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	H8SX
Core Size	32-Bit Single-Core
Speed	50MHz
Connectivity	EBI/EMI, I²C, IrDA, SCI, SmartCard, USB
Peripherals	DMA, LVD, POR, PWM, WDT
Number of I/O	92
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	40K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 8x10b; D/A 2x8b
Oscillator Type	External
Operating Temperature	-20°C ~ 75°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/r5f61664rn50fpv

- 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 structure of the exception vector table is shown in figure 2.2.

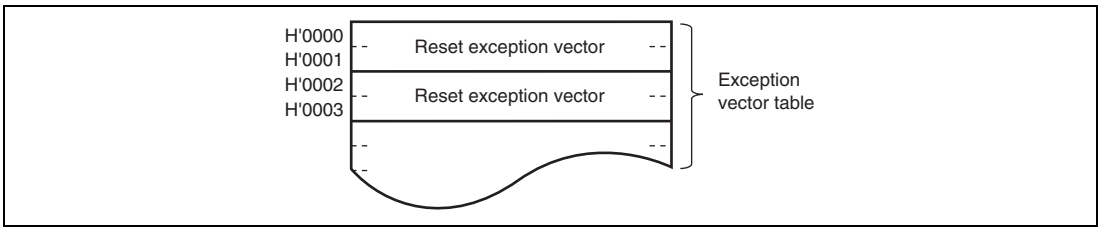

Figure 2.2 Exception Vector Table (Normal Mode)

The memory indirect (@@aa:8) and extended memory indirect (@@vec:7) addressing modes are used in the JMP and JSR instructions. An 8-bit absolute address included in the instruction code specifies a memory location. Execution branches to the contents of the memory location.

- Stack Structure

  The stack structure of PC at a subroutine branch and that of PC and CCR at an exception handling are shown in figure 2.3. The PC contents are saved or restored in 16-bit units.

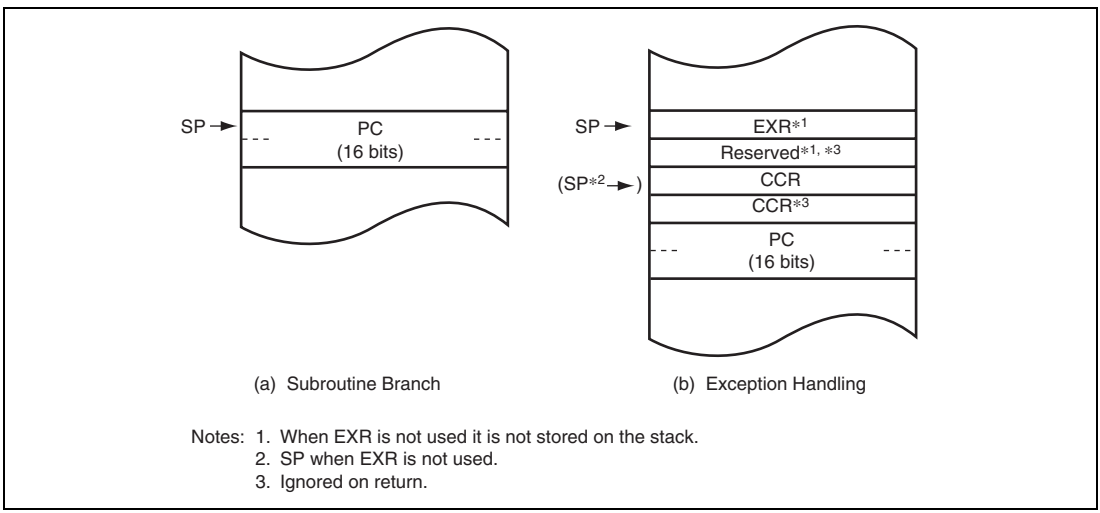

Figure 2.3 Stack Structure (Normal Mode)

9.5.6 Endian and Data Alignment

Data sizes for the CPU and other internal bus masters are byte, word, and longword. The bus controller has a data alignment function, and controls whether the upper byte data bus (D15 to D8) or lower data bus (D7 to D0) is used according to the bus specifications for the area being accessed (8-bit access space or 16-bit access space), the data size, and endian format when accessing external address space.

(1) 8-Bit Access Space

With the 8-bit access space, the lower byte data bus (D7 to D0) is always used for access. The amount of data that can be accessed at one time is one byte: a word access is performed as two byte accesses, and a longword access, as four byte accesses.

Figures 9.11 and 9.12 illustrate data alignment control for the 8-bit access space. Figure 9.11 shows the data alignment when the data endian format is specified as big endian. Figure 9.12 shows the data alignment when the data endian format is specified as little endian.

