



Welcome to [E-XFL.COM](http://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	HCS12
Core Size	16-Bit
Speed	25MHz
Connectivity	CANbus, EBI/EMI, I <sup>2</sup> C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	91
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	12K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	112-LQFP
Supplier Device Package	112-LQFP (20x20)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12kt256mpve">https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12kt256mpve</a>

## Chapter 1

### MC9S12KT256 Device Overview (MC9S12KT256V1)

1.1	Introduction	17
1.1.1	Features	17
1.1.2	Modes of Operation	19
1.1.3	MC9S12KT256 Block Diagram	21
1.2	Signal Description	22
1.2.1	Signal Properties Summary	24
1.2.2	Detailed Signal Descriptions	27
1.2.3	Power Supply Pins	36
1.3	Memory Map and Register Definition	38
1.3.1	Device Memory Map	38
1.3.2	Detailed Register Map	40
1.3.3	Part ID Assignments	64
1.4	System Clock Description	65
1.5	Modes of Operation	65
1.5.1	Chip Configuration Summary	65
1.5.2	Security	67
1.5.3	Low Power Modes	68
1.6	Resets and Interrupts	68
1.6.1	Vectors	68
1.6.2	Resets	71

## Chapter 2

### 256 Kbyte ECC Flash Module (S12FTS256K2ECCV1)

2.1	Introduction	73
2.1.1	Glossary	74
2.1.2	Features	74
2.1.3	Modes of Operation	74
2.1.4	Block Diagram	74
2.2	External Signal Description	75
2.3	Memory Map and Register Definition	76
2.3.1	Module Memory Map	76
2.3.2	Register Descriptions	80
2.4	Functional Description	95
2.4.1	Flash Command Operations	95
2.5	Operating Modes	111
2.5.1	Wait Mode	111
2.5.2	Stop Mode	111
2.5.3	Background Debug Mode	111
2.6	Flash Module Security	112
2.6.1	Unsecuring the MCU using Backdoor Key Access	112
2.6.2	Unsecuring the Flash Module in Special Single-Chip Mode using BDM	113
2.7	Resets	113

## 0x0140–0x017F CAN0 (MSCAN)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0150–0x0153	CAN0IDAR0–CAN0IDAR3	R W	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
0x0154–0x0157	CAN0IDMR0–CAN0IDMR3	R W	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
0x0158–0x015B	CAN0IDAR4–CAN0IDAR7	R W	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
0x015C–0x015F	CAN0IDMR4–CAN0IDMR7	R W	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
0x0160–0x016F	CAN0RXFG	R W	FOREGROUND RECEIVE BUFFER (see Table 1-6)							
0x0170–0x017F	CAN0TXFG	R W	FOREGROUND TRANSMIT BUFFER (see Table 1-6)							

**Table 1-6. Detailed MSCAN Foreground Receive and Transmit Buffer Layout**

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00	Extended ID	R	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21
	Standard ID	R	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3
	CANxRIDR0	W								
0x01	Extended ID	R	ID20	ID19	ID18	SRR=1	IDE=1	ID17	ID16	ID15
	Standard ID	R	ID2	ID1	ID0	RTR	IDE=0			
	CANxRIDR1	W								
0x02	Extended ID	R	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
	Standard ID	R								
	CANxRIDR2	W								
0x03	Extended ID	R	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR
	Standard ID	R								
	CANxRIDR3	W								
0x04–0x0B	CANxRDSR0–CANxRDSR7	R W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0x0C	CANRxDLR	R W					DLC3	DLC2	DLC1	DLC0
0x0D	Reserved	R W								
0x0E	CANxRTSRH	R W	TSR15	TSR14	TSR13	TSR12	TSR11	TSR10	TSR9	TSR8
0x0F	CANxRTSRL	R W	TSR7	TSR6	TSR5	TSR4	TSR3	TSR2	TSR1	TSR0
0x10	Extended ID	R	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21
	CANxTIDR0	W								
	Standard ID	R W	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3
0x10	Extended ID	R	ID20	ID19	ID18	SRR=1	IDE=1	ID17	ID16	ID15
	CANxTIDR1	W								
	Standard ID	R W	ID2	ID1	ID0	RTR	IDE=0			

If we define:

- FCLK as the clock of the Flash timing control block,
- Tbus as the period of the bus clock, and
- INT(x) as taking the integer part of x (e.g. INT(4.323)=4).

Then, FCLKDIV register bits PRDIV8 and FDIV[5:0] are to be set as described in Figure 3-23.

For example, if the oscillator clock frequency is 950 kHz and the bus clock frequency is 10 MHz, FCLKDIV bits FDIV[5:0] must be set to 4 (000100) and bit PRDIV8 set to 0. The resulting FCLK frequency is then 190 kHz. As a result, the Flash program and erase algorithm timings are increased over the optimum target by:

$$(200 - 190)/200 \times 100 = 5\%$$

Program and erase command execution time will increase proportionally with the period of FCLK. Because of the impact of clock synchronization on the accuracy of the functional timings, programming or erasing the Flash memory cannot be performed if the bus clock runs at less than 1 MHz. Programming or erasing the Flash memory with  $FCLK < 150$  kHz must be avoided. Setting FCLKDIV to a value such that  $FCLK < 150$  kHz can destroy the Flash memory due to overstress. Setting FCLKDIV to a value such that  $(1/FCLK + T_{bus}) < 5\mu s$  can result in incomplete programming or erasure of the Flash memory cells.

