Renesas Electronics America Inc - R5F72147GDFP#V1 Datasheet

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Not For New Designs
Core Processor	SH2A-FPU
Core Size	32-Bit Single-Core
Speed	100MHz
Connectivity	CANbus, Ethernet, I²C, SCI, SPI, USB
Peripherals	DMA, PWM, WDT
Number of I/O	112
Program Memory Size	1MB (1M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128K 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 (24x24)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/r5f72147gdfp-v1

Table 4.3 Example of Relationship between Clock Operating Mode and Frequency Range

PLL Multiplication Ratio	FRQCR/MCLKCR/ACLKCR Division Ratio Setting					Clock Ratio					Clock Frequency (MHz)*					
	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ	Input Clock	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ
×16	1/4	1/8	1/8	1/4	1/4	4	2	2	4	4	10	40	20	20	40	40
	1/4	1/4	1/8	1/4	1/4	4	4	2	4	4		40	40	20	40	40
	1/4	1/4	1/4	1/4	1/4	4	4	4	4	4		40	40	40	40	40
	1/2	1/8	1/8	1/4	1/4	8	2	2	4	4		80	20	20	40	40
	1/2	1/8	1/8	1/2	1/4	8	2	2	8	4		80	20	20	80	40
	1/2	1/4	1/8	1/4	1/4	8	4	2	4	4		80	40	20	40	40
	1/2	1/4	1/8	1/2	1/4	8	4	2	8	4		80	40	20	80	40
	1/2	1/4	1/4	1/4	1/4	8	4	4	4	4		80	40	40	40	40
	1/2	1/4	1/4	1/2	1/4	8	4	4	8	4		80	40	40	80	40
	1/1	1/8	1/8	1/4	1/4	16	2	2	4	4		160	20	20	40	40
	1/1	1/8	1/8	1/2	1/4	16	2	2	8	4		160	20	20	80	40
	1/1	1/4	1/8	1/4	1/4	16	4	2	4	4		160	40	20	40	40
	1/1	1/4	1/8	1/2	1/4	16	4	2	8	4		160	40	20	80	40
	1/1	1/4	1/4	1/4	1/4	16	4	4	4	4		160	40	40	40	40
	1/1	1/4	1/4	1/2	1/4	16	4	4	8	4		160	40	40	80	40

