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	Obsolete
Core Processor	H8S/2000
Core Size	16-Bit
Speed	34MHz
Connectivity	I²C, IrDA, SCI, SmartCard
Peripherals	DMA, POR, PWM, WDT
Number of I/O	96
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	24K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 16x10b; D/A 6x8b
Oscillator Type	External
Operating Temperature	-20°C ~ 75°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/df2371vfq34v

		Pin No.					
Type	**Symbol**	**H8S/2378 0.18µm F-ZTAT Group, H8S/2378R 0.18µm F-ZTAT Group (LQFP-144)**	**H8S/2378 0.18µm F-ZTAT Group, H8S/2378R 0.18µm F-ZTAT Group (LGA-145)**	**H8S/2377 H8S/2377R**	**H8S/2375 H8S/2373 H8S/2375R H8S/2373R**	**I/O**	**Function**
Programmable pulse generator (PPG)	PO15 to PO0	49 to 42, 58 to 51	N5, M6, L5, M5, N4, L4, M4, L3, M8, N7, K8, K7, K6, N6, M7, L6	49 to 42, 58 to 51	49 to 42, 58 to 51	Output	Pulse output pins.
8-bit timer (TMR)	TMO0 TMO1	105, 106	C12, C13	105, 106	105, 106	Output	Waveform output pins with output compare function.
	TMCI0 TMCI1	83, 104	J10, D10	83, 104	83, 104	Input	External event input pins.
	TMRI0 TMRI1	82, 81	K13, J12	82, 81	82, 81	Input	Counter reset input pins.
Watchdog timer (WDT)	$\overline{\text{WDTOVF}}$	39	M3	39	39	Output	Counter overflow signal output pin in watchdog timer mode.
Serial communication interface (SCI)/ smart card interface (SCI_0 with IrDA function)	TxD4 TxD3 TxD2 TxD1 TxD0/ IrTxD	54, 33, 133, 141, 142	K6, L2, A6, B3, C4	54, 33, 133, 141, 142	54, 33, 133, 141, 142	Output	Data output pins.
	RxD4 RxD3 RxD2 RxD1 RxD0/ IrRxD	55, 59, 134, 139, 140	K7, L7, B5, C5, A4	55, 59, 134, 139, 140	55, 59, 134, 139, 140	Input	Data input pins.
	SCK4 SCK3 SCK2 SCK1 SCK0	138, 61, 135, 137, 138	B4, N8, C6, A5, B4	138, 61, 135, 137, 138	138, 61, 135, 137, 138	Input/ output	Clock input/output pins.

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address[*1] Advanced Mode	IPR	Priority	DTC Activation	DMAC Activation
IIC2	IICI0	116	H'01D0	IPRK6 to IPRK4	High	—	—
	Reserved for system use	117	H'01D4			—	—
	IICI1	118	H'01D8			—	—
	Reserved for system use	119	H'01DC			—	—
	Reserved for system use	120	H'01E0	IPRK2 to IPRK0		—	—
		121	H'01E4			—	—
		122	H'01E8			—	—
		123	H'01EC			—	—
		124	H'01F0			—	—
		125	H'01F4			—	—
		126	H'01F8			—	—
		127	H'01EC		Low	—	—

Notes: 1. Lower 16 bits of the start address.
2. Not supported for the H8S/2375, H8S/2375R, H8S/2373, and H8S/2373R.

6.3.1 Bus Width Control Register (ABWCR)

ABWCR designates each area in the external address space as either 8-bit access space or 16-bit access space.

Bit	Bit Name	Initial Value*	R/W	Description
7	ABW7	1/0	R/W	Area 7 to 0 Bus Width Control
6	ABW6	1/0	R/W	These bits select whether the corresponding area is to be designated as 8-bit access space or 16-bit access space.
5	ABW5	1/0	R/W	
4	ABW4	1/0	R/W	
3	ABW3	1/0	R/W	0: Area n is designated as 16-bit access space
2	ABW2	1/0	R/W	
1	ABW1	1/0	R/W	1: Area n is designated as 8-bit access space
0	ABW0	1/0	R/W	(n = 7 to 0)

Note: * In modes 2 and 4, ABWCR is initialized to 1. In modes 1 and 7, ABWCR is initialized to 0.

6.3.2 Access State Control Register (ASTCR)

ASTCR designates each area in the external address space as either 2-state access space or 3-state access space.

