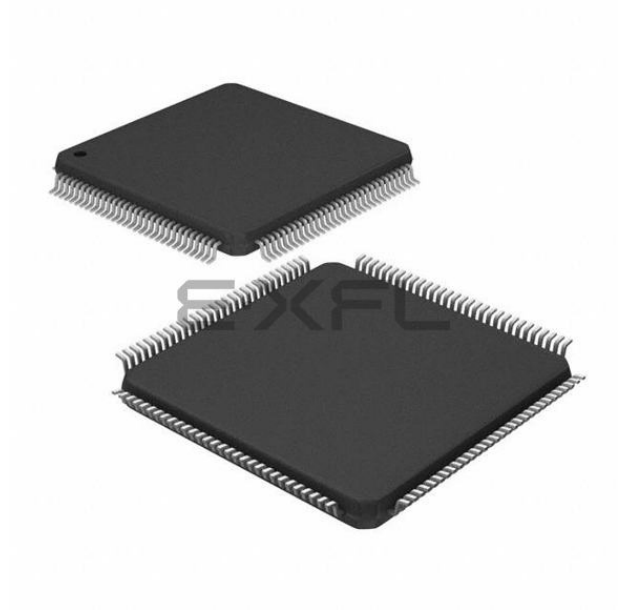


Welcome to E-XFL.COM

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	Last Time Buy
Core Processor	H8SX
Core Size	32-Bit Single-Core
Speed	48MHz
Connectivity	SCI, SSU
Peripherals	DMA, PWM, WDT
Number of I/O	82
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	12K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-20°C ~ 75°C (TA)
Mounting Type	Surface Mount
Package / Case	120-LQFP
Supplier Device Package	120-LQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/df61582n48fpv

1.2 Block Diagram

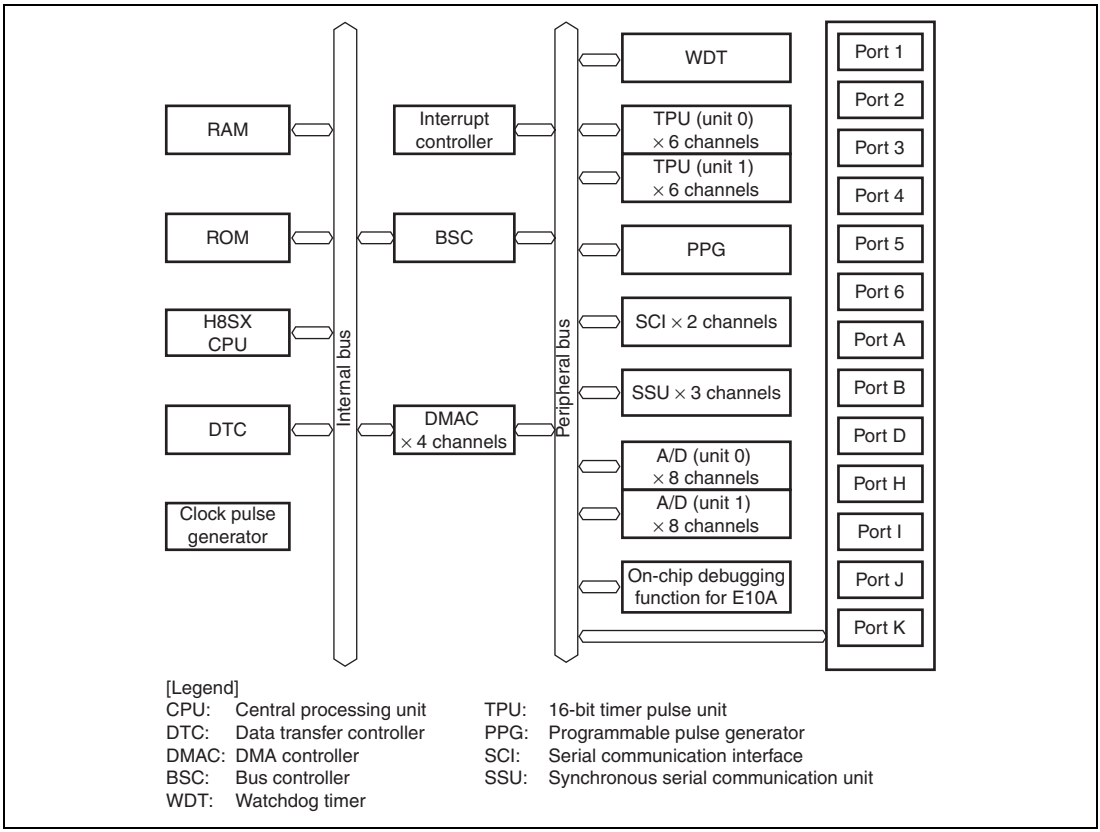
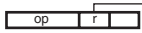
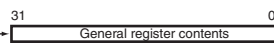
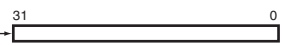
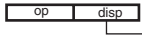
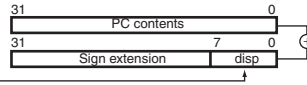
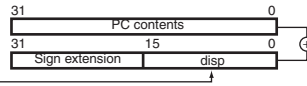
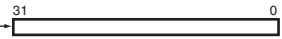
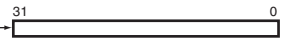
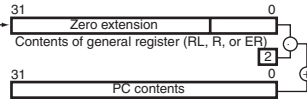
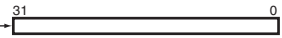
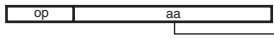
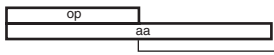
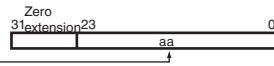
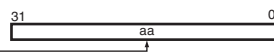
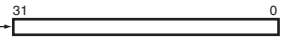
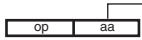
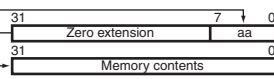
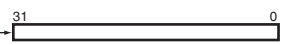
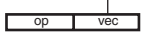
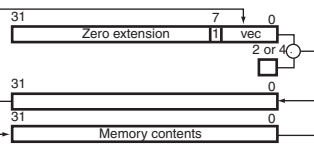
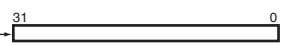


