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	86
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 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	128-BFQFP
Supplier Device Package	128-QFP (14x20)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/df2328bvf25v

2.2 CPU Operating Modes

The H8S/2329 and H8S/2328 Group CPU has advanced operating mode. Advanced mode supports a maximum 16-Mbyte total address space (architecturally a maximum 16-Mbyte program area and a maximum of 4 Gbytes for program and data areas combined). The mode is selected by the mode pins of the microcontroller.

Advanced Mode

Address Space: Linear access is provided to a 16-Mbyte maximum address space (architecturally a maximum 16-Mbyte program area and a maximum 4-Gbyte data area, with a maximum of 4 Gbytes for program and data areas combined).

Extended Registers (En): The extended registers (E0 to E7) can be used as 16-bit registers, or as the upper 16-bit segments of 32-bit registers or address registers.

Instruction Set: All instructions and addressing modes can be used.

2.4 Register Configuration

2.4.1 Overview

The CPU has the internal registers shown in figure 2.4. There are two types of registers: general registers and control registers.

General Registers (Rn) and Extended Registers (En)

	15	07	07	0
ER0	E0	R0H	R0L	
ER1	E1	R1H	R1L	
ER2	E2	R2H	R2L	
ER3	E3	R3H	R3L	
ER4	E4	R4H	R4L	
ER5	E5	R5H	R5L	
ER6	E6	R6H	R6L	
ER7 (SP)	E7	R7H	R7L	

Control Registers (CR)

23	0
PC	

	7	6	5	4	3	2	1	0
EXR	T	—	—	—	—	I2	I1	I0

	7	6	5	4	3	2	1	0
CCR	I	UI	H	U	N	Z	V	C

Legend:

SP:	Stack pointer	H:	Half-carry flag
PC:	Program counter	U:	User bit
EXR:	Extended control register	N:	Negative flag
T:	Trace bit	Z:	Zero flag
I2 to I0:	Interrupt mask bits	V:	Overflow flag
CCR:	Condition-code register	C:	Carry flag
I:	Interrupt mask bit		
UI:	User bit or interrupt mask bit*		

Note: * In the H8S/2329 Group and H8S/2328 Group, this bit cannot be used as an interrupt mask.

Figure 2.4 CPU Registers

Bit 3 MXC1	Bit 2 MXC0	Description
0	0	8-bit shift (Initial value) • When 8-bit access space is designated: Row address A_{23} to A_8 used for comparison • When 16-bit access space is designated: Row address A_{23} to A_9 used for comparison
	1	9-bit shift • When 8-bit access space is designated: Row address A_{23} to A_9 used for comparison • When 16-bit access space is designated: Row address A_{23} to A_{10} used for comparison
1	0	10-bit shift • When 8-bit access space is designated: Row address A_{23} to A_{10} used for comparison • When 16-bit access space is designated: Row address A_{23} to A_{11} used for comparison
	1	—

Bits 1 and 0—Refresh Cycle Wait Control 1 and 0 (RLW1, RLW0): These bits select the number of wait states to be inserted in a DRAM interface CAS-before-RAS refresh cycle. This setting is used for all areas designated as DRAM space. Wait input by the $\overline{\text{WAIT}}$ pin is disabled.

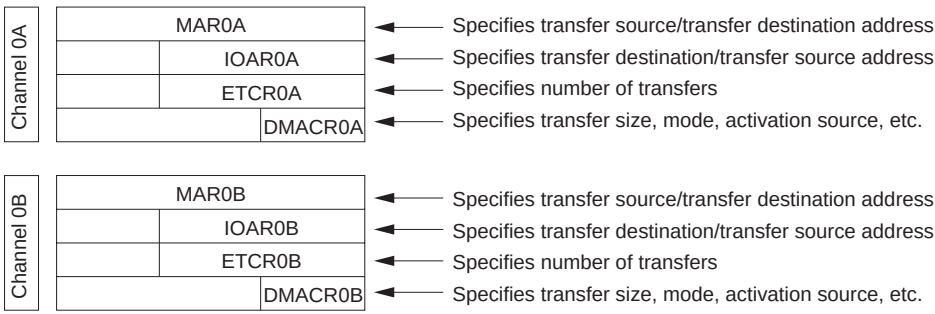
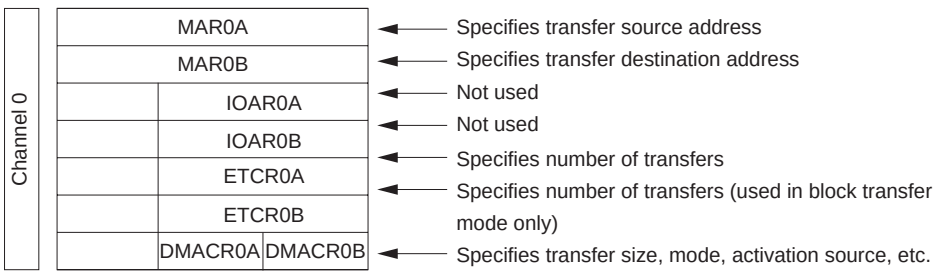
Bit 1 RLW1	Bit 0 RLW0	Description
0	0	No wait state inserted (Initial value)
	1	1 wait state inserted
1	0	2 wait states inserted
	1	3 wait states inserted