Bit	Bit Name	Initial Value	R/W	Description
7	AST7	1	R/W	Area 7 to 0 Access State Control
6	AST6	1	R/W	These bits select whether the corresponding area is to be designated as 2-state access space or 3-state access space. Wait state insertion is enabled or disabled at the same time.
5	AST5	1	R/W	
4	AST4	1	R/W	
3	AST3	1	R/W	
2	AST2	1	R/W	
1	AST1	1	R/W	
0	AST0	1	R/W	0: Area n is designated as 2-state access space Wait state insertion in area n access is disabled 1: Area n is designated as 3-state access space Wait state insertion in area n access is enabled (n = 7 to 0)

8.4.9 EXDMAC Bus Cycles (Dual Address Mode)

Normal Transfer Mode (Cycle Steal Mode): Figure 8.15 shows an example of transfer when $\overline{\text{ETEND}}$ output is enabled, and word-size, normal transfer mode (cycle steal mode) is performed from external 16-bit, 2-state access space to external 16-bit, 2-state access space.

After one byte or word has been transferred, the bus is released. While the bus is released, one CPU, DMAC, or DTC bus cycle is initiated.

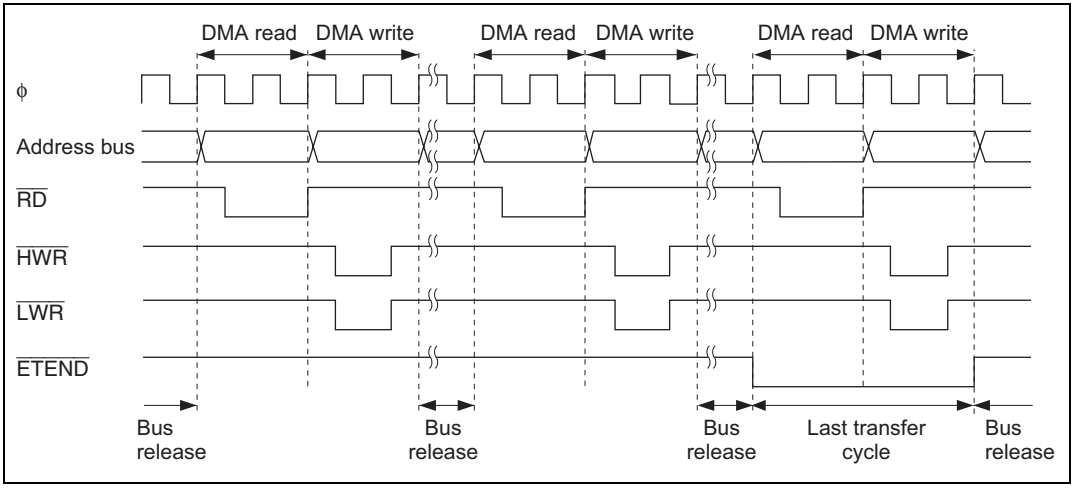

Figure 8.15 Example of Normal Transfer Mode (Cycle Steal Mode) Transfer

Origin of Activation Source	Activation Source	Vector Number	DTC Vector Address	DTCE*	Priority
TPU_3	TGI3A	56	H'0470	DTCED5	High
	TGI3B	57	H'0472	DTCED4	
	TGI3C	58	H'0474	DTCED3	
	TGI3D	59	H'0476	DTCED2	
TPU_4	TGI4A	64	H'0480	DTCED1	
	TGI4B	65	H'0482	DTCED0	
TPU_5	TGI5A	68	H'0488	DTCEE7	
	TGI5B	69	H'048A	DTCEE6	
TMR_0	CMIA0	72	H'0490	DTCEE3	
	CMIB0	73	H'0492	DTCEE2	
TMR_1	CMIA1	76	H'0498	DTCEE1	
	CMIB1	77	H'049A	DTCEE0	
DMAC	DMTEND0A	80	H'04A0	DTCEF7	
	DMTEND0B	81	H'04A2	DTCEF6	
	DMTEND1A	82	H'04A4	DTCEF5	
	DMTEND1B	83	H'04A6	DTCEF4	
SCI_0	RXI0	89	H'04B2	DTCEF3	
	TXI0	90	H'04B4	DTCEF2	
SCI_1	RXI1	93	H'04BA	DTCEF1	
	TXI1	94	H'04BC	DTCEF0	
SCI_2	RXI2	97	H'04C2	DTCEG7	
	TXI2	98	H'04C4	DTCEG6	
SCI_3	RXI3	101	H'04CA	DTCEF5	
	TXI3	102	H'04CC	DTCEF4	
SCI_4	RXI4	105	H'04D2	DTCEG3	
	TXI4	106	H'04D4	DTCEG2	Low

Note: * DTCE bits with no corresponding interrupt are reserved, and 0 should be written to. When clearing the software standby state or all-module-clocks-stop mode with an interrupt, write 0 to the corresponding DTCE bit.

