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	25MHz
Connectivity	SCI, SmartCard
Peripherals	DMA, POR, PWM, WDT
Number of I/O	106
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 12x10b; D/A 4x8b
Oscillator Type	Internal
Operating Temperature	-20°C ~ 75°C (TA)
Mounting Type	Surface Mount
Package / Case	144-BFQFP
Supplier Device Package	144-QFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/d12332vfc25v

Item	Page	Revision (See Manual for Details)
6.3.5 Chip Select Signals	153	Description amended …Enabling or disabling of $\overline{CS}_n$ signal output is performed by setting the data direction register (DDR) bit for the port corresponding to the particular $\overline{CS}_n$ pin, the CS/67 enable bit (CS/67E), and the CS25 enable bit (CS25E). In expanded mode with on-chip ROM disabled, the $\overline{CS}_0$ pin is placed in the output state after a reset. Pins $\overline{CS}_1$ to $\overline{CS}_7$ are placed in the input state after a reset, so the corresponding DDR bits as well as bits CS/67E and CS25E should be set to 1 when outputting signals $\overline{CS}_1$ to $\overline{CS}_7$. In expanded mode with on-chip ROM enabled, pins $\overline{CS}_0$ to $\overline{CS}_7$ are all placed in the input state after a reset, so the corresponding DDR bits as well as bits CS/67E and CS25E should be set to 1 when outputting signals $\overline{CS}_1$ to $\overline{CS}_7$. For details, …
Section 13 Watchdog Timer	579 to 594	Note shown below deleted Note: The $\overline{WDTOVF}$ pin function cannot be used in the F-ZTAT version.
14.2.8 Bit Rate Register (BRR) Table 14.3 BRR Settings for Various Bit Rates (Asynchronous Mode)	616	Table 14.3 amended

	ϕ = 25 MHz		
Bit Rate (bits/s)	**n**	**N**	**Error (%)**
110	3	110	–0.02
150	3	80	0.47
300	2	162	–0.15
600	2	80	0.47
1200	1	162	–0.15
2400	1	80	0.47
4800	0	162	–0.15
9600	0	80	0.47
19200	0	40	–0.76
31250	0	24	1.00
38400	0	19	1.73

19.4.1 Features	738	Description amended • Reprogramming capability The flash memory can be reprogrammed min. 100 times.

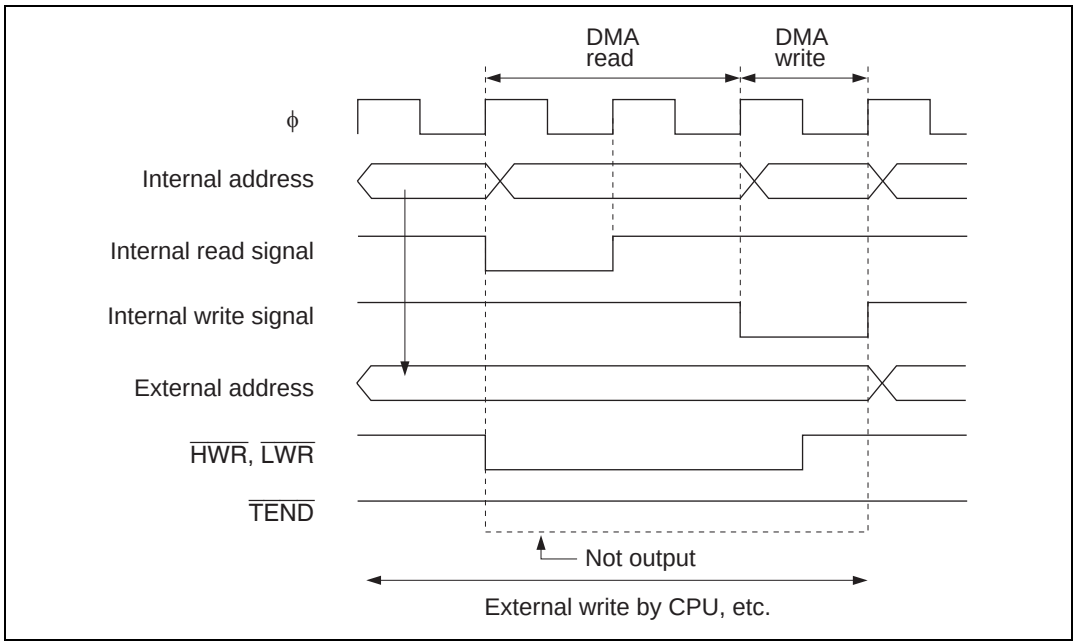


Figure 7.42 Example in Which Low Level is Not Output at $\overline{\text{TEND}}$ Pin

Activation by Falling Edge on $\overline{\text{DREQ}}$ Pin: $\overline{\text{DREQ}}$ pin falling edge detection is performed in synchronization with DMAC internal operations. The operation is as follows:

[1] Activation request wait state: Waits for detection of a low level on the $\overline{\text{DREQ}}$ pin, and switches to [2].

