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	H8/300H
Core Size	16-Bit
Speed	20MHz
Connectivity	SCI, SmartCard
Peripherals	DMA, PWM, WDT
Number of I/O	35
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-20°C ~ 75°C (TA)
Mounting Type	Surface Mount
Package / Case	100-BFQFP
Supplier Device Package	100-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/d13007fi20v

2.5 Data Formats

The H8/300H CPU can process 1-bit, 4-bit (BCD), 8-bit (byte), 16-bit (word), and 32-bit (longword) data. Bit-manipulation instructions operate on 1-bit data by accessing bit n (n = 0, 1, 2, …, 7) of byte operand data. The DAA and DAS decimal-adjust instructions treat byte data as two digits of 4-bit BCD data.

2.5.1 General Register Data Formats

Figures 2.6 and 2.7 show the data formats in general registers.

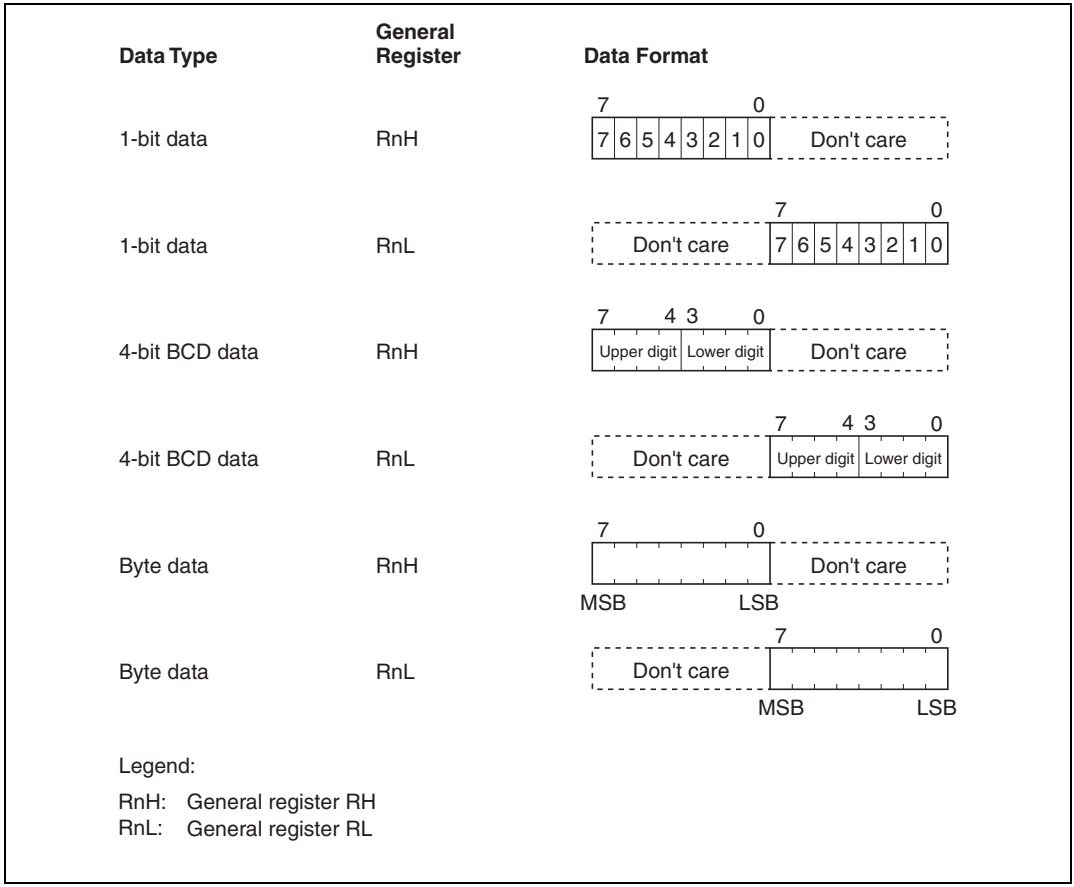


Figure 2.6 General Register Data Formats (1)

