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/2600
Core Size	16-Bit
Speed	20MHz
Connectivity	CANbus, SCI, SmartCard
Peripherals	Motor Control PWM, POR, PWM, WDT
Number of I/O	72
Program Memory Size	384KB (384K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 12x10b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-20°C ~ 75°C (TA)
Mounting Type	Surface Mount
Package / Case	128-BFQFP
Supplier Device Package	128-QFP (14x20)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/df2630f20v

7.2.4 Bus Control Register H (BCRH)

Bit	:	7	6	5	4	3	2	1	0
		ICIS1	ICIS0	BRSTRM	BRSTS1	BRSTS0	—	—	—
Initial value	:	1	1	0	1	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BCRH is an 8-bit readable/writable register that selects enabling or disabling of idle cycle insertion, and the memory interface for area 2 to 5, and 0.

BCRH is initialized to H'D0 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—Idle Cycle Insert 1 (ICIS1): Selects whether or not one idle cycle state is to be inserted between bus cycles when successive external read cycles are performed in different areas.

Bit 7

ICIS1	Description
0	Idle cycle not inserted in case of successive external read cycles in different areas
1	Idle cycle inserted in case of successive external read cycles in different areas (Initial value)

Bit 6—Idle Cycle Insert 0 (ICIS0): Selects whether or not one idle cycle state is to be inserted between bus cycles when successive external read and external write cycles are performed .

Bit 6

ICIS0	Description
0	Idle cycle not inserted in case of successive external read and external write cycles
1	Idle cycle inserted in case of successive external read and external write cycles (Initial value)

8.2.3 DTC Source Address Register (SAR)

Bit	:	23	22	21	20	19	- - -	4	3	2	1	0
Initial value:		*	*	*	*	*	- - -	*	*	*	*	*
R/W	:	—	—	—	—	—	- - -	—	—	—	—	—

*: Undefined

SAR is a 24-bit register that designates the source address of data to be transferred by the DTC. For word-size transfer, specify an even source address.

8.2.4 DTC Destination Address Register (DAR)

Bit	:	23	22	21	20	19	- - -	4	3	2	1	0
Initial value	:	*	*	*	*	*	- - -	*	*	*	*	*
R/W	:	—	—	—	—	—	- - -	—	—	—	—	—

*: Undefined

DAR is a 24-bit register that designates the destination address of data to be transferred by the DTC. For word-size transfer, specify an even destination address.

8.2.5 DTC Transfer Count Register A (CRA)

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W	:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		← CRAH →								← CRAL →							

*: Undefined

CRA is a 16-bit register that designates the number of times data is to be transferred by the DTC.

8.3.9 Operation Timing

Figures 8-10 to 8-12 show an example of DTC operation timing.

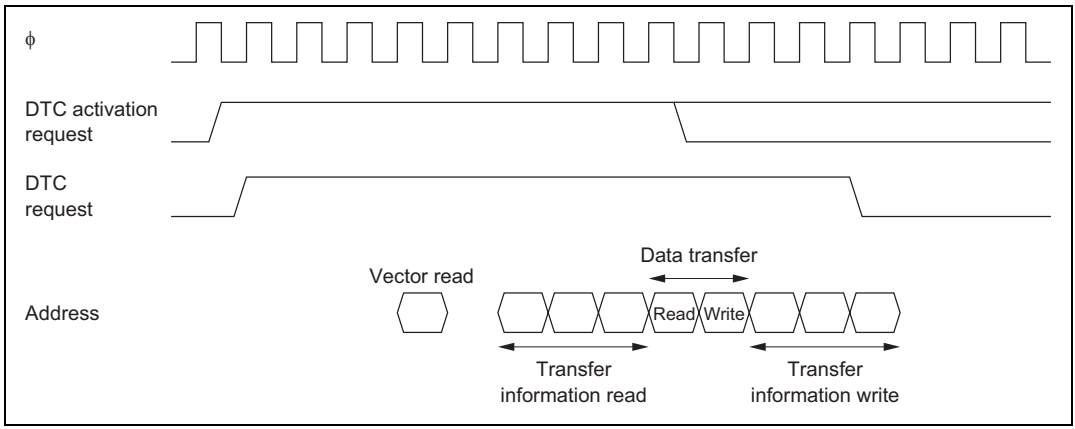

Figure 8-10 DTC Operation Timing (Example in Normal Mode or Repeat Mode)