7.2 Register Descriptions (1) (Short Address Mode)

Short address mode transfer can be performed for channels A and B independently.

Short address mode transfer is specified for each channel by clearing the FAE bit in DMABCR to 0, as shown in table 7.4. Short address mode or full address mode can be selected for channels 1 and 0 independently by means of bits FAE1 and FAE0.

Table 7.4 Short Address Mode and Full Address Mode (For 1 Channel: Example of Channel 0)

FAE0	Description
0	Short address mode specified (channels A and B operate independently) 
1	Full address mode specified (channels A and B operate in combination) 

7.3.4 DMA Control Register (DMACR)

DMACR is a 16-bit readable/writable register that controls the operation of each DMAC channel. In full address mode, DMACRA and DMACRB have different functions.

DMACR is initialized to H'0000 by a reset, and in hardware standby mode.

DMACRA

Bit	:	15	14	13	12	11	10	9	8
		DTSZ	SAID	SAIDE	BLKDIR	BLKE	—	—	—
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

DMACRB

Bit	:	7	6	5	4	3	2	1	0
		—	DAID	DAIDE	—	DTF3	DTF2	DTF1	DTF0
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—Data Transfer Size (DTSZ): Selects the size of data to be transferred at one time.

Bit 15 DTSZ	Description	
0	Byte-size transfer	(Initial value)
1	Word-size transfer	

7.5.15 NMI Interrupts and DMAC

When an NMI interrupt is requested, burst mode transfer in full address mode is interrupted. An NMI interrupt does not affect the operation of the DMAC in other modes.

In full address mode, transfer is enabled for a channel when both the DTE bit and the DTME bit are set to 1. With burst mode setting, the DTME bit is cleared when an NMI interrupt is requested.

If the DTME bit is cleared during burst mode transfer, the DMAC discontinues transfer on completion of the 1-byte or 1-word transfer in progress, then releases the bus, which passes to the CPU.

The channel on which transfer was interrupted can be restarted by setting the DTME bit to 1 again. Figure 7.36 shows the procedure for continuing transfer when it has been interrupted by an NMI interrupt on a channel designated for burst mode transfer.

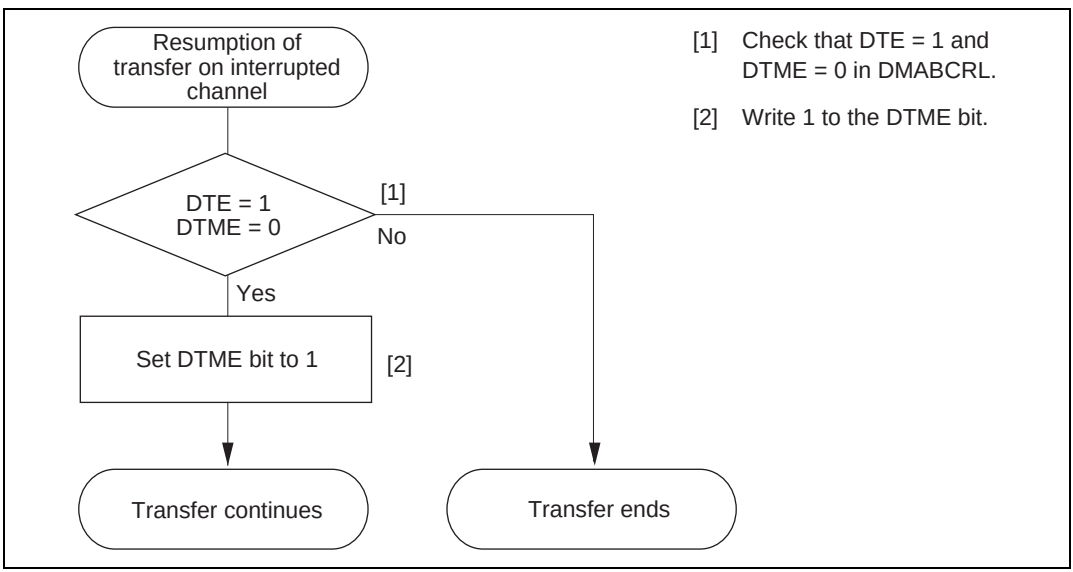


Figure 7.36 Example of Procedure for Continuing Transfer on Channel Interrupted by NMI Interrupt