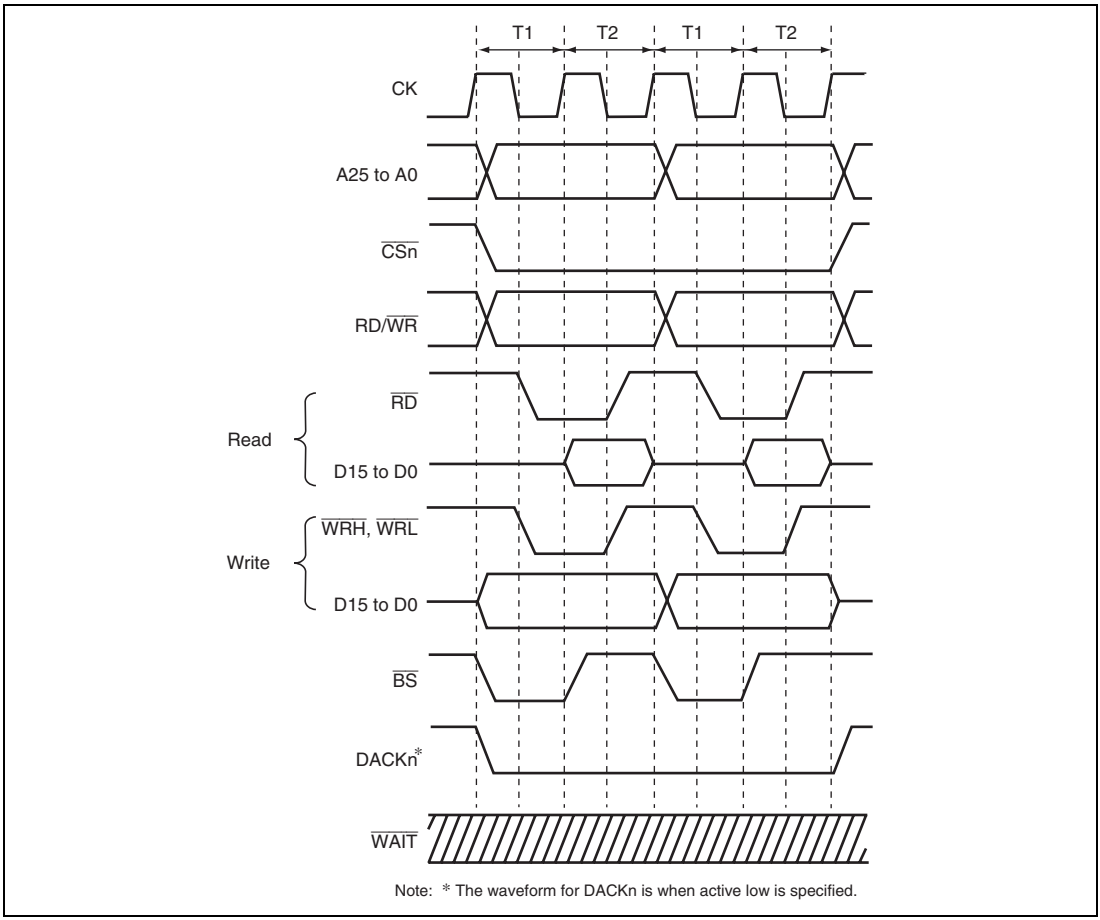

Figure 9.4 Continuous Access for Normal Space 2
Bus Width = 16 Bits, Longword Access, CSnWCR.WM Bit = 1
(Access Wait = 0, Cycle Wait = 0)

9.5.5 MPX-I/O Interface

Access timing for the MPX space is shown below. In the MPX space, $\overline{CS5}$, $\overline{AH}$, $\overline{RD}$, and $\overline{WRxx}$ signals control the accessing. The basic access for the MPX space consists of 2 cycles of address output followed by an access to a normal space. The bus width for the address output cycle or the data input/output cycle is fixed to 8 bits or 16 bits. Alternatively, it can be 8 bits or 16 bits depending on the address to be accessed.

Output of the addresses D15 to D0 or D7 to D0 is performed from cycle Ta2 to cycle Ta3. Because cycle Ta1 has a high-impedance state, collisions of addresses and data can be avoided without inserting idle cycles, even in continuous access cycles. Address output is increased to 3 cycles by setting the MPXW bit in CS5WCR to 1.

The RD/$\overline{WR}$ signal is output at the same time as the $\overline{CS5}$ signal; it is high in the read cycle and low in the write cycle.

The data cycle is the same as that in a normal space access.

The delay cycles the number of which is specified by SW[1:0] are inserted between cycle Ta3 and cycle T1. The delay cycles the number of which is specified by HW[1:0] are added after cycle T2.

Choose to detect DREQ by either the edge or level of the signal input with the DL and DS bits in CHCR_0 to CHCR_3 as shown in table 10.6. The source of the transfer request does not have to be the data transfer source or destination.

Table 10.6 Selecting External Request Detection with DL and DS Bits

CHCR		
DL bit	**DS bit**	**Detection of External Request**
0	0	Low level detection
	1	Falling edge detection
1	0	High level detection
	1	Rising edge detection

When DREQ is accepted, the DREQ pin enters the request accept disabled state (non-sensitive period). After issuing acknowledge DACK signal for the accepted DREQ, the DREQ pin again enters the request accept enabled state.

When DREQ is used by level detection, there are following two cases by the timing to detect the next DREQ after outputting DACK.

Overrun 0: Transfer is terminated after the same number of transfer has been performed as requests.

Overrun 1: Transfer is terminated after transfers have been performed for (the number of requests plus 1) times.

The DO bit in CHCR selects this overrun 0 or overrun 1.

Table 10.7 Selecting External Request Detection with DO Bit

CHCR	
DO bit	**External Request**
0	Overrun 0
1	Overrun 1

10.4.5 Number of Bus Cycles and DREQ Pin Sampling Timing

(1) Number of Bus Cycles

When the DMAC is the bus master, the number of bus cycles is controlled by the bus state controller (BSC) in the same way as when the CPU is the bus master. For details, see section 9, Bus State Controller (BSC).

(2) DREQ Pin Sampling Timing

Figures 10.13 to 10.16 show the DREQ input sampling timings in each bus mode.

