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	Active
Core Processor	HCS12
Core Size	16-Bit
Speed	32MHz
Connectivity	CANbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	49
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	6K x 8
Voltage - Supply (Vcc/Vdd)	1.72V ~ 5.5V
Data Converters	A/D 10x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12p128j0mlh

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0004 Reserved	R	0	0	0	0	0	0	0	0
	W								
0x0005 Reserved	R	0	0	0	0	0	0	0	0
	W								
0x0006 Reserved	R	0	0	0	0	0	0	0	0
	W								
0x0007 Reserved	R	0	0	0	0	0	0	0	0
	W								
0x0008 PORTE	R	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
	W								
0x0009 DDRE	R	DDRE7	DDRE6	DDRE5	DDRE4	DDRE3	DDRE2	0	0
	W								
0x000A 0x000B Non-PIM Address Range	R	Non-PIM Address Range							
	W								
0x000C PUCR	R	0	BKPUE	0	PUPEE	0	0	PUPBE	PUPAE
	W								
0x000D RDRIV	R	0	0	0	RDPE	0	0	RDPB	RDPA
	W								
0x000E–0x001B Non-PIM Address Range	R	Non-PIM Address Range							
	W								
0x001C ECLKCTL	R	NECLK	NCLKX2	DIV16	EDIV4	EDIV3	EDIV2	EDIV1	EDIV0
	W								
0x001D Reserved	R	0	0	0	0	0	0	0	0
	W								
0x001E IRQCR	R	IRQE	IRQEN	0	0	0	0	0	0
	W								
0x001F Reserved	R	0	0	0	0	0	0	0	0
	W								

= Unimplemented or Reserved

2.3.12 ECLK Control Register (ECLKCTL)

Address 0x001C Access: User read/write(1)

	7	6	5	4	3	2	1	0
R W	NECLK	NCLKX2	DIV16	EDIV4	EDIV3	EDIV2	EDIV1	EDIV0
Reset:	Mode Depen-dent	1	0	0	0	0	0	0
Special single-chip	0	1	0	0	0	0	0	0
Normal single-chip	1	1	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 2-10. ECLK Control Register (ECLKCTL)

1. Read: Anytime
   Write: Anytime

Table 2-12. ECLKCTL Register Field Descriptions

Field	Description
7 NECLK	**No ECLK**—Disable ECLK output This bit controls the availability of a free-running clock on the ECLK pin. This clock has a fixed rate of equivalent to the internal bus clock. 1 ECLK disabled 0 ECLK enabled
6 NCLKX2	**No ECLKX2**—Disable ECLKX2 output This bit controls the availability of a free-running clock on the ECLKX2 pin. This clock has a fixed rate of twice the internal bus clock. 1 ECLKX2 disabled 0 ECLKX2 enabled
5 DIV16	**Free-running ECLK pre-divider**—Divide by 16 This bit enables a divide-by-16 stage on the selected EDIV rate. 1 Divider enabled: ECLK rate = EDIV rate divided by 16 0 Divider disabled: ECLK rate = EDIV rate
4-0 EDIV	**Free-running ECLK Divider**—Configure ECLK rate These bits determine the rate of the free-running clock on the ECLK pin. 00000 ECLK rate = bus clock rate 00001 ECLK rate = bus clock rate divided by 2 00010 ECLK rate = bus clock rate divided by 3,... 11111 ECLK rate = bus clock rate divided by 32

2.3.13 PIM Reserved Register

2.3.64 Port AD Pull Up Enable Register (PER1AD)

Address 0x0277 Access: User read/write(1)

	7	6	5	4	3	2	1	0
R W	PER1AD7	PER1AD6	PER1AD5	PER1AD4	PER1AD3	PER1AD2	PER1AD1	PER1AD0
Reset	0	0	0	0	0	0	0	0

Figure 2-62. Port AD Pull Up Enable Register (PER1AD)

1. Read: Anytime
   Write: Anytime

Table 2-58. PER1AD Register Field Descriptions

Field	Description
7-0 PER1AD	**Port AD pull-up enable**—Enable pull-up device on input pin This bit controls whether a pull device on the associated port input pin is active. If a pin is used as output this bit has no effect. 1 Pull device enabled 0 Pull device disabled

2.3.65 PIM Reserved Registers

Address 0x0278-0x27F Access: User read(1)

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved u = Unaffected by reset

Figure 2-63. PIM Reserved Registers

1. Read: Always reads 0x00
   Write: Unimplemented

2.4 Functional Description

2.4.1 General

Each pin except PE0, PE1, and BKGD can act as general purpose I/O. In addition each pin can act as an output or input of a peripheral module.

2.4.2 Registers

A set of configuration registers is common to all ports with exception of the ATD port (Table 2-59). All registers can be written at any time, however a specific configuration might not become active.

3.5.1 Chip Bus Control

The S12PMMC controls the address buses and the data buses that interface the bus masters (CPU12, S12SBDM) with the rest of the system (master buses). In addition the MMC handles all CPU read data bus swapping operations. All internal resources are connected to specific target buses (see Figure 3-12).