[2] Transfer wait state: Waits for DMAC data transfer to become possible, and switches to [3].

[3] Activation request disabled state: Waits for detection of a high level on the $\overline{\text{DREQ}}$ pin, and switches to [1].

After DMAC transfer is enabled, a transition is made to [1]. Thus, initial activation after transfer is enabled is performed on detection of a low level.

Activation Source Acceptance: At the start of activation source acceptance, a low level is detected in both $\overline{\text{DREQ}}$ pin falling edge sensing and low level sensing. Similarly, in the case of an internal interrupt, the interrupt request is detected. Therefore, a request is accepted from an internal interrupt or $\overline{\text{DREQ}}$ pin low level that occurs before execution of the DMABCRL write to enable transfer.

When the DMAC is activated, take any necessary steps to prevent an internal interrupt or $\overline{\text{DREQ}}$ pin low level remaining from the end of the previous transfer, etc.

Section 8 Data Transfer Controller

8.1 Overview

The chip includes a data transfer controller (DTC). The DTC can be activated for data transfer by an interrupt or software.

8.1.1 Features

The features of the DTC are:

- Transfer possible over any number of channels
  - Transfer information is stored in memory
  - One activation source can trigger a number of data transfers (chain transfer)
  - Chain transfer execution can be set after data transfer (when counter = 0)
- Selection of transfer modes
  - Normal, repeat, and block transfer modes available
  - Incrementing, decrementing, and fixing of source and destination addresses can be selected
- Direct specification of 16-Mbyte address space possible
  - 24-bit transfer source and destination addresses can be specified
- Transfer can be set in byte or word units
- A CPU interrupt can be requested for the interrupt that activated the DTC
  - An interrupt request can be issued to the CPU after one data transfer ends
  - An interrupt request can be issued to the CPU after all the specified data transfers have ended
- Activation by software is possible
- Module stop mode can be set
  - The initial setting enables DTC registers to be accessed. DTC operation is halted by setting module stop mode

8.3.6 Repeat Mode

In repeat mode, one operation transfers one byte or one word of data.

From 1 to 256 transfers can be specified. Once the specified number of transfers have ended, the initial state of the transfer counter and the address register specified as the repeat area is restored, and transfer is repeated. In repeat mode the transfer counter value does not reach H'00, and therefore CPU interrupts cannot be requested when DISEL = 0.

Table 8.7 lists the register information in repeat mode and figure 8.7 shows the memory map in repeat mode.

Table 8.7 Register Information in Repeat Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates source address
DTC destination address register	DAR	Designates destination address
DTC transfer count register AH	CRAH	Holds number of transfers
DTC transfer count register AL	CRAL	Transfer counter
DTC transfer count register B	CRB	Not used