Data Size	Access Address	Access Count	Bus Cycle	Data Size	Strobe signal LHWR/LUB, LLWR/LLB, RD; Data bus D15 D8 D7 D0
Byte	n	1	1st	Byte	7 0
Word	n	2	1st	Byte	15 8
			2nd	Byte	7 0
Longword	n	4	1st	Byte	31 24
			2nd	Byte	23 16
			3rd	Byte	15 8
			4th	Byte	7 0

Figure 9.11 Access Sizes and Data Alignment Control for 8-Bit Access Space (Big Endian)

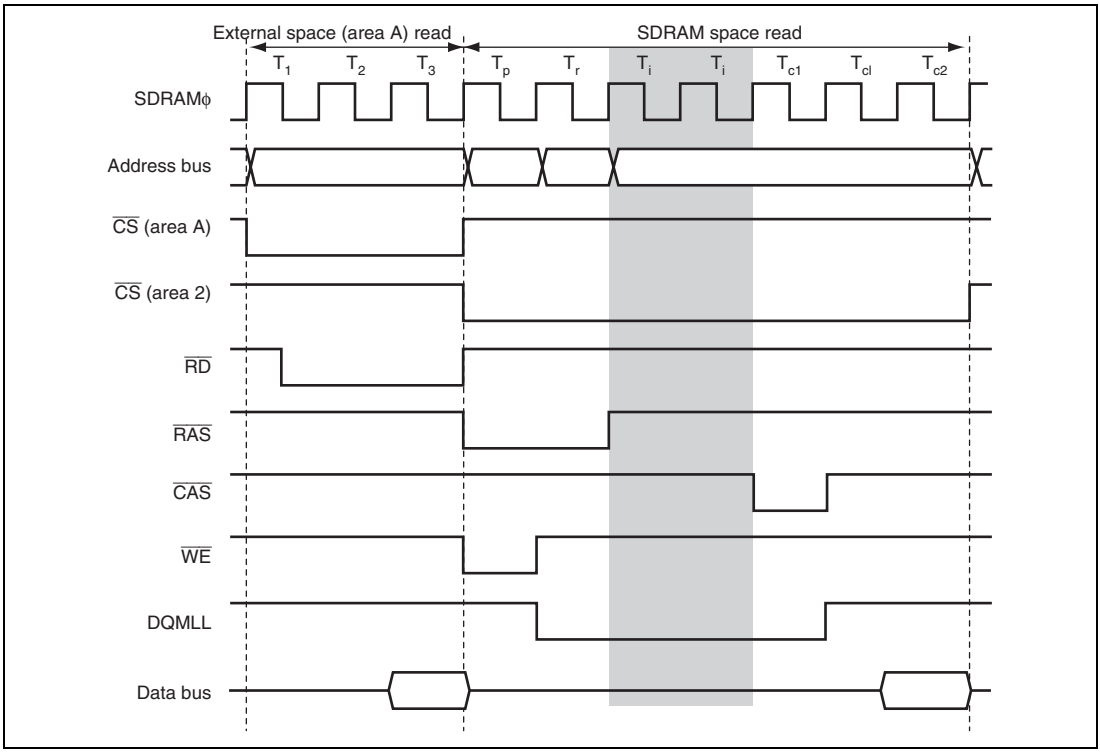

Figure 9.100 Example of SDRAM Full Access after External Read (CAS Latency = 2)

10.5.2 Transfer Modes

(1) Normal Transfer Mode

In normal transfer mode, one data access size of data is transferred at a single transfer request. Up to 4 Gbytes can be specified as a total transfer size by DTCR. DBSR is ignored in normal transfer mode.

The $\overline{\text{TEND}}$ signal is output only in the last DMA transfer.

Figure 10.7 shows an example of the signal timing in normal transfer mode and figure 10.8 shows the operation in normal transfer mode.

