM0n interrupt.

(7) Parity bit

The parity bit is used to confirm that transfer data occurs with no errors.

The parity bit is added to respective data of the master address, slave address, control, message length, and data bits.

Even parity is used. When the number of bits having the value 1 is odd, the parity bit is 1. When the number of bits having the value 1 is even, the parity bit is 0.

(8) Acknowledge bit

In normal communications (single unit to single unit communications), the acknowledge bit is added in the following positions to confirm that data is correctly accepted.

- At the end of the slave address field
- At the end of the control field
- At the end of the message length field
- At the end of the data field

The acknowledge bit is defined below.

- 0: indicates that the transfer data is acknowledged. (ACK)
- 1: indicates that the transfer data is not acknowledged. (NAK)

Note that the acknowledge bit is ignored in the case of broadcast communications.

(a) Acknowledge bit at the End of the Slave Address Field

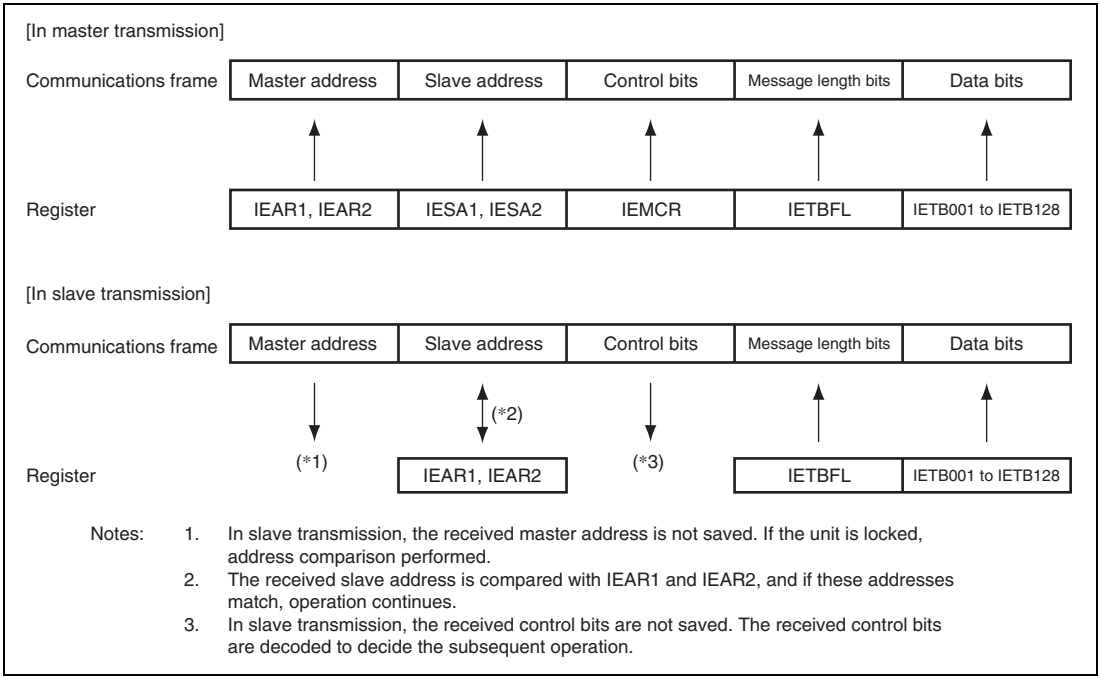
The acknowledge bit at the end of the slave address field becomes NAK in the following cases and transfer is stopped.

- When the parity of the master address or slave address bits is incorrect
- When a timing error (an error in bit format) occurs
- When there is no slave unit

23.4 Data Format

23.4.1 Transmission Format

Figure 23.6 shows the relationship between the transfer format and each register during the IEBus data transmission.



Notes:
1. In slave transmission, the received master address is not saved. If the unit is locked, address comparison performed.
2. The received slave address is compared with IEAR1 and IEAR2, and if these addresses match, operation continues.
3. In slave transmission, the received control bits are not saved. The received control bits are decoded to decide the subsequent operation.