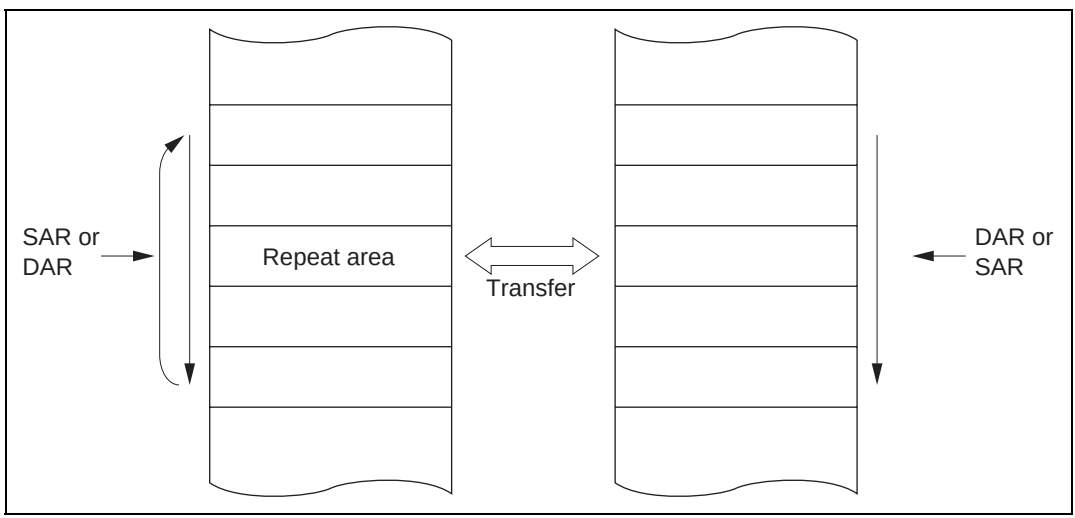


Figure 8.7 Memory Map in Repeat Mode

Pin	Selection Method and Pin Functions
$P1_5/PO_{13}/TIOCB_1/TCLKC$	The pin function is switched as shown below according to the combination of the TPU channel 1 setting (by bits MD3 to MD0 in TMDR1, bits IOB3 to IOB0 in TIOR1, and bits CCLR1 and CCLR0 in TCR1), bits TPSC2 to TPSC0 in TCR0, TCR2, TCR4, and TCR5, bit NDER13 in NDERH, and bit P15DDR.

TPU Channel 1 Setting	Table Below (1)	Table Below (2)		
P15DDR	—	0	1	1
NDER13	—	—	0	1
Pin function	$TIOCB_1$ output	$P1_5$ input	$P1_5$ output	PO_{13} output
		$TIOCB_1$ input *1		
	TCLKC input *2			

Notes: 1. $TIOCB_1$ input when MD3 to MD0 = B'0000 or B'01xx, and IOB3 to IOB0 = B'10xx.

2. TCLKC input when the setting for either TCR0 or TCR2 is: TPSC2 to TPSC0 = B'110; or when the setting for either TCR4 or TCR5 is TPSC2 to TPSC0 = B'101.
TCLKC input when channels 2 and 4 are set to phase counting mode.

TPU Channel 1 Setting	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

Pin	Selection Method and Pin Functions
$P2_0/PO_0/TIOCA_3$	The pin function is switched as shown below according to the combination of the TPU channel 3 setting (by bits MD3 to MD0 in TMDR3, bits IOA3 to IOA0 in TIOR3H, and bits CCLR2 to CCLR0 in TCR3), bit NDER0 in NDERL, and bit P20DDR.

TPU Channel 3 Setting	Table Below (1)	Table Below (2)		
P20DDR	—	0	1	1
NDER0	—	—	0	1
Pin function	$TIOCA_3$ output	$P2_0$ input	$P2_0$ output	PO_0 output
		$TIOCA_3$ input *1		

TPU Channel 3 Setting	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00		
CCLR2 to CCLR0	—	—	—	—	Other than B'001	B'001
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 2 output	—

x: Don't care

Notes: 1. $TIOCA_3$ input when MD3 to MD0 = B'0000, and IOA3 to IOA0 = B'10xx.

2. $TIOCB_3$ output is disabled.

9.8 Port 7

9.8.1 Overview

Port 7 is a 6-bit I/O port. Port 7 pins also function as 8-bit timer I/O pins ($TMRI_0$, $TMCI_0$, TMO_0, $TMRI_1$, $TMCI_1$, TMO_1). Port 7 pin functions are the same in all operating modes. Port 7 uses Schmitt-triggered input.

Figure 9.7 shows the port 7 pin configuration.

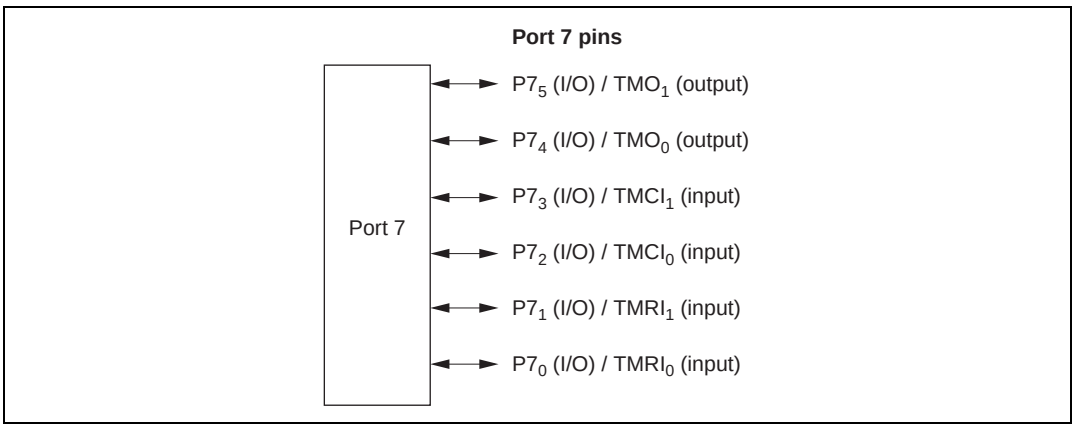


Figure 9.7 Port 7 Pin Functions

14.2.3 Transmit Shift Register (TSR)

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

TSR is a register used to transmit 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 with the LSB (bit 0).

