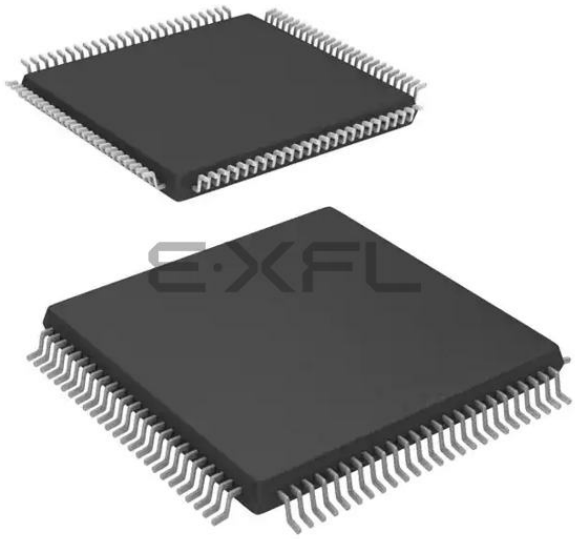


Welcome to E-XFL.COM

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	H8/300H
Core Size	16-Bit
Speed	25MHz
Connectivity	SCI, SmartCard
Peripherals	DMA, PWM, WDT
Number of I/O	70
Program Memory Size	384KB (384K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
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-TQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/df3028x25v

6.3 Operation

6.3.1 Area Division

The external address space is divided into areas 0 to 7. Each area has a size of 128 kbytes in the 1-Mbyte modes, or 2-Mbytes in the 16-Mbyte modes. Figure 6.2 shows a general view of the memory map.