Figure 1.1 Block Diagram of H8SX/1582

Table 2.15 Effective Address Calculation for Branch Instructions

No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
1	Register indirect 		
2	Program-counter relative with 8-bit displacement  Program-counter relative with 16-bit displacement 	 	 
3	Program-counter relative with index register 		
4	24-bit absolute address  32-bit absolute address 	 	 
5	Memory indirect 		
6	Extended memory indirect 		

2.8.13 MOVA Instruction

The MOVA instruction stores the effective address in a general register.

1. Firstly, data is obtained by the addressing mode shown in item 2of table 2.14.
2. Next, the effective address is calculated using the obtained data as the index by the addressing mode shown in item 5 of table 2.14. The obtained data is used instead of the general register. The result is stored in a general register. For details, see H8SX Family Software Manual.

7.2.4 DMA Transfer Count Register (DTCR)

DTCR is a 32-bit readable/writable register that specifies the size of data to be transferred (total transfer size).

To transfer 1-byte data in total, set H'00000001 in DTCR. When H'00000000 is set in this register, it means that the total transfer size is not specified and data is transferred with the transfer counter stopped (free running mode). When H'FFFFFFFF is set, the total transfer size is 4 Gbytes (4,294,967,295), which is the maximum size. While data is being transferred, this register indicates the remaining transfer size. The value corresponding to its data access size is subtracted every time data is transferred (byte: –1, word: –2, and longword: –4).

Although DTCR can always be read from by the CPU, it must be read from in longwords and must not be written to while data for the channel is being transferred.

Bit	31	30	29	28	27	26	25	24
Bit Name								
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	23	22	21	20	19	18	17	16
Bit Name								
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	15	14	13	12	11	10	9	8
Bit Name								
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	7	6	5	4	3	2	1	0
Bit Name								
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

When an interrupt by an extended repeat area overflow is used in block transfer mode, the following should be taken into consideration.

When a transfer is stopped by an interrupt by an extended repeat area overflow, the address register must be set so that the block size is a power of 2 or the block size boundary is aligned with the extended repeat area boundary. When an overflow on the extended repeat area occurs during a transfer of one block, the interrupt by the overflow is suspended and the transfer overruns.

Figure 7.16 shows examples when the extended repeat area function is used in block transfer mode.