Port	Description	Pins	Mode 4[*1]	Mode 5[*1]	Mode 6	Mode 7
Port F	• 8-bit I/O port	$PF_1/\overline{BACK}$	When BRLE = 0 (after reset): I/O port When BRLE = 1: $\overline{BREQ}$ input, $\overline{BACK}$ output			I/O ports
		$PF_0/\overline{BREQ}$				
Port G	• 5-bit I/O port	$PG_4/\overline{CS_0}$	When DDR = 0[*3]: input port When DDR = 1[*4]: $\overline{CS0}$ output			I/O ports
		$PG_3/\overline{CS_1}$	When DDR = 0 (after reset): input port When CS167E = 0 and DDR = 1: output port When CS167E = 1 and DDR = 1: $\overline{CS1}$ output			
		$PG_2/\overline{CS_2}$	When DDR = 0 (after reset): input port When CS25E = 0 and DDR = 1: output port When CS25E = 1 and DDR = 1: $\overline{CS2}$ output			
		$PG_1/\overline{CS_3}$	When DDR = 0 (after reset): input port When CS25E = 0 and DDR = 1: output port When CS25E = 1 and DDR = 1: $\overline{CS3}$ output			
		$PG_0/\overline{CAS}$[*2]	DRAM space set: $\overline{CAS}$ output Otherwise (after reset): I/O port			

Notes: 1. Only modes 4 and 5 are provided in the ROMless version.
2. The $\overline{DACK1}$, $\overline{DACK0}$, $\overline{TEND1}$, $\overline{DREQ1}$, $\overline{TEND0}$, $\overline{DREQ0}$ and $\overline{LCAS}$ are not supported in the H8S/2321.
3. After a reset in mode 6.
4. After a reset in mode 4 or 5.

9.6 Port 5

9.6.1 Overview

Port 5 is a 4-bit I/O port. Port 5 pins also function as SCI I/O pins (TxD_2, RxD_2, and SCK_2), the A/D converter input pin ($\overline{ADTRG}$), interrupt input pins ($\overline{IRQ_4}$ to $\overline{IRQ_7}$), and bus control signal I/O pins ($\overline{WAIT}$ and $\overline{BREQO}$). The pin functions can be switched by means of settings in PFCR2 and SYSCR. $\overline{IRQ_4}$ to $\overline{IRQ_7}$ only are Schmitt-triggered inputs. Figure 9.5 shows the port 5 pin configuration.