10.12.4 Port D Pull-up Control Register (PDPCR)

PDPCR controls on/off states of the input pull-up MOS of port D. PDPCR is valid in mode 7.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7PCR	0	R/W	When PDDDR = 0 (input port), the input pull-up MOS of the input pin is on when the corresponding bit is set to 1.
6	PD6PCR	0	R/W	
5	PD5PCR	0	R/W	
4	PD4PCR	0	R/W	
3	PD3PCR	0	R/W	
2	PD2PCR	0	R/W	
1	PD1PCR	0	R/W	
0	PD0PCR	0	R/W	

10.12.5 Pin Functions

Port D pins also function as the pins for data I/Os. The correspondence between the register specification and the pin functions is shown below.

- PD7/D15, PD6/D14, PD5/D13, PD4/D12, PD3/D11, PD2/D10, PD1/D9, PD0/D8

  The pin function is switched as shown below according to the operating mode, bit EXPE, and bit PDDDR.

Operating mode	1, 2, 4	7		
EXPE	—	0		1
PDnDDR	—	0	1	—
Pin function	Data I/O	PDn input	PDn output	Data I/O

Legend: n = 7 to 0

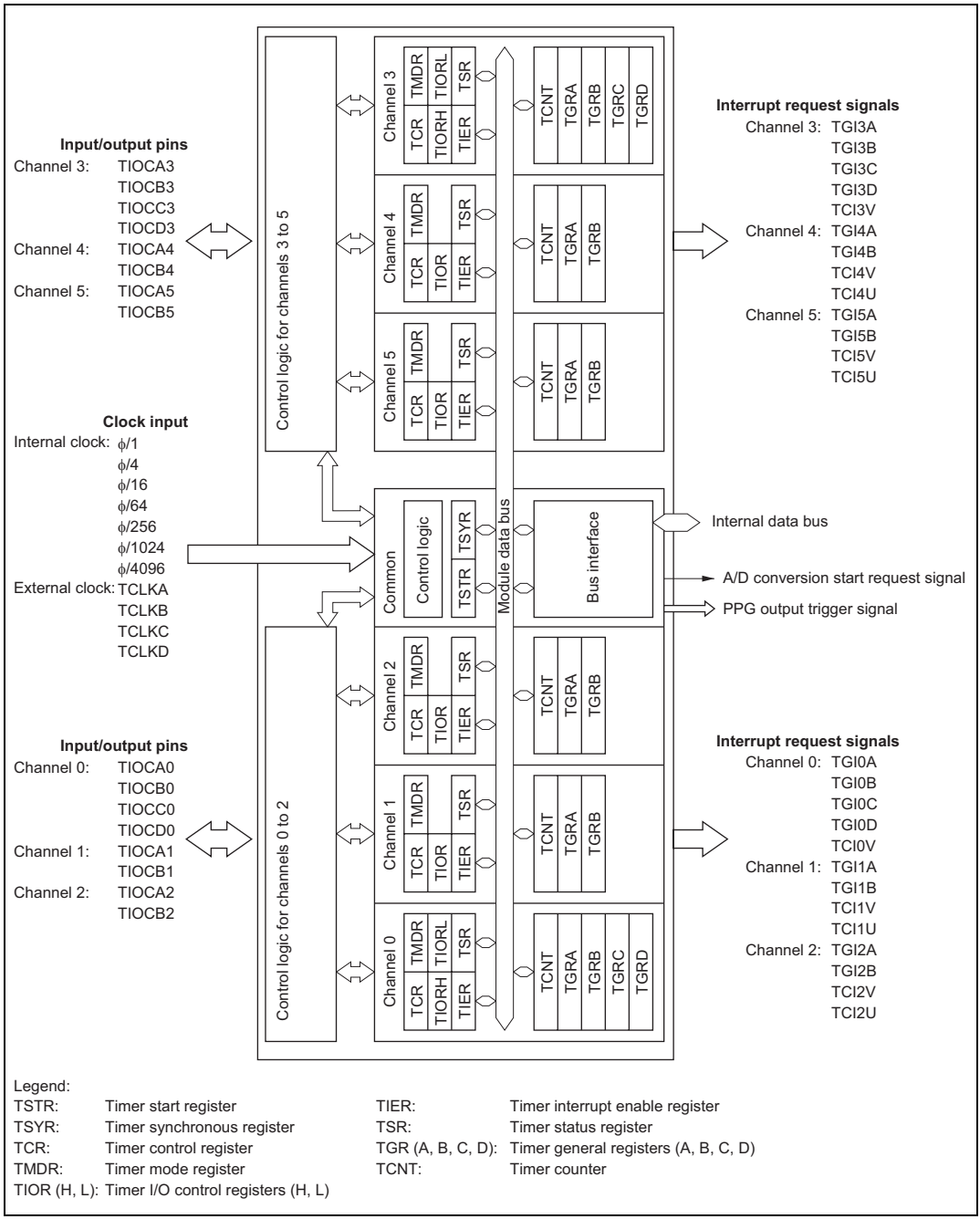

Figure 11.1 Block Diagram of TPU

- Break detection: Break can be detected by reading the RxD pin level directly in case of a framing error
- Average transfer rate generator (only for H8S/2378R Group): The following transfer rate can be selected (SCI_2 only)

  115.152 or 460.606 kbps at 10.667-MHz operation

  115.196, 460.784, or 720 kbps at 16-MHz operation

  720 kbps at 32-MHz operation

Clocked Synchronous mode

- Data length: 8 bits
- Receive error detection: Overrun errors detected

Smart Card Interface

- Automatic transmission of error signal (parity error) in receive mode
- Error signal detection and automatic data retransmission in transmit mode
- Direct convention and inverse convention both supported

already been written to TDR during serial transmission, the SCI transfers the written data to TSR to continue transmission. Although TDR can be read or written to by the CPU at all times, to achieve reliable serial transmission, write transmit data to TDR for only once after confirming that the TDRE bit in SSR is set to 1.

