User's Manual

32

SH7734

Additional Document
for User's Manual: Hardware

Renesas 32-Bit RISC Microcomputer
SuperH™ RISC engine Family / SH-4A Series

Renesas Electronics
www.renesas.com

Rev.1.10 Aug 2012

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(2012.4)

Contents

Section 1 Stream Interface (STIF)

This LSI has a 2-channel stream interface (STIF).

1.1 Features

- Two-channel bidirectional interface
- Supports TS packets (packet size: 188 bytes).
- Supports TTS packets (packet size: 192 bytes).
- Supports PS packets (packet size: specified by the size register).
- 8-bit parallel transfer or 1-bit serial transfer is selectable.
- Transfer direction is settable for each channel.
- Polarity of each clock signal, request signal, synchronizing signal, and data enable signal is selectable.
- A PCR clock recovery module (PCRRCV) incorporated

Figure 1.1 shows a block diagram of the STIF.

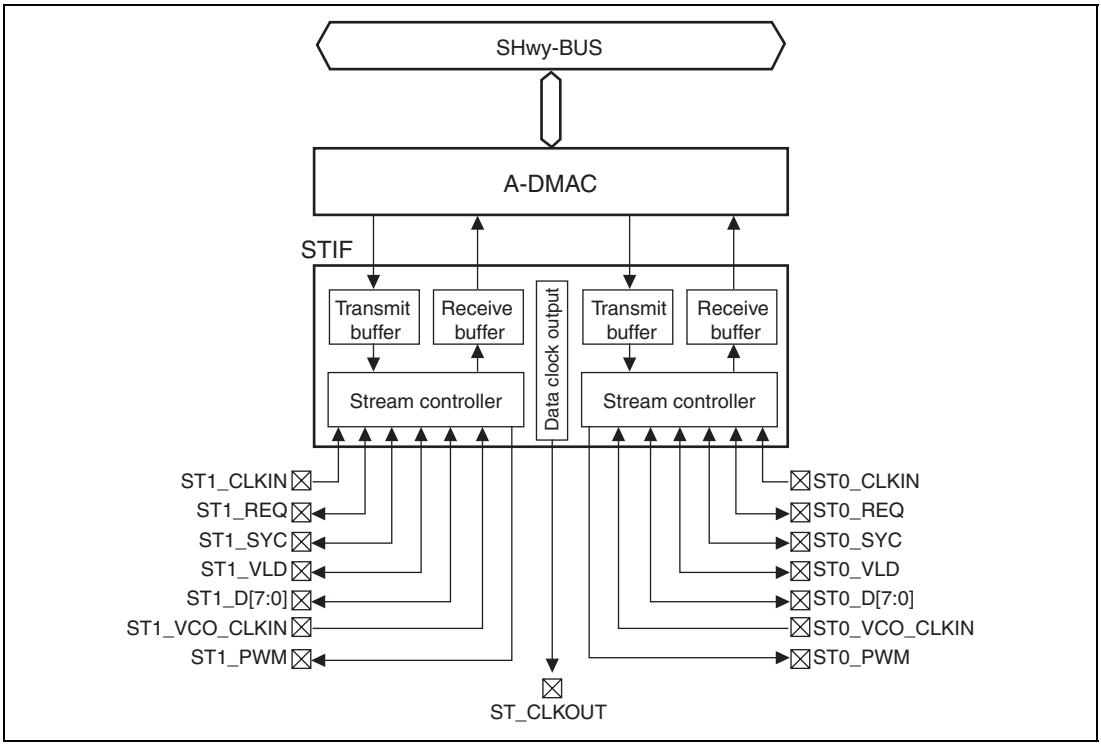


Figure 1.1 Block Diagram of STIF

1.2 Input/Output Pins

Table 1.1 shows the pin configuration of the STIF.

Table 1.1 Pin Configuration

Name	I/O	Function
ST_CLKOUT	Output	Data clock output (common to channels) This is defined by bits [15:12] in FRQCR2 of the CPG.
ST0_CLKIN	Input	Data clock input
ST0_REQ	I/O	Request signal
ST0_SYC	I/O	Synchronizing signal
ST0_VLD	I/O	Data enable
ST0_D[7:0]	I/O	Data (ST0_D[0] is used in serial mode)
ST0_VCO_CLKIN	Input	The MPEG base clock is input from the external 27-MHz voltage controlled oscillator (VCO).
ST0_PWM	Output	The 27-MHz VCO is controlled through the low-pass filter (LPF).
ST1_CLKIN	Input	Data clock input
ST1_REQ	I/O	Request signal
ST1_SYC	I/O	Synchronizing signal
ST1_VLD	I/O	Data enable
ST1_D[7:0]	I/O	Data (ST1_D[0] is used in serial mode)
ST1_VCO_CLKIN	Input	The MPEG base clock is input from the external 27-MHz VCO.
ST1_PWM	Output	The 27-MHz VCO is controlled through the LPF.

1.3 Register Descriptions

Table 1.2 (1) shows the register configurations. Table 1.2 (2) shows the register states in each processing mode.

Table 1.2 (1) Register Configuration

Channel	Register Name	Abbreviation	Address	Access Size
0	STIF mode select register	STMDR_0	H'FFEE 0000	32
	STIF control register	STCTLR_0	H'FFEE 0004	32
	STIF internal counter control register	STCNTCR_0	H'FFEE 0008	32
	STIF internal counter set register	STCNTVR_0	H'FFEE 000C	32
	STIF status register	STSTR_0	H'FFEE 0010	32
	STIF interrupt enable register	STIER_0	H'FFEE 0014	32
	STIF transfer size register	STSIZER_0	H'FFEE 0018	32
	STIFPWM mode register	STPWMMR_0	H'FFEE 0020	32
	STIFPWM control register	STPWMCR_0	H'FFEE 0024	32
	STIFPWM register	STPWMR_0	H'FFEE 0028	32
	STIFPCR0 register	STPCR0R_0	H'FFEE 002C	32
	STIFPCR1 register	STPCR1R_0	H'FFEE 0030	32
	STIFSTC0 register	STSTC0R_0	H'FFEE 0034	32
	STIFSTC1 register	STSTC1R_0	H'FFEE 0038	32
	STIF lock control register	STLKCR_0	H'FFEE 003C	32
	STIF debugging status register	STDBGR_0	H'FFEE 0060	32
1	STIF mode select register	STMDR_1	H'FFEE8000	32
	STIF control register	STCTLR_1	H'FFEE 8004	32
	STIF internal counter control register	STCNTCR_1	H'FFEE 8008	32
	STIF internal counter set register	STCNTVR_1	H'FFEE 800C	32
	STIF status register	STSTR_1	H'FFEE 8010	32
	STIF interrupt enable register	STIER_1	H'FFEE 8014	32
	STIF transfer size register	STSIZER_1	H'FFEE 8018	32
	STIFPWM mode register	STPWMMR_1	H'FFEE 8020	32
	STIFPWM control register	STPWMCR_1	H'FFEE 8024	32
	STIFPWM register	STPWMR_1	H'FFEE 8028	32
	STIFPCR0 register	STPCR0R_1	H'FFEE 802C	32

Channel	Register Name	Abbreviation	Address	Access Size
1	STIFPCR1 register	STPCR1R_1	H'FFEE 8030	32
	STIFSTC0 register	STSTC0R_1	H'FFEE 8034	32
	STIFSTC1 register	STSTC1R_1	H'FFEE 8038	32
	STIF lock control register	STLKCR_1	H'FFEE 803C	32
	STIF debugging status register	STDBGR_1	H'FFEE 8060	32

Table 1.2 (2) State of Registers in Each Operating Mode (Common to Each Channel)

Abbr.	Power-on Reset	Manual Reset	Sleep	Software Standby	Module Standby	Deep Standby
STMDR	Initialized	Initialized	Retained	Retained	Retained	Initialized
STCTLR	Initialized	Initialized	Retained	Retained	Retained	Initialized
STCNTCR	Initialized	Initialized	Retained	Retained	Retained	Initialized
STCNTVR	Initialized	Initialized	Retained	Retained	Retained	Initialized
STSTR	Initialized	Initialized	Retained	Retained	Retained	Initialized
STIER	Initialized	Initialized	Retained	Retained	Retained	Initialized
STSIZER	Initialized	Initialized	Retained	Retained	Retained	Initialized
STPWMMR	Initialized	Initialized	Retained	Retained	Retained	Initialized
STPWMCR	Initialized	Initialized	Retained	Retained	Retained	Initialized
STPWMR	Initialized	Initialized	Retained	Retained	Retained	Initialized
STPCR0R	Initialized	Initialized	Retained	Retained	Retained	Initialized
STPCR1R	Initialized	Initialized	Retained	Retained	Retained	Initialized
STSTC0R	Initialized	Initialized	Retained	Retained	Retained	Initialized
STSTC1R	Initialized	Initialized	Retained	Retained	Retained	Initialized
STLKCR	Initialized	Initialized	Retained	Retained	Retained	Initialized
STDBGR	Initialized	Initialized	Retained	Retained	Retained	Initialized

1.3.1 STIF Mode Select Register (STMDR)

STMDR is a 32-bit register that selects operating mode, etc. of the on-chip STIF module. STMDR is initialized to H'00000000 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
14	LSBSEL	0	R/W	Selects MSB first or LSB first in serial mode. 0: MSB-first data input/output 1: LSB-first data input/output
13	EDGSEL	0	R/W	Selects input/output timing of STn_REQ, STn_SYC, STn_VLD, and STn_D[7:0]. 0: Output and sampled at the rising edge of the synchronizing clock 1: Output and sampled at the falling edge of the synchronizing clock The synchronizing clock is defined by the CLKSEL bit in this register and bits [15:12] in FRQCR2 of the CPG.
12	CLKSEL	0	R/W	Selects synchronizing clock for stream transmit mode 0: STn_SYC, STn_VLD, and STn_D[7:0] are output in synchronization with ST_CLKOUT. STn_REQ is sampled in synchronization with ST_CLKOUT 1: STn_SYC, STn_VLD, and STn_D[7:0] are output in synchronization with STn_CLKIN. STn_REQ is sampled in synchronization with STn_CLKIN.
11 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
7	REQACTSEL	0	R/W	Selects the active polarity of STn_REQ. 0: Active-high 1: Active-low
6	VLDACTSEL	0	R/W	Selects the active polarity of STn_VLD. 0: Active-high 1: Active-low
5	SYCACTSEL	0	R/W	Selects the active polarity of STn_SYC. 0: Active-high 1: Active-low
4	IOSEL	0	R/W	Selects stream input or output direction. 0: Input (from an external device to this LSI) 1: Output (from this LSI to an external device)
3	IFMDSEL3	0	R/W	These bits select operating mode.
2	IFMDSEL2	0	R/W	0000: TS serial mode 1
1	IFMDSEL1	0	R/W	0001: TS parallel mode 1
0	IFMDSEL0	0	R/W	0010: TS serial mode 2 0011: TS parallel mode 2 0100: TS serial mode 3 0101: TS parallel mode 3 0110: Reserved (setting prohibited) 0111: Reserved (setting prohibited) 1000: TTS serial mode 1001: TTS parallel mode 1010: Reserved (setting prohibited) 1011: Reserved (setting prohibited) 1100: PS serial mode 1101: PS parallel mode 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)

[Legend] n = 0, 1

1.3.2 STIF Control Register (STCTLR)

STCTLR is a 32-bit register that sets the recovery processing switching threshold value and enables/disables DMA transfer requests. STCTLR is initialized to H'00000000 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 10 9	RCVTM2 RCVTM1 RCVTM0	0 0 0	R/W R/W R/W	These bits set the recovery processing switching threshold value for packet output in TS mode 1 or TS mode 2. These bits are valid when RCV = 1. 000: Approximately 0.625 seconds 001: Approximately 1.25 seconds 010: Approximately 2.5 seconds 011: Approximately 5 seconds 100: Approximately 10 seconds 101: Approximately 20 seconds 110: Approximately 40 seconds 111: Approximately 80 seconds The recovery functions are processed as follows: Recovery function (1) When the internal counter value exceeds the timestamp value and the difference is smaller than the set threshold value, the packet is output immediately. Recovery function (2) When the internal counter value exceeds the timestamp value and the difference is larger than the set threshold value, the packet is discarded and the recovery processing restarts with the next packet. (The next packet is output immediately, and the packet's timestamp is reloaded to the internal counter for timestamp at the same time.) Recovery function (3) When the internal counter value is under the timestamp value but the difference is larger than the set threshold value, the packet is discarded and the recovery processing restarts with the next packet. (The next packet is output immediately, and the packet's timestamp is reloaded to the internal counter for timestamp at the same time.)

Bit	Bit Name	Initial Value	R/W	Description
8	RCV	0	R/W	Enables the recovery functions when outputting packets in TS mode 1 or TS mode 2. 0: Recovery functions disabled 1: Recovery functions enabled
7	TRICK	0	R/W	Enables the function that transfers stream independently of timestamp when outputting packets in TS mode 1 or TS mode 2. 0: Transfer function disabled 1: Transfer function enabled
6 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	REQEN	0	R/W	Enables or disables DMA transfer requests to the A-DMAC. 0: Disabled 1: Enabled
1	EN	0	R/W	Enables or disables stream input/output. 0: Disabled 1: Enabled
0	SRST	0	R/W	Setting this bit to 1 causes the internal state of this module to be initialized with register settings retained. When a TS packet is received for the first time after the initialization, the timestamp value of the TS packet is reloaded to the internal counter for timestamp. While 1 is read from this bit, the initialization is in progress. This bit is automatically cleared to 0. Whenever the STMDR setting is modified, be sure to set SRST to 1 to initialize this module and then set EN and REQEN to 1 to enable stream transfer.

1.3.3 STIF Internal Counter Control Register (STCNTCR)

STCNTCR is a 32-bit register to control the internal counter for timestamp. STCNTCR is initialized to H'00000000 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	CRD	0	R/W	Setting this bit to 1 causes the internal counter value for timestamp to be read to STCNTVR. This bit is automatically cleared to 0.
2	CSTP	0	R/W	Stops the internal counter for timestamp. 0: Count operation is continued. 1: Counter is stopped with its value retained.
1	CSET	0	R/W	Setting this bit to 1 causes the STCNTVR value to be reloaded to the internal counter for timestamp. This bit is automatically cleared to 0.
0	CRST	0	R/W	Setting this bit to 1 causes the internal counter for timestamp to be initialized to H'00000000. This bit is automatically cleared to 0.

1.3.4 STIF Internal Counter Set Register (STCNTVR)

STCNTVR is a 32-bit register that reads or reloads the value of the internal counter for timestamp in combination with the settings of the CRD and CSET bits in STCNTCR. STCNTVR is initialized to H'00000000 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	VLU31 to VLU0	All 0	R/W	Internal counter value for timestamp

1.3.5 STIF Status Register (STSTR)

STSTR is a 32-bit register that indicates the status of the recovery functions, packet transmission/reception, and PCR clock recovery. STSTR is initialized to H'00000000 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	LKZF	0	R/W	Indicates whether the PLL error amount (internal STC - internal PCR) falls within threshold value range LKCYC when a PCR packet is received. 0: Within threshold value range (PLL error amount (internal STC - internal PCR) =< LKCYC) 1: Outside threshold value range (PLL error amount (internal STC - internal PCR) > LKCYC) This bit is cleared to 0 by writing 1.

Bit	Bit Name	Initial Value	R/W	Description
11	LKF	0	R/W	Indicates PLL lock status. 0: PLL unlocked In the case of ULCNT >= ULREF due to continued LKZF = 1 state (outside threshold value range) when a PCR packet is received with PLL locked (LKF = 1) 1: PLL locked In the case of LKCNT >= LKREF due to continued LKZF = 0 state (within threshold value range) when a PCR packet is received with PLL unlocked (LKF = 0) This bit is cleared to 0 by writing 1.
10	DISF	0	R/W	Status flag bit that indicates the discontinuity_indicator (table 1.6) of received PCR_PID. This bit is set to 1 upon completion of transfer (internal PCR → STC → internal STC). This bit is cleared to 0 by writing 1.
9	UNZF	0	R/W	This bit is set to 1 when data transfer from internal PCR register to STC counter and data transfer from STC counter to internal STC register are completed and the comparison of the upper data of received PCR_PID does not match (internal STC - internal PCR exceeded the acceptable comparison result range specified by PWMCYC; see figure 1.9). This bit is also set to 1 when PCR_PID (table 1.6) is received after "discont." Furthermore, if the PWM control variable with a bit width of the effective comparison bit count "n" specified by PWMCYC is -(2^n), this bit is set to 1 as invalid PWM control variable (ILGL = 1 hereinafter) in the same manner as above. This bit is cleared to 0 by writing 1.
8	PCRF	0	R/W	This bit is set to 1 when data transfer from internal PCR register to STC counter and data transfer from STC counter to internal STC register are completed. These transfers take place when a packet whose PCR_PID was detected satisfies the conditions in table 1.5. This bit is cleared to 0 by writing 1.

Bit	Bit Name	Initial Value	R/W	Description
7	TENDF	0	R/W	Indicates completion of output packet transfer for the transfer data size specified by STSIZER in PS mode. This bit is cleared to 0 by writing 1.
6	RENDF	0	R/W	Indicates completion of input packet transfer for the transfer data size specified by STSIZER in PS mode. This bit is cleared to 0 by writing 1.
5	RCVF3	0	R/W	This bit is set to 1 when recovery function (3) is activated when outputting a packet in TS mode 1 or TS mode 2. This bit is cleared to 0 by writing 1.
4	RCVF2	0	R/W	This bit is set to 1 when recovery function (2) is activated when outputting a packet in TS mode 1 or TS mode 2. This bit is cleared to 0 by writing 1.
3	RCVF1	0	R/W	This bit is set to 1 when recovery function (1) is activated when outputting a packet in TS mode 1 or TS mode 2. This bit is cleared to 0 by writing 1.
2	UPF	0	R/W	This bit is set to 1 when a packet shorter than 188 bytes is received in TS mode 1 or TS mode 2. Such packets are discarded. This bit is cleared to 0 by writing 1.
1	OPF	0	R/W	When a packet of 189 bytes or longer is received in TS mode 1 or TS mode 2, this bit is set to 1 when reception of the 189th byte is completed. Packets exceeding 188 bytes are discarded. This bit is cleared to 0 by writing 1.
0	$\overline{\text{O}}$VF	0	R/W	This bit is set to 1 when data read by the A-DMAC is delayed and therefore the receive data which came later is discarded in TS mode 1 or TS mode 2. This bit is cleared to 0 by writing 1.

1.3.6 STIF Interrupt Enable Register (STIER)

STIER is a 32-bit register to control various interrupt requests. STIER is initialized to H'00000000 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	LKZE	0	R/W	Enables or disables LKZF interrupt requests. 0: LKZF interrupt requests are disabled. 1: LKZF interrupt requests are enabled.
11	LKE	0	R/W	Enables or disables LKF interrupt requests. 0: LKF interrupt requests are disabled. 1: LKF interrupt requests are enabled.
10	DISE	0	R/W	Enables or disables DISF interrupt requests. 0: DISF interrupt requests are disabled. 1: DISF interrupt requests are enabled.
9	UNZE	0	R/W	Enables or disables UNZF interrupt requests. 0: UNZF interrupt requests are disabled. 1: UNZF interrupt requests are enabled.
8	PCRE	0	R/W	Enables or disables PCRF interrupt requests. 0: PCRF interrupt requests are disabled. 1: PCRF interrupt requests are enabled.
7	TENDE	0	R/W	Enables or disables TENDF interrupt requests. 0: TENDF interrupt requests are disabled. 1: TENDF interrupt requests are enabled.
6	RENDE	0	R/W	Enables or disables RENDF interrupt requests. 0: RENDF interrupt requests are disabled. 1: RENDF interrupt requests are enabled.
5	RCVE3	0	R/W	Enables or disables RCVF3 interrupt requests. 0: RCVF3 interrupt requests are disabled. 1: RCVF3 interrupt requests are enabled.

Bit	Bit Name	Initial Value	R/W	Description
4	RCVE2	0	R/W	Enables or disables RCVF2 interrupt requests. 0: RCVF2 interrupt requests are disabled. 1: RCVF2 interrupt requests are enabled.
3	RCVE1	0	R/W	Enables or disables RCVF1 interrupt requests. 0: RCVF1 interrupt requests are disabled. 1: RCVF1 interrupt requests are enabled.
2	UPE	0	R/W	Enables or disables UPF interrupt requests. 0: UPF interrupt requests are disabled. 1: UPF interrupt requests are enabled.
1	OPE	0	R/W	Enables or disables OPF interrupt requests. 0: OPF interrupt requests are disabled. 1: OPF interrupt requests are enabled.
0	$\overline{O}$VE	0	R/W	Enables or disables $\overline{O}$VF interrupt requests. 0: $\overline{O}$VF interrupt requests are disabled. 1: $\overline{O}$VF interrupt requests are enabled.

1.3.7 STIF Transfer Size Register (STSIZER) (n = 0,1)

STSIZER is a 32-bit register that specifies a transfer byte count for PS mode. STSIZER is initialized to H'FFFFFFFF by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	SIZE31 to SIZE0	All 1	R/W	Transfer byte count for PS mode

1.3.8 STIFPWM Mode Register (STPWMMR)

STPWMMR is a 32-bit register that selects PWM mode, sets PWM control cycle, reference bit shift amount, and reference clock, enables/disables PID filtering, and sets the PID of a PCR packet to be filtered. STPWMMR is initialized to H'00000000 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28 to 16	PID12 to PID0	All 0	R/W	These bits set the PID (PCR_PID) of filtering target PCR packet.
15	PIDEN	0	R/W	Enables or disables PCR packet filtering. 0: Filtering is disabled. 1: Filtering is enabled.
14	PWMUEN	0	R/W	Selects whether to reflect the PWM control difference (internal STC register - internal PCR register) in the PWM control output according to the comparison of the residual upper bits (comparison target bits) in the comparison of bits 0 to 11. The comparison result of target bits is reflected in UNZF. This bit is valid only when PWMSEL is 0. 0: When the comparison results in a mismatch, PWM control variable is reflected in PWM control. [Match: UNZF = 0] PWM control variable is reflected in PWM output control. [Mismatch: UNZF = 1] PWM control variable is reflected in PWM output control. 1: When the comparison results in a mismatch, PWM control variable is not reflected in PWM control. [Match: UNZF = 0] PWM control variable is reflected in PWM output control. [Mismatch: UNZF = 1] PWM control variable is not reflected in PWM output control.

Bit	Bit Name	Initial Value	R/W	Description
13	PWMSEL	0	R/W	Selects difference (internal STC register - internal PCR register) result or PWMR register value for use as the input to selector 2 (figure 1.9). Also selects PCR arrival pulse or PWMWP as the pulse for reflecting the selector 2 output in the PWM output. 0: PWM control mode is set for the difference (internal STC - internal PCR) result control. PCR arrival pulse and PWMWP are available. 1: PWM control mode is set for the PWMR register control. Only PWMWP is available.
12	PWMSEL2	0	R/W	Selects selector 1 output (figure 1.9) or addition (selector 1 output + internal PWM register value) result for use as input to the internal PWM register. 0: Selector 1 output is set for input to the internal PWM register. 1: Addition (selector 1 output + internal PWM register value) result is set for input to the internal PWM register.
11	PWMCYC3	0	R/W	These bits set a PWM control cycle value based on the PWM reference clock that is set by the PWMDIV bits. See table 1.3. This setting should be modified only when the PIDEN bit is 0.
10	PWMCYC2	0	R/W	
9	PWMCYC1	0	R/W	
8	PWMCYC0	0	R/W	
7	PWMSFT3	0	R/W	These bits set a reference bit position that is used to specify the PWM control variable (internal STC register - internal PCR register). As shown in figure 1.9, the reference bit position of the PWM control variable varies with the PWMSFT value. This setting should be modified only when the PIDEN bit is 0. Reference bit position / Reference bit position 0000: 0 / 1000: 8 0001: 1 / 1001: 9 0010: 2 / 1010: 10 0011: 3 / 1011: 11 0100: 4 / 1100: 12 0101: 5 / 1101: 13 0110: 6 / 1110: 14 0111: 7 / 1111: 15
6	PWMSFT2	0	R/W	
5	PWMSFT1	0	R/W	
4	PWMSFT0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
3	PWMDIV3	0	R/W	These bits set the reference clock of the PWM control output (PWMOUT) as a system clock (Bϕ) division count. Set a division count between 1 and 1024. If a value outside the range is set, the operation of this LSI is not guaranteed. This setting should be modified only when the PIDEN bit is 0. 0000: 1 1000: 256 0001: 2 1001: 512 0010: 4 1010: 1024 0011: 8 1011: 2048 (invalid) 0100: 16 1100: 4096 (invalid) 0101: 32 1101: 8192 (invalid) 0110: 64 1110: 16384 (invalid) 0111: 128 1111: 32768 (invalid)
2	PWMDIV2	0	R/W	
1	PWMDIV1	0	R/W	
0	PWMDIV0	0	R/W	

Table 1.3 PWM Control Cycle

Bits 11 to 8	Description			
PWMCYC3 to PWMCYC0	PWM Cycle (× PWM reference clock)	Acceptable Comparison Bit Count n	Acceptable Comparison (Internal STC - Internal PCR) Result Range *1	PWM Control Variable*2 (ILGL = 1)
0000	2	0	—	—
0001	4	1	–1 to +1	–2
0010	8	2	–3 to +3	–4
0011	16	3	–7 to +7	–8
0100	32	4	–15 to +15	–16
0101	64	5	–31 to +31	–32
0110	128	6	–63 to +63	–64
0111	256	7	–127 to +127	–128
1000	512	8	–255 to +255	–256
1001	1024	9	–511 to +511	–512
1010	2048	10	–1023 to +1023	–1024
1011	4096	11	–2047 to +2047	–2048
1100	8192	12	–4095 to +4095	–4096
1101	16384	13	–8191 to +8191	–8192
1110	32768	14	–16383 to +16383	–16384
1111	65536	15	–32767 to +32767	–32768

Notes: 1. When PWMSEL = 0, if the comparison (internal STC - internal PCR) result exceeds the acceptable comparison result range, the UNZF bit is set to 1.

2. If the PWM control variable is -(2^n), it is treated as an invalid PWM control variable (ILGL = 1) and the UNZF bit is set to 1. The PWM control variable is selected by the PWMSEL bit.