When transmission of one byte is completed, the next transmit data is transferred from TDR to TSR, and transmission started, automatically. However, data transfer from TDR to TSR is not performed if the TDRE bit in SSR is set to 1.

TSR cannot be directly read or written to by the CPU.

14.2.4 Transmit Data Register (TDR)

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

TDR is an 8-bit register that stores data for serial transmission.

When the SCI detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts serial transmission. Continuous serial transmission can be carried out by writing the next transmit data to TDR during serial transmission of the data in TSR.

TDR can be read or written to by the CPU at all times.

TDR is initialized to H'FF by a reset, and in standby mode or module stop mode.

Retransfer Operations (Except Block Transfer Mode): Retransfer operations are performed by the SCI in receive mode and transmit mode as described below.

- Retransfer operation when SCI is in receive mode

  Figure 15.11 illustrates the retransfer operation when the SCI is in receive mode.

[1] If an error is found when the received parity bit is checked, the PER bit in SSR is automatically set to 1. If the RIE bit in SCR is enabled at this time, an ERI interrupt request is generated. The PER bit in SSR should be kept cleared to 0 until the next parity bit is sampled.

[2] The RDRF bit in SSR is not set for a frame in which an error has occurred.

[3] If no error is found when the received parity bit is checked, the PER bit in SSR is not set.

[4] If no error is found when the received parity bit is checked, the receive operation is judged to have been completed normally, and the RDRF flag in SSR is automatically set to 1. If the RIE bit in SCR is enabled at this time, an RXI interrupt request is generated.

If DMAC or DTC data transfer by an RXI source is enabled, the contents of RDR can be read automatically. When the RDR data is read by the DMAC or DTC, the RDRF flag is automatically cleared to 0.

[5] When a normal frame is received, the pin retains the high-impedance state at the timing for error signal transmission.

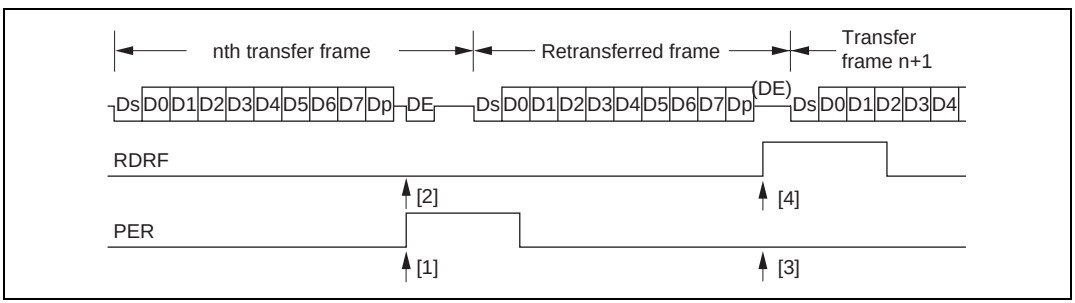
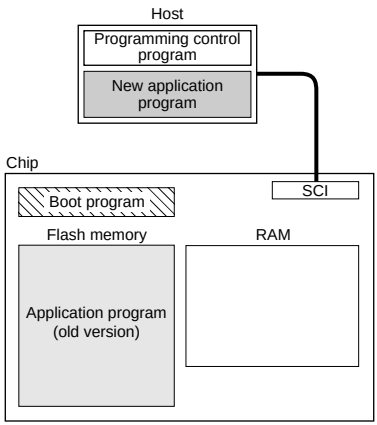


Figure 15.11 Retransfer Operation in SCI Receive Mode

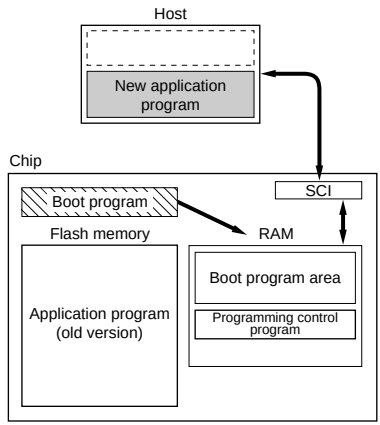
19.4.4 On-Board Programming Modes

- Boot mode

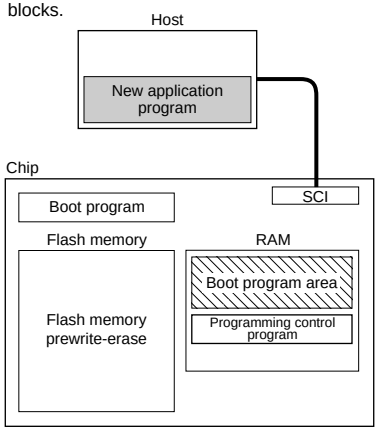
1. Initial state
   The old program version or data remains written in the flash memory. The user should prepare the programming control program and new application program beforehand in the host.