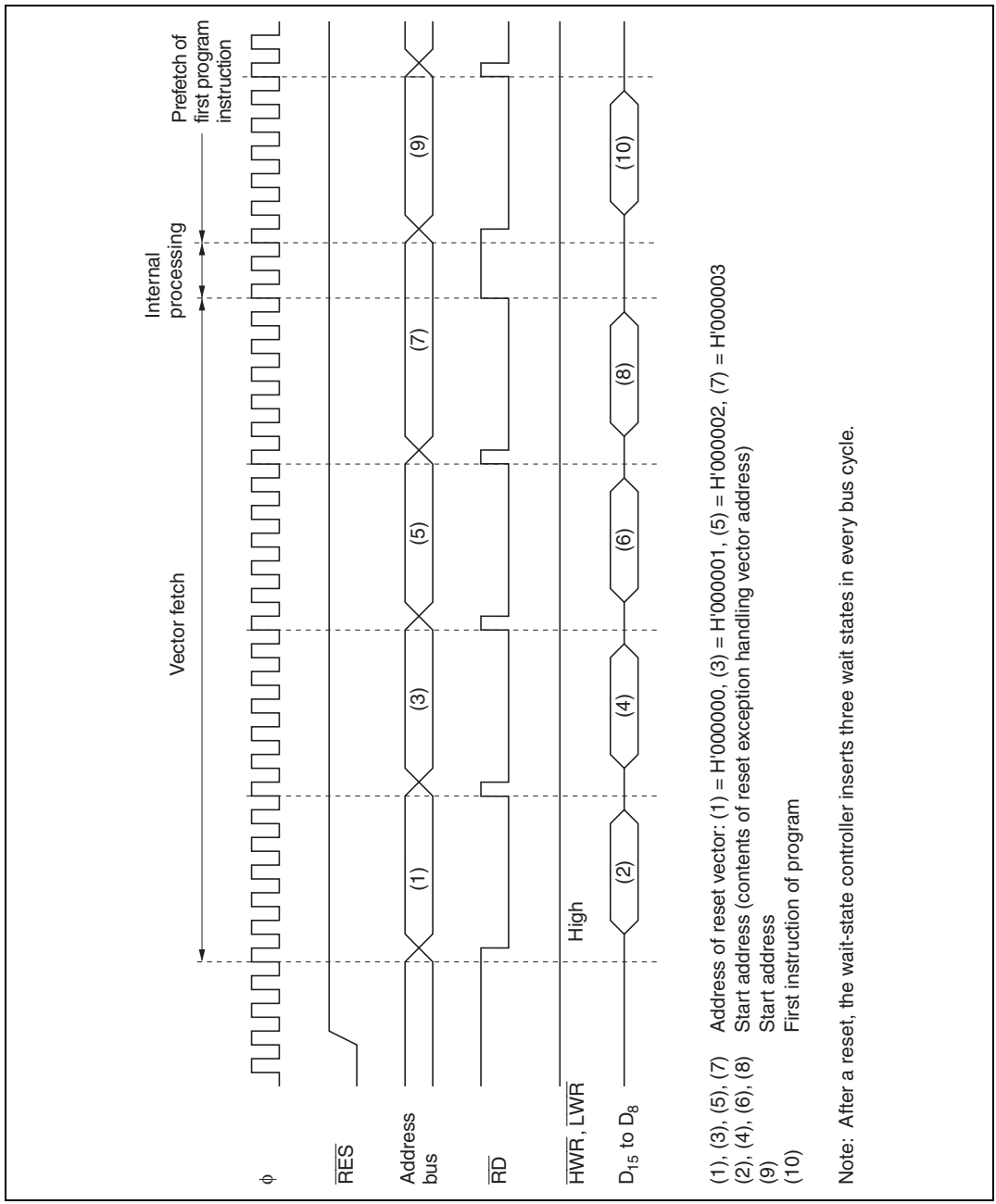


Figure 4.2 Reset Sequence (Modes 1 and 3)

WCRH

Bit	7	6	5	4	3	2	1	0
	W71	W70	W61	W60	W51	W50	W41	W40
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 7 and 6—Area 7 Wait Control 1 and 0 (W71, W70): These bits select the number of program wait states when area 7 in external space is accessed while the AST7 bit in ASTCR is set to 1.

Bit 7 W71	Bit 6 W70	Description
0	0	Program wait not inserted when external space area 7 is accessed
	1	1 program wait state inserted when external space area 7 is accessed
1	0	2 program wait states inserted when external space area 7 is accessed
	1	3 program wait states inserted when external space area 7 is accessed (Initial value)

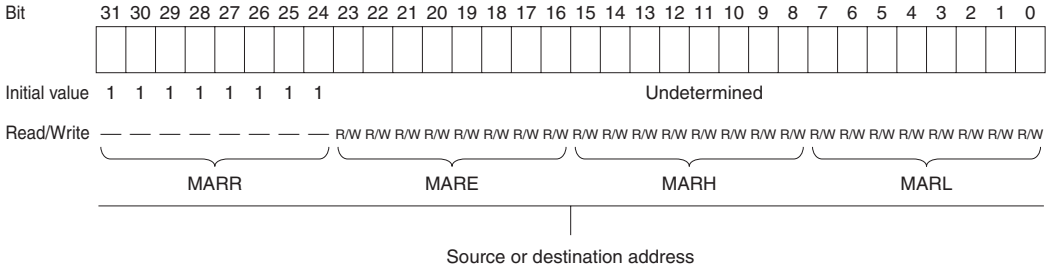
Bits 5 and 4—Area 6 Wait Control 1 and 0 (W61, W60): These bits select the number of program wait states when area 6 in external space is accessed while the AST6 bit in ASTCR is set to 1.

Bit 5 W61	Bit 4 W60	Description
0	0	Program wait not inserted when external space area 6 is accessed
	1	1 program wait state inserted when external space area 6 is accessed
1	0	2 program wait states inserted when external space area 6 is accessed
	1	3 program wait states inserted when external space area 6 is accessed (Initial value)

7.2.1 Memory Address Registers (MAR)

A memory address register (MAR) is a 32-bit readable/writable register that specifies a source or destination address. The transfer direction is determined automatically from the activation source.

An MAR consists of four 8-bit registers designated MARR, MARE, MARH, and MARL. All bits of MARR are reserved; they cannot be modified and are always read as 1.



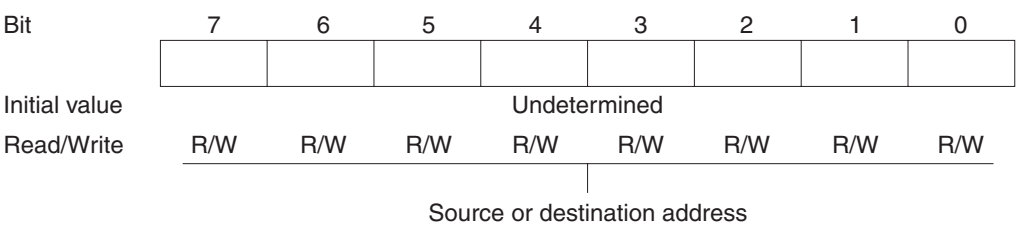
An MAR functions as a source or destination address register depending on how the DMAC is activated: as a destination address register if activation is by a receive-data-full interrupt from the serial communication interface (SCI) (channel 0) or by a conversion-end interrupt from the A/D converter, and as a source address register otherwise.