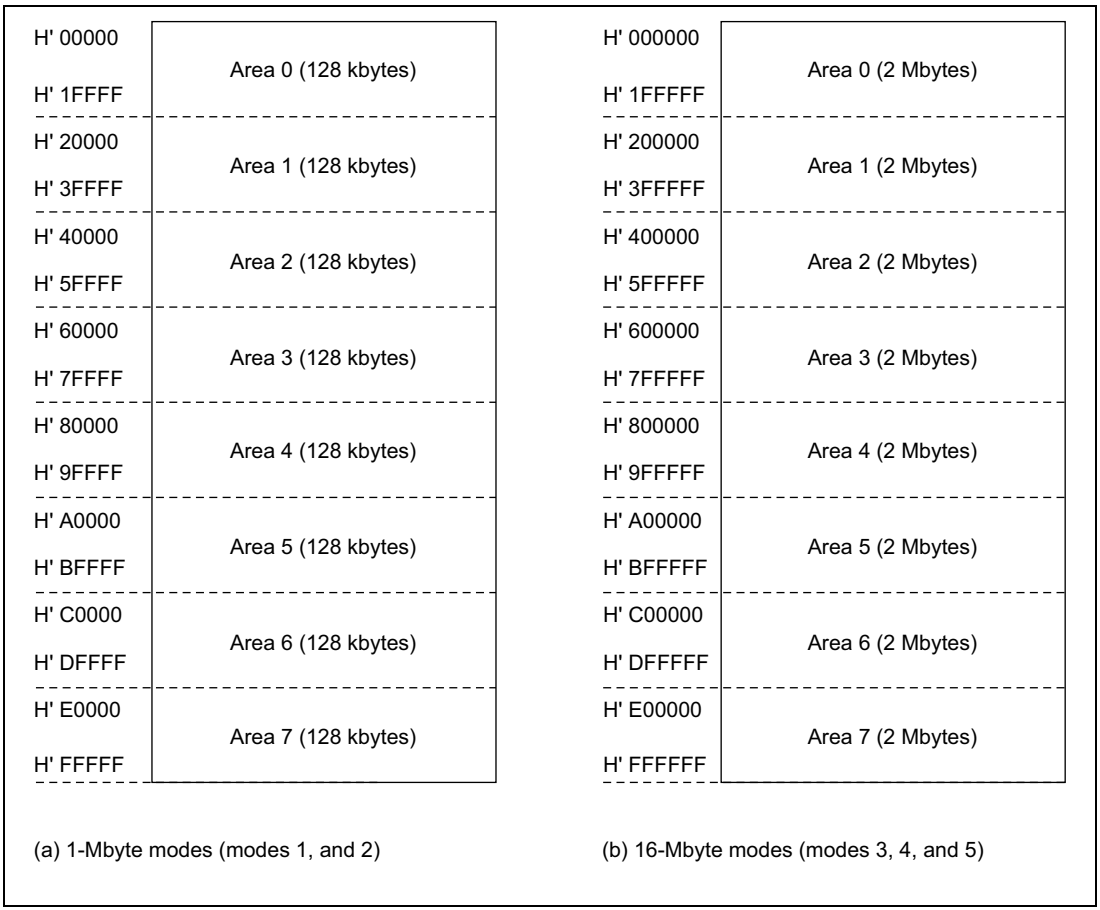


Figure 6.2 Access Area Map for Each Operating Mode

Chip select signals ($\overline{CS}_0$ to $\overline{CS}_7$) can be output for areas 0 to 7. The bus specifications for each area are selected in ABWCR, ASTCR, WCRH, and WCRL.

In 16-Mbyte mode, the area division units can be selected with the RDEA bit in BCR.

Areas 2 to 5: In external expansion mode, areas 2 to 5 are entirely external space.
When area 2 to 5 external space is accessed, signals $\overline{CS}_2$ to $\overline{CS}_5$ can be output.
Basic bus interface or DRAM interface can be selected for areas 2 to 5. With the DRAM interface, signals $\overline{CS}_2$ to $\overline{CS}_5$ are used as $\overline{RAS}$ signals.
The size of areas 2 to 5 is 128 kbytes in modes 1 and 2, and 2 Mbytes in modes 3, 4, and 5.

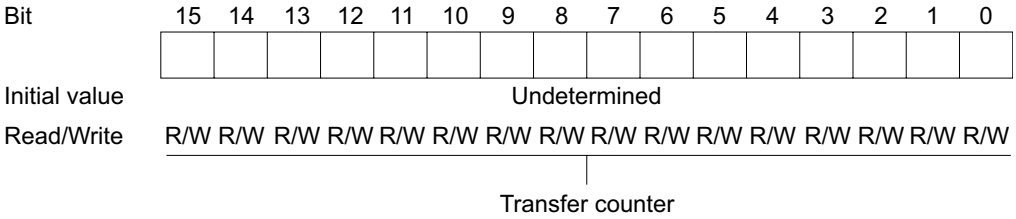
Area 7: Area 7 includes the on-chip RAM and registers. In external expansion mode, the space excluding the on-chip RAM and registers is external space. The on-chip RAM is enabled when the RAME bit in the system control register (SYSCR) is set to 1; when the RAME bit is cleared to 0, the on-chip RAM is disabled and the corresponding space becomes external space .
When area 7 external space is accessed, the $\overline{CS}_7$ signal can be output.
Only the basic bus interface can be used for the area 7 memory interface.
The size of area 7 is 128 kbytes in modes 1 and 2, and 2 Mbytes in modes 3, 4, and 5.

7.3.3 Execute Transfer Count Registers (ETCR)

An execute transfer count register (ETCR) is a 16-bit readable/writable register that specifies the number of transfers to be executed. The functions of these registers differ between normal mode and block transfer mode.

Normal Mode

- ETCRA



ETCRB: Is not used in normal mode.

In normal mode ETCRA functions as a 16-bit transfer counter. The count is decremented by 1 each time one transfer is executed. The transfer ends when the count reaches H'0000. ETCRB is not used.

Figure 7.17 shows the timing when the DMAC is activated by level-sensitive low $\overline{\text{DREQ}}$ input in normal mode.