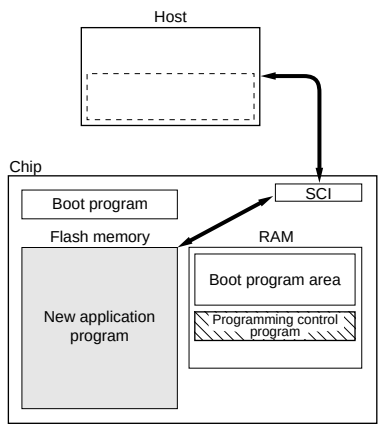
2. Programming control program transfer
   When boot mode is entered, the boot program in the chip (originally incorporated in the chip) is started and the programming control program in the host is transferred to RAM via SCI communication. The boot program required for flash memory erasing is automatically transferred to the RAM boot program area.



3. Flash memory initialization
   The erase program in the boot program area (in RAM) is executed, and the flash memory is initialized (to H'FF). In boot mode, entire flash memory erasure is performed, without regard to blocks.



4. Writing new application program
   The programming control program transferred from the host to RAM is executed, and the new application program in the host is written into the flash memory.



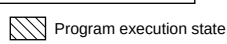


Figure 19.4 Boot Mode

a transition to program mode or erase mode. When actually programming a flash memory area, the RAMS bit should be cleared to 0.

2. A RAM area cannot be erased by execution of software in accordance with the erase algorithm while flash memory emulation in RAM is being used.
3. Block area EB0 includes the vector table. When performing RAM emulation, the vector table is needed by the overlap RAM.

19.10 Interrupt Handling when Programming/Erasing Flash Memory

All interrupts, including NMI input, are disabled when flash memory is being programmed or erased (when the P or E bit is set in FLMCR1), and while the boot program is executing in boot mode*1, to give priority to the program or erase operation. There are three reasons for this:

1. Interrupt during programming or erasing might cause a violation of the programming or erasing algorithm, with the result that normal operation could not be assured.
2. In the interrupt exception handling sequence during programming or erasing, the vector would not be read correctly*2, possibly resulting in MCU runaway.
3. If an interrupt occurred during boot program execution, it would not be possible to execute the normal boot mode sequence.

For these reasons, in on-board programming mode alone there are conditions for disabling interrupts, as an exception to the general rule. However, this provision does not guarantee normal erasing and programming or MCU operation. All interrupt requests, including NMI, must therefore be restricted inside and outside the MCU when programming or erasing flash memory. The NMI interrupt is also disabled in the error-protection state while the P or E bit remains set in FLMCR1.

Notes: 1. Interrupt requests must be disabled inside and outside the MCU until the programming control program has completed programming.
2. The vector may not be read correctly in this case for the following two reasons:
   - If flash memory is read while being programmed or erased (while the P or E bit is set in FLMCR1), correct read data will not be obtained (undetermined values will be returned).
   - If the interrupt entry in the vector table has not been programmed yet, interrupt exception handling will not be executed correctly.

19.14.2 Flash Memory Control Register 2 (FLMCR2)

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

FLMCR2 is an 8-bit register that controls the flash memory operating modes. FLMCR2 is initialized to H'00 by a reset, and in hardware standby mode and software standby mode.

When on-chip flash memory is disabled, a read will return H'00 and writes are invalid.

Bit 7—Flash Memory Error (FLER): Indicates that an error has occurred during an operation on flash memory (programming or erasing). When FLER is set to 1, flash memory goes to the error-protection state.

Bit 7 FLER	Description	
0	Flash memory is operating normally	(Initial value)
	Flash memory program/erase protection (error protection) is disabled	
	[Clearing condition]	
	Reset or hardware standby mode	
1	An error has occurred during flash memory programming/erasing	
	Flash memory program/erase protection (error protection) is enabled	
	[Setting condition]	
	See section 19.17.3, Error Protection	

Bits 6 to 0—Reserved: These bits cannot be modified and are always read as 0.