1.3.9 STIFPWM Control Register (STPWMCR)

STPWMCR is a 32-bit register that specifies the generation of write pulses of the internal PCR and STC registers. STPWMCR is initialized to H'00000000 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	STCXP	0	R/W	Setting this bit to 1 causes the STC counter value to be transferred to STSTC0R and STSTC1R registers. This bit is automatically cleared to 0.
7	PWMBRS	0	R/W	Setting this bit to 1 causes the internal PWM register value to be transferred to the STPWM register (PWMB). This bit is automatically cleared to 0.
6	PWMBWP	0	R/W	Setting this bit to 1 causes the STPWM register (PWMB) value to be reflected in the internal PWM register. PWM control is immediately performed with the value that is set in the internal PWM register. Loading with this bit can preferentially be performed independently of the PWMSEL and PWMUEN settings, except when the PWM control variable is an invalid value as described in the UNZF bit field of STSTR. If this bit is set to 1 together with the PWMWP bit, PWMBWP takes precedence. This bit is automatically cleared to 0.
5	PWMRS	0	R/W	Setting this bit to 1 causes the difference (internal STC register - internal PCR register) result to be transferred to the STPWM register (PWM). The difference result is masked by the PWMCYC bits (validity comparison bits) of STPWMMR as shown in figure 1.9. This bit is automatically cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
4	PWMWP	0	R/W	Setting this bit to 1 causes the selector 2 output to be reflected in the internal PWM register. Loading with this bit can be performed preferentially without depending on the PWMSEL and PWMUEN settings, except when the PWM control variable is an invalid value as described in the UNZF bit field of STSTR. If this bit is set to 1 together with the PWMBWP bit, PWMBWP takes precedence. This bit is automatically cleared to 0.
3	STCRS	0	R/W	Setting this bit to 1 causes the internal STC register value to be transferred to STSTC0R and STSTC1R registers. This bit is automatically cleared to 0.
2	STCWP	0	R/W	Setting this bit to 1 causes the STSTC0R and STSTC1R register values to be transferred to the internal STC register. If the transfer conflicts with the data write after PCR is received, the transfer using the write pulse of this register takes precedence. This bit is automatically cleared to 0.
1	PCRRS	0	R/W	Setting this bit to 1 causes the internal PCR register value to be transferred to STPCR0R and STPCR1R registers. This bit is automatically cleared to 0.
0	PCRWP	0	R/W	Setting this bit to 1 causes the STPCR0R and STPCR1R register values to be transferred to the internal PCR register. If the transfer conflicts with the data write after PCR is received, the transfer using the write pulse of this register takes precedence. This bit is automatically cleared to 0.

1.3.10 STIFPWM Register (STPWMR)

STPWMR is a 32-bit register that directly sets PWM control variable. STPWMR is initialized to H'00000000 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	PWMB15 to PWMB0	All 0	R/W	These bits set a PWM control value equivalent to the comparison (internal STC register - internal PCR register) result. To reflect the PWMB value in the PWMOUT output pin as a PWM control variable, set the PWMBWP bit in STPWMCR to 1. Set a PWMB value of two's complement (bit n = sign bit) out of the acceptable comparison bit count n specified by the PWMCYC bits. The acceptable setting range is -(2^n - 1) to +(2^n - 1) where n = acceptable comparison bit count specified by the PWMCYC bits. Do not set the value -(2^n). If -(2^n) is set and is attempted to reflect in the PWM control by setting PWMBWP to 1, it is treated as an invalid PWM control variable setting and the UNZF bit is set to 1. The setting is not reflected in the PWM control output. Note that reflection in the PWM control output is not performed until PWMB is set to a value other than -(2^n).
15 to 0	PWM15 to PWM0	All 0	R/W	These bits set a PWM control value equivalent to the comparison (internal STC register - internal PCR register) result. To reflect the PWM value in the PWMOUT output pin as a PWM control variable, set the PWMWP bit in STPWMCR to 1 with PWMSEL = 1. Set a PWM value of two's complement (bit n = sign bit) out of the acceptable comparison bit count n specified by the PWMCYC bits. The acceptable setting range is -(2^n - 1) to +(2^n - 1) where n = acceptable comparison bit count specified by the PWMCYC bits. Set a value so that the selector 2 output is not the value -(2^n). When the selector 2 output is -(2^n) and the PWM value is reflected in the PWM control by setting PWMWP to 1, the PWM value is treated as an invalid PWM control variable setting and the UNZF bit is set to 1. The setting is not reflected in the PWM control output.

1.3.11 STIFPCR0, STIFPCR1 Registers (STPCR0R, STPCR1R)

STPCR0R and STPCR1R are 32-bit registers that interface with the internal PCR register. STPCR0R and STPCR1R are initialized to H'00000000 by a power-on reset. These registers compose a 42-bit register including PCR base (33 bits) and PCR extension (9 bits). The PCR base and PCR extension are stored in PCRB32 to PCRB0 and PCRX8 to PCRX0 respectively. Reading or writing this 42-bit register does not cause the read/write result to be reflected directly in clock recovery.

- STPCR0R

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	PCRB32 to PCRB23	All 0	R/W	PCR Base

- STPCR1R

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	PCRB22 to PCRB0	All 0	R/W	PCR Base
8 to 0	PCRX8 to PCRX0	All 0	R/W	PCR Extension

Note: If PCR_PID arrives during data read, the read value is overwritten and becomes undefined. Therefore, confirm that there is no PCR_PID during data read with the PCRF bit in STSTR. Specifically, set the PCRF bit to 0 and then start reading the receive data. Whenever PCRF is 1, take this procedure.

1.3.12 STIFSTC0, STIFSTC1 Registers (STSTC0R, STSTC1R)

STSTC0R and STSTC1R are 32-bit registers that interface with the internal STC register. STSTC0R and STSTC1R are initialized to H'00000000 by a power-on reset. These registers compose a 42-bit register including STC base (33 bits) and STC extension (9 bits). The STC base and STC extension are stored in STCB32 to STCB0 and STCX8 to STCX0 respectively. Reading or writing this 42-bit register does not cause the read/write result to be reflected directly in clock recovery.

- STSTC0R

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	STCB32 to STCB23	All 0	R/W	STC Base

- STSTC1R

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	STCB22 to STCB0	All 0	R/W	STC Base
8 to 0	STCX8 to STCX0	All 0	R/W	STC Extension

Note: If PCR_PID arrives during data read, the read value is overwritten and becomes undefined. Therefore, confirm that there is no PCR_PID during data read with the PCRF bit in STSTR. Specifically, set the PCRF bit to 0 and then start reading the receive data. Whenever PCRF is 1, take this procedure.

1.3.13 STIF Lock Control Register (STLKCR)

STLKCR is a 32-bit register to control PLL frequency lock. STLKCR is initialized to H'00000000 by a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25	LKWP	0	R/W	Setting this bit to 1 causes the LKCNT value to be reflected in the internal LKCNT. If this operation conflicts with the count up or clear operation of the internal LKCNT, writing by LKWP takes precedence. This bit is automatically cleared to 0.
24	ULWP	0	R/W	Setting this bit to 1 causes the ULCNT value to be reflected in the internal ULCNT. If this operation conflicts with the count up or clear operation of the internal ULCNT, writing by ULWP takes precedence. This bit is automatically cleared to 0.
23	ULCNT3	0	R/W	Setting ULWP to 1 causes the ULCNT value to be written to the internal ULCNT. When read, these bits indicate the state below. - The count of continuous LKZF = 1 states (outside the threshold value range) in the PLL lock state (LKF = 1) These bits are cleared to 0 when (1) ULCNT >= ULREF (when LKF = 1, it is cleared to 0), (2) the ULCNT value falls within the threshold value range (LKZF = 0), or (3) "discont" occurs.
22	ULCNT2	0	R/W	
21	ULCNT1	0	R/W	
20	ULCNT0	0	R/W	
19	LKCNT3	0	R/W	Setting LKLP to 1 causes the LKCNT value to be written to the internal LKCNT. When read, these bits indicate the state below. - The count of continuous LKZF = 0 states (within the threshold value range) in the PLL unlock state (LKF = 0) These bits are cleared to 0 when (1) LKCNT >= LKREF (when LKF = 0, it is set to 1) , (2) the LKCNT value exceeds the threshold value range (LKZF = 1), or (3) "discont" occurs.
18	LKCNT2	0	R/W	
17	LKCNT1	0	R/W	
16	LKCNT0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
15	GAIN3	0	R/W	These bits are used to control the right-shift amount that gains the error amount to be input to the adder from selector 1. Since the error amount is expressed as a two's complement, an arithmetic shift is used for right shift. That is, the most significant sign bit is copied to the bits that become short by the right shift for refill. Select a value between 0 and 10 for the right-shift amount of error amount. If a value outside the range is set, the setting is invalid and the operation is not guaranteed.
14	GAIN2	0	R/W	
13	GAIN1	0	R/W	
12	GAIN0	0	R/W	

Right-shift amount	Right-shift amount
0000: 0	1000: 8
0001: 1	1001: 9
0010: 2	1010: 10
0011: 3	1011: 11 (invalid)
0100: 4	1100: 12 (invalid)
0101: 5	1101: 13 (invalid)
0110: 6	1110: 14 (invalid)
0111: 7	1111: 15 (invalid)

11	LKCYC3	0	R/W	These bits set a PLL lock threshold value. For PLL lock threshold values, see table 1.4. Set an LKCYC value that is not larger than PWMCYC (LKCYC =< PWMCYC). If a value larger than PWMCYC is set, the operation is not guaranteed.
10	LKCYC2	0	R/W	
9	LKCYC1	0	R/W	
8	LKCYC0	0	R/W	
7	ULREF3	0	R/W	These bits specify a reference value for the number of continuous LKZF = 1 states (outside the threshold value range) when PLL is locked (LKF = 1). This value is compared with the ULCNT value. When ULCNT >= ULREF, the LKF bit in STSTR is set to 0.
6	ULREF2	0	R/W	
5	ULREF1	0	R/W	
4	ULREF0	0	R/W	
3	LKREF3	0	R/W	These bits specify a reference value for the number of continuous LKZF = 0 states (within the threshold value range) when PLL is unlocked (LKF = 0). This value is compared with the LKCNT value. When LKCNT >= LKREF, the LKF bit in STSTR is set to 1.
2	LKREF2	0	R/W	
1	LKREF1	0	R/W	
0	LKREF0	0	R/W	

Table 1.4 PLL Lock Threshold Value

Bits 11 to 8		Description		
LKCYC3 to LKCYC0	PLL Lock Threshold Value (× PWM Reference Clock)	Acceptable Comparison Bit Count n	Acceptable Comparison (Internal STC - Internal PCR) Result Range*1	PWM Control Variable*2 (ILGL = 1)
0000	2	0	—	—
0001	4	1	–1 to +1	–2
0010	8	2	–3 to +3	–4
0011	16	3	–7 to +7	–8
0100	32	4	–15 to +15	–16
0101	64	5	–31 to +31	–32
0110	128	6	–63 to +63	–64
0111	256	7	–127 to +127	–128
1000	512	8	–255 to +255	–256
1001	1024	9	–511 to +511	–512
1010	2048	10	–1023 to +1023	–1024
1011	4096	11	–2047 to +2047	–2048
1100	8192	12	–4095 to +4095	–4096
1101	16384	13	–8191 to +8191	–8192
1110	32768	14	–16383 to +16383	–16384
1111	65536	15	–32767 to +32767	–32768

Notes: 1. When the comparison (internal STC - internal PCR) result falls within the acceptable comparison result range, the LKZF bit is set to 0.

2. If the PWM control variable is -(2^n), it is treated as an invalid PWM control variable (ILGL = 1) and the LKZF bit is set to 1. The PWM control variable is selected by the PWMSEL and PWMSEL2 bits.

1.3.14 STIF Debugging Status Register (STDBGR)

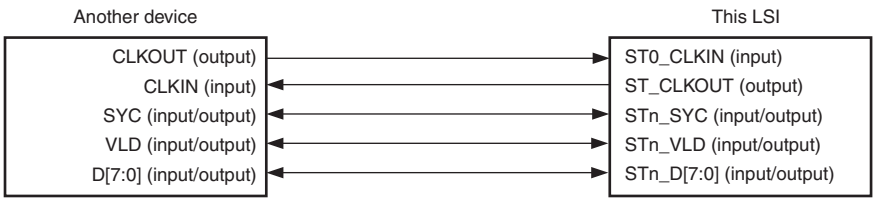
STDBGR is a 32-bit register that indicates the first four bytes of an input or output packet. STDBGR is provided for debugging. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	STMON31 to STMON0	All 0	R	The 4-byte timestamp of a packet that is input or output in TS mode is stored. This means that when the STIF receives a TS packet (188 bytes) from an external device, the 4-byte timestamp that will be added to the beginning of the TS packet by the STIF is stored. When the STIF outputs a TS packet to an external device, the 4-byte timestamp that was added to the beginning of the TTS packet (192 bytes) is stored. This register cannot be used in TTS mode or PS mode.

1.4 Examples of Clock Connection to Another Device

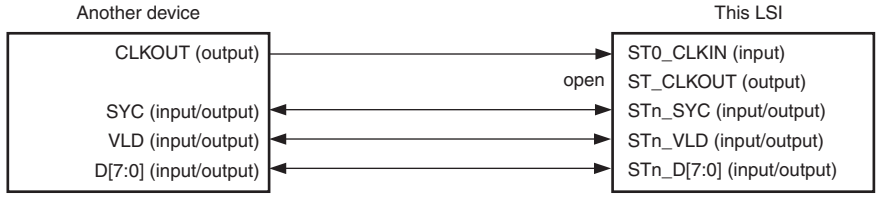
Examples of clock connection to another device are illustrated below.

1.4.1 A Basic Example



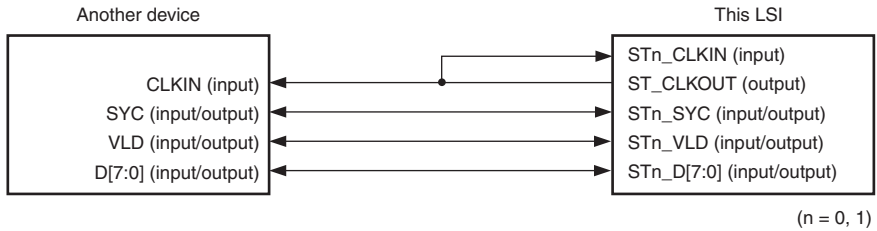
- When this LSI receives a stream, it is received in synchronization with STn_CLKIN.
- When this LSI sends a stream, it is sent in synchronization with ST_CLKOUT.

1.4.2 An Example of Clock Connection when Another Device Has No Clock Input



- When this LSI receives a stream, it is received in synchronization with STn_CLKIN.
- When this LSI sends a stream, it is sent in synchronization with STn_CLKIN.

1.4.3 An Example of Clock Connection when Another Device Has No Clock Output



- When this LSI receives a stream, it is received in synchronization with STn_CLKIN.
- When this LSI sends a stream, it is sent in synchronization with ST_CLKOUT.

1.5 Input/Output Timing

Figures 1.2 to 1.7 show the operation overview and input/output timing of each mode.