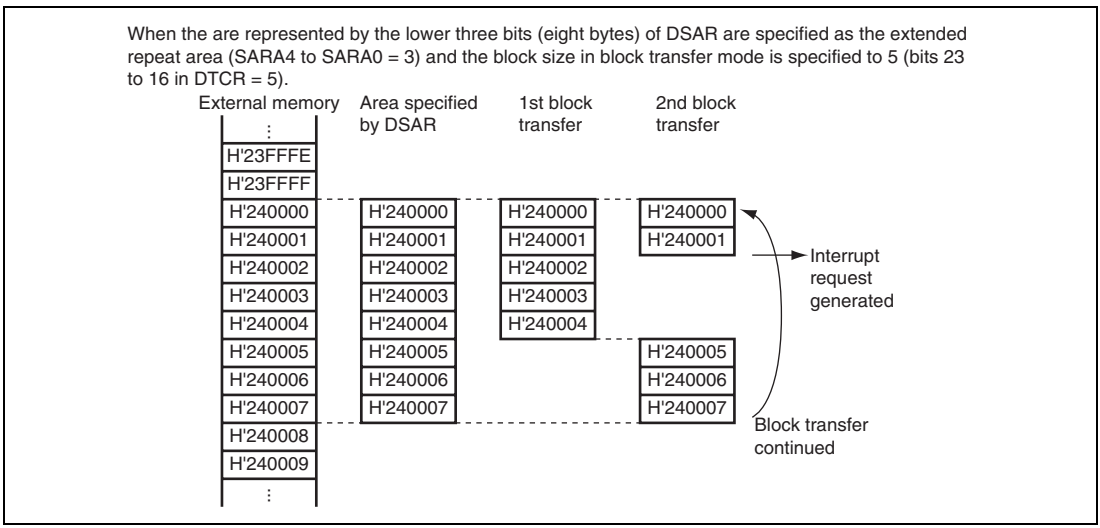


Figure 7.16 Example of Extended Repeat Area Function in Block Transfer Mode

(9) DTIF Bit in DMDR

The DTIF bit in DMDR is set to 1 after the total transfer size of transfers is completed. When both the DTIF and DTIE bits in DMDR are set to 1, a transfer end interrupt by the transfer counter is requested to the CPU or DTC.

The DTIF bit is set to 1 when the ACT bit in DMDR is cleared to 0 to stop a transfer after the bus cycle is completed.

The DTIF bit is automatically cleared to 0 and a transfer request is cleared if the transfer is resumed by setting the DTE bit to 1 during interrupt handling.

For details on interrupts, see section 7.7, Interrupt Sources.

7.4.8 Priority of Channels

The channels of the DMAC are given following priority levels: channel 0 > channel 1 > channel 2 > channel3. Table 7.5 shows the priority levels among the DMAC channels.

Table 7.5 Priority among DMAC Channels

Channel	Priority
Channel 0	High
Channel 1	
Channel 2	
Channel 3	Low

The channel having highest priority other than the channel being transferred is selected when a transfer is requested from other channels. The selected channel starts the transfer after the channel being transferred releases the bus. At this time, when a bus master other than the DMAC requests the bus, the cycle for the bus master is inserted.

In a burst transfer or a block transfer, channels are not switched.

8.9.3 DMAC Transfer End Interrupt

When DTC is activated by the DMAC transfer end interrupt, the data in the DTE bit in DMDR has priority over the DTC control regardless of the transfer counter or the DISEL bit. Therefore, an interrupt to CPU may not be generated even if the DTC transfer counter becomes 0.

8.9.4 DTCE Bit Setting

For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR. If all interrupts are disabled, multiple activation sources can be set at one time (only at the initial setting) by writing data after executing a dummy read on the relevant register.

8.9.5 Chain Transfer

When chain transfer is used, clearing of the activation source or DTCER is performed when the last of the chain of data transfers is executed. SCI and high-speed A/D converter interrupt/activation sources, on the other hand, are cleared when the DTC reads or writes to the relevant register.

Therefore, when the DTC is activated by an interrupt or activation source, if a read/write of the relevant register is not included in the last chained data transfer, the interrupt or activation source will be retained.

8.9.6 Transfer Information Start Address, Source Address, and Destination Address

The transfer information start address to be specified in the vector table should be address 4n. If an address other than address 4n is specified, the lower 2 bits of the address are regarded as 0s.

The source and destination addresses specified in SAR and DAR, respectively, will be transferred in the divided bus cycles depending on the address and data size.

8.9.7 Endian

The DTC supports both big-endian and little-endian formats. The endian format should be the same when the transfer information is written to and when the transfer information is read by the DTC.