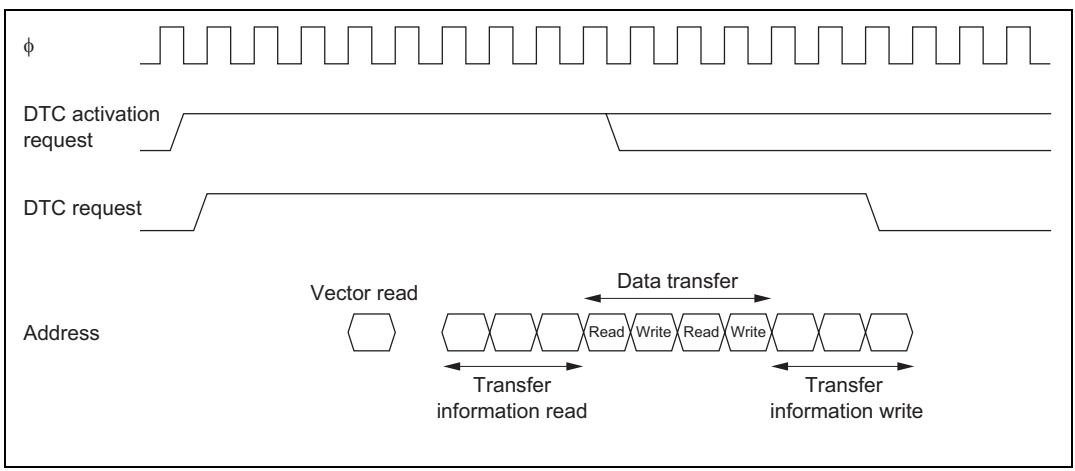

Figure 8-11 DTC Operation Timing (Example of Block Transfer Mode, with Block Size of 2)

8.4 Interrupts

An interrupt request is issued to the CPU when the DTC finishes the specified number of data transfers, or a data transfer for which the DISEL bit was set to 1. In the case of interrupt activation, the interrupt set as the activation source is generated. These interrupts to the CPU are subject to CPU mask level and interrupt controller priority level control.

In the case of activation by software, a software activated data transfer end interrupt (SWDTEND) is generated.

When the DISEL bit is 1 and one data transfer has ended, or the specified number of transfers have ended, after data transfer ends, the SWDTE bit is held at 1 and an SWDTEND interrupt is generated. The interrupt handling routine should clear the SWDTE bit to 0.

When the DTC is activated by software, an SWDTEND interrupt is not generated during a data transfer wait or during data transfer even if the SWDTE bit is set to 1.

8.5 Usage Notes

Module Stop: When the MSTPA6 bit in MSTPCRA is set to 1, the DTC clock stops, and the DTC enters the module stop state. However, 1 cannot be written in the MSTPA6 bit while the DTC is operating.

On-Chip RAM: The MRA, MRB, SAR, DAR, CRA, and CRB registers are all located in on-chip RAM. When the DTC is used, the RAME bit in SYSCR must not be cleared to 0.

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

Port	Description	Pins	Mode 4	Mode 5	Mode 6	Mode 7
Port E	• 8-bit I/O port • On-chip MOS input pull-up	PE7/D7 PE6/D6 PE5/D5 PE4/D4 PE3/D3 PE2/D2 PE1/D1 PE0/D0	In 8-bit-bus mode: I/O port In 16-bit-bus mode: data bus input/output			I/O port
Port F	• 6-bit I/O port • Schmitt-triggered input (PF3, PF0)	PF7/ϕ	When DDR = 0: input port When DDR = 1 (after reset): ϕ output			When DDR = 0 (after reset): input port When DDR = 1: ϕ output
		PF6/$\overline{AS}$ PF5/$\overline{RD}$ PF4/$\overline{HWR}$ PF3/$\overline{LWR}$/$\overline{ADTRG}$/$\overline{IRQ3}$	$\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$ outputs $\overline{ADTRG}$, $\overline{IRQ3}$ input			I/O port $\overline{ADTRG}$, $\overline{IRQ3}$ input
		PF0/$\overline{IRQ2}$	$\overline{IRQ2}$ input, I/O port			
Port H	• 8-bit I/O port	PH7/PWM1H PH6/PWM1G PH5/PWM1F PH4/PWM1E PH3/PWM1D PH2/PWM1C PH1/PWM1B PH0/PWM1A	Function as both Motor Control PWM Timer output pins and 8-bit I/O port.			
Port J	• 8-bit I/O port	PJ7/PWM2H PJ6/PWM2G PJ5/PWM2F PJ4/PWM2E PJ3/PWM2D PJ2/PWM2C PJ1/PWM2B PJ0/PWM2A	Function as both Motor Control PWM Timer output pins and 8-bit I/O port.			