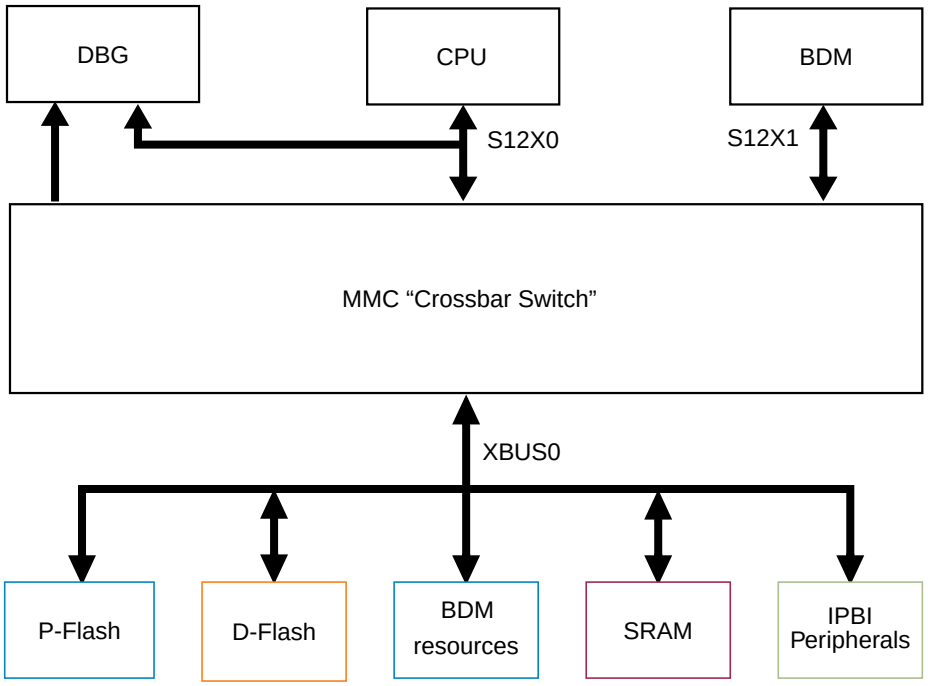

Figure 3-12. S12I platform

3.5.1.1 Master Bus Prioritization regarding Access Conflicts on Target Buses

The arbitration scheme allows only one master to be connected to a target at any given time. The following rules apply when prioritizing accesses from different masters to the same target bus:

- CPU12 always has priority over BDM.
- BDM has priority over CPU12 when its access is stalled for more than 128 cycles. In the later case the CPU will be stalled after finishing the current operation and the BDM will gain access to the bus.

3.5.2 Interrupts

The MMC does not generate any interrupts.

Table 7-5. CPMUCLKS Descriptions (continued)

Field	Description
3 PRE	**RTI Enable During Pseudo Stop Bit** — PRE enables the RTI during Pseudo Stop Mode. 0 RTI stops running during Pseudo Stop Mode. 1 RTI continues running during Pseudo Stop Mode if RTIOSCSEL=1. **Note:** If PRE=0 or RTIOSCSEL=0 then the RTI will go static while Stop Mode is active. The RTI counter will not be reset.
2 PCE	**COP Enable During Pseudo Stop Bit** — PCE enables the COP during Pseudo Stop Mode. 0 COP stops running during Pseudo Stop Mode 1 COP continues running during Pseudo Stop Mode if COPOSCSEL=1 **Note:** If PCE=0 or COPOSCSEL=0 then the COP will go static while Stop Mode is active. The COP counter will not be reset.
1 RTIOSCSEL	**RTI Clock Select**— RTIOSCSEL selects the clock source to the RTI. Either IRCCLK or OSCCLK. Changing the RTIOSCSEL bit re-starts the RTI time-out period. RTIOSCSEL can only be set to 1, if UPOSC=1. UPOSC= 0 clears the RTIOSCSEL bit. 0 RTI clock source is IRCCLK. 1 RTI clock source is OSCCLK.
0 COPOSCSEL	**COP Clock Select**— COPOSCSEL selects the clock source to the COP. Either IRCCLK or OSCCLK. Changing the COPOSCSEL bit re-starts the COP time-out period. COPOSCSEL can only be set to 1, if UPOSC=1. UPOSC= 0 clears the COPOSCSEL bit. 0 COP clock source is IRCCLK. 1 COP clock source is OSCCLK

7.3.2.12 S12CPMU COP Timer Arm/Reset Register (CPMUARMCOP)

This register is used to restart the COP time-out period.

0x003F

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset	0	0	0	0	0	0	0	0

Figure 7-15. S12CPMU CPMUARMCOP Register

Read: Always reads $00

Write: Anytime

When the COP is disabled (CR[2:0] = “000”) writing to this register has no effect.

When the COP is enabled by setting CR[2:0] nonzero, the following applies:

Writing any value other than $55 or $AA causes a COP reset. To restart the COP time-out period write $55 followed by a write of $AA. These writes do not need to occur back-to-back, but the sequence ($55, $AA) must be completed prior to COP end of time-out period to avoid a COP reset. Sequences of $55 writes are allowed. When the WCOP bit is set, $55 and $AA writes must be done in the last 25% of the selected time-out period; writing any value in the first 75% of the selected period will cause a COP reset.

7.3.2.13 High Temperature Control Register (CPMUHTCTL)

The CPMUHTCTL register configures the temperature sense features.

0x02F0

	7	6	5	4	3	2	1	0
R	0	0	VSEL	0	HTE	HTDS	HTIE	HTIF
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Read: Anytime

Write: VSEL, HTE, HTIE and HTIF are write anytime, HTDS is read only

7.3.2.19 High Temperature Trimming Register (CPMUHTTR)

The CPMUHTTR register configures the trimming of the S12CPMU temperature sense.

0x02F7

	7	6	5	4	3	2	1	0
R	HTOE	0	0	0	HTTR3	HTTR2	HTTR1	HTTR0
W								
Reset	0	0	0	0	F	F	F	F

After de-assert of System Reset a trim value is automatically loaded from the Flash memory. See Device specification for details.

= Unimplemented or Reserved

Read: Anytime

Write: Anytime

Field	Description
7 HTOE	**High Temperature Offset Enable Bit** — If set the temperature sense offset is enabled. 0 The temperature sense offset is disabled. HTTR[3:0] bits don't care. 1 The temperature sense offset is enabled. HTTR[3:0] select the temperature offset.
3–0 HTTR[3:0]	**High Temperature Trimming Bits** — See Table 1-27 for trimming effects.