(2) PB1

The pin function is switched as shown below according to the PB1DDR bit setting.

		Setting
		I/O Port
Module Name	Pin Function	PB1DDR
I/O port	PB1 output	1
	PB1 input (initial setting)	0

(3) PB0

The pin function is switched as shown below according to the PB0DDR bit setting.

		Setting
		I/O Port
Module Name	Pin Function	PB0DDR
I/O port	PB0 output	1
	PB0 input (initial setting)	0

9.2.7 Port D

(1) PD7/$\overline{\text{SCS1}}$

The pin function is switched as shown below according to the combination of the SSU_1 and the PD7DDR bit settings.

		Setting	
		SSU_2	I/O Port
Module Name	Pin Function	$\overline{\text{SCS1}}$_OE	PD7DDR
SSU_1	$\overline{\text{SCS1}}$ output	1	—
I/O port	PD7 output	0	1
	PD7 input (initial setting)	0	0

(b) Examples of waveform output operation

Figure 10.7 shows an example of 0-output and 1-output.

In this example, TCNT has been designated as a free-running counter, and settings have been made so that 1 is output by compare match A, and 0 is output by compare match B. When the set level and the pin level match, the pin level does not change.

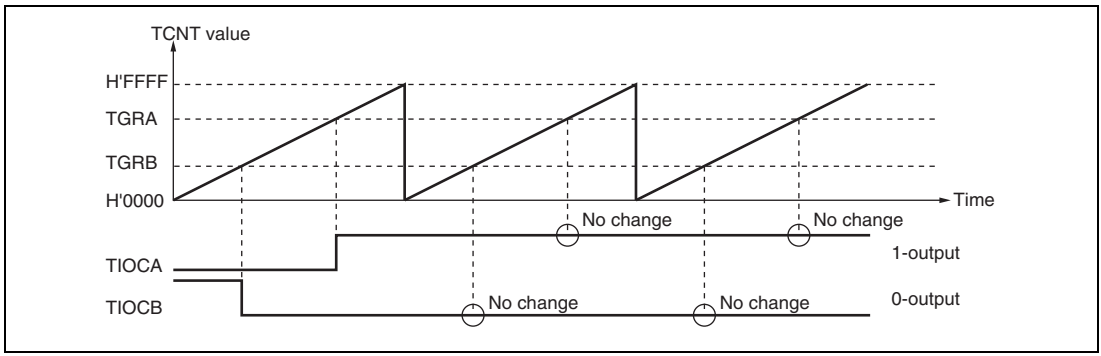


Figure 10.7 Example of 0-Output/1-Output Operation

Figure 10.8 shows an example of toggle output.

In this example, TCNT has been designated as a periodic counter (with counter clearing performed by compare match B), and settings have been made so that output is toggled by both compare match A and compare match B.

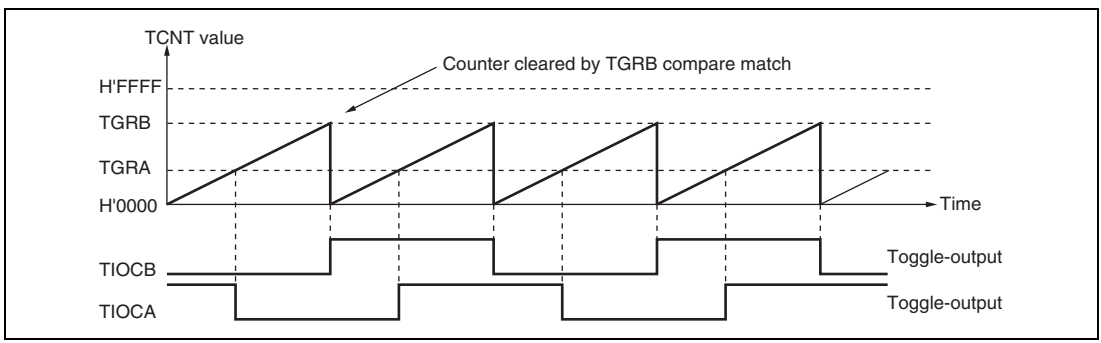


Figure 10.8 Example of Toggle Output Operation

11.3.1 Next Data Enable Registers H, L (NDERH, NDERL)

NDERH and NDERL enable/disable pulse output on a bit-by-bit basis.