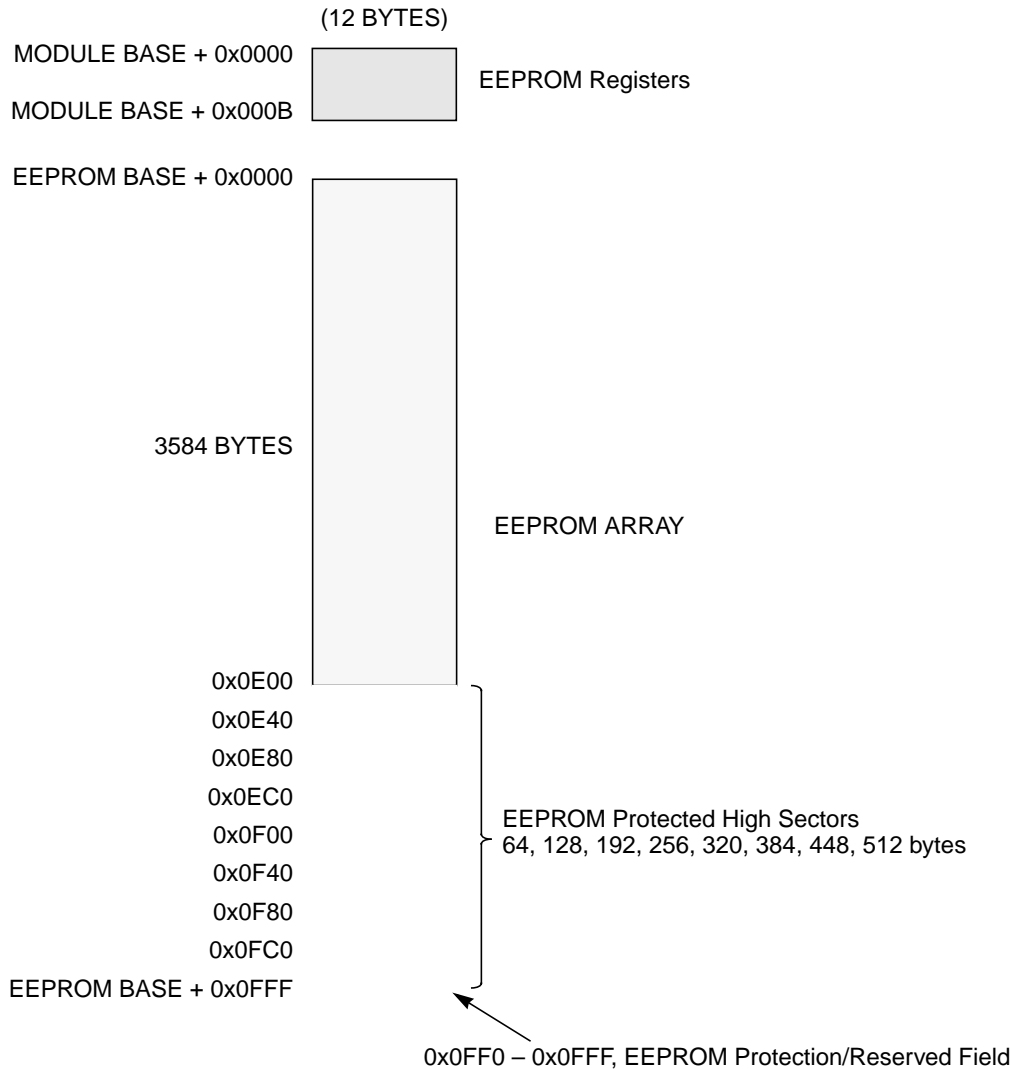
If the FCLKDIV register is written, the FDIVLD bit is set automatically. If the FDIVLD bit is 0, the FCLKDIV register has not been written since the last reset. Flash commands will not be executed if the FCLKDIV register has not been written to.

**Table 4-1. EEPROM Protection/Reserved Field**

Address Offset	Size (Bytes)	Description
0x0FF0 – 0x0FFC	13	Reserved
0x0FFD	1	EEPROM protection byte
0x0FFE – 0x0FFF	2	Reserved

The EEPROM module has hardware interlocks which protect data from accidental corruption. A protected sector is located at the higher address end of the EEPROM array, just below address 0x0FFF. The protected sector in the EEPROM array can be sized from 64 bytes to 512 bytes. In addition, the EPOPEN bit in the EPROT register, described in Section 4.3.2.5, “EEPROM Protection Register (EPROT)”, can be set to globally protect the entire EEPROM array.

Chip security is defined at the MCU level.



**Figure 4-2. EEPROM Memory Map**

### 5.3.5.7 Port H Interrupt Enable Register (PIEH)

Module Base + 0x0026

	7	6	5	4	3	2	1	0
R	PIEH7	PIEH6	PIEH5	PIEH4	PIEH3	PIEH2	PIEH1	PIEH0
W								
Reset	0	0	0	0	0	0	0	0

Figure 5-37. Port H Interrupt Enable Register (PIEH)

Read: Anytime. Write: Anytime.

This register disables or enables on a per pin basis the edge sensitive external interrupt associated with port H.

Table 5-34. PIEH Field Descriptions

Field	Description
7–0 PIEH[7:0]	<b>Interrupt Enable Port H</b> 0 Interrupt is disabled (interrupt flag masked). 1 Interrupt is enabled.

### 5.3.5.8 Port H Interrupt Flag Register (PIFH)

Module Base + 0x0027

	7	6	5	4	3	2	1	0
R	PIFH7	PIFH6	PIFH5	PIFH4	PIFH3	PIFH2	PIFH1	PIFH0
W								
Reset	0	0	0	0	0	0	0	0

Figure 5-38. Port H Interrupt Flag Register (PIFH)

Read: Anytime. Write: Anytime.

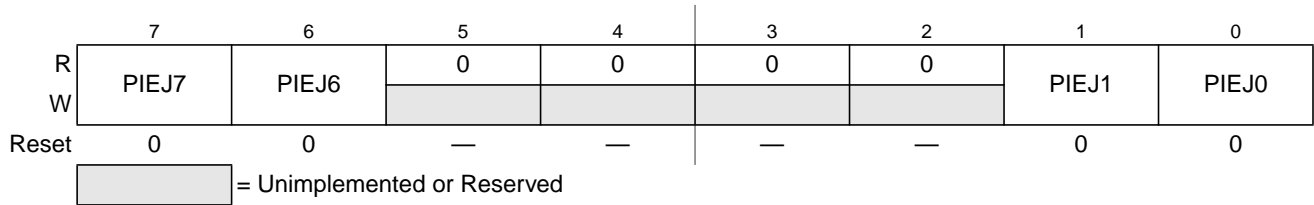
Each flag is set by an active edge on the associated input pin. This could be a rising or a falling edge based on the state of the PPSH register. To clear this flag, write “1” to the corresponding bit in the PIFH register. Writing a “0” has no effect.

Table 5-35. PIFH Field Descriptions

Field	Description
7–0 PIFH[7:0]	<b>Interrupt Flags Port H</b> 0 No active edge pending. Writing a “0” has no effect. 1 Active edge on the associated bit has occurred (an interrupt will occur if the associated enable bit is set). Writing a “1” clears the associated flag.