15.3.4 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data. To perform serial data transmission, the SCI first transfers transmit data from TDR to TSR, then sends the data to the TxD pin starting. TSR cannot be directly accessed by the CPU.

15.3.5 Serial Mode Register (SMR)

SMR is used to set the SCI's serial transfer format and select the on-chip baud rate generator clock source.

Some bit functions of SMR differ in normal serial communication interface mode and Smart Card interface mode.

Normal Serial Communication Interface Mode (When SMIF in SCMR is 0)

Bit	Bit Name	Initial Value	R/W	Description
7	C/$\overline{A}$	0	R/W	Communication Mode 0: Asynchronous mode 1: Clocked synchronous mode
6	CHR	0	R/W	Character Length (enabled only in asynchronous mode) 0: Selects 8 bits as the data length. 1: Selects 7 bits as the data length. LSB-first is fixed and the MSB (bit 7) of TDR is not transmitted in transmission. In clocked synchronous mode, a fixed data length of 8 bits is used.

Bit	Bit Name	Initial Value	R/W	Description
2	TEND	1	R	Transmit End This bit is set to 1 when no error signal has been sent back from the receiving end and the next transmit data is ready to be transferred to TDR. [Setting conditions] • When the TE bit in SCR is 0 and the ERS bit is also 0 • If the ERS bit is 0 and the TDRE bit is 1 after the specified interval after transmission of 1-byte data Timing to set this bit differs according to the register settings. GM = 0, BLK = 0: 2.5 etu*2 after transmission GM = 0, BLK = 1: 1.5 etu*2 after transmission GM = 1, BLK = 0: 1.0 etu*2 after transmission GM = 1, BLK = 1: 1.0 etu*2 after transmission [Clearing conditions] • When 0 is written to TEND after reading TEND = 1 • When the DMAC or DTC is activated by a TXI interrupt and writes data to TDR
1	MPB	0	R	Multiprocessor Bit This bit is not used in Smart Card interface mode.
0	MPBT	0	R/W	Multiprocessor Bit Transfer Write 0 to this bit in Smart Card interface mode.