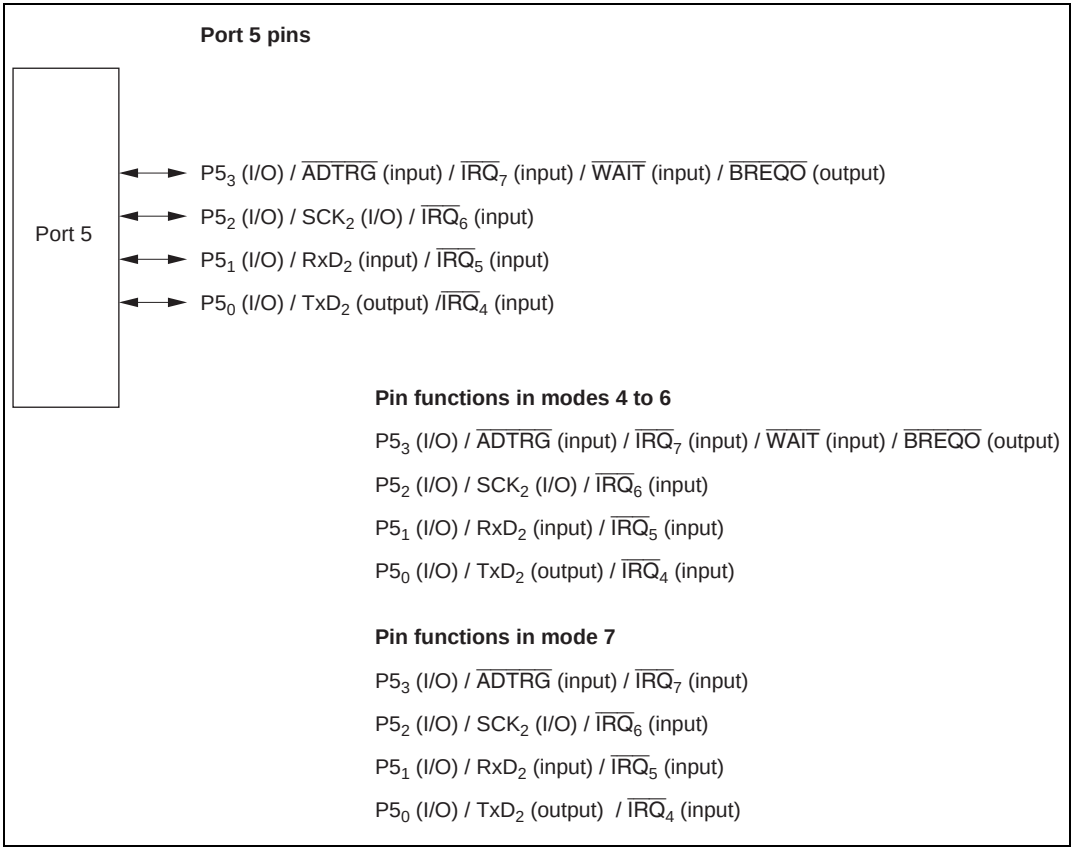


Figure 9.5 Port 5 Pin Functions

Bits 7 to 5—Counter Clear 2 to 0 (CCLR2 to CCLR0): These bits select the TCNT counter clearing source.

Channel	Bit 7 CCLR2	Bit 6 CCLR1	Bit 5 CCLR0	Description
0, 3	0	0	0	TCNT clearing disabled (Initial value)
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*1
	1	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRC compare match/input capture*2
		1	0	TCNT cleared by TGRD compare match/input capture*2
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*1

Channel	Bit 7 Reserved*3	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2, 4, 5	0	0	0	TCNT clearing disabled (Initial value)
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*1

Notes: 1. Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.
2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority, and compare match/input capture does not occur.
3. Bit 7 is reserved in channels 1, 2, 4, and 5. It is always read as 0 and cannot be modified.

Section 14 Serial Communication Interface (SCI)

14.1 Overview

The chip is equipped with a serial communication interface (SCI) that can handle both asynchronous and synchronous serial communication. A function is also provided for serial communication between processors (multiprocessor communication function).

14.1.1 Features

SCI features are listed below.

- Choice of asynchronous or synchronous serial communication mode

  Asynchronous mode
  - Serial data communication executed using an asynchronous system in which synchronization is achieved character by character
  - Serial data communication can be carried out with standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA)
  - A multiprocessor communication function is provided that enables serial data communication with a number of processors
  - Choice of 12 serial data transfer formats

    Data length: 7 or 8 bits

    Stop bit length: 1 or 2 bits

    Parity: Even, odd, or none

    Multiprocessor bit: 1 or 0
  - Receive error detection: Parity, overrun, and framing errors
  - Break detection: Break can be detected by reading the RxD pin level directly in case of a framing error

  Synchronous mode
  - Serial data communication synchronized with a clock
  - Serial data communication can be carried out with other chips that have a synchronous communication function
  - One serial data transfer format

    Data length: 8 bits
  - Receive error detection: Overrun errors detected

With the above processing, interrupt handling or data transfer by the DMAC* or DTC is possible.

If reception ends and the RDRF flag is set to 1 while the RIE bit is set to 1 and interrupt requests are enabled, a receive data full interrupt (RXI) request will be generated. If an error occurs in reception and either the ORER flag or the PER flag is set to 1, a transmit/receive-error interrupt (ERI) request will be generated.

If the DMAC* or DTC is activated by an RXI request, the receive data in which the error occurred is skipped, and only the number of bytes of receive data set in the DMAC* or DTC are transferred.

For details, see Interrupt Operation and Data Transfer Operation by DMAC* or DTC below.

If a parity error occurs during reception and the PER is set to 1, the received data is still transferred to RDR, and therefore this data can be read.

Notes: For details of operation in block transfer mode, see section 14.3.2, Operation in Asynchronous Mode.
* The DMAC is not supported in the H8S/2321.

Mode Switching Operation: When switching from receive mode to transmit mode, first confirm that the receive operation has been completed, then start from initialization, clearing RE bit to 0 and setting TE bit to 1. The RDRF flag or the PER and ORER flags can be used to check that the receive operation has been completed.

When switching from transmit mode to receive mode, first confirm that the transmit operation has been completed, then start from initialization, clearing TE bit to 0 and setting RE bit to 1. The TEND flag can be used to check that the transmit operation has been completed.

Fixing Clock Output: When the GSM bit in SMR is set to 1, the clock output can be fixed with bits CKE1 and CKE0 in SCR. At this time, the minimum clock pulse width can be made the specified width.