- NDERH

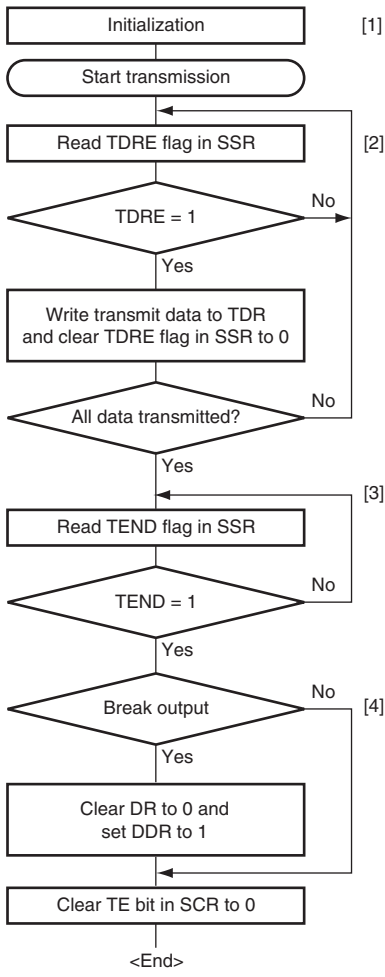
Bit	7	6	5	4	3	2	1	0
Bit Name	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8
Initial Value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- NDERL

Bit	7	6	5	4	3	2	1	0
Bit Name	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
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

- NDERH

Bit	Bit Name	Initial Value	R/W	Description
7	NDER15	0	R/W	Next Data Enable 15 to 8
6	NDER14	0	R/W	When a bit is set to 1, the value in the corresponding NDRH bit is transferred to the PODRH bit by the selected output trigger. Values are not transferred from NDRH to PODRH for cleared bits.
5	NDER13	0	R/W	
4	NDER12	0	R/W	
3	NDER11	0	R/W	
2	NDER10	0	R/W	
1	NDER9	0	R/W	
0	NDER8	0	R/W	



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin. After the TE bit is set to 1, a 1 is output for a frame, and transmission is enabled.

[2] SCI state check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:

To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and clear the TDRE flag to 0. However, the TDRE flag is checked and cleared automatically when the DTC or DMAC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set DDR for the port corresponding to the TxD pin to 1, clear DR to 0, then clear the TE bit in SCR to 0.

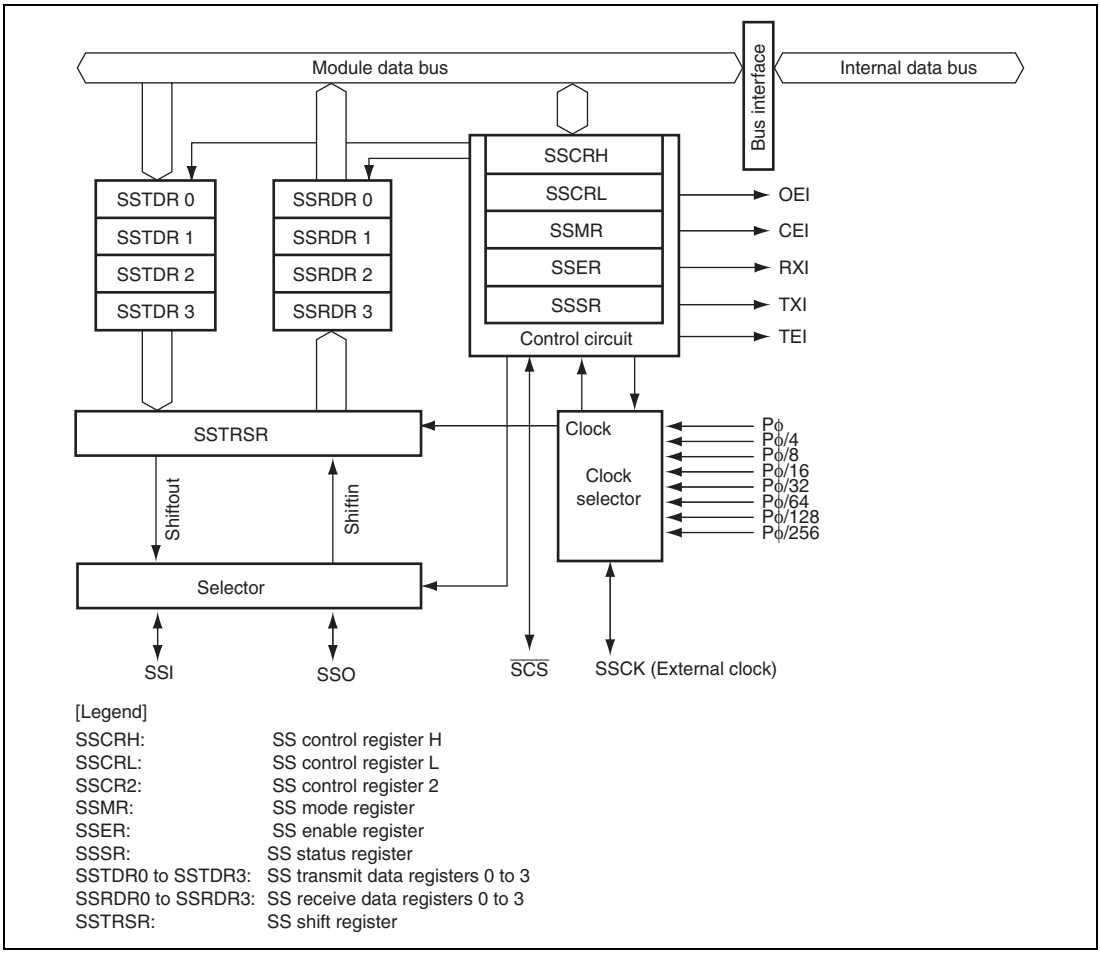
Figure 13.7 Sample Serial Transmission Flowchart