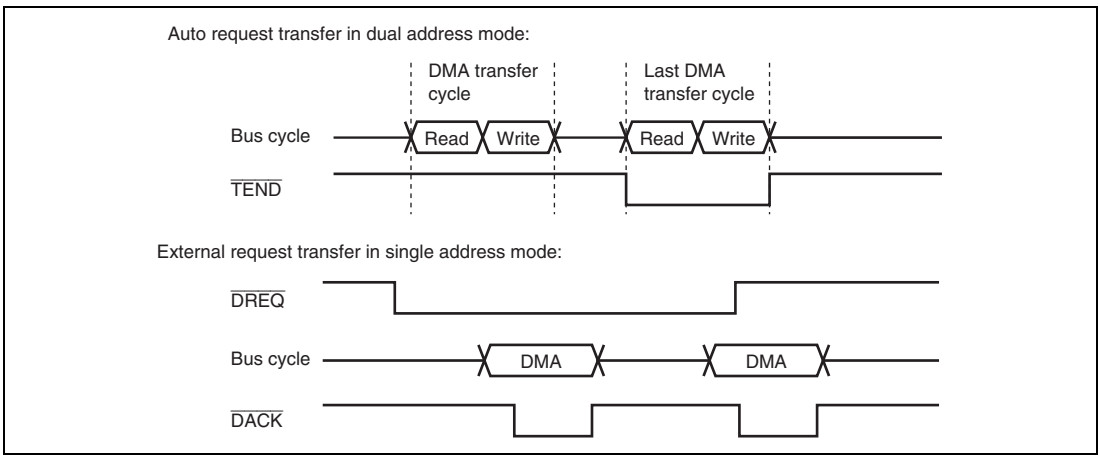

Figure 10.7 Example of Signal Timing in Normal Transfer Mode

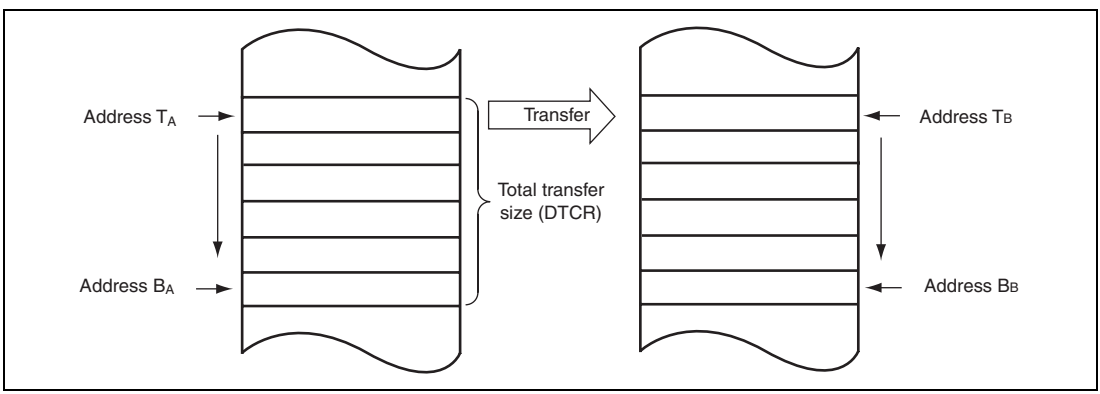

Figure 10.8 Operations in Normal Transfer Mode

10.5.5 Extended Repeat Area Function

The source and destination address sides can be specified as the extended repeat area. The contents of the address register repeat addresses within the area specified as the extended repeat area. For example, to use a ring buffer as the transfer target, the contents of the address register should return to the start address of the buffer every time the contents reach the end address of the buffer (overflow on the ring buffer address). This operation can automatically be performed using the extended repeat area function of the DMAC.

The extended repeat areas can be specified independently to the source address register (DSAR) and destination address register (DDAR).

The extended repeat area on the source address is specified by bits SARA4 to SARA0 in DACR. The extended repeat area on the destination address is specified by bits DARA4 to DARA0 in DACR. The extended repeat area sizes for each side can be specified independently.

A DMA transfer is stopped and an interrupt by an extended repeat area overflow can be requested to the CPU when the contents of the address register reach the end address of the extended repeat area. When an overflow on the extended repeat area set in DSAR occurs while the SARIE bit in DACR is set to 1, the ESIF bit in DMDR is set to 1 and the DTE bit in DMDR is cleared to 0 to stop the transfer. At this time, if the ESIE bit in DMDR is set to 1, an interrupt by an extended repeat area overflow is requested to the CPU. When the DARIE bit in DACR is set to 1, an overflow on the extended repeat area set in DDAR occurs, meaning that the destination side is a target. During the interrupt handling, setting the DTE bit in DMDR resumes the transfer.

Cluster transfer mode: One cluster data is transferred at a single transfer request
Up to 32-byte data can be set as cluster size

- Selection of extended repeat area function (to transfer data such as ring buffer data by fixing the upper bit value in the transfer address register and repeating the address values in a specified range)

  For the extended repeat area, 1 bit (2 bytes) to 27 bits (128 Mbytes) can be set independently for the transfer source or destination.
- Selection of address update methods: Increment/decrement by 1, 2 or 4, fixed, or offset addition

  When offset addition is used to update addresses, the mid-addresses can be skipped during data transfer.
- Transfer of word or longword data to addresses beyond each data boundary

  Data can be divided into an optimal data size (byte or word) according to addresses when transferring data.
- Two kinds of interrupts requested to the CPU

  Transfer end interrupt: Requested after the number of data set by the transfer counter has been completely transferred
  Transfer escape end interrupt: Requested when the remaining transfer size is smaller than the size set for a single transfer request, after a repeat-size transfer is completed, or when an extended repeat area overflow occurs.
- Acceptance of a transfer request can be reported to an external device via the $\overline{\text{EDRAK}}$ pin (only for the EXDMAC).
- Operation of EXDMAC, connected to a dedicated bus, in parallel with a bus master such as the CPU, DTC, or DMAC (only for the EXDMAC).
- Module stop state can be set.