### 5.3.6.7 Port J Interrupt Enable Register (PIEJ)

Module Base + 0x002E



**Figure 5-45. Port J Interrupt Enable Register (PIEJ)**

Read: Anytime. Write: Anytime.

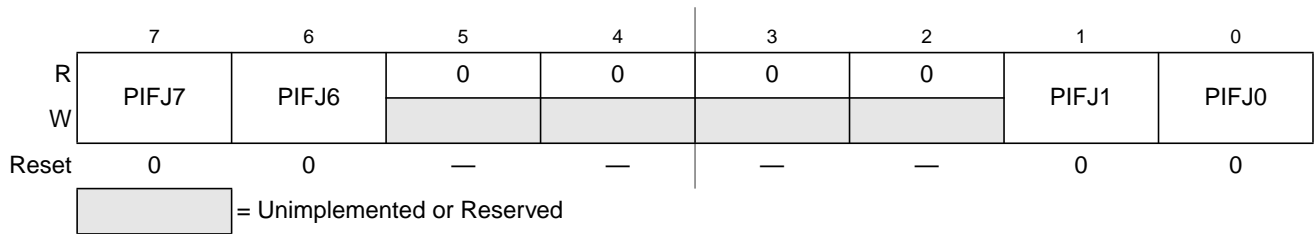
This register disables or enables on a per pin basis the edge sensitive external interrupt associated with port J.

**Table 5-40. PIEJ Field Descriptions**

Field	Description
7, 6, 1, 0 PIEJ[7:6] PIEJ[1:0]	<b>Interrupt Enable Port J</b> 0 Interrupt is disabled (interrupt flag masked). 1 Interrupt is enabled.

### 5.3.6.8 Port J Interrupt Flag Register (PIFJ)

Module Base + 0x002F



**Figure 5-46. Port J Interrupt Flag Register (PIFJ)**

Read: Anytime. Write: Anytime.

Each flag is set by an active edge on the associated input pin. This could be a rising or a falling edge based on the state of the PPSJ register. To clear this flag, write “1” to the corresponding bit in the PIFJ register. Writing a “0” has no effect.

**Table 5-41. PIFJ Field Descriptions**

Field	Description
7, 6, 1, 0 PIFJ[7:6] PIFJ[1:0]	<b>Interrupt Flags Port J</b> 0 No active edge pending. Writing a “0” has no effect. 1 Active edge on the associated bit has occurred (an interrupt will occur if the associated enable bit is set). Writing a “1” clears the associated flag.

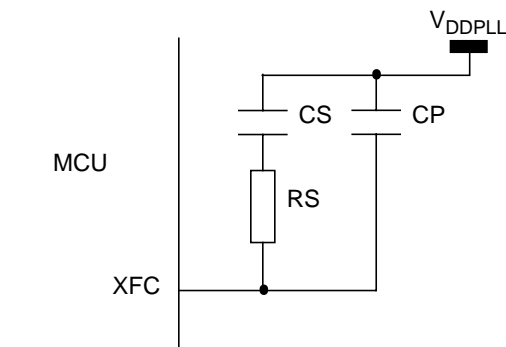


Figure 6-2. PLL Loop Filter Connections

### 6.2.3 $\overline{\text{RESET}}$ — Reset Pin

$\overline{\text{RESET}}$  is an active low bidirectional reset pin. As an input it initializes the MCU asynchronously to a known start-up state. As an open-drain output it indicates that a system reset (internal to MCU) has been triggered.

## 6.3 Memory Map and Register Definition

This section provides a detailed description of all registers accessible in the CRGV4.

### 6.3.1 Module Memory Map

Table 6-1 gives an overview on all CRGV4 registers.

Table 6-1. CRGV4 Memory Map

Address Offset	Use	Access
0x0000	CRG Synthesizer Register (SYNR)	R/W
0x0001	CRG Reference Divider Register (REFDV)	R/W
0x0002	CRG Test Flags Register (CTFLG) <sup>1</sup>	R/W
0x0003	CRG Flags Register (CRGFLG)	R/W
0x0004	CRG Interrupt Enable Register (CRGINT)	R/W
0x0005	CRG Clock Select Register (CLKSEL)	R/W
0x0006	CRG PLL Control Register (PLLCTL)	R/W
0x0007	CRG RTI Control Register (RTICTL)	R/W
0x0008	CRG COP Control Register (COPCTL)	R/W
0x0009	CRG Force and Bypass Test Register (FORBYP) <sup>2</sup>	R/W
0x000A	CRG Test Control Register (CTCTL) <sup>3</sup>	R/W
0x000B	CRG COP Arm/Timer Reset (ARMCOP)	R/W

<sup>1</sup> CTFLG is intended for factory test purposes only.

<sup>2</sup> FORBYP is intended for factory test purposes only.

<sup>3</sup> CTCTL is intended for factory test purposes only.



running at minimum operating frequency; this mode of operation is called self-clock mode. This requires CME = 1 and SCME = 1. If the MCU was clocked by the PLL clock prior to entering self-clock mode, the PLLSEL bit will be cleared. If the external clock signal has stabilized again, the CRG will automatically select OSCCLK to be the system clock and return to normal mode. See Section 6.4.4, “Clock Quality Checker” for more information on entering and leaving self-clock mode.

### NOTE

In order to detect a potential clock loss, the CME bit should be always enabled (CME=1).

If CME bit is disabled and the MCU is configured to run on PLL clock (PLLCLK), a loss of external clock (OSCCLK) will not be detected and will cause the system clock to drift towards the VCO's minimum frequency  $f_{SCM}$ . As soon as the external clock is available again the system clock ramps up to its PLL target frequency. If the MCU is running on external clock any loss of clock will cause the system to go static.

## 6.4.8 Low-Power Operation in Run Mode

The RTI can be stopped by setting the associated rate select bits to 0.

The COP can be stopped by setting the associated rate select bits to 0.

## 6.4.9 Low-Power Operation in Wait Mode

The WAI instruction puts the MCU in a low power consumption stand-by mode depending on setting of the individual bits in the CLKSEL register. All individual wait mode configuration bits can be superposed. This provides enhanced granularity in reducing the level of power consumption during wait mode. Table 6-10 lists the individual configuration bits and the parts of the MCU that are affected in wait mode.

