



Welcome to [E-XFL.COM](#)

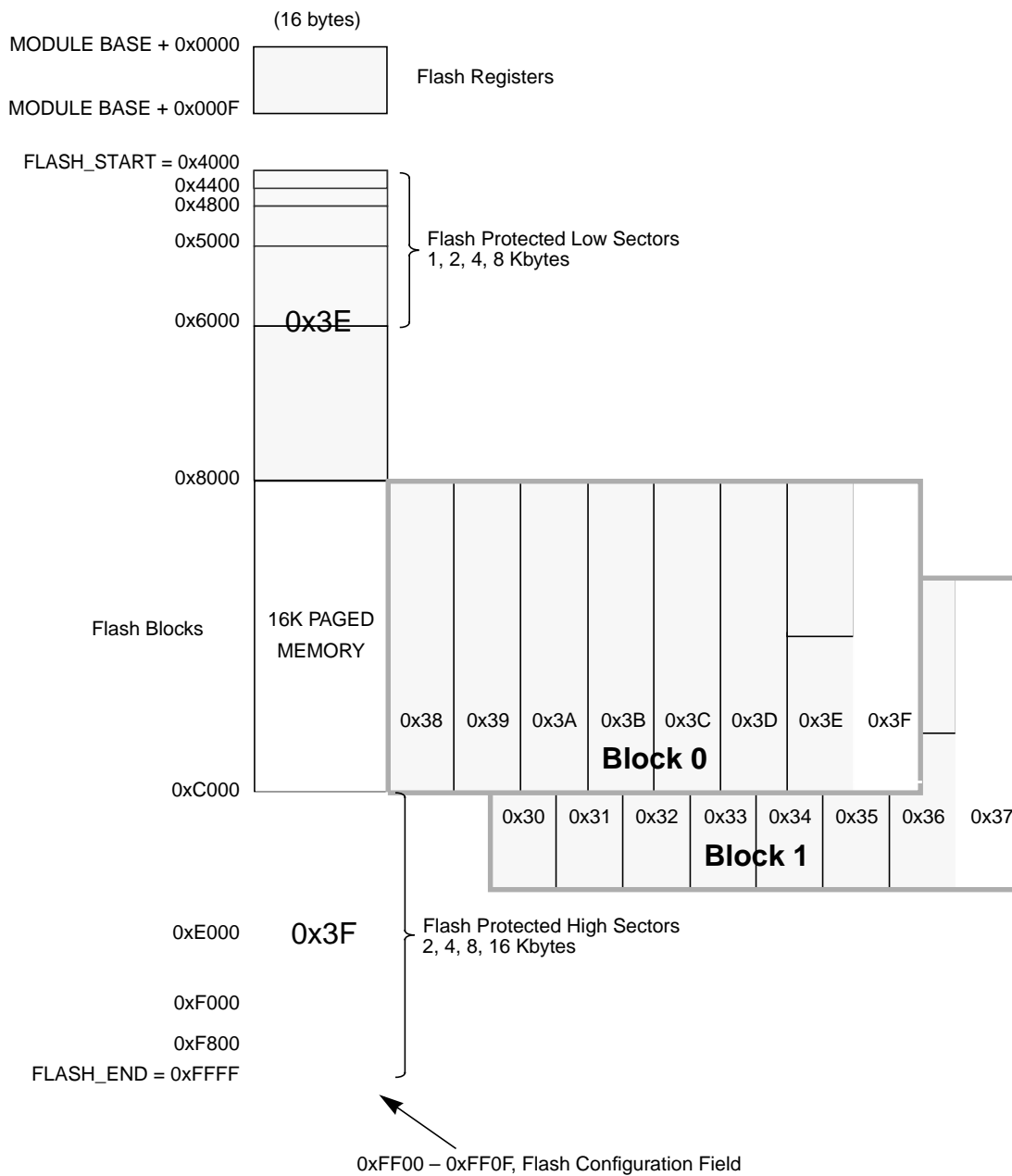
### What is "[Embedded - Microcontrollers](#)"?

"[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	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	59
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 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12kg256vfue">https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12kg256vfue</a>



Note: 0x30–0x3F correspond to the PPAGE register content

### Figure 2-2. Flash Memory Map

be stored in the lower six bits of FDATALO. The faulty parity bits remain readable until the start of the next command write sequence.

### 2.3.2.12 RESERVED1

This register is reserved for factory testing and is not accessible.

Module Base + 0x000C

	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

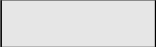
 = Unimplemented or Reserved

Figure 2-19. RESERVED1

All bits read 0 and are not writable.

### 2.3.2.13 RESERVED2

This register is reserved for factory testing and is not accessible.

Module Base + 0x000D

	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 2-20. RESERVED2

All bits read 0 and are not writable.

### 2.3.2.14 RESERVED3

This register is reserved for factory testing and is not accessible.

Module Base + 0x000E

	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 2-21. RESERVED3

Except for the sector erase abort command, a buffered command will wait for the active operation to be completed before being launched. The sector erase abort command is launched when the CBEIF flag is cleared as part of a sector erase abort command write sequence. After a command is launched, the completion of the command operation is indicated by the setting of the CCIF flag. The CCIF flag only sets when all active and buffered commands have been completed.

