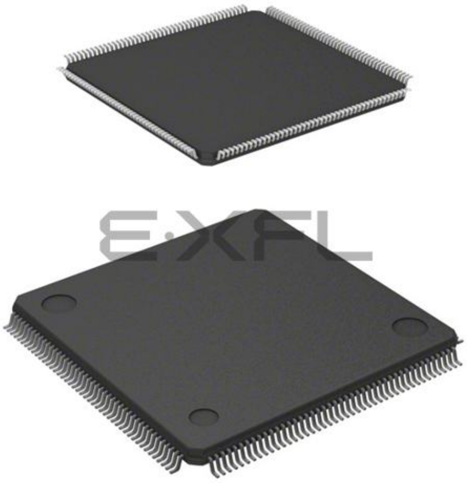
Renesas Electronics America Inc - R5F72145GDFA#V1 Datasheet



What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Not For New Designs
Core Processor	SH2A-FPU
Core Size	32-Bit Single-Core
Speed	100MHz
Connectivity	CANbus, Ethernet, I²C, SCI, SPI, USB
Peripherals	DMA, PWM, WDT
Number of I/O	112
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	64K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 8x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	176-LQFP
Supplier Device Package	176-LFQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/r5f72145gdfa-v1

(11) Absolute Address

When data is accessed by an absolute address, the absolute address value should be placed in the memory table in advance. That value is transferred to the register by loading the immediate data during the execution of the instruction, and the data is accessed in register indirect addressing mode.

With the SH-2A/SH2A-FPU, when data is referenced using an absolute address not exceeding 28 bits, it is also possible to transfer immediate data located in the instruction code to a register and to reference the data in register indirect addressing mode. However, when referencing data using an absolute address of 21 to 28 bits, an OR instruction must be used after the data is transferred to a register.

Table 2.7 Absolute Address Accessing

Classification	SH-2A/SH2A-FPU CPU	Example of Other CPU

Up to 20 bits	`MOVI20 #H'12345,R1`
	`MOV.B @R1,R0`
		`MOV.B @H'12345,R0`

21 to 28 bits	`MOVI20S #H'12345,R1`
	`OR #H'67,R1`
	`MOV.B @R1,R0`
		`MOV.B @H'1234567,R0`

29 bits or more	`MOV.L @(disp,PC),R1`
	`MOV.B @R1,R0`
	....
	`.DATA.L H'12345678`
		`MOV.B @H'12345678,R0`

(12) 16-Bit/32-Bit Displacement

When data is accessed by 16-bit or 32-bit displacement, the displacement value should be placed in the memory table in advance. That value is transferred to the register by loading the immediate data during the execution of the instruction, and the data is accessed in the indexed indirect register addressing mode.

5.6 Exceptions Triggered by Instructions

5.6.1 Types of Exceptions Triggered by Instructions

Exception handling can be triggered by trap instructions, slot illegal instructions, general illegal instructions, integer division exceptions, and floating-point operation instructions, as shown in table 5.9.

Table 5.9 Types of Exceptions Triggered by Instructions

Type	Source Instruction	Comment
Trap instruction	TRAPA	
Slot illegal instructions	Undefined code placed immediately after a delayed branch instruction (delay slot), instructions that rewrite the PC, 32-bit instructions, RESBANK instruction, DIVS instruction, and DIVU instruction	Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF Instructions that rewrite the PC: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF, JSR/N, RTV/N 32-bit instructions: BAND.B, BANDNOT.B, BCLR.B, BLD.B, BLDNOT.B, BOR.B, BORNOT.B, BSET.B, BST.B, BXOR.B, MOV.B@disp12, MOV.W@disp12, FMOV.S@disp12, FMOV.D@disp12, MOV.L@disp12, MOVI20, MOVI20S, MOVU.B, MOVU.W.
General illegal instructions	Undefined code anywhere besides in a delay slot	
Integer division exceptions	Division by zero	DIVU, DIVS
	Negative maximum value ÷ (–1)	DIVS
Floating-point operation instructions	Starts when detecting invalid operation exception defined by IEEE754, division-by-zero exception, overflow, underflow, or inexact exception.	FADD, FSUB, FMUL, FDIV, FMAC, FCMP/EQ, FCMP/GT, FLOAT, FTRC, FCNVDS, FCNVSD, FSQRT