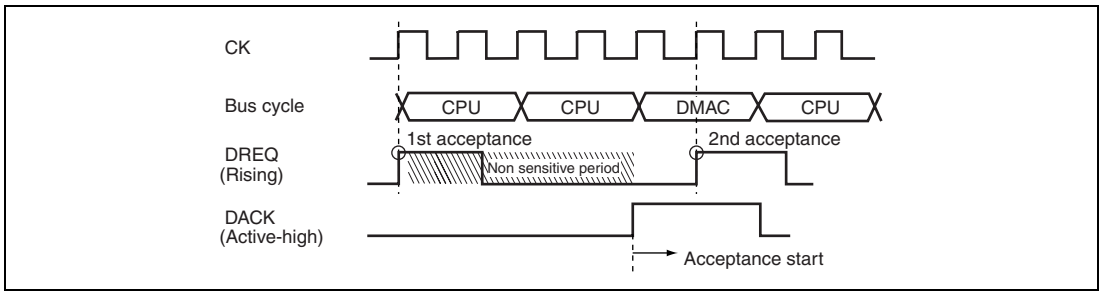

Figure 10.13 Example of DREQ Input Detection in Cycle Steal Mode Edge Detection

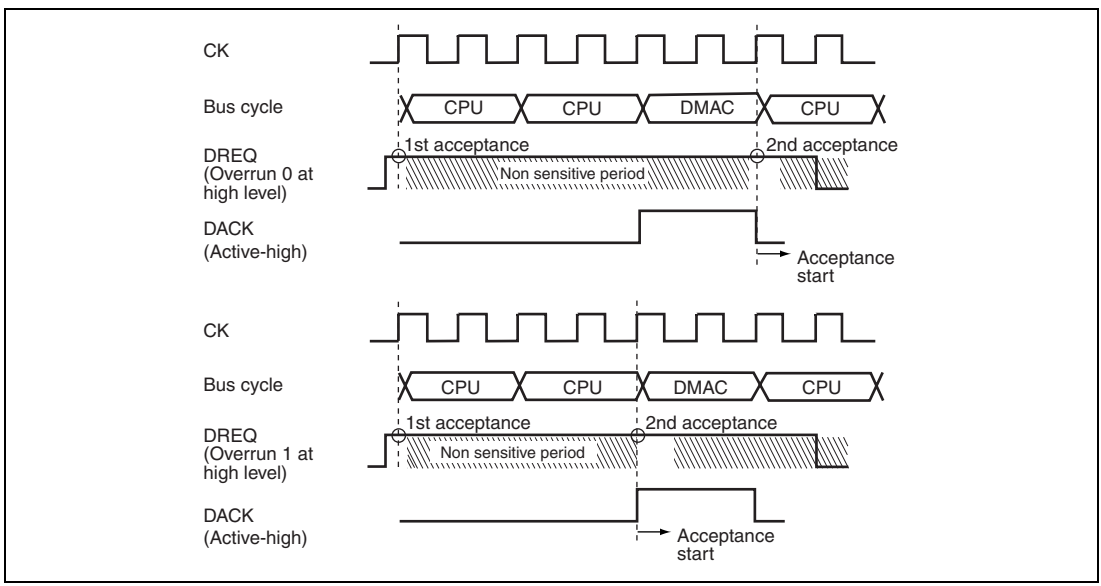

Figure 10.14 Example of DREQ Input Detection in Cycle Steal Mode Level Detection

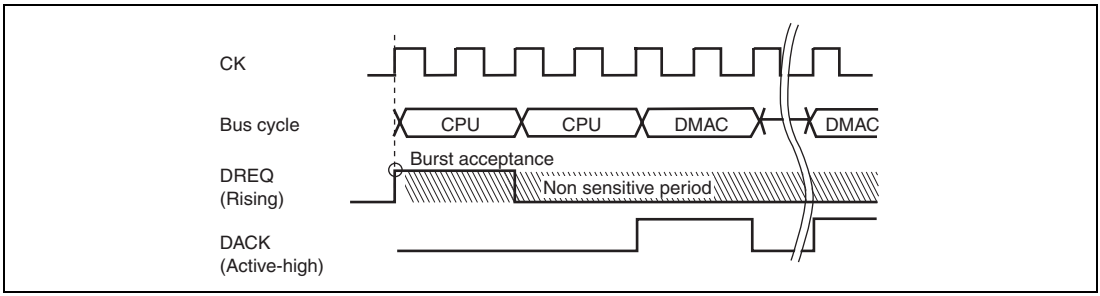

Figure 10.15 Example of DREQ Input Detection in Burst Mode Edge Detection

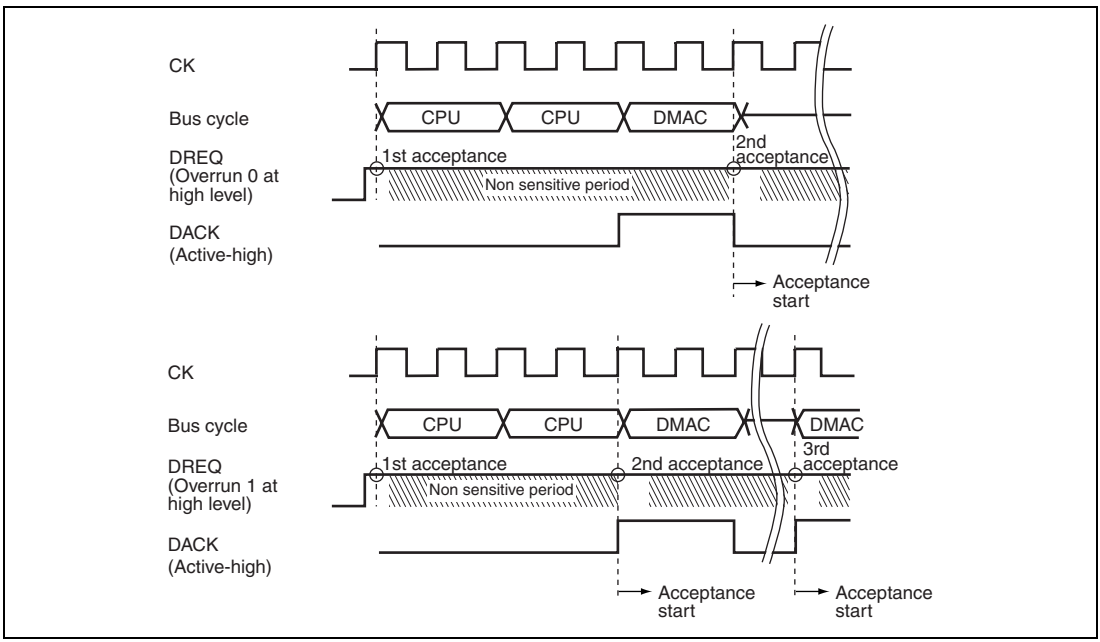

Figure 10.16 Example of DREQ Input Detection in Burst Mode Level Detection

Table 11.26 TIORH_4 (Channel 4)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_4 Function**	**TIOC4A 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.

Table 11.42 Setting of Bits BTE1 and BTE0

Bit 1	Bit 0	
BTE1	**BTE0**	**Description**
0	0	Enables transfer from the buffer registers to the temporary registers*1 and does not link the transfer with interrupt skipping operation.
0	1	Disables transfer from the buffer registers to the temporary registers.
1	0	Links transfer from the buffer registers to the temporary registers with interrupt skipping operation.*2
1	1	Setting prohibited