The MAR value is incremented or decremented each time one byte or word is transferred, automatically updating the source or destination memory address. For details, see section 7.3.4, Data Transfer Control Registers (DTCR).

The MARs are not initialized by a reset or in standby mode.

7.2.2 I/O Address Registers (IOAR)

An I/O address register (IOAR) is an 8-bit readable/writable register that specifies a source or destination address. The IOAR value is the lower 8 bits of the address. The upper 16 address bits are all 1 (H'FFFF).



An IOAR functions as a source or destination address register depending on how the DMAC is activated: as a source address register if activation is by a receive-data-full interrupt from the SCI

7.4.14 DMAC States in Reset State, Standby Modes, and Sleep Mode

When the chip is reset or enters hardware standby mode or software standby mode, the DMAC is initialized and halts. DMAC operations continue in sleep mode. Figure 7.24 shows the timing of a cycle-steal transfer in sleep mode.

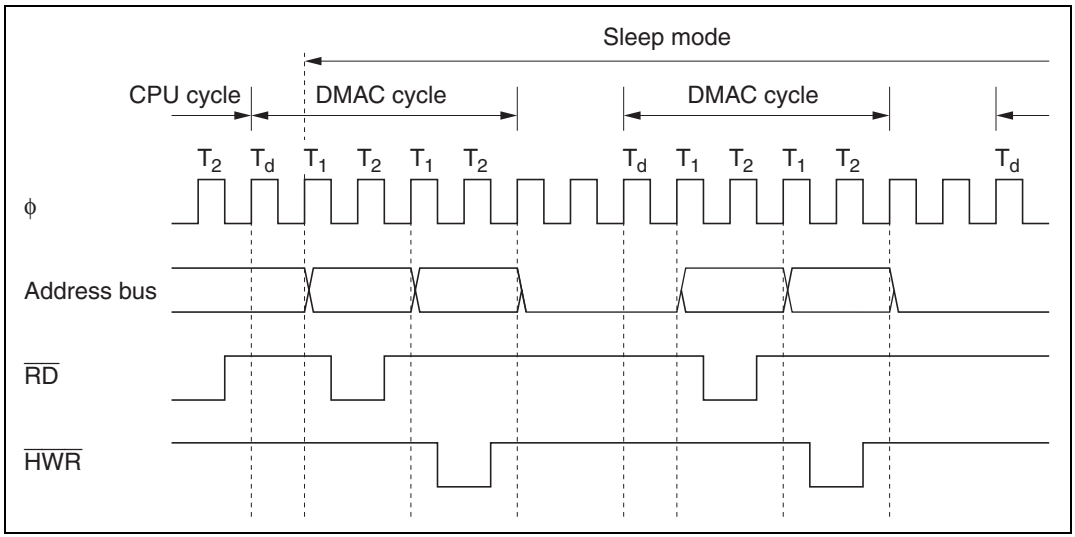


Figure 7.24 Timing of Cycle-Steal Transfer in Sleep Mode

7.5 Interrupts

The DMAC generates only DMA-end interrupts. Table 7.13 lists the interrupts and their priority.

Table 7.13 DMAC Interrupts

Interrupt	Description		Interrupt Priority
	Short Address Mode	Full Address Mode	
DEND0A	End of transfer on channel 0A	End of transfer on channel 0	High
DEND0B	End of transfer on channel 0B	—	
DEND1A	End of transfer on channel 1A	End of transfer on channel 1	
DEND1B	End of transfer on channel 1B	—	Low

7.6.5 Note on Activating DMAC by Internal Interrupts

When using an internal interrupt to activate the DMAC, make sure that the interrupt selected as the activating source does not occur during the interval after it has been selected but before the DMAC has been enabled. The on-chip supporting module that will generate the interrupt should not be activated until the DMAC has been enabled. If the DMAC must be enabled while the on-chip supporting module is active, follow the procedure in figure 7.26.