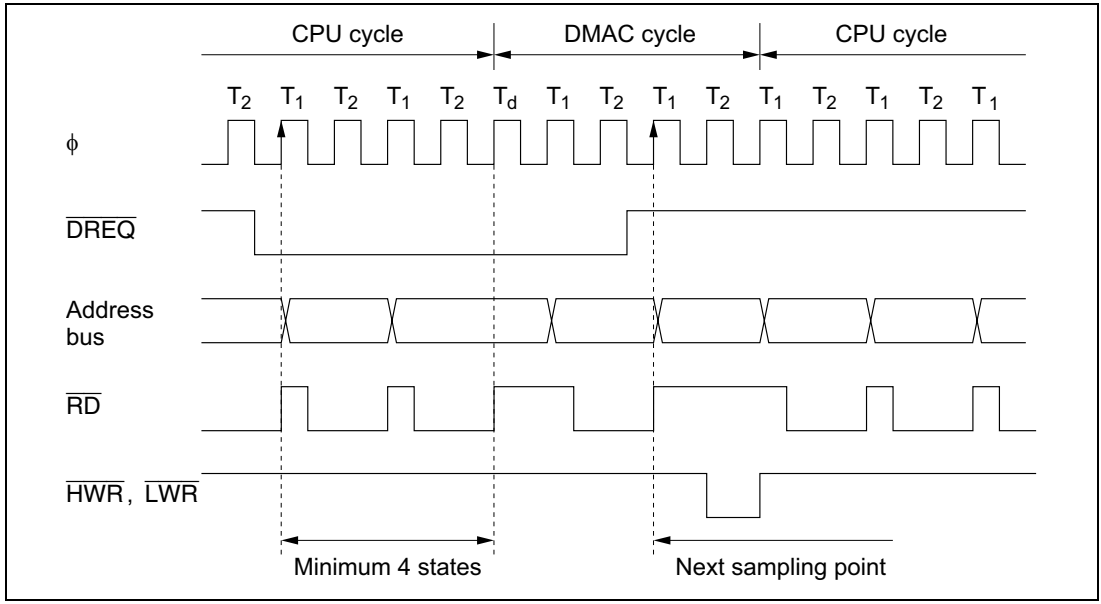


Figure 7.17 Timing of DMAC Activation by Low $\overline{\text{DREQ}}$ Level in Normal Mode

8.3.2 Register Descriptions

Table 8.3 summarizes the registers of port 2.

Table 8.3 Port 2 Registers

Address*	Name	Abbreviation	R/W	Initial Value	
				Modes 1 to 4	Modes 5 to 7
H'EE001	Port 2 data direction register	P2DDR	W	H'FF	H'00
H'FFFD1	Port 2 data register	P2DR	R/W	H'00	H'00
H'EE03C	Port 2 input pull-up MOS control register	P2PCR	R/W	H'00	H'00

Note: * Lower 20 bits of the address in advanced mode.

Port 2 Data Direction Register (P2DDR): P2DDR is an 8-bit write-only register that can select input or output for each pin in port 2.

Bit		7	6	5	4	3	2	1	0
		$P2_7DDR$	$P2_6DDR$	$P2_5DDR$	$P2_4DDR$	$P2_3DDR$	$P2_2DDR$	$P2_1DDR$	$P2_0DDR$
Modes 1 to 4	Initial value	1	1	1	1	1	1	1	1
	Read/Write	—	—	—	—	—	—	—	—
Modes 5 to 7	Initial value	0	0	0	0	0	0	0	0
	Read/Write	W	W	W	W	W	W	W	W

Port 2 data direction 7 to 0
These bits select input or output for port 2 pins

Modes 1 to 4 (Expanded Modes with On-Chip ROM Disabled): P2DDR values are fixed at 1. Port 2 functions as an address bus.

Mode 5 (Expanded Modes with On-Chip ROM Enabled): Following a reset, port 2 is an input port. A pin in port 2 becomes an address output pin if the corresponding P2DDR bit is set to 1, and a generic input port if this bit is cleared to 0.

Modes 6 and 7 (Single-Chip Mode): Port 2 functions as an input/output port. A pin in port 2 becomes an output port if the corresponding P2DDR bit is set to 1, and an input port if this bit is cleared to 0.

In modes 1 to 4, P2DDR bits are always read as 1, and cannot be modified.

In modes 5 to 7, P2DDR is a write-only register. Its value cannot be read. All bits return 1 when read.

Modes 6 and 7 (Single-Chip Mode): Port 8 is a generic input/output port. A pin in port 8 becomes an output port if the corresponding P8DDR bit is set to 1, and an input port if this bit is cleared to 0.

P8DDR is a write-only register. Its value cannot be read. All bits return 1 when read.

P8DDR is initialized to H'F0 in modes 1 to 4, and to H'E0 in modes 5 to 7, by a reset and in hardware standby mode. In software standby mode P8DDR retains its previous setting. Therefore, if a transition is made to software standby mode while port 8 is functioning as an input/output port and a P8DDR bit is set to 1, the corresponding pin maintains its output state.