13.4.6 Serial Data Reception (Asynchronous Mode)

Figure 13.8 shows an example of the operation for reception in asynchronous mode. In serial reception, the SCI operates as described below.

1. The SCI monitors the communication line, and if a start bit is detected, performs internal synchronization, stores receive data in RSR, and checks the parity bit and stop bit.

Figure 14.1 shows a block diagram of the SSU.



[Legend]

SSCRH:	SS control register H
SSCRL:	SS control register L
SSCR2:	SS control register 2
SSMR:	SS mode register
SSER:	SS enable register
SSSR:	SS status register
SSTDR0 to SSTDR3:	SS transmit data registers 0 to 3
SSRDR0 to SSRDR3:	SS receive data registers 0 to 3
SSTRSR:	SS shift register

Figure 14.1 Block Diagram of SSU

Bit	7	6	5	4	3	2	1	0
Bit Name								
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	7	6	5	4	3	2	1	0
Bit Name								
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	7	6	5	4	3	2	1	0
Bit Name								
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	7	6	5	4	3	2	1	0
Bit Name								
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

Table 14.2 Correspondence Between DATS Bit Setting and SSTDR

	DATS[1:0] (SSCRL[1:0])			
SSTDR	**00**	**01**	**10**	**11 (Setting Invalid)**
0	Valid	Valid	Valid	Invalid
1	Invalid	Valid	Valid	Invalid
2	Invalid	Invalid	Valid	Invalid
3	Invalid	Invalid	Valid	Invalid

Table 14.5 Communication Modes and Pin States of SSCK Pin

Communication Mode	Register Setting			Pin State
	SSUMS	MSS	SCKS	SSCK
SSU communication mode	0	0	0	—
			1	Input
		1	0	—
			1	Output
Clock synchronous communication mode	1	0	0	—
			1	Input
		1	0	—
			1	Output

[Legend]

—: Not used as SSU pin (can be used as I/O port)

Table 14.6 Communication Modes and Pin States of $\overline{\text{SCS}}$ Pin

Communication Mode	Register Setting				Pin State
	SSUMS	MSS	CSS1	CSS0	$\overline{\text{SCS}}$
SSU communication mode	0	0	×	×	Input
		1	0	0	—
			0	1	Input
			1	0	Automatic input/output
			1	1	Output
Clock synchronous communication mode	1	×	×	×	—

[Legend]

×: Don't care

—: Not used as SSU pin (can be used as I/O port)

17.7.2 Programming/Erasing Interface Parameters

The programming/erasing interface parameters specify the operating frequency, storage place for program data, start address of programming destination, and erase block number, and exchanges the execution result. These parameters use the general registers of the CPU (ER0 and ER1) or the on-chip RAM area. The initial values of programming/erasing interface parameters are undefined at a power-on reset or a transition to software standby mode.