**Table 6-10. MCU Configuration During Wait Mode**

	PLLWAI	CWAI	SYSWAI	RTIWAI	COPWAI	ROAWAI
<b>PLL</b>	stopped	—	—	—	—	—
<b>Core</b>	—	stopped	stopped	—	—	—
<b>System</b>	—	—	stopped	—	—	—
<b>RTI</b>	—	—	—	stopped	—	—
<b>COP</b>	—	—	—	—	stopped	—
<b>Oscillator</b>	—	—	—	—	—	reduced <sup>1</sup>

<sup>1</sup> Refer to oscillator block description for availability of a reduced oscillator amplitude.

After executing the WAI instruction the core requests the CRG to switch MCU into wait mode. The CRG then checks whether the PLLWAI, CWAI and SYSWAI bits are asserted (see Figure 6-23). Depending on the configuration the CRG switches the system and core clocks to OSCCLK by clearing the PLLSEL bit, disables the PLL, disables the core clocks and finally disables the remaining system clocks. As soon as all clocks are switched off wait mode is active.

### 8.3.2.5 ATD Control Register 4 (ATDCTL4)

This register selects the conversion clock frequency, the length of the second phase of the sample time and the resolution of the A/D conversion (i.e.: 8-bits or 10-bits). Writes to this register will abort current conversion sequence but will not start a new sequence.

Module Base + 0x0004

	7	6	5	4	3	2	1	0
R	SRES8	SMP1	SMP0	PRS4	PRS3	PRS2	PRS1	PRS0
W								
Reset	0	0	0	0	0	1	0	1

Figure 8-7. ATD Control Register 4 (ATDCTL4)

Read: Anytime

Write: Anytime

Table 8-10. ATDCTL4 Field Descriptions

Field	Description
7 SRES8	<b>A/D Resolution Select</b> — This bit selects the resolution of A/D conversion results as either 8 or 10 bits. The A/D converter has an accuracy of 10 bits; however, if low resolution is required, the conversion can be speeded up by selecting 8-bit resolution. 0 10-bit resolution 1 8-bit resolution
6–5 SMP[1:0]	<b>Sample Time Select</b> — These two bits select the length of the second phase of the sample time in units of ATD conversion clock cycles. Note that the ATD conversion clock period is itself a function of the prescaler value (bits PRS4–0). The sample time consists of two phases. The first phase is two ATD conversion clock cycles long and transfers the sample quickly (via the buffer amplifier) onto the A/D machine's storage node. The second phase attaches the external analog signal directly to the storage node for final charging and high accuracy. Table 8-11 lists the lengths available for the second sample phase.
4–0 PRS[4:0]	<b>ATD Clock Prescaler</b> — These 5 bits are the binary value prescaler value PRS. The ATD conversion clock frequency is calculated as follows: $\text{ATDclock} = \frac{[\text{BusClock}]}{[\text{PRS} + 1]} \times 0.5$ <b>Note:</b> The maximum ATD conversion clock frequency is half the bus clock. The default (after reset) prescaler value is 5 which results in a default ATD conversion clock frequency that is bus clock divided by 12. Table 8-12 illustrates the divide-by operation and the appropriate range of the bus clock.

Table 8-11. Sample Time Select

SMP1	SMP0	Length of 2nd Phase of Sample Time
0	0	2 A/D conversion clock periods
0	1	4 A/D conversion clock periods
1	0	8 A/D conversion clock periods
1	1	16 A/D conversion clock periods

Table 10-9. CANRFLG Register Field Descriptions

Field	Description
7 WUPIF	<b>Wake-Up Interrupt Flag</b> — If the MSCAN detects CAN bus activity while in sleep mode (see Section 10.4.5.4, “MSCAN Sleep Mode,”) and WUPE = 1 in CANTCTL0 (see Section 10.3.2.1, “MSCAN Control Register 0 (CANCTL0)”), the module will set WUPIF. If not masked, a wake-up interrupt is pending while this flag is set. 0 No wake-up activity observed while in sleep mode 1 MSCAN detected activity on the CAN bus and requested wake-up
6 CSCIF	<b>CAN Status Change Interrupt Flag</b> — This flag is set when the MSCAN changes its current CAN bus status due to the actual value of the transmit error counter (TEC) and the receive error counter (REC). An additional 4-bit (RSTAT[1:0], TSTAT[1:0]) status register, which is split into separate sections for TEC/REC, informs the system on the actual CAN bus status (see Section 10.3.2.6, “MSCAN Receiver Interrupt Enable Register (CANRIER)”). If not masked, an error interrupt is pending while this flag is set. CSCIF provides a blocking interrupt. That guarantees that the receiver/transmitter status bits (RSTAT/TSTAT) are only updated when no CAN status change interrupt is pending. If the TECs/RECs change their current value after the CSCIF is asserted, which would cause an additional state change in the RSTAT/TSTAT bits, these bits keep their status until the current CSCIF interrupt is cleared again. 0 No change in CAN bus status occurred since last interrupt 1 MSCAN changed current CAN bus status
5:4 RSTAT[1:0]	<b>Receiver Status Bits</b> — The values of the error counters control the actual CAN bus status of the MSCAN. As soon as the status change interrupt flag (CSCIF) is set, these bits indicate the appropriate receiver related CAN bus status of the MSCAN. The coding for the bits RSTAT1, RSTAT0 is: 00 RxOK: $0 \leq \text{receive error counter} \leq 96$ 01 RxWRN: $96 < \text{receive error counter} \leq 127$ 10 RxERR: $127 < \text{receive error counter}$ 11 Bus-off <sup>1</sup> : transmit error counter $> 255$
3:2 TSTAT[1:0]	<b>Transmitter Status Bits</b> — The values of the error counters control the actual CAN bus status of the MSCAN. As soon as the status change interrupt flag (CSCIF) is set, these bits indicate the appropriate transmitter related CAN bus status of the MSCAN. The coding for the bits TSTAT1, TSTAT0 is: 00 TxOK: $0 \leq \text{transmit error counter} \leq 96$ 01 TxWRN: $96 < \text{transmit error counter} \leq 127$ 10 TxERR: $127 < \text{transmit error counter} \leq 255$ 11 Bus-Off: transmit error counter $> 255$
1 OVRIF	<b>Overrun Interrupt Flag</b> — This flag is set when a data overrun condition occurs. If not masked, an error interrupt is pending while this flag is set. 0 No data overrun condition 1 A data overrun detected
0 RXF <sup>2</sup>	<b>Receive Buffer Full Flag</b> — RXF is set by the MSCAN when a new message is shifted in the receiver FIFO. This flag indicates whether the shifted buffer is loaded with a correctly received message (matching identifier, matching cyclic redundancy code (CRC) and no other errors detected). After the CPU has read that message from the RxFG buffer in the receiver FIFO, the RXF flag must be cleared to release the buffer. A set RXF flag prohibits the shifting of the next FIFO entry into the foreground buffer (RxFG). If not masked, a receive interrupt is pending while this flag is set. 0 No new message available within the RxFG 1 The receiver FIFO is not empty. A new message is available in the RxFG