### 2.4.1.3 Valid Flash Commands

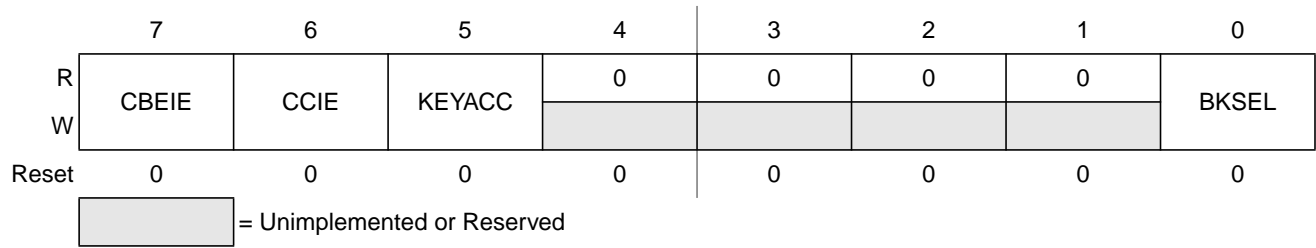
Table 2-21 summarizes the valid Flash commands along with the effects of the commands on the Flash block.

**Table 2-21. Valid Flash Command Description**

FCMDB	NVM Command	Function on Flash Memory
0x05	Erase Verify	Verify all memory bytes in the Flash block are erased. If the Flash block is erased, the BLANK flag in the FSTAT register will set upon command completion.
0x06	Data Compress	Compress data from a selected portion of the Flash block. The resulting signature is stored in the FDATA register.
0x20	Program	Program a word (two bytes) in the Flash block.
0x40	Sector Erase	Erase all memory bytes in a sector of the Flash block.
0x41	Mass Erase	Erase all memory bytes in the Flash block. A mass erase of the full Flash block is only possible when FPLDIS, FPHDIS, and FPOPEN bits in the FPROT register are set prior to launching the command.
0x47	Sector Erase Abort	Abort the sector erase operation. The sector erase operation will terminate according to a set procedure. The Flash sector must not be considered erased if the ACCERR flag is set upon command completion.

A Flash word must be in the erased state before being programmed.  
Cumulative programming of bits within a Flash word is not allowed and will result in invalid data stored.

Module Base + 0x0003



**Figure 3-7. Flash Configuration Register (FCNFG)**

CBEIE, CCIE, KEYACC, DFDIE and BKSEL bits are readable and writable while all remaining bits read 0 and are not writable. KEYACC is only writable if KEYEN (see Section 3.3.2.2) is set to the enabled state.

**Table 3-10. FCNFG Field Descriptions**

Field	Description
7 CBEIE	<b>Command Buffer Empty Interrupt Enable</b> — The CBEIE bit enables an interrupt in case of an empty command buffer in the Flash module. 0 Command buffer empty interrupt disabled. 1 An interrupt will be requested whenever the CBEIF flag (see Section 3.3.2.7, “Flash Status Register (FSTAT)”) is set.
6 CCIE	<b>Command Complete Interrupt Enable</b> — The CCIE bit enables an interrupt in case all commands have been completed in the Flash module. 0 Command complete interrupt disabled. 1 An interrupt will be requested whenever the CCIF flag (see Section 3.3.2.7, “Flash Status Register (FSTAT)”) is set.
5 KEYACC	<b>Enable Security Key Writing</b> 0 Flash writes are interpreted as the start of a command write sequence. 1 Writes to Flash array are interpreted as keys to open the backdoor. Reads of the Flash array return invalid data.
3 DFDIE	<b>Double Fault Detect Interrupt Enable</b> — The DFDIE bit enables an interrupt in case a double bit fault is detected during a Flash block operation. 0 Double bit fault detect interrupt disabled. 1 An interrupt will be requested whenever the DFDIF flag is set (see Section 3.3.2.7, “Flash Status Register (FSTAT)”).

## 3.3.2.5 Flash Protection Register (FPROT)

The banked FPROT register defines which Flash sectors are protected against program or erase operations.

All bits in the FPROT register are readable and writable with restrictions except for RNV[6] which is only readable (see Section 3.3.2.6, “Flash Protection Restrictions”).

During reset, the banked FPROT registers are loaded from the Flash Configuration Field at the address shown in Table 3-11. To change the Flash protection that will be loaded during the reset sequence, the upper sector of the Flash memory must be unprotected, then the Flash Protect/Security byte located as described in Table 3-2 must be reprogrammed. If the DFDIF flag in the FSTAT register is set while reading

### 3.4.1.3.3 Program Command

The program command is used to program a previously erased word in the Flash memory using an embedded algorithm. If the word to be programmed is in a protected area of the Flash block, the PVIOL flag in the FSTAT register will set and the program command will not launch. After the program command has successfully launched, the CCIF flag in the FSTAT register will set after the program operation has completed unless a second command has been buffered.

A summary of the launching of a program operation is shown in Figure 3-27.