			Interrupt Vector					
Interrupt Source		Number	Vector	Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
MTU2	MTU2_3	TGIA_3	180	H'000002D0 to H'000002D3	0 to 15 (0)	IPR10 (7 to 4)	1	High
		TGIB_3	181	H'000002D4 to H'000002D7			2	
		TGIC_3	182	H'000002D8 to H'000002DB			3	
		TGID_3	183	H'000002DC to H'000002DF			4	
		TCIV_3	184	H'000002E0 to H'000002E3	0 to 15 (0)	IPR10 (3 to 0)	—	
	MTU2_4	TGIA_4	188	H'000002F0 to H'000002F3	0 to 15 (0)	IPR11 (15 to 12)	1	
		TGIB_4	189	H'000002F4 to H'000002F7			2	
		TGIC_4	190	H'000002F8 to H'000002FB			3	
		TGID_4	191	H'000002FC to H'000002FF			4	
		TCIV_4	192	H'00000300 to H'00000303	0 to 15 (0)	IPR11 (11 to 8)	—	
	MTU2_5	TGIU_5	196	H'00000310 to H'00000313	0 to 15 (0)	IPR11 (7 to 4)	1	
		TGIV_5	197	H'00000314 to H'00000317			2	
		TGIW_5	198	H'00000318 to H'0000031B			3	
POE2	OEI1		200	H'00000320 to H'00000323	0 to 15 (0)	IPR11 (3 to 0)	1	
	OEI2		201	H'00000324 to H'00000327			2	Low

7.3.7 Break Address Register_2 (BAR_2)

BAR_2 is a 32-bit readable/writable register. BAR_2 specifies the address used as a break condition in channel 2. The control bits CD2_1 and CD2_0 in the break bus cycle register_2 (BBR_2) select one of the three address buses for a break condition of channel 2. BAR_2 is initialized to H'00000000 by a power-on reset, but retains its previous value by a manual reset or in software standby mode or sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BA2_31	BA2_30	BA2_29	BA2_28	BA2_27	BA2_26	BA2_25	BA2_24	BA2_23	BA2_22	BA2_21	BA2_20	BA2_19	BA2_18	BA2_17	BA2_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
	BA2_15	BA2_14	BA2_13	BA2_12	BA2_11	BA2_10	BA2_9	BA2_8	BA2_7	BA2_6	BA2_5	BA2_4	BA2_3	BA2_2	BA2_1	BA2_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	BA2_31 to BA2_0	All 0	R/W	Break Address 2 Store an address on the CPU address bus (FAB or MAB) or IAB specifying break conditions of channel 2. When the C bus and instruction fetch cycle are selected by BBR_2, specify an FAB address in bits BA2_31 to BA2_0. When the C bus and data access cycle are selected by BBR_2, specify an MAB address in bits BA2_31 to BA0_2.

Note: When setting the instruction fetch cycle as a break condition, clear the LSB in BAR_2 to 0.

7.6 Usage Notes

1. The CPU can read from or write to the UBC registers via the I bus. Accordingly, during the period from executing an instruction to rewrite the UBC register till the new value is actually rewritten, the desired break may not occur. In order to know the timing when the UBC register is changed, read from the last written register. Instructions after then are valid for the newly written register value.
2. The UBC cannot monitor access to the C bus and I bus cycles in the same channel.
3. When a user break and another exception occur at the same instruction, which has higher priority is determined according to the priority levels defined in table 5.1 in section 5, Exception Handling. If an exception with a higher priority occurs, the user break does not occur.
4. Note the following when a break occurs in a delay slot.