<sup>1</sup> Redundant Information for the most critical CAN bus status which is “bus-off”. This only occurs if the Tx error counter exceeds a number of 255 errors. Bus-off affects the receiver state. As soon as the transmitter leaves its bus-off state the receiver state skips to RxOK too. Refer also to TSTAT[1:0] coding in this register.

<sup>2</sup> To ensure data integrity, do not read the receive buffer registers while the RXF flag is cleared. For MCUs with dual CPUs, reading the receive buffer registers while the RXF flag is cleared may result in a CPU fault condition.

Table 12-7. Example SPI Baud Rate Selection (25 MHz Bus Clock) (continued)

SPPR2	SPPR1	SPPR0	SPR2	SPR1	SPR0	Baud Rate Divisor	Baud Rate
1	0	0	1	1	1	1280	19.53 kHz
1	0	1	0	0	0	12	2.08333 MHz
1	0	1	0	0	1	24	1.04167 MHz
1	0	1	0	1	0	48	520.83 kHz
1	0	1	0	1	1	96	260.42 kHz
1	0	1	1	0	0	192	130.21 kHz
1	0	1	1	0	1	384	65.10 kHz
1	0	1	1	1	0	768	32.55 kHz
1	0	1	1	1	1	1536	16.28 kHz
1	1	0	0	0	0	14	1.78571 MHz
1	1	0	0	0	1	28	892.86 kHz
1	1	0	0	1	0	56	446.43 kHz
1	1	0	0	1	1	112	223.21 kHz
1	1	0	1	0	0	224	111.61 kHz
1	1	0	1	0	1	448	55.80 kHz
1	1	0	1	1	0	896	27.90 kHz
1	1	0	1	1	1	1792	13.95 kHz
1	1	1	0	0	0	16	1.5625 MHz
1	1	1	0	0	1	32	781.25 kHz
1	1	1	0	1	0	64	390.63 kHz
1	1	1	0	1	1	128	195.31 kHz
1	1	1	1	0	0	256	97.66 kHz
1	1	1	1	0	1	512	48.83 kHz
1	1	1	1	1	0	1024	24.41 kHz
1	1	1	1	1	1	2048	12.21 kHz

### NOTE

In slave mode of SPI S-clock speed DIV2 is not supported.

### 12.3.2.4 SPI Status Register (SPISR)

Module Base 0x0003

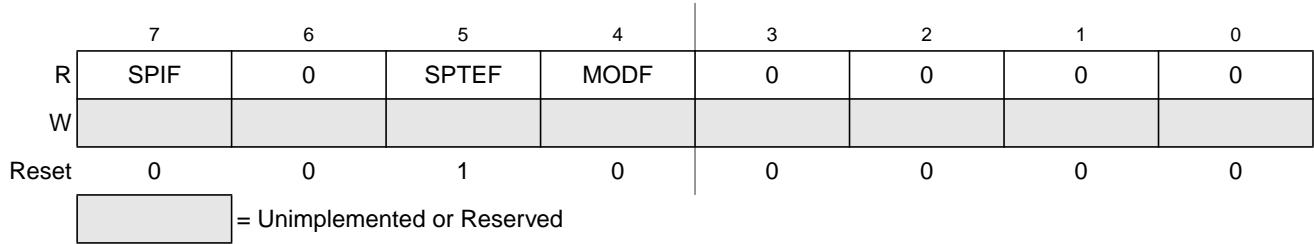


Figure 12-6. SPI Status Register (SPISR)

Read: anytime

Write: has no effect

Table 12-8. SPISR Field Descriptions

Field	Description
7 SPIF	<b>SPIF Interrupt Flag</b> — This bit is set after a received data byte has been transferred into the SPI Data Register. This bit is cleared by reading the SPISR register (with SPIF set) followed by a read access to the SPI Data Register. 0 Transfer not yet complete 1 New data copied to SPIDR
5 SPTEF	<b>SPI Transmit Empty Interrupt Flag</b> — If set, this bit indicates that the transmit data register is empty. To clear this bit and place data into the transmit data register, SPISR has to be read with SPTEF = 1, followed by a write to SPIDR. Any write to the SPI Data Register without reading SPTEF = 1, is effectively ignored. 0 SPI Data register not empty 1 SPI Data register empty
4 MODF	<b>Mode Fault Flag</b> — This bit is set if the $\overline{SS}$ input becomes low while the SPI is configured as a master and mode fault detection is enabled, MODFEN bit of SPICR2 register is set. Refer to MODFEN bit description in Section 12.3.2.2, “SPI Control Register 2 (SPICR2).” The flag is cleared automatically by a read of the SPI Status Register (with MODF set) followed by a write to the SPI Control Register 1. 0 Mode fault has not occurred. 1 Mode fault has occurred.

### 12.3.2.5 SPI Data Register (SPIDR)

Module Base 0x0005

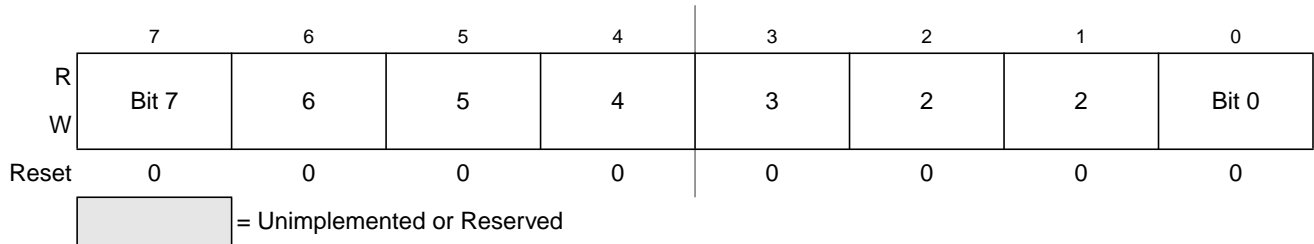


Figure 12-7. SPI Data Register (SPIDR)

Read: anytime; normally read only after SPIF is set

## 13.4 Functional Description

### 13.4.1 PWM Clock Select

There are four available clocks: clock A, clock B, clock SA (scaled A), and clock SB (scaled B). These four clocks are based on the bus clock.