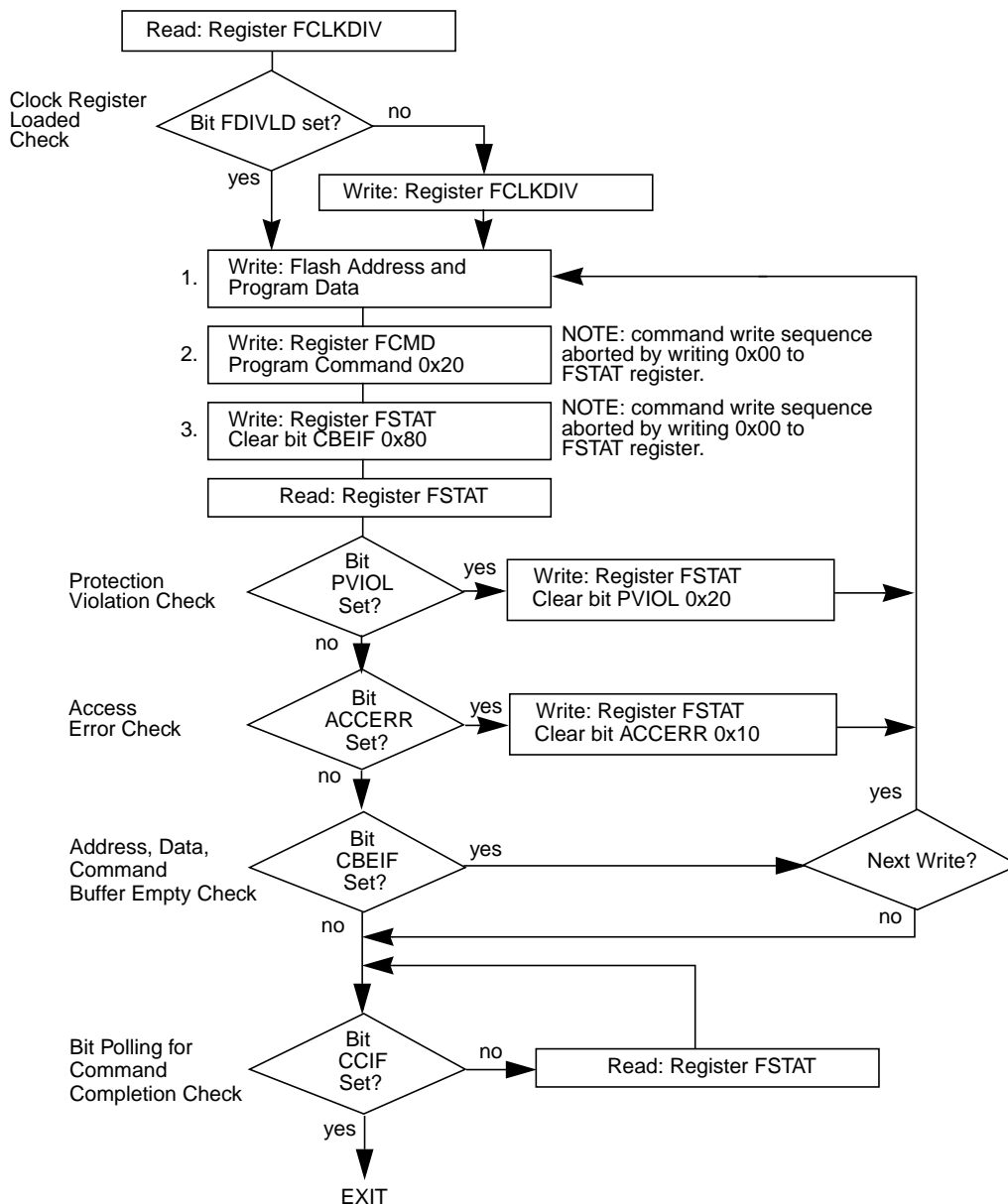


Figure 3-27. Example Program Command Flow

### 3.4.1.3.4 Sector Erase Command

The sector erase command is used to erase the addressed sector in the Flash memory using an embedded algorithm. If the Flash sector to be erased is in a protected area of the Flash block, the PVIOL flag in the FSTAT register will set and the sector erase command will not launch. After the sector erase command has successfully launched, the CCIF flag in the FSTAT register will set after the sector erase operation has completed unless a second command has been buffered.