Port 8 Data Register (P8DR): P8DR is an 8-bit readable/writable register that stores output data for port 8. When port 8 functions as an output port, the value of this register is output. When a bit in P8DDR is set to 1, if port 8 is read the value of the corresponding P8DR bit is returned. When a bit in P8DDR is cleared to 0, if port 8 is read the corresponding pin logic level is read.

Bits 7 to 5 are reserved. They are fixed at 1, and cannot be modified.

Bit	7	6	5	4	3	2	1	0
	—	—	—	$P8_4$	$P8_3$	$P8_2$	$P8_1$	$P8_0$
Initial value	1	1	1	0	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W	R/W

Reserved bits

Port 8 data 4 to 0
These bits store data for port 8 pins

P8DR is initialized to H'E0 by a reset and in hardware standby mode. In software standby mode it retains its previous setting.

8.11 Port A

8.11.1 Overview

Port A is an 8-bit input/output port that is also used for output (TP_7 to TP_0) from the programmable timing pattern controller (TPC), input and output, ($TIOCB_2$, $TIOCA_2$, $TIOCB_1$, $TIOCA_1$, $TIOCB_0$, $TIOCA_0$, TCLKD, TCLKC, TCLKB, TCLKA) by the 16-bit timer, input (TCLKD, TCLKC, TCLKB, TCLKA) to the 8-bit timer, output ($\overline{TEND}_1$, $\overline{TEND}_0$) from the DMA controller (DMAC), and address output (A_{23} to A_{20}). A reset or hardware standby transition leaves port A as an input port, except that in modes 3 and 4, one pin is always used for A_{20} output. See table 8.19 to 8.21 for the selection of pin functions.

Usage of pins for TPC, 16-bit timer, 8-bit timer, and DMAC input and output is described in the sections on those modules. For output of address bits A_{23} to A_{20} in modes 3, 4, and 5, see section 6.2.4, Bus Release Control Register (BRCR). Pins not assigned to any of these functions are available for generic input/output. Figure 8.10 shows the pin configuration of port A.

Pins in port A can drive one TTL load and a 30-pF capacitive load. They can also drive a darlington transistor pair. Port A has Schmitt-trigger inputs.

8.11.2 Register Descriptions

Table 8.18 summarizes the registers of port A.

Table 8.18 Port A Registers

Address*	Name		R/W	Initial Value: Modes 1, 2, 5, 6, and 7	Initial Value: Modes 3 and 4
H'EE009	Port A data direction register	PADDR	W	H'00	H'80
H'FFFD9	Port A data register	PADR	R/W	H'00	H'00

Note: * Lower 20 bits of the address in advanced mode.

Port A Data Direction Register (PADDR): PADDR is an 8-bit write-only register that can select input or output for each pin in port A. When pins are used for TPC output, the corresponding PADDR bits must also be set.

Bit		7	6	5	4	3	2	1	0
		PA_7DDR	PA_6DDR	PA_5DDR	PA_4DDR	PA_3DDR	PA_2DDR	PA_1DDR	PA_0DDR
Modes 3 and 4	Initial value	1	0	0	0	0	0	0	0
	Read/Write	—	W	W	W	W	W	W	W
Modes 1, 2, 5, 6, and 7	Initial value	0	0	0	0	0	0	0	0
	Read/Write	W	W	W	W	W	W	W	W

Port A data direction 7 to 0
These bits select input or output for port A pins

The pin functions that can be selected for pins PA_7 to PA_4 differ between modes 1, 2, 6, and 7, and modes 3 to 5. For the method of selecting the pin functions, see tables 8.19 and 8.20.

The pin functions that can be selected for pins PA_3 to PA_0 are the same in modes 1 to 7. For the method of selecting the pin functions, see table 8.21.

When port A functions as an input/output port, a pin in port A becomes an output port if the corresponding PADDR bit is set to 1, and an input port if this bit is cleared to 0. In modes 3 and 4, PA_7DDR is fixed at 1 and PA_7 functions as the A_{20} address output pin.

PADDR is a write-only register. Its value cannot be read. All bits return 1 when read.