Figure 15.8 shows the timing for fixing the clock output. In this example, GSM is set to 1, CKE1 is cleared to 0, and the CKE0 bit is controlled.

- Before branching to the programming control program (RAM area H'FFE400 to H'FFFBFF), the chip terminates transmit and receive operations by the on-chip SCI (channel 1) (by clearing the RE and TE bits in SCR to 0), but the adjusted bit rate value remains set in BRR. The transmit data output pin, TxD1, goes to the high-level output state (P31DDR = 1, P31DR = 1).

- The contents of the CPU's internal general registers are undefined at this time, so these registers must be initialized immediately after branching to the programming control program. In particular, since the stack pointer (SP) is used implicitly in subroutine calls, etc., a stack area must be specified for use by the programming control program.

- Initial settings must also be made for the other on-chip registers.

- Boot mode can be entered by making the pin settings shown in table 19.30 and executing a reset-start.

- Boot mode can be cleared by driving the reset pin low, waiting at least 20 states, then setting the FWE pin and mode pins, and executing reset release*1. Boot mode can also be cleared by a WDT overflow reset.

- Do not change the mode pin input levels in boot mode, and do not drive the FWE pin low while the boot program is being executed or while flash memory is being programmed or erased*2.

- If the mode pin input levels are changed (for example, from low to high) during a reset, the state of ports with multiplexed address functions and bus control output pins ($\overline{AS}$, $\overline{RD}$, $\overline{HWR}$) will change according to the change in the microcomputer's operating mode*3.

  Therefore, care must be taken to make pin settings to prevent these pins from becoming output signal pins during a reset, or to prevent collision with signals outside the microcomputer.

Notes: 1. Mode pins and FWE pin input must satisfy the mode programming setup time (t_{MDS} = 200 ns) with respect to the reset release timing, as shown in figures 19.56 to 19.58.
2. For further information on FWE application and disconnection, see section 19.21, Flash Memory Programming and Erasing Precautions.
3. See section 9, I/O Ports.

19.22.8 Pin Configuration

The flash memory is controlled by means of the pins shown in tables 19.47.

Table 19.47 Flash Memory Pins

Pin Name	Abbreviation	I/O	Function
Reset	$\overline{\text{RES}}$	Input	Reset
Flash write enable	FWE	Input	Flash program/erase protection by hardware
Mode 2	MD2	Input	Sets MCU operating mode
Mode 1	MD1	Input	Sets MCU operating mode
Mode 0	MD0	Input	Sets MCU operating mode
Port 64	P64	Input	Sets MCU operating mode in PROM mode
Port 65	P65	Input	Sets MCU operating mode in PROM mode
Port 66	P66	Input	Sets MCU operating mode in PROM mode
Transmit data	TxD1	Output	Serial transmit data output
Receive data	RxD1	Input	Serial receive data input

FLER bit setting conditions are as follows:

- When flash memory is read during programming/erasing (including a vector read or instruction fetch)
- Immediately after exception handling (excluding a reset) during programming/erasing
- When a SLEEP instruction (including software standby) is executed during programming/erasing
- When a bus master other than the CPU (the DMAC or DTC) has control of the bus during programming/erasing

Error protection is released only by a reset and in hardware standby mode.

Figure 19.73 shows the flash memory state transition diagram.