   If a pre-execution break is set at a delay slot instruction, the break is not generated until immediately before execution of the branch destination.
5. User breaks are disabled during UBC module standby mode. Do not read from or write to the UBC registers during UBC module standby mode; the values are not guaranteed.
6. Do not set an address within an interrupt exception handling routine whose interrupt priority level is at least 15 (including user break interrupts) as a break address.
7. Do not set break after instruction execution for the SLEEP instruction or for the delayed branch instruction where the SLEEP instruction is placed at its delay slot.
8. When setting a break for a 32-bit instruction, set the address where the upper 16 bits are placed. If the address of the lower 16 bits is set and a break before instruction execution is set as a break condition, the break is handled as a break after instruction execution.
9. Do not set a pre-execution break for an instruction that immediately follows a DIVU or DIVS instruction. If such a break is set and an interrupt or other exception occurs during execution of the DIVU or DIVS instruction, the pre-execution break will still occur even though execution of the DIVU or DIVS instruction is suspended.
10. Do not set a pre- and post-execution break for the same address at the same time. For example, if a pre-execution break for channel 0 and a post -execution break for channel 1 are set for the same address at the same time, the condition match flags on channel 1 after instruction execution will be set even though a pre-execution break has occurred on channel 0.

Table 9.12 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (2)-2

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 Bits)	**01 (12 Bits)**	**10 (10 Bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A27	A17		Unused
A16	A26	A16		
A15	A25*[2]	A25*[2]	A13(BA1)	Specifies bank
A14	A24*[2]	A24*[2]	A12(BA0)	
A13	A23	A13	A11	Address
A12	A22	L/H*[1]	A10/AP	Specifies address/precharge
A11	A21	A11	A9	Address
A10	A20	A10	A8	
A9	A19	A9	A7	
A8	A18	A8	A6	
A7	A17	A7	A5	
A6	A16	A6	A4	
A5	A15	A5	A3	
A4	A14	A4	A2	
A3	A13	A3	A1	
A2	A12	A2	A0	
A1	A11	A1		Unused
A0	A10	A0		
Example of connected memory				
512-Mbit product (4 Mwords × 32 bits × 4 banks, column 10 bits product): 1				
256-Mbit product (4 Mwords × 16 bits × 4 banks, column 10 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.

2. Bank address specification

Table 11.15 TIOR_2 (Channel 2)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_2 Function**	**TIOC2B Pin Function**
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

Bit	Bit Name	Initial Value	R/W	Description
6	TTGE2	0	R/W	A/D Converter Start Request Enable 2 Enables or disables generation of A/D converter start requests by TCNT_4 underflow (trough) in complementary PWM mode. In channels 0 to 3, bit 6 is reserved. It is always read as 0 and the write value should always be 0. 0: A/D converter start request generation by TCNT_4 underflow (trough) disabled 1: A/D converter start request generation by TCNT_4 underflow (trough) enabled
5	TCIEU	0	R/W	Underflow Interrupt Enable Enables or disables interrupt requests (TCIU) by the TCFU flag when the TCFU flag in TSR is set to 1 in channels 1 and 2. In channels 0, 3, and 4, bit 5 is reserved. It is always read as 0 and the write value should always be 0. 0: Interrupt requests (TCIU) by TCFU disabled 1: Interrupt requests (TCIU) by TCFU enabled
4	TCIEV	0	R/W	Overflow Interrupt Enable Enables or disables interrupt requests (TCIV) by the TCFV flag when the TCFV flag in TSR is set to 1. 0: Interrupt requests (TCIV) by TCFV disabled 1: Interrupt requests (TCIV) by TCFV enabled
3	TGIED	0	R/W	TGR Interrupt Enable D Enables or disables interrupt requests (TGID) by the TGFD bit when the TGFD bit in TSR is set to 1 in channels 0, 3, and 4. In channels 1 and 2, bit 3 is reserved. It is always read as 0 and the write value should always be 0. 0: Interrupt requests (TGID) by TGFD bit disabled 1: Interrupt requests (TGID) by TGFD bit enabled