Pin	Pin Functions and Selection Method
$PB_0/TP_8/TMO_0/\overline{CS_7}$	Bits OIS3/2 and OS1/0 in 8TCSR0, bit CS7E in CSCR, bit NDER8 in NDERB, and bit PB_0DDR select the pin function as follows.

OIS3/2 and OS1/0	All 0				Not all 0
CS7E	0			1	—
PB_0DDR	0	1	1	—	—
NDER8	—	0	1	—	—
Pin function	PB_0 input	PB_0 output	TP_8 output	$\overline{CS_7}$ output	TMO_0 output

Bit 7—Transmit Interrupt Enable (TIE): Enables or disables the transmit-data-empty interrupt (TXI) requested when the TDRE flag in SSR is set to 1 due to transfer of serial transmit data from TDR to TSR.

Bit 7 TIE	Description	
0	Transmit-data-empty interrupt request (TXI) is disabled*	(Initial value)
1	Transmit-data-empty interrupt request (TXI) is enabled	

Note: * TXI interrupt requests can be cleared by reading the value 1 from the TDRE flag, then clearing it to 0; or by clearing the TIE bit to 0.

Bit 6—Receive Interrupt Enable (RIE): Enables or disables the receive-data-full interrupt (RXI) requested when the RDRF flag in SSR is set to 1 due to transfer of serial receive data from RSR to RDR; also enables or disables the receive-error interrupt (ERI).

Bit 6 RIE	Description
0	Receive-data-full (RXI) and receive-error (ERI) interrupt requests are disabled* (Initial value)
1	Receive-data-full (RXI) and receive-error (ERI) interrupt requests are enabled

Note: * RXI and ERI interrupt requests can be cleared by reading the value 1 from the RDRF, FER, PER, or ORER flag, then clearing the flag to 0; or by clearing the RIE bit to 0.

Bit 5—Transmit Enable (TE): Enables or disables the start of SCI serial transmitting operations.

Bit 5 TE	Description	
0	Transmitting disabled*1	(Initial value)
1	Transmitting enabled*2	

Notes: 1. The TDRE flag is fixed at 1 in SSR.
2. In the enabled state, serial transmission starts when the TDRE flag in SSR is cleared to 0 after writing of transmit data into TDR. Select the transmit format in SMR before setting the TE bit to 1.

Figure 13.19 shows an example of SCI receive operation.

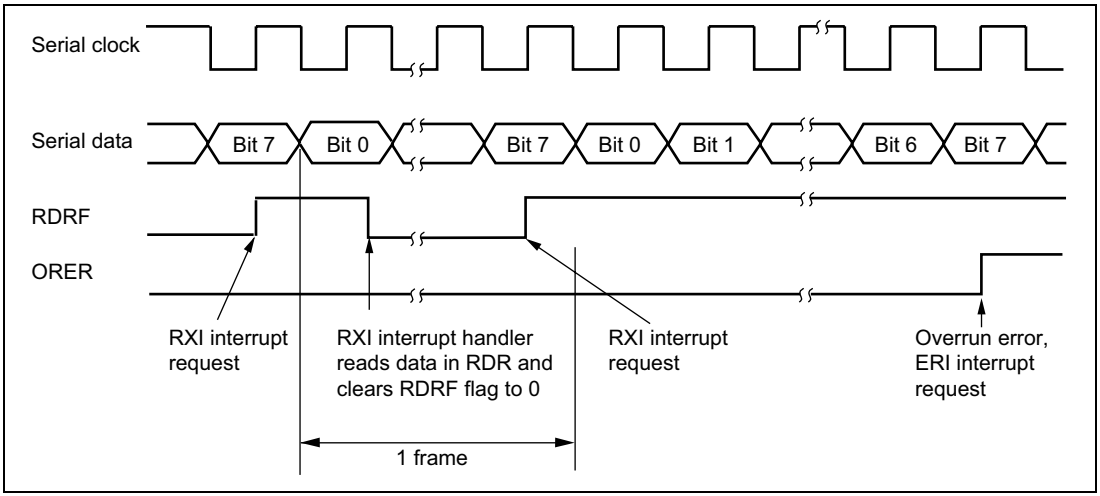


Figure 13.19 Example of SCI Receive Operation

- If a parity error is detected during reception, a low error signal level is output for a1 etu period 10.5 etu after the start bit.
- If an error signal is detected during transmission, the same data is transmitted automatically after the elapse of 2 etu or longer.
- Only asynchronous communication is supported; there is no synchronous communication function.