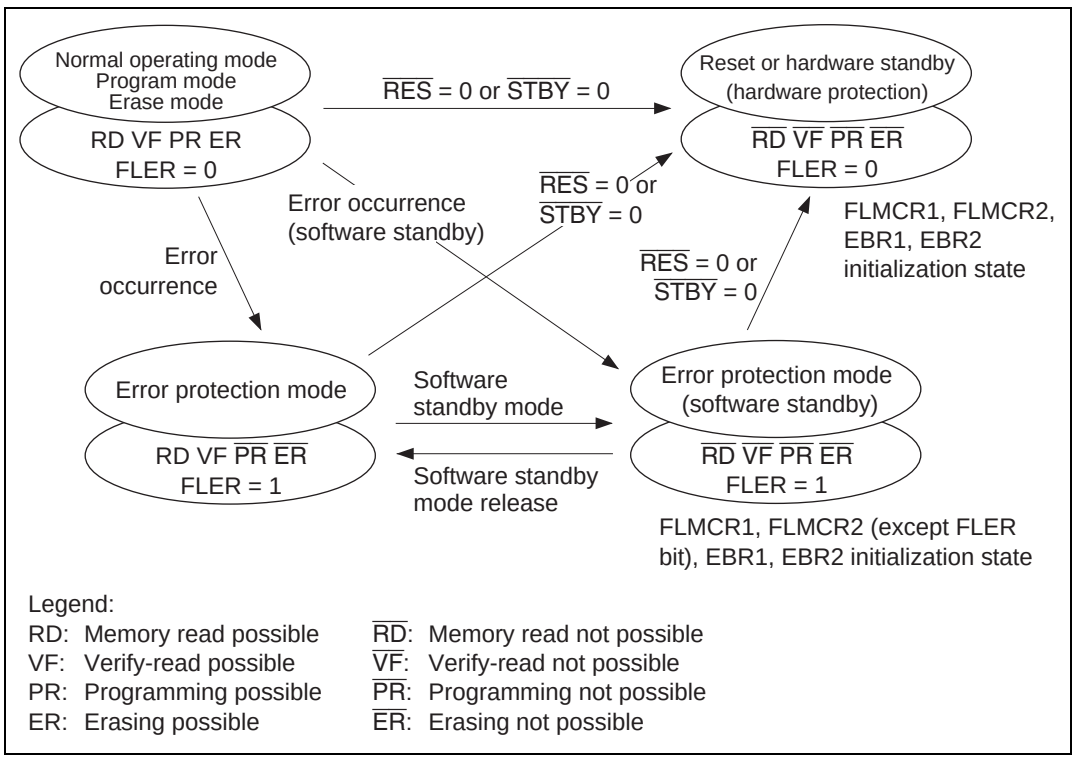


Figure 19.73 Flash Memory State Transitions

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
Bcc	BVS d:8	2					
	BPL d:8	2					
	BMI d:8	2					
	BGE d:8	2					
	BLT d:8	2					
	BGT d:8	2					
	BLE d:8	2					
	BRA d:16 (BT d:16)	2					1
	BRN d:16 (BF d:16)	2					1
	BHI d:16	2					1
	BLS d:16	2					1
	BCC d:16 (BHS d:16)	2					1
	BCS d:16 (BLO d:16)	2					1
	BNE d:16	2					1
	BEQ d:16	2					1
	BVC d:16	2					1
	BVS d:16	2					1
	BPL d:16	2					1
	BMI d:16	2					1
	BGE d:16	2					1
	BLT d:16	2					1
	BGT d:16	2					1
	BLE d:16	2					1
BCLR	BCLR #xx:3,Rd	1					
	BCLR #xx:3,@ERd	2			2		
	BCLR #xx:3,@aa:8	2			2		
	BCLR #xx:3,@aa:16	3			2		
	BCLR #xx:3,@aa:32	4			2		
	BCLR Rn,Rd	1					
	BCLR Rn,@ERd	2			2		
	BCLR Rn,@aa:8	2			2		
	BCLR Rn,@aa:16	3			2		
	BCLR Rn,@aa:32	4			2		

Instruction	1	2	3	4	5	6	7	8	9
STC EXR,@(d:16,ERd)	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA					
STC CCR,@(d:32,ERd)	R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	W:W EA			
STC EXR,@(d:32,ERd)	R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	W:W EA			
STC CCR,@–ERd	R:W 2nd	R:W NEXT	Internal operation, 1 state	W:W EA					
STC EXR,@–ERd	R:W 2nd	R:W NEXT	Internal operation, 1 state	W:W EA					
STC CCR,@aa:16	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA					
STC EXR,@aa:16	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA					
STC CCR,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	W:W EA				
STC EXR,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	W:W EA				
STM.L(ERn–ERn+1),@–SP	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M stack (H)*3	W:W stack (L)*3				
STM.L(ERn–ERn+2),@–SP	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M stack (H)*3	W:W stack (L)*3				
STM.L(ERn–ERn+3),@–SP	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M stack (H)*3	W:W stack (L)*3				
STMAC MACH,ERd	Cannot be used in the chip								
STMAC MACL,ERd									
SUB.B Rs,Rd	R:W NEXT								
SUB.W #xx:16,Rd	R:W 2nd	R:W NEXT							
SUB.W Rs,Rd	R:W NEXT								
SUB.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT						
SUB.L ERs,ERd	R:W NEXT								
SUBS #1/2/4,ERd	R:W NEXT								
SUBX #xx:8,Rd	R:W NEXT								
SUBX Rs,Rd	R:W NEXT								
TAS @ERd*8	R:W 2nd	R:W NEXT	R:B:M EA	W:B EA					
TRAPA #x:2 Advanced	R:W NEXT	Internal operation, 1 state	W:W stack (L)	W:W stack (H)	W:W stack (EXR)	R:W:M VEC	R:W VEC+2	Internal operation, 1 state	R:W*7
XOR.B #xx8,Rd	R:W NEXT								
XOR.B Rs,Rd	R:W NEXT								
XOR.W #xx:16,Rd	R:W 2nd	R:W NEXT							
XOR.W Rs,Rd	R:W NEXT								
XOR.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT						