Bit	Trimming Effect
HTTR[3]	Increases V_{HT} twice of HTTR[2]
HTTR[2]	Increases V_{HT} twice of HTTR[1]
HTTR[1]	Increases V_{HT} twice of HTTR[0]
HTTR[0]	Increases V_{HT} (to compensate Temperature Offset)

8.2.2 TXCAN — CAN Transmitter Output Pin

TXCAN is the MSCAN transmitter output pin. The TXCAN output pin represents the logic level on the CAN bus:

0 = Dominant state
1 = Recessive state

8.2.3 CAN System

A typical CAN system with MSCAN is shown in Figure 8-2. Each CAN station is connected physically to the CAN bus lines through a transceiver device. The transceiver is capable of driving the large current needed for the CAN bus and has current protection against defective CAN or defective stations.

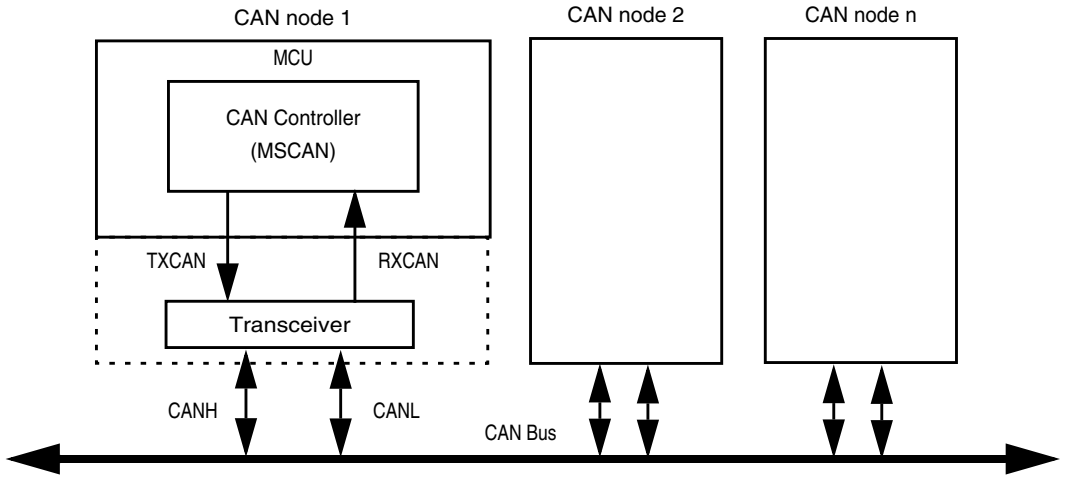

Figure 8-2. CAN System

8.3 Memory Map and Register Definition

This section provides a detailed description of all registers accessible in the MSCAN.

8.3.1 Module Memory Map

Figure 8-3 gives an overview on all registers and their individual bits in the MSCAN memory map. The *register address* results from the addition of *base address* and *address offset*. The *base address* is determined at the MCU level and can be found in the MCU memory map description. The *address offset* is defined at the module level.

The MSCAN occupies 64 bytes in the memory space. The base address of the MSCAN module is determined at the MCU level when the MCU is defined. The register decode map is fixed and begins at the first address of the module address offset.

Module Base + 0x0000 — Access: User read/write(1)

	7	6	5	4	3	2	1	0
R	RXFRM	RXACT	CSWAI	SYNCH	TIME	WUPE	SLPRQ	INITRQ
W								
Reset:	0	0	0	0	0	0	0	1

= Unimplemented

Figure 8-4. MSCAN Control Register 0 (CANCTL0)

1. Read: Anytime
   Write: Anytime when out of initialization mode; exceptions are read-only RXACT and SYNCH, RXFRM (which is set by the module only), and INITRQ (which is also writable in initialization mode)

NOTE

The CANCTL0 register, except WUPE, INITRQ, and SLPRQ, is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK = 1). This register is writable again as soon as the initialization mode is exited (INITRQ = 0 and INITAK = 0).

Table 8-3. CANCTL0 Register Field Descriptions

Field	Description
7 RXFRM(1)	**Received Frame Flag** — This bit is read and clear only. It is set when a receiver has received a valid message correctly, independently of the filter configuration. After it is set, it remains set until cleared by software or reset. Clearing is done by writing a 1. Writing a 0 is ignored. This bit is not valid in loopback mode. 0 No valid message was received since last clearing this flag 1 A valid message was received since last clearing of this flag
6 RXACT	**Receiver Active Status** — This read-only flag indicates the MSCAN is receiving a message. The flag is controlled by the receiver front end. This bit is not valid in loopback mode. 0 MSCAN is transmitting or idle[2] 1 MSCAN is receiving a message (including when arbitration is lost)(2)
5 CSWAI(3)	**CAN Stops in Wait Mode** — Enabling this bit allows for lower power consumption in wait mode by disabling all the clocks at the CPU bus interface to the MSCAN module. 0 The module is not affected during wait mode 1 The module ceases to be clocked during wait mode
4 SYNCH	**Synchronized Status** — This read-only flag indicates whether the MSCAN is synchronized to the CAN bus and able to participate in the communication process. It is set and cleared by the MSCAN. 0 MSCAN is not synchronized to the CAN bus 1 MSCAN is synchronized to the CAN bus
3 TIME	**Timer Enable** — This bit activates an internal 16-bit wide free running timer which is clocked by the bit clock rate. If the timer is enabled, a 16-bit time stamp will be assigned to each transmitted/received message within the active TX/RX buffer. Right after the EOF of a valid message on the CAN bus, the time stamp is written to the highest bytes (0x000E, 0x000F) in the appropriate buffer (see Section 8.3.3, "Programmer's Model of Message Storage"). The internal timer is reset (all bits set to 0) when disabled. This bit is held low in initialization mode. 0 Disable internal MSCAN timer 1 Enable internal MSCAN timer

- For transmit buffers, anytime when TXEx flag is set (see Section 8.3.2.7, "MSCAN Transmitter Flag Register (CANTFLG)") and the corresponding transmit buffer is selected in CANTBSEL (see Section 8.3.2.11, "MSCAN Transmit Buffer Selection Register (CANTBSEL)").
- Unimplemented for receive buffers.