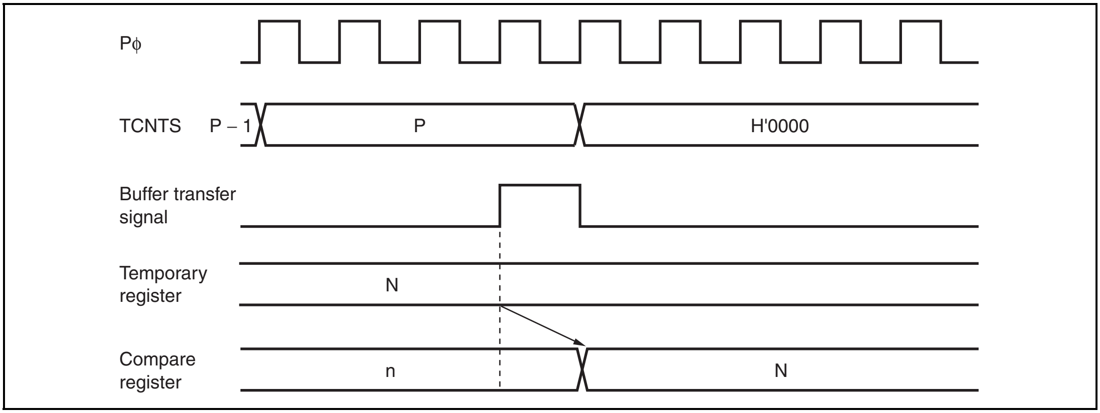


Figure 11.108 Transfer Timing from Temporary Register to Compare Register

11.6.2 Interrupt Signal Timing

(1) TGF Flag Setting Timing in Case of Compare Match

Figures 11.109 and 110 show the timing for setting of the TGF flag in TSR on compare match, and TGI interrupt request signal timing.

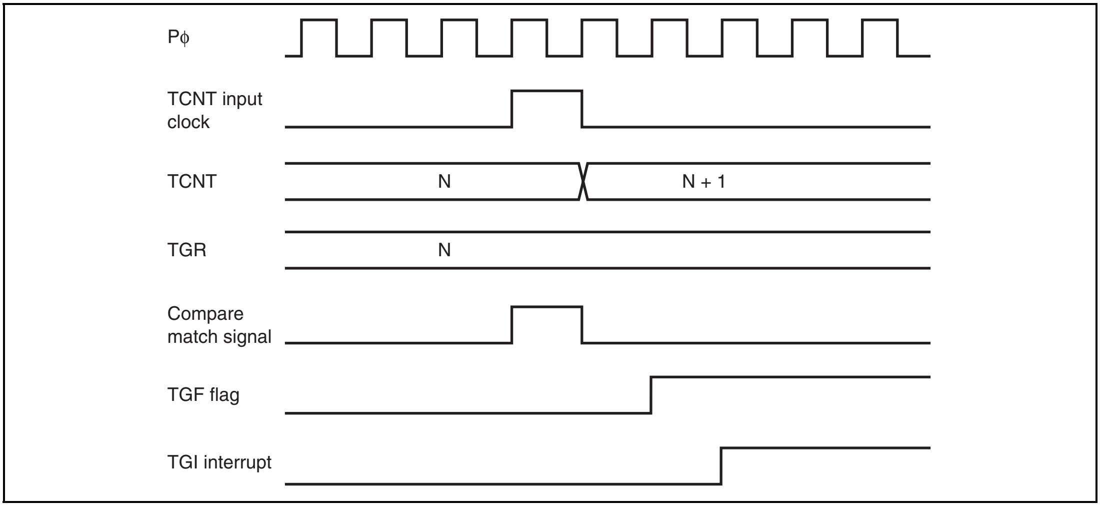


Figure 11.109 TGI Interrupt Timing (Compare Match)