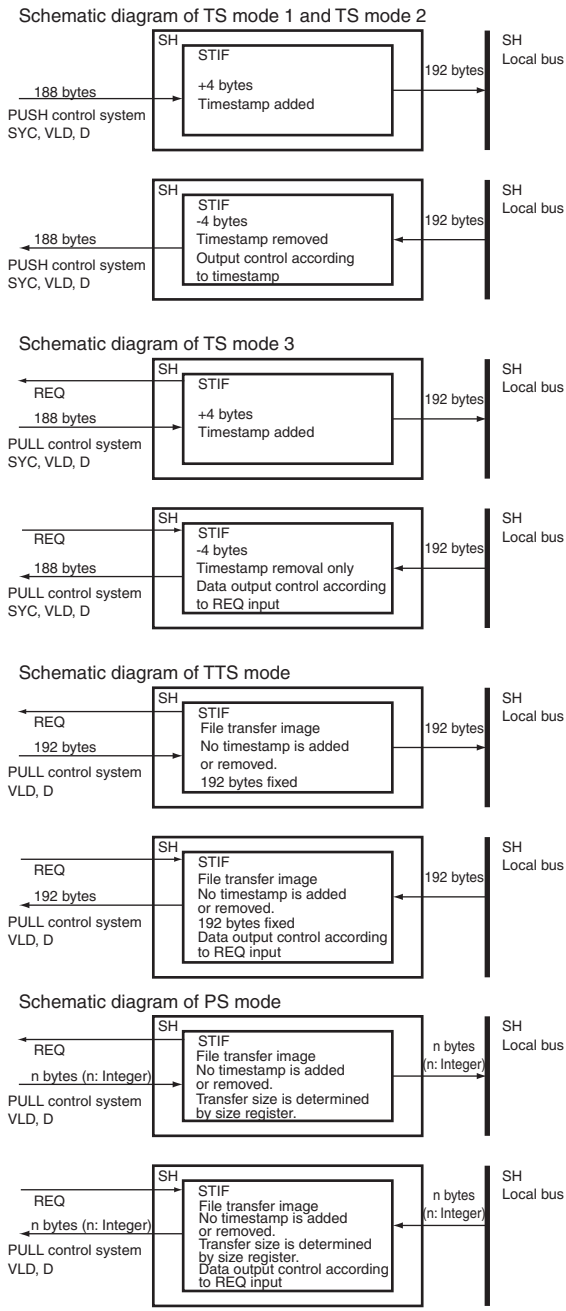


Figure 1.2 Operation Overview

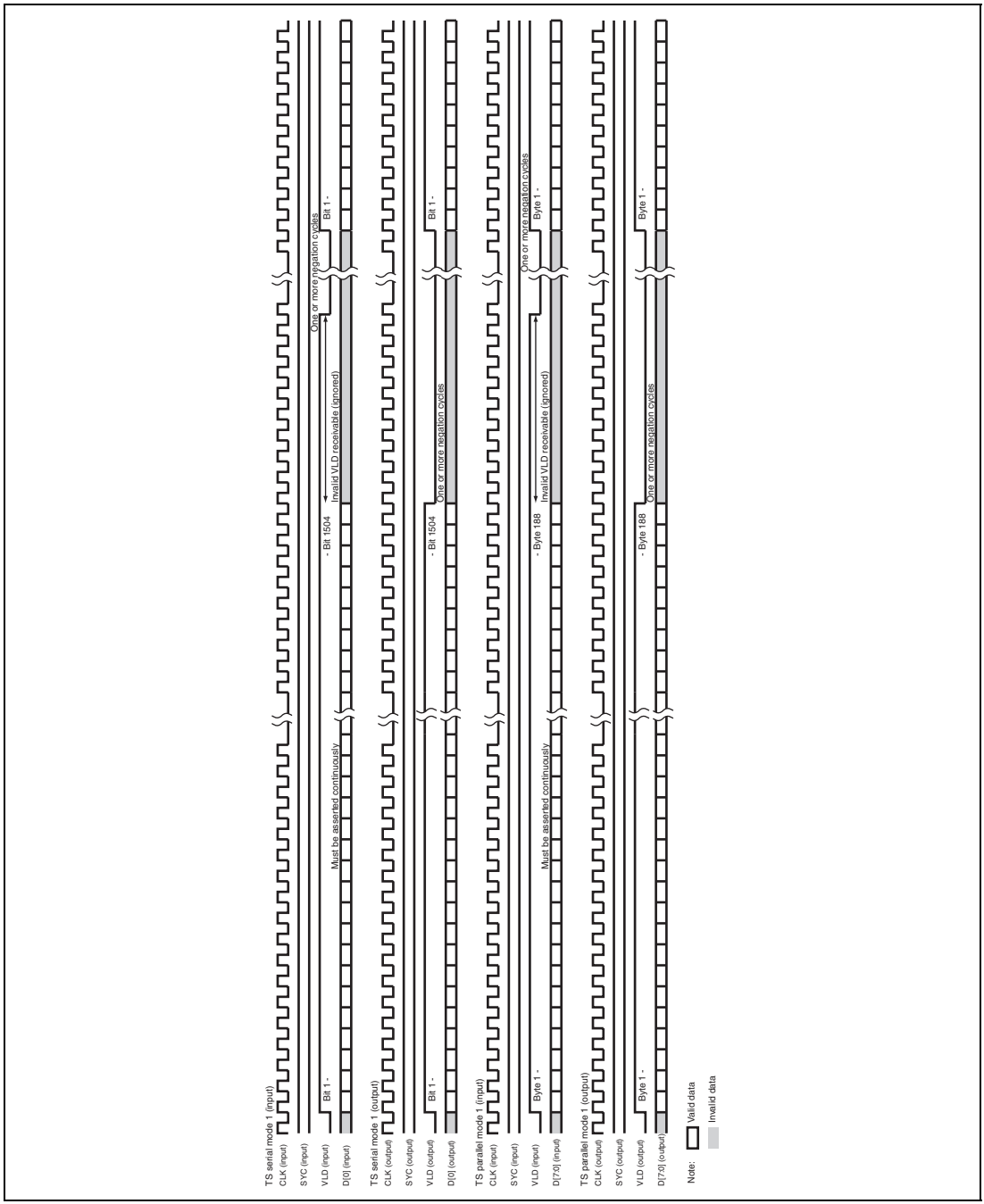


Figure 1.3 TS Mode 1

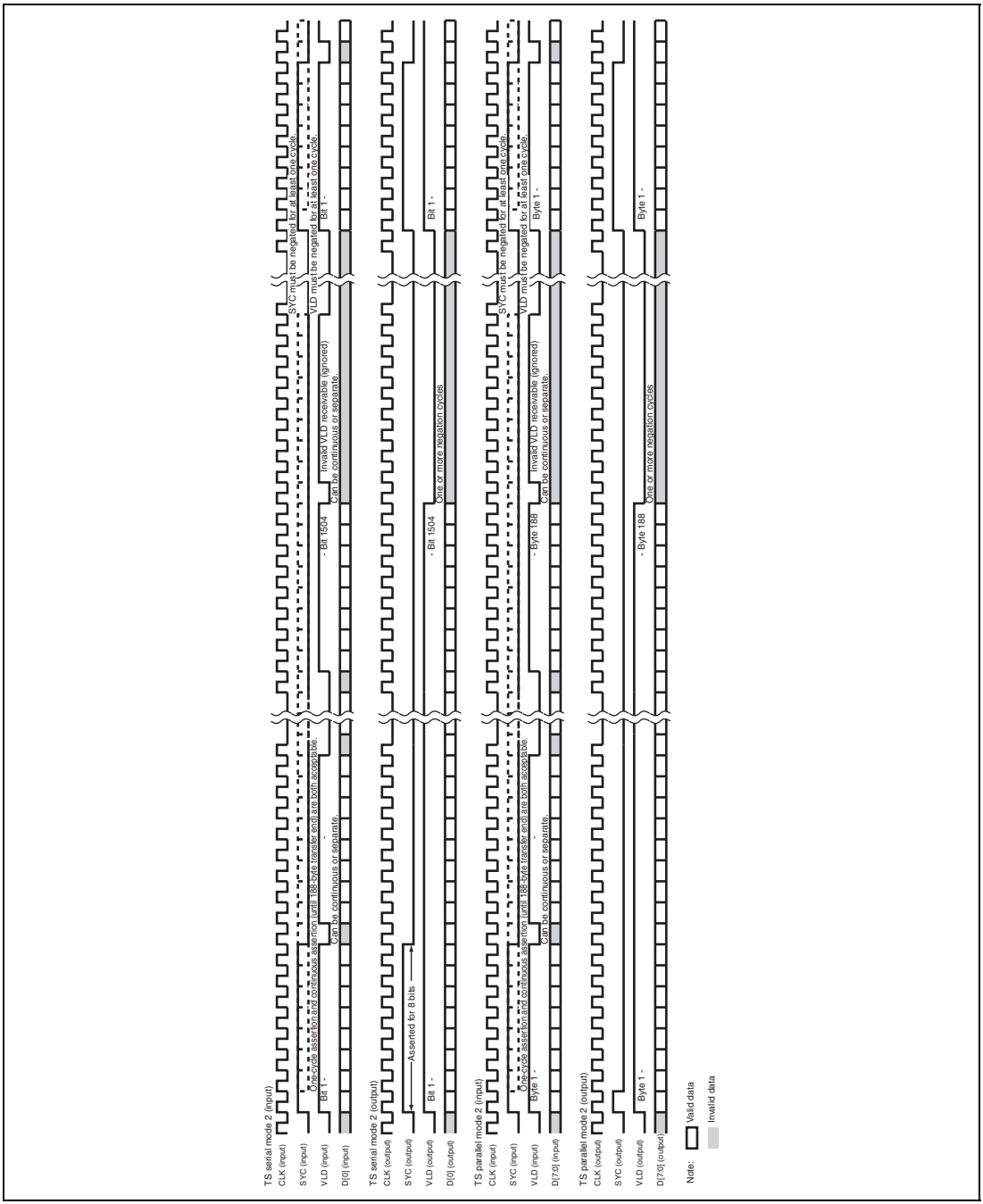


Figure 1.4 TS Mode 2

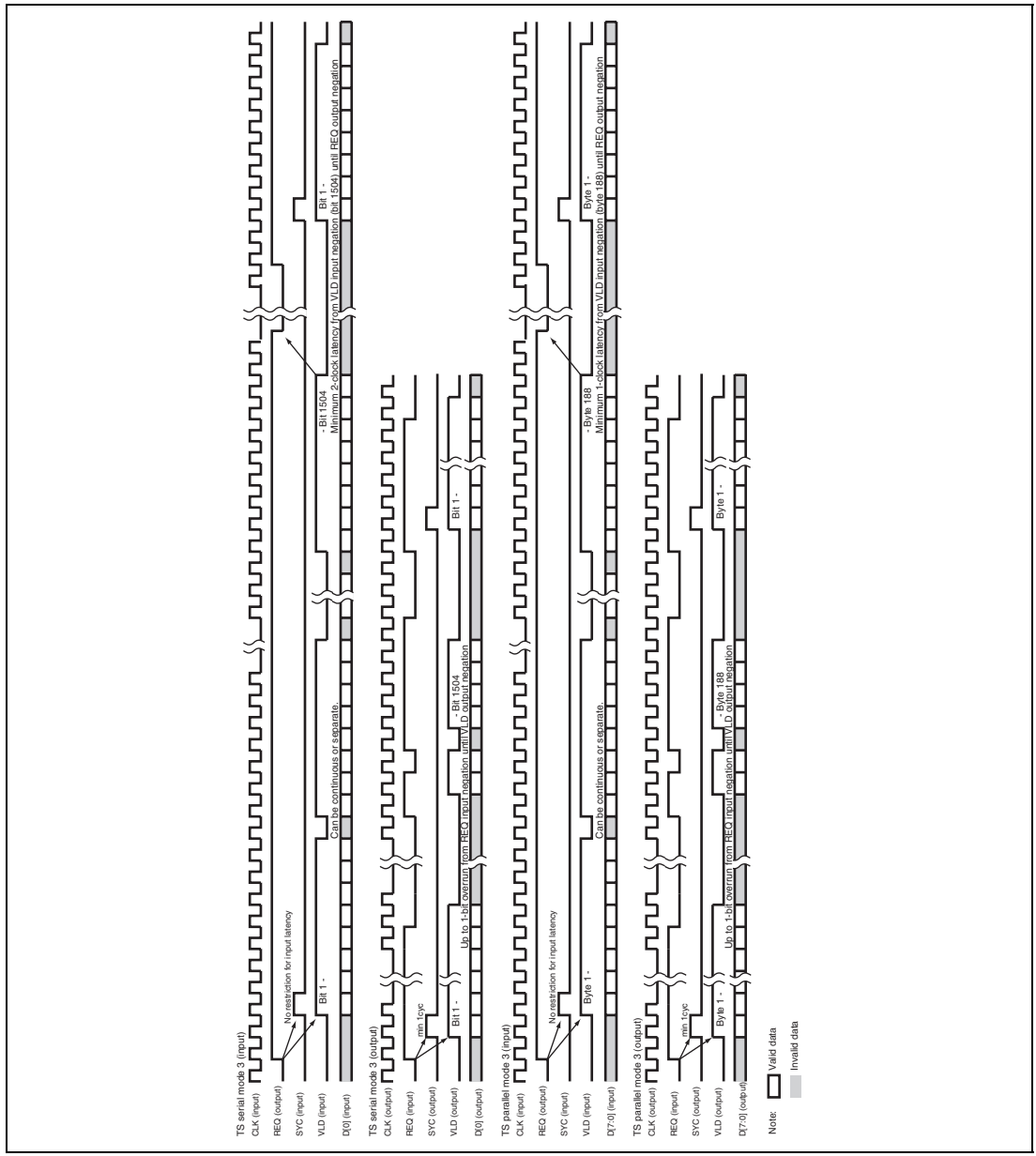


Figure 1.5 TS Mode 3

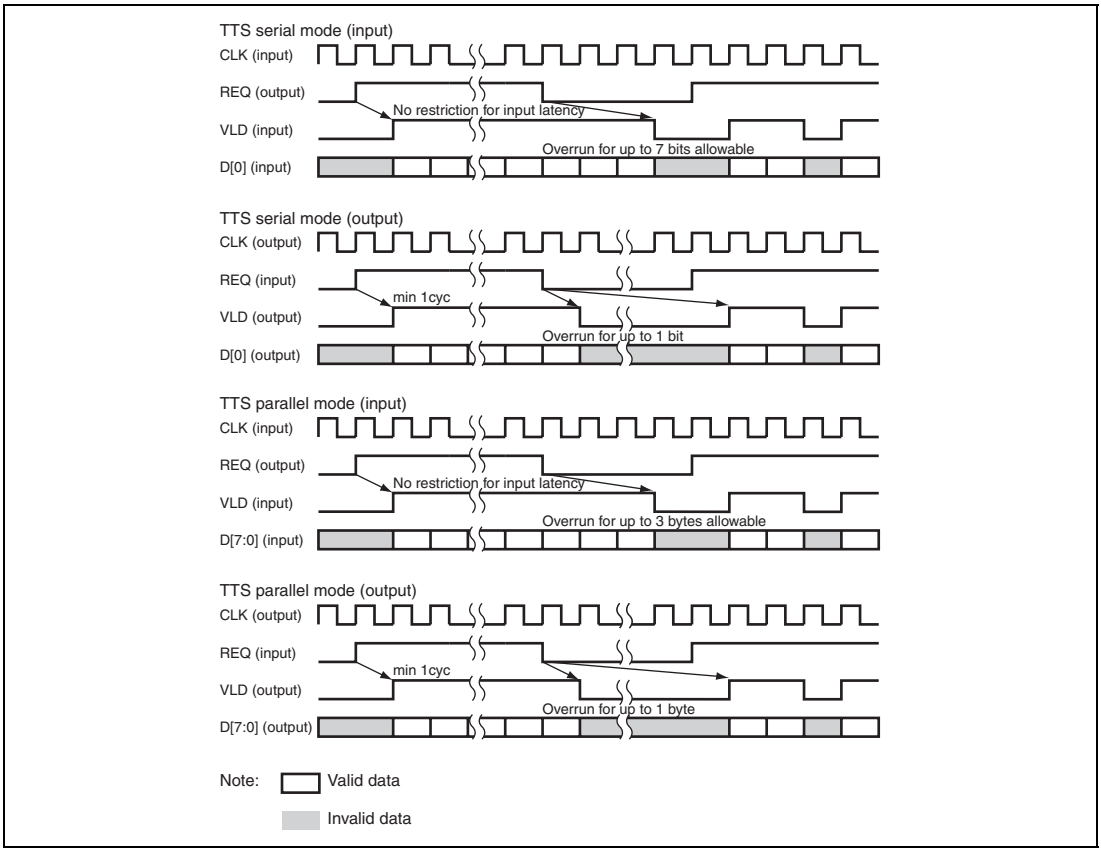


Figure 1.6 TTS Mode

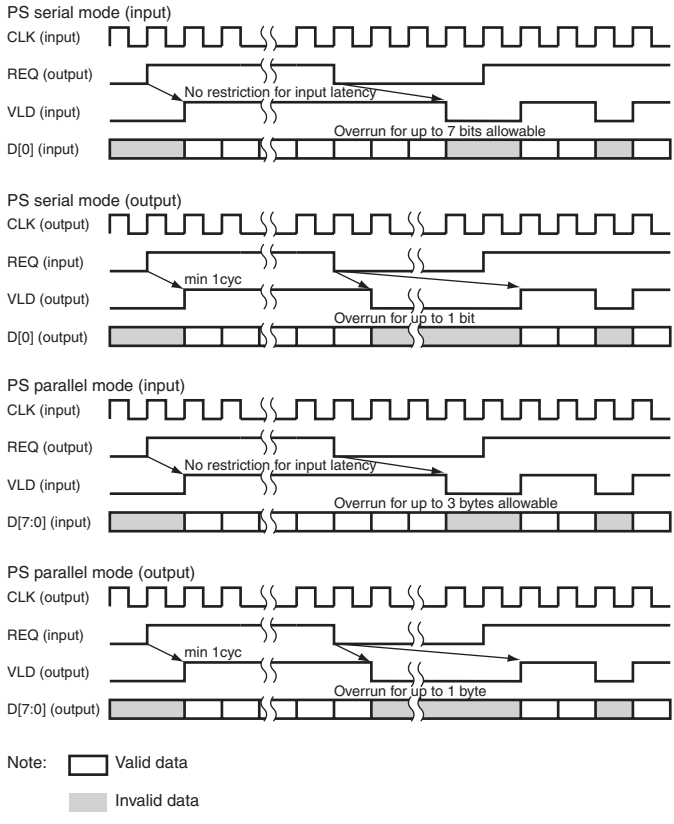


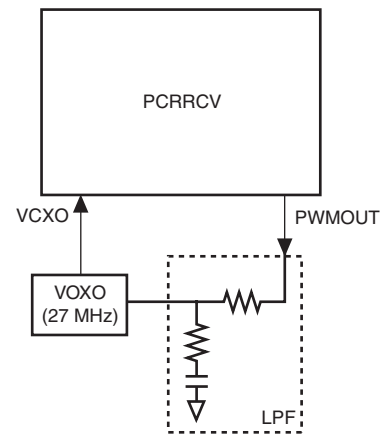
Figure 1.7 PS Mode

1.6 PCR Clock Recovery Module (PCRRCV)

The PCR clock recovery module (PCRRCV) is a circuit to provide a PWM (pulse wave modulation) output for controlling the external VCXO circuit according to the difference between 42-bit program_clock_reference (PCR) and system reference clock (STC). The PCR consists of program_clock_reference_base (PCR base) and program_clock_reference_extension (PCR extension) of adaptation_field in an input transport (TS) packet.

The PCRRCV has the following features.

- A 42-bit internal STC counter triggered by the clock input from the external VCXO circuit
- PWM output for controlling the external VCXO circuit can be output on the STn_PWMOUT pin.
- VCXO control is selectable from PWM control mode according to the difference between PCR and STC in a TS packet or PWM control mode by directly setting STPWMR.
- PWM control accuracy and PWM cycle can be set in STPWMMR.



[Legend]

VCXO:	VCXO clock input pin
LPF:	Low Pass Filter
PWMOUT:	PWM control output pin
VOXO (27 MHz):	Voltage control oscillator

Figure 1.8 STn_VCO_CLKIN and STn_PWMOUT Connections

1.6.1 Operation of PCR Clock Recovery

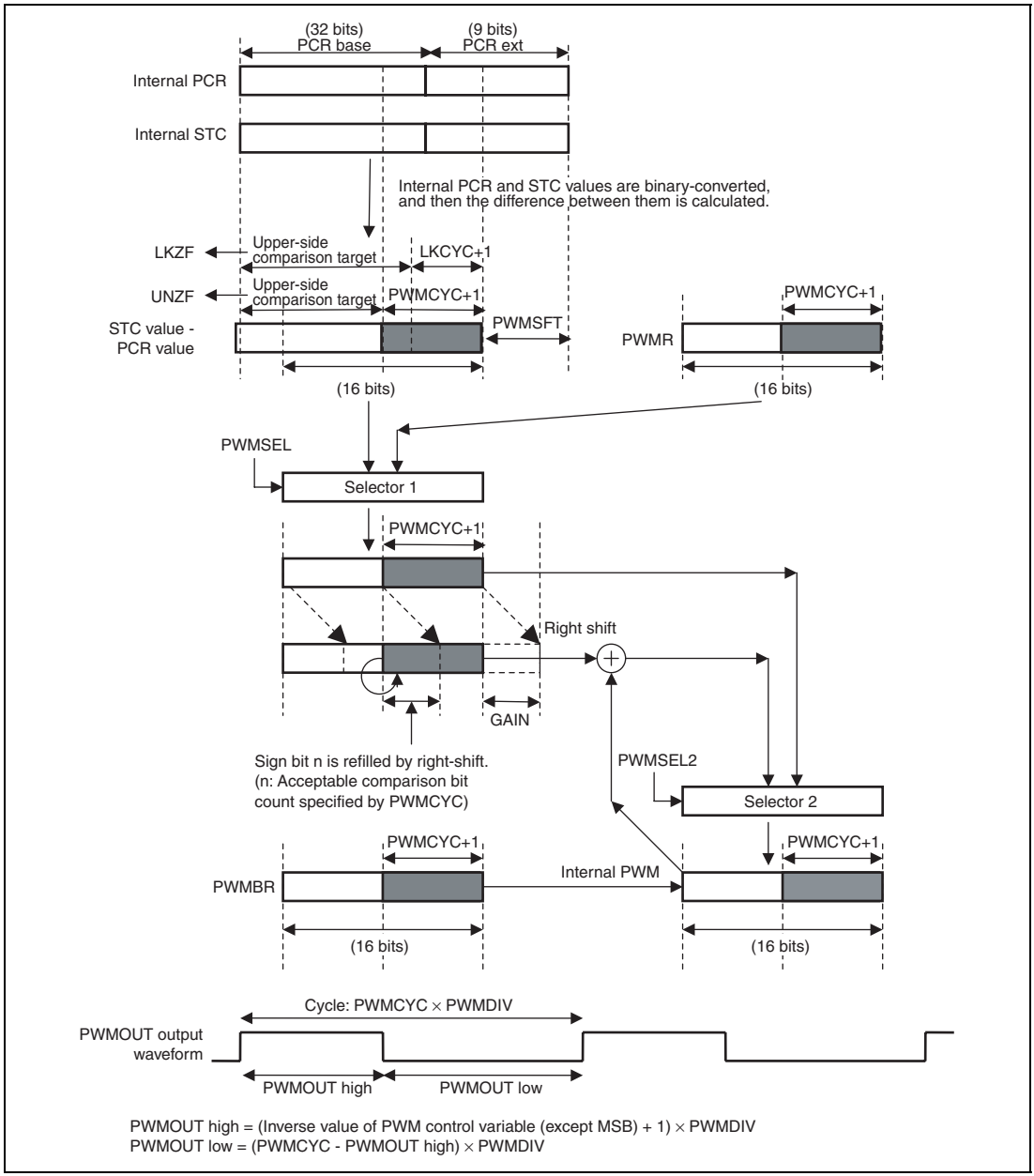


Figure 1.9 Illustration of Register Settings

- Register Settings
  A. Specify the acceptable comparison bit count n of the PWM control variable using the PWMCYC3 to PWMCYC0 bits in STPWMMR.
  B. Set a PLL lock threshold value using the LKCYC3 to LKCYC0 bits in STLKCR.
  C. Specify the shift amount of the PWM control variable reference bit (LSB of PWM control variable) using the PWMSFT3 to PWMSFT0 bits in STPWMMR. The shifted bits (PWMSFT width bits in figure 1.9) do not fall within the PWM control variable comparison range.
  D. In the PWM control variable calculation, the PCR base and PCR extension of the internal STC and PCR registers are converted to binary values, and the converted values are masked with the PWMSFT width bits, and then the difference between the values is calculated. The binary conversion is performed using the following expression (1).

     Internal STC/PCR binary conversion: (PCR_base × 300 + PCR_ext) & (PWMSFT width mask) …(1)

     The upper comparison result except (PWMCYC + 1) of the difference result is reflected in the UNZF bit in STSTR.

     The upper comparison result except (LKCYC + 1) of the difference result is reflected in the LKZF bit in STSTR.
  E. Specify the right-shift amount of selector 1 output using the GAIN3 to GAIN0 bits in STLKCR. Since an arithmetic shift is used for the right shift, the overflowing bits are discarded on the lower side and the sign bit (bit n when the acceptable comparison bit count = n) is refilled on the upper side.
  F. Switch selector 1 and selector 2 using the PWMSEL and PWMSEL2 bits in STPWMMR to select the path to the internal PWM register.
  G. Specify the PWM reference clock of the PWMOUT output pin using the PWMDIV3 to PWMDIV0 bits in STPWMMR. The PWM reference clock has a cycle of system clock × PWMDIV. The PWM cycle of the PWMOUT output pin has a clock cycle of PWMCYC × PWMDIV.
  H. The PWMOUT output waveform that depends on the PWM control variable is as follows:

     PWMOUT high =
     (Inverse value of PWM control variable (except MSB) + 1) × PWMDIV …(2a)

     PWMOUT low = (PWMCYC - PWMOUT low) × PWMDIV …(2b)

     When the acceptable comparison bit count is n, the MSB becomes bit n of the PWM control variable.

     The PWM control variable is expressed as a two's complement. Whether to include the upper comparison target in the PWM control range is specified by the PWMUEN bit in STPWMMR.

I. The PWM control variable can directly be set by the STPWMR register when the PWMSEL bit in STPWMMR is set to 1. In this case, PWMCYC acts as a number of acceptable comparison bits as in the case of internal STC register value - PCR register value (figure 1.9).

1.6.2 PCR Clock Recovery Operation

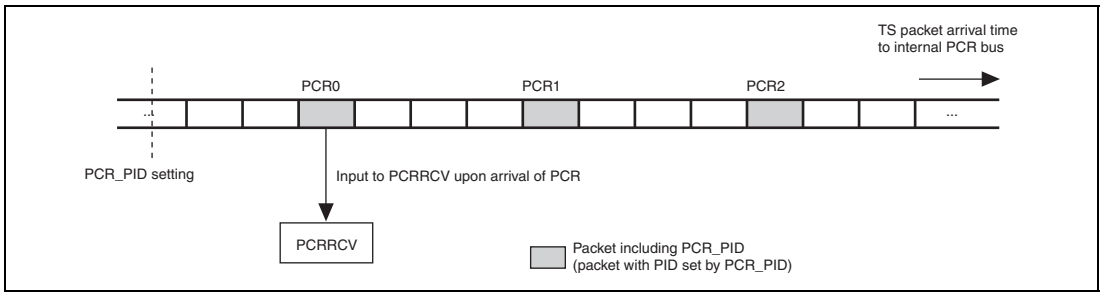


Figure 1.10 Overview of TS Packet

- Example 1: Clock recovery using the PCRRCV hardware
  A. Set PWMSEL to 0 to enable the clock recovery using the PCRRCV hardware.
  B. Set the PWMCYC, PWMSFT, PWMDIV, and PWMUEN bits in STPWMMR for the PWM control output.
  C. Set the PID bits in STPWMMR to PCR_PID of a packet including PCR for recovering clock to the PCRRCV. The clock recovery starts upon the PID setting. For this reason, if the PCR continuity is impaired by a reset or channel change, set PIDEN to 0 and then set a PID.
  D. When a packet including PCR_PID arrives, the PCRRCV extracts the 42-bit program_clock_reference (PCR) from the adaptation_field of the packet. For packet conditions for extracting PCR, see table 1.5. When a TS packet that satisfies the conditions in table 1.5 arrives, a PCR arrival pulse is generated in the PCRRCV.
  E. When a PCR arrival pulse is generated, the PCR extracted from the packet is transferred to the internal PCR register and STC counter. The STC counter (STC) value is also transferred to the internal STC register at the same time. After that, the STC counter starts counting by the VCXO clock input. The STC counter value incremented from the previous PCR until the present time by the VCXO clock is set in the internal STC register, and the currently arrived PCR is set in the internal PCR register and STC counter.
  F. The PWM control variable, which is the comparison result between the internal STC and PCR registers (STC value - PCR value), is calculated by the PWMCYC, PWMSFT, and PWMDIV bits in STPWMMR. When the comparison result reflection conditions in table

1.6 are satisfied, the comparison result is reflected in the PWM output control. Setting PWMUEN to 1 disables the PWM control variable outside the acceptable comparison result range specified by the PWMCYC bits to be reflected in the PWM control output. This function can prevent an abnormal PCR (caused by PCR error due to transmission error or protocol violation at the transmitter side) from being reflected in clock recovery. For measures against abnormal PCRs, see the procedure examples 1 and 2 described later.

G. The PWMOUT waveform with a reflected PWM control variable is output from the PWMOUT pin as shown in figure 1.9.

H. The internal STC and PCR register values are compared each time a PCR_PID packet arrives. By configuring a feedback circuit including an external low-pass filter (LPF) and an external VCXO, which makes the comparison (STC value - PCR value) result to be 0, the VCXO clock frequency is adjusted.

- Example 2: Clock recovery using the PCRRCV and software

A. Set PWMSEL to 1 to disable the clock recovery using the PCRRCV hardware. Also set PCRE to 1 to enable interrupt requests made by each PCR arrival pulse.

B. The PCRRCV settings and transfers of the PCRRCV internal registers are the same as those described in steps B to E of Example 1. Notes 1 to 4 for table 1.6 apply to the case of the first arrival of PCR after the PCR continuity is lost. Since the STC counter that has no continuity with the arrived internal PCR register is transferred to the internal STC register, it is not appropriate to calculate the difference (STC value - PCR value) by the CPU.

C. Confirm that transfer between the PCRRCV internal registers has been completed (PCRF = 1), and then set PCRF to 0. After that, set the STCRS and PCRRS bits to 1 to enable transfer from the internal STC register to STSTC0R and STSTC1R registers, as well as transfer from the internal PCR register to STPCR0R and STPCR1R registers. Then read STSTC0R/STSTC1R and STPCR0R/STPCR1R. Furthermore, to obtain the STC counter value for setting to the MPEG2 decoder, set the STCXP bit to 1 to enable transfer from the STC counter to STSTC0R and STSTC1R registers. Then read STSTC0R and STSTC1R.

D. A PCRF read value of 0 means that no PCR_PID packet arrived during the register read stage in step C. This shows that the reading of STSTC0R/STSTC1R and STPCR0R/STPCR1R in step C was successful. On the other hand, a PCRF read value of 1 shows an arrival of PCR_PID packet during the register read stage in step C. It is not certain that the read STSTC0R/STSTC1R and STPCR0R/STPCR1R are the values of the internal STC and PCR registers that arrived previously or those that arrived during the register read stage. Therefore, go back to step C and repeat the procedure.

E. The CPU converts the read STSTC0R/STSTC1R and STPCR0R/STPCR1R values to binary ones with STCbin and PCRbin respectively using the expression (1) in section 1.6.1, Operation of PCR Clock Recovery, step D, and then calculates the difference (STCbin - PCRbin). To be set in STPWMR as a PWM control variable, specify the difference (STCbin - PCRbin) as a two's complement of the acceptable comparison bit count n (bit n: sign bit) specified by the PWMCYC bits. Set a value within the range of -(2^n - 1) to +(2^n - 1) for the difference. Do not set the value -2^n. The CPU also determines the handling of PCR data errors shown in step F of Example 1.

F. The PWM control variable that is set in STPWMR can be reflected in the PWM control output by writing 1 to PWMWP.

G. For the principle of VCXO clock frequency adjustment using the PWM control output, the descriptions in steps G and H of Example 1 apply.

Table 1.5 PCR Extraction Conditions

transport_error_ indicator	adaptation_field_ control	adaptation_field_length	PCR_flag	PCR Extraction*
0	00	don't care	don't care	Impossible
	01	don't care	don't care	Impossible
	10	0 ≤ len < 7	don't care	Impossible
		7 ≤ len < H'FF	0	Impossible
			1	Possible
	11	0 ≤ len < 7	don't care	Impossible
		7 ≤ len < H'FF	0	Impossible
			1	Possible
1	don't care	don't care	don't care	Impossible

Note: * When PCR extraction is possible, PCR is extracted and a PCR arrival pulse is generated.

Table 1.6 Internal PCR and STC Registers Comparison Result Reflection Conditions *[1]

Reflection	Comparison Result Reflection Conditions
Not reflected*[2]	Arrival of PCR_PID after "discont" *[3]
	Arrival of PCR_PID whose upper comparison result does not match when PWMUEN = 1
Reflected	Arrival of PCR_PID in other cases

Notes: 1. When PWMSEL = 0, the reflection conditions in this table are effective. When PWMSEL = 1, the PWM control variable can be reflected in the PWM control output by writing 1 to the PWMWP bit.
2. Since the PWM control variable is not reflected, the PWMOUT output waveform is maintained.
3. There are four patterns of PCR_PID arrival after "discont."
(1) The first PCR_PID arrival after reset cancellation
(2) The first PCR_PID arrival after PCR flush cancellation
(3) When discontinuity_indicator of the arrived PCR_PID packet = 1
(4) Arrival of PCR_PID after the PID bits in the STPWMMR register were modified *[4]
4. Set PIDEN to 0 before modifying the PID bits in the STPWMMR register. With this setting, arrival of PCR_PID is treated as arrival after "discont."

- Hardware measures against arrival of abnormal PCRs

  Initial setting (1) Set an LKCYC value not larger than the PWMCYC value. The LKCYC value must be larger than the steady-state deviation (error amount) of the PLL that is configured with an external circuit. Determine the LKCYC value with a margin from the PLL's steady-state deviation. Otherwise, the operation of this LSI is not guaranteed.

  Initial setting (2) Set an LKREF value until the LKF bit is set to 1 when the PLL error amount falls within the threshold value range. The LKREF bits, which are usually set to 1 or larger, are provided for setting to enhance the stability against arrival of abnormal PCRs in the PLL pull-in state. Note that the larger the LKREF value becomes, the higher stability is obtained, but the PLL pull-in time becomes longer.

  Initial setting (3) Set a ULREF value until the LKF bit is set to 0 when the PLL error amount exceeds the threshold value limit. The LKREF bits, which are usually set to 1 or larger, are provided for setting to enhance the stability against arrival of abnormal PCRs in the PLL lock state. It is recommended that the LKREF value be larger than the maximum of the number of continuous arrivals of system-dependent abnormal PCRs.

1.7 Usage Notes

1. When a stream from an external device to the STIF is being transferred to memory via the A-DMAC, perform the following processing in the order shown below to pause and resume transfer.

   To pause transfer:

   (1) Clear the C[i]C_E bit in the channel [i] processing control register (C[i]C) to 0.

   (2) Poll the interrupt request flag or wait for an interrupt until the C[i]I_DI bit in the channel [i] processing interrupt request register (C[i]I) becomes 1.

   To resume transfer:

   (3) Set the C[i]C_E bit in the channel [i] processing control register (C[i]C) to 1.

2. When a stream from memory to the A-DMAC is being transferred to an external device via the STIF, perform the following processing in the order shown below to pause and resume transfer.

   To pause transfer:

   (1) Clear the C[i]C_E bit in the channel [i] processing control register (C[i]C) to 0.

   (2) Poll the interrupt request flag or wait for an interrupt until the C[i]I_DI bit in the channel [i] processing interrupt request register (C[i]I) becomes 1.

   To resume transfer:

   (3) Set the C[i]C_E bit in the channel [i] processing control register (C[i]C) to 1.

3. When a stream from an external device to the STIF is being transferred to memory via the A-DMAC, perform the following processing in the order shown below to forcibly terminate and resume transfer.

   To forcibly terminate transfer:

   (1) Clear both the REQEN and EN bits in the STIF control register (STCTLR) to 0.

   (2) Clear the C[i]C_E bit to 0 and set the C[i]C_R bit to 1 in the channel [i] processing control register (C[i]C).

   (3) Prepare a dummy descriptor. (Specify data transfer from the relevant STIF channel to memory and set a transfer size of 192 bytes. Other settings and the transfer data contents can be set as desired.)

   (4) Set the pointer address of the above dummy descriptor in the channel [i] processing descriptor start address register (C[i]DSA) and channel [i] processing descriptor current address register (C[i]DCA).

   (5) Set the C[i]C_E bit to 1 and clear the C[i]C_R bit to 0 in the channel [i] processing control register (C[i]C).

(6) According to the STIF channel to be stopped, poll the interrupt request flag until bits 14 to 8 or 6 to 0 in address H'FF88 05E0 become 0000000. (If STIF0 is to be stopped, bits 14 to 8 are polled. If STIF1 is to be stopped, bits 6 to 0 are polled.)

(7) Clear the C[i]C_E bit to 0 and set the C[i]C_R bit to 1 in the channel [i] processing control register (C[i]C).

(8) Set the SRST bit in the STIF control register (STCTLR) to 1.

To resume transfer:

(9) Set the C[i]C_E bit to 1 and clear the C[i]C_R bit to 0 in the channel [i] processing control register (C[i]C).

(10) Set both the REQEN and EN bits in the STIF control register (STCTLR) to 1.

4. When a stream from memory to the A-DMAC is being transferred to an external device via the STIF, perform the following processing in the order shown below to forcibly terminate and resume transfer.

To forcibly terminate transfer:

(1) Clear both the REQEN and EN bits in the STIF control register (STCTLR) to 0.

(2) Clear the C[i]C_E bit to 0 and set the C[i]C_R bit to 1 in the channel [i] processing control register (C[i]C).

(3) Prepare a dummy descriptor. (Specify data transfer from the relevant STIF channel to memory and set a transfer size of 192 bytes. Other settings and the transfer data contents can be set as desired.)

(4) Set the pointer address of the above dummy descriptor in the channel [i] processing descriptor start address register (C[i]DSA) and channel [i] processing descriptor current address register (C[i]DCA).

(5) Set the C[i]C_E bit to 1 and clear the C[i]C_R bit to 0 in the channel [i] processing control register (C[i]C).

(6) According to the STIF channel to be stopped, poll the interrupt request flag until bits 14 to 8 or 6 to 0 in address H'FF88 05E0 become 0000000. (If STIF0 is to be stopped, bits 14 to 8 are polled. If STIF1 is to be stopped, bits 6 to 0 are polled.)

(7) Clear the C[i]C_E bit to 0 and set the C[i]C_R bit to 1 in the channel [i] processing control register (C[i]C).

(8) Set the SRST bit in the STIF control register (STCTLR) to 1.

To resume transfer:

(9) Set both the REQEN and EN bits in the STIF control register (STCTLR) to 1.

(10) Set the C[i]C_E bit to 1 and clear the C[i]C_R bit to 0 in the channel [i] processing control register (C[i]C).

Section 2 Stream Interface (STIF) Electrical Characteristics

2.1 STIF Module Signal Timing (1)

Table 2.1 STIF Module Signal Timing (1)

Conditions: V_{CC} = V_{CC} (PLL) = DV12 = UV12 = 1.1 to 1.3 V, V_{CC}Q = DV33 = 3.1 to 3.5 V,
AV12 = 1.1 to 1.3 V, AV33 = 3.1 to 3.5 V,
V_{SS} = V_{SS} (PLL) = DG12 = UG12 = V_{SS}Q = DG33 = AG12 = AG33 = 0 V,
Ta = –20 to 70°C (regular specifications),
–40 to 85°C (wide temperature specifications)

Item		Symbol	Min.	Max.	Unit	Figure
STn_CLKIN clock input cycle	Parallel mode	$t_{ST\_CKIN\_CYC}$	20.0	400	ns*	2.1
	Serial mode		13.2	400		
STn_CLKIN clock input high pulse width	Parallel mode	$t_{ST\_CKIN\_H}$	0.4	0.6	$t_{ST\_CKIN\_CYC}$	
	Serial mode		0.4	0.6		
STn_CLKIN clock input low pulse width	Parallel mode	$t_{ST\_CKIN\_L}$	0.4	0.6	$t_{ST\_CKIN\_CYC}$	
	Serial mode		0.4	0.6		
STn_CLKIN clock input rise time	Parallel mode	$t_{ST\_CKIN\_r}$	—	0.1	$t_{ST\_CKIN\_CYC}$	
	Serial mode		—	0.1		
STn_CLKIN clock input fall time	Parallel mode	$t_{ST\_CKIN\_f}$	—	0.1	$t_{ST\_CKIN\_CYC}$	
	Serial mode		—	0.1		

Note: * Furthermore, the period must be such that the frequency of STn_CLKIN (in MHz) is less than or equal to 80% of the frequency of clks1 (in MHz).