PGDR—Port G Data Register H'FF6F Port G

Bit	7	6	5	4	3	2	1	0
	—	—	—	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR
Initial value	Undefined	Undefined	Undefined	0	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W	R/W

Stores output data for port G pins (PG_4 to PG_0)

PAPCR—Port A MOS Pull-Up Control Register H'FF70 Port A

Bit	7	6	5	4	3	2	1	0
	PA7PCR	PA6PCR	PA5PCR	PA4PCR	PA3PCR	PA2PCR	PA1PCR	PA0PCR
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

Controls the MOS input pull-up function incorporated into port A on a bit-by-bit basis

PBPCR—Port B MOS Pull-Up Control Register H'FF71 Port B

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
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Controls the MOS input pull-up function incorporated into port B on a bit-by-bit basis

PCPCR—Port C MOS Pull-Up Control Register H'FF72 **Port C**

Bit	7	6	5	4	3	2	1	0
	PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR
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

Controls the MOS input pull-up function incorporated into port C on a bit-by-bit basis

PDPCR—Port D MOS Pull-Up Control Register H'FF73 **Port D**

Bit	7	6	5	4	3	2	1	0
	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR
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

Controls the MOS input pull-up function incorporated into port D on a bit-by-bit basis

PEPCR—Port E MOS Pull-Up Control Register H'FF74 **Port E**

Bit	7	6	5	4	3	2	1	0
	PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR
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

Controls the MOS input pull-up function incorporated into port E on a bit-by-bit basis