Bit	Bit Name	Initial Value	R/W	Description
14	POE2F	0	R/(W)*[1]	POE2 Flag Indicates that a high impedance request has been input to the $\overline{\text{POE2}}$ pin. [Clearing conditions] • By writing 0 to POE2F after reading POE2F = 1 (when the falling edge is selected by bits 5 and 4 in ICSR1) • By writing 0 to POE2F after reading POE2F = 1 after a high level input to $\overline{\text{POE2}}$ is sampled at $P\phi/8$, $P\phi/16$, or $P\phi/128$ clock (when low-level sampling is selected by bits 5 and 4 in ICSR1) [Setting condition] • When the input set by bits 5 and 4 in ICSR1 occurs at the $\overline{\text{POE2}}$ pin
13	POE1F	0	R/(W)*[1]	POE1 Flag Indicates that a high impedance request has been input to the $\overline{\text{POE1}}$ pin. [Clearing conditions] • By writing 0 to POE1F after reading POE1F = 1 (when the falling edge is selected by bits 3 and 2 in ICSR1) • By writing 0 to POE1F after reading POE1F = 1 after a high level input to $\overline{\text{POE1}}$ is sampled at $P\phi/8$, $P\phi/16$, or $P\phi/128$ clock (when low-level sampling is selected by bits 3 and 2 in ICSR1) [Setting condition] • When the input set by bits 3 and 2 in ICSR1 occurs at the $\overline{\text{POE1}}$ pin

Section 16 Serial Communication Interface (SCI)

This LSI has four channels of independent serial communication interface (SCI). The SCI can handle both asynchronous and clock synchronous serial communication. In asynchronous serial communication mode, serial data communication can be carried out with standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communications Interface Adapter (ACIA). A function is also provided for serial communication between processors (multiprocessor communication function).

16.1 Features

- Choice of asynchronous or clock synchronous serial communication mode
- Asynchronous mode:
  - Serial data communication is performed by start-stop in character units. The SCIF can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communications interface adapter (ACIA), or any other communications chip that employs a standard asynchronous serial system. There are twelve selectable serial data communication formats.
  - Data length: 7 or 8 bits
  - Stop bit length: 1 or 2 bits
  - Parity: Even, odd, or none
  - Multiprocessor communications
  - Receive error detection: Parity, overrun, and framing errors
  - Break detection: Break is detected by reading the RXD pin level directly when a framing error occurs.
- Clock synchronous mode:
  - Serial data communication is synchronized with a clock signal. The SCIF can communicate with other chips having a clock synchronous communication function.
  - Data length: 8 bits
  - Receive error detection: Overrun errors
- Full duplex communication: The transmitting and receiving sections are independent, so the SCI can transmit and receive simultaneously. Both sections use double buffering, so high-speed continuous data transfer is possible in both the transmit and receive directions.
- On-chip baud rate generator with selectable bit rates
- Internal or external transmit/receive clock source: From either baud rate generator (internal clock) or SCK pin (external clock)
- Choice of LSB-first or MSB-first data transfer (except for 7-bit data in asynchronous mode)

17.3.1 Receive Shift Register (SCRSR)

SCRSR receives serial data. Data input at the RXD pin is loaded into SCRSR in the order received, LSB (bit 0) first, converting the data to parallel form. When one byte has been received, it is automatically transferred to the receive FIFO data register (SCFRDR).

The CPU cannot read or write to SCRSR directly.

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

17.3.2 Receive FIFO Data Register (SCFRDR)

SCFRDR is a register that stores serial receive data. The SCIF completes the reception of one byte of serial data by moving the received data from the receive shift register (SCRSR) into SCFRDR for storage. Continuous reception is possible until 16 bytes are stored. The CPU can read but not write to SCFRDR. If data is read when there is no receive data in the SCFRDR, the value is undefined.

When SCFRDR is full of receive data, subsequent serial data is lost.

SCFRDR is initialized to an undefined value by a power-on reset.

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

18.3.2 RSPI Slave Select Polarity Register (SSLP)

SSLP sets the polarity of the SSL0 to SSL7 signals of the RSPI. SSLP can always be read from or written to by the CPU. If the contents of SSLP are changed by the CPU while the RSPI function is enabled by setting the SPE bit in the RSPI control register (SPCR) to 1, subsequent operations cannot be guaranteed.