(4) Shift Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code					No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Advanced
SHAL	SHAL.B Rd	B		2									—	—	↕	↕	0	↕	1
	SHAL.B #2,Rd	B		2									—	—	↕	↕	↕	↕	1
	SHAL.W Rd	W		2									—	—	↕	↕	0	↕	1
	SHAL.W #2,Rd	W		2									—	—	↕	↕	↕	↕	1
	SHAL.L ERd	L		2									—	—	↕	↕	0	↕	1
	SHAL.L #2,ERd	L		2									—	—	↕	↕	↕	↕	1
SHAR	SHAR.B Rd	B		2									—	—	↕	↕	0	↕	1
	SHAR.B #2,Rd	B		2									—	—	↕	↕	↕	↕	1
	SHAR.W Rd	W		2									—	—	↕	↕	0	↕	1
	SHAR.W #2,Rd	W		2									—	—	↕	↕	↕	↕	1
	SHAR.L ERd	L		2									—	—	↕	↕	0	↕	1
	SHAR.L #2,ERd	L		2									—	—	↕	↕	↕	↕	1
SHLL	SHLL.B Rd	B		2									—	—	↕	↕	0	↕	1
	SHLL.B #2,Rd	B		2									—	—	↕	↕	↕	↕	1
	SHLL.W Rd	W		2									—	—	↕	↕	0	↕	1
	SHLL.W #2,Rd	W		2									—	—	↕	↕	↕	↕	1
	SHLL.L ERd	L		2									—	—	↕	↕	0	↕	1
	SHLL.L #2,ERd	L		2									—	—	↕	↕	0	↕	1

Module	Register	Abbreviation	R/W	Initial Value	Address*1
DTC	DTC mode register A	MRA	—*3	Undefined	—*4
	DTC mode register B	MRB	—*3	Undefined	—*4
	DTC source address register	SAR	—*3	Undefined	—*4
	DTC destination address register	DAR	—*3	Undefined	—*4
	DTC transfer count register A	CRA	—*3	Undefined	—*4
	DTC transfer count register B	CRB	—*3	Undefined	—*4
	DTC enable register	DTCER	R/W	H'00	H'FF30 to H'FF35
	DTC vector register	DTVECR	R/W	H'00	H'FF37
	Module stop control register	MSTPCR	R/W	H'3FFF	H'FF3C
DMAC0	Memory address register 0A	MAR0A	R/W	Undefined	H'FEE0
	I/O address register 0A	IOAR0A	R/W	Undefined	H'FEE4
	Transfer count register 0A	ETCR0A	R/W	Undefined	H'FEE6
	Memory address register 0B	MAR0B	R/W	Undefined	H'FEE8
	I/O address register 0B	IOAR0B	R/W	Undefined	H'FEEC
	Transfer count register 0B	ETCR0B	R/W	Undefined	H'FEEE
DMAC1	Memory address register 1A	MAR1A	R/W	Undefined	H'FEF0
	I/O address register 1A	IOAR1A	R/W	Undefined	H'FEF4
	Transfer count register 1A	ETCR1A	R/W	Undefined	H'FEF6
	Memory address register 1B	MAR1B	R/W	Undefined	H'FEF8
	I/O address register 1B	IOAR1B	R/W	Undefined	H'FEFC
	Transfer count register 1B	ETCR1B	R/W	Undefined	H'FEFE
Both DMAC channels	DMA write enable register	DMAWER	R/W	H'00	H'FF00
	DMA terminal control register	DMATCR	R/W	H'00	H'FF01
	DMA control register 0A	DMACR0A	R/W	H'00	H'FF02
	DMA control register 0B	DMACR0B	R/W	H'00	H'FF03
	DMA control register 1A	DMACR1A	R/W	H'00	H'FF04
	DMA control register 1B	DMACR1B	R/W	H'00	H'FF05
	DMA band control register	DMABCR	R/W	H'0000	H'FF06
	Module stop control register	MSTPCR	R/W	H'3FFF	H'FF3C

TIER5—Timer Interrupt Enable Register 5 H'FEA4 TPU5

Bit	7	6	5	4	3	2	1	0
	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
Initial value	0	1	0	0	0	0	0	0
Read/Write	R/W	—	R/W	R/W	—	—	R/W	R/W