Clock A and B can be software selected to be 1, 1/2, 1/4, 1/8,..., 1/64, 1/128 times the bus clock. Clock SA uses clock A as an input and divides it further with a reloadable counter. Similarly, clock SB uses clock B as an input and divides it further with a reloadable counter. The rates available for clock SA are software selectable to be clock A divided by 2, 4, 6, 8,..., or 512 in increments of divide by 2. Similar rates are available for clock SB. Each PWM channel has the capability of selecting one of two clocks, either the pre-scaled clock (clock A or B) or the scaled clock (clock SA or SB).

The block diagram in Figure 13-18 shows the four different clocks and how the scaled clocks are created.

#### 13.4.1.1 Prescale

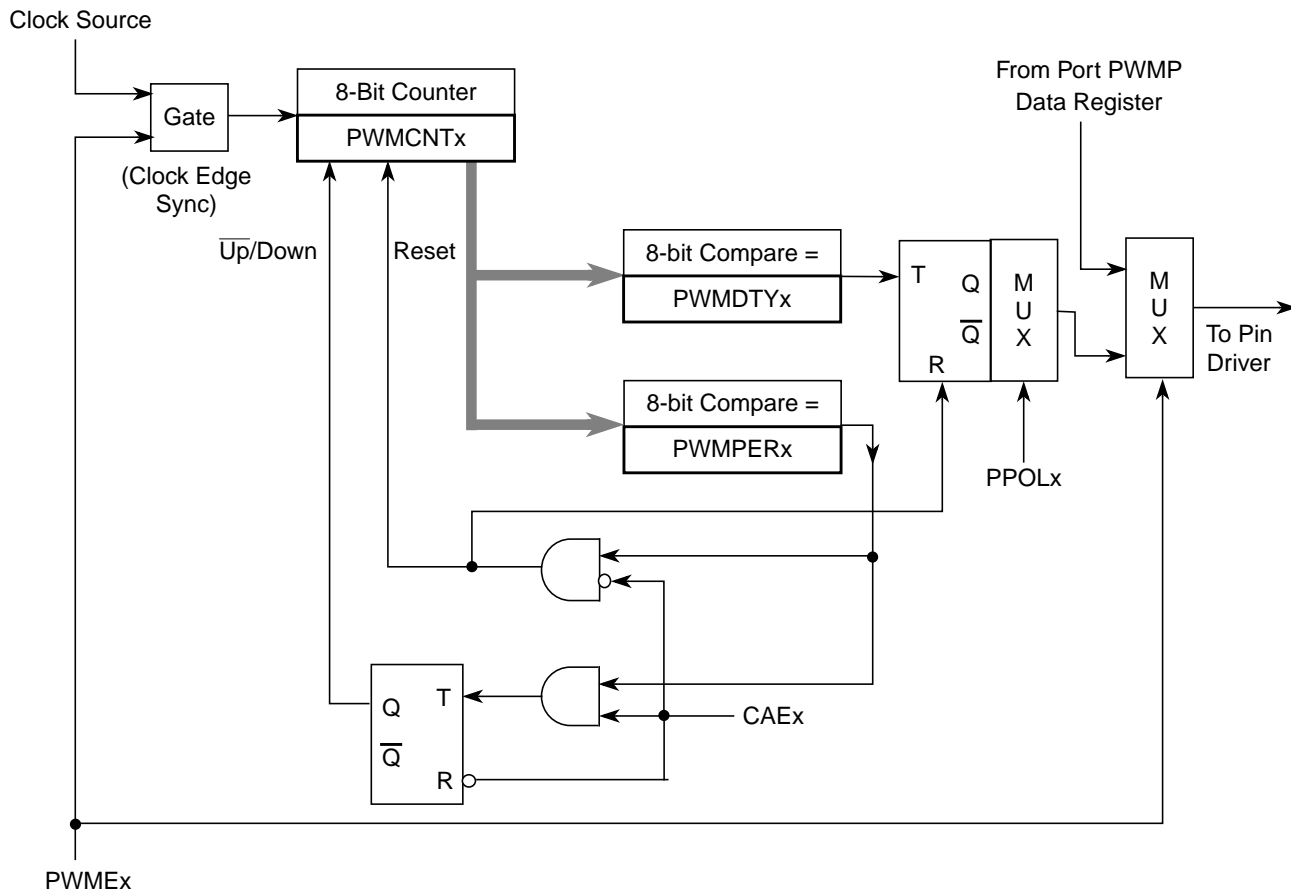
The input clock to the PWM prescaler is the bus clock. It can be disabled whenever the part is in freeze mode by setting the PFRZ bit in the PWMCTL register. If this bit is set, whenever the MCU is in freeze mode (freeze mode signal active) the input clock to the prescaler is disabled. This is useful for emulation in order to freeze the PWM. The input clock can also be disabled when all eight PWM channels are disabled (PWME7-0 = 0). This is useful for reducing power by disabling the prescale counter.

Clock A and clock B are scaled values of the input clock. The value is software selectable for both clock A and clock B and has options of 1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, or 1/128 times the bus clock. The value selected for clock A is determined by the PCKA2, PCKA1, PCKA0 bits in the PWMPRCLK register. The value selected for clock B is determined by the PCKB2, PCKB1, PCKB0 bits also in the PWMPRCLK register.

#### 13.4.1.2 Clock Scale

The scaled A clock uses clock A as an input and divides it further with a user programmable value and then divides this by 2. The scaled B clock uses clock B as an input and divides it further with a user programmable value and then divides this by 2. The rates available for clock SA are software selectable to be clock A divided by 2, 4, 6, 8,..., or 512 in increments of divide by 2. Similar rates are available for clock SB.

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. Shown below in Figure 13-19 is the block diagram for the PWM timer.