Note: 1. Only 0 can be written, to clear the flag. Alternately, use the bit clear instruction to clear the flag.
2. Elementary time unit (etu): Transfer duration for one bit

15.4.5 Data Transmission (Asynchronous Mode)

Figure 15.6 shows an example of the operation for transmission in asynchronous mode. In transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if is cleared to 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit is set to 1 at this time, a transmit data empty interrupt request (TXI) is generated. Because the TXI interrupt routine writes the next transmit data to TDR before transmission of the current transmit data has finished, continuous transmission can be enabled.
3. Data is sent from the TxD pin in the following order: start bit, transmit data, parity bit or multiprocessor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks the TDRE flag at the timing for sending the stop bit.
5. If the TDRE flag is 0, the data is transferred from TDR to TSR, the stop bit is sent, and then serial transmission of the next frame is started.
6. If the TDRE flag is 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then the “mark state” is entered in which 1 is output. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated.

Figure 15.7 shows a sample flowchart for transmission in asynchronous mode.

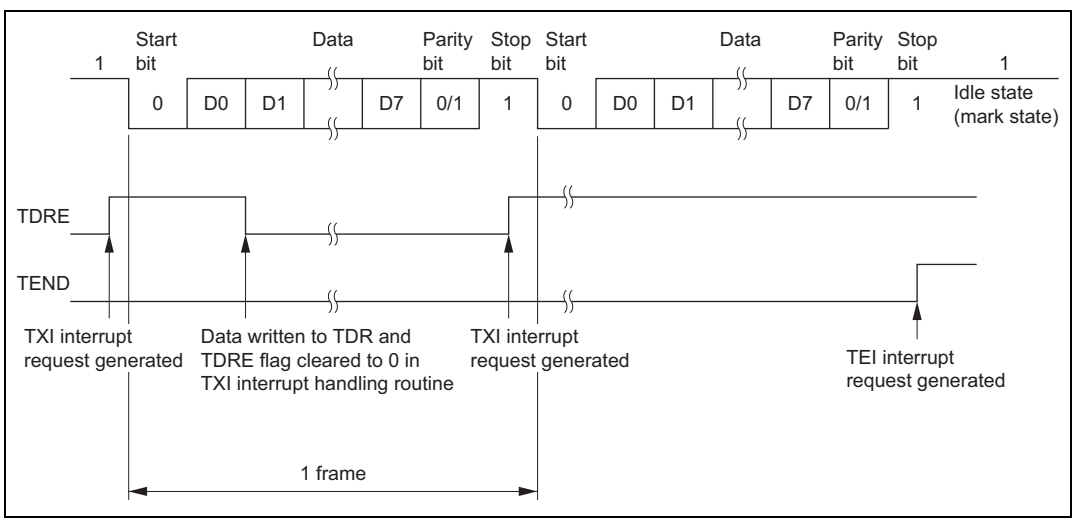

Figure 15.6 Example of Operation in Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)

When turning on the power or switching between Smart Card interface mode and software standby mode, the following procedures should be followed in order to maintain the clock duty cycle.

Powering On: To secure the clock duty cycle from power-on, the following switching procedure should be followed.

1. The initial state is port input and high impedance. Use a pull-up resistor or pull-down resistor to fix the potential.
2. Fix the SCK pin to the specified output level with the CKE1 bit in SCR.
3. Set SMR and SCMR, and switch to smart card mode operation.
4. Set the CKE0 bit in SCR to 1 to start clock output.

When Changing from Smart Card Interface Mode to Software Standby Mode:

1. Set the data register (DR) and data direction register (DDR) corresponding to the SCK pin to the value for the fixed output state in software standby mode.
2. Write 0 to the TE bit and RE bit in the serial control register (SCR) to halt transmit/receive operation. At the same time, set the CKE1 bit to the value for the fixed output state in software standby mode.
3. Write 0 to the CKE0 bit in SCR to halt the clock.
4. Wait for one serial clock period.
   During this interval, clock output is fixed at the specified level, with the duty cycle preserved.
5. Make the transition to the software standby state.

When Returning to Smart Card Interface Mode from Software Standby Mode:

1. Exit the software standby state.
2. Write 1 to the CKE0 bit in SCR and output the clock. Signal generation is started with the normal duty cycle.