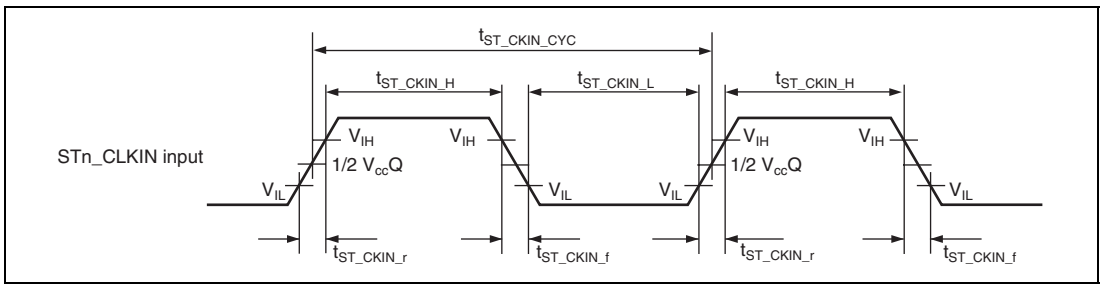


Figure 2.1 STIF Module Signal Timing (1)

2.1.1 STIF Module Signal Timing (2)

Table 2.2 STIF Module Signal Timing (2)

Conditions: V_{CC} = V_{CC} (PLL) = DV12 = UV12 = 1.1 to 1.3 V, $V_{CC}Q$ = DV33 = 3.1 to 3.5 V,
AV12 = 1.1 to 1.3 V, AV33 = 3.1 to 3.5 V,
V_{SS} = V_{SS} (PLL) = DG12 = UG12 = $V_{SS}Q$ = DG33 = AG12 = AG33 = 0 V,
Ta = –20 to 70°C (regular specifications),
–40 to 85°C (wide temperature specifications)

Item		Symbol	Min.	Max.	Unit	Figure
ST_CLKOUT clock output cycle	Parallel mode	$t_{ST\_CKOUT\_CYC}$	20.0	400	ns*	2.2
	Serial mode		13.2	400		
ST_CLKOUT clock output high pulse width	Parallel mode	$t_{ST\_CKOUT\_H}$	6.75	—	ns	
	Serial mode		3	—		
ST_CLKOUT clock output low pulse width	Parallel mode	$t_{ST\_CKOUT\_L}$	6.75	—	ns	
	Serial mode		3	—		
ST_CLKOUT clock output rise time	Parallel mode	$t_{ST\_CKOUT\_r}$	—	2.75	ns	
	Serial mode		—	2.75		
ST_CLKOUT clock output fall time	Parallel mode	$t_{ST\_CKOUT\_f}$	—	2.75	ns	
	Serial mode		—	2.75		

Note: * Furthermore, the period must be such that the frequency of ST_CLKOUT (in MHz) is less than or equal to 80% of the frequency of clks1 (in MHz).

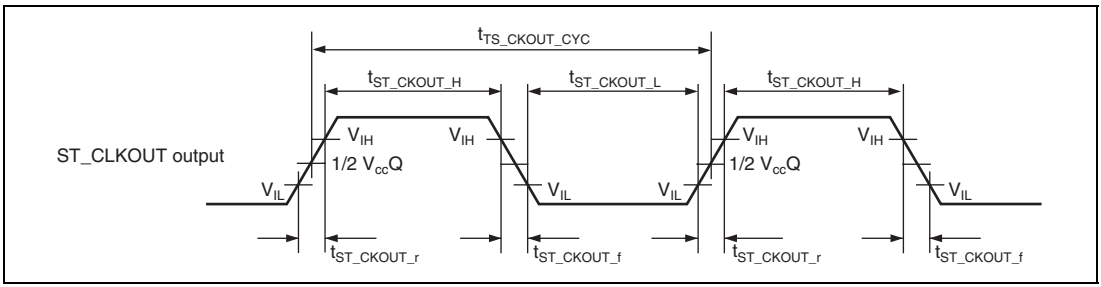


Figure 2.2 STIF Module Signal Timing (2)

2.1.2 STIF Module Signal Timing (3)
(With Stream Input/Output Set Synchronized with STn_CLKIN Rise Time)

Table 2.3 STIF Module Signal Timing (3)

Conditions: V_{CC} = V_{CC} (PLL) = DV12 = UV12 = 1.1 to 1.3 V, V_{CC}Q = DV33 = 3.1 to 3.5 V,
AV12 = 1.1 to 1.3 V, AV33 = 3.1 to 3.5 V,
V_{SS} = V_{SS} (PLL) = DG12 = UG12 = V_{SS}Q = DG33 = AG12 = AG33 = 0 V,
Ta = –20 to 70°C (regular specifications),
–40 to 85°C (wide temperature specifications)

Item	Symbol	Min.	Max.	Unit	Figure
STn_SYC output delay time 1	t_{STSD1}	0	11.5	ns	2.3
STn_VLD output delay time 1	t_{STVD1}	0	11.5	ns	
STn_REQ output delay time 1	t_{STRD1}	0	11.5	ns	
STn_Dm output delay time 1	t_{STDD1}	0	11.5	ns	
STn_SYC input setup time 1	t_{STSS1}	4	—	ns	
STn_SYC input hold time 1	t_{STSH1}	3	—	ns	
STn_VLD input setup time 1	t_{STVS1}	4	—	ns	
STn_VLD input hold time 1	t_{STVH1}	3	—	ns	
STn_REQ input setup time 1	t_{STRS1}	4	—	ns	
STn_REQ input hold time 1	t_{STRH1}	3	—	ns	
STn_Dm input setup time 1	t_{STDS1}	4	—	ns	
STn_Dm input hold time 1	t_{STDH1}	3	—	ns	

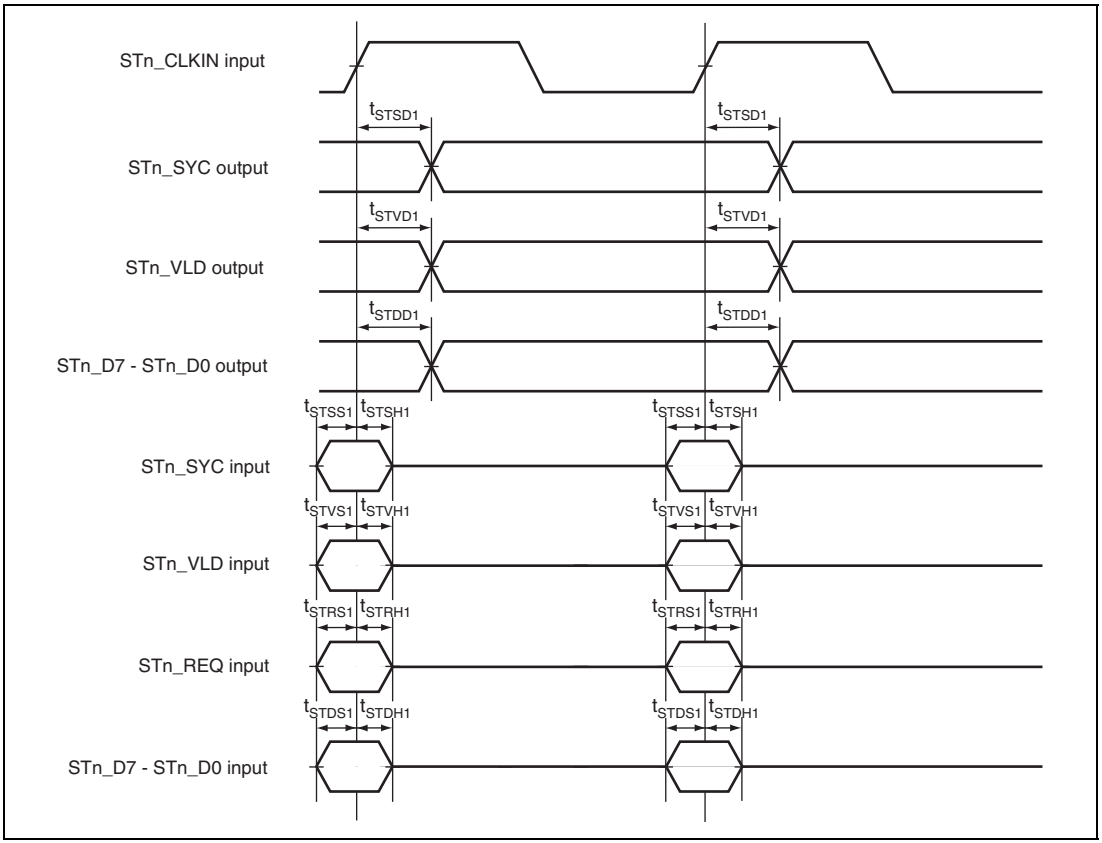


Figure 2.3 STIF Module Signal Timing (3)

2.1.3 STIF Module Signal Timing (4)
(With Stream Input/Output Set Synchronized with STn_CLKIN Fall Time)

Table 2.4 STIF Module Signal Timing (4)

Conditions: V_{CC} = V_{CC} (PLL) = DV12 = UV12 = 1.1 to 1.3 V, V_{CC}Q = DV33 = 3.1 to 3.5 V,
AV12 = 1.1 to 1.3 V, AV33 = 3.1 to 3.5 V,
V_{SS} = V_{SS} (PLL) = DG12 = UG12 = V_{SS}Q = DG33 = AG12 = AG33 = 0 V,
Ta = –20 to 70°C (regular specifications),
–40 to 85°C (wide temperature specifications)

Item	Symbol	Min.	Max.	Unit	Figure
STn_SYC output delay time 2	t_{STSD2}	0	11.5	ns	2.4
STn_VLD output delay time 2	t_{STVD2}	0	11.5	ns	
STn_REQ output delay time 2	t_{STRD2}	0	11.5	ns	
STn_Dm output delay time 2	t_{STDD2}	0	11.5	ns	
STn_SYC input setup time 2	t_{STSS2}	4	—	ns	
STn_SYC input hold time 2	t_{STSH2}	3	—	ns	
STn_VLD input setup time 2	t_{STVS2}	4	—	ns	
STn_VLD input hold time 2	t_{STVH2}	3	—	ns	
STn_REQ input setup time 2	t_{STRS2}	4	—	ns	
STn_REQ input hold time 2	t_{STRH2}	3	—	ns	
STn_Dm input setup time 2	t_{STDS2}	4	—	ns	
STn_Dm input hold time 2	t_{STDH2}	3	—	ns	

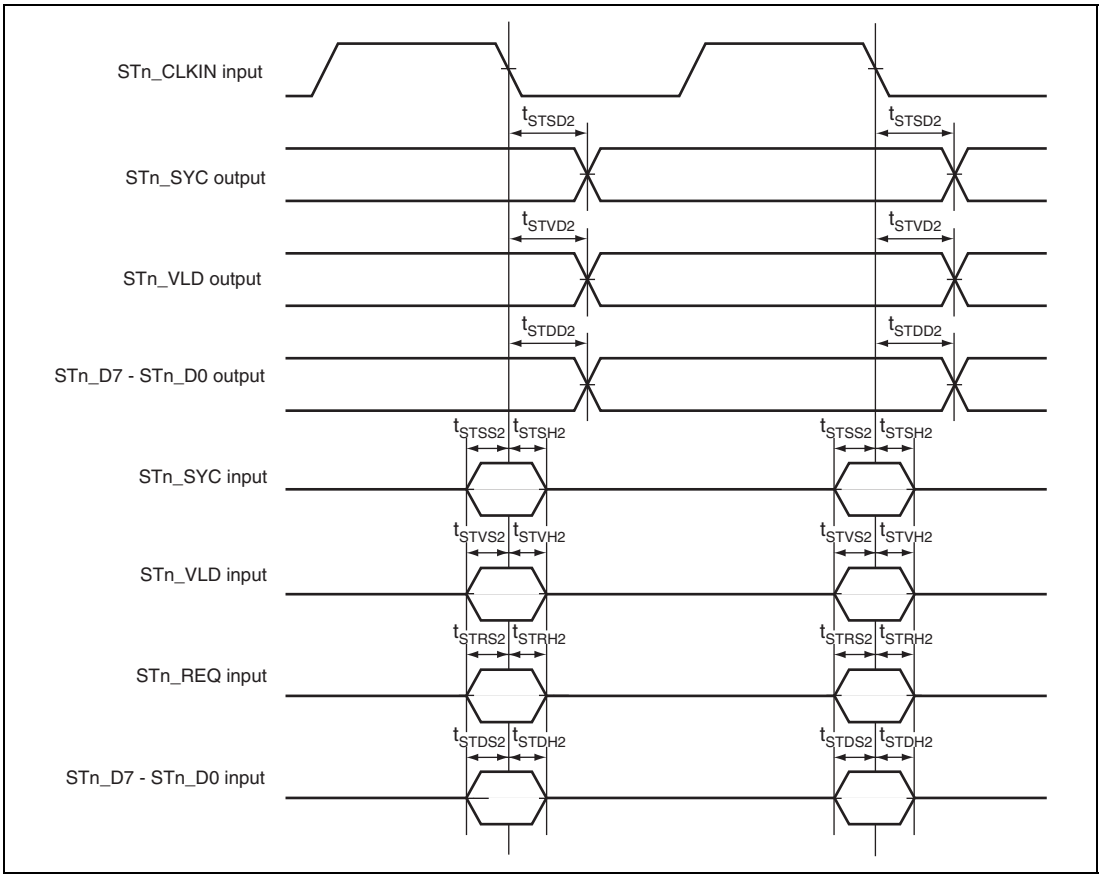


Figure 2.4 STIF Module Signal Timing (4)

2.1.4 STIF Module Signal Timing (5)
(With Stream Input/Output Set Synchronized with ST_CLKOUT Rise Time)

Table 2.5 STIF Module Signal Timing (5)

Conditions: V_{CC} = V_{CC} (PLL) = DV12 = UV12 = 1.1 to 1.3 V, V_{CC}Q = DV33 = 3.1 to 3.5 V,
AV12 = 1.1 to 1.3 V, AV33 = 3.1 to 3.5 V,
V_{SS} = V_{SS} (PLL) = DG12 = UG12 = V_{SS}Q = DG33 = AG12 = AG33 = 0 V,
Ta = –20 to 70°C (regular specifications),
–40 to 85°C (wide temperature specifications)

Item	Symbol	Min.	Max.	Unit	Figure
STn_SYC output delay time 5	t_{STSD5}	-2	5	ns	2.5
STn_VLD output delay time 5	t_{STVD5}	-2	5	ns	
STn_Dm output delay time 5	t_{STDD5}	-2	5	ns	
STn_REQ input setup time 5	t_{STRS5}	11.5	—	ns	
STn_REQ input hold time 5	t_{STRH5}	0.5	—	ns	

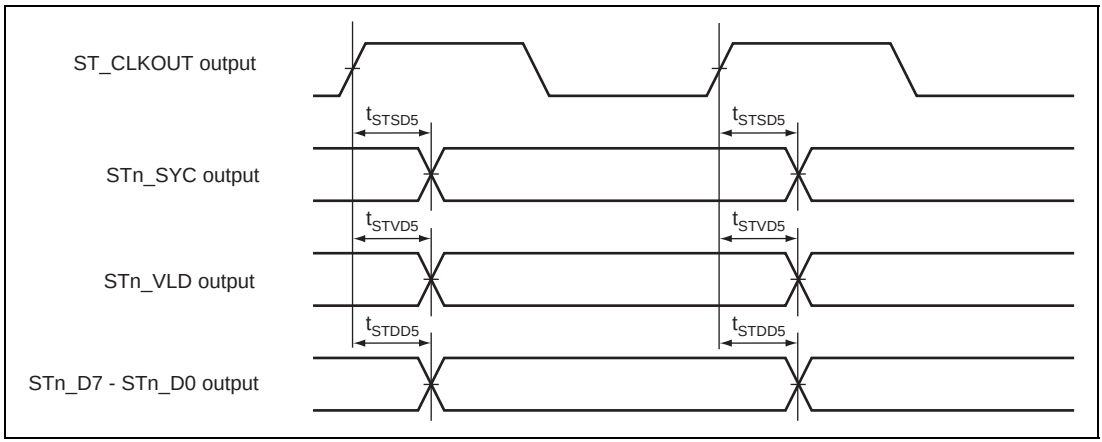


Figure 2.5 STIF Module Signal Timing (5)

2.1.5 STIF Module Signal Timing (6)
(With Stream Input/Output Set Synchronized with ST_CLKOUT Fall Time)

Table 2.6 STIF Module Signal Timing (6)

Conditions: V_{CC} = V_{CC} (PLL) = DV12 = UV12 = 1.1 to 1.3 V, V_{CC}Q = DV33 = 3.1 to 3.5 V,
AV12 = 1.1 to 1.3 V, AV33 = 3.1 to 3.5 V,
V_{SS} = V_{SS} (PLL) = DG12 = UG12 = V_{SS}Q = DG33 = AG12 = AG33 = 0 V,
Ta = –20 to 70°C (regular specifications),
–40 to 85°C (wide temperature specifications)

Item	Symbol	Min.	Max.	Unit	Figure
STn_SYC output delay time 6	t_{STSD6}	-2	5	ns	2.6
STn_VLD output delay time 6	t_{STVD6}	-2	5	ns	
STn_Dm output delay time 6	t_{STDD6}	-2	5	ns	
STn_REQ input setup time 6	t_{STRS6}	11.5	—	ns	
STn_REQ input hold time 6	t_{STRH6}	0.5	—	ns	

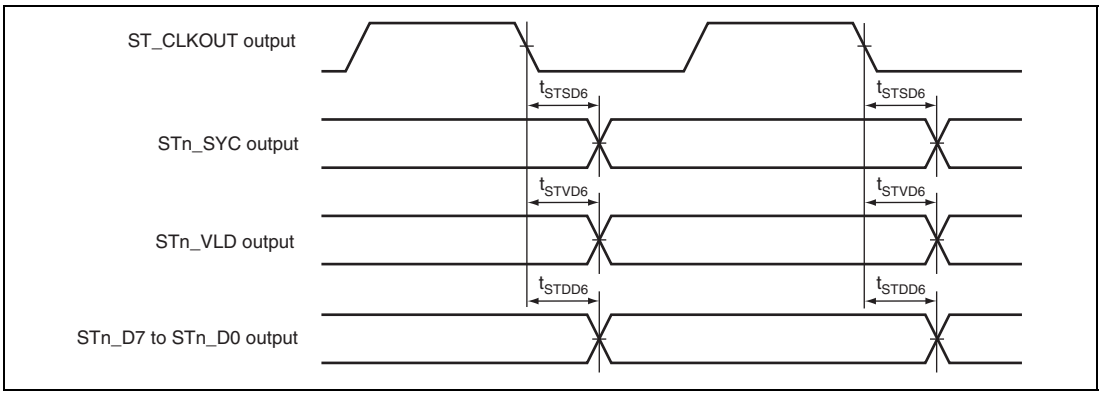


Figure 2.6 STIF Module Signal Timing (6)

Section 3 DMAC That Works with Forward Error Correction Core (A-DMAC)

3.1 Overview

The A-DMAC is a high-level descriptor-mode DMAC having error correction function. This DMAC provides data transfer with memory via an internal shared bus (SHwy bus) and data transfer with an external MPEG device via STIF.

3.1.1 Features

The functions and features of this A-DMAC are as follows:

(1) Channels for checksum processing

- Number of channels: 2
- Transfer direction: Memory← →memory, memory← →STIF
- Descriptor structure: Structure that enables checksum operation, etc., to be continuously performed
- Error check: Checksum calculation function

(2) FEC channels

- Number of channels: 1
- Descriptor structure: Structure that enables processing of any number of data items with a small number of buffers
- Error correction (FEC): XOR calculation function

(3) Other features

- Supported endian: Big endian/little endian
- Number of STIFs connected: Two channels
- Channel arbitration: Round robin scheduling that provides highly efficient use of channels and buses
- Channel operation: Parallel processing

3.1.2 Overall Configuration of the A-DMAC

The A-DMAC is configured as shown in figure 3.1. Table 3.1 gives an overview of A-DMAC submodules.

The A-DMAC is connected to the SHwy bus via the SHwy bus interface, to the STIF0 via the STIF0 interface, and to the STIF1 via the STIF1 interface. The SHwy bus is a shared bus in this LSI operating on the SHwy clock (clks). The STIF is an I/O port for MPEG-2 TS/PS format data. The STIF0 is fixed at CH0 and the STIF1 fixed at CH1.

The A-DMAC has two channels for checksum operation that operate on descriptors. Aside from these channels, the A-DMAC has an FEC channel dedicated for FEC operation. This FEC channel performs XOR operation of FEC operation.

These modules operate in parallel. For example, when the bus for channel 0 for checksum processing is accessed, channel 1 for checksum processing can perform checksum operation.

The arbiter is a module that arbitrates the requests sent from each checksum processing channel and each initiator of the FEC channel. The arbiter arbitrates requests from an initiator in round robin scheduling. If you want to execute CH0 and CH1 simultaneously and raise the priority of CH0 or CH1, the arbiter controls the priorities in descriptor ring units (example: When the descriptor of CH0 or CH1, whichever has a lower priority, runs dry, the arbiter piles up the next descriptor after a certain idling) or controls the priorities by suspending channel processing of lower priority.

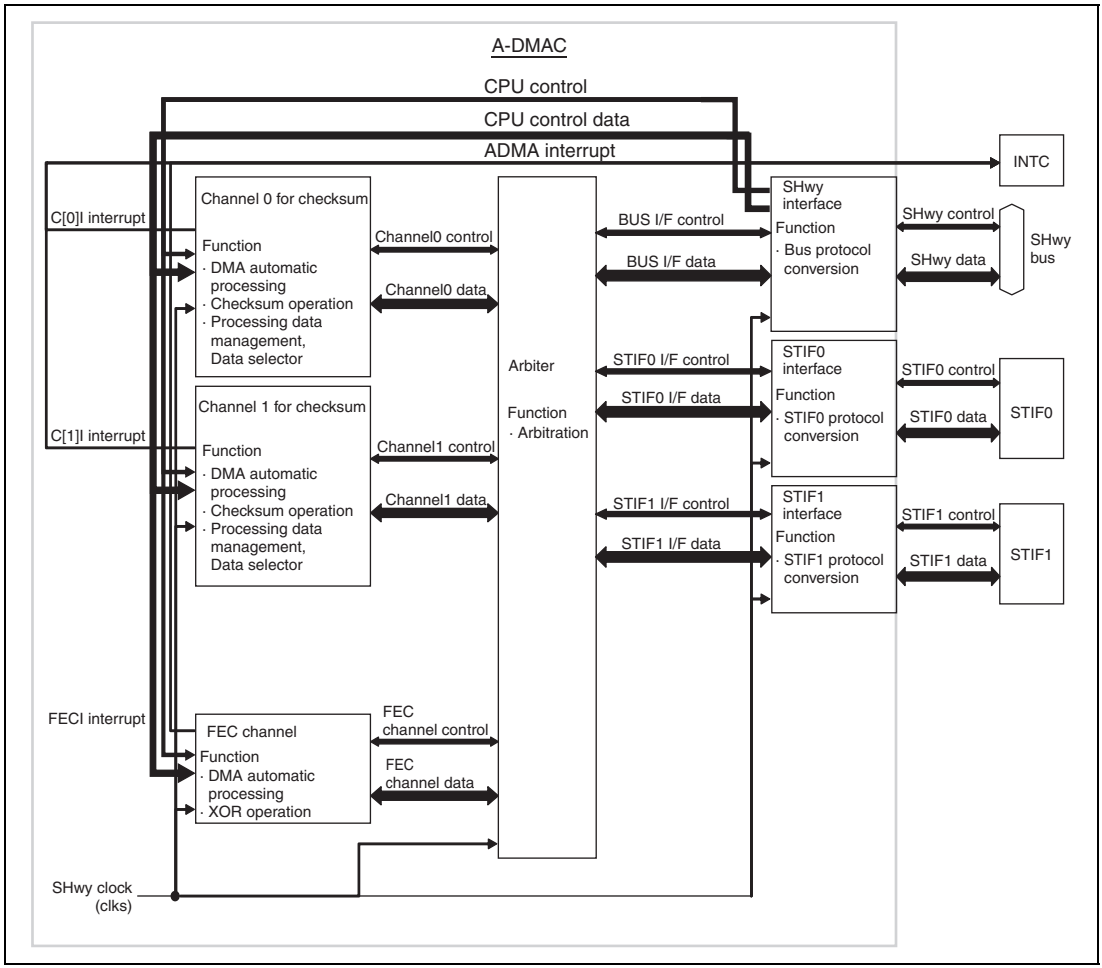


Figure 3.1 Block Diagram of A-DMAC

Table 3.1 A-DMAC Submodules

Submodule Name	Function
Channel for checksum processing	• DMA automatic processing based on descriptors • Checksum operation • Continuous execution of checksum
FEC channel	• DMA automatic processing based on descriptors • XOR operation for any number of data items
Arbiter	• Arbitrates requests from the channel for checksum processing and FEC channel. • Channel arbitration mode is round robin scheduling.
SHwy bus interface	• Conversion between SHwy bus protocol and A-DMAC protocol • Distribution of register R/W requests from the CPU to each module
STIF interface	• Conversion between STIF protocol and A-DMAC protocol

3.1.3 Restrictions on the A-DMAC

The following restrictions apply to the A-DMAC:

- The A-DMAC supports only register access in 32-bit units.
- If the channel processor, or FEC processor is running, write to registers related to the appropriate processor is inhibited. However, you can write data to the following two registers by verifying them after the write even if the appropriate processor is running. Write data repeatedly till verify succeeds.
  - Channel [i] processing control register (C[i]C) (However, do not rewrite the C[i]C_R bit of the running channel processor.)
  - Channel [i] processing interrupt request register (C[i]I)
- Descriptors of data size 0 are inhibited.

3.2 Register Descriptions

The A-DMAC has the following registers.

Do not write to any addresses other than listed below. Operations cannot be guaranteed if writing is attempted. Values read from addresses other than those listed below are undefined.