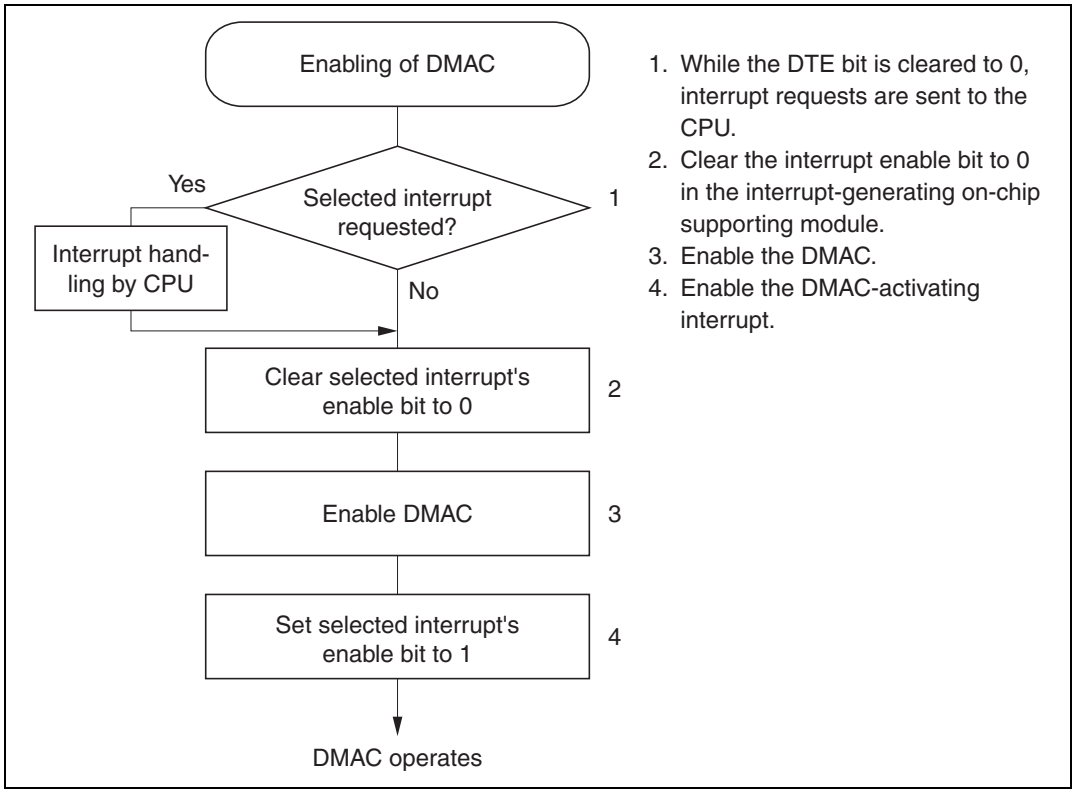


Figure 7.26 Procedure for Enabling DMAC while On-Chip Supporting Module Is Operating (Example)

If the DTE bit is set to 1 but the DTME bit is cleared to 0, the DMAC is halted and the selected activating source cannot generate a CPU interrupt. If the DMAC is halted by an NMI interrupt, for example, the selected activating source cannot generate CPU interrupts. To terminate DMAC operations in this state, clear the DTE bit to 0 to allow CPU interrupts to be requested. To continue DMAC operations, carry out steps 2 and 4 in figure 7.26 before and after setting the DTME bit to 1.

Bit 5—Parity Enable (PE): In asynchronous mode, this bit enables or disables the addition of a parity bit to transmit data, and the checking of the parity bit in receive data. In synchronous mode, the parity bit is neither added nor checked, regardless of the PE bit setting.

Bit 5 PE	Description	
0	Parity bit not added or checked	(Initial value)
1	Parity bit added and checked*	

Note: * When PE bit is set to 1, an even or odd parity bit is added to transmit data according to the even or odd parity mode selection by the O/$\overline{E}$ bit, and the parity bit in receive data is checked to see that it matches the even or odd mode selected by the O/$\overline{E}$ bit.

Bit 4—Parity Mode (O/$\overline{E}$): Selects even or odd parity. The O/$\overline{E}$ bit setting is only valid when the PE bit is set to 1, enabling parity bit addition and checking, in asynchronous mode. The O/$\overline{E}$ bit setting is ignored in synchronous mode, or when parity addition and checking is disabled in asynchronous mode.

Bit 4 O/$\overline{E}$	Description	
0	Even parity*[1]	(Initial value)
1	Odd parity*[2]	

Notes: 1. When even parity is selected, the parity bit added to transmit data makes an even number of 1s in the transmitted character and parity bit combined. Receive data must have an even number of 1s in the received character and parity bit combined.
2. When odd parity is selected, the parity bit added to transmit data makes an odd number of 1s in the transmitted character and parity bit combined. Receive data must have an odd number of 1s in the received character and parity bit combined.

Bit 3—Stop Bit Length (STOP): Selects one or two stop bits in asynchronous mode. This setting is used only in asynchronous mode. In synchronous mod no stop bit is added, so the STOP bit setting is ignored.

Bit 3 STOP	Description	
0	1 stop bit*[1]	(Initial value)
1	2 stop bits*[2]	

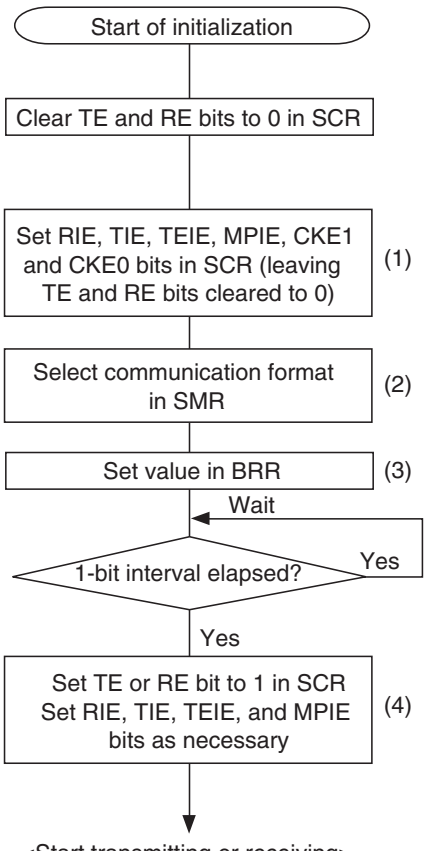
Notes: 1. One stop bit (with value 1) is added to the end of each transmitted character.
2. Two stop bits (with value 1) are added to the end of each transmitted character.

Transmitting and Receiving Data:

- SCI Initialization (Synchronous Mode): Before transmitting or receiving data, clear the TE and RE bits to 0 in SCR, then initialize the SCI as follows.

  When changing the communication mode or format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 sets the TDRE flag to 1 and initializes TSR. Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags, or RDR, which retain their previous contents.

Figure 13.15 shows a sample flowchart for initializing the SCI.



(1) Set the clock source in SCR. Clear the RIE, TIE, TEIE, MPIE, TE, and RE bits to 0.

(2) Select the communication format in SMR.

(3) Write the value corresponding to the bit rate in BRR.
This step is not necessary when an external clock is used.

(4) Wait for at least the interval required to transmit or receive one bit, then set the TE or RE bit to 1 in SCR. Set the RIE, TIE, TEIE, and MPIE bits. Setting the TE or RE bit enables the SCI to use the TxD or RxD pin.

Note: In simultaneous transmitting and receiving, the TE and RE bits should be cleared to 0 or set to 1 simultaneously.