Note: 1. Data is transferred according to the MD3 to MD0 bit setting in TMDR. For details, refer to section 11.4.8, Complementary PWM Mode.

2. When interrupt skipping is disabled (the T3AEN and T4VEN bits are cleared to 0 in the timer interrupt skipping set register (TITCR) or the skipping count set bits (3ACOR and 4VCOR) in TITCR are cleared to 0)), be sure to disable link of buffer transfer with interrupt skipping (clear the BTE1 bit in the timer buffer transfer set register (TBTER) to 0). If link with interrupt skipping is enabled while interrupt skipping is disabled, buffer transfer will not be performed.

Bit	Bit Name	Initial Value	R/W	Description
0	MTU2CH34HIZ	0	R/W	MTU2 Channels 3 and 4 Output High-Impedance Specifies whether to place the high-current pins for the MTU2 in high-impedance state. 0: Does not place the pins in high-impedance state [Clearing conditions] • Power-on reset • By writing 0 to MTU2CH34HIZ after reading MTU2CH34HIZ = 1 1: Places the pins in high-impedance state [Setting condition] • By writing 1 to MTU2CH34HIZ

13.3.7 Port Output Enable Control Register 1 (POECR1)

POECR1 is an 8-bit readable/writable register that controls high-impedance state of the pins.