Table 3.2 Register Configuration

Name	Abbreviation	Number of Bits	Address	Access Size
Channel 0 processing control register	C0C	32	H'FF880440	32
Channel 0 processing mode register	C0M	32	H'FF880444	32
Channel 0 processing interrupt request register	C0I	32	H'FF880448	32
Channel 0 processing descriptor start address register	C0DSA	32	H'FF88047C	32
Channel 0 processing descriptor current address register	C0DCA	32	H'FF880480	32
Channel 0 processing descriptor 0 register	C0D0	32	H'FF880484	32
Channel 0 processing descriptor 1 register	C0D1	32	H'FF880488	32
Channel 0 processing descriptor 2 register	C0D2	32	H'FF88048C	32
Channel 0 processing descriptor 3 register	C0D3	32	H'FF880490	32
Channel 0 processing descriptor 4 register	C0D4	32	H'FF880494	32
Channel 1 processing control register	C1C	32	H'FF8804B0	32
Channel 1 processing mode register	C1M	32	H'FF8804B4	32
Channel 1 processing interrupt request register	C1I	32	H'FF8804B8	32
Channel 1 processing descriptor start address register	C1DSA	32	H'FF8804EC	32
Channel 1 processing descriptor current address register	C1DCA	32	H'FF8804F0	32
Channel 1 processing descriptor 0 register	C1D0	32	H'FF8804F4	32
Channel 1 processing descriptor 1 register	C1D1	32	H'FF8804F8	32
Channel 1 processing descriptor 2 register	C1D2	32	H'FF8804FC	32
Channel 1 processing descriptor 3 register	C1D3	32	H'FF880500	32
Channel 1 processing descriptor 4 register	C1D4	32	H'FF880504	32
FEC DMAC processing control register	FECC	32	H'FF880590	32
FEC DMAC processing interrupt request register	FECI	32	H'FF880594	32

Name	Abbreviation	Number of Bits	Address	Access Size
FEC DMAC processing descriptor start address register	FECDSA	32	H'FF880598	32
FEC DMAC processing descriptor current address register	FECDCA	32	H'FF88059C	32
FEC DMAC processing descriptor 0 register	FECD00	32	H'FF8805A0	32
FEC DMAC processing descriptor 1 register	FECD01D0A	32	H'FF8805A4	32
FEC DMAC processing descriptor 2 register	FECD02S0A	32	H'FF8805A8	32
FEC DMAC processing descriptor 3 register	FECD03S1A	32	H'FF8805AC	32

Table 3.3 Register States in Each Operating Mode

Abbr.	Power-on Reset	Manual Reset	Sleep	Software Standby	Module Standby	Deep Standby
C0C	Initialized	Initialized	Retained	Retained	Retained	Initialized
C0M	Initialized	Initialized	Retained	Retained	Retained	Initialized
C0I	Initialized	Initialized	Retained	Retained	Retained	Initialized
C0DSA	Initialized	Initialized	Retained	Retained	Retained	Initialized
C0DCA	Initialized	Initialized	Retained	Retained	Retained	Initialized
C0D0	Initialized	Initialized	Retained	Retained	Retained	Initialized
C0D1	Initialized	Initialized	Retained	Retained	Retained	Initialized
C0D2	Initialized	Initialized	Retained	Retained	Retained	Initialized
C0D3	Initialized	Initialized	Retained	Retained	Retained	Initialized
C0D4	Initialized	Initialized	Retained	Retained	Retained	Initialized
C1C	Initialized	Initialized	Retained	Retained	Retained	Initialized
C1M	Initialized	Initialized	Retained	Retained	Retained	Initialized
C1I	Initialized	Initialized	Retained	Retained	Retained	Initialized
C1DSA	Initialized	Initialized	Retained	Retained	Retained	Initialized
C1DCA	Initialized	Initialized	Retained	Retained	Retained	Initialized
C1D0	Initialized	Initialized	Retained	Retained	Retained	Initialized
C1D1	Initialized	Initialized	Retained	Retained	Retained	Initialized
C1D2	Initialized	Initialized	Retained	Retained	Retained	Initialized
C1D3	Initialized	Initialized	Retained	Retained	Retained	Initialized
C1D4	Initialized	Initialized	Retained	Retained	Retained	Initialized
FECC	Initialized	Initialized	Retained	Retained	Retained	Initialized
FECI	Initialized	Initialized	Retained	Retained	Retained	Initialized
FECDSA	Initialized	Initialized	Retained	Retained	Retained	Initialized
FECDCA	Initialized	Initialized	Retained	Retained	Retained	Initialized
FECD00	Initialized	Initialized	Retained	Retained	Retained	Initialized
FECD01D0A	Initialized	Initialized	Retained	Retained	Retained	Initialized
FECD02S0A	Initialized	Initialized	Retained	Retained	Retained	Initialized
FECD03S1A	Initialized	Initialized	Retained	Retained	Retained	Initialized

3.2.1 Channel [i] Processing Control Register (C[i]C) (i = 0, 1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	C[i]C_R
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	C[i]C_DWF	—	—	—	C[i]C_VLD	—	—	—	C[i]C_EIE	—	—	—	C[i]C_E
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	C[i]C_R	0	R/W	Reset Writing 1 to this bit when the channel [i] processor is halted causes the channel [i] calculation sequence to be reset. This bit is automatically and immediately set to 0. Setting both this bit and the C[i]C_E bit to 1 causes channel [i] processing to be newly started.
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	C[i]C_DWF	0	R	WAIT State Flag after Descriptor Processing End 0: Non-WAIT state 1: WAIT state There are two methods for understanding the processing state of the DMAC channel [i] descriptor. In one, when the DMAC channel [i] descriptor is set, C[i]DWE is set to 1 and then C[i]DIE is set to 1 to accept the "1 descriptor processing end" interrupt request. In the other, the processing state is observed till this bit is set to 1.
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
8	C[i]C_VLD	0	R/W	Variable-Length Descriptor Control Flag 0: Fixed-length descriptor (32 bytes) 1: Variable-length descriptor (16/32 bytes) The A-DMAC channel uses the 32-byte fixed length structure or 16/32-byte variable-length structure. If this bit is set to 0 to define the descriptor as the fixed-length, the descriptor is always read as 32 bytes. If this bit is set to 1 to define the descriptor as the variable-length, the first 16 bytes are read, and then C[i]D4 and the dummy 12 bytes are read according to the contents of the C[i]CSM[1:0] bits in C[i]D0.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	C[i]C_EIE	0	R/W	"Processing End" Interrupt Request Enable When processing ends, specifies whether to enable or disable the "processing end" interrupt request. 0: Disables the "processing end" interrupt request. 1: Enables the "processing end" interrupt request. A-DMAC channel [i] processing end means fetching of depleted descriptors (invalid descriptors (descriptors where C[i]F0 is set to 0)).
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
0	C[i]C_E	0	R/W	Execution Request Setting this bit to 1 causes channel [i] processing to be started. Setting this bit to 0 causes channel [i] processing to be suspended. When 0 is written to this bit, 0 is read immediately but the channel [i] processor does not stop immediately. That is, the processor stops after it writes back to the descriptor being processed. To understand the channel operating state, set the C[i]C_EIE bit to 1 to accept the "operation end" interrupt request or poll the "operation end" interrupt request flag. To start new processing, the channel [i] of the STIF must be initialized. 0: Channel [i] processing is halted. 1: Channel [i] processing is in progress. Determine if channel [i] processing is suspended when the processor writes back to the descriptor.

3.2.2 Channel [i] Processing Mode Register (C[i]M) (i = 0, 1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	C[i]M_LIE	—	—	—	—
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	C[i]M_LIE	0	R/W	"Last Data Descriptor End Processing" Interrupt Request Enable When last data (C[i]F2=1) descriptor end processing ends, specifies whether to enable or disable the interrupt request. 0: Disables the "last data descriptor processing end" interrupt request. 1: Enables the "last data descriptor processing end" interrupt request.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

3.2.3 Channel [i] Processing Interrupt Request Register (C[i]I) (i = 0, 1)

The C[i]I bit in INT2B13 is asserted as negation of logical OR of all bits in this register. For details on INT2B13, see section 7, INTC/INTC2 of the SH7734 User's Manual: Hardware.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

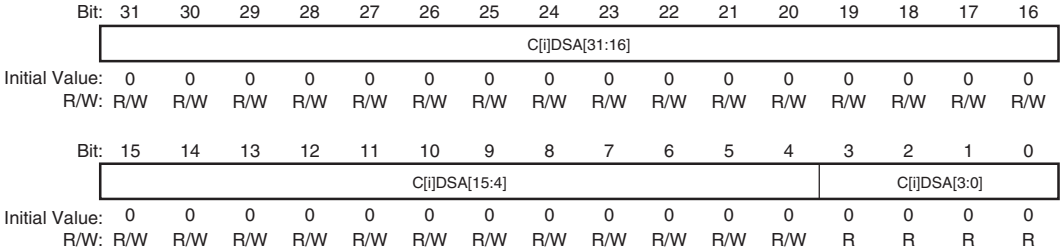
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	C[i]I_DI	—	—	—	C[i]I_LI	—	—	—	—	—	—	—	C[i]I_EI
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	C[i]I_DI	0	R/W	"1 Descriptor Processing End" Interrupt Request (interrupt request to notify you that the processor ended descriptor processing and wrote back the descriptor) This bit is cleared to 0 by writing 1 to it. When 0 is written to this bit, the current state is retained. 0: The "1 descriptor processing end" interrupt is not requested. 1: The "1 descriptor processing end" interrupt is requested.
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
8	C[i]I_LI	0	R/W	"Continuous Data Last Descriptor Processing End" Interrupt Request (interrupt request to notify you that processing described in the descriptor where C[i]F2 is set to 1 ended) This bit is cleared to 0 by writing 1 to it. When 0 is written to this bit, the current state is retained. 0: The "last descriptor processing end" interrupt is not requested. 1: The "last descriptor processing end" interrupt is requested.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	C[i]I_EI	0	R/W	"Processing End" Interrupt Request This bit indicates whether the "processing end" interrupt is requested. This bit is cleared to 0 by writing 1 to it. When 0 is written to this bit, the current state is retained. 0: The "processing end" interrupt is not requested. 1: The "processing end" interrupt is requested. "Processing end" means fetching of depleted descriptors (invalid descriptors (descriptors where C[i]F0 is set to 0)).

3.2.4 Channel [i] Processing Descriptor Start Address Register (C[i]DSA) (i = 0, 1)

Do not write any value to this register when C[i]C_E is set to 1.



Bit	Bit Name	Initial Value	R/W	Description
31 to 4	C[i]DSA[31:4]	All 0	R/W	Descriptor Ring Start Address
3 to 0	C[i]DSA[3:0]	All 0	R	Specify a descriptor ring start address. Set a 16-byte boundary address value.

3.2.5 Channel [i] Processing Descriptor Current Address Register (C[i]DCA) (i = 0, 1)

Do not write any value to this register when C[i]C_E is set to 1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	C[i]DCA[31:16]															
Initial Value:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	C[i]DCA[15:4]												C[i]DCA[3:0]			
Initial Value:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	C[i]DCA[31:4]	All 0	R/W	Descriptor Current Address
3 to 0	C[i]DCA[3:0]	All 0	R	Specify the start address of descriptor processing. Set a 16-byte boundary address value. When descriptor processing is in progress, these bits indicate the address of descriptor currently being processed. After descriptor write-back, these bits indicate the address of the next descriptor.

3.2.6 Channel [i] Processing Descriptor 0 Register (C[i]D0) [Control] (i = 0, 1)

Do not write any value to this register when C[i]C_E is set to 1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	C[i]CRDO[3:0]				C[i]CHDO[3:0]				C[i]SO[3:0]				C[i]DA	C[i]SA	C[i]CSM[1:0]	
Initial Value:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	C[i]F2	C[i]F1	C[i]F0
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	C[i]CRDO[3:0]	All 0	R/W	Transfer Data Destination Data Sequence Specify a swap method for writing transfer data from the A-DMAC to memory such as the STIF and SDRAM. Specify a swap method for writing the checksum operation result obtained from body data in the C[i]CHDO3 to C[i]CHDO0 bits, not these bits.

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	C[i]CRDO[3:0]	All 0	R/W	• When the destination is not the STIF (C[i]DA bit = 0) C[i]CRDO3: Data swap in two-byte units (longword swap in word units) 0: As-is 1: Swap C[i]CRDO2: Data swap in one-byte units (word swap in byte units) 0: As-is 1: Swap C[i]CRDO1: Inversion of bit 1 at address when one or two bytes are accessed 0: As-is 1: Inversion C[i]CRDO0: Inversion of bit 0 at address when one byte is accessed 0: As-is 1: Inversion C[i]CRDO1 and C[i]CRDO0 function for endian adjustment. Note that if an endian different from the endian of this LSI is used, up to three different addresses are accessed from the address where the start and end addresses are specified when an area is allocated. • When the destination is the STIF (C[i]DA bit = 1) C[i]CRDO3: Data swap in two-byte units (longword swap in word units) 0: As-is 1: Swap C[i]CRDO2: Data swap in one-byte units (word swap in byte units) 0: As-is 1: Swap C[i]CRDO1: Data swap in one-bit units (byte swap in one-bit units) 0: As-is 1: Swap C[i]CRDO0: Set this bit to 0 as reserved.

Bit	Bit Name	Initial Value	R/W	Description
27 to 24	C[i]CHDO[3:0]	All 0	R/W	Checksum Operation Result Destination Data Sequence Specify a swap method for writing the checksum operation result from the A-DMAC to memory such as SDRAM. Specify a swap method for writing body data after checksum operation in the C[i]CRDO3 to C[i]CRDO0 bits. C[i]CHDO3: Data swap in two-byte units (longword swap in word units) 0: As-is 1: Swap C[i]CHDO2: Data swap in one-byte units (word swap in byte units) 0: As-is 1: Swap C[i]CHDO1: Inversion of bit 1 at address when one or two bytes are accessed 0: As-is 1: Inversion C[i]CHDO0: Inversion of bit 0 at address when one byte is accessed 0: As-is 1: Inversion C[i]CHDO1 and C[i]CHDO0 function for endian adjustment. Note that if an endian different from the endian of this LSI is used, up to three different addresses are accessed from the address where the start and end addresses are specified when an area is allocated.

Bit	Bit Name	Initial Value	R/W	Description
23 to 20	C[i]SO[3:0]	All 0	R/W	Source Data Sequence Specify a swap method for reading data from memory such as the STIF and SDRAM to the A-DMAC. • When the source is not the STIF (C[i]SA bit = 0) C[i]SO3: Data swap in two-byte units (longword swap in word units) 0: As-is 1: Swap C[i]SO2: Data swap in one-byte units (word swap in byte units) 0: As-is 1: Swap C[i]SO1: Inversion of bit 1 at address when one or two bytes are accessed 0: As-is 1: Inversion C[i]SO0: Inversion of bit 0 at address when one byte is accessed 0: As-is 1: Inversion C[i]SO1 and C[i]SO0 function for endian adjustment. Note that if an endian different from the endian of this LSI is used, up to three different addresses are accessed from the address where the start and end addresses are specified when an area is allocated. • When the source is the STIF (C[i]SA bit = 1) C[i]SO3: Data swap in two-byte units (longword swap in word units) 0: As-is 1: Swap C[i]SO2: Data swap in one-byte units (word swap in byte units) 0: As-is 1: Swap C[i]SO1: Data swap in one-bit units (byte swap in one-bit units) 0: As-is 1: Swap C[i]SO0: Set this bit to 0 as reserved.

Bit	Bit Name	Initial Value	R/W	Description
19	C[i]DA	0	R/W	Destination Attribute Specifies whether the data read source uses the channel [i] (the destination address is not used) of the STIF or the destination address (memory such as SDRAM). 0: Uses the destination address (memory such as SDRAM). 1: Uses the channel [i] of the STIF
18	C[i]SA	0	R/W	Source Attribute Specifies whether the data read source uses the channel [i] (the source address is not used) of the STIF or the source address (memory such as SDRAM). 0: Uses the source address (memory such as SDRAM). 1: Uses the channel [i] of the STIF
17, 16	C[i]CSM[1:0]	00	R/W	Checksum Mode 00: Checksum (not initialized, not written back) Not beginning of data Not end of data 01: Checksum (not initialized, written back) Not beginning of data End of data 10: Checksum (initialized, not written back) Beginning of data Not end of data 11: Checksum (initialized, written back) Beginning of data End of data
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2	C[i]F2	0	R/W	Descriptor Execution Flag 2 When splitting continuous data into several descriptors for execution, set this bit to 1 in the descriptor that processes the last data part (because the pointer in the A-DMAC must be initialized to process the next descriptor). 0: Non-last descriptor that processes continuous data 1: Last descriptor that processes continuous data
1	C[i]F1	0	R/W	Descriptor Execution Flag 1 When this bit is 1, the A-DMAC regards this descriptor as the last descriptor in the descriptor ring area. When processing of this descriptor ends, the A-DMAC returns to the beginning (descriptor start address) of the descriptor ring area. 0: Non-last descriptor ring 1: Last descriptor ring
0	C[i]F0	0	R/W	Descriptor Execution Flag 0 When this bit is 0, the A-DMAC ends processing because this descriptor is invalid. When this bit is 1, this descriptor is valid. When this bit is 1 (valid descriptor), the A-DMAC sets this bit to 0 and writes back to the original address after processing of this descriptor ends. 0: Invalid descriptor 1: Valid descriptor

3.2.7 Channel [i] Processing Descriptor 1 Register (C[i]D1) [Source Address] (i = 0, 1)

Do not write any value to this register when C[i]C_E is set to 1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	C[i]D1[31:16]															
Initial Value:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	C[i]D1[15:0]															
Initial Value:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	C[i]D1[31:0]	All 0	R/W	Source Address Specify a source address. This register is used when source access involves memory reference. It is not used in the STIF. When copying a key or initial vector from the source, set 0000 in the lower four bits (16-byte boundary). When splitting continuous data into several descriptors for execution, specify the same source address in all the descriptors.

3.2.8 Channel [i] Processing Descriptor 2 Register (C[i]D2) [Destination Address] (i = 0, 1)

Do not write any value to this register when C[i]C_E is set to 1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	C[i]D2[31:16]															
Initial Value:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	C[i]D2[15:0]															
Initial Value:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	C[i]D2[31:0]	All 0	R/W	Transfer Data Destination Address Specify a destination address to which to write back the transfer data. When splitting continuous data into several descriptors for execution, specify the same source address in all the descriptors.

3.2.9 Channel [i] Processing Descriptor 3 Register (C[i]D3) [Data Length] (i = 0, 1)

Do not write any value to this register when C[i]C_E is set to 1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	C[i]DWE	C[i]DIE	—	—	—	—	—	—	—	—	—	—	—	—
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	C[i]D3[15:0]															
Initial Value:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
29	C[i]DWE	0	R/W	"1 Descriptor Processing End" Interrupt Release Wait Enable If this bit is 1 and the "1 descriptor processing" interrupt is requested, the A-DMAC waits for the interrupt release before it moves to next descriptor processing. 0: Does not observe the "1 descriptor processing end" interrupt. 1: Enables "1 descriptor processing end" interrupt release wait.
28	C[i]DIE	0	R/W	"1 Descriptor Processing End" Interrupt Request Enable Specifies whether to enable or disable the "1 descriptor processing end" interrupt when processing of this descriptor ends. Processing does not end even if the "1 descriptor processing end" interrupt request is enabled. 0: Disables the "1 descriptor processing end" interrupt request. 1: Enables the "1 descriptor processing end" interrupt request.
27 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	C[i]D3[15:0]	All 0	R/W	Target Data Size (Byte Length) The target data size range that can be specified in these bits is as follows: 0 < C[I]D3[15:0] <= 2^16-96 Basically, set a multiple of the length of block to be processed. For checksum, set a multiple of 2 bytes. When using continuous data over several descriptors, set the total of sizes specified in each descriptor in multiples of the block length. Also set the total size not to exceed 2^32.

3.2.10 Channel [i] Processing Descriptor 4 Register (C[i]D4) [Checksum Value Write Address] (i = 0, 1)

Do not write any value to this register when C[i]C_E is set to 1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	C[i]D4[31:16]															
Initial Value:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	C[i]D4[15:1]															-
Initial Value:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	C[i]D4[31:1]	All 0	R/W	Write address of the checksum calculation result. The lower four bits should be 0. Set a 16-byte boundary address.
0	—	0	R	

3.2.11 FEC DMAC Processing Control Register (FECC)

A suspension direction is evaluated after descriptor processing in progress is completed (descriptor write-back). If FECC_E is 1 when the suspension direction is evaluated, the FEC DMAC enters the WAIT cycle. If the FEC DMAC is restarted because 1 was written to FECC_E during the WAIT cycle, the FEC DMAC moves to next descriptor read unless the descriptor written back just before is the last descriptor. If the descriptor is the last descriptor, the FEC DMAC ends processing and enters the IDLE state.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	FECC_R	—	—	—	FECC_DWF	—	—	—	FECC_DWE	—	—	—	FECC_DIE
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	FECC_LIE	—	—	—	FECC_NIE	—	—	—	FECC_EIE	—	—	—	FECC_E
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28	FECC_R	0	R/W	Reset Writing 1 to this bit during stop causes the FEC processing sequence to be reset. This bit is automatically and immediately set to 0. Setting both this bit and the FECC_E bit to 1 causes FEC processing to be newly started except when the following bits are set to 1: FECC_DWE, FECC_DIE, FECC_LIE, FECC_NIE, FECC_EIE, FECC_E, FECDSA, and FECDCA
27 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
24	FECC_DWF	0	R	WAIT State Flag after Descriptor Processing End 0: Non-WAIT state 1: WAIT state There are two methods for understanding the processing state of the FEC DMAC descriptor. In one, when the FEC DMAC is executed, FEC_DWE is set to 1 and then FECC_DIE is set to 1 to accept the "1 descriptor processing end" interrupt request. In the other, the processing state is observed till this bit is set to 1.
23 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	FECC_DWE	0	R/W	"1 Descriptor Processing End" Interrupt Request Release Wait Enable When this bit is 1, the FEC DMAC ends processing of this descriptor and enters the WAIT state unless FECC_DIE is 0 after write-back. If the "1 descriptor processing end" interrupt is requested, the FEC DMAC waits for release of the interrupt before it moves to processing of the next descriptor. If this descriptor is the last descriptor when the interrupt is released, the FEC DMAC ends processing and enters the IDLE state. If this descriptor is not the last descriptor, the FEC DMAC reads the next descriptor. 0: The FEC DMAC does not enter the WAIT state when the "1 descriptor processing end" interrupt is requested. 1: The FEC DMAC enters the WAIT state when the "1 descriptor processing end" interrupt is requested.
19 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	FECC_DIE	0	R/W	"1 Descriptor Processing End" Interrupt Request Enable Specifies whether to enable or disable the "1 descriptor processing end" interrupt request when processing of this descriptor ends. Processing does not end even if this interrupt is requested. This bit functions as the FECI_DI mask. 0: Disables the "1 descriptor processing end" interrupt request. 1: Enables the "1 descriptor processing end" interrupt request.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	FECC_LIE	0	R/W	"Last Descriptor Processing End" Notification Interrupt Request Enable Specifies whether to enable or disable the "last descriptor processing end" interrupt request when processing of the last descriptor ends (FECI_LI mask). 0: Disables the "last descriptor processing end" interrupt request. 1: Enables the "last descriptor processing end" interrupt request.
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	FECC_NIE	0	R/W	"Invalid Descriptor" Notification Interrupt Request Enable Specifies whether to enable or disable the "invalid descriptor" notification interrupt request when the invalid descriptor is fetched (FECI_NI mask). 0: Disables the "invalid descriptor" notification interrupt request. 1: Enables the "invalid descriptor" notification interrupt request.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	FECC_EIE	0	R/W	"Processing End" Interrupt Request Enable Specifies whether to enable or disable the "processing end" interrupt request when processing ends (FECI_EI mask). 0: Disables the "processing end" interrupt request. 1: Enables the "processing end" interrupt request.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
0	FECC_E	0	R/W	Execution Request Setting this bit to 1 causes FEC processing to be started. Setting this bit to 0 during FEC processing causes FEC processing to be suspended. After FEC processing ends, this bit is automatically set to 0. There are two methods for understanding the FEC DMAC operating state. In one, when the FEC DMAC is executed, FECC_EIE is set to 1 to accept the "operation end" interrupt request. In the other, the operating state is observed till the key of this bit is set to 0. 0: FEC processing is halted. 1: FEC processing is in progress.

3.2.12 FEC DMAC Processing Interrupt Request Register (FECI)

The FECI bit in INT2B13 is asserted as negation of logical OR of all bits in this register. For details on INT2B13, see section 7, INTC/INTC2 of the SH7734 User's Manual: Hardware.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

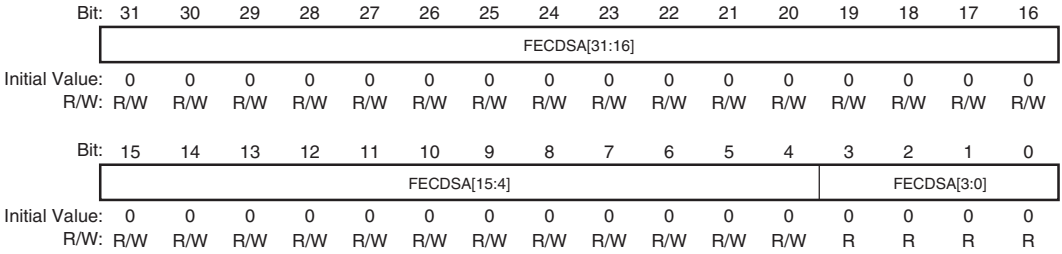
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	FECI_DI	—	—	—	FECI_LI	—	—	—	FECI_NI	—	—	—	FECI_EI
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	FECI_DI	0	R/W	"1 Descriptor Processing End" Interrupt Notification Request This interrupt notifies you that the FEC DMAC ended 1 descriptor processing and wrote back the descriptor. This bit is cleared to 0 by writing 1 to it. When 0 is written to this bit, the current state is retained. This interrupt is masked by the FECC_DIE bit of the descriptor. 0: The "1 descriptor processing end" interrupt is not requested. 1: The "1 descriptor processing end" interrupt is requested.
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	FECI_LI	0	R/W	"Last Descriptor (descriptor where FECD00_F2 is 1) Processing End" Interrupt Notification Request This interrupt notifies you that the FEC DMAC wrote back the last descriptor and ended last descriptor processing. The FEC DMAC enters the IDLE state after it ended last descriptor processing. This bit is cleared to 0 by writing 1 to it. When 0 is written to this bit, the current state is retained. If this bit is set, the FEC DMAC is in the initial state because descriptors ran dry. In this case, replenish new descriptors and then restart the FEC DMAC. This interrupt is masked by the FECC_LIE bit of the FECC. 0: The "last descriptor processing end" interrupt is not requested. 1: The "last descriptor processing end" interrupt is requested.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	FECI_NI	0	R/W	"Invalid Descriptor (descriptor where FECD00_F0 is 0) Interrupt Interrupt request for read end notification. When this interrupt request is made, the FEC DMAC ends processing and enters the IDLE state. This bit is cleared to 0 by writing 1 to it. When 0 is written to this bit, the current state is retained. This interrupt is masked by the FECC_NIE bit of the FECC. If this bit is set, the FEC DMAC is in the initial state because descriptors ran dry. In this case, replenish new descriptors and then restart the FEC DMAC. 0: The "invalid descriptor processing end" interrupt is not requested. 1: The "invalid descriptor processing end" interrupt is requested.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	FECI_EI	0	R/W	"Processing End" Interrupt Request This interrupt notifies you that processing ended due to the FECI_LI or FECI_NI interrupt source and the FEC DMAC is now in the IDLE state. This bit is cleared to 0 by writing 1 to it. When 0 is written to this bit, the current state is retained. This interrupt is masked by the FECC_EIE bit of the FECC. If this bit is set, the FEC DMAC is in the initial state because descriptors ran dry. In this case, replenish new descriptors and then restart the FEC DMAC. 0: The "processing end" interrupt is not requested. 1: The "processing end" interrupt is requested.