Notes: 1. Pins for I^2C bus interface.
I^2C bus interface is available as an option in the H8S/2638, H8S/2639, and H8S/2630.
2. The PPG output is not implemented in the H8S/2635 Group.
3. The DA output is not implemented in the H8S/2635 Group.

Port B MOS Pull-Up Control Register (PBPCR)

Bit	:	7	6	5	4	3	2	1	0
		PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR
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

PBPCR is an 8-bit readable/writable register that controls the MOS input pull-up function incorporated into port B on an individual bit basis.

In modes 4 to 6, if a pin is in the input state in accordance with the settings in PFCR, in the TPU's TIOR, and in DDR, setting the corresponding PBPCR bit to 1 turns on the MOS input pull-up for that pin.

In mode 7, if a pin is in the input state in accordance with the settings in the TPU's TIOR and in DDR, setting the corresponding PBPCR bit to 1 turns on the MOS input pull-up for that pin.

PBPCR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Port B Open Drain Control Register (PBODR)

Bit	:	7	6	5	4	3	2	1	0
		PB7ODR	PB6ODR	PB5ODR	PB4ODR	PB3ODR	PB2ODR	PB1ODR	PB0ODR
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

PBODR is an 8-bit readable/writable register that controls the PMOS on/off state for each port B pin (PB7 to PB0).

When pins are not address outputs in accordance with the setting of bits AE3 to AE0 in PFCR, setting a PBODR bit makes the corresponding port B pin an NMOS open-drain output, while clearing the bit to 0 makes the pin a CMOS output.

PBODR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Bit 6—Receive Data Register Full (RDRF): Indicates that the received data is stored in RDR.

Bit 6

RDRF	Description
0	[Clearing conditions] (Initial value) • When 0 is written to RDRF after reading RDRF = 1 • When the DTC is activated by an RXI interrupt and reads data from RDR
1	[Setting condition] • When serial reception ends normally and receive data is transferred from RSR to RDR

Note: RDR and the RDRF flag are not affected and retain their previous values when an error is detected during reception or when the RE bit in SCR is cleared to 0.
If reception of the next data is completed while the RDRF flag is still set to 1, an overrun error will occur and the receive data will be lost.

Bit 5—Overrun Error (ORER): Indicates that an overrun error occurred during reception, causing abnormal termination.

Bit 5

ORER	Description
0	[Clearing condition] (Initial value)*1 • When 0 is written to ORER after reading ORER = 1
1	[Setting condition] • When the next serial reception is completed while RDRF = 1*2

Notes: 1. The ORER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.
2. The receive data prior to the overrun error is retained in RDR, and the data received subsequently is lost. Also, subsequent serial reception cannot be continued while the ORER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either.

13.2.8 Bit Rate Register (BRR)

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

BRR is an 8-bit register that sets the serial transfer bit rate in accordance with the baud rate generator operating clock selected by bits CKS1 and CKS0 in SMR.

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

BRR is initialized to H'FF by a reset and in standby mode.

As baud rate generator control is performed independently for each channel, different values can be set for each channel.

Table 13-3 shows sample BRR settings in asynchronous mode, and table 13-4 shows sample BRR settings in clocked synchronous mode.

Table 13-3 BRR Settings for Various Bit Rates (Asynchronous Mode)

	ϕ = 4 MHz			ϕ = 4.9152 MHz			ϕ = 5 MHz		
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	2	70	0.03	2	86	0.31	2	88	–0.25
150	1	207	0.16	1	255	0.00	2	64	0.16
300	1	103	0.16	1	127	0.00	1	129	0.16
600	0	207	0.16	0	255	0.00	1	64	0.16
1200	0	103	0.16	0	127	0.00	0	129	0.16
2400	0	51	0.16	0	63	0.00	0	64	0.16
4800	0	25	0.16	0	31	0.00	0	32	–1.36
9600	0	12	0.16	0	15	0.00	0	15	1.73
19200	—	—	—	0	7	0.00	0	7	1.73
31250	0	3	0.00	0	4	–1.70	0	4	0.00
38400	—	—	—	0	3	0.00	0	3	1.73

14.3.6 Data Transfer Operations

Initialization: Before transmitting and receiving data, initialize the SCI as described below. Initialization is also necessary when switching from transmit mode to receive mode, or vice versa.