### Figure 13-19. PWM Timer Channel Block Diagram

Each PWM channel has an enable bit (PWME<sub>x</sub>) to start its waveform output. When any of the PWME<sub>x</sub> bits are set (PWME<sub>x</sub> = 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 due to the synchronization of PWME<sub>x</sub> and the clock source. An exception to this is when channels are concatenated. Refer to Section 13.4.2.7, “PWM 16-Bit Functions” for more detail.

The first PWM cycle after enabling the channel can be irregular.

## 14.4.2 Input Capture

Clearing the I/O (input/output) select bit, IOSx, configures channel x as an input capture channel. The input capture function captures the time at which an external event occurs. When an active edge occurs on the pin of an input capture channel, the timer transfers the value in the timer counter into the timer channel registers, TCx.

The minimum pulse width for the input capture input is greater than two bus clocks.

An input capture on channel x sets the CxF flag. The CxI bit enables the CxF flag to generate interrupt requests. Timer module must stay enabled (TEN bit of TSCR1 must be set to one) while clearing CxF (writing one to CxF).

## 14.4.3 Output Compare

Setting the I/O select bit, IOSx, configures channel x as an output compare channel. The output compare function can generate a periodic pulse with a programmable polarity, duration, and frequency. When the timer counter reaches the value in the channel registers of an output compare channel, the timer can set, clear, or toggle the channel pin. An output compare on channel x sets the CxF flag. The CxI bit enables the CxF flag to generate interrupt requests. Timer module must stay enabled (TEN bit of TSCR1 register must be set to one) while clearing CxF (writing one to CxF).

The output mode and level bits, OMx and OLx, select set, clear, toggle on output compare. Clearing both OMx and OLx disconnects the pin from the output logic.

Setting a force output compare bit, FOCx, causes an output compare on channel x. A forced output compare does not set the channel flag.

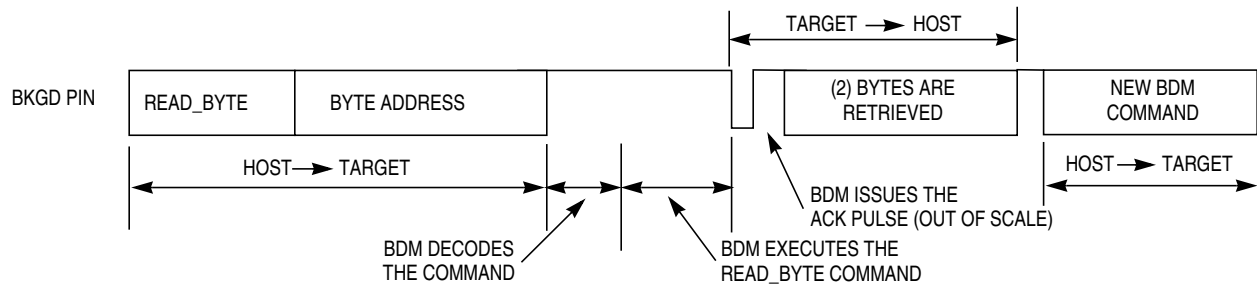
A successful output compare on channel 7 overrides output compares on all other output compare channels. The output compare 7 mask register masks the bits in the output compare 7 data register. The timer counter reset enable bit, TCRE, enables channel 7 output compares to reset the timer counter. A channel 7 output compare can reset the timer counter even if the IOC7 pin is being used as the pulse accumulator input.

Writing to the timer port bit of an output compare pin does not affect the pin state. The value written is stored in an internal latch. When the pin becomes available for general-purpose output, the last value written to the bit appears at the pin.

When TCRE is set and TC7 is not equal to 0, then TCNT will cycle from 0 to TC7. When TCNT reaches TC7 value, it will last only one bus cycle then reset to 0.



Figure 16-11 shows the ACK handshake protocol in a command level timing diagram. The READ\_BYTE instruction is used as an example. First, the 8-bit instruction opcode is sent by the host, followed by the address of the memory location to be read. The target BDM decodes the instruction. A bus cycle is grabbed (free or stolen) by the BDM and it executes the READ\_BYTE operation. Having retrieved the data, the BDM issues an ACK pulse to the host controller, indicating that the addressed byte is ready to be retrieved. After detecting the ACK pulse, the host initiates the byte retrieval process. Note that data is sent in the form of a word and the host needs to determine which is the appropriate byte based on whether the address was odd or even.



**Figure 16-11. Handshake Protocol at Command Level**

Differently from the normal bit transfer (where the host initiates the transmission), the serial interface ACK handshake pulse is initiated by the target MCU by issuing a falling edge in the BKGD pin. The hardware handshake protocol in Figure 16-10 specifies the timing when the BKGD pin is being driven, so the host should follow this timing constraint in order to avoid the risk of an electrical conflict in the BKGD pin.

### NOTE

The only place the BKGD pin can have an electrical conflict is when one side is driving low and the other side is issuing a speedup pulse (high). Other “highs” are pulled rather than driven. However, at low rates the time of the speedup pulse can become lengthy and so the potential conflict time becomes longer as well.

The ACK handshake protocol does not support nested ACK pulses. If a BDM command is not acknowledge by an ACK pulse, the host needs to abort the pending command first in order to be able to issue a new BDM command. When the CPU enters WAIT or STOP while the host issues a command that requires CPU execution (e.g., WRITE\_BYTE), the target discards the incoming command due to the WAIT or STOP being detected. Therefore, the command is not acknowledged by the target, which means that the ACK pulse will not be issued in this case. After a certain time the host should decide to abort the ACK sequence in order to be free to issue a new command. Therefore, the protocol should provide a mechanism in which a command, and therefore a pending ACK, could be aborted.

### NOTE

Differently from a regular BDM command, the ACK pulse does not provide a time out. This means that in the case of a WAIT or STOP instruction being executed, the ACK would be prevented from being issued. If not aborted, the ACK would remain pending indefinitely. See the handshake abort procedure described in Section 16.4.8, “Hardware Handshake Abort Procedure.”

- Data associated with event B trigger modes
- Detail report mode stores address and data for all cycles except program (P) and free (f) cycles
- Current instruction address when in profiling mode
- BGND is not considered a change-of-flow (cof) by the debugger

## 17.1.2 Modes of Operation

There are two main modes of operation: breakpoint mode and debug mode. Each one is mutually exclusive of the other and selected via a software programmable control bit.