Bit:	7	6	5	4	3	2	1	0
	MTU2 PB4ZE	MTU2 PB3ZE	MTU2 PB2ZE	MTU2 PB1ZE	MTU2 PE3ZE	MTU2 PE2ZE	MTU2 PE1ZE	MTU2 PE0ZE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * Can be modified only once after a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
7	MTU2PB4ZE	0	R/W*	MTU2PB4 High-Impedance Enable Specifies whether to place the PB4/TIOC0D pin for channel 0 in the MTU2 in high-impedance state when either POE8F or MTU2CH0HIZ bit is set to 1. 0: Does not place the pin in high-impedance state 1: Places the pin in high-impedance state

(2) Writing to WRCSR

WRCSR must be written by a word access to address H'FFFE0004. It cannot be written by byte transfer or longword transfer instructions.

Procedures for writing 0 to WOVF (bit 7) and for writing to RSTE (bit 6) and RSTS (bit 5) are different, as shown in figure 15.3.

To write 0 to the WOVF bit, write H'A5 to the upper byte and write the write data to the lower byte. This clears the WOVF bit to 0. The RSTE and RSTS bits are not affected. To write to the RSTE and RSTS bits, the upper byte must be H'5A and the lower byte must be the write data. The values of bits 6 and 5 of the lower byte are transferred to the RSTE and RSTS bits, respectively. The WOVF bit is not affected.

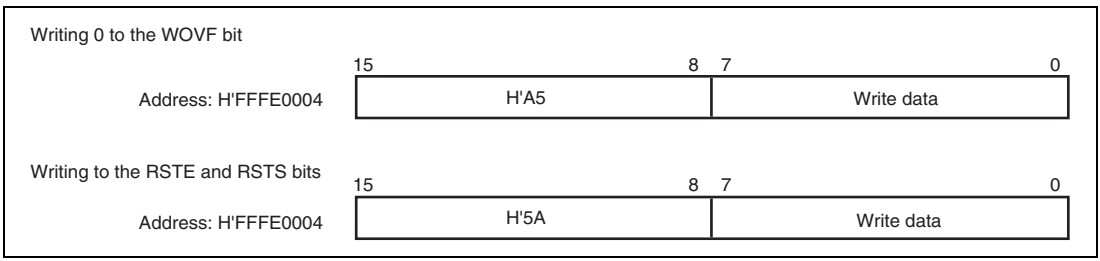

Figure 15.3 Writing to WRCSR

(3) Reading from WTCNT, WTCSR, and WRCSR

WTCNT, WTCSR, and WRCSR are read in a method similar to other registers. WTCSR is allocated to address H'FFFE0000, WTCNT to address H'FFFE0002, and WRCSR to address H'FFFE0004. Byte transfer instructions must be used for reading from these registers.

17.3.5 Serial Mode Register (SCSMR)

SCSMR specifies the SCIF serial communication format and selects the clock source for the baud rate generator.

The CPU can always read and write to SCSMR. SCSMR is initialized to H'0000 by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	-	CKS[1:0]	
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/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	C/$\overline{A}$	0	R/W	Communication Mode Selects whether the SCIF operates in asynchronous or clocked synchronous mode. 0: Asynchronous mode 1: Clocked synchronous mode
6	CHR	0	R/W	Character Length Selects 7-bit or 8-bit data length in asynchronous mode. In clocked synchronous mode, the data length is always 8 bits, regardless of the CHR setting. 0: 8-bit data 1: 7-bit data* Note: * When 7-bit data is selected, the MSB (bit 7) of the transmit FIFO data register is not transmitted.

7. Interrupt Sources

- Maskable interrupt sources are provided.
  - RSPI receive interrupt (receive buffer full)
  - RSPI transmit interrupt (transmit buffer empty)
  - RSPI error interrupt (mode fault and overrun)

8. Other Features

- Loopback mode is provided.
- The CMOS/open drain output switchover function is provided.
- The RSPI disable (initialization) function is provided.

18.1.1 Internal Block Diagram

Figure 18.1 shows an RSPI block diagram.

(1-4) Transfer Operation Flowchart (CPHA = 1)

Figure 18.26 shows an example of transfer operation flowchart for the RSPI during clock synchronous operation.