[1] Clear the TE and RE bits in SCR to 0.

[2] Clear the error flags ERS, PER, and ORER in SSR to 0.

[3] Set the GM, BLK, O/$\overline{E}$, BCP1, BCP0, CKS1, CKS0 bits in SMR. Set the PE bit to 1.

[4] Set the SMIF, SDIR, and SINV bits in SCMR.
When the SMIF bit is set to 1, the TxD and RxD pins are both switched from ports to SCI pins, and are placed in the high-impedance state.

[5] Set the value corresponding to the bit rate in BRR.

[6] Set the CKE0 and CKE1 bits in SCR. Clear the TIE, RIE, TE, RE, MPIE, and TEIE bits to 0.
If the CKE0 bit is set to 1, the clock is output from the SCK pin.

[7] Wait at least one bit interval, then set the TIE, RIE, TE, and RE bits in SCR. Do not set the TE bit and RE bit at the same time, except for self-diagnosis.

19.2.3 PWM Polarity Registers 1 and 2 (PWPR1, PWPR2)

PWPR1

Bit	7	6	5	4	3	2	1	0
	OPS1H	OPS1G	OPS1F	OPS1E	OPS1D	OPS1C	OPS1B	OPS1A
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

PWPR2

Bit	7	6	5	4	3	2	1	0
	OPS2H	OPS2G	OPS2F	OPS2E	OPS2D	OPS2C	OPS2B	OPS2A
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

PWPR is an 8-bit read/write register that selects the PWM output polarity. PWPR1 controls outputs PWM1H to PWM1A, and PWPR2 controls outputs PWM2H to PWM2A.

PWPR is initialized to H'00 upon reset, and in standby mode, watch mode*, subactive mode*, subsleep mode*, and module stop mode.

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions, and H8S/2635 Group only.
These functions cannot be used with the other versions.

Bits 7 to 0—Output Polarity Select (OPS): Each of these bits selects the polarity of the corresponding PWM output.

Bits 7 to 0:

OPS	Description	
0	PWM direct output	(Initial value)
1	PWM inverse output	

Table 21B-7 Flash Memory Erase Blocks

Block (Size)	Addresses
EB0 (4 kbytes)	H'000000 to H'000FFF
EB1 (4 kbytes)	H'001000 to H'001FFF
EB2 (4 kbytes)	H'002000 to H'002FFF
EB3 (4 kbytes)	H'003000 to H'003FFF
EB4 (4 kbytes)	H'004000 to H'004FFF
EB5 (4 kbytes)	H'005000 to H'005FFF
EB6 (4 kbytes)	H'006000 to H'006FFF
EB7 (4 kbytes)	H'007000 to H'007FFF
EB8 (32 kbytes)	H'008000 to H'00FFFF
EB9 (64 kbytes)	H'010000 to H'01FFFF
EB10 (64 kbytes)	H'020000 to H'02FFFF
EB11 (64 kbytes)	H'030000 to H'03FFFF
EB12 (64 kbytes)*	H'040000 to H'04FFFF
EB13 (64 kbytes))*	H'050000 to H'05FFFF

Note: * This function is not available in the H8S/2638 and H8S/2639.

21B.7.5 RAM Emulation Register (RAMER)

RAMER specifies the area of flash memory to be overlapped with part of RAM when emulating real-time flash memory programming. RAMER initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode. RAMER settings should be made in user mode or user program mode.

Flash memory area divisions are shown in table 21B-8. To ensure correct operation of the emulation function, the ROM for which RAM emulation is performed should not be accessed immediately after this register has been modified. Normal execution of an access immediately after register modification is not guaranteed.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	RAMS	RAM2	RAM1	RAM0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

21B.9.2 Program-Verify Mode

In program-verify mode, the data written in program mode is read to check whether it has been correctly written in the flash memory.