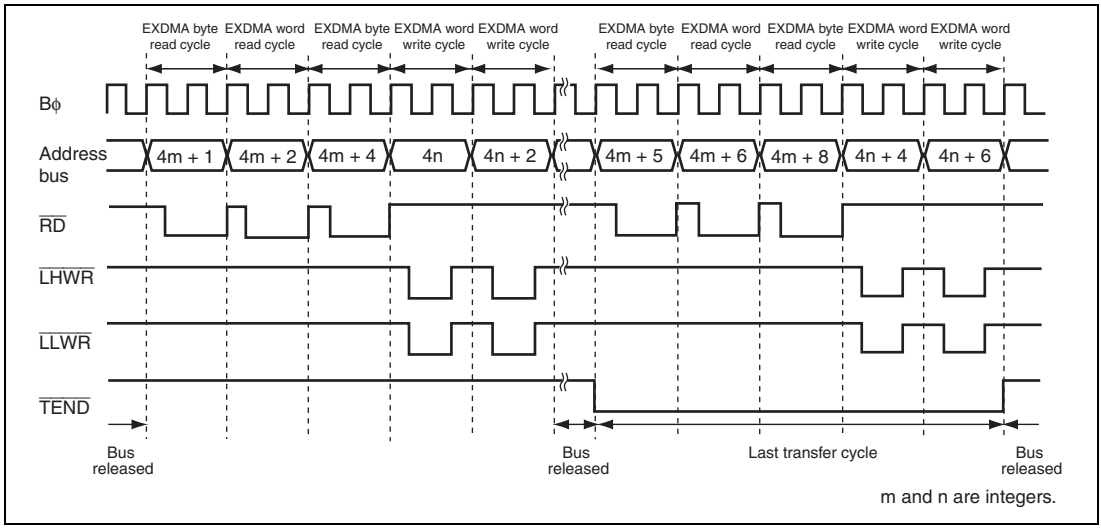

Figure 11.25 Example of Normal Transfer Mode (Cycle Steal Mode) Transfer (Transfer Source EDSAR = Odd Address, Source Address Incremented)

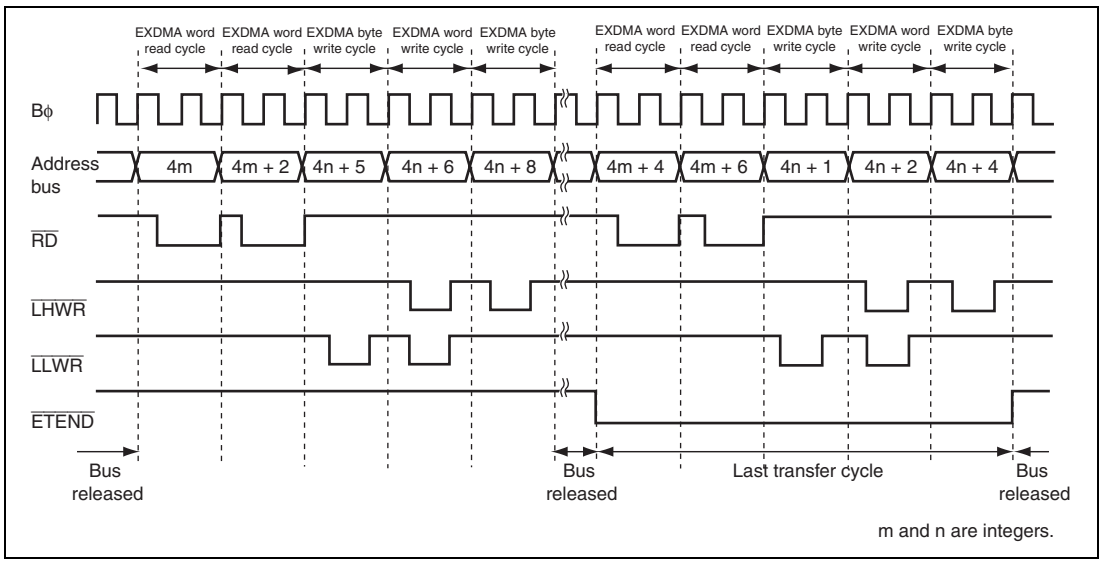

Figure 11.26 Example of Normal Transfer Mode (Cycle Steal Mode) Transfer (Transfer Destination EDDAR = Odd Address, Destination Address Decremented)

12.5.6 Block Transfer Mode

In block transfer mode, one operation transfers one block of data. Either the transfer source or the transfer destination is designated as a block area by the DTS bit in MRB.