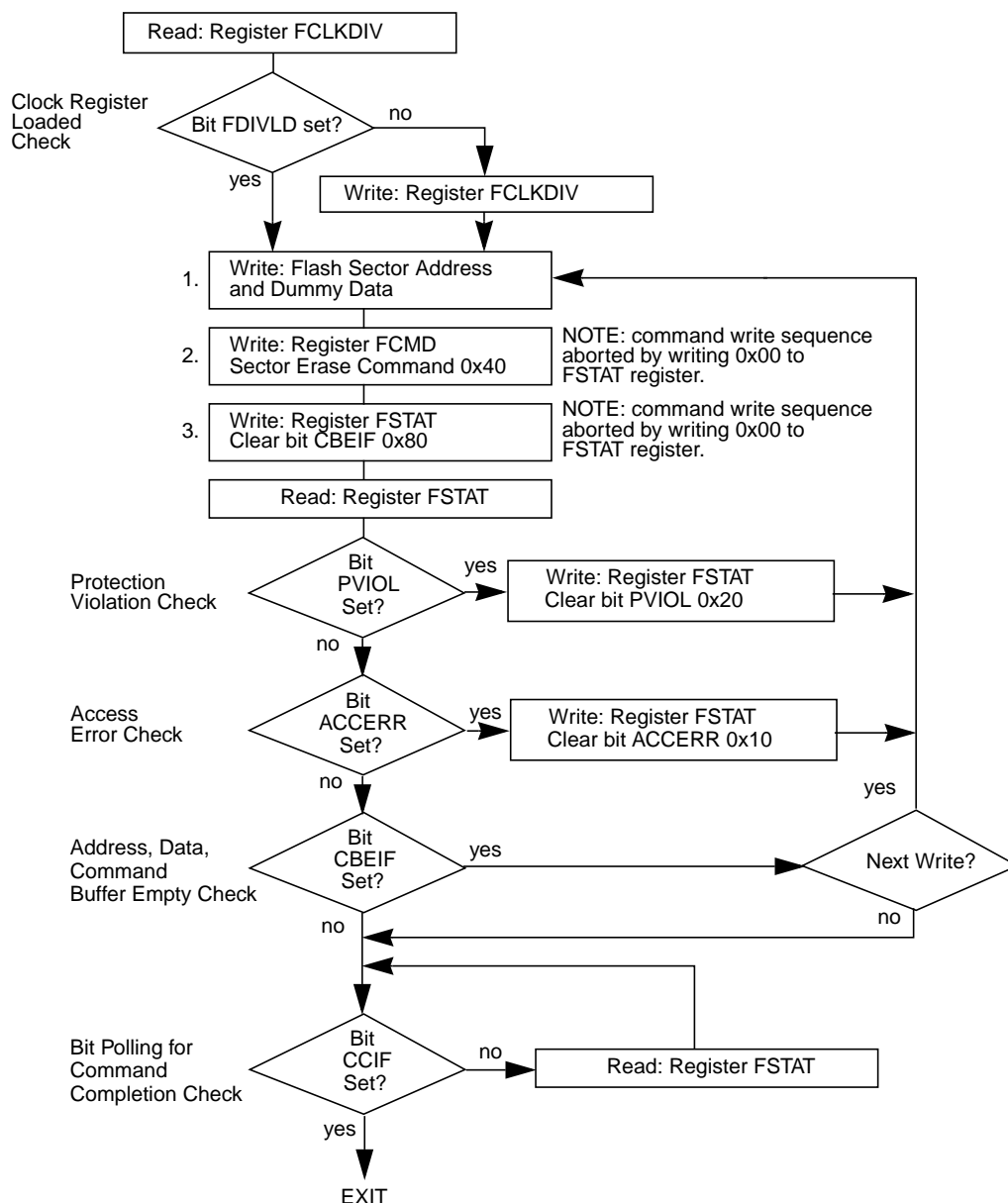


Figure 3-28. Example Sector Erase Command Flow

### 5.3.3.3 Port M Data Direction Register (DDRM)

Module Base + 0x0012

	7	6	5	4	3	2	1	0
R	DDRM7	DDRM6	DDRM5	DDRM4	DDRM3	DDRM2	DDRM1	DDRM0
W								
Reset	0	0	0	0	0	0	0	0

**Figure 5-17. Port M Data Direction Register (DDRM)**

Read: Anytime. Write: Anytime.

This register configures each port M pin as either input or output. The CAN forces the I/O state to be an output for each port line associated with an enabled output (TXCAN4, TXCAN1 and TXCAN0). It also forces the I/O state to be an input for each port line associated with an enabled input (RXCAN4, RXCAN1 and RXCAN0). In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled.

**Table 5-12. DDRM Field Descriptions**

Field	Description
7–0 DDRM[7:0]	<b>Data Direction Port M</b> 0 Associated pin is configured as input. 1 Associated pin is configured as output.

### 5.3.3.4 Port M Reduced Drive Register (RDRM)

Module Base + 0x0013

	7	6	5	4	3	2	1	0
R	RDRM7	RDRM6	RDRM5	RDRM4	RDRM3	RDRM2	RDRM1	RDRM0
W								
Reset	0	0	0	0	0	0	0	0

**Figure 5-18. Port M Reduced Drive Register (RDRM)**

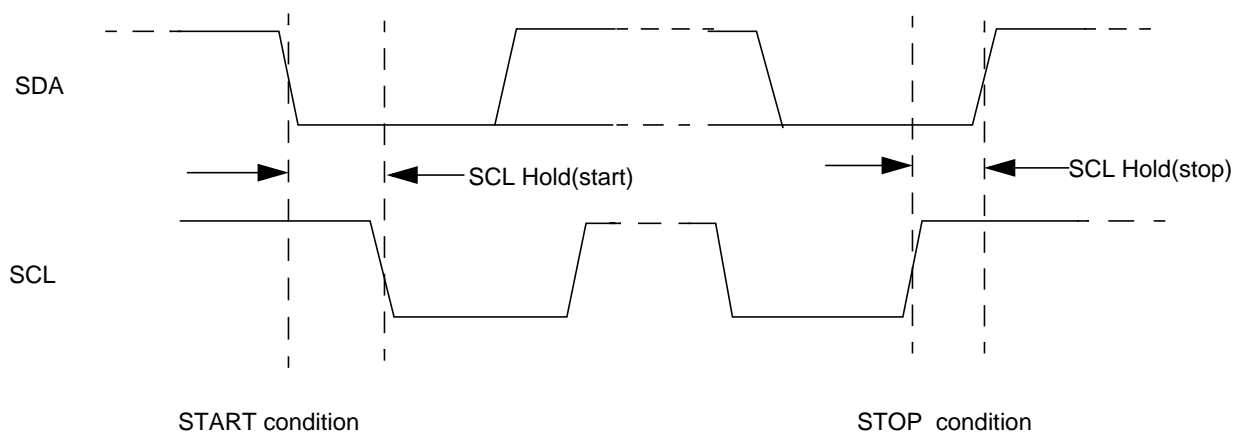
Read: Anytime. Write: Anytime.

This register configures the drive strength of each port M output pin as either full or reduced. If the port is used as input this bit is ignored.

**Table 5-13. RDRM Field Descriptions**

Field	Description
7–0 RDRM[7:0]	<b>Reduced Drive Port M</b> 0 Full drive strength at output. 1 Associated pin drives at about 1/6 of the full drive strength.