After the elapse of the given programming time, clear the P bit in FLMCR1, then wait for at least (t_{cp}) µs before clearing the PSU bit to exit program mode. After exiting program mode, the watchdog timer setting is also cleared. The operating mode is then switched to program-verify mode by setting the PV bit in FLMCR1. Before reading in program-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of (t_{spv}) µs or more. When the flash memory is read in this state (verify data is read in 16-bit units), the data at the latched address is read. Wait at least (t_{spvr}) µs after the dummy write before performing this read operation. Next, the originally written data is compared with the verify data, and reprogram data is computed (see figure 21B-12) and transferred to RAM. After verification of 128 bytes of data has been completed, exit program-verify mode, wait for at least (t_{cpv}) µs, then clear the SWE bit in FLMCR1. If reprogramming is necessary, set program mode again, and repeat the program/program-verify sequence as before. The maximum number of repetitions of the program/program-verify sequence is indicated by the maximum programming count (N). Leave a wait time of at least (t_{cswe}) µs after clearing SWE.

Notes on Program/Program-Verify Procedure

1. In order to perform 128-byte-unit programming, the lower 8 bits of the write start address must be H'00 or H'80.

2. When performing continuous writing of 128-byte data to flash memory, byte-unit transfer should be used.

   128-byte data transfer is necessary even when writing fewer than 128 bytes of data. Write H'FF data to the extra addresses.

3. Verify data is read in word units.

4. The write pulse is applied and a flash memory write executed while the P bit in FLMCR1 is set. In the chip, write pulses should be applied as follows in the program/program-verify procedure to prevent voltage stress on the device and loss of write data reliability.
   a. After write pulse application, perform a verify-read in program-verify mode and apply a write pulse again for any bits read as 1 (reprogramming processing). When all the 0-write bits in the 128-byte write data are read as 0 in the verify-read operation, the program/program-verify procedure is completed. In the chip, the number of loops in reprogramming processing is guaranteed not to exceed the maximum value of the maximum programming count (N).

3. The H8S/2635 has 192 kbytes of on-chip ROM. The H8S/2634 has 128 kbytes of on-chip ROM.

Table 21C-3 Operating Modes and ROM (Mask ROM Version)

		Mode Pins			
	Operating Mode	**MD2**	**MD1**	**MD0**	**On-Chip ROM**
Mode 0	—	0	0	0	—
Mode 1				1	
Mode 2			1	0	
Mode 3				1	
Mode 4	Advanced expanded mode with on-chip ROM disabled	1	0	0	Disabled
Mode 5	Advanced expanded mode with on-chip ROM disabled			1	
Mode 6	Advanced expanded mode with on-chip ROM enabled		1	0	Enabled (192 kbytes/ 128 kbytes)*
Mode 7	Advanced single-chip mode			1	Enabled (192 kbytes/ 128 kbytes)*

Note: * The H8S/2635 has 192 kbytes of on-chip ROM. The H8S/2634 has 128 kbytes of on-chip ROM.

- Programmer mode

  Flash memory can be programmed/erased in programmer mode, using a PROM programmer, as well as in on-board programming mode.