Reset: Undefined because of RAM-based implementation

Figure 8-25. Receive/Transmit Message Buffer — Standard Identifier Mapping

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
IDR0 0x00X0	R	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3
	W								
IDR1 0x00X1	R	ID2	ID1	ID0	RTR	IDE (=0)			
	W								
IDR2 0x00X2	R								
	W								
IDR3 0x00X3	R								
	W								

= Unused, always read 'x'

8.3.3.1 Identifier Registers (IDR0–IDR3)

The identifier registers for an extended format identifier consist of a total of 32 bits; ID[28:0], SRR, IDE, and RTR bits. The identifier registers for a standard format identifier consist of a total of 13 bits; ID[10:0], RTR, and IDE bits.

Module Base + 0x000C

	7	6	5	4	3	2	1	0
R	Bit 7	6	5	4	3	2	1	Bit 0
W	0	0	0	0	0	0	0	0
Reset	0	0	0	0	0	0	0	0

Figure 10-15. PWM Channel Counter Registers (PWMCNT0)

Module Base + 0x000D

	7	6	5	4	3	2	1	0
R	Bit 7	6	5	4	3	2	1	Bit 0
W	0	0	0	0	0	0	0	0
Reset	0	0	0	0	0	0	0	0

Figure 10-16. PWM Channel Counter Registers (PWMCNT1)

Module Base + 0x000E

	7	6	5	4	3	2	1	0
R	Bit 7	6	5	4	3	2	1	Bit 0
W	0	0	0	0	0	0	0	0
Reset	0	0	0	0	0	0	0	0

Figure 10-17. PWM Channel Counter Registers (PWMCNT2)

Module Base + 0x000F

	7	6	5	4	3	2	1	0
R	Bit 7	6	5	4	3	2	1	Bit 0
W	0	0	0	0	0	0	0	0
Reset	0	0	0	0	0	0	0	0

Figure 10-18. PWM Channel Counter Registers (PWMCNT3)

Module Base + 0x00010

	7	6	5	4	3	2	1	0
R	Bit 7	6	5	4	3	2	1	Bit 0
W	0	0	0	0	0	0	0	0
Reset	0	0	0	0	0	0	0	0

Figure 10-19. PWM Channel Counter Registers (PWMCNT4)