The block size is 1 to 256 bytes (1 to 256 words, or 1 to 256 longwords). When the transfer of one block ends, the block size counter (CRAL) and address register (SAR when DTS = 1 or DAR when DTS = 0) specified as the block area is restored to the initial state. The other address register is then incremented, decremented, or left fixed. From 1 to 65,536 transfers can be specified. When the specified number of transfers ends, an interrupt is requested to the CPU.

Table 12.8 lists the register function in block transfer mode. Figure 12.9 shows the memory map in block transfer mode.

Table 12.8 Register Function in Block Transfer Mode

Register	Function	Written Back Value
SAR	Source address	DTS =0: Incremented/decremented/fixed*
		DTS = 1: SAR initial value
DAR	Destination address	DTS = 0: DAR initial value
		DTS =1: Incremented/decremented/fixed*
CRAH	Block size storage	CRAH
CRAL	Block size counter	CRAH
CRB	Block transfer counter	CRB – 1

Note: * Transfer information writeback is skipped.

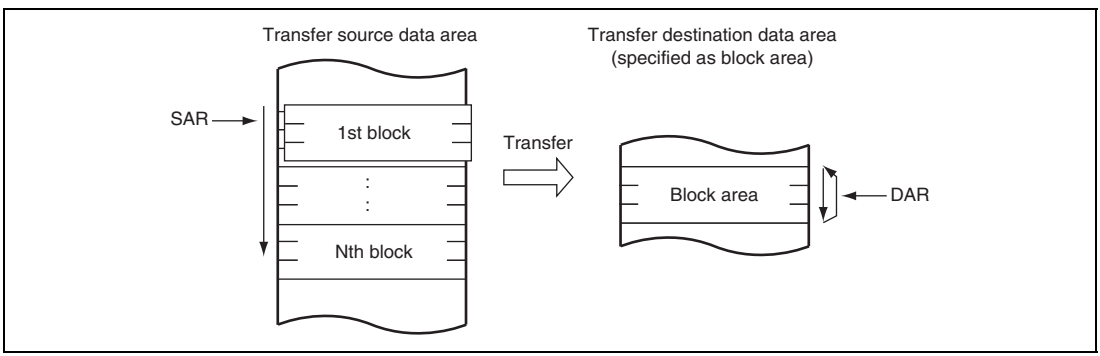

Figure 12.9 Memory Map in Block Transfer Mode (When Transfer Destination is Specified as Block Area)

(5) P33/PO11/TIOCC0/TIOCD0/TCLKB-A/$\overline{\text{DREQ1}}$-B/$\overline{\text{EDREQ3}}$

The pin function is switched as shown below according to the combination of the TPU and PPG register settings and P33DDR bit setting.

		Setting		
		TPU	PPG	I/O Port
Module Name	Pin Function	TIOCD0_OE	PO11_OE	P33DDR
TPU	TIOCD0 output	1	—	—
PPG	PO11 output	0	1	—
I/O port	P33 output	0	0	1
	P33 input (initial value)	0	0	0

(6) P32/PO10/TIOCC0/TCLKA-A/$\overline{\text{DACK0}}$-B/$\overline{\text{EDACK2}}$

The pin function is switched as shown below according to the combination of the EXDMAC, DMAC, TPU, and PPG register settings and P32DDR bit setting.

		Setting				
		EXDMAC	DMAC	TPU	PPG	I/O Port
Module Name	Pin Function	$\overline{\text{EDACK2}}$_OE	$\overline{\text{DACK0B}}$_OE	TIOCC0_OE	PO10_OE	P32DDR
EXDMAC	$\overline{\text{EDACK2}}$ output	1	—	—	—	—
DMAC	$\overline{\text{DACK0}}$-B output	0	1	—	—	—
TPU	TIOCC0 output	0	0	1	—	—
PPG	PO10 output	0	0	0	1	—
I/O port	P32 output	0	0	0	0	1
	P32 input (initial value)	0	0	0	0	0