14.3.2 Pin Connections

Figure 14.2 shows a pin connection diagram for the smart card interface.

In communication with a smart card, since both transmission and reception are carried out on a single data transmission line, the TxD pin and RxD pin should both be connected to this line. The data transmission line should be pulled up to V_{CC} with a resistor.

When the smart card uses the clock generated on the smart card interface, the SCK pin output is input to the CLK pin of the smart card. If the smart card uses an internal clock, this connection is unnecessary.

The reset signal should be output from one of the H8/3028 Group's generic ports.

In addition to these pin connections, power and ground connections will normally also be necessary.

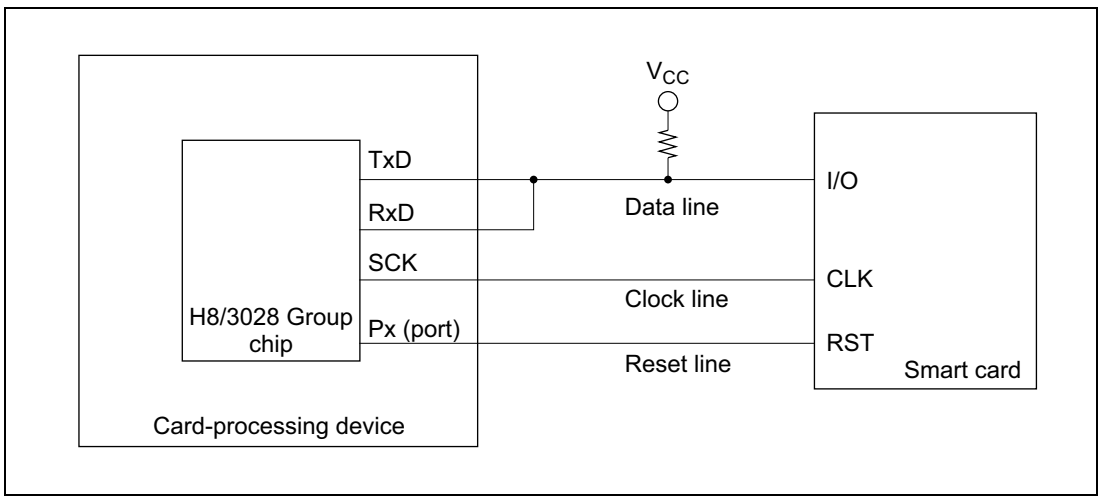


Figure 14.2 Smart Card Interface Connection Diagram

Note: A loop-back test can be performed by setting both RE and TE to 1 without connecting a smart card.

Examples of Operation in GSM Mode: When switching between smart card interface mode and software standby mode, use the following procedures to maintain the clock duty cycle.

- Switching from smart card interface mode to software standby mode

1. Set the $P9_4$ data register (DR) and data direction register (DDR) to the values for the fixed output state in software standby mode.
2. Write 0 in the TE and RE bits in the serial control register (SCR) to stop transmit/receive operations. At the same time, set the CKE1 bit to the value for the fixed output state in software standby mode.
3. Write 0 in the CKE0 bit in SCR to stop the clock.
4. Wait for one serial clock cycle. During this period, the duty cycle is preserved and clock output is fixed at the specified level.
5. Write H'00 in the serial mode register (SMR) and smart card mode register (SCMR).
6. Make the transition to the software standby state.

- Returning from software standby mode to smart card interface mode

1. Clear the software standby state.
2. Set the CKE1 bit in SCR to the value for the fixed output state at the start of software standby (the current $P9_4$ pin state).
3. Set smart card interface mode and output the clock. Clock signal generation is started with the normal duty cycle.