21C.4.2 Block Diagram

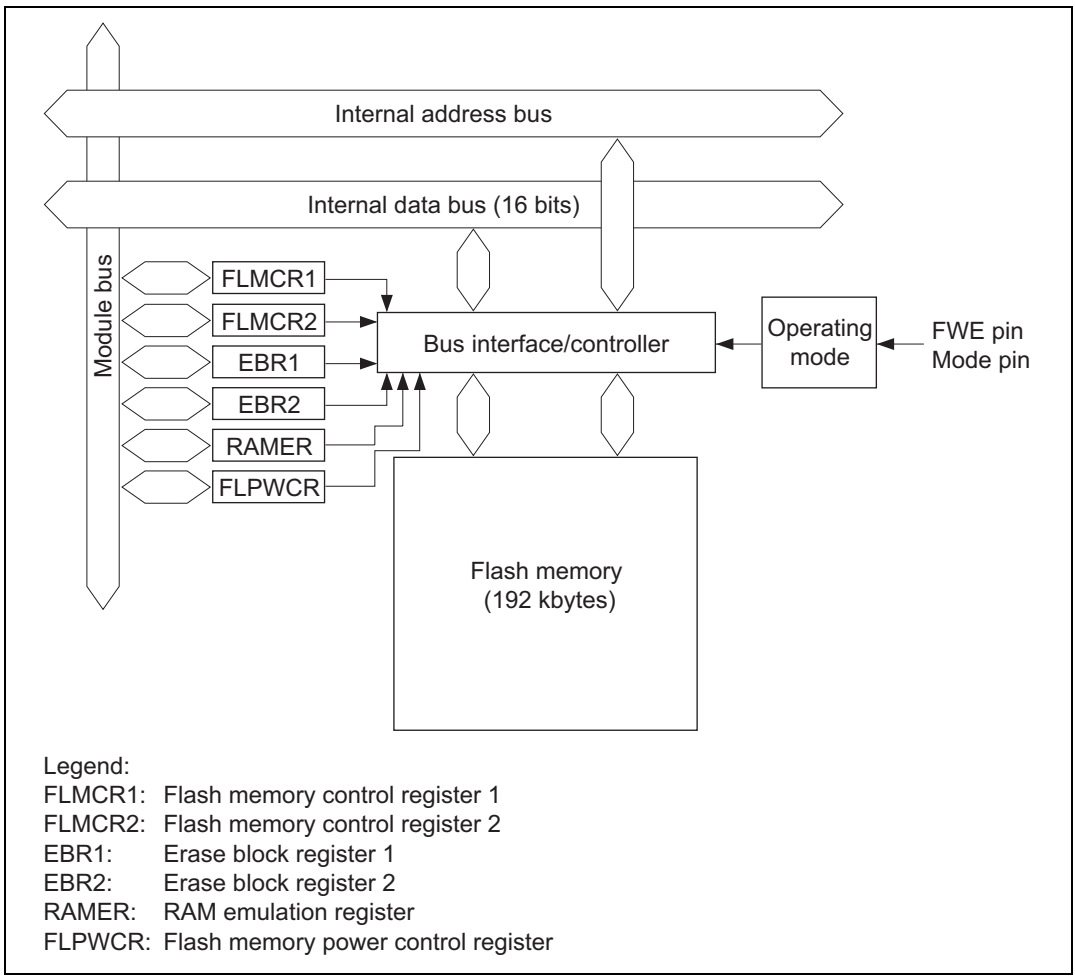

Figure 21C-2 Block Diagram of Flash Memory

Table A-6 Instruction Execution Cycles

Instruction	1	2	3	4	5	6	7	8	9
ADD.B #xx:8,Rd	R:W NEXT								
ADD.B Rs,Rd	R:W NEXT								
ADD.W #xx:16,Rd	R:W 2nd	R:W NEXT							
ADD.W Rs,Rd	R:W NEXT								
ADD.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT						
ADD.L ERs,ERd	R:W NEXT								
ADDS #1/2/4,ERd	R:W NEXT								
ADDX #xx:8,Rd	R:W NEXT								
ADDX Rs,Rd	R:W NEXT								
AND.B #xx:8,Rd	R:W NEXT								
AND.B Rs,Rd	R:W NEXT								
AND.W #xx:16,Rd	R:W 2nd	R:W NEXT							
AND.W Rs,Rd	R:W NEXT								
AND.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT						
AND.L ERs,ERd	R:W 2nd	R:W NEXT							
ANDC #xx:8,CCR	R:W NEXT								
ANDC #xx:8,EXR	R:W 2nd	R:W NEXT							
BAND #xx:3,Rd	R:W NEXT								
BAND #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BAND #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BAND #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BAND #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BRA d:8 (BT d:8)	R:W NEXT	R:W EA							
BRN d:8 (BF d:8)	R:W NEXT	R:W EA							
BHI d:8	R:W NEXT	R:W EA							
BLS d:8	R:W NEXT	R:W EA							
BCC d:8 (BHS d:8)	R:W NEXT	R:W EA							
BCS d:8 (BLO d:8)	R:W NEXT	R:W EA							
BNE d:8	R:W NEXT	R:W EA							
BEQ d:8	R:W NEXT	R:W EA							
BVC d:8	R:W NEXT	R:W EA							
BVS d:8	R:W NEXT	R:W EA							
BPL d:8	R:W NEXT	R:W EA							
BMI d:8	R:W NEXT	R:W EA							
BGE d:8	R:W NEXT	R:W EA							
BLT d:8	R:W NEXT	R:W EA							
BGT d:8	R:W NEXT	R:W EA							