Bit	Bit Name	Initial value	R/W	Description
3	TGIED	0	R/W	TGR Interrupt Enable D Enables/disables interrupt requests (TGID) by the TGFD bit when the TGFD bit in TSR is set to 1 in channels 0 and 3. In channels 1, 2, 4, and 5, bit 3 is reserved. It is always read as 0 and cannot be modified. 0: Interrupt requests (TGID) by TGFD bit disabled 1: Interrupt requests (TGID) by TGFD bit enabled
2	TGIEC	0	R/W	TGR Interrupt Enable C Enables/disables interrupt requests (TGIC) by the TGFC bit when the TGFC bit in TSR is set to 1 in channels 0 and 3. In channels 1, 2, 4, and 5, bit 2 is reserved. It is always read as 0 and cannot be modified. 0: Interrupt requests (TGIC) by TGFC bit disabled 1: Interrupt requests (TGIC) by TGFC bit enabled
1	TGIEB	0	R/W	TGR Interrupt Enable B Enables/disables interrupt requests (TGIB) by the TGFB bit when the TGFB bit in TSR is set to 1. 0: Interrupt requests (TGIB) by TGFB bit disabled 1: Interrupt requests (TGIB) by TGFB bit enabled
0	TGIEA	0	R/W	TGR Interrupt Enable A Enables/disables interrupt requests (TGIA) by the TGFA bit when the TGFA bit in TSR is set to 1. 0: Interrupt requests (TGIA) by TGFA bit disabled 1: Interrupt requests (TGIA) by TGFA bit enabled

Note: * The bit 7 in TIER of unit 1 is a reserved bit. This bit is always read as 0 and the initial value should not be changed.

14.3.5 Timer Status Register (TSR)

TSR indicates the status of each channel. The TPU has six TSR registers, one for each channel.

Bit	7	6	5	4	3	2	1	0
Bit Name	TCFD	—	TCFU	TCFV	TGFD	TGFC	TGFB	TGFA
Initial Value	1	1	0	0	0	0	0	0
R/W	R	—	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to bits 5 to 0, to clear flags.

15.3.1 Next Data Enable Registers H, L (NDERH, NDERL)

NDERH and NDERL enable/disable pulse output on a bit-by-bit basis.

- NDERH

Bit	7	6	5	4	3	2	1	0
Bit Name	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8
Initial Value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- NDERL

Bit	7	6	5	4	3	2	1	0
Bit Name	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
Initial Value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

19.8 IrDA Operation

If the IrDA function is enabled using the IrE bit in IrCR, the TxD5 and RxD5 pins in SCI_5 are allowed to encode and decode the waveform based on the IrDA Specifications version 1.0 (function as the IrTxD and IrRxD pins)*. Connecting these pins to the infrared data transceiver achieves infrared data communication based on the system defined by the IrDA Specifications version 1.0.

In the system defined by the IrDA Specifications version 1.0, communication is started at a transfer rate of 9600 bps, which can be modified later as required. Since the IrDA interface provided by this LSI does not incorporate the capability of automatic modification of the transfer rate, the transfer rate must be modified through programming.

Figure 19.36 is the IrDA block diagram.

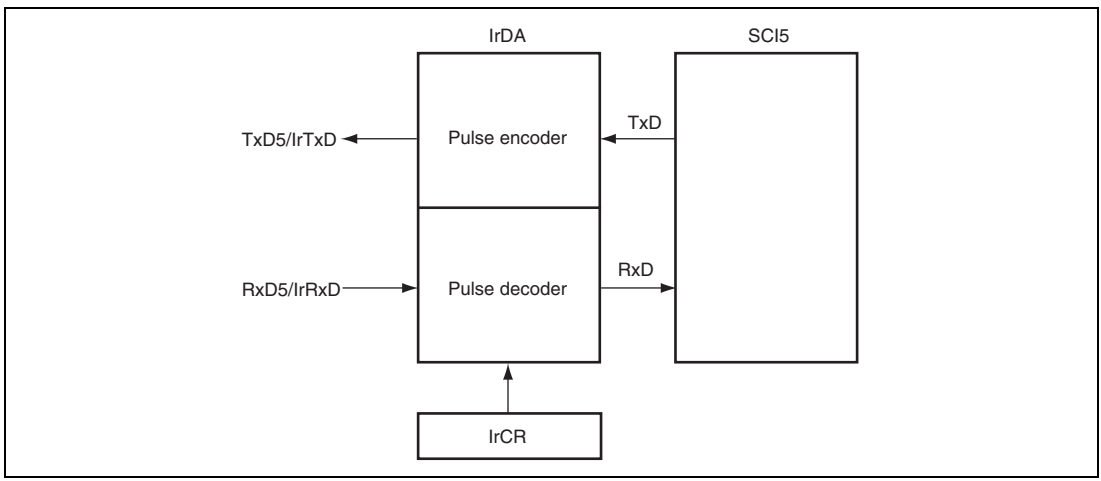

Figure 19.36 IrDA Block Diagram

Note: * The IrDA function should be used when the ABCS bit in SEMR_5 is set to 0 and the ACS3 to ACS0 bits in SEMR_5 are set to B'0000.