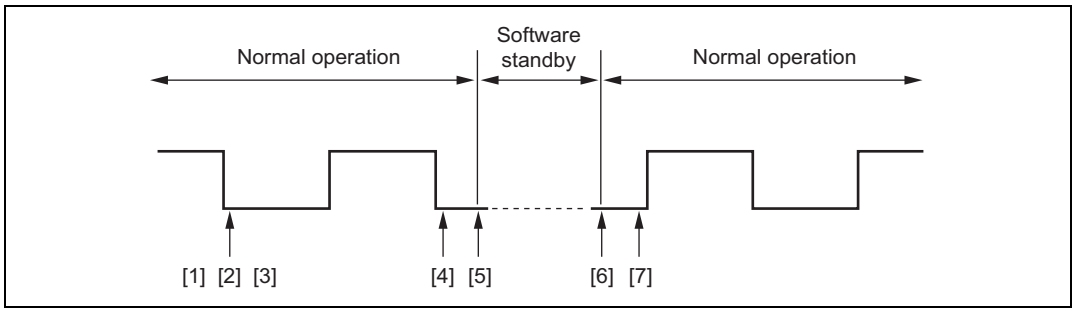

Figure 15.32 Clock Halt and Restart Procedure

15.8 IrDA Operation

When the IrDA function is enabled with bit IrE in IrCR, the SCI_0 TxD0 and RxD0 signals are subjected to waveform encoding/decoding conforming to IrDA specification version 1.0 (IrTxD and IrRxD pins). By connecting these pins to an infrared transceiver/receiver, it is possible to implement infrared transmission/reception conforming to the IrDA specification version 1.0 system.

In the IrDA specification version 1.0 system, communication is started at a transfer rate of 9600 bps, and subsequently the transfer rate can be varied as necessary. As the IrDA interface in this LSI does not include a function for varying the transfer rate automatically, the transfer rate setting must be changed by software.

Figure 15.33 shows a block diagram of the IrDA function.

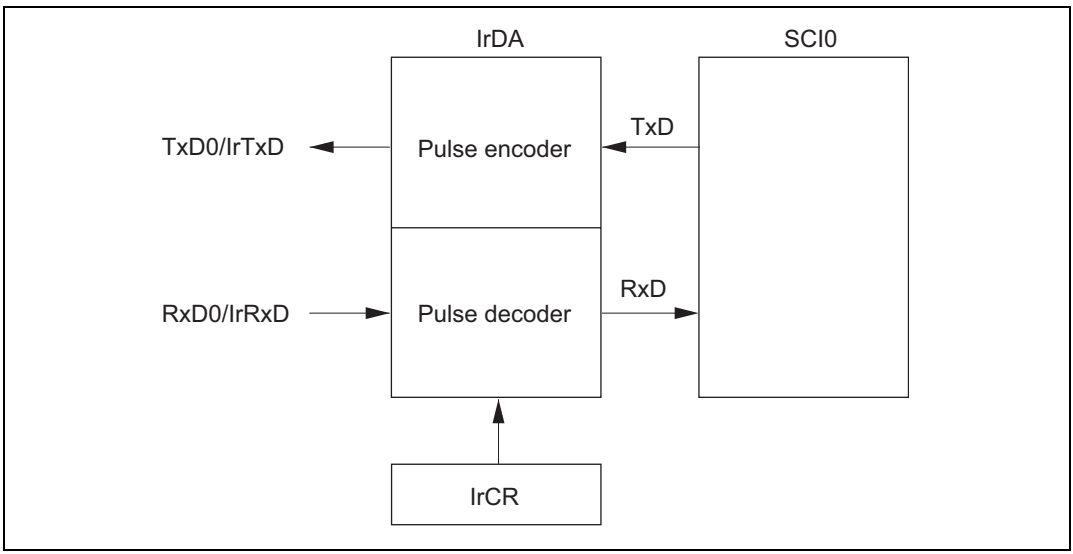

Figure 15.33 Block Diagram of IrDA

Transmission: In transmission, the output signal (UART frame) from the SCI is converted to an IR frame by the IrDA interface (see figure 15.34).

When the serial data is 0, a high pulse of 3/16 the bit rate (interval equivalent to the width of one bit) is output (initial value). The high-level pulse can be varied according to the setting of bits IrCKS2 to IrCKS0 in IrCR.

16.7 Usage Notes

(1) Issue (retransmit) the start/stop conditions after the fall of the ninth clock is confirmed.