**Figure 9-5. SCL Divider and SDA Hold**

The equation used to generate the divider values from the IBFD bits is:

$$\text{SCL Divider} = \text{MUL} \times \{2 \times (\text{scl2tap} + [(\text{SCL\_Tap} - 1) \times \text{tap2tap}] + 2)\}$$

The SDA hold delay is equal to the CPU clock period multiplied by the SDA Hold value shown in Table 9-5. The equation used to generate the SDA Hold value from the IBFD bits is:

$$\text{SDA Hold} = \text{MUL} \times \{\text{scl2tap} + [(\text{SDA\_Tap} - 1) \times \text{tap2tap}] + 3\}$$

The equation for SCL Hold values to generate the start and stop conditions from the IBFD bits is:

$$\text{SCL Hold(start)} = \text{MUL} \times [\text{scl2start} + (\text{SCL\_Tap} - 1) \times \text{tap2tap}]$$

$$\text{SCL Hold(stop)} = \text{MUL} \times [\text{scl2stop} + (\text{SCL\_Tap} - 1) \times \text{tap2tap}]$$

**Table 9-5. IIC Divider and Hold Values (Sheet 1 of 5)**

IBC[7:0] (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
<b>MUL=1</b>				
00	20	7	6	11
01	22	7	7	12
02	24	8	8	13
03	26	8	9	14
04	28	9	10	15
05	30	9	11	16
06	34	10	13	18
07	40	10	16	21
08	28	7	10	15
09	32	7	12	17
0A	36	9	14	19
0B	40	9	16	21
0C	44	11	18	23
0D	48	11	20	25
0E	56	13	24	29
0F	68	13	30	35
10	48	9	18	25

- If there are one or more message buffers scheduled for transmission (TXEx = 0), the MSCAN will continue to transmit until all transmit message buffers are empty (TXEx = 1, transmitted successfully or aborted) and then goes into sleep mode.
- If the MSCAN is receiving, it continues to receive and goes into sleep mode as soon as the CAN bus next becomes idle.
- If the MSCAN is neither transmitting nor receiving, it immediately goes into sleep mode.

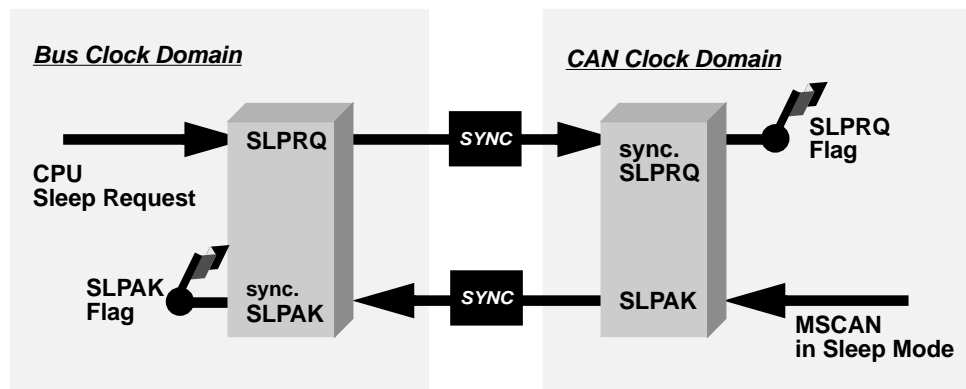


Figure 10-44. Sleep Request / Acknowledge Cycle

#### NOTE

The application software must avoid setting up a transmission (by clearing one or more TXEx flag(s)) and immediately request sleep mode (by setting SLPRQ). Whether the MSCAN starts transmitting or goes into sleep mode directly depends on the exact sequence of operations.

If sleep mode is active, the SLPRQ and SLPK bits are set (Figure 10-44). The application software must use SLPK as a handshake indication for the request (SLPRQ) to go into sleep mode.

When in sleep mode (SLPRQ = 1 and SLPK = 1), the MSCAN stops its internal clocks. However, clocks that allow register accesses from the CPU side continue to run.

If the MSCAN is in bus-off state, it stops counting the 128 occurrences of 11 consecutive recessive bits due to the stopped clocks. The TXCAN pin remains in a recessive state. If RXF = 1, the message can be read and RXF can be cleared. Shifting a new message into the foreground buffer of the receiver FIFO (RxFG) does not take place while in sleep mode.

It is possible to access the transmit buffers and to clear the associated TXE flags. No message abort takes place while in sleep mode.

If the WUPE bit in CANCLT0 is not asserted, the MSCAN will mask any activity it detects on CAN. The RXCAN pin is therefore held internally in a recessive state. This locks the MSCAN in sleep mode (Figure 10-45). WUPE must be set before entering sleep mode to take effect.

In master mode, with slave select output enabled the  $\overline{SS}$  line is always deasserted and reasserted between successive transfers for at least minimum idle time.

### 12.4.3.3 CPHA = 1 Transfer Format

Some peripherals require the first SCK edge before the first data bit becomes available at the data out pin, the second edge clocks data into the system. In this format, the first SCK edge is issued by setting the CPHA bit at the beginning of the 8-cycle transfer operation.

The first edge of SCK occurs immediately after the half SCK clock cycle synchronization delay. This first edge commands the slave to transfer its first data bit to the serial data input pin of the master.

A half SCK cycle later, the second edge appears on the SCK pin. This is the latching edge for both the master and slave.

When the third edge occurs, the value previously latched from the serial data input pin is shifted into the LSB or MSB of the SPI shift register, depending on LSBFE bit. After this edge, the next bit of the master data is coupled out of the serial data output pin of the master to the serial input pin on the slave.

This process continues for a total of 16 edges on the SCK line with data being latched on even numbered edges and shifting taking place on odd numbered edges.

Data reception is double buffered, data is serially shifted into the SPI shift register during the transfer and is transferred to the parallel SPI Data Register after the last bit is shifted in.

After the 16th SCK edge:

- Data that was previously in the SPI Data Register of the master is now in the data register of the slave, and data that was in the data register of the slave is in the master.
- The SPIF flag bit in SPISR is set indicating that the transfer is complete.