Figure 13.15 Sample Flowchart for SCI Initialization

13.5 Usage Notes

13.5.1 Notes on Use of SCI

Note the following points when using the SCI.

TDR Write and TDRE Flag: The TDRE flag in SSR is a status flag indicating the loading of transmit data from TDR to TSR. The SCI sets the TDRE flag to 1 when it transfers data from TDR to TSR.

Data can be written into TDR regardless of the state of the TDRE flag. If new data is written in TDR when the TDRE flag is 0, the old data stored in TDR will be lost because this data has not yet been transferred to TSR. Before writing transmit data in TDR, be sure to check that the TDRE flag is set to 1.

Simultaneous Multiple Receive Errors: Table 13.13 shows the state of the SSR status flags when multiple receive errors occur simultaneously. When an overrun error occurs the RSR contents are not transferred to RDR, so receive data is lost.

Table 13.13 SSR Status Flags and Transfer of Receive Data

SSR Status Flags				Receive Data Transfer	
RDRF	ORER	FER	PER	RSR → RDR	Receive Errors
1	1	0	0	Not transferred	Overrun error
0	0	1	0	Transferred	Framing error
0	0	0	1	Transferred	Parity error
1	1	1	0	Not transferred	Overrun error + framing error
1	1	0	1	Not transferred	Overrun error + parity error
0	0	1	1	Transferred	Framing error + parity error
1	1	1	1	Not transferred	Overrun error + framing error + parity error

Break Detection and Processing: Break signals can be detected by reading the RxD pin directly when a framing error (FER) is detected. In the break state the input from the RxD pin consists of all 0s, so the FER flag is set and the parity error flag (PER) may also be set. In the break state the SCI receiver continues to operate, so if the FER flag is cleared to 0 it will be set to 1 again.

14.1.2 Block Diagram

Figure 14.1 shows a block diagram of the smart card interface.

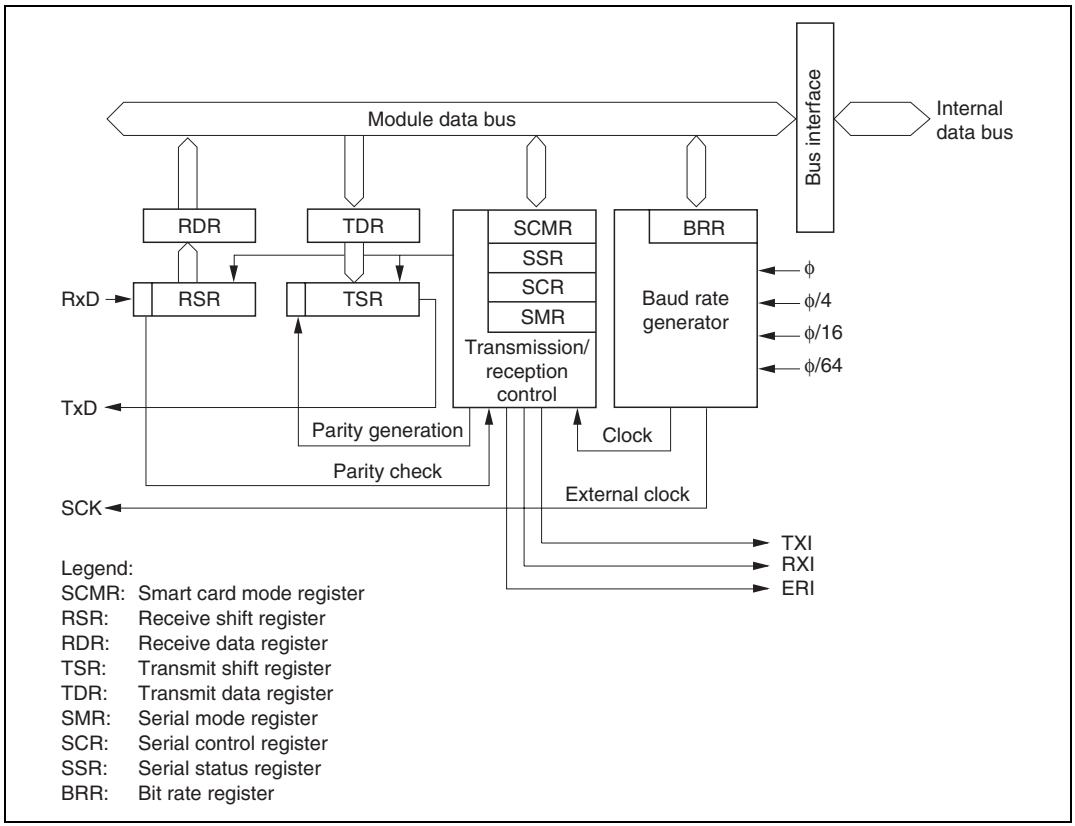


Figure 14.1 Block Diagram of Smart Card Interface

14.1.3 Pin Configuration

Table 14.1 shows the smart card interface pins.

Table 14.1 Smart Card Interface Pins

Pin Name	Abbreviation	I/O	Function
Serial clock pin	SCK	I/O	Clock input/output
Receive data pin	RxD	Input	Receive data input
Transmit data pin	TxD	Output	Transmit data output

14.3.4 Register Settings

Table 14.3 shows a bit map of the registers used in the smart card interface. Bits indicated as 0 or 1 must be set to the value shown. The setting of other bits is described in this section.

Table 14.3 Smart Card Interface Register Settings

		Bit							
Register	**Address**[1]	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
SMR	H'FFFB0	GM	0	1	O/$\overline{E}$	1	0	CKS1	CKS0
BRR	H'FFFB1	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0
SCR	H'FFFB2	TIE	RIE	TE	RE	0	0	CKE1[2]	CKE0
TDR	H'FFFB3	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0
SSR	H'FFFB4	TDRE	RDRF	ORER	ERS	PER	TEND	0	0
RDR	H'FFFB5	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0
SCMR	H'FFFB6	—	—	—	—	SDIR	SINV	—	SMIF

Notes: — Unused bit.
1. Lower 20 bits of the address in advanced mode.
2. When GM is cleared to 0 in SMR, the CKE1 bit must also be cleared to 0.