Bit:	7	6	5	4	3	2	1	0
	–	–	–	–	SSL3P	SSL2P	SSL1P	SSL0P
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 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	SSL3P	0	R/W	SSL Signal Polarity Setting These bits set the polarity of the SSL signals. SSLiP (where i is 3 to 0) indicates the active polarity of the SSLi signal. 0: SSLi signal set to active-0 1: SSLi signal set to active-1
2	SSL2P	0	R/W	
1	SSL1P	0	R/W	
0	SSL0P	0	R/W	

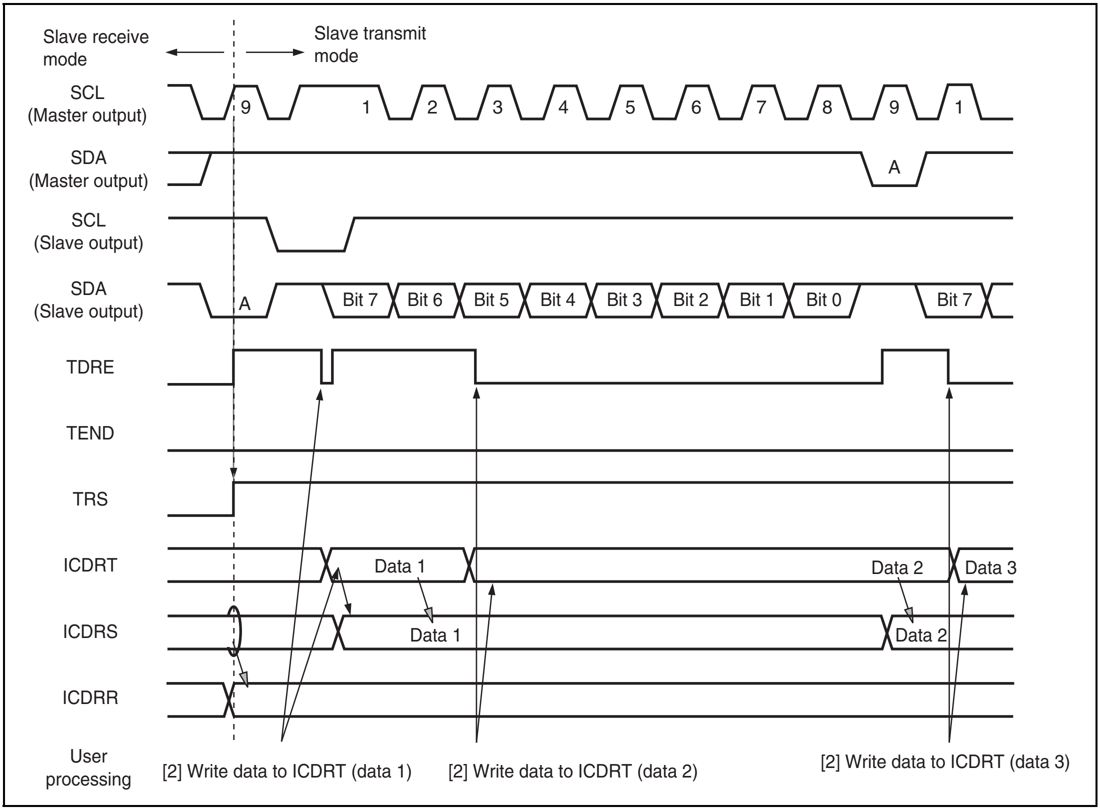


Figure 19.9 Slave Transmit Mode Operation Timing (1)

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PD13MD[2:0]	000*	R/W	PD13 Mode Select the function of the PD13/D13/TIOC4BS pin. 000: PD13 I/O (port) 001: D13 I/O (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: TIOC4BS I/O (MTU2S) 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PD12MD[2:0]	000*	R/W	PD12 Mode Select the function of the PD12/D12/TIOC4AS pin. 000: PD12 I/O (port) 001: D12 I/O (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited) 101: TIOC4AS I/O (MTU2S) 110: Setting prohibited 111: Setting prohibited

Note: * The initial value is 001 during the on-chip ROM disabled external extension mode.

27.10.2 State in which Interrupts are Ignored

In the following mode or period, the AUD is in module standby mode and cannot operate. The NMI or maskable interrupt requests are ignored.