Figure 12-10 shows two clocking variations for CPHA = 1. The diagram may be interpreted as a master or slave timing diagram because the SCK, MISO, and MOSI pins are connected directly between the master and the slave. The MISO signal is the output from the slave, and the MOSI signal is the output from the master. The  $\overline{SS}$  line is the slave select input to the slave. The  $\overline{SS}$  pin of the master must be either high or reconfigured as a general-purpose output not affecting the SPI.

The  $\overline{SS}$  line can remain active low between successive transfers (can be tied low at all times). This format is sometimes preferred in systems having a single fixed master and a single slave that drive the MISO data line.

- Back-to-back transfers in master mode

In master mode, if a transmission has completed and a new data byte is available in the SPI Data Register, this byte is send out immediately without a trailing and minimum idle time.

The SPI interrupt request flag (SPIF) is common to both the master and slave modes. SPIF gets set one half SCK cycle after the last SCK edge.

### NOTE

When PWMSCLA = \$00, PWMSCLA value is considered a full scale value of 256. Clock A is thus divided by 512.

Any value written to this register will cause the scale counter to load the new scale value (PWMSCLA).

Module Base + 0x0008

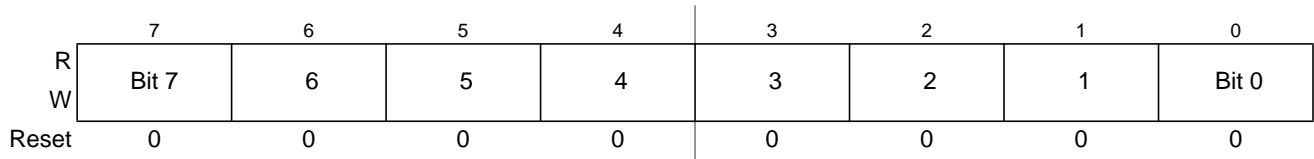


Figure 13-11. PWM Scale A Register (PWMSCLA)

Read: Anytime

Write: Anytime (causes the scale counter to load the PWMSCLA value)

### 13.3.2.10 PWM Scale B Register (PWMSCLB)

PWMSCLB is the programmable scale value used in scaling clock B to generate clock SB. Clock SB is generated by taking clock B, dividing it by the value in the PWMSCLB register and dividing that by two.

$$\text{Clock SB} = \text{Clock B} / (2 * \text{PWMSCLB})$$

### NOTE

When PWMSCLB = \$00, PWMSCLB value is considered a full scale value of 256. Clock B is thus divided by 512.

Any value written to this register will cause the scale counter to load the new scale value (PWMSCLB).

Module Base + 0x0009

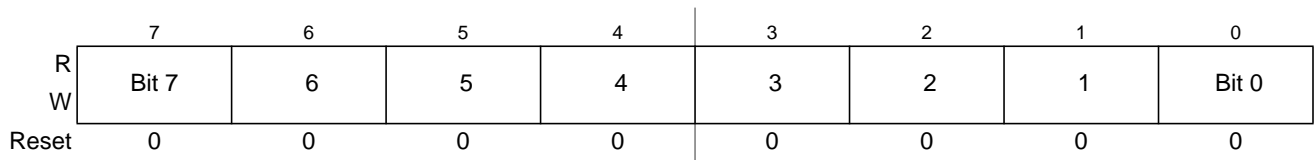


Figure 13-12. PWM Scale B Register (PWMSCLB)

Read: Anytime

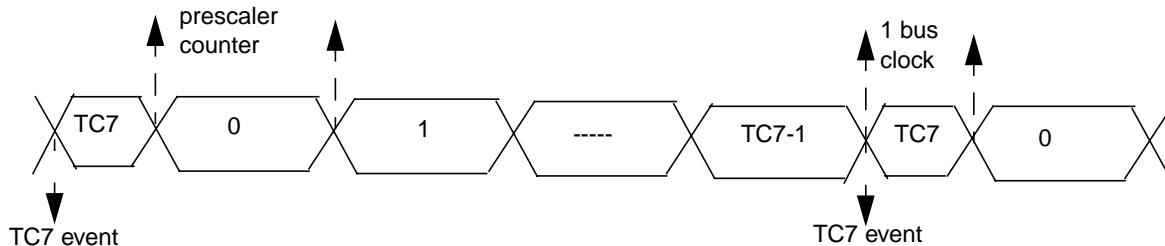
Write: Anytime (causes the scale counter to load the PWMSCLB value).

### 13.3.2.11 Reserved Registers (PWMSCNTx)

The registers PWMSCNTA and PWMSCNTB are reserved for factory testing of the PWM module and are not available in normal modes.

Note: in Figure 14-29, if PR[2:0] is equal to 0, one prescaler counter equal to one bus clock

**Figure 14-29. The TCNT cycle diagram under TCRE=1 condition**



### 14.4.4 Pulse Accumulator

The pulse accumulator (PACNT) is a 16-bit counter that can operate in two modes:

Event counter mode — Counting edges of selected polarity on the pulse accumulator input pin, PAI.

Gated time accumulation mode — Counting pulses from a divide-by-64 clock. The PAMOD bit selects the mode of operation.

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

### 14.4.5 Event Counter Mode

Clearing the PAMOD bit configures the PACNT for event counter operation. An active edge on the IOC7 pin increments the pulse accumulator counter. The PEDGE bit selects falling edges or rising edges to increment the count.

#### NOTE

The PACNT input and timer channel 7 use the same pin IOC7. To use the IOC7, disconnect it from the output logic by clearing the channel 7 output mode and output level bits, OM7 and OL7. Also clear the channel 7 output compare 7 mask bit, OC7M7.