Check SCLO in the I²C control register B (IICRB) to confirm the fall of the ninth clock.

When the start/stop conditions are issued (retransmitted) at the specific timing under the following condition (i) or (ii), such conditions may not be output successfully. This does not occur in other cases.

(i) When the rising of SCL falls behind the time specified in section 16.6, Bit Synchronous Circuit, by the load of the SCL bus (load capacitance or pull-up resistance)

(ii) When the bit synchronous circuit is activated by extending the low period of eighth and ninth clocks, that is driven by the slave device

(2) Control WAIT in the I²C bus mode register (ICMR) to be set to 0.

When WAIT is set to 1, and SCL is driven low for two or more transfer clocks by the slave device at the eighth and ninth clocks, the high period of ninth clock may be shortened. This does not occur in other cases.

(3) I²C bus interface 2 (IIC2) master receive mode

When operating in master receive mode with RDRF set to 1, SCL is driven low at the falling edge of the eighth clock cycle. However, when ICDRR is read near the falling edge of the eighth clock cycle, SCL is only fixed low for one clock cycle at the eighth clock cycle of the next receive data, after which SCL is no longer fixed and the ninth clock cycle is output, even if ICDRR is not read. This causes the receive data to overflow.

The following methods can be used to prevent this from occurring.

— In master receive mode, complete processing to read ICDRR before the rising edge of the eighth clock cycle.

— In master receive mode, set RCVD to 1 and perform communication processing one byte at a time.

(4) Limitations on transfer rate setting values when using I²C bus interface 2 (IIC2) in multi-master mode

When operating in multi-master mode and the IIC transfer rate setting of the MCU is slower than that of another master device, an SCL of an unanticipated width may by output occasionally. To prevent this, set the transfer rate to a value 1/1.8 or greater than the fastest transfer rate among the other master devices. For example, if the fastest transfer rate setting among the other master devices is 400 kbps, set the IIC transfer rate of the MCU to 223 kbps (400/1.8) or higher.

(5) Limitations on use of bit manipulation instructions to set MST and TRS when using I²C bus interface 2 (IIC2) in multi-master mode

21.3 Register Descriptions

The registers/parameters which control flash memory are shown as follows.

- Flash code control status register (FCCS)
- Flash program code select register (FPCS)
- Flash erase code select register (FECS)
- Flash key code register (FKEY)
- Flash MAT select register (FMATS)
- Flash transfer destination address register (FTDAR)
- Download pass and fail result (DPFP)
- Flash pass and fail result (FPFR)
- Flash multipurpose address area (FMPAR)
- Flash multipurpose data destination area (FMPDR)
- Flash erase Block select (FEBS)
- Flash program and erase frequency control (FPEFEQ)
- Flash vector address control register (FVACR)

There are several operating modes for accessing flash memory, for example, read mode/program mode.

There are two memory MATs: user MAT and user boot MAT. The dedicated registers/parameters are allocated for each operating mode and MAT selection. The correspondence of operating modes and registers/parameters for use is shown in table 21.3.

- Area-last address (four byte): Last address of the area
  There are as many groups of data representing the start and last addresses as there are areas.
- SUM (one byte): Checksum

(i) Erased Block Information Inquiry

The boot program will return the number of erased blocks and their addresses.

Command

H'26

- Command, H'26, (two bytes): Inquiry regarding erased block information

Response

H'36	Size	Number of blocks	
Block start address			Block last address
...			
SUM			

- Response, H'36, (one byte): Response to the number of erased blocks and addresses
- Size (three byte): The number of bytes that represents the number of blocks, block-start addresses, and block-last addresses.
- Number of blocks (one byte): The number of erased blocks
- Block start address (four bytes): Start address of a block
- Block last Address (four bytes): Last address of a block
  There are as many groups of data representing the start and last addresses as there are areas.
- SUM (one byte): Checksum

(j) Programming Unit Inquiry

The boot program will return the programming unit used to program data.

Command

H'27

- Command, H'27, (one byte): Inquiry regarding programming unit

Response

H'37	Size	Programming unit	SUM

- Response, H'37, (one byte): Response to programming unit inquiry
- Size (one byte): The number of bytes that indicate the programming unit, which is fixed to 2
- Programming unit (two bytes): A unit for programming
  This is the unit for reception of programming.
- SUM (one byte): Checksum

26.4.2 Control Signal Timing

The control signal timings are shown below.