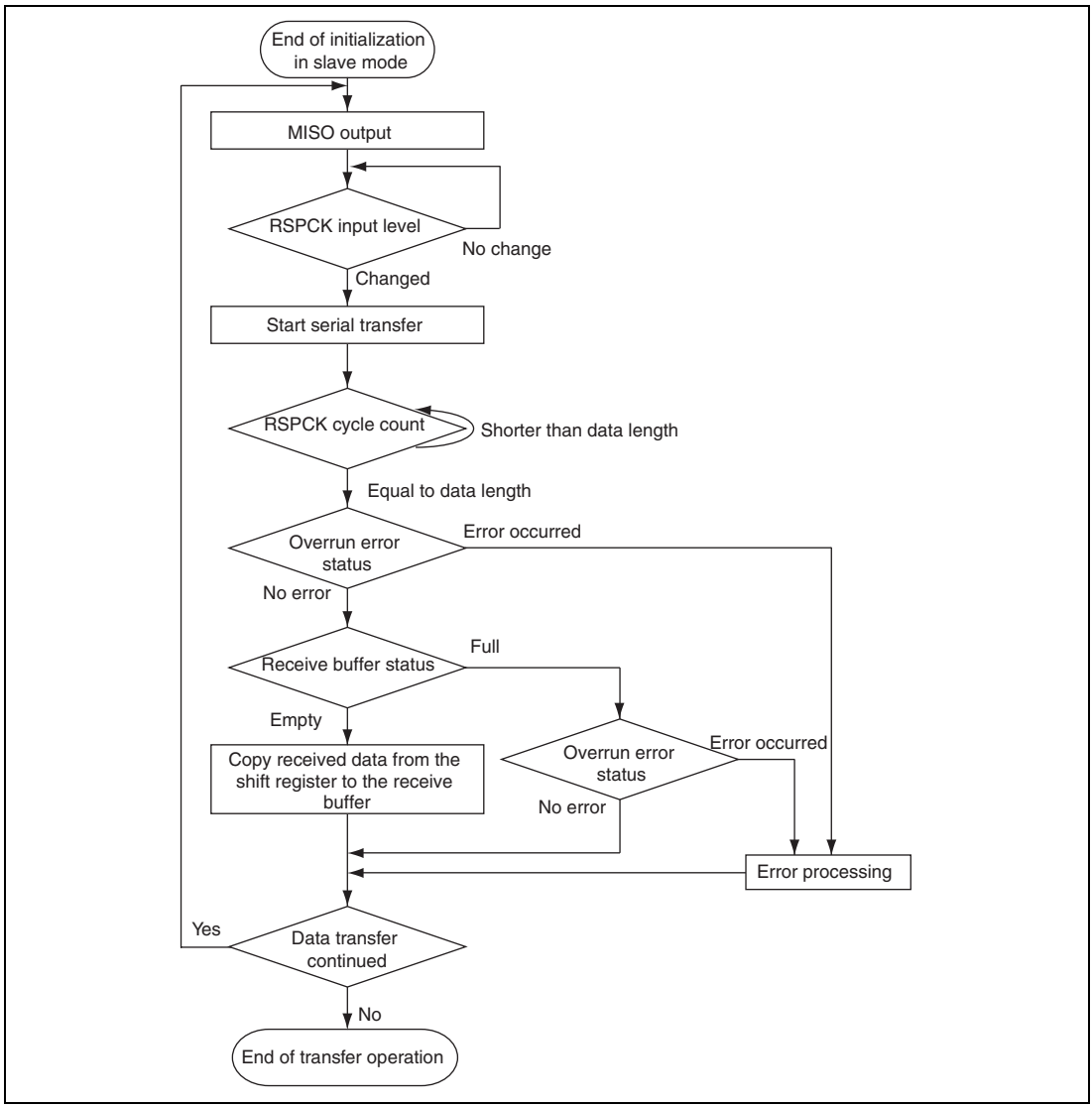

Figure 18.26 Example of Transfer Operation Flowchart in Slave Mode (CPHA = 1)

19.3.8 I²C Bus Receive Data Register (ICDRR)

ICDRR is an 8-bit register that stores the receive data. When data of one byte is received, ICDRR transfers the receive data from ICDRS to ICDRR and the next data can be received. ICDRR is a receive-only register, therefore the CPU cannot write to this register.