C.8 Port B

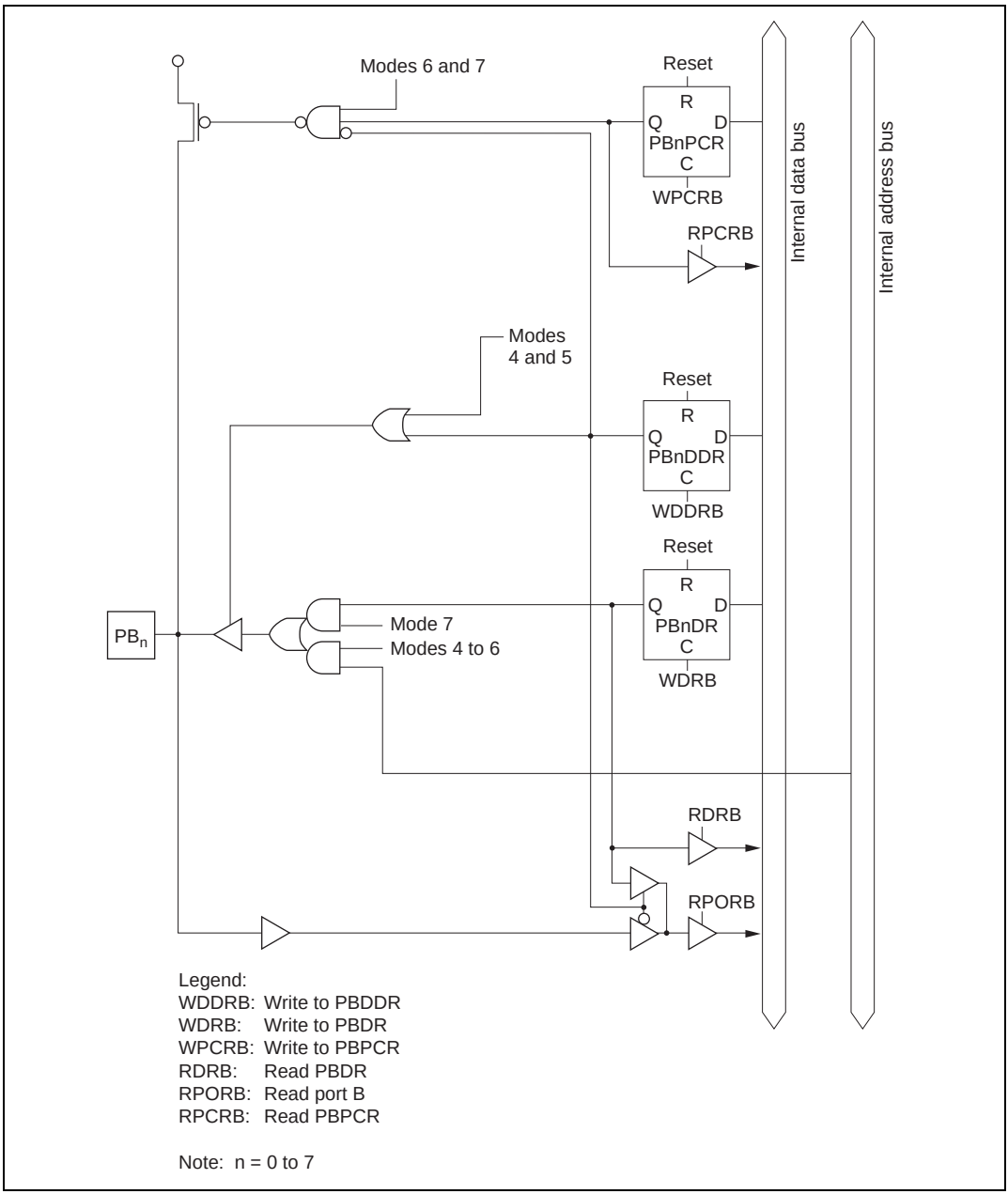


Figure C.8 Port B Block Diagram (Pins PB_0 to PB_7)

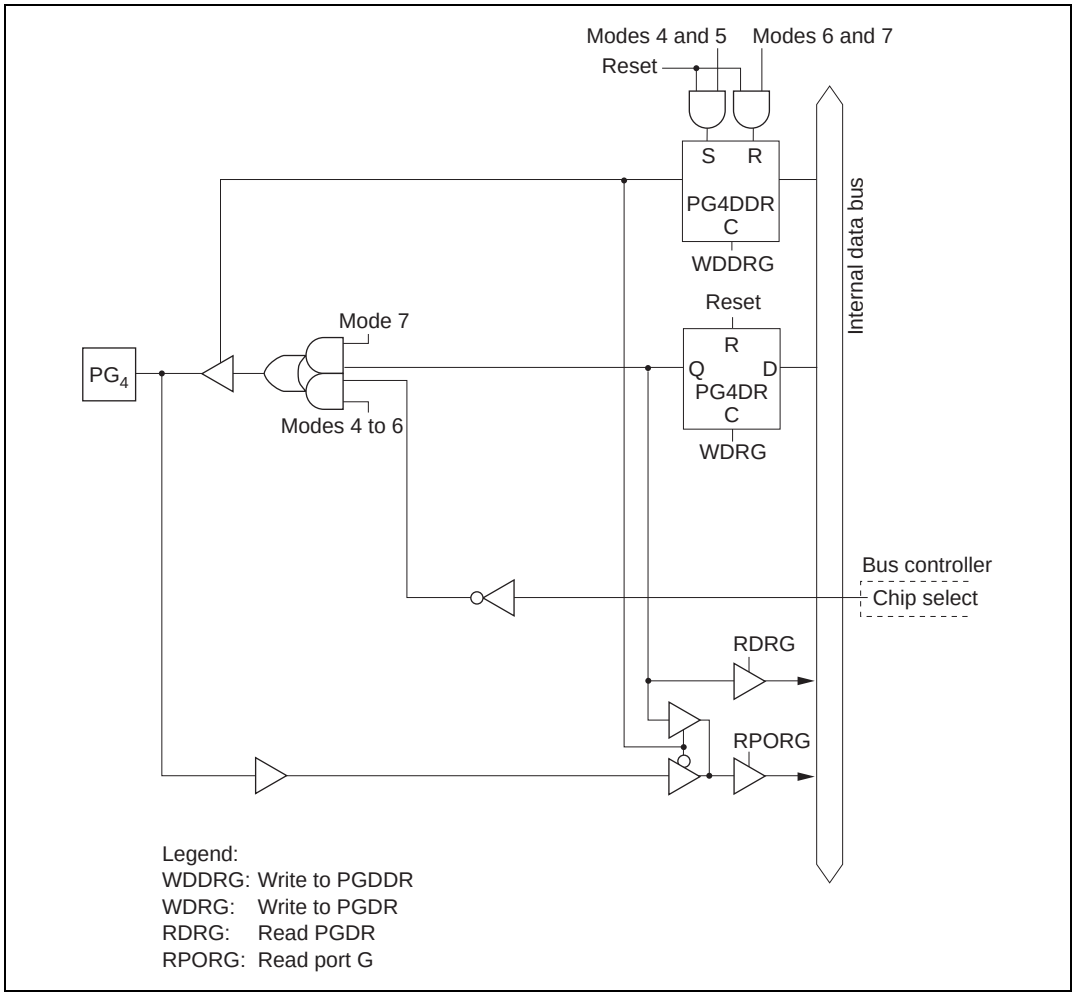


Figure C.13 (d) Port G Block Diagram (Pin PG_4)