3.2.13 FEC DMAC Processing Descriptor Start Address Register (FECDSA)

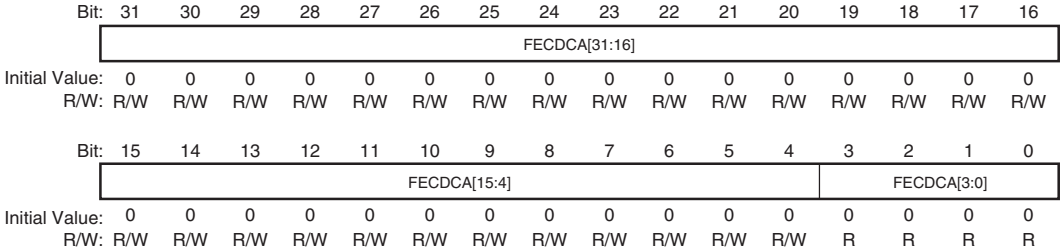
Do not write any value to this register when FECC_E is set to 1.



Bit	Bit Name	Initial Value	R/W	Description
31 to 4	FECDSA[31:4]	All 0	R/W	Descriptor Ring Start Address
3 to 0	FECDSA[3:0]	All 0	R	Specify a descriptor ring start address. Set a 16-byte boundary address value.

3.2.14 FEC DMAC Processing Descriptor Current Address Register (FECDCA)

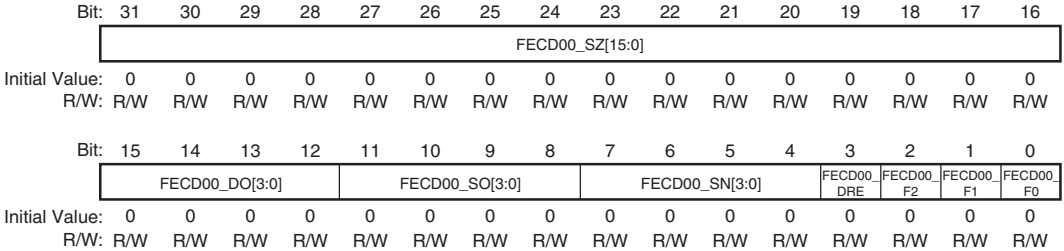
Do not write any value to this register when FECC_E is set to 1.



Bit	Bit Name	Initial Value	R/W	Description
31 to 4	FECDSA[31:4]	All 0	R/W	Descriptor Current Address
3 to 0	FECDSA[3:0]	All 0	R	Specify the start address of descriptor processing. Set a 16-byte boundary address value. When descriptor processing is in progress, these bits indicate the address of descriptor currently being processed. After descriptor write-back, these bits indicate the address of the next descriptor. When the FEC DMAC enters the IDLE state after it has processed the descriptor where the last flag is set, this register indicates the address of the next descriptor of the descriptor where the last flag is set.

3.2.15 FEC DMAC Processing Descriptor 0 Register (FECD00) [Control]

Do not write any value to this register when FECC_E is set to 1.



Bit	Bit Name	Initial Value	R/W	Description
31 to 16	FECD00_SZ[15:0]	All 0	R/W	Data Size (Byte Length) Specify the byte size of data to be processed. Set a value from 0 to 65504. Do not set a value from 65505 to 65536.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	FECD00_DO[3:0]	All 0	R/W	These bits function when the destination is read or written. FECD00_DO3: Data swap in two-byte units (longword swap in word units) 0: As-is 1: Swap FECD00_DO2: Data swap in one-byte units (word swap in byte units) 0: As-is 1: Swap FECD00_DO1: Inversion of bit 1 at address when one or two bytes are accessed 0: As-is 1: Inversion FECD00_DO0: Inversion of bit 0 at address when one byte is accessed 0: As-is 1: Inversion FECD00_DO1 and FECD00_DO0 function for endian adjustment. Note that if an endian different from the endian of this LSI is used, up to three different addresses are accessed from the address where the start and end addresses are specified when an area is allocated.

Bit	Bit Name	Initial Value	R/W	Description
11 to 8	FECD00_SO[3:0]	All 0	R/W	These bits function when the source is read. FECD00_SO3: Data swap in two-byte units (longword swap in word units) 0: As-is 1: Swap FECD00_SO2: Data swap in one-byte units (word swap in byte units) 0: As-is 1: Swap FECD00_SO1: Inversion of bit 1 at address when one or two bytes are accessed 0: As-is 1: Inversion FECD00_SO0: Inversion of bit 0 at address when one byte is accessed 0: As-is 1: Inversion FECD00_SO1 and FECD00_SO0 function for endian adjustment. Note that if an endian different from the endian of this LSI is used, up to three different addresses are accessed from the address where the start and end addresses are specified when an area is allocated.
7 to 4	FECD00_SN[3:0]	All 0	R/W	Number of Source Addresses Specify the number of source addresses subject to FEC operation. 0000: The number of source addresses is 1. 0001: The number of source addresses is 2. Others: Reserved (setting prohibited)

Bit	Bit Name	Initial Value	R/W	Description
3	FECD00_DRE	0	R/W	Destination Read Enable 0: Does not read the destination. 1: Reads the destination and updates the read values.
2	FECD00_F2	0	R/W	Descriptor Execution Flag 2 When 1, this bit explicitly indicates that this descriptor is the last descriptor that should operate. (Another method for explicitly indicating that this descriptor is the last descriptor is to place an invalid descriptor immediately after this descriptor.) 0: This descriptor is not the last descriptor that should operate. 1: This descriptor is the last descriptor that should operate.
1	FECD00_F1	0	R/W	Descriptor Execution Flag 1 When this bit is 1, the FEC DMAC regards this descriptor as the last descriptor in the descriptor ring area and returns to the beginning (descriptor start address) of the descriptor ring area when processing of this descriptor ends. 0: This descriptor is not regarded as the last descriptor in the descriptor ring area. 1: This descriptor is regarded as the last descriptor in the descriptor ring area.
0	FECD00_F0	0	R/W	Descriptor Execution Flag 0 When this bit is 0, processing of this descriptor ends because this descriptor is invalid. If the descriptor where FECD00_F0 is 0 is processed, the FEC DMAC suspends FEC processing on the assumption that FECC_E is 0. When this bit is 1, this descriptor is valid. If this descriptor is valid, the FEC DMAC sets this bit to 0 and writes back to the original address when processing of this descriptor ends. 0: This descriptor is invalid. 1: This descriptor is valid.

3.2.16 FEC DMAC Processing Descriptor 1 Register (FECD01D0A) [Destination Address]

Do not write any value to this register when FECC_E is set to 1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FECD01D0A[31:16]															
Initial Value:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FECD01D0A[15:0]															
Initial Value:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	FECD01D0A[31:0]	All 0	R/W	Destination Address Specify a processing data write-back destination address.

3.2.17 FEC DMAC Processing Descriptor 2 Register (FECD02S0A) [Source 0 Address]

Do not write any value to this register when FECC_E is set to 1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FECD02S0A[31:16]															
Initial Value:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FECD02S0A[15:0]															
Initial Value:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	FECD02S0A[31:0]	All 0	R/W	Specify the start address of source 0 data.

3.2.18 FEC DMAC Processing Descriptor 3 Register (FECD03S1A) [Source 1 Address]

Do not write any value to this register when FECC_E is set to 1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FECD03S1A[31:16]															
Initial Value:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FECD03S1A[15:0]															
Initial Value:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	FECD03S1A[31:0]	All 0	R/W	Specify the start address of source 1 data.

3.3 Functions

Table 3.4 lists A-DMAC security/network functions.

Table 3.4 A-DMAC Security/Network Functions

Classification	Item	Description	Conforming/ Supported Standard
Error detection	Checksum	• 1's complement sum operation	RFC1071 support
Error correction	FEC	• FEC XOR operation • Support of any number of FEC matrixes	RFC2733 Pro-MPEG support

3.3.1 DMAC Channel Function

The A-DMAC has two DMAC channels for checksum processing.

3.3.2 Checksum

Checksum is a data error detection scheme. Checksum splits the entered data in 16-bit units and calculates their 1's complement sum to detect an error.

The A-DMAC has a function to calculate the 1's complement sum of data obtained via DMA transfer.

3.3.3 FEC Channel

The A-DMAC has one channel for FEC operation. This channel can perform XOR operation for the data obtained via DMA transfer and write back to memory because it is of a descriptor structure that can cope with FEC operation of any number of rows.

3.3.4 FEC Operation

FEC is an error correction method. This method enables the receiving side to repair the lost packet without requesting packet retransmission. When repairing the lost packet by using FEC, the transmitting side uses the original packet group to generate a redundant packet (FEC packet). When transmitting 100 packets, for example, the transmitting side generates a 10 x 10 packet matrix, XORs the ten original packets aligned to one row or column, and generates one FEC packet per row (column). In this example, 20 FEC packets are generated. The transmitting side transmits the original data packet group and FEC packets to the receiving side. To check whether the original packets are lost, the receiving side aligns the original packets and FEC packets to the matrix as in the transmitting side. If a lost original packet is found, the transmitting side can repair the packet by XORing the other packets in the row and column to which the lost packet belongs with the FEC packets. Like this, the transmitting side and receiving side need to share the number of rows and columns of matrix aligned to generate FEC packets before transmitting and receiving the packets.

The A-DMAC has the XOR calculation function used for FEC operation and supports the following FEC specifications of RFC2733 and Pro-MPEG:

- XOR calculation of any number of rows (columns)
  A variable-length descriptor supports the FEC structure of theoretically infinite length.
- One-dimensional FEC
  Not only two-dimensional FEC but also one-dimensional FEC is supported because processing is performed per row (column).

The CPU must perform the following operations:

- FEC matrix alignment
- Lost packet detection
- Unification of the lengths of packets that constitute a row (column)
  (Packets less than the maximum packet length are padded with 0.)
- Repair of a portion of timestamp and payload type from the result obtained from the A-DMAC

3.4 Channel Operation

3.4.1 Descriptor Format

The A-DMAC can automatically perform DMA transfer between memory and the STIF without the CPU based on the descriptor containing information such as a buffer pointer and its data size. The A-DMAC automatically performs operations such as reading data from memory and writing the data to the STIF according to the information stored in this descriptor.

Figure 3.2 shows the descriptor format. The gray parts in the figure are ignored when descriptor processing is started and "0" is written back to these parts after descriptor processing ends. For details on each bit, see section 3.2.6, Channel [i] Processing Descriptor 0 Register (C[i]D0) [Control] (i = 0, 1), to section 3.2.10, Channel [i] Processing Descriptor 4 Register (C[i]D4) [Checksum Value Write Address] (i = 0, 1).

Bit / Address	31	30	29	28	27-26	25-24	23-20	19	18	17-16	15-3	2-1	0
0	CRDO[3:0]				CHDO[3:0]		SO[3:0]	DA	SA	CSM[1:0]		F[2:0]	
+4	D1 [31:0]												
+8	D2 [31:0]												
+12			DWE	DIE							D3 [15:0]		
+16	D4 [31:1]												
+20													
+24													
+28													

Figure 3.2 Descriptor Format

A descriptor is 16/32-byte variable length or 32-byte fixed length. Select variable length or fixed length from the variable-length descriptor control flag (C[i]C_VLD) of the channel [i] processing control register (C[i]C). If C[i]C_VLD is set to 1 to operate the descriptor as the variable-length descriptor, the remaining 16 bytes are read when the following conditions are met:

- Checksum calculation result write-back is set (C[i]CSM0 = 1).

3.4.2 Basic Channel Operation

When "1" is written to the C[i]C_E bit of the channel [i] processing control register (C[i]C), channel [i] reads the descriptor from the C[i]DCA31 to C[i]DCA4 addresses. If a fixed length is set in C[i]C_VLD, the first 32 bytes are continuously read. If variable length is set, the remaining 16 bytes are read according to the previously explained conditions.

If the C[i]F0 flag of the first longword of a descriptor is 1, the descriptor is fetched to the appropriate register of channel [i] processing descriptor 0 (C[i]D0) to channel [i] processing descriptor 4 (C[i]D4). After 1-descriptor processing ends, channel [i] sets the C[i]F0 flag to 0 and writes back to the original area.

Any number of descriptors can be allocated onto memory in the ring form. Processing is started from the descriptor allocated to the address indicated by the channel [i] processing descriptor current address register (C[i]DCA). If descriptors where the C[i]F0 flag of channel [i] processing descriptor 0 (C[i]D0) is set to 1 continue, channel [i] processes them one after another. If the C[i]F1 flag of channel [i] processing descriptor 0 (C[i]D0) is 1, channel [i] assumes that the end of descriptor ring was detected and processes the descriptor allocated to the address indicated by the channel [i] processing descriptor start address register (C[i]DSA). To end descriptor processing, allocate the invalid descriptor where the C[i]F0 flag of channel [i] processing descriptor 0 (C[i]D0) is set to 0.

If processing for single continuous data is divided into several descriptors, the data size of each processing must be saved between several descriptor processing. Conversely, to handle different data, the data size of each processing must be initialized. For this reason, whether the descriptor currently being executed handles the end of continuous data must be indicated in the descriptor. Set this in the C[i]F2 flag.

The A-DMAC does not allow you to set data size 0 in C[i]D315 to C[i]D30.

3.4.3 Checksum

Figure 3.3 shows an example of descriptors that execute only checksum. In figure 3.3, (a) is an example of continuously allocating checksum descriptors each of which completes one processing and (b) is an example of splitting processing into several checksum descriptors and allocating the last checksum descriptor that completes processing.

In the descriptor that performs checksum operation, set the data size in multiples of two bytes. However, if split descriptors are used, a value other than a multiple of two bytes can be set as the data size in each descriptor but the total number of data sizes specified in the split descriptors must be set so that it becomes a multiple of 2. (If processing is split into several descriptors and an odd size is specified in the non-last descriptor, the A-DMAC waits for the next descriptor without processing data of the last one byte.)

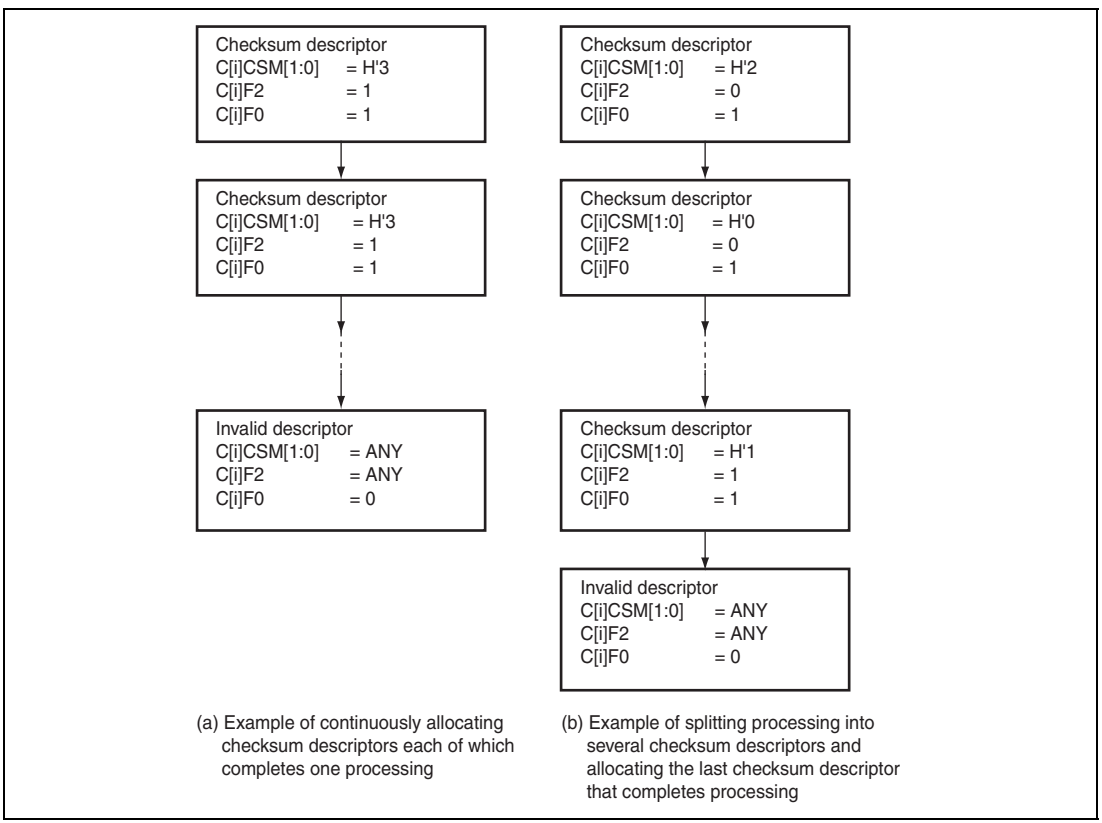


Figure 3.3 Examples of Allocating Checksum Descriptors

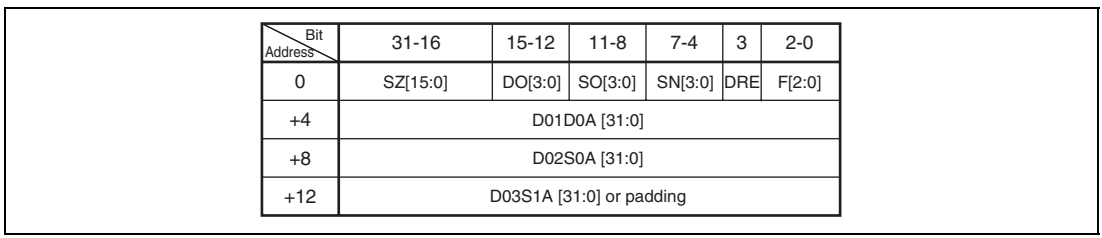
3.5 FEC Channel Operation

3.5.1 Descriptor Format for FEC Channel

Figure 3.4 shows the descriptor format for the FEC channel. The FEC channel can automatically perform DMA transfer with memory without the CPU according to descriptor information.

Two source addresses can be specified in a descriptor but linking descriptors in the ring form provides FEC processing of any number of rows (columns).

Only SHwy bus can access FEC channel data due to its application, so information indicating the source and destination directions (SHwy bus or STIF) is not included in the descriptor. The size of data subject to FEC operation must match the data size set in the FECD0_SZ15 to FECD0_SZ0 flag of FEC DMAC processing descriptor 0 (FECD00) in the first longword of the FEC descriptor. For this reason, when processing data less than the data size set in the FECD0_SZ15 to FECD0_SZ0 flag, you must pad the data with 0 and perform FEC processing.



Bit / Address	31-16	15-12	11-8	7-4	3	2-0
0	SZ[15:0]	DO[3:0]	SO[3:0]	SN[3:0]	DRE	F[2:0]
+4	D01D0A [31:0]					
+8	D02S0A [31:0]					
+12	D03S1A [31:0] or padding					

Figure 3.4 FEC DMAC Descriptor Format

3.5.2 Basic FEC Channel Operation

When "1" is written to the FECC_E bit of the FEC DMAC processing control register (FECC), the FEC channel starts descriptor read. If the FECD00_F0 flag in the first longword is "1", descriptors are fetched in turn to the appropriate register, starting from FECD00 in the first longword.

After descriptor read is completed, the FEC channel reads data from memory space indicated by a source address and performs FEC operation (XOR calculation). After XOR calculation with all source addresses is completed, the FEC channel writes back the result to destination address space. After 1-descriptor processing ends, the FEC channel sets the FECD00_F0 flag to "0" and writes back to the original area.

To support the FEC matrix operation of any number of rows and columns, the FEC channel installed on the A-DMAC temporarily writes back the FEC operation result of source rows/columns that can be processed by one descriptor to the destination address. If the FEC matrix consists of two rows, the FEC matrix operation ends in one descriptor. If the FEC matrix consists of 3 rows/columns or more, the FEC channel splits the matrix into several descriptors and performs FEC matrix processing. If this processing must be split into several descriptors, use the FECD00_DRE bit to control the FEC operation.

Figure 3.5 shows an example of descriptor configuration where the FEC matrix operation is split into several descriptors for execution. In the first descriptor that starts the FEC operation, FECD00_DRE is set to 0 because the operation result is not yet written. In the second and subsequent descriptors, the FEC operation is continued. In other words, to XOR the calculation result of the previous descriptor with the current descriptor source, FECD00_DRE is set to 1. Piling up such descriptors till the number of rows or columns for the FEC matrix operation is met makes it possible to obtain the last XOR operation result of the target row (column).

Any number of descriptors can be allocated onto memory in the ring form. Processing is started from the descriptor allocated to the address indicated by the FEC DMAC processing descriptor current address register (FECDCA). If descriptors where the FECD00_F0 flag of FEC DMAC processing descriptor 0 (FECD00) is set to 1 continue, the FEC channel processes them one after another. If the FECD00_F1 flag of FECD00 is 1, the FEC channel assumes that the end of descriptor ring was detected and processes the descriptor allocated to the address indicated by the FEC DMAC processing descriptor start address register (FECDSA). To end descriptor processing, allocate the invalid descriptor where the FECD00_F0 flag of FECD00 is set to "0" or allocate the last descriptor where the FECD00_F2 flag of FECD00 is set to "1".

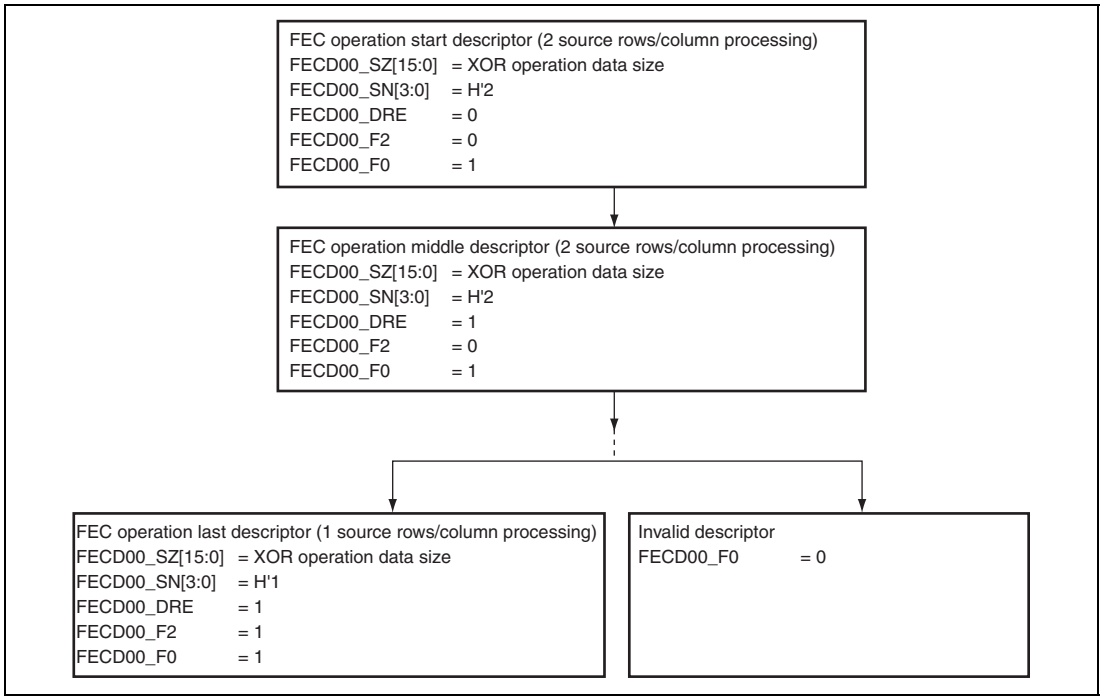


Figure 3.5 Example of FEC Descriptor Configuration

3.6 Usage Notes

3.6.1 Data Transfer Size Set in Descriptor for A-DMAC Channel Operation

The data transfer size to be set in a descriptor for A-DMAC channel operation is as explained in the description of the C[i]D3[15:0] bits in section 3.2.9, Channel [i] Processing Descriptor 3 Register (C[i]D3) [Data Length] (i = 0, 1). Note however that the following restrictions are applied when the relevant A-DMAC channel is operating together with an STIF in PS mode.

When operating together with an STIF in PS mode, a descriptor for dummy data transfer must be prepared to satisfy the expression of (data transfer size specified by A-DMAC descriptor) = (setting of the SIZE[31:0] bits in STSIZER) + dummy data = multiple of 192 bytes. This means that the sum of the C[i]D3[15:0] bits for each descriptor should be a multiple of 192 bytes when data transfer is divided to be performed by multiple descriptors.

Accordingly, in PS output mode, data including dummy data is transferred from the A-DMAC to the STIF, but the dummy data is deleted inside the STIF and only the data that should really be transferred will be output from the STIF to an external device.

Similarly, in PS input mode, the STIF retrieves only the amount of data that should really be transferred from an external device. However, the A-DMAC transfers the data that should really be transferred followed by dummy data. Software should ignore the dummy data stored in memory.

Section 4 IEBus™ Controller

4.1 Overview

The IEBus is a logical unit supporting the IEBus™ (Inter Equipment Bus™) specifications.

Note: IEBus™ is a trademark of Renesas Electronics Corporation.

4.1.1 Features

The IEBus has the following features.

- IEBus protocol control (layer 2)
  - Half duplex asynchronous communication
  - Multi-master method
  - Broadcast communication function
  - Two modes with different transfer rates available

Table 4.1 Two Modes with Different Transfer Rates

Mode	6.29-MHz Operation	Maximum Number of Bytes to be Transferred (byte/frame)
0	Approximately 4.1 kbps	16
1	Approximately 18 kbps	32

- Built-in data transmit and receive buffers
  - Transmit buffer size: 32 bytes
  - Receive buffer size: 32 bytes
  - Continuous transmission and reception up to 32 bytes, or the maximum number of bytes to be transferred of mode 1
- IEBus clock frequency
  - 6.29 MHz

4.1.2 Block Diagram

Figure 4.1 is a block diagram of the IEBus. Table 4.2 shows functions in each block.