20.3.25 Endpoint Information Register (EPIR)

This register sets the information for each endpoint. Each endpoint needs five bytes to store the information. Writing data should be done in sequence starting at logical endpoint 0. Do not write data of more than 50 bytes (five bytes multiplied by ten endpoints) to this register. The information should be written to this register only once at reset and no data should be written after that. Description of writing data for one endpoint is shown below.

Although this register consists of one register to which data is written sequentially for one address, the write data for the endpoint 0 is described as EPIR00 to EPIR05 (EPIR endpoint number in write order) to make the explanation understood easier. Write should start at EPIR00.

Bit	7	6	5	4	3	2	1	0
Bit Name	D7	D6	D5	D4	D3	D2	D1	D0
Initial Value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	W	W	W	W	W	W	W	W

- EPIR00

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	D7 to D4	Undefined	W	Endpoint Number [Enable setting range] 0 to 3
3, 2	D3, D2	Undefined	W	Endpoint Configuration Number [Enable setting range] 0 or 1
1, 0	D1, D0	Undefined	W	Endpoint Interface Number [Enable setting range] 0 to 3

- EPIR01

Bit	Bit Name	Initial Value	R/W	Description
7, 6	D7, D6	Undefined	W	Endpoint Alternate Number [Possible setting range] 0 or 1
5, 4	D5, D4	Undefined	W	Endpoint Transmission [Possible setting range] 0: Control 1: Setting prohibited 2: Bulk 3: Interrupt
3	D3	Undefined	W	Endpoint Transmission Direction [Possible setting range] 0: Out 1: In
2 to 0	D2 to D0	Undefined	W	Reserved [Possible setting range] Fixed to 0.

- EPIR02

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	D7 to D1	Undefined	W	Endpoint Maximum Packet Size [Possible setting range] 0 to 64
0	D0	Undefined	W	Reserved [Possible setting range] Fixed to 0.

- EPIR03

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Reserved [Possible setting range] Fixed to 0.

20.3.27 Transceiver Test Register 1 (TRNTREG1)

TRNTREG1 is a test register that can monitor the on-chip transceiver input signal.

Setting bits PTSTE and txenl in TRNTREG0 to 1 enables monitoring the on-chip transceiver input signal. Table 20.5 shows the relationship between pin input and TRNTREG1 monitoring value.

Bit	7	6	5	4	3	2	1	0
Bit Name	—	—	—	—	—	xver_data	dpls	dmns
Initial Value	0	0	0	0	0	—*	—*	—*
R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	xver_data	—*	R	On-Chip Transceiver Input Signal Monitor
1	dpls	—*	R	xver_data: Monitors the differential input level (xver_data) signal of the on-chip transceiver.
0	dmns	—*	R	dpls: Monitors the USD+ (dpls) signal of the on-chip transceiver.
				dmns: Monitors the USD- (dmns) signal of the on-chip transceiver.

Note: * Determined by the state of pins, VBUS, USD+, and USD-

22.3.2 A/D Control/Status Register for Unit 0 (ADCSR_0)

ADCSR_0 controls A/D conversion operations.

Bit	7	6	5	4	3	2	1	0
Bit Name	ADF	ADIE	ADST	-	CH3	CH2	CH1	CH0
Initial Value	0	0	0	0	0	0	0	0
R/W	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Only 0 can be written to this bit, to clear the flag.