In the breakpoint mode there are two sub-modes of operation:

- Dual address mode, where a match on either of two addresses will cause the system to enter background debug mode (BDM) or initiate a software interrupt (SWI).
- Full breakpoint mode, where a match on address and data will cause the system to enter background debug mode (BDM) or initiate a software interrupt (SWI).

In debug mode, there are several sub-modes of operation.

- Trigger modes

There are many ways to create a logical trigger. The trigger can be used to capture bus information either starting from the trigger or ending at the trigger. Types of triggers (A and B are registers):

- A only
- A or B
- A then B
- Event only B (data capture)
- A then event only B (data capture)
- A and B, full mode
- A and not B, full mode
- Inside range
- Outside range

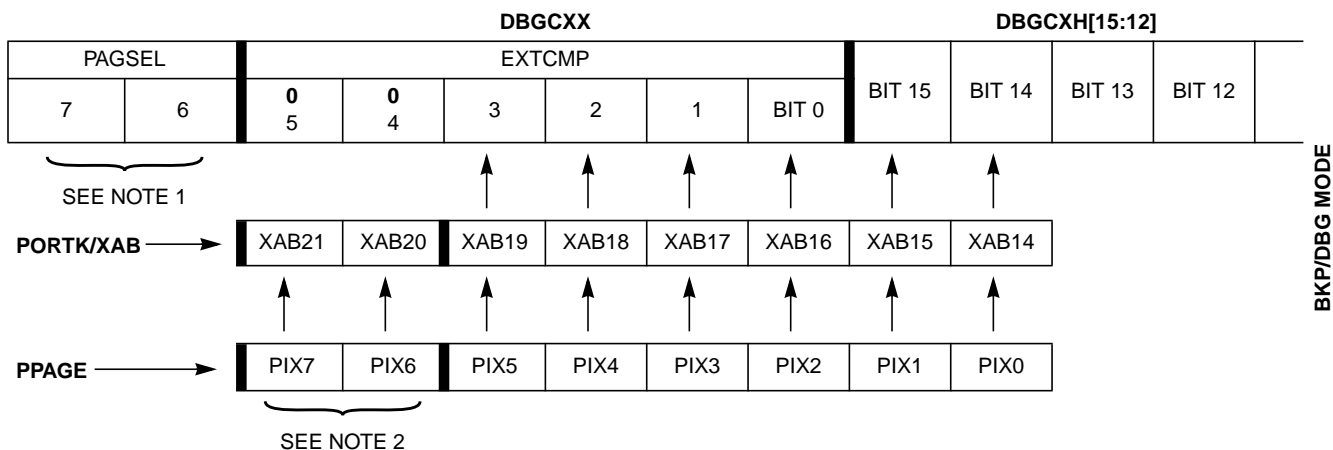
- Capture modes

There are several capture modes. These determine which bus information is saved and which is ignored.

- Normal: save change-of-flow program fetches
- Loop1: save change-of-flow program fetches, ignoring duplicates
- Detail: save all bus operations except program and free cycles
- Profile: poll target from external device

## 17.1.3 Block Diagram

Figure 17-1 is a block diagram of this module in breakpoint mode. Figure 17-2 is a block diagram of this module in debug mode.



## NOTES:

1. In BKP and DBG mode, PAGSEL selects the type of paging as shown in Table 17-11.
2. Current HCS12 implementations are limited to six PPAGE bits, PIX[5:0]. Therefore, EXTCMP[5:4] = 00.

**Figure 17-10. Comparator C Extended Comparison in BKP/DBG Mode**

## 17.3.2.6 Debug Comparator C Register (DBGCC)

Module Base + 0x0026

Starting address location affected by INITRG register setting.

	15	14	13	12	11	10	9	8
R	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

**Figure 17-11. Debug Comparator C Register High (DBGCCCH)**

Module Base + 0x0027

Starting address location affected by INITRG register setting.

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

= Unimplemented or Reserved

**Figure 17-12. Debug Comparator C Register Low (DBGCCCL)**

Table 20-6. External Stretch Bit Definition

Stretch Bit EXSTR1	Stretch Bit EXSTR0	Number of E Clocks Stretched
0	0	0
0	1	1
1	0	2
1	1	3

### 20.3.2.5 Reserved Test Register 0 (MTST0)

Module Base + 0x0014

Starting address location affected by INITRG register setting.

	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

Figure 20-7. Reserved Test Register 0 (MTST0)

Read: Anytime

Write: No effect — this register location is used for internal test purposes.

### 20.3.2.6 Reserved Test Register 1 (MTST1)

Module Base + 0x0017

Starting address location affected by INITRG register setting.

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

= Unimplemented or Reserved

Figure 20-8. Reserved Test Register 1 (MTST1)

Read: Anytime

Write: No effect — this register location is used for internal test purposes.

**Table A-8. Supply Current Characteristics**

Conditions are shown in Table A-4 unless otherwise noted						
Num	Rating	Symbol	Min	Typ	Max	Unit
1	Run supply currents Single Chip, Internal regulator enabled	$I_{DD5}$	—	—	65	mA
2	Wait Supply current All modules enabled only RTI enabled <sup>1</sup>	$I_{DDW}$	— —	— —	40 5	mA
3	Pseudo Stop Current (RTI and COP enabled) <sup>1,2</sup> —40°C 27°C 70°C 85°C "C" Temp Option 100°C 105°C "V" Temp Option 120°C 125°C "M" Temp Option 140°C	$I_{DDPS}$	— — — — — — — — —	90 130 155 180 250 295 470 520 1000	350 — — — 1200 — 2400 — 5000	μA
4	Pseudo Stop Current (RTI and COP disabled) <sup>1,2</sup> —40°C 27°C 70°C 85°C "C" Temp Option 100°C 105°C "V" Temp Option 120°C 125°C "M" Temp Option 140°C	$I_{DDPS}$	— — — — — — — — —	40 80 105 130 200 245 420 470 800	200 — — — 1000 — 2000 — 5000	μA
5	Stop Current <sup>2</sup> —40°C 27°C 70°C 85°C "C" Temp Option 100°C 105°C "V" Temp Option 120°C 125°C "M" Temp Option 140°C	$I_{DD5}$	— — — — — — — — —	20 60 85 110 180 225 400 450 600	100 — — — 800 — 1800 — 5000	μA

<sup>1</sup> PLL off

<sup>2</sup> All those low power dissipation levels  $T_J = T_A$  can be assumed.