Since registers of the CPU except for R0 are saved in the stack area during download of an on-chip program, initialization, programming, or erasing, allocate the stack area before performing these operations (the maximum stack size is 128 bytes). The return value of the processing result is written in R0. The programming/erasing interface parameters are used in download control, initialization before programming or erasing, programming, and erasing. Table 17.4 shows the usable parameters and target modes. The meaning of the bits in the flash pass and fail result parameter (FPFR) varies in initialization, programming, and erasure.

Table 17.4 Parameters and Target Modes

Parameter	Download	Initialization	Programming	Erasure	R/W	Initial Value	Allocation
DPFR	O	—	—	—	R/W	Undefined	On-chip RAM*
FPFR	—	O	O	O	R/W	Undefined	R0L of CPU
FPEFEQ	—	O	—	—	R/W	Undefined	ER0 of CPU
FMPAR	—	—	O	—	R/W	Undefined	ER1 of CPU
FMPDR	—	—	O	—	R/W	Undefined	ER0 of CPU
FEBS	—	—	—	O	R/W	Undefined	ER0 of CPU

Note: * A single byte of the start address of the on-chip RAM specified by FTDAR

Download Control: The on-chip program is automatically downloaded by setting the SCO bit in FCCS to 1. The on-chip RAM area to download the on-chip program is the 4-kbyte area starting from the start address specified by FTDAR. Download is set by the programming/erasing interface registers, and the download pass and fail result parameter (DPFR) indicates the return value.

Table 17.10 Usable Area for Programming in User Boot Mode

	Storable/Executable Area		Selected MAT		
Item	**On-Chip RAM**	**User Boot MAT**	**User MAT**	**User Boot MAT**	**Embedded Program Storage MAT**
Storage area for program data	O	×*[1]	—	—	—
Operation for selecting on-chip program to be downloaded	O	O		O	
Operation for writing H'A5 to FKEY	O	O		O	
Execution of writing 1 to SCO bit in FCCS (download)	O	×			O
Operation for clearing FKEY	O	O		O	
Decision of download result	O	O		O	
Operation for download error	O	O		O	
Operation for setting initialization parameter	O	O		O	
Execution of initialization	O	×		O	
Decision of initialization result	O	O		O	
Operation for initialization error	O	O		O	
NMI handling routine	O	×		O	
Operation for disabling interrupts	O	O		O	
Switching memory MATs by FMATS	O	×	O		
Operation for writing H'5A to FKEY	O	×	O		
Operation for setting programming parameter	O	×	O		
Execution of programming	O	×	O		
Decision of programming result	O	×	O		
Operation for programming error	O	×*[2]	O		
Operation for clearing FKEY	O	×	O		
Switching memory MATs by FMATS	O	×		O	

Notes: 1. Transferring the program data to the on-chip RAM beforehand enables this area to be used.
2. Switching memory MATs by FMATS by a program in the on-chip RAM enables this area to be used.

(5) Transition to Programming/Erasing State

The boot program will transfer the erasing program, and erase the user MATs and user boot MATs in that order. On completion of this erasure, ACK will be returned and will enter the programming/erasing state.

The host should select the device code, clock mode, and new bit rate with device selection, clock-mode selection, and new bit-rate selection commands, and then send the command for the transition to programming/erasing state. These procedures should be carried out before sending of the programming selection command or program data.

Command	H'40

- Command, H'40, (one byte): Transition to programming/erasing state

Response	H'06

- Response, H'06, (one byte): Response to transition to programming/erasing state
  The boot program will send ACK when the user MAT and user boot MAT have been erased by the transferred erasing program.

Error Response	H'C0	H'51

- Error response, H'C0, (one byte): Error response for user boot MAT blank check
- Error code, H'51, (one byte): Erasing error
  An error occurred and erasure was not completed.

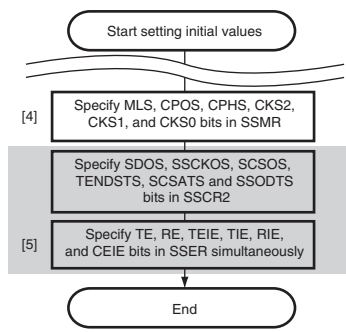
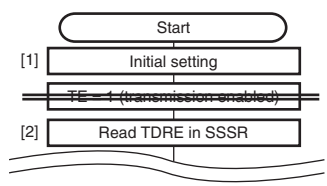
(6) Command Error

A command error will occur when a command is undefined, the order of commands is incorrect, or a command is unacceptable. Issuing a clock-mode selection command before a device selection or an inquiry command after the transition to programming/erasing state command, are examples.