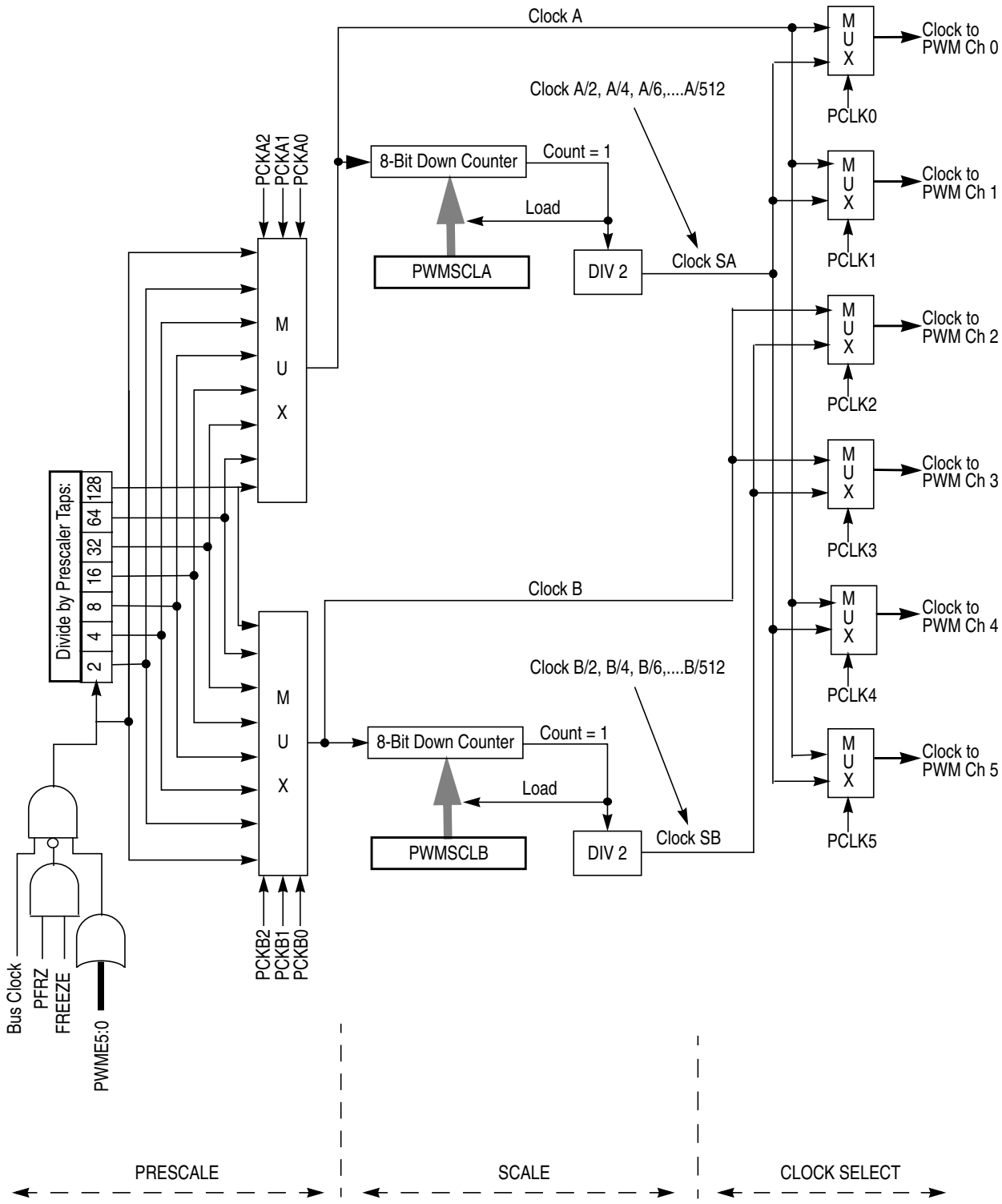

Figure 10-34. PWM Clock Select Block Diagram

NOTE

Changing clock control bits while channels are operating can cause irregularities in the PWM outputs.

10.4.2 PWM Channel Timers

The main part of the PWM module are the actual timers. Each of the timer channels has a counter, a period register and a duty register (each are 8 bit). The waveform output period is controlled by a match between the period register and the value in the counter. The duty is controlled by a match between the duty register and the counter value and causes the state of the output to change during the period. The starting polarity of the output is also selectable on a per channel basis. Figure 10-35 shows a block diagram for PWM timer.

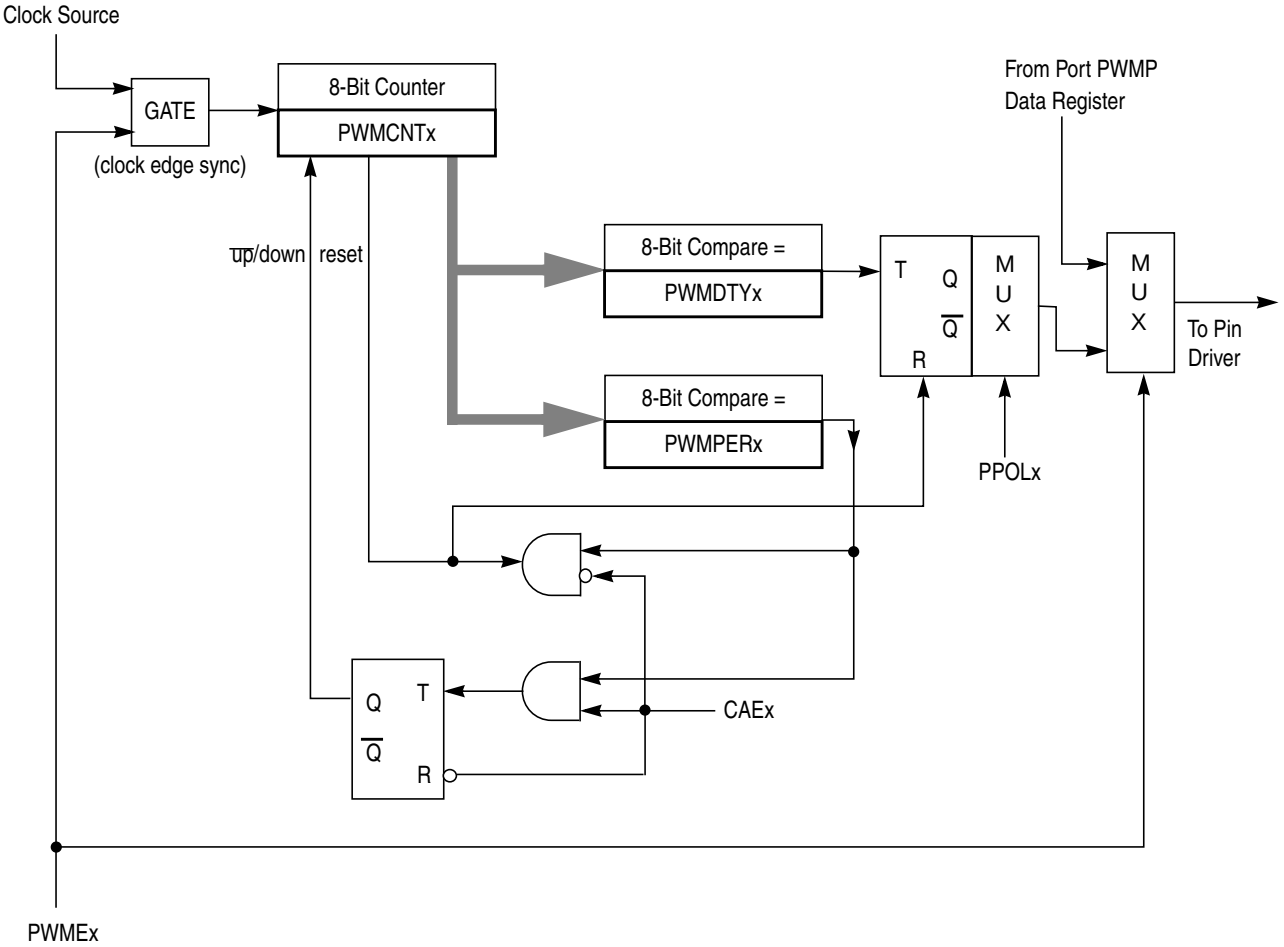

Figure 10-35. PWM Timer Channel Block Diagram

10.4.2.1 PWM Enable

Each PWM channel has an enable bit (PWMEx) to start its waveform output. When any of the PWMEx bits are set (PWMEx = 1), the associated PWM output signal is enabled immediately. However, the actual PWM waveform is not available on the associated PWM output until its clock source begins its next cycle