The Pulse Accumulator counter register reflect the number of active input edges on the PACNT input pin since the last reset.

The PAOVF bit is set when the accumulator rolls over from 0xFFFF to 0x0000. The pulse accumulator overflow interrupt enable bit, PAOVI, enables the PAOVF flag to generate interrupt requests.

#### NOTE

The pulse accumulator counter can operate in event counter mode even when the timer enable bit, TEN, is clear.

The commands are described as follows:

- **ACK\_ENABLE** — enables the hardware handshake protocol. The target will issue the ACK pulse when a CPU command is executed by the CPU. The ACK\_ENABLE command itself also has the ACK pulse as a response.
- **ACK\_DISABLE** — disables the ACK pulse protocol. In this case, the host needs to use the worst case delay time at the appropriate places in the protocol.

The default state of the BDM after reset is hardware handshake protocol disabled.

All the read commands will ACK (if enabled) when the data bus cycle has completed and the data is then ready for reading out by the BKGD serial pin. All the write commands will ACK (if enabled) after the data has been received by the BDM through the BKGD serial pin and when the data bus cycle is complete. See Section 16.4.3, “BDM Hardware Commands,” and Section 16.4.4, “Standard BDM Firmware Commands,” for more information on the BDM commands.

The ACK\_ENABLE sends an ACK pulse when the command has been completed. This feature could be used by the host to evaluate if the target supports the hardware handshake protocol. If an ACK pulse is issued in response to this command, the host knows that the target supports the hardware handshake protocol. If the target does not support the hardware handshake protocol the ACK pulse is not issued. In this case, the ACK\_ENABLE command is ignored by the target because it is not recognized as a valid command.

The BACKGROUND command will issue an ACK pulse when the CPU changes from normal to background mode. The ACK pulse related to this command could be aborted using the SYNC command.

The GO command will issue an ACK pulse when the CPU exits from background mode. The ACK pulse related to this command could be aborted using the SYNC command.

The GO\_UNTIL command is equivalent to a GO command with exception that the ACK pulse, in this case, is issued when the CPU enters into background mode. This command is an alternative to the GO command and should be used when the host wants to trace if a breakpoint match occurs and causes the CPU to enter active background mode. Note that the ACK is issued whenever the CPU enters BDM, which could be caused by a breakpoint match or by a BGND instruction being executed. The ACK pulse related to this command could be aborted using the SYNC command.

The TRACE1 command has the related ACK pulse issued when the CPU enters background active mode after one instruction of the application program is executed. The ACK pulse related to this command could be aborted using the SYNC command.

The TAGGO command will not issue an ACK pulse because this would interfere with the tagging function shared on the same pin.

If an interrupt is pending when a TRACE1 command is issued, the interrupt stacking operation occurs but no user instruction is executed. Upon return to standard BDM firmware execution, the program counter points to the first instruction in the interrupt service routine.

## 16.4.11 Instruction Tagging

The instruction queue and cycle-by-cycle CPU activity are reconstructible in real time or from trace history that is captured by a logic analyzer. However, the reconstructed queue cannot be used to stop the CPU at a specific instruction. This is because execution already has begun by the time an operation is visible outside the system. A separate instruction tagging mechanism is provided for this purpose.

The tag follows program information as it advances through the instruction queue. When a tagged instruction reaches the head of the queue, the CPU enters active BDM rather than executing the instruction.

### NOTE

Tagging is disabled when BDM becomes active and BDM serial commands are not processed while tagging is active.

Executing the BDM TAGGO command configures two system pins for tagging. The  $\overline{\text{TAGLO}}$  signal shares a pin with the  $\overline{\text{LSTRB}}$  signal, and the  $\overline{\text{TAGHI}}$  signal shares a pin with the BKGD signal.

Table 16-7 shows the functions of the two tagging pins. The pins operate independently, that is the state of one pin does not affect the function of the other. The presence of logic level 0 on either pin at the fall of the external clock (ECLK) performs the indicated function. High tagging is allowed in all modes. Low tagging is allowed only when low strobe is enabled (LSTRB is allowed only in wide expanded modes and emulation expanded narrow mode).

**Table 16-7. Tag Pin Function**

$\overline{\text{TAGHI}}$	$\overline{\text{TAGLO}}$	Tag
1	1	No tag
1	0	Low byte
0	1	High byte
0	0	Both bytes

## 16.4.12 Serial Communication Time-Out

The host initiates a host-to-target serial transmission by generating a falling edge on the BKGD pin. If BKGD is kept low for more than 128 target clock cycles, the target understands that a SYNC command was issued. In this case, the target will keep waiting for a rising edge on BKGD in order to answer the SYNC request pulse. If the rising edge is not detected, the target will keep waiting forever without any time-out limit.

Consider now the case where the host returns BKGD to logic one before 128 cycles. This is interpreted as a valid bit transmission, and not as a SYNC request. The target will keep waiting for another falling edge marking the start of a new bit. If, however, a new falling edge is not detected by the target within 512 clock cycles since the last falling edge, a time-out occurs and the current command is discarded without affecting memory or the operating mode of the MCU. This is referred to as a soft-reset.

- <sup>1</sup> The DBG module is designed for backwards compatibility to existing BKP modules. Register and bit names have changed from the BKP module. This column shows the DBG register name, as well as the BKP register name for reference.
- <sup>2</sup> Comparator C can be used to enhance the BKP mode by providing a third breakpoint.

### 17.3.2.1 Debug Control Register 1 (DBGC1)

#### NOTE

All bits are used in DBG mode only.

Module Base + 0x0020

Starting address location affected by INITRG register setting.