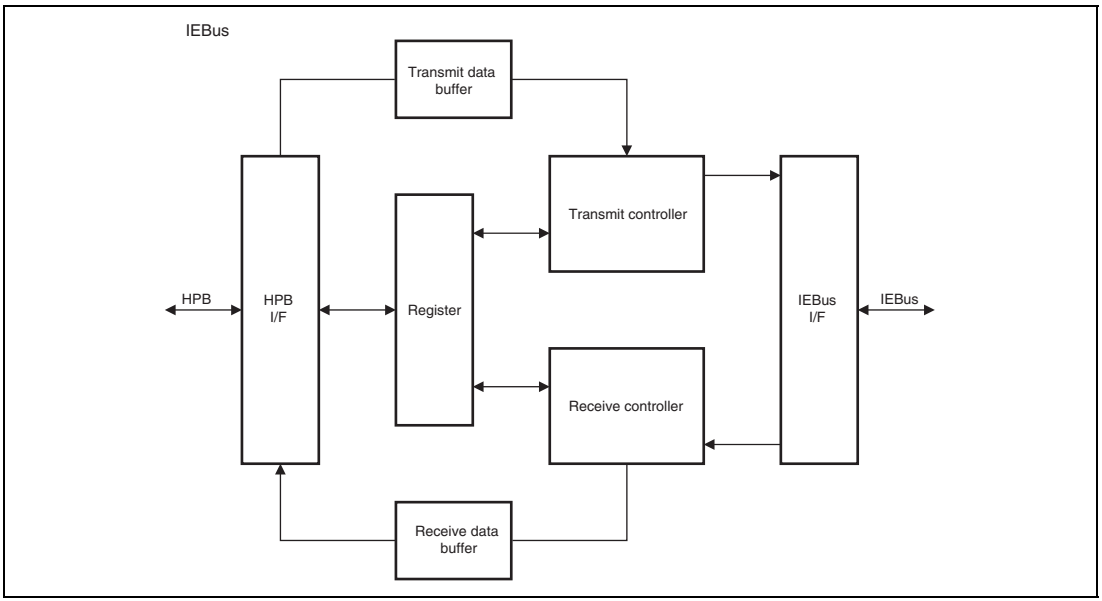


Figure 4.1 Block Diagram of IEBus

Table 4.2 Functions in Each Block

Block		Function
1	HPB I/F	HPB interface conversion function
		• Data width: 8 bits • IEBus register access
2	IEBus I/F	Interface function conforming to the IEBus specifications
		• Transmission of data from the transmit controller in the IEBus bit format via IEBus • Extraction and transfer of frame data from the IEBus bit format to the receive controller.
3	Register	IEBus control registers
		• Registers for controlling the IEBus • Readable and writable via the HPB
4	Transmit controller	Transmits data from transmit buffer to IEBus
		• Generation and transmission of transmit frames based on header information from registers and data from transmit buffer. • Detection of a transmit error.
5	Receive controller	Stores data from IEBus in receive buffer
		• Storing of header information of received frame in register, and data in receive buffer. • Detection of a receive error.
6	Transmit data buffer	Data transmit buffer
		• Buffer for storing data to be transmitted to IEBus. • Buffer size: 32 bytes
7	Receive data buffer	Data receive buffer
		• Buffer for storing data received from IEBus. • Buffer size: 32 bytes

4.1.3 Input/Output Pins

Table 4.3 shows the IEBus pin configuration.

Table 4.3 Pin Configuration

Pin Name	I/O	Function
IECLK	Input	IEBus clock signal
IERX	Input	IEBus data input signal
IETX	Output	IEBus data output signal

4.1.4 Register Configuration

The base address of the IEBus registers is as follows;
H'FFFC 9000

Do not write to any addresses other than listed below. Operations cannot be guaranteed if writing is attempted. Values read from addresses other than those listed below are undefined.

Table 4.4 Register Configuration

Name	Abbreviation	R/W	Offset Address from Base Address	Size
IEBus control register	IECTR	R/W	H'000	8
IEBus command register	IECMR	R/W	H'001	8
IEBus master control register	IEMCR	R/W	H'002	8
IEBus local address register 1	IEAR1	R/W	H'003	8
IEBus local address register 2	IEAR2	R/W	H'004	8
IEBus slave address setting register 1	IESA1	R/W	H'005	8
IEBus slave address setting register 2	IESA2	R/W	H'006	8
IEBus transmit telegram length register	IETBFL	R/W	H'007	8
—	IETBR	R	H'008	8
IEBus receive master address register 1	IEMA1	R	H'009	8
IEBus receive master address register 2	IEMA2	R	H'00A	8
IEBus receive control field register	IERCTL	R	H'00B	8
IEBus receive telegram length register	IERBFL	R	H'00C	8
—	IERBR	R	H'00D	8
—	IELA1	R	H'00E	8
—	IELA2	R	H'00F	8
IEBus general flag register	IEFLG	R	H'010	8
IEBus transmit status register	IETSR	R	H'011	8
IEBus transmit interrupt enable register	IEIET	R/W	H'012	8

Name	Abbreviation	R/W	Offset Address from Base Address	Size
—	IETEF	R	H'013	8
IEBus receive status register	IERSR	R	H'014	8
IEBus receive interrupt enable register	IEIER	R/W	H'015	8
—	IEREF	R	H'016	8
IEBus transmit data buffer	IETB01 to IETB32	*1	H'100 to H'11F	8
IEBus receive data buffer	IERB01 to IERB32	*2	H'200 to H'21F	8

Notes: 1. During master transmission (when MRQ in IEFLG = 1), the IEBus transmit data buffer cannot be read (the read value is undefined) and cannot be written to.

2. During slave reception (when SRE in IEFLG = 1 and RXBSY in IERSR = 0), the IEBus receive data buffer cannot be read.

Table 4.5 Register States in Each Operating Mode

Abbr.	Power-on Reset	Manual Reset	Sleep	Software Standby	Module Standby	Deep Standby
IECTR	Initialized	Initialized	Retained	Retained	Retained	Initialized
IECMR	Initialized	Initialized	Retained	Retained	Retained	Initialized
IEMCR	Initialized	Initialized	Retained	Retained	Retained	Initialized
IEAR1	Initialized	Initialized	Retained	Retained	Retained	Initialized
IEAR2	Initialized	Initialized	Retained	Retained	Retained	Initialized
IESA1	Initialized	Initialized	Retained	Retained	Retained	Initialized
IESA2	Initialized	Initialized	Retained	Retained	Retained	Initialized
IETBFL	Initialized	Initialized	Retained	Retained	Retained	Initialized
IETBR	Initialized	Initialized	Retained	Retained	Retained	Initialized
IEMA1	Initialized	Initialized	Retained	Retained	Retained	Initialized
IEMA2	Initialized	Initialized	Retained	Retained	Retained	Initialized
IERCTL	Initialized	Initialized	Retained	Retained	Retained	Initialized
IERBFL	Initialized	Initialized	Retained	Retained	Retained	Initialized
IERBR	Initialized	Initialized	Retained	Retained	Retained	Initialized
IELA1	Initialized	Initialized	Retained	Retained	Retained	Initialized
IELA2	Initialized	Initialized	Retained	Retained	Retained	Initialized
IEFLG	Initialized	Initialized	Retained	Retained	Retained	Initialized
IETSR	Initialized	Initialized	Retained	Retained	Retained	Initialized
IEIET	Initialized	Initialized	Retained	Retained	Retained	Initialized
IETEF	Initialized	Initialized	Retained	Retained	Retained	Initialized
IERSR	Initialized	Initialized	Retained	Retained	Retained	Initialized
IEIER	Initialized	Initialized	Retained	Retained	Retained	Initialized
IEREF	Initialized	Initialized	Retained	Retained	Retained	Initialized
IETB01 to IETB32	Initialized	Initialized	Retained	Retained	Retained	Initialized
IERB01 to IERB32	Initialized	Initialized	Retained	Retained	Retained	Initialized

4.2 Register Descriptions

Legend:

Initial value: Register value after a reset

—: Undefined value

R/W: Readable/writable. The written value can be read.

R/WC0: Readable/writable. Writing 0 initializes the bit. Writing 1 is ignored.

R: Read-only. The write value should always be 0.

—/W: Write-only. The read value is undefined.

4.2.1 IEBus Control Register (IECTR)

IECTR controls IEBus operation.

Bit:	7	6	5	4	3	2	1	0
	—	IOL	DEE	—	RE	—	—	—
Initial value:	—	0	0	—	0	—	—	—
R/W:	R	R/W	R/W	R	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	R	Reserved This bit is always read as an undefined value. The write value should always be 0.
6	IOL	0	R/W	Input/Output Level Selects the input and output level (positive or negative logic) for the IERX and IETX pins. 0: Negative logic (logical 1: low, logical 0: high) 1: Positive logic (logical 1: high, logical 0: low)
5	DEE	0	R/W	Broadcast Receive Error Interrupt Enable Setting this bit to 1 generates a broadcast receive error interrupt when the receive buffer is not ready (when the RE bit in IECTR is not set to 1 or the RXBSY flag in IERSR is set) during broadcast reception of the control field. At this time, the master address is stored in the IEBus receive master address registers 1 and 2, instead of the receive control field register. If this bit is 0, a broadcast receive error interrupt is not generated, reception is aborted, and the standby state is entered when the receive buffer is not ready during broadcast reception of the control field. The master address is not stored. 0: Does not generate a broadcast receive error interrupt to the control field. 1: Generates a broadcast receive error interrupt to the control field.
4	—	—	R	Reserved This bit is always read as an undefined value. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
3	RE	0	R/W	Receive Enable Enables or disables the start of IEBus reception. This bit should be set with initial setting before frame reception. Modification before reception of the control field is valid. However, modification after reception of control field is invalid, and the value before modification is valid. 0: Disables reception. 1: Enables reception.
2 to 0	—	—	R	Reserved These bits are always read as an undefined value. The write value should always be 0.

4.2.2 IEBus Command Register (IECMR)

IECMR issues a command for controlling IEBus communication.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	CMD		
Initial value:	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	-/W	-/W	-/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	—	R	Reserved These bits are always read as an undefined value. The write value should always be 0.
2 to 0	CMD	—	—/W	Command Bit Issues a command for controlling IEBus communication. While the CMX bit in IEFLG is set after the command issuance, command execution is in progress. When the CMX bit is cleared to 0, the operating state is entered. H'0: No operation H'1: Reserved*[1] H'2: Requests master communication. H'3: Aborts master communication. *[2] H'4: Undefined*[3] H'5: Reserved*[1] H'6: Reserved*[1] H'7: Undefined*[3]

Notes: 1. Setting prohibited.

2. This command is valid only during master communication (MRQ in IEFLG = 1). In the other times, command issuance is ignored. If this command is issued during master communication, the communication controller immediately enters the standby state. At the same time, a master communication request which has been issued is ended (MRQ in IEFLG = 0).
3. Undefined bit. Issuance of this command does not affect operation.

4.2.3 IEBus Master Control Register (IEMCR)

IEMCR sets the master communication conditions.

Bit:	7	6	5	4	3	2	1	0
	SS	RN			CTL			
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

Bit	Bit Name	Initial Value	R/W	Description
7	SS	0	R/W	Broadcast/Normal Communication Select Selects broadcast or normal communication in master communication. 0: Broadcast communication 1: Normal communication
6 to 4	RN	000	R/W	Number of Retransmission Times Sets the number of automatic retransmission times when arbitration is lost during master communication. If arbitration is lost for the specified number of times, the TXEAL bit in the IEBus transmit status register (IETSR) is set, a transmit error occurs, and transmission is ended. If arbitration is won during retransmission, the value of retransmission times is automatically restored to the initial value after the master address is transmitted. H'0: Zero H'1: Once H'2: Two times H'3: Three times H'4: Four times H'5: Five times H'6: Six times H'7: Seven times

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	CTL	0000	R/W	Control Bit Sets the control bit in the control field in master communication. H'0: Reserved* H'1: Undefined* H'2: Undefined* H'3: Reserved* H'4: Reserved* H'5: Reserved* H'6: Reserved* H'7: Reserved* H'8: Undefined* H'9: Undefined* H'A: Reserved* H'B: Reserved* H'C: Reserved* H'D: Undefined* H'E: Undefined* H'F: Data write

Note: * Setting prohibited.

4.2.4 IEBus Local Address Register 1 (IEAR1)

IEAR1 sets the lower 4 bits of the local address and communication mode.

Bit:	7	6	5	4	3	2	1	0
	IARL4				IMD		—	—
Initial value:	0	0	0	0	0	0	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	IARL4	0000	R/W	IEBus Local Address Lower 4 Bits Sets the lower 4 bits of the local address. During master communication, the local address is a master address field value. During slave communication, the local address is compared with the received slave address.
3, 2	IMD	00	R/W	IEBus Communication Mode Selects the IEBus communication mode. H'0: Communication mode 0 H'1: Communication mode 1 H'2: Reserved* H'3: Undefined*
1, 0	—	—	R	Reserved These bits are always read as an undefined value. The write value should always be 0.

Note: * Setting prohibited.

4.2.5 IEBus Local Address Register 2 (IEAR2)

IEAR2 sets the upper 8 bits of the local address.

Bit:	7	6	5	4	3	2	1	0
	IARU8							
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

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	IARU8	All 0	R/W	IEBus Local Address Upper 8 Bits Sets the upper 8 bits of the local address. During master communication, the local address is a master address field value. During slave communication, the local address is compared with the received slave address.

4.2.6 IEBus Slave Address Setting Register 1 (IESA1)

IESA1 sets the lower 4-bit address of the communication destination slave unit.

Bit:	7	6	5	4	3	2	1	0
	ISAL4				—	—	—	—
Initial value:	0	0	0	0	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	ISAL4	0000	R/W	IEBus Slave Address Lower 4 Bits Sets the lower 4-bit address of the communication destination slave unit.
3 to 0	—	—	R	Reserved These bits are always read as an undefined value. The write value should always be 0.

4.2.7 IEBus Slave Address Setting Register 2 (IESA2)

IESA2 sets the upper 8-bit address of the communication destination slave unit.

Bit:	7	6	5	4	3	2	1	0
	ISAU8							
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

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	ISAU8	All 0	R/W	IEBus Slave Address Upper 8 Bits Sets the upper 8-bit address of the communication destination slave unit.

4.2.8 IEBus Transmit Telegram Length Register (IETBFL)

IETBFL sets the telegram length for master transmission.

Bit:	7	6	5	4	3	2	1	0
	TBFL							
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

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	TBFL	All 0	R/W	Transmit Telegram Length Bit Sets the telegram length for master transmission. The set value should be equal to or smaller than the maximum number of bytes to be transferred in the communication mode. H'01: 1 byte H'02: 2 bytes : H'1F: 31 bytes H'20: 32 bytes H'21: Reserved* : H'FF: Reserved* H'00: Reserved*

Note: * Setting prohibited.

4.2.9 IETBR

All bits of IETBR are reserved.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	—	R	Reserved These bits are always read as an undefined value. The write value should always be 0.

4.2.10 IEBus Receive Master Address Register 1 (IEMA1)

IEMA1 indicates the lower 4-bit address of the communication destination master unit in slave or broadcast reception.

Bit:	7	6	5	4	3	2	1	0
	IMAL4				—	—	—	—
Initial value:	0	0	0	0	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	IMAL4	0000	R	IEBus Receive Master Address Lower 4 Bits Indicates the lower 4-bit address of the communication destination master unit in slave or broadcast reception. IEMA1 is valid when slave or broadcast reception is started (contents are rewritten when the RXS bit in IERSR is set). When a broadcast receive error interrupt is enabled by the DEE bit in IECTR, if the receive buffer is not ready when the control field is received, a receive error interrupt is generated, and the lower 4 bits of the master address are stored in the IEBus receive master address register 1 (IEMA1).
3 to 0	—	—	R	Reserved These bits are always read as an undefined value. The write value should always be 0.

4.2.11 IEBus Receive Master Address Register 2 (IEMA2)

IEMA2 indicates the upper 8-bit address of the communication destination master unit in slave or broadcast reception.

Bit:	7	6	5	4	3	2	1	0
	IMAU8							
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 to 0	IMAU8	All 0	R	IEBus Receive Master Address Upper 8 Bits Indicates the upper 8-bit address of the communication destination master unit in slave or broadcast reception. IEMA2 is valid when slave or broadcast reception is started (contents are rewritten when the RXS bit in IERSR is set). When a broadcast receive error interrupt is enabled by the DEE bit in IECTR, if the receive buffer is not ready when the control field is received, a receive error interrupt is generated, and the upper 8 bits of the master address are stored in the IEBus receive master address register 2 (IEMA2).

4.2.12 IEBus Receive Control Field Register (IERCTL)

IERCTL indicates the control field value in slave or broadcast reception.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	RCTL			
Initial value:	—	—	—	—	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	—	R	Reserved These bits are always read as an undefined value. The write value should always be 0.
3 to 0	RCTL	0000	R	IEBus Receive Control Field Indicates the control field value in slave or broadcast reception. IERCTL is valid when slave or broadcast reception is started (contents are rewritten when the RXS bit in IERSR is set).

4.2.13 IEBus Receive Telegram Length Register (IERBFL)

IERBFL indicates the telegram length field value in slave or broadcast reception.

Bit:	7	6	5	4	3	2	1	0
	RBFL							
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 to 0	RBFL	All 0	R	IEBus Receive Telegram Length Indicates the telegram length field value in slave or broadcast reception. IERBFL is valid when slave or broadcast reception is started (contents are rewritten when the RXS bit in IERSR is set).

4.2.14 IERBR

All bits of IERBR are reserved.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	—	R	Reserved These bits are always read as an undefined value. The write value should always be 0.

4.2.15 IELA1

All bits of IELA1 are reserved.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	—	R	Reserved These bits are always read as an undefined value. The write value should always be 0.

4.2.16 IELA2

All bits of IELA2 are reserved.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	—	R	Reserved These bits are always read as an undefined value. The write value should always be 0.

4.2.17 IEBus General Flag Register (IEFLG)

IEFLG indicates the IEBus command execution state and detects slave address match and broadcast reception.

Bit:	7	6	5	4	3	2	1	0
	CMX	MRQ	—	SRE	—	—	RSS	GG
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	CMX	0	R	Command Execution State Indicates the command execution state. Setting condition: • A master communication request command is issued while either the MRQ or SRE bit in IEFLG is set. Clearing condition: • A command execution is completed. 0: Command execution has been completed. 1: Command execution is in progress.
6	MRQ	0	R	Master Communication Request Indicates whether or not a master communication request command is being issued. Setting condition: • A master communication request command is issued and the CMX bit in IEFLG is cleared to 0. Clearing condition: • A master communication is completed. 0: A master communication request command is not being issued. 1: A master communication request command is being issued.
5	—	—	R	Reserved This bit is always read as an undefined value. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
4	SRE	0	R	Slave reception state Indicates the slave or broadcast reception execution state. Setting condition: • Slave or broadcast reception is started while the RE bit in IECTR is 1. Clearing condition: • Slave or broadcast reception is completed. 0: Slave or broadcast reception is not performed. 1: Slave or broadcast reception is in progress.
3, 2	—	—	R	Reserved These bits are always read as an undefined value. The write value should always be 0.
1	RSS	0	R	Received Broadcast Bit Indicates the received broadcast bit value. IEFLG is valid when slave or broadcast reception is started (contents are rewritten when the RXS bit in IERSR is set). The previous value is retained until slave or broadcast reception is started. 0: Received broadcast bit is 0. 1: Received broadcast bit is 1.
0	GG	0	R	Simultaneous Broadcast Reception Acknowledge This bit is set when H'FFF is acknowledged in slave address field in broadcast reception. As with the received broadcast bit (RSS), IEFLG is valid when slave or broadcast reception is started (contents are rewritten when the RXS bit in IERSR is set). The previous value is retained until slave or broadcast reception is started. For normal slave reception, this bit is cleared to 0. 0: (1) Slave normal reception (2) H'FFF has not been acknowledged in the slave address field in broadcast reception. 1: H'FFF is acknowledged in the slave address field in broadcast reception.

4.2.18 IEBus Transmit Status Register (IETSR)

IETSR detects interrupt sources such as transmit start, transmit successful end, and transmit erroneous end. Each IETSR source corresponds to the respective bits in the IEBus transmit interrupt enable register (IEIET) and enables or disables the pertinent interrupts.

IETSR is cleared to 0 by writing 1 to each bit.

Bit:	7	6	5	4	3	2	1	0
	—	TXS	TXF	—	TXEAL	TXET TME	TXERO	TXEACK
Initial value:	—	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	—	—	R	Reserved This bit is always read as an undefined value. The write value should always be 0.
6	TXS	0	R	Transmit Start Indicates that the IEBus has started transmission. Setting condition: • The master address field is entirely transmitted after winning arbitration in master transmission.
5	TXF	0	R	Transmit Successful End Indicates that the number of bytes specified by the telegram length bits has been transmitted thus completing transmission successfully. Setting condition: • The number of bytes specified by the telegram length bits is transmitted.
4	—	—	R	Reserved This bit is always read as an undefined value. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
3	TXEAL	0	R	Arbitration Lost When arbitration is lost in master communication, the IEBus retransmits data from the first bit for the number of times specified by RN bits in IEMCR. If arbitration is lost for the specified number of times, this bit is set, and the standby state is entered. If arbitration is won during the specified number of retransmission, this bit is not set to 1. This bit is set only when arbitration is lost and the standby state is entered. Setting condition: • Arbitration is lost in data transmission, and transmission ends.
2	TXETTME	0	R	Transmit Timing Error When data is not transferred at the timing specified by the IEBus protocol during data transmission, this bit is set and the IEBus enters the standby state. Setting condition: • A timing error is generated during data transmission.
1	TXERO	0	R	Maximum Number of Transmit Frame Bytes to be Transferred Over Indicates that transmission has not successfully ended either because the maximum number of bytes defined for the communication mode were transmitted as a result of data retransmission triggered by NAK reception from the receive unit during data transmission, or because the telegram length was greater than the maximum number of bytes to be transferred and the IEBus enters the standby state. Setting condition: • Transmission did not end successfully despite the maximum number of bytes defined for the communication mode having been transmitted.

Bit	Bit Name	Initial Value	R/W	Description
0	TXEACK	0	R	Acknowledge Bit Indicates data received in the acknowledge bit in the data field. (1) Acknowledge bit in any field other than data field When NAK is received, communication is aborted, and the standby state is entered. In this case, this bit is set to 1. (2) Acknowledge bit in data field When NAK is received from a receive unit in data field transmission, up to the maximum number of bytes defined in communication mode are retransmitted until ACK is received from the receive unit. In this case, if ACK is received from the receive unit during transmission, this bit is not set and transmission is continued. If ACK is not received and communication ends, this bit is set to 1. Note: This flag is invalid for broadcast communication. Setting condition: A 1 (NAK) is detected in the acknowledge bit.

4.2.19 IEBus Transmit Interrupt Enable Register (IEIET)

IEIET enables and disables interrupts such as transmit start, transmit successful end, and transmit erroneous end of IETSR.

Bit:	7	6	5	4	3	2	1	0
	—	TXSE	TXFE	—	TXEALE	TXETTMEE	TXEROE	TXEACKE
Initial value:	—	0	0	—	0	0	0	0
R/W:	R	R/W	R/W	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	R	Reserved This bit is always read as an undefined value. The write value should always be 0.
6	TXSE	0	R/W	Transmit Start Interrupt Enable Enables or disables a transmit start (TXS) interrupt. 0: Disables a transmit start (TXS) interrupt. 1: Enables a transmit start (TXS) interrupt.
5	TXFE	0	R/W	Transmit Successful End Interrupt Enable Enables or disables a transmit successful end (TXF) interrupt. 0: Disables a transmit successful end (TXF) interrupt. 1: Enables a transmit successful end (TXF) interrupt.
4	—	—	R	Reserved This bit is always read as an undefined value. The write value should always be 0.
3	TXEALE	0	R/W	Arbitration Lost Interrupt Enable Enables or disables an arbitration lost (TXEAL) interrupt. 0: Disables an arbitration lost (TXEAL) interrupt. 1: Enables an arbitration lost (TXEAL) interrupt.

Bit	Bit Name	Initial Value	R/W	Description
2	TXETTMEE	0	R/W	Transmit Timing Error Interrupt Enable Enables or disables a transmit timing error (TXETTME) interrupt. 0: Disables a transmit timing error (TXETTME) interrupt. 1: Enables a transmit timing error (TXETTME) interrupt.
1	TXEROE	0	R/W	Maximum Number of Transmit Frame Bytes to be Transferred Over Interrupt Enable Enables or disables a maximum number of transmit frame bytes to be transferred over (TXERO) interrupt. 0: Disables a maximum number of transmit frame bytes to be transferred over (TXERO) interrupt. 1: Enables a maximum number of transmit frame bytes to be transferred over (TXERO) interrupt.
0	TXEACKE	0	R/W	Acknowledge Bit Interrupt Enable Enables or disables an acknowledge bit (TXEACK) interrupt. 0: Disables an acknowledge bit (TXEACK) interrupt. 1: Enables an acknowledge bit (TXEACK) interrupt.

4.2.20 IETEF

All bits of IETEF are reserved.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	—	R	Reserved These bits are always read as an undefined value. The write value should always be 0.

4.2.21 IEBus Receive Status Register (IERSR)

IERSR detects interrupt sources such as receive busy, receive start, receive successful end, and receive erroneous end. Each IERSR source corresponds to the respective bits in the IEBus receive interrupt enable register (IEIER) and enables or disables the pertinent interrupts.

IERSR is cleared to 0 by writing 1 to each bit.

Bit:	7	6	5	4	3	2	1	0
	RXBSY	RXS	RXF	RXEDE	RXEOVE	RXERTME	RXEDLE	RXEPE
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	RXBSY	0	R	Receive Busy Indicates that the receive data buffer (IERB01 to IERB32) has held the received data. This bit should be cleared after all the received data have been read. The subsequent data cannot be received while this bit is set. Setting condition: • All the received data are written to the receive data buffer.

Bit	Bit Name	Initial Value	R/W	Description
6	RXS	0	R	Receive Start Indicates that the IEBus has started reception. Setting condition: • In slave reception, up to the telegram length field is successfully received from the master unit.
5	RXF	0	R	Receive Successful End Indicates that the number of bytes specified by the telegram length bits has been received thus completing reception successfully. Setting condition: • The number of bytes specified by the telegram length bits has been received.
4	RXEDE	0	R	Broadcast Receive Error Indicates that data cannot be received because the receive buffer is not ready (when the RE bit is not set to 1 or the RXBSY flag is set) when the control field is received in broadcast reception. This bit is effective when the DEE bit in IECTR is 1. Setting condition: • Data cannot be received in broadcast reception.

Bit	Bit Name	Initial Value	R/W	Description
3	RXEOVE	0	R	Overrun Control Flag This flag controls overrun during data reception. The IEBus sets the RXEOVE flag when the IEBus receives a data byte and starts receiving its parity bit while the RXBSY flag is not cleared, that is, the previously received data has not been read. If the OVE bit remains set until transmission of the acknowledge bit, the IEBus determines that an overrun error has occurred and returns NAK to the communication destination. Then, the communication destination retransmits data until it has transmitted the maximum number of frame bytes to be transferred. However, if the OVE flag remains set, the IEBus determines that an overrun error has not been resolved and continues transmission of NAK. When the OVE flag is cleared, the IEBus determines that an overrun error has been resolved and transmits ACK to fetch subsequent data. For broadcast reception, if the OVE bit is set in acknowledge bit transmission, the standby state is immediately entered. This flag is valid only when the receive start flag (RXS) is set. If an overrun error occurs before the receive start flag is set, communication is aborted and the standby state is entered. At this time, this bit is not set. Setting condition: A data byte is received and reception of the parity of the data is started while the RXBSY flag for the previous data is not cleared.

Bit	Bit Name	Initial Value	R/W	Description
2	RXERTME	0	R	Receive Timing Error Indicates that a timing error has occurred during data reception. When data is not successfully received at the timing specified by the IEBus protocol during data reception, this bit is set, and the IEBus enters the standby state. This flag is valid only when the receive start flag (RXS) is set. When a timing error occurs before the receive start flag is set, communication is aborted and the standby state is entered. At this time, this bit is not set. Setting condition: A timing error occurs during data reception.
1	RXEDLE	0	R	Maximum Number of Receive Frame Bytes to be Transferred Over Indicates that a maximum number of receive frame bytes to be transferred over has occurred. Indicates that reception has not successfully ended either because the maximum number of bytes defined for the communication mode were received as a result of data re-reception due to a parity error or an overrun error during data reception, or because the telegram length was greater than the maximum number of bytes to be transferred. The IEBus sets this bit and the standby state is entered. This flag is valid only when the receive start flag (RXS) is set. When a maximum number of receive frame bytes to be transferred over occurs before the receive start flag is set, communication is aborted and the standby state is entered. At this time, this bit is not set. Setting condition: Reception does not successfully end within the maximum number of bytes defined for the communication mode.