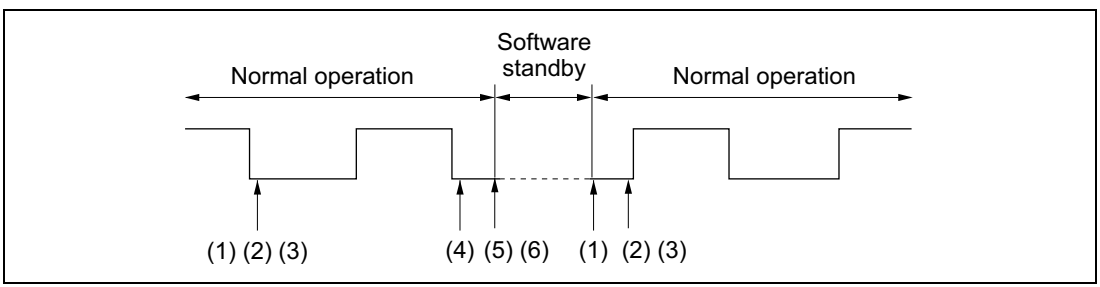


Figure 14.10 Procedure for Stopping and Restarting the Clock

Use the following procedure to secure the clock duty cycle after powering on.

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

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)
D: Clock duty cycle (L = 0 to 1.0)
L: Frame length (L =10)
F: Absolute deviation of clock frequency

From the above equation, if F = 0 and D = 0.5, the receive margin is as follows.

When D = 0.5 and F = 0:

$$M = (0.5 - 1/2 \times 372) \times 100\%$$
$$= 49.866\%$$

Retransmission: Retransmission is performed by the SCI in receive mode and transmit mode as described below.

- Retransmission when SCI is in Receive Mode
  Figure 14.12 illustrates retransmission when the SCI is in receive mode.

1. If an error is found when the received parity bit is checked, the PER bit is automatically set to 1. If the RIE bit in SCR is set to the enable state, an ERI interrupt is requested. The PER bit should be cleared to 0 in SSR before the next parity bit sampling timing.
2. The RDRF bit in SSR is not set for the frame in which the error has occurred.
3. If no error is found when the received parity bit is checked, the PER bit is not set to 1 in SSR.
4. If no error is found when the received parity bit is checked, the receive operation is assumed to have been completed normally, and the RDRF bit is automatically set to 1 in SSR. If the RIE bit in SCR is set to the enable state, an RXI interrupt is requested. If RXI is enabled as a DMA transfer activation source, the RDR contents can be read automatically. When the DMAC reads the RDR data, the RDRF flag is automatically cleared to 0.
5. When a normal frame is received, the data pin is held in the high-impedance state at the error signal transmission timing.

15.4.2 Scan Mode (SCAN = 1)

Scan mode is useful for monitoring analog inputs in a group of one or more channels. When the ADST bit is set to 1 by software or external trigger input, A/D conversion starts on the first channel in the group (AN_0 when CH2 = 0, AN_4 when CH2 = 1). When two or more channels are selected, after conversion of the first channel ends, conversion of the second channel (AN_1 or AN_5) starts immediately. A/D conversion continues cyclically on the selected channels until the ADST bit is cleared to 0. The conversion results are transferred for storage into the A/D data registers corresponding to the channels.

When the mode or analog input channel selection must be changed during analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1. A/D conversion will start again from the first channel in the group. The ADST bit can be set at the same time as the mode or channel selection is changed.

Typical operations when three channels in group 0 (AN_0 to AN_2) are selected in scan mode are described next. Figure 15.4 shows a timing diagram for this example.

1. Scan mode is selected (SCAN = 1), scan group 0 is selected (CH2 = 0), analog input channels AN_0 to AN_2 are selected (CH1 = 1, CH0 = 0), and A/D conversion is started (ADST = 1).
2. When A/D conversion of the first channel (AN_0) is completed, the result is transferred into ADDRA. Next, conversion of the second channel (AN_1) starts automatically.
3. Conversion proceeds in the same way through the third channel (AN_2).
4. When conversion of all selected channels (AN_0 to AN_2) is completed, the ADF flag is set to 1 and conversion of the first channel (AN_0) starts again. If the ADIE bit is set to 1, an ADI interrupt is requested at this time.
5. Steps 2 to 4 are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops. After that, if the ADST bit is set to 1, A/D conversion starts again from the first channel (AN_0).

18.3 Flash Memory Version Register Description

18.3.1 Flash Memory Control Register 1 (FLMCR1)

Bit	7	6	5	4	3	2	1	0
	FWE	SWE	ESU	PSU	EV	PV	E	P
Initial value	—*	0	0	0	0	0	0	0
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Determined by the state of the FWE pin.