Figure 23.6 Relationship between Transfer Format and Each Register during IEBus Data Transmission

25.3.15 Decoding Stoppage Source Status Register (CBUFST1)

The decoding stoppage source status register (CBUFST1) indicates that decoding/buffering has been stopped due to some errors.

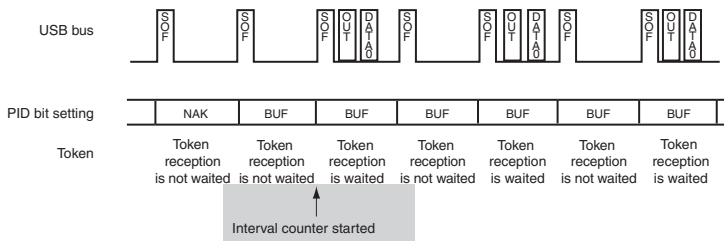
A bit in this register can only be set when the corresponding bit in the CROMCTL3 register is set to 1.

Bit:	7	6	5	4	3	2	1	0
	BUF_ECC	BUF_EDC	-	BUF_MD	BUF_MIN	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	BUF_ECC	0	R	Indicates that decoding and buffering have been stopped because of an error that is not correctable by using the ECC.
6	BUF_EDC	0	R	Indicates that decoding and buffering have been stopped because the post-correction EDC check indicated an error.
5	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
4	BUF_MD	0	R	Indicates that decoding and buffering have been stopped because the current sector is in a mode or form differing from that of the previous sectors.
3	BUF_MIN	0	R	Indicates that decoding and buffering have been stopped because a non-sequential minutes, seconds, or frames (1/75 second) value has been encountered.
2 to 0	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
13	ADST	0	R/W	A/D Start Starts or stops A/D conversion. This bit remains set to 1 during A/D conversion. 0: A/D conversion is stopped 1: Single mode: A/D conversion starts. This bit is automatically cleared to 0 when A/D conversion ends on the selected channel. Multi mode: A/D conversion starts. This bit is automatically cleared to 0 when A/D conversion is completed cycling through the selected channels. Scan mode: A/D conversion starts. A/D conversion is continuously performed until this bit is cleared to 0 by software, by a power-on reset as well as by a transition to deep standby mode, software standby mode or module standby mode.
12 to 9	TRGS[3:0]	0000	R/W	Timer Trigger Select These bits enable or disable starting of A/D conversion by a trigger signal. 0000: Start of A/D conversion by external trigger input is disabled 0001: A/D conversion is started by conversion trigger TRGAN from the multi-function timer pulse unit 2 0010: A/D conversion is started by conversion trigger TRG0N from the multi-function timer pulse unit 2 0011: A/D conversion is started by conversion trigger TRG4AN from the multi-function timer pulse unit 2 0100: A/D conversion is started by conversion trigger TRG4BN from the multi-function timer pulse unit 2 1001: A/D conversion is started by $\overline{\text{ADTRG}}$ Other than above: Setting prohibited