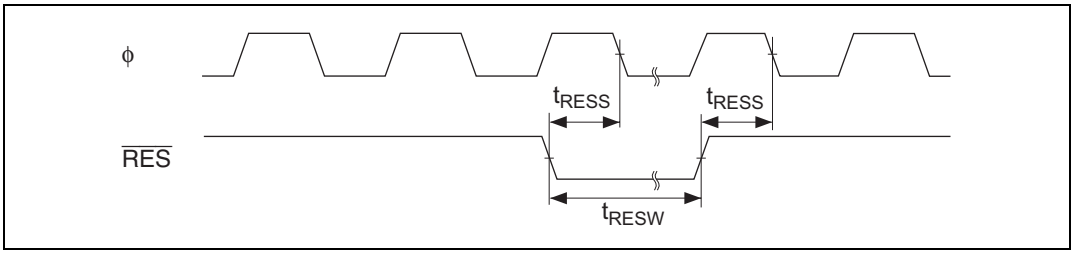

Figure 26.5 Reset Input Timing

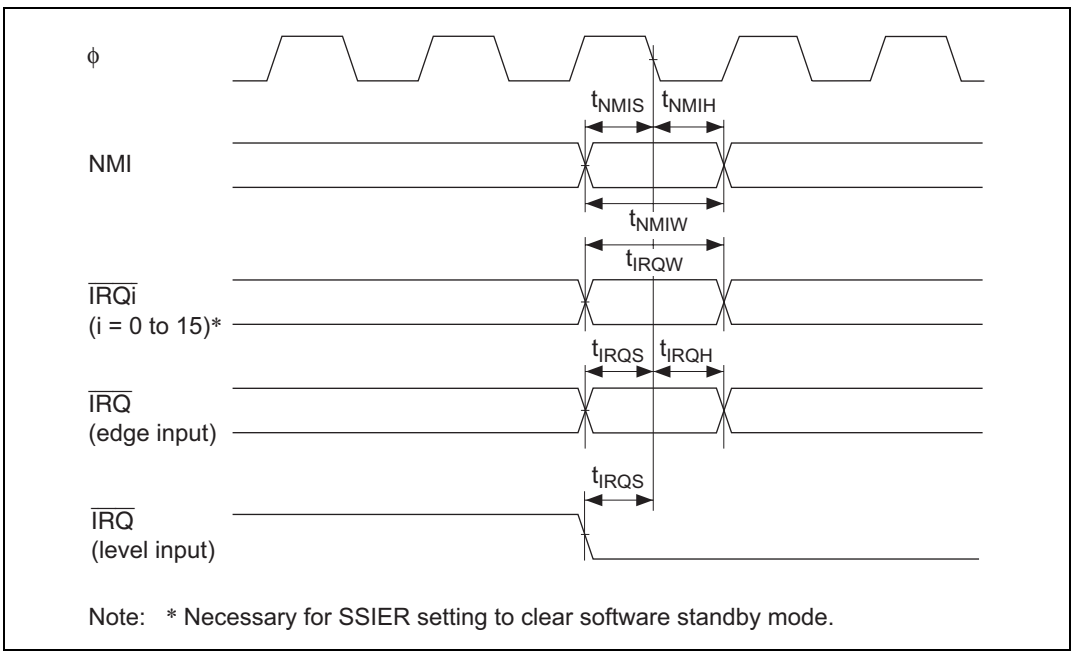

Figure 26.6 Interrupt Input Timing

Instruction	1	2	3	4	5	6	7	8	9
BLD #xx:3,@aa:8	R:W 2nd	R:B EA	R:W NEXT						
BLD #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W NEXT					
BLD #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W NEXT				
BNOT #xx:3,Rd	R:W NEXT								
BNOT #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BNOT #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BNOT #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
BNOT #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BNOT Rn,Rd	R:W NEXT								
BNOT Rn,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BNOT Rn,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BNOT Rn,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
BNOT Rn,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BOR #xx:3,Rd	R:W NEXT								
BOR #xx:3,@ERd	R:W 2nd	R:B EA	R:W NEXT						
BOR #xx:3,@aa:8	R:W 2nd	R:B EA	R:W NEXT						
BOR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W NEXT					
BOR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W NEXT				
BSET #xx:3,Rd	R:W NEXT								
BSET #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					