Serial Mode Register (SMR) Settings: Clear the GM bit to 0 when using the normal smart card interface mode, or set to 1 when using GSM mode. Clear the O/$\overline{E}$ bit to 0 if the smart card is of the direct convention type, or set to 1 if of the inverse convention type.

Bits CKS1 and CKS0 select the clock source of the built-in baud rate generator. See section 14.3.5, Clock.

Bit Rate Register (BRR) Settings: BRR is used to set the bit rate. See section 14.3.5, Clock, for the method of calculating the value to be set.

Serial Control Register (SCR) Settings: The TIE, RIE, TE, and RE bits have their normal serial communication functions. See section 13, Serial Communication Interface, for details. The CKE1 and CKE0 bits specify clock output. To disable clock output, clear these bits to 00; to enable clock output, set these bits to 01. Clock output is not performed when the GM bit is set to 1 in SMR. Clock output can also be fixed low or high.

Smart Card Mode Register (SCMR) Settings: Clear both the SDIR bit and SINV bit cleared to 0 if the smart card is of the direct convention type, and set both to 1 if of the inverse convention type. To use the smart card interface, set the SMIF bit to 1.

14.4 Usage Notes

The following points should be noted when using the SCI as a smart card interface.

Receive Data Sampling Timing and Receive Margin in Smart Card Interface Mode: In smart card interface mode, the SCI operates on a base clock with a frequency of 372 times the transfer rate. In reception, the SCI synchronizes internally with the fall of the start bit, which it samples on the base clock. Receive data is latched at the rising edge of the 186th base clock pulse. The timing is shown in figure 14.11.

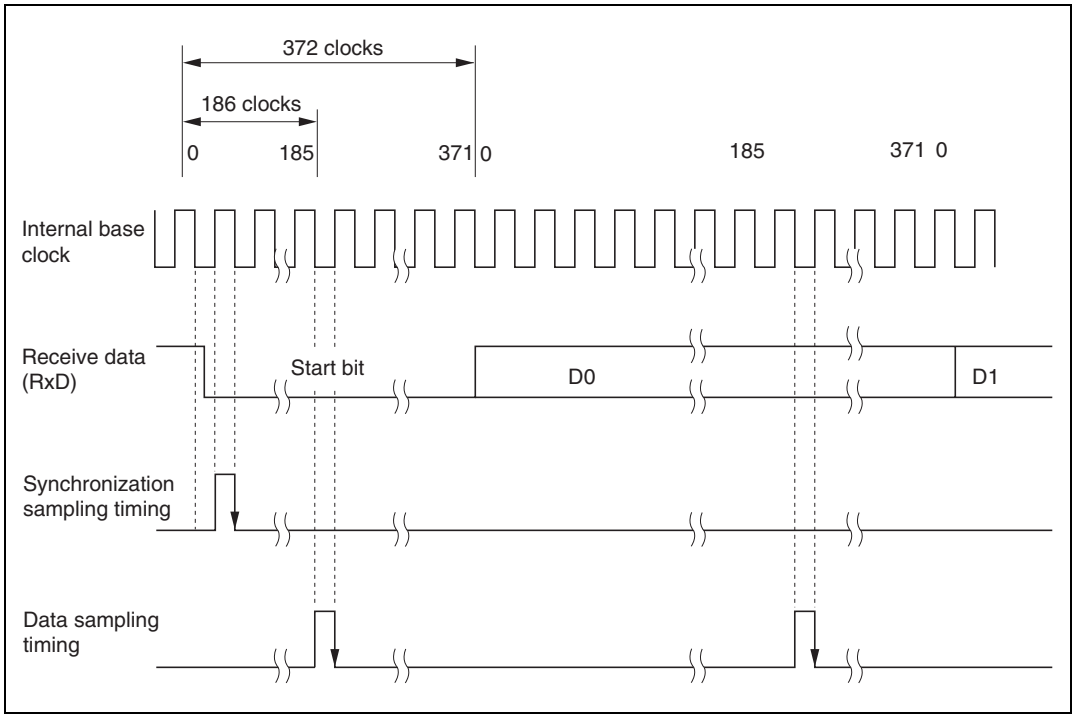


Figure 14.11 Receive Data Sampling Timing in Smart Card Interface Mode

The receive margin can therefore be expressed as follows.

Receive margin in smart card interface mode:

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}\,(1 + F) \right| \times 100\%$$

M: Receive margin (%)
N: Ratio of clock frequency to bit rate (N = 372)

Bit 7—Trigger Enable (TRGE): Enables or disables starting of A/D conversion by an external trigger or 8-bit timer compare match.

Bit 7 TRGE	Description	
0	Starting of A/D conversion by an external trigger or 8-bit timer compare match is disabled	(Initial value)
1	A/D conversion is started at the falling edge of the external trigger signal ($\overline{\text{ADTRG}}$) or by an 8-bit timer compare match	

External trigger pin and 8-bit timer selection are performed by the 8-bit timer. For details, see section 10, 8-Bit Timers.

Bits 6 to 1—Reserved: These bits cannot be modified and are always read as 1.

Bit 0—Reserved: This bit can be read or written, but should not be set to 1.

16.1.2 Block Diagram

Figure 16.1 shows a block diagram of the D/A converter.