- Boot mode
- The program in the embedded program stored MAT is being executed immediately after the LSI is started in user boot mode

27.10.3 Programming-/Erasure-Suspended Area

The data stored in the programming-suspended or erasure-suspended area is undetermined. To avoid malfunction due to undefined read data, ensure that no instruction is executed or no data is read from the programming-suspended or erasure-suspended area.

To avoid instruction fetch from the programming-suspended or erasure-suspended area, which may be caused by prefetch by the ROM cache, ensure that no instruction is fetched within 16 bytes from the start address of the programming-suspended or erasure-suspended area.

During ROM cache prefetch, the destination of a branch instruction is also accessed. The destination must not be in the programming-suspended or erasure-suspended area.

27.10.4 Compatibility with Programming/Erasing Program of Conventional F-ZTAT SH Microcomputers

The flash memory programming/erasing program used for conventional F-ZTAT SH microcontrollers does not work with this LSI.

27.10.5 FWE Pin State

Ensure that the FWE pin level does not change during programming or erasure. If the FWE level goes low, the current programming or erasure terminates abnormally and the FRDY bit is set to 1 (the erasure or programming error bit in FASTATR0 is set), and then FENTRYR is cleared. To reprogram ROM, do it after erasing data with the FWE pin at the high level.

In a transition from single-chip mode to user program mode, issue an FCU command after driving the FWE pin high, making sure that the FWE bit in FPMON is set to 1, and setting the FENTRYR register.

No.	Pin Name	Type
11	PB12/IRQ2/$\overline{\text{POE1}}$/SCL	INPUT
10	PB13/IRQ3/$\overline{\text{POE2}}$/SDA	INPUT
9	PB14/IRQ6	OUTPUT
8	PB14/IRQ6	CONTROL
7	PB14/IRQ6	INPUT
6	PB15/IRQ7	OUTPUT
5	PB15/IRQ7	CONTROL
4	PB15/IRQ7	INPUT
3	VBUS	OUTPUT
2	VBUS	CONTROL
1	VBUS	INPUT
0	NMI	INPUT
To TDO		

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
DMAC	RSAR_0								
	RDAR_0								
	RDMATCR_0								
	SAR_1								
	DAR1								
	DMATCR_1								
	CHCR_1	TC	—	—	RLD	—	—	—	—
		DO	TL	—	—	HE	HIE	AM	AL
		DM[1:0]		SM[1:0]		RS[3:0]			
		DL	DS	TB	TS[1:0]		IE	TE	DE
	RSAR_1								
	RDAR_1								

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
RCAN-ET	RFPR0	Initialized	Retained	Retained	Initialized	Retained
	MBIMR0	Initialized	Retained	Retained	Initialized	Retained
	UMSR0	Initialized	Retained	Retained	Initialized	Retained
	MB[0]. CONTROL0H	—	Retained	—	—	Retained
	MB[0]. CONTROL0L	—	Retained	—	—	Retained
	MB[0]. LAFMH	—	Retained	—	—	Retained
	MB[0]. LAFML	—	Retained	—	—	Retained
	MB[0]. MSG_DATA[0]	—	Retained	—	—	Retained
	MB[0]. MSG_DATA[1]	—	Retained	—	—	Retained
	MB[0]. MSG_DATA[2]	—	Retained	—	—	Retained
	MB[0]. MSG_DATA[3]	—	Retained	—	—	Retained
	MB[0]. MSG_DATA[4]	—	Retained	—	—	Retained
	MB[0]. MSG_DATA[5]	—	Retained	—	—	Retained
	MB[0]. MSG_DATA[6]	—	Retained	—	—	Retained
	MB[0]. MSG_DATA[7]	—	Retained	—	—	Retained
	MB[0]. CONTROL1H	Initialized	Retained	Retained	Retained	Retained
	MB[0]. CONTROL1L	Initialized	Retained	Retained	Retained	Retained
	MB[1].	Same as MB[0]				
	MB[2].	Same as MB[0]				