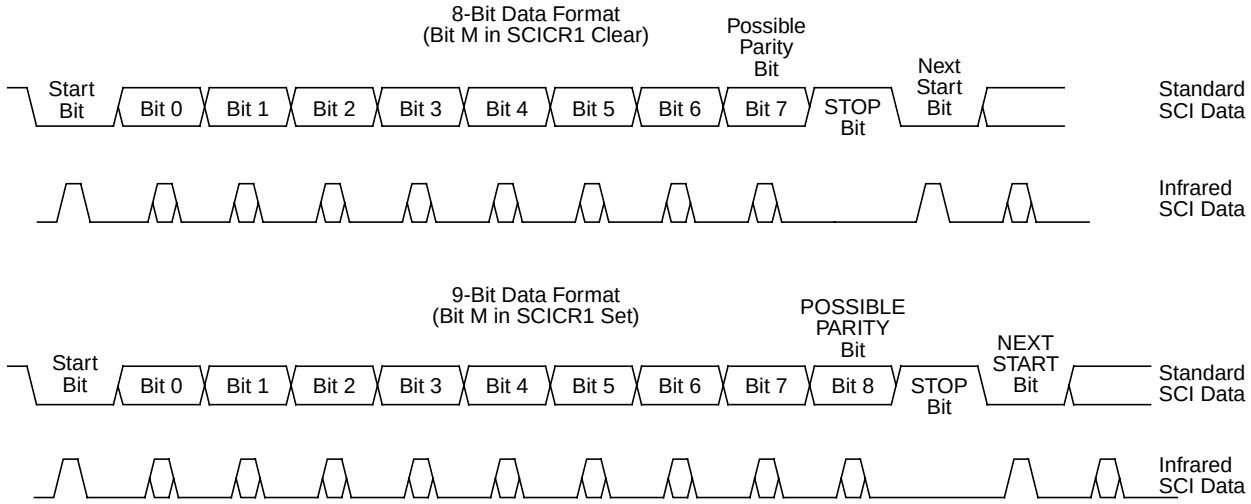

Figure 11-15. SCI Data Formats

Each data character is contained in a frame that includes a start bit, eight or nine data bits, and a stop bit. Clearing the M bit in SCI control register 1 configures the SCI for 8-bit data characters. A frame with eight data bits has a total of 10 bits. Setting the M bit configures the SCI for nine-bit data characters. A frame with nine data bits has a total of 11 bits.

Table 11-14. Example of 8-Bit Data Formats

Start Bit	Data Bits	Address Bits	Parity Bits	Stop Bit
1	8	0	0	1
1	7	0	1	1
1	7	1[1]	0	1

1. The address bit identifies the frame as an address character. See Section 11.4.6.6, “Receiver Wakeup”.

When the SCI is configured for 9-bit data characters, the ninth data bit is the T8 bit in SCI data register high (SCIDRH). It remains unchanged after transmission and can be used repeatedly without rewriting it. A frame with nine data bits has a total of 11 bits.

Table 11-15. Example of 9-Bit Data Formats

Start Bit	Data Bits	Address Bits	Parity Bits	Stop Bit
1	9	0	0	1
1	8	0	1	1
1	8	1[1]	0	1

1. The address bit identifies the frame as an address character. See Section 11.4.6.6, “Receiver Wakeup”.

The main element of the SPI system is the SPI data register. The n-bit[1] data register in the master and the n-bit[1] data register in the slave are linked by the MOSI and MISO pins to form a distributed 2n-bit[1] register. When a data transfer operation is performed, this 2n-bit[1] register is serially shifted n[1] bit positions by the S-clock from the master, so data is exchanged between the master and the slave. Data written to the master SPI data register becomes the output data for the slave, and data read from the master SPI data register after a transfer operation is the input data from the slave.

A read of SPISR with SPTEF = 1 followed by a write to SPIDR puts data into the transmit data register. When a transfer is complete and SPIF is cleared, received data is moved into the receive data register. This data register acts as the SPI receive data register for reads and as the SPI transmit data register for writes. A common SPI data register address is shared for reading data from the read data buffer and for writing data to the transmit data register.

The clock phase control bit (CPHA) and a clock polarity control bit (CPOL) in the SPI control register 1 (SPICR1) select one of four possible clock formats to be used by the SPI system. The CPOL bit simply selects a non-inverted or inverted clock. The CPHA bit is used to accommodate two fundamentally different protocols by sampling data on odd numbered SCK edges or on even numbered SCK edges (see Section 12.4.3, “Transmission Formats”).

The SPI can be configured to operate as a master or as a slave. When the MSTR bit in SPI control register1 is set, master mode is selected, when the MSTR bit is clear, slave mode is selected.

NOTE

A change of CPOL or MSTR bit while there is a received byte pending in the receive shift register will destroy the received byte and must be avoided.

12.4.1 Master Mode

The SPI operates in master mode when the MSTR bit is set. Only a master SPI module can initiate transmissions. A transmission begins by writing to the master SPI data register. If the shift register is empty, data immediately transfers to the shift register. Data begins shifting out on the MOSI pin under the control of the serial clock.