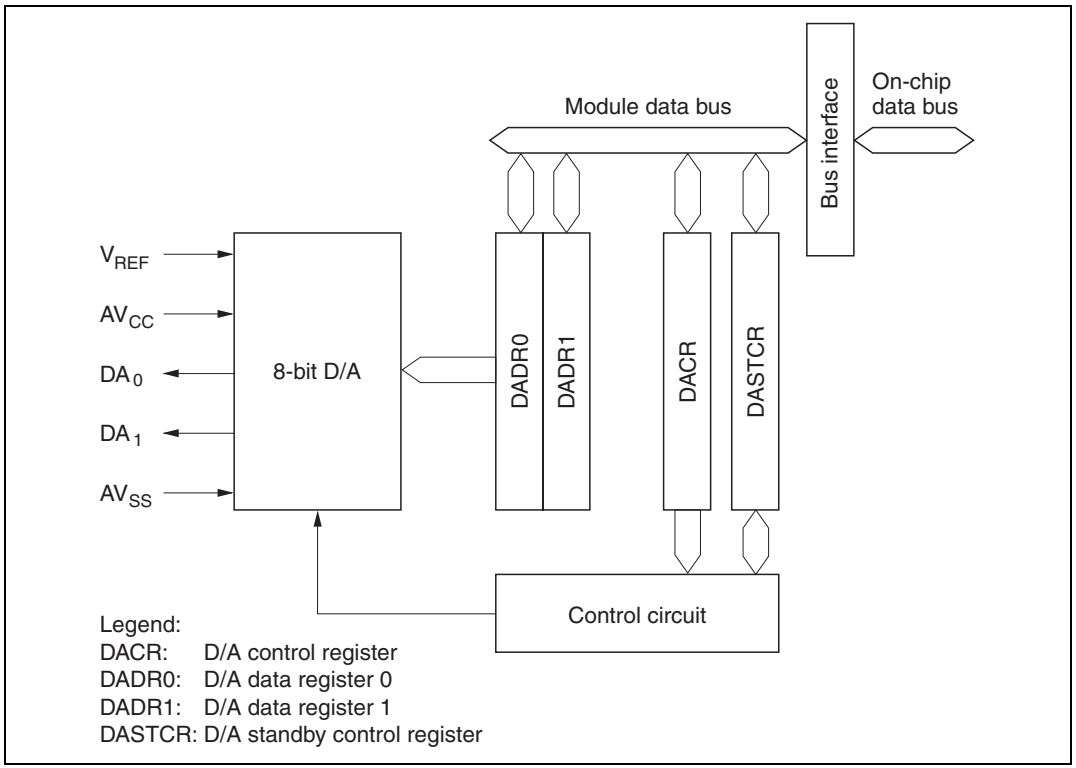


Figure 16.1 D/A Converter Block Diagram

16.1.3 Pin Configuration

Table 16.1 summarizes the D/A converter's input and output pins.

Table 16.1 D/A Converter Pins

Pin Name	Abbreviation	I/O	Function
Analog power supply pin	AV_{CC}	Input	Analog power supply and reference voltage
Analog ground pin	AV_{SS}	Input	Analog ground and reference voltage
Analog output pin 0	DA_0	Output	Analog output, channel 0
Analog output pin 1	DA_1	Output	Analog output, channel 1
Reference voltage input pin	V_{REF}	Input	Analog reference voltage

19.4 Software Standby Mode

19.4.1 Transition to Software Standby Mode

To enter software standby mode, execute the SLEEP instruction while the SSBY bit is set to 1 in SYSCR.

In software standby mode, current dissipation is reduced to an extremely low level because the CPU, clock, and on-chip supporting modules all halt. The DMAC and on-chip supporting modules are reset and halted. As long as the specified voltage is supplied, however, CPU register contents and on-chip RAM data are retained. The settings of the I/O ports and DRAM interface* are also held.

When the WDT is used as a watchdog timer (WT/$\overline{IT}$ = 1), the TME bit must be cleared to 0 before setting SSBY. Also, when setting TME to 1, SSBY should be cleared to 0.

Clear the BRLE bit in BRCR (inhibiting bus release) before making a transition to software standby mode.

Note: * RTCNT and bits 7 and 6 of RTMCSR are initialized. Other bits and registers hold their previous states.

19.4.2 Exit from Software Standby Mode

Software standby mode can be exited by input of an external interrupt at the NMI, $\overline{IRQ_0}$, $\overline{IRQ_1}$, or $\overline{IRQ_2}$ pin, or by input at the $\overline{RES}$ or $\overline{STBY}$ pin.

Exit by Interrupt: When an NMI, IRQ_0, IRQ_1, or IRQ_2 interrupt request signal is received, the clock oscillator begins operating. After the oscillator settling time selected by bits STS2 to STS0 in SYSCR, stable clock signals are supplied to the entire chip, software standby mode ends, and interrupt exception handling begins. Software standby mode is not exited if the interrupt enable bits of interrupts IRQ_0, IRQ_1, and IRQ_2 are cleared to 0, or if these interrupts are masked in the CPU.

Exit by $\overline{RES}$ Input: When the $\overline{RES}$ input goes low, the clock oscillator starts and clock pulses are supplied immediately to the entire chip. The $\overline{RES}$ signal must be held low long enough for the clock oscillator to stabilize. When $\overline{RES}$ goes high, the CPU starts reset exception handling.

Exit by $\overline{STBY}$ Input: Low input at the $\overline{STBY}$ pin causes a transition to hardware standby mode.