ICDRR is initialized to H'FF by a power-on reset.

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

19.3.9 I²C Bus Shift Register (ICDRS)

ICDRS is a register that is used to transfer/receive data. In transmission, data is transferred from ICDRT to ICDRS and the data is sent from the SDA pin. In reception, data is transferred from ICDRS to ICDRR after data of one byte is received. This register cannot be read directly from the CPU.

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

Table 24.6 Pin Input Values and Monitored USBTRNTREG1 Values

Register Setting		Pin Input Value		Monitored USBTRNTREG1 Value			
PTSTE	**SUSPEND**	**USD+**	**USD-**	**xver_data**	**dpls**	**dmns**	**Remarks**
0	×	×	×	0	0	0	Cannot be monitored when VBUS = 0 or PTSTE = 0.
1	0	0	0	×	0	0	Can be monitored when VBUS = 1 and PTSTE = 1.
1	0	0	1	0	0	1	
1	0	1	0	1	1	0	
1	0	1	1	×	1	1	
1	1	0	0	0	0	0	
1	1	0	1	0	0	1	
1	1	1	0	0	1	0	
1	1	1	1	0	1	1	

[Legend]

×: Don't care

24.11 Usage Notes

24.11.1 Receiving Setup Data

For USBEPDR0s that receives 8-byte setup data, note the following:

1. Since the USB always receives the setup command, writing from the USB bus has priority over reading from the CPU. When the USB starts receiving the next setup command while the CPU is reading data after data reception, the USB forcibly invalidates reading from the CPU to start writing. Therefore, the value that is read after starting reception is undefined.
2. USBEPDR0s must be read in 8-byte units. When reading is stopped in the middle, the data that is received by the next setup command cannot be read correctly.

24.11.2 Clearing FIFO

If the connected USB cable is disconnected during communication, the data being received or transmitted may remain in the FIFO. Therefore, clear the FIFO immediately after the USB cable is connected.

Do not clear the FIFO that is receiving data from or transmitting data to the host.

24.11.3 Overreading or Overwriting Data Registers

Note the following when reading or writing the data registers of this module:

(1) Receive Data Register

Do not read data that exceeds the valid receive data size from the receive data register. That is, data that exceeds the number of bytes specified in the receive data size register must not be read. For USBEPDR1 and USBEPDR4 that have two FIFOs, the maximum number of bytes that can be read at one time is 64 bytes. After reading data on the currently selected side, write 1 to the EPxRDFN bit in USBTRGx to change the current side to another side. This allows the number of bytes for the new side to be used as the receive data size, enabling the next data to be read.

25.4.7 Flow Control

The EtherC supports flow control functions conforming to IEEE802.3x for full-duplex operation. The flow control is available for both receive and transmit operations. When transmitting PAUSE frames, flow control can be performed in the following two procedures:

(1) Transmitting Automatic PAUSE Frames

For receive frames, PAUSE frames are automatically transmitted when the volume of data written to the receive FIFO (in the E-DMAC) reaches the value set in the flow control start FIFO threshold setting register (FCFTR) in the E-DMAC. The TIME parameter contained in the PAUSE frame is set by the automatic PAUSE frame register (APR). The automatic PAUSE frame transmission is repeated until the volume of data in the receive FIFO becomes less than the FCFTR value as the receive data is read from the FIFO. The upper limit of PAUSE frame retransmission counts can also be set in the automatic PAUSE frame retransmit count register (TPAUSER). In this case, PAUSE frame transmission is repeated until the volume of receive FIFO data becomes less than the FCFTR value, or the transmit count reaches the TPAUSER value. Transmission of automatic PAUSE frames is enabled when the TXF bit in the EtherC mode register (ECMR) is 1.

(2) Transmitting Manual PAUSE Frames

PAUSE frames are transmitted by software instructions. When a Timer value is written to the manual PAUSE frame register (MPR), manual PAUSE frame transmission is started. With this method, PAUSE frame transmission is carried out only once.

(3) Receiving PAUSE Frames

After a PAUSE frame is received, the next frame is not transmitted until the time indicated by the Timer value elapses. However, the ongoing transmission of a frame is continued. Reception of PAUSE frames is enabled when the RXF bit in ECMR is set to 1.