- Serial clock

  The SPR2, SPR1, and SPR0 baud rate selection bits, in conjunction with the SPPR2, SPPR1, and SPPR0 baud rate preselection bits in the SPI baud rate register, control the baud rate generator and determine the speed of the transmission. The SCK pin is the SPI clock output. Through the SCK pin, the baud rate generator of the master controls the shift register of the slave peripheral.
- MOSI, MISO pin

  In master mode, the function of the serial data output pin (MOSI) and the serial data input pin (MISO) is determined by the SPC0 and BIDIROE control bits.
- $\overline{SS}$ pin

  If MODFEN and SSOE are set, the $\overline{SS}$ pin is configured as slave select output. The $\overline{SS}$ output becomes low during each transmission and is high when the SPI is in idle state.

  If MODFEN is set and SSOE is cleared, the $\overline{SS}$ pin is configured as input for detecting mode fault error. If the $\overline{SS}$ input becomes low this indicates a mode fault error where another master tries to

1. n depends on the selected transfer width, please refer to Section 12.3.2.2, “SPI Control Register 2 (SPICR2)

12.4.7.5.2 SPIF

SPIF occurs when new data has been received and copied to the SPI data register. After SPIF is set, it does not clear until it is serviced. SPIF has an automatic clearing process, which is described in Section 12.3.2.4, “SPI Status Register (SPISR)”.

12.4.7.5.3 SPTEF

SPTEF occurs when the SPI data register is ready to accept new data. After SPTEF is set, it does not clear until it is serviced. SPTEF has an automatic clearing process, which is described in Section 12.3.2.4, “SPI Status Register (SPISR)”.

Address & Name		7	6	5	4	3	2	1	0
0x000A FCCOBHI	R	CCOB15	CCOB14	CCOB13	CCOB12	CCOB11	CCOB10	CCOB9	CCOB8
	W								
0x000B FCCOBLO	R	CCOB7	CCOB6	CCOB5	CCOB4	CCOB3	CCOB2	CCOB1	CCOB0
	W								
0x000C FRSV1	R	0	0	0	0	0	0	0	0
	W								
0x000D FRSV2	R	0	0	0	0	0	0	0	0
	W								
0x000E FRSV3	R	0	0	0	0	0	0	0	0
	W								
0x000F FRSV4	R	0	0	0	0	0	0	0	0
	W								
0x0010 FOPT	R	NV7	NV6	NV5	NV4	NV3	NV2	NV1	NV0
	W								
0x0011 FRSV5	R	0	0	0	0	0	0	0	0
	W								
0x0012 FRSV6	R	0	0	0	0	0	0	0	0
	W								
0x0013 FRSV7	R	0	0	0	0	0	0	0	0
	W								

= Unimplemented or Reserved

Figure 13-4. FTMRC128K1 Register Summary (continued)

13.3.2.1 Flash Clock Divider Register (FCLKDIV)

The FCLKDIV register is used to control timed events in program and erase algorithms.

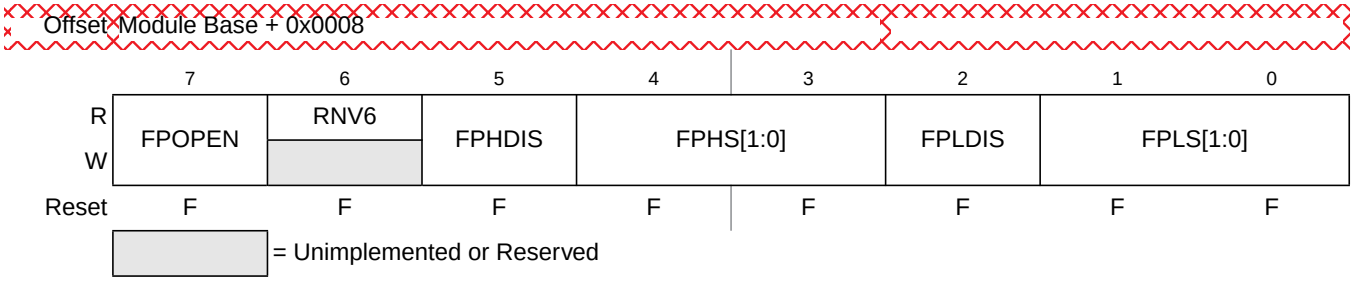

Figure 13-13. Flash Protection Register (FPROT)

The (unreserved) bits of the FPROT register are writable with the restriction that the size of the protected region can only be increased (see Section 13.3.2.9.1, “P-Flash Protection Restrictions,” and Table 13-20).

During the reset sequence, the FPROT register is loaded with the contents of the P-Flash protection byte in the Flash configuration field at global address 0x3_FF0C located in P-Flash memory (see Table 13-3) as indicated by reset condition ‘F’ in Figure 13-13. To change the P-Flash protection that will be loaded during the reset sequence, the upper sector of the P-Flash memory must be unprotected, then the P-Flash protection byte must be reprogrammed. If a double bit fault is detected while reading the P-Flash phrase containing the P-Flash protection byte during the reset sequence, the FPOPEN bit will be cleared and remaining bits in the FPROT register will be set to leave the P-Flash memory fully protected.

Trying to alter data in any protected area in the P-Flash memory will result in a protection violation error and the FPVIOL bit will be set in the FSTAT register. The block erase of a P-Flash block is not possible if any of the P-Flash sectors contained in the same P-Flash block are protected.