FLMCR1 is an 8-bit register used for flash memory operating mode control.

Program-verify mode or erase-verify mode for addresses H'00000 to H'5FFFF is entered by setting the SWE bit when FWE = 1, then setting the PV or EV bit. Program mode for addresses H'00000 to H'5FFFF is entered by setting the SWE bit when FWE = 1, then setting the PSU bit, and finally setting the P bit. Erase mode for addresses H'00000 to H'5FFFF is entered by setting the SWE bit when FWE = 1, then setting the ESU bit, and finally setting the E bit. FLMCR1 is initialized by a reset, and in hardware standby mode and software standby mode. Its initial value is H'80 when a high level is input to the FWE pin, and H'00 when a low level is input. In mode 6 the FWE pin must be fixed low since flash memory on-board programming modes are not supported. When the on-chip flash memory is disabled, a read access to this register will return H'00, and writes are invalid.

When setting bits 6 to 0 in this register, one bit must be set one at a time. Writes to the SWE bit in FLMCR1 are enabled only when FWE = 1; writes to bits ESU, PSU, EV, and PV only when FWE = 1 and SWE = 1; writes to the E bit only when FWE = 1, SWE = 1, and ESU = 1; and writes to the P bit only when FWE = 1, SWE = 1, and PSU = 1.

Notes: 1. The programming and erase flowcharts must be followed when setting the bits in this register to prevent erroneous programming or erasing.
2. Transitions are made to program mode, erase mode, program-verify mode, and erase-verify mode according to the settings in this register. When reading flash memory as normal on-chip ROM, bits 6 to 0 in this register must be cleared.

Bit 7—Flash Write Enable (FWE): Sets hardware protection against flash memory programming/erasing.

Bit 7 FWE	Description
0	When a low level is input to the FWE pin (hardware-protected state)
1	When a high level is input to the FWE pin

- Same trigger for TPC output groups 0 and 1
  - — Address H'FFFA5

Bit	7	6	5	4	3	2	1	0
	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0
Initial value	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

Store the next output data for TPC output group 1 | Store the next output data for TPC output group 0

  - — Address H'FFFA7

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	—	—	—	—

- Different triggers for TPC output groups 0 and 1
  - — Address H'FFFA5

Bit	7	6	5	4	3	2	1	0
	NDR7	NDR6	NDR5	NDR4	—	—	—	—
Initial value	0	0	0	0	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	—	—	—	—

Store the next output data for TPC output group 1

  - — Address H'FFFA7

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	NDR3	NDR2	NDR1	NDR0
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Store the next output data for TPC output group 0

Pin Name	Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus-Released Mode	Program Execution Mode
PA_6 to PA_4	3 to 5	T	T	• Address output[*5] (SSOE=0) T (SSOE=1) Keep • Otherwise[*6] Keep	• Address output[*5] T • Otherwise[*6] Keep	• Address output A_{23} to A_{21} • Otherwise I/O port
PA_7	1, 2	T	T	Keep	Keep	I/O port
	3, 4	L	T	(SSOE=0) T (SSOE=1) Keep	T	A_{20}
	5	L	T	• When A20E = 0 SSOE = 0 T SSOE = 1 Keep • When A20E = 1 Keep	• When A20E = 0 T • When A20E = 1 Keep	• When A20E = 0 A_{20} • When A20E = 1 I/O port
	6, 7	T	T	Keep	—	I/O port
PB_1 to PB_0	1 to 5	T	T	• CS output[*7] (SSOE=0) T (SSOE=1) H • Otherwise[*8] Keep	• CS output[*7] T • Otherwise[*8] Keep	• CS output $\overline{CS}_7$ to $\overline{CS}_6$ • Otherwise I/O port
	6, 7	T	T	Keep	—	I/O port
PB_2	1 to 5	T	T	• RAS_5 output[*9] (SSOE=0) T (SSOE=1) H • CS output[*10] (SSOE=0) T (SSOE=1) H • Otherwise[*11] Keep	• RAS_5 output[*9] T • CS output[*10] T • Otherwise[*11] Keep	• RAS_5 output $\overline{RAS}_5$ • CS output $\overline{CS}_5$ • Otherwise I/O port
	6, 7	T	T	Keep	—	I/O port