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Renesas serial peripheral interface	SPDR_1	SPD31	SPD30	SPD29	SPD28	SPD27	SPD26	SPD25	SPD24
		SPD23	SPD22	SPD21	SPD20	SPD19	SPD18	SPD17	SPD16
		SPD15	SPD14	SPD13	SPD12	SPD11	SPD10	SPD9	SPD8
		SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
	SPSCR_1	—	—	—	—	—	—	SPSLN1	SPSLN0
	SPSSR_1	—	—	—	—	—	—	SPCP1	SPCP0
	SPBR_1	SPR7	SPR6	SPR5	SPR4	SPR3	SPR2	SPR1	SPR0
	SPDCR_1	TXDMY	SPLW1	SPLW0	—	—	—	—	—
	SPCKD_1	—	—	—	—	—	SCKDL2	SCKDL1	SCKDL0
	SSLND_1	—	—	—	—	—	SLNDL2	SLNDL1	SLNDL0
	SPND_1	—	—	—	—	—	SPNDL2	SPNDL1	SPNDL0
	SPCMD_10	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	—	—	—	BRDV1	BRDV0	CPOL	CPHA
	SPCMD_11	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	—	—	—	BRDV1	BRDV0	CPOL	CPHA
	SPCMD_12	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	—	—	—	BRDV1	BRDV0	CPOL	CPHA
	SPCMD_13	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	—	—	—	BRDV1	BRDV0	CPOL	CPHA
	SPBFCR_1	TXRST	RXRST	TXTRG[1]	TXTRG[0]	—	RXTRG[2]	RXTRG[1]	RXTRG[0]
	SPBFDR_1	—	—	—	—	T[3]	T[2]	T[1]	T[0]
		—	—	R[5]	R[4]	R[3]	R[2]	R[1]	R[0]
	SPCR_2	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	—	—
	SSLP_2	—	—	—	—	—	—	—	SSL0P
	SPPCR_2	—	—	MOIFE	MOIFV	—	—	—	SPLP
	SPSR_2	SPRF	TEND	SPTEF	—	—	MODF	—	OVRF
	SPDR_2	SPD31	SPD30	SPD29	SPD28	SPD27	SPD26	SPD25	SPD24
		SPD23	SPD22	SPD21	SPD20	SPD19	SPD18	SPD17	SPD16
		SPD15	SPD14	SPD13	SPD12	SPD11	SPD10	SPD9	SPD8
		SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0

Item	Page	Revision (See Manual for Details)
27.4.4 FIFO Buffer Memory (1) FIFO Buffer Memory Allocation • Buffer memory specifications (single/double setting)	1507	Description amended Either a single or double buffer can be selected for PIPE1 to PIPE5, using the DBLB bit in PIPECFG.
(2) FIFO Port Functions (a) FIFO Port Selection	1508	Description amended Also, the bus width to be accessed should be selected using the MBW bit. The buffer memory access direction conforms to the DIR bit in PIPECFG. The ISEL bit determines this only for the DCP.
Figure 27.11 Relationship between Frames and Expected Token Reception when IITV = 1	1523	Figure amended 
34.1 Register Addresses (by functional module, in order of the corresponding section numbers)	1733	Table amended

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
USB 2.0 host/function module	Pipe 1 transaction counter enable register	PIPE1TRE	16	H'FFFFC090	16
	Pipe 1 transaction counter register	PIPE1TRN	16	H'FFFFC092	16
	Pipe 2 transaction counter enable register	PIPE2TRE	16	H'FFFFC094	16
	Pipe 2 transaction counter register	PIPE2TRN	16	H'FFFFC096	16
	Pipe 3 transaction counter enable register	PIPE3TRE	16	H'FFFFC098	16
	Pipe 3 transaction counter register	PIPE3TRN	16	H'FFFFC09A	16
	Pipe 4 transaction counter enable register	PIPE4TRE	16	H'FFFFC09C	16
	Pipe 4 transaction counter register	PIPE4TRN	16	H'FFFFC09E	16
	Pipe 5 transaction counter enable register	PIPE5TRE	16	H'FFFFC0A0	16
	Pipe 5 transaction counter register	PIPE5TRN	16	H'FFFFC0A2	16
	~~USB AC characteristics switching register 1~~	~~USBACSWR1~~	~~16~~	~~H'FFFFC0C2~~	~~16~~
	Device address 0 configuration register	DEVADD0	16	H'FFFFC0D0	16
	Device address 1 configuration register	DEVADD1	16	H'FFFFC0D2	16
	Device address 2 configuration register	DEVADD2	16	H'FFFFC0D4	16
	Device address 3 configuration register	DEVADD3	16	H'FFFFC0D6	16
	Device address 4 configuration register	DEVADD4	16	H'FFFFC0D8	16
	Device address 5 configuration register	DEVADD5	16	H'FFFFC0DA	16