TGR Interrupt Enable A

0	Interrupt request (TGIA) by TGFA bit disabled
1	Interrupt request (TGIA) by TGFA bit enabled

TGR Interrupt Enable B

0	Interrupt request (TGIB) by TGFB bit disabled
1	Interrupt request (TGIB) by TGFB bit enabled

Overflow Interrupt Enable

0	Interrupt request (TCIV) by TCFV disabled
1	Interrupt request (TCIV) by TCFV enabled

Underflow Interrupt Enable

0	Interrupt request (TCIU) by TCFU disabled
1	Interrupt request (TCIU) by TCFU enabled

A/D Conversion Start Request Enable

0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

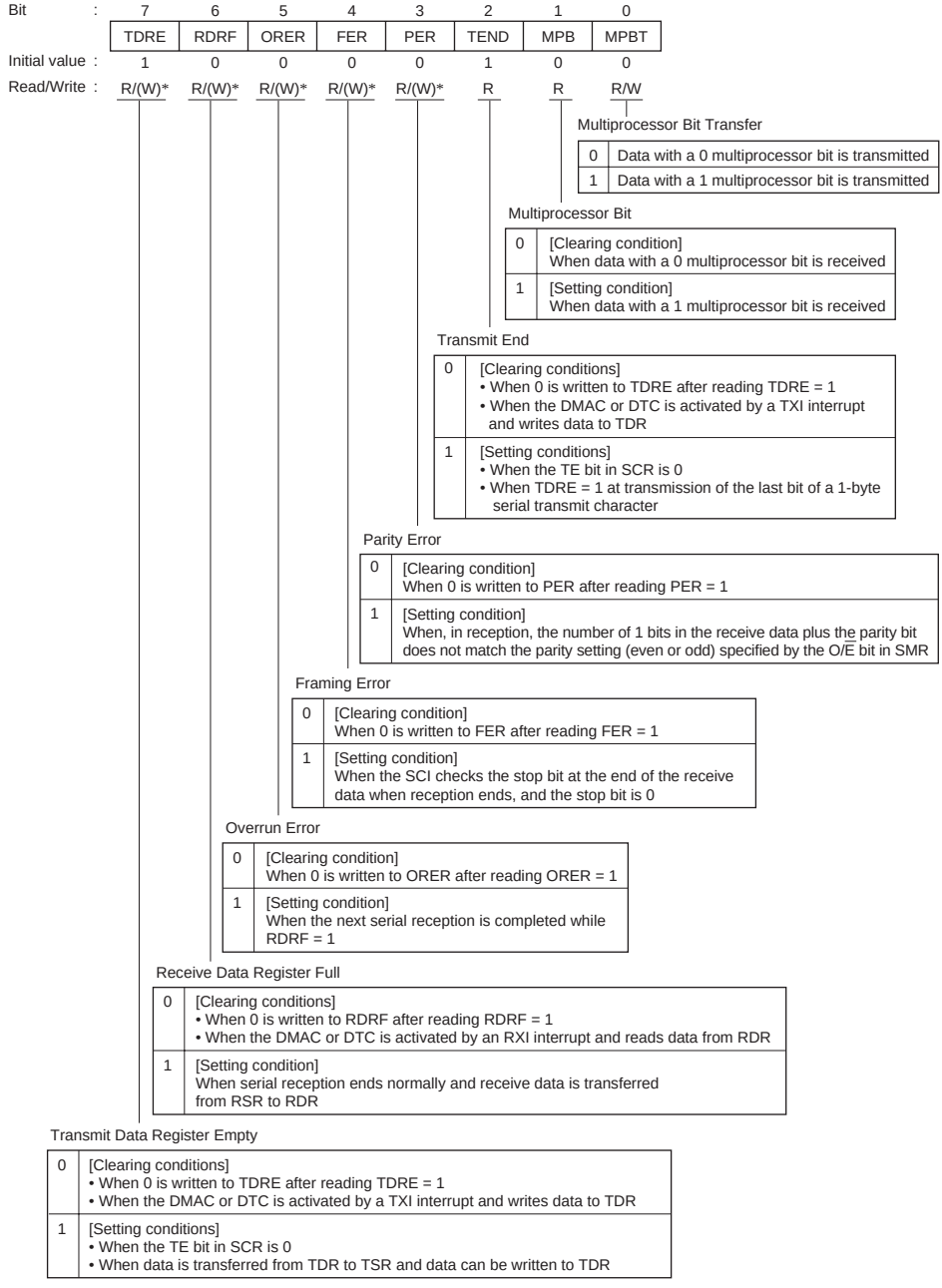
BRR0—Bit Rate Register 0 **H'FF79** **SCI0, Smart Card Interface 0**

Bit	:	7	6	5	4	3	2	1	0
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

Sets the serial transfer bit rate

Note: For details, see section 14.2.8, Bit Rate Register (BRR).

SSR0—Serial Status Register 0 H'FF7C SCI0



Note: * Can only be written with 0 for flag clearing.