MC8[1]—Message Control 8[1]	**H'F860**	**HCAN0**
MC8[2]—Message Control 8[2]	**H'F861**	**HCAN0**
MC8[3]—Message Control 8[3]	**H'F862**	**HCAN0**
MC8[4]—Message Control 8[4]	**H'F863**	**HCAN0**
MC8[5]—Message Control 8[5]	**H'F864**	**HCAN0**
MC8[6]—Message Control 8[6]	**H'F865**	**HCAN0**
MC8[7]—Message Control 8[7]	**H'F866**	**HCAN0**
MC8[8]—Message Control 8[8]	**H'F867**	**HCAN0**

MC8[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 bytes
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC8[2]

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

MC8[3]

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

MD12[1]—Message Data 12[1]	**H'F910**	**HCAN0**
MD12[2]—Message Data 12[2]	**H'F911**	**HCAN0**
MD12[3]—Message Data 12[3]	**H'F912**	**HCAN0**
MD12[4]—Message Data 12[4]	**H'F913**	**HCAN0**
MD12[5]—Message Data 12[5]	**H'F914**	**HCAN0**
MD12[6]—Message Data 12[6]	**H'F915**	**HCAN0**
MD12[7]—Message Data 12[7]	**H'F916**	**HCAN0**
MD12[8]—Message Data 12[8]	**H'F917**	**HCAN0**

	Bit	7	6	5	4	3	2	1	0
MDx[1]									
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[2]									
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[3]									
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[4]									
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[5]									
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[6]									
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[7]									
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[8]									
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

SCKCR—System Clock Control Register H'FDE6 System

Bit	7	6	5	4	3	2	1	0
	PSTOP	—	—	—	STCS	SCK2	SCK1	SCK0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	—	—	—	R/W	R/W	R/W	R/W

System Clock Select

0	0	0	Bus master in high-speed mode
		1	Medium-speed clock is ϕ/2
	1	0	Medium-speed clock is ϕ/4
		1	Medium-speed clock is ϕ/8
1	0	0	Medium-speed clock is ϕ/16
		1	Medium-speed clock is ϕ/32
	1	—	—

Frequency Multiplication Factor Switching Mode Select

0	Specified multiplication factor is valid after transition to software standby mode, watch mode*, or subactive mode*
1	Specified multiplication factor is valid immediately after STC bits are rewritten

ϕ Clock Output Disable

DDR	0	1	1
PSTOP	—	0	1
Hardware standby mode	High impedance	High impedance	High impedance
Software standby mode, watch mode*, and direct transition	High impedance	Fixed high	Fixed high
Sleep mode and subsleep mode*	High impedance	ϕ output	Fixed high
High-speed mode, medium-speed mode, and subactive mode*	High impedance	ϕ output	Fixed high

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions, and H8S/2635 Group only.
These functions cannot be used with the other versions.

Appendix F Product Code Lineup

Table F-1 H8S/2636, H8S/2638, H8S/2639, and H8S/2630 Product Code Lineup

Product Type		Part No.	Mark Code	Functions	Packages
H8S/2636	F-ZTAT version	HD64F2636	HD64F2636F	No subclock function	128-pin QFP (FP-128B)
			HD64F2636UF	Subclock function	128-pin QFP (FP-128B)
	Mask ROM version	HD6432636	HD6432636F	No subclock function	128-pin QFP (FP-128B)
			HD6432636UF	Subclock function	128-pin QFP (FP-128B)
H8S/2638	F-ZTAT version	HD64F2638	HD64F2638F	No subclock function or I^2C bus interface	128-pin QFP (FP-128B)
			HD64F2638UF	Subclock function, no I^2C bus interface	128-pin QFP (FP-128B)
			HD64F2638WF	Subclock function and I^2C bus interface	128-pin QFP (FP-128B)
	Mask ROM version	HD6432638	HD6432638F	No subclock function or I^2C bus interface	128-pin QFP (FP-128B)
			HD6432638UF	Subclock function, no I^2C bus interface	128-pin QFP (FP-128B)
			HD6432638WF	Subclock function and I^2C bus interface	128-pin QFP (FP-128B)
H8S/2639	F-ZTAT version	HD64F2639	HD64F2639UF	Subclock function, no I^2C bus interface	128-pin QFP (FP-128B)
			HD64F2639WF	Subclock function and I^2C bus interface	128-pin QFP (FP-128B)
	Mask ROM version	HD6432639	HD6432639UF	Subclock function, no I^2C bus interface	128-pin QFP (FP-128B)
			HD6432639WF	Subclock function and I^2C bus interface	128-pin QFP (FP-128B)