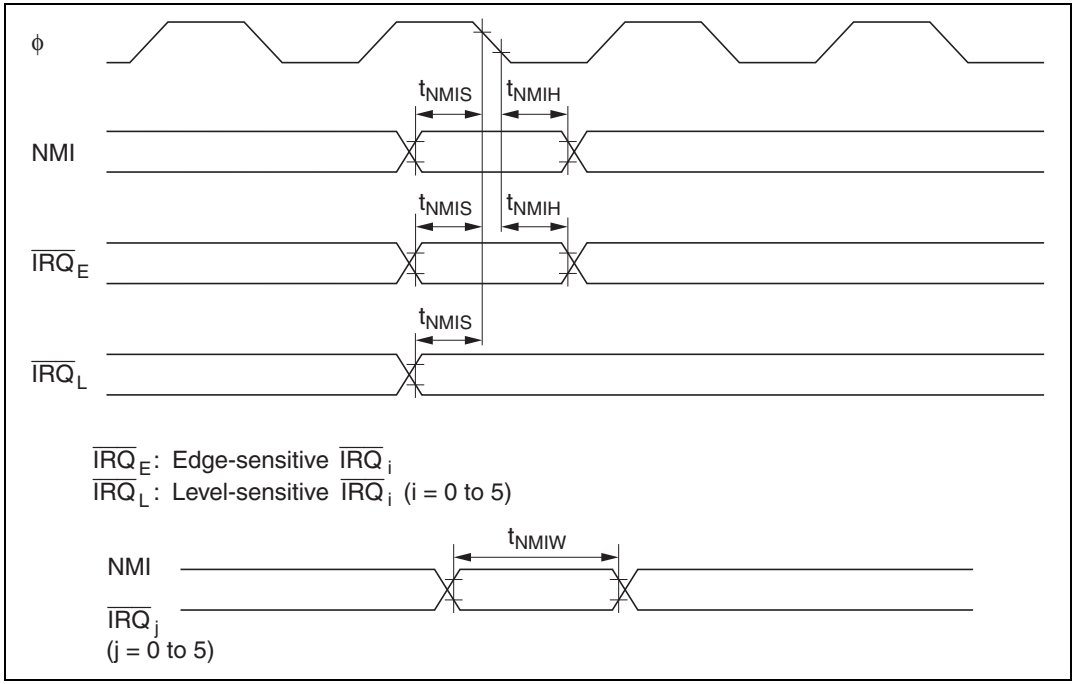


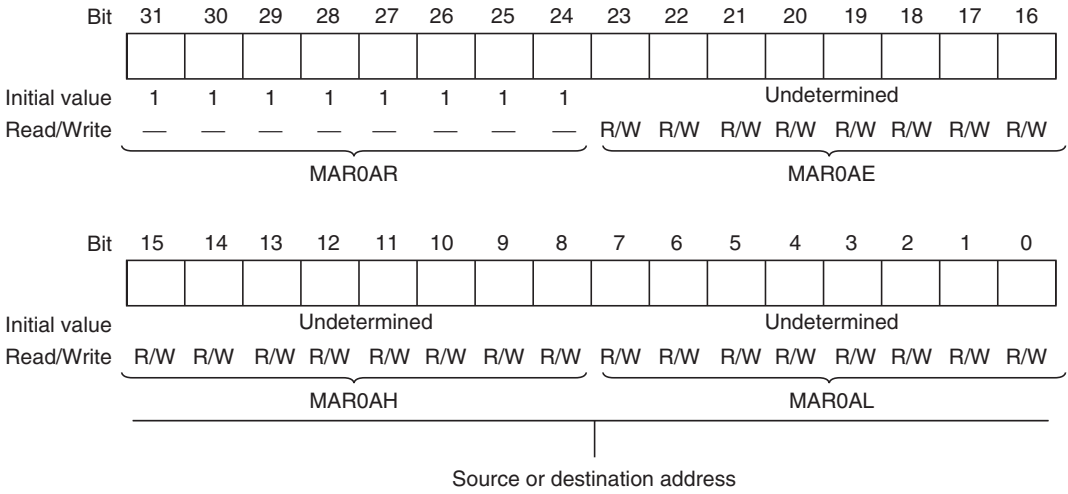
Figure 20.7 Interrupt Input Timing

20.3.3 Bus Timing

Bus timing is shown as follows:

- Basic bus cycle: two-state access
  Figure 20.8 shows the timing of the external two-state access cycle.
- Basic bus cycle: three-state access
  Figure 20.9 shows the timing of the external three-state access cycle.
- Basic bus cycle: three-state access with one wait state
  Figure 20.10 shows the timing of the external three-state access cycle with one wait state inserted.
- Burst ROM access timing: burst cycle two-state
  Figure 20.11 shows the timing of the burst cycle two-state access.
- Burst ROM access timing: burst cycle three-state
  Figure 20.12 shows the timing of the burst cycle three-state access.
- Bus-release mode timing
  Figure 20.13 shows the bus-release mode timing.

MAR0A R/E/H/L—Memory Address Register 0A R/E/H/L H'FFF20 H'FFF21 H'FFF22 H'FFF23 DMAC0



16TCNT1H/L—Timer Counter 1H/L **H'FFF72 H'FFF73** **16-Bit Timer Channel 1**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	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

Note: Bit functions are the same as for 16-bit timer channel 0.

GRA1H/L—General Register A1 H/L **H'FFF74 H'FFF75** **16-Bit Timer Channel 1**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	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

Note: Bit functions are the same as for 16-bit timer channel 0.

GRB1H/L—General Register B1 H/L **H'FFF76 H'FFF77** **16-Bit Timer Channel 1**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	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

Note: Bit functions are the same as for 16-bit timer channel 0.

16TCR2—Timer Control Register 2 **H'FFF78** **16-Bit Timer Channel 2**

Bit	7	6	5	4	3	2	1	0
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	1	0	0	0	0	0	0	0
Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Notes: 1. Bit functions are the same as for 16-bit timer channel 0.
2. The settings of bits CKEG1 and CKEG0 and bits TPSC2 to TPSC0 in 16TCR2 are invalid when phase counting mode is selected for channel 2.