FLMCR2—Flash Memory Control Register 2 **H'FFC9** **Flash Memory**
(Valid only in F-ZTAT version)

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

Flash Memory Error

0	Flash memory is operating normally Flash memory program/erase protection (error protection) is disabled [Clearing condition] Reset or hardware standby mode
1	An error has occurred during flash memory programming/erasing Flash memory program/erase protection (error protection) is enabled [Setting condition] See section 19.8.3, Error Protection for H8S/2339 F-ZTAT, and section 19.17.3, Error Protection for H8S/2338 F-ZTAT.

EBR1—Erase Block Register 1 **H'FFCA** **Flash Memory**
EBR2—Erase Block Register 2 **H'FFCB** **Flash Memory**
(Valid only in F-ZTAT version)

Bit	7	6	5	4	3	2	1	0
EBR1	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0
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

Bit	7	6	5	4	3	2	1	0
EBR2	—	—	EB13*	EB12*	EB11	EB10	EB9	EB8
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	— (R/W)*	— (R/W)*	R/W	R/W	R/W	R/W

Note: * Valid only in H8S/2339 F-ZTAT.

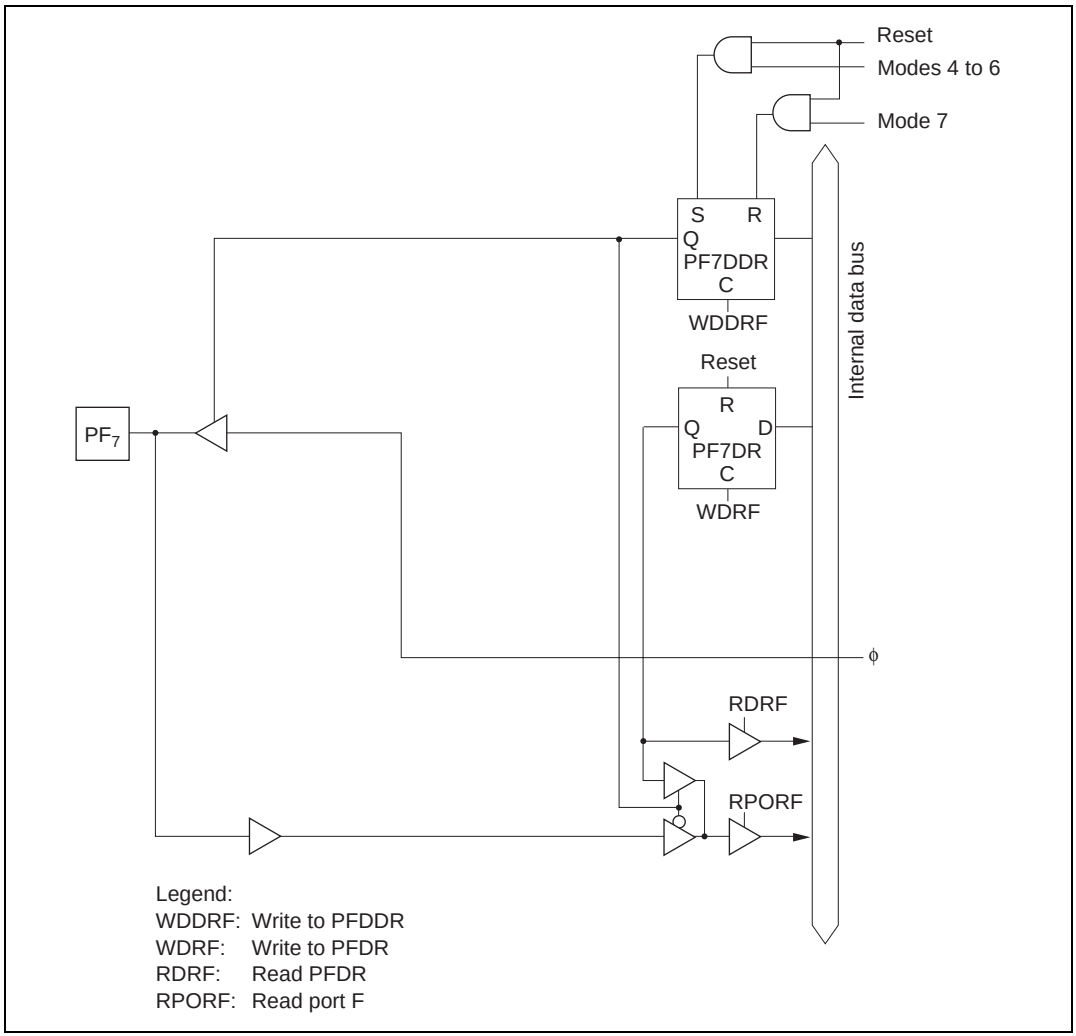


Figure C.15 (h) Port F Block Diagram (Pin PF_7)