Bit	Bit Name	Initial Value	R/W	Description
7	ADF	0	R/(W)*	A/D End Flag A status flag that indicates the end of A/D conversion. [Setting conditions] • Completion of A/D conversion in single mode • Completion of A/D conversion on all specified channels in scan mode [Clearing conditions] • Writing of 0 after reading ADF = 1 (When the CPU is used to clear this flag by writing 0 while the corresponding interrupt is enabled, be sure to read the flag after writing 0 to it.) • Reading from ADDR after activation of the DMAC or DTC by an ADI interrupt
6	ADIE	0	R/W	A/D Interrupt Enable Setting this bit to 1 enables ADI interrupts by ADF.
5	ADST	0	R/W	A/D Start Clearing this bit to 0 stops A/D conversion, and the A/D converter enters wait state. Setting this bit to 1 starts A/D conversion. In single mode, this bit is cleared to 0 automatically when A/D conversion on the specified channel ends. In scan mode, A/D conversion continues sequentially on the specified channels until this bit is cleared to 0 by software, a reset, or hardware standby mode. Note: Do not write to ADST when activation is by an external trigger. For details, see section 22.7.3, Notes on A/D activation by an External Trigger.
4	—	0	R/W	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
3	CH3	0	R/W	Channel Select 3 to 0
2	CH2	0	R/W	Selects analog input together with bits SCANE and SCANS in ADCR.
1	CH1	0	R/W	
0	CH0	0	R/W	• When SCANE = 0 and SCANS = x 0000: AN0 0001: AN1 0010: AN2 0011: AN3 0100: AN4 0101: AN5 0110: AN6 0111: AN7 1xxx: Setting prohibited • When SCANE = 1 and SCANS = 0 0000: AN0 0001: AN0 and AN1 0010: AN0 to AN2 0011: AN0 to AN3 0100: AN4 0101: AN4 and AN5 0110: AN4 to AN6 0111: AN4 to AN7 1xxx: Setting prohibited • When SCANE = 1 and SCANS = 1 0000: AN0 0001: AN0 and AN1 0010: AN0 to AN2 0011: AN0 to AN3 0100: AN0 to AN4 0101: AN0 to AN5 0110: AN0 to AN6 0111: AN0 to AN7 1xxx: Setting prohibited

[Legend]

x: Don't care

Note: * Only 0 can be written to this bit, to clear the flag.

(f) Memory Read

The boot program will return the data in the specified address.

Command

H'52	Size	Area	Read address	
Read size				SUM

- Command: H'52 (1 byte): Memory read
- Size (1 byte): Amount of data that represents the area, read address, and read size (fixed at 9)
- Area (1 byte)

  H'00: User boot MAT

  H'01: User MAT

  An address error occurs when the area setting is incorrect.
- Read address (4 bytes): Start address to be read from
- Read size (4 bytes): Size of data to be read
- SUM (1 byte): Checksum

Response

H'52	Read size						
Data	···						
SUM							

- Response: H'52 (1 byte): Response to memory read
- Read size (4 bytes): Size of data to be read
- Data (n bytes): Data for the read size from the read address
- SUM (1 byte): Checksum

Error Response

H'D2	ERROR

- Error response: H'D2 (1 byte): Error response to memory read
- ERROR: (1 byte): Error code

  H'11: Sum check error

  H'2A: Address error

  The read address is not in the MAT.

  H'2B: Size error

  The read size exceeds the MAT.

29.2 Register Bits

Register addresses and bit names of the on-chip peripheral modules are described below.

Each line covers eight bits, and 16-bit and 32-bit registers are shown as 2 or 4 lines, respectively.

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	Module
TCR_4	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR_4
TCR_5	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR_5
TCSR_4	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0	TMR_4
TCSR_5	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0	TMR_5
TCORA_4									TMR_4
TCORA_5									TMR_5
TCORB_4									TMR_4
TCORB_5									TMR_5
TCNT_4									TMR_4
TCNT_5									TMR_5
TCCR_4	—	—	—	—	TMRIS	—	ICKS1	ICKS0	TMR_4
TCCR_5	—	—	—	—	TMRIS	—	ICKS1	ICKS0	TMR_5
CRCCR	DORCLR	—	—	—	—	LMS	G1	G0	CRC
CRCDIR									
CRCDOR									
TCR_6	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR_6
TCR_7	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR_7
TCSR_6	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0	TMR_6
TCSR_7	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0	TMR_7
TCORA_6									TMR_6
TCORA_7									TMR_7
TCORB_6									TMR_6
TCORB_7									TMR_7
TCNT_6									TMR_6
TCNT_7									TMR_7

Item	Page	Revision (See Manual for Details)
28.2.4 Deep Standby Control Register (DPSBYCR)	1246	Amended DPSBYCR controls deep software standby mode. DPSBYCR is not initialized by the internal reset signal upon exit from deep software standby mode.
	1248	Amended Descriptions for bit 5 in the register table: On-chip RAM Power Off 2 RAMCUT 2, 1, and 0 control the internal power supply to the on-chip RAM and USB in deep software standby mode. For details, see descriptions of the RAMCUT0 bit
		Amended Descriptions for bit 4 in the register table: On-chip RAM Power Off 1 RAMCUT 2, 1, and 0 control the internal power supply to the on-chip RAM and USB in deep software standby mode. For details, see descriptions of the RAMCUT0 bit.
		Amended Descriptions for bit 0 in the register table: On-chip RAM Power Off 0 RAMCUT 2, 1, and 0 control the internal power supply to the on-chip RAM and USB in deep software standby mode. For details, see descriptions of the RAMCUT0 bit.
28.2.5 Deep Standby Wait Control Register (DPSWCR)	1249	Amended DPSWCR is not initialized by the internal reset signal upon exit from deep software standby mode.