Table 13-16. FPROT Field Descriptions

Field	Description
7 FPOPEN	**Flash Protection Operation Enable** — The FPOPEN bit determines the protection function for program or erase operations as shown in Table 13-17 for the P-Flash block. 0 When FPOPEN is clear, the FPHDIS and FPLDIS bits define unprotected address ranges as specified by the corresponding FPHS and FPLS bits 1 When FPOPEN is set, the FPHDIS and FPLDIS bits enable protection for the address range specified by the corresponding FPHS and FPLS bits
6 RNV[6]	**Reserved Nonvolatile Bit** — The RNV bit should remain in the erased state for future enhancements.
5 FPHDIS	**Flash Protection Higher Address Range Disable** — The FPHDIS bit determines whether there is a protected/unprotected area in a specific region of the P-Flash memory ending with global address 0x3_FFFF. 0 Protection/Unprotection enabled 1 Protection/Unprotection disabled
4–3 FPHS[1:0]	**Flash Protection Higher Address Size** — The FPHS bits determine the size of the protected/unprotected area in P-Flash memory as shown inTable 13-18. The FPHS bits can only be written to while the FPHDIS bit is set.
2 FPLDIS	**Flash Protection Lower Address Range Disable** — The FPLDIS bit determines whether there is a protected/unprotected area in a specific region of the P-Flash memory beginning with global address 0x3_8000. 0 Protection/Unprotection enabled 1 Protection/Unprotection disabled
1–0 FPLS[1:0]	**Flash Protection Lower Address Size** — The FPLS bits determine the size of the protected/unprotected area in P-Flash memory as shown in Table 13-19. The FPLS bits can only be written to while the FPLDIS bit is set.

13.3.2.9.1 P-Flash Protection Restrictions

The general guideline is that P-Flash protection can only be added and not removed. Table 13-20 specifies all valid transitions between P-Flash protection scenarios. Any attempt to write an invalid scenario to the FPROT register will be ignored. The contents of the FPROT register reflect the active protection scenario. See the FPHS and FPLS bit descriptions for additional restrictions.

Table 13-20. P-Flash Protection Scenario Transitions

From Protection Scenario	To Protection Scenario[1]							
	0	1	2	3	4	5	6	7
0	X	X	X	X				
1		X		X				
2			X	X				
3				X				
4				X	X			
5			X	X	X	X		
6		X		X	X		X	
7	X	X	X	X	X	X	X	X

1. Allowed transitions marked with X, see Figure 13-14 for a definition of the scenarios.

13.3.2.10 D-Flash Protection Register (DFPROT)

The DFPROT register defines which D-Flash sectors are protected against program and erase operations.

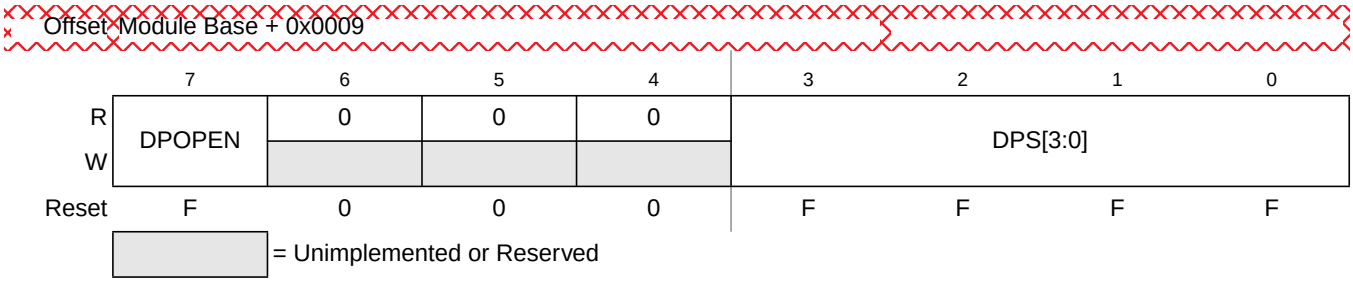

Figure 13-15. D-Flash Protection Register (DFPROT)

The (unreserved) bits of the DFPROT register are writable with the restriction that protection can be added but not removed. Writes must increase the DPS value and the DPOPEN bit can only be written from 1 (protection disabled) to 0 (protection enabled). If the DPOPEN bit is set, the state of the DPS bits is irrelevant.

During the reset sequence, the DFPROT register is loaded with the contents of the D-Flash protection byte in the Flash configuration field at global address 0x3_FF0D located in P-Flash memory (see Table 13-3) as indicated by reset condition F in Figure 13-15. To change the D-Flash protection that will be loaded during the reset sequence, the P-Flash sector containing the D-Flash protection byte must be unprotected, then the D-Flash protection byte must be programmed. If a double bit fault is detected while reading the

Table A-10. Peripheral Configurations for Run & Wait Current Measurement

Peripheral	Configuration
MSCAN	configured to loop-back mode using a bit rate of 1Mbit/s
SPI	configured to master mode, continously transmit data (0x55 or 0xAA) at 1Mbit/s
SCI	configured into loop mode, continously transmit data (0x55) at speed of 57600 baud
PWM	configured to toggle its pins at the rate of 40kHz
ATD	the peripheral is configured to operate at its maximum specified frequency and to continuously convert voltages on all input channels in sequence.
DBG	the module is enabled and the comparators are configured to trigger in outside range.The range covers all the code executed by the core.
TIM	the peripheral shall be configured to output compare mode, pulse accumulator and modulus counter enabled.
COP & RTI	enabled

Table A-11. Run and Wait Current Characteristics

Conditions are: V_{DDR}=5.5V, T_A=125°C, see Table A-9. and Table A-10.							
Num	**C**	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	P	IDD Run Current	I_{DDR}		18	20	mA
2	P	IDD Wait Current	I_{DDW}		11	12	mA

Table A-12. Full Stop Current Characteristics

Conditions are: VDDR=5.5V, API see Table A-9.							
Num	**C**	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
Stop Current API disabled							
1	P	150°C	I_{DDS}		250	1100	µA
2	P	-40°C	I_{DDS}		15	35	µA
3	P	25°C,	I_{DDS}		25	50	µA
Stop Current API enabled							
4	C	150°C,	I_{DDS}		270		µA
5	C	-40°C	I_{DDS}		20		µA
6	C	25°C	I_{DDS}		40		µA