Bit	Bit Name	Initial Value	R/W	Description
0	RXEPE	0	R	Parity Error Indicates that a parity error has occurred during data field reception. If a parity error occurs before data field reception, the IEBus immediately enters the standby state. RXEPE is not set. If a parity error occurs during data field reception before the maximum number of frame bytes for transfer has yet been received, RXEPE is not set yet. If a parity error occurs, the IEBus returns NAK to the communication destination using the acknowledge bit. At this time, the communication destination retransmits data until it has completely transmitted the maximum number of frame bytes to be transferred. However, if a parity error is resolved during re-reception and reception is successfully completed, RXEPE is not set. RXEPE is set if a parity error is not resolved when reception is aborted and the standby state is entered before the number of bytes specified by the telegram length is received. For broadcast reception, if a parity error occurs during data field reception, the IEBus sets RXEPE and immediately enters the standby state. This flag is valid only when the receive start flag (RXS) is set. If a parity error occurs before receive start flag is set, communication is aborted and the standby state is entered. At this time, this bit is not set. Setting condition: After receiving the maximum number of frame bytes to be transferred, a parity bit in the data field of the finally received byte is not an even parity.

4.2.22 IEBus Receive Interrupt Enable Register (IEIER)

IEIER enables or disables interrupts such as receive busy, receive start, receive successful end, and receive erroneous end of IERSR.

Bit:	7	6	5	4	3	2	1	0
	RXBSYE	RXSE	RXFE	RXEDEE	RXEOVEE	RXERTMEE	RXEDLEE	RXEPEE
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

Bit	Bit Name	Initial Value	R/W	Description
7	RXBSYE	0	R/W	Receive Busy Interrupt Enable Enables or disables a receive busy (RXBSY) interrupt. 0: Disables a receive busy (RXBSY) interrupt. 1: Enables a receive busy (RXBSY) interrupt.
6	RXSE	0	R/W	Receive Start Interrupt Enable Enables or disables a receive start (RXS) interrupt. 0: Disables a receive start (RXS) interrupt. 1: Enables a receive start (RXS) interrupt.
5	RXFE	0	R/W	Receive Successful End Interrupt Enable Enables or disables a receive successful end (RXF) interrupt. 0: Disables a receive successful end (RXF) interrupt. 1: Enables a receive successful end (RXF) interrupt.
4	RXEDEE	0	R/W	Broadcast Receive Error Interrupt Enable Enables or disables a broadcast receive error (RXEDE) interrupt. 0: Disables a broadcast receive error (RXEDE) interrupt. 1: Enables a broadcast receive error (RXEDE) interrupt.

Bit	Bit Name	Initial Value	R/W	Description
3	RXEOVEE	0	R/W	Overrun Control Flag Interrupt Enable Enables or disables an overrun control flag (RXEOVE) interrupt. 0: Disables an overrun control flag (RXEOVE) interrupt. 1: Enables an overrun control flag (RXEOVE) interrupt.
2	RXERTMEE	0	R/W	Receive Timing Error Interrupt Enable Enables or disables a receive timing error (RXERTME) interrupt. 0: Disables a receive timing error (RXERTME) interrupt. 1: Enables a receive timing error (RXERTME) interrupt.
1	RXEDLEE	0	R/W	Maximum Number of Receive Frame Bytes to be Transferred Over Interrupt Enable Enables or disables a maximum number of receive frame bytes to be transferred over (RXEDLE) interrupt. 0: Disables a maximum number of receive frame bytes to be transferred over (RXEDLE) interrupt. 1: Enables a maximum number of receive frame bytes to be transferred over (RXEDLE) interrupt.
0	RXEPEE	0	R/W	Parity Error Interrupt Enable Enables or disables a parity error (RXEPE) interrupt. 0: Disables a parity error (RXEPE) interrupt. 1: Enables a parity error (RXEPE) interrupt.

4.2.23 IEREF

All bits of IEREF are reserved.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	—	R	Reserved These bits are always read as an undefined value. The write value should always be 0.

4.2.24 IEBus Transmit Data Buffer (IETB01 to IETB32)

IETB01 to IETB32 are 32-byte (8 bits × 32) buffer registers to which data to be transmitted in master transmission is written.

Bit:	7	6	5	4	3	2	1	0
	TBn							
Initial value:	—	—	—	—	—	—	—	—
R/W:	*	*	*	*	*	*	*	*

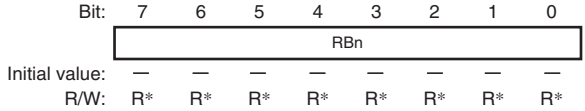
Bit	Bit Name	Initial Value	R/W	Description
7 to 0	TBn	—	*	IEBus Transmit Data Buffer Data to be transmitted in the data field in master transmission is written to TB01 to TB32. The leading data is written to TB01, and data is written to TB02, TB03, …, in order of transmission. For 32-byte transmission, the last data is written to TB32.

Notes: n = 01 to 32

* During master transmission (when MRQ in IEFLG = 1), these bits cannot be read (the read value is undefined), and cannot be written to.

4.2.25 IEBus Receive Data Buffer (IERB01 to IERB32)

IERB01 to IERB32 are 32-byte (8 bits × 32) buffer registers in which received data is stored in slave reception.



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	RBn	—	R*	IEBus Receive Data Buffer RB01 to RB32 can be read when the RXBSY bit in the IEBus receive status register is 1. Data read from RB01 to RB32 is a data field value in slave reception. The leading byte is written to RB01, and data is written to RB02, RB03, …, in order of reception. For 32-byte reception, the last data is written to RB32.

Notes: n = 01 to 32

* During slave reception (when SRE in IEFLG = 1 and RXBSY in IERSR = 0), these bits cannot be read.

4.3 Operation

4.3.1 Interface

Figure 4.2 shows an IEBus communication frame bit format (concept).

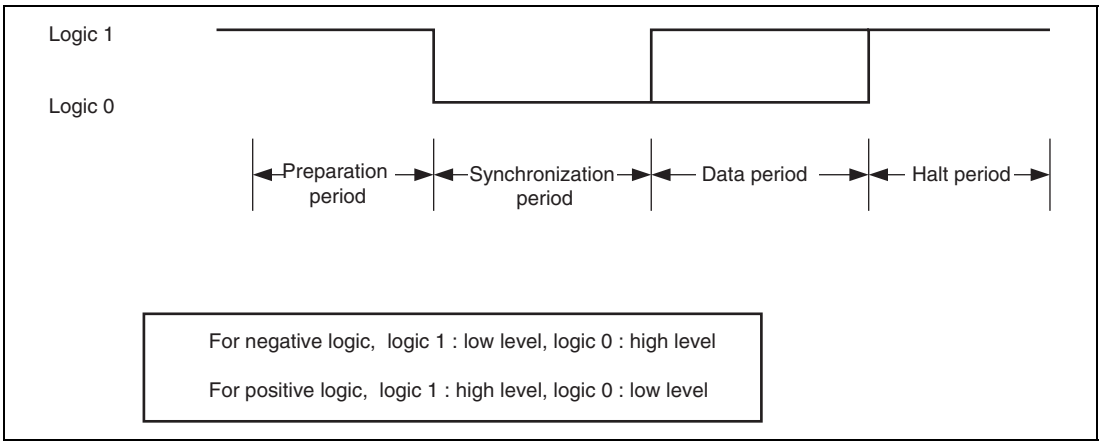


Figure 4.2 IEBus Bit Format (Concept)

Each period of the bit format for positive logic is described below.

Preparation period: First logic 1 period (high)

Synchronization period: Subsequent logic 0 period (low)

Data period: Bit value period (logic 1: high, logic 0: low)

Halt period: Last logic 1 period (high)

For negative logic, the positive logic levels are inverted.

Lengths of synchronization period and the data period are almost equal.

The IEBus is synchronized bit by bit. The lengths of the entire bit and periods in the bit depend on the type of transfer bit and whether the pertinent unit is a master or a slave.

4.3.2 Data Format

(1) Master Transmission

Figure 4.3 shows a relationship between the transfer format in IEBus data transmission and registers.

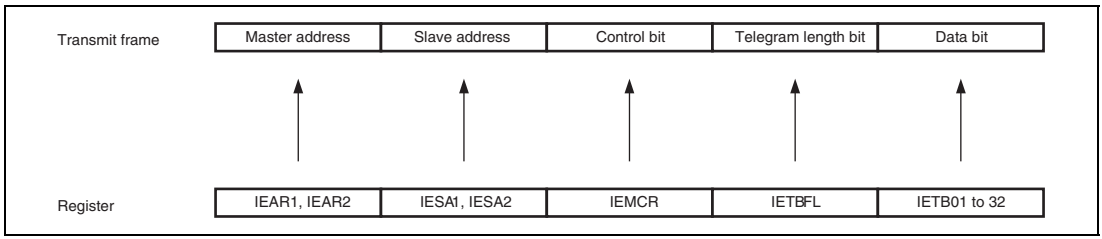


Figure 4.3 Relationship between Transfer Signal Format in Transmission and Registers

(2) Slave Reception

Figure 4.4 shows a relationship between the transfer format in IEBus data reception and registers.

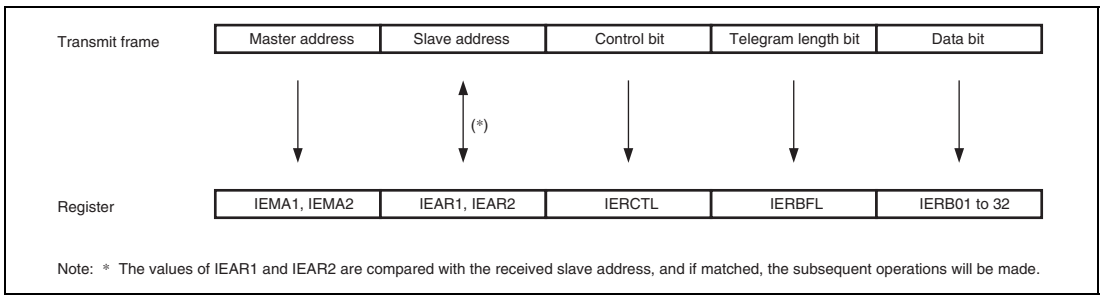


Figure 4.4 Relationship between Transfer Signal Format in Reception and Registers

4.3.3 Software Control Flow

(1) Initial Setting

Figure 4.5 shows an initial setting flow.

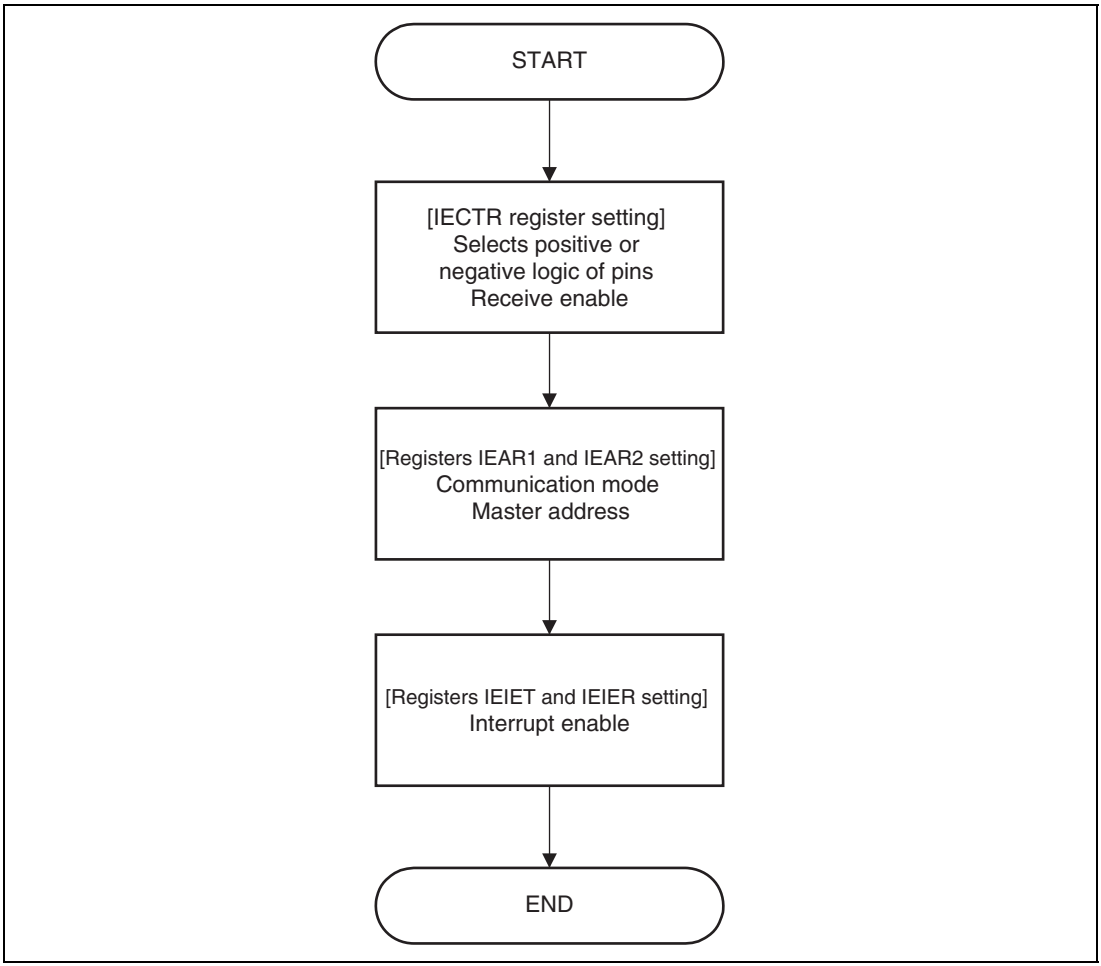


Figure 4.5 Initial Setting Flow

(2) Master Transmission

Figure 4.6 shows a master transmission flow.

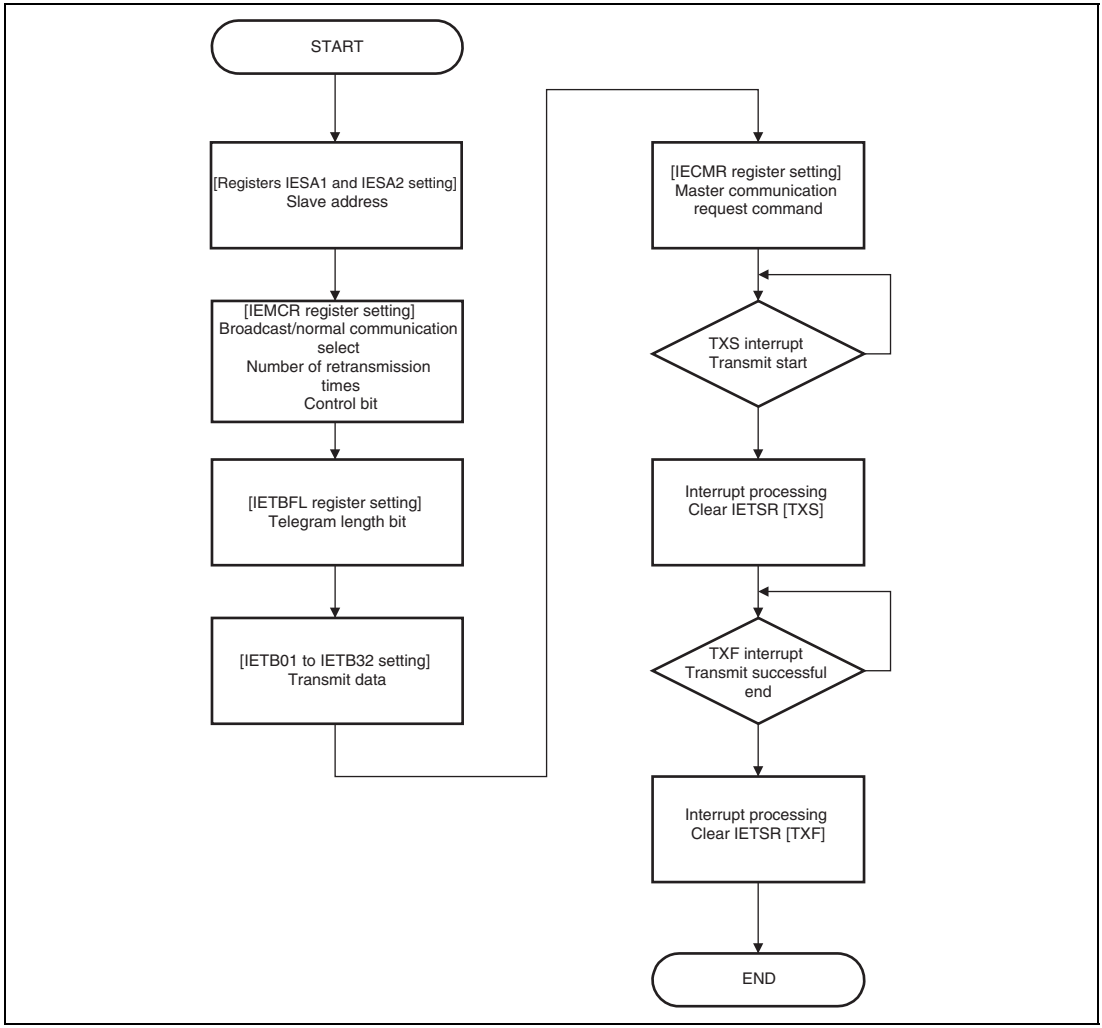


Figure 4.6 Master Transmission Flow (Normal Case)

(3) Slave Reception

Figure 4.7 shows a slave reception flow.

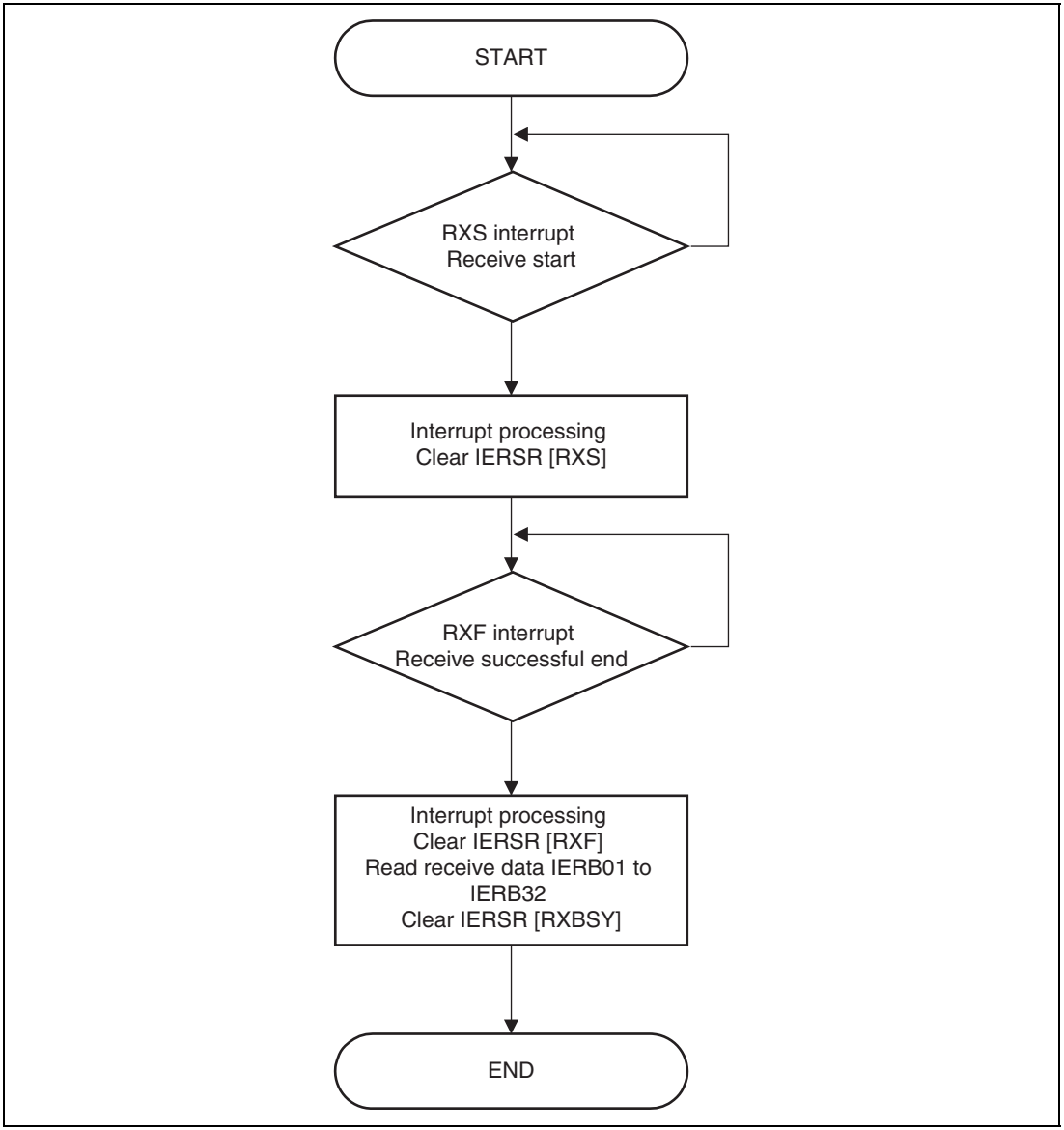


Figure 4.7 Slave Reception Flow (Normal Case)

4.3.4 Operational Timing

(1) Master Transmission

Figure 4.8 shows a master transmission timing.

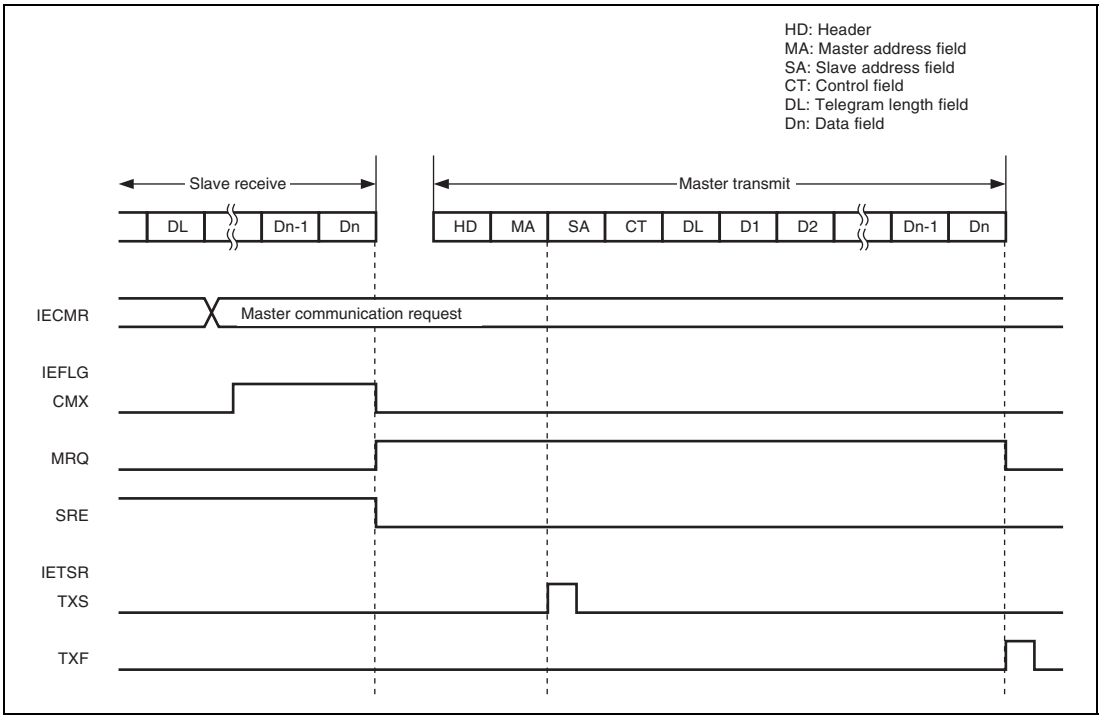


Figure 4.8 Master Transmission Timing

(2) Slave Reception

Figure 4.9 shows a slave reception timing.

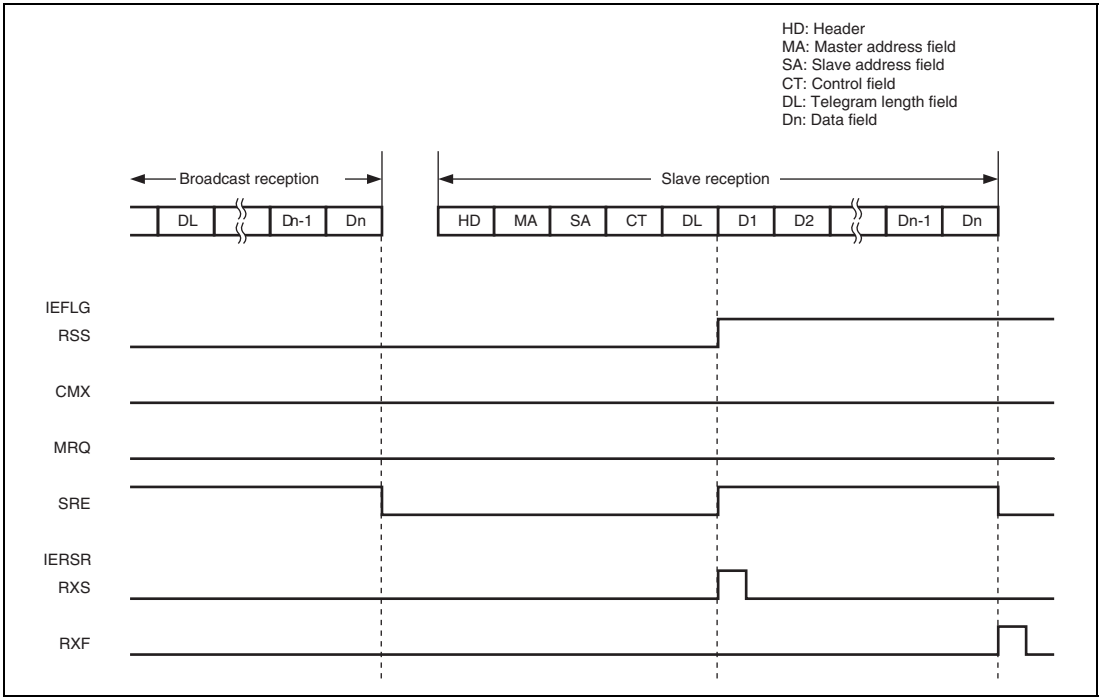


Figure 4.9 Slave Reception Timing

Section 5 Enabling the IEBus™ Controller

The IEBus™ Controller is disabled after a power-on reset. Execute the following procedure to enable the IEBus™ Controller after a power-on reset. The procedure must be executed after each power-on reset.

- Write H'05B5E760 twice consecutively to the address H'FFC8 0094 (in the P4 area) or H'1FC8 0094 (in area 7).

Note: IEBus™ is a trademark of Renesas Electronics Corporation.

Main Revisions and Additions in this Edition

Item	Page	Revision (See Manual for Details)
—	—	First edition issued

SH7734 Additional Document for User's Manual: Hardware

Publication Date: Rev.1.10 Aug 22, 2012

Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.3

SH7734

Renesas Electronics Corporation

R01UH0380EJ0110