Error Response	H'80	H'xx

- Error response, H'80, (one byte): Command error
- Command, H'xx, (one byte): Received command

Register Abbreviation	Reset	Sleep	Module Stop	All-Module-Clock-Stop	Software Standby	Hardware Standby
TGRA_0	Initialized	—	—	—	—	TPU_0
TGRB_0	Initialized	—	—	—	—	
TGRC_0	Initialized	—	—	—	—	
TGRD_0	Initialized	—	—	—	—	
TCR_1	Initialized	—	—	—	—	TPU_1
TMDR_1	Initialized	—	—	—	—	
TIOR_1	Initialized	—	—	—	—	
TIER_1	Initialized	—	—	—	—	
TSR_1	Initialized	—	—	—	—	
TCNT_1	Initialized	—	—	—	—	
TGRA_1	Initialized	—	—	—	—	
TGRB_1	Initialized	—	—	—	—	
TCR_2	Initialized	—	—	—	—	TPU_2
TMDR_2	Initialized	—	—	—	—	
TIOR_2	Initialized	—	—	—	—	
TIER_2	Initialized	—	—	—	—	
TSR_2	Initialized	—	—	—	—	
TCNT_2	Initialized	—	—	—	—	
TGRA_2	Initialized	—	—	—	—	
TGRB_2	Initialized	—	—	—	—	
TCR_3	Initialized	—	—	—	—	TPU_3
TMDR_3	Initialized	—	—	—	—	
TIORH_3	Initialized	—	—	—	—	
TIORL_3	Initialized	—	—	—	—	
TIER_3	Initialized	—	—	—	—	
TSR_3	Initialized	—	—	—	—	
TCNT_3	Initialized	—	—	—	—	
TGRA_3	Initialized	—	—	—	—	
TGRB_3	Initialized	—	—	—	—	
TGRC_3	Initialized	—	—	—	—	
TGRD_3	Initialized	—	—	—	—	

Item	Page	Revision (See Manual for Details)
Section 14 Synchronous Serial Communication Unit (SSU) 14.3.5 SS Status Register (SSSR) • BIt 6, 3 to 0	492 to 494	Amended (When the CPU is used to clear this flag by writing 0 while the corresponding interrupt is enabled, be sure to read the flag after writing 0 to it.)
14.3.7 SS Transmit Data Registers 0 to 3 (SSTDR0 to SSTDR3)	496	Added Be sure not to access invalid SSTDRs.
Table 14.2 Correspondence Between DATS Bit Setting and SSTDR	497	Added
14.3.8 SS Receive Data Registers 0 to 3 (SSRDR0 to SSRDR3)	198	Added Be sure not to access invalid SSRDRs
Table 14.3 Correspondence Between DATS Bit Setting and SSRDR	499	Added
14.4.5 SSU Mode Table 14.5 Communication Modes and Pin States of SSCK Pin Table 14.6 Communication Modes and Pin States of $\overline{SCS}$ Pin	503	Amended [Legend] —: Not used as SSU pin (can be used as I/O port)
Figure 14.4 Example of Initial Settings in SSU Mode	504	Amended 
Figure 14.6 Flowchart Example of Data Transmission (SSU Mode)	507	Deleted 

Item	Page	Revision (See Manual for Details)
Section 18 Clock Pulse Generator 18.1 Register Description 18.1.1 System Clock Control Register (SCKCR)	641	Amended SCKCR controls Bϕ clock output and frequencies of the system, peripheral module, and external clocks, and selects the Bϕ clock to be output.

Bit	Bit Name	Description
15	PSTOP1	Bϕ Clock Output Enable Controls Bϕ output on PA7. • Normal operation 0: Bϕ output 1: Fixed high • Software standby mode X: Fixed high • Hardware standby mode X: Hi-Z

18.5 Usage Notes 18.5.1 Notes on Clock Pulse Generator	645	Deleted ~~5. When Iϕ > Pϕ is specified by SCKCR, signals from the peripheral modules must be synchronized with the system clock. When CPU instructions are used to clear the interrupt source flag of a peripheral module, the flag must be read after being cleared to 0.~~

Index