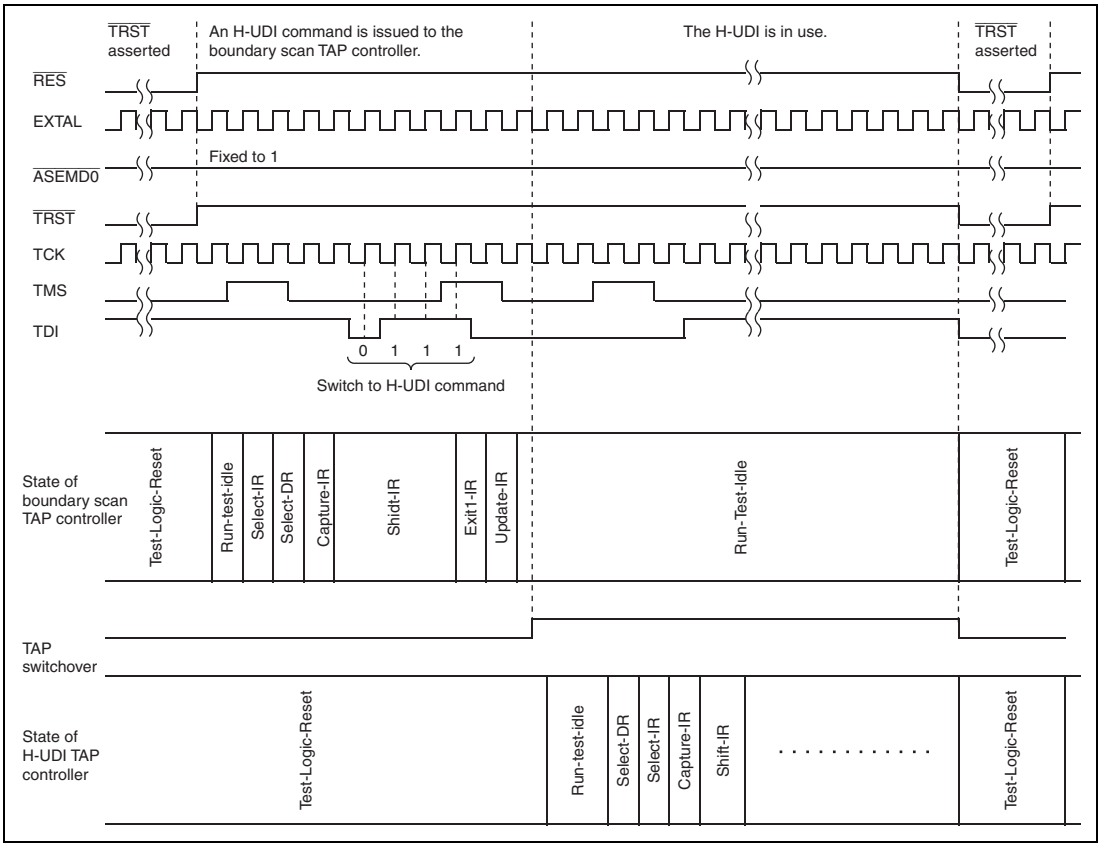

Figure 31.2 Switchover Sequence from Boundary Scan TAP Controller to H-UDI

31.4 H-UDI TAP Controller

The H-UDI of this LSI provides the TAP controller for the H-UDI functions (hereafter referred to as H-UDI TAP controller), separately from the boundary scan TAP controller.

This H-UDI TAP controller is enabled by issuing the switch to H-UDI command to the boundary scan TAP controller.

Table 31.3 lists the commands that the H-UDI TAP controller supports. If a command longer than 4 bits is issued from the TDI pin, the last 4 bits of the serial data become valid. Operation is not guaranteed if a reserved value defined in the table is input.

Table 31.3 Commands that H-UDI TAP Controller Supports

TI3	TI2	TI1	TI0	Description
0	0	0	0	Reserved
0	0	0	1	Reserved
0	0	1	0	Reserved
0	0	1	1	Reserved
0	1	0	0	Reserved
0	1	0	1	Reserved
0	1	1	0	H-UDI reset negate
0	1	1	1	H-UDI reset assert
1	0	0	0	Reserved
1	0	0	1	Reserved
1	0	1	0	Reserved
1	0	1	1	H-UDI interrupt
1	1	0	0	Reserved
1	1	0	1	Reserved
1	1	1	0	IDCODE (Initial value)
1	1	1	1	BYPASS