	7	6	5	4	3	2	1	0
R	DBGEN	ARM	TRGSEL	BEGIN	DBGBRK	0	CAPMOD	
W								
Reset	0	0	0	0	0	0	0	0


 = Unimplemented or Reserved

Figure 17-4. Debug Control Register (DBGC1)

#### NOTE

This register cannot be written if BKP mode is enabled (BKABEN in DBGC2 is set).

Table 17-3. DBGC1 Field Descriptions

Field	Description
7 DBGEN	<b>DBG Mode Enable Bit</b> — The DBGEN bit enables the DBG module for use in DBG mode. This bit cannot be set if the MCU is in secure mode. 0 DBG mode disabled 1 DBG mode enabled
6 ARM	<b>Arm Bit</b> — The ARM bit controls whether the debugger is comparing and storing data in the trace buffer. See Section 17.4.2.4, “Arming the DBG Module,” for more information. 0 Debugger unarmed 1 Debugger armed <b>Note:</b> This bit cannot be set if the DBGEN bit is not also being set at the same time. For example, a write of 01 to DBGEN[7:6] will be interpreted as a write of 00.
5 TRGSEL	<b>Trigger Selection Bit</b> — The TRGSEL bit controls the triggering condition for comparators A and B in DBG mode. It serves essentially the same function as the TAGAB bit in the DBGC2 register does in BKP mode. See Section 17.4.2.1.2, “Trigger Selection,” for more information. TRGSEL may also determine the type of breakpoint based on comparator A and B if enabled in DBG mode (DBGBRK = 1). Please refer to Section 17.4.3.1, “Breakpoint Based on Comparator A and B.” 0 Trigger on any compare address match 1 Trigger before opcode at compare address gets executed (tagged-type)



### 17.3.2.5 Debug Comparator C Extended Register (DBGCCX)

Module Base + 0x0025

Starting address location affected by INITRG register setting.

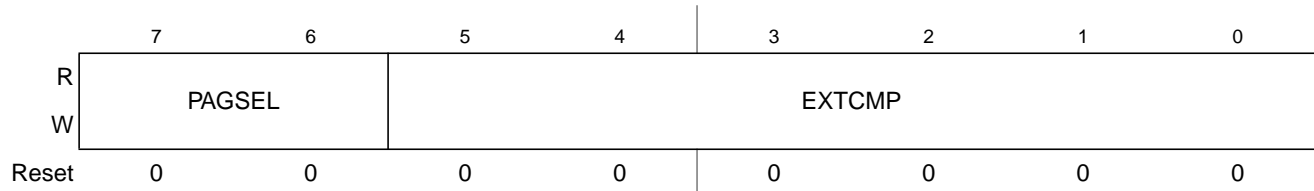


Figure 17-9. Debug Comparator C Extended Register (DBGCCX)

Table 17-10. DBGCCX Field Descriptions

Field	Description
7:6 PAGSEL	<b>Page Selector Field</b> — In both BKP and DBG mode, PAGSEL selects the type of paging as shown in Table 17-11. DPAGE and EPAGE are not yet implemented so the value in bit 7 will be ignored (i.e., PAGSEL values of 10 and 11 will be interpreted as values of 00 and 01, respectively).
5:0 EXTCMP	<b>Comparator C Extended Compare Bits</b> — The EXTCMP bits are used as comparison address bits as shown in Table 17-11 along with the appropriate PPAGE, DPAGE, or EPAGE signal from the core. <b>Note:</b> Comparator C can be used when the DBG module is configured for BKP mode. Extended addressing comparisons for comparator C use PAGSEL and will operate differently to the way that comparator A and B operate in BKP mode.

Table 17-11. PAGSEL Decoding<sup>1</sup>

PAGSEL	Description	EXTCMP	Comment
00	Normal (64k)	Not used	No paged memory
01	PPAGE (256 — 16K pages)	EXTCMP[5:0] is compared to address bits [21:16] <sup>2</sup>	PPAGE[7:0] / XAB[21:14] becomes address bits [21:14] <sup>1</sup>
10 <sup>3</sup>	DPAGE (reserved) (256 — 4K pages)	EXTCMP[3:0] is compared to address bits [19:16]	DPAGE / XAB[21:14] becomes address bits [19:12]
11 <sup>2</sup>	EPAGE (reserved) (256 — 1K pages)	EXTCMP[1:0] is compared to address bits [17:16]	EPAGE / XAB[21:14] becomes address bits [17:10]

<sup>1</sup> See Figure 17-10.

<sup>2</sup> Current HCS12 implementations have PPAGE limited to 6 bits. Therefore, EXTCMP[5:4] should be set to 00.

<sup>3</sup> Data page (DPAGE) and Extra page (EPAGE) are reserved for implementation on devices that support paged data and extra space.

least six addresses higher than address A (or B is lower than A) and there are not changes of flow to put these in the queue at the same time, then this operation should trigger properly.

#### 17.4.2.5.4 Event-Only B (Store Data)

In the event-only B trigger mode, if the match condition for B is met, the B flag in DBGSC is set and a trigger occurs. The event-only B trigger mode is considered a begin-trigger type and the BEGIN bit in DBGSC1 is ignored. Event-only B is incompatible with instruction tagging (TRGSEL = 1), and thus the value of TRGSEL is ignored. Please refer to Section 17.4.2.7, “Storage Memory,” for more information.

This trigger mode is incompatible with the detail capture mode so the detail capture mode will have priority. TRGSEL and BEGIN will not be ignored and this trigger mode will behave as if it were “B only”.

#### 17.4.2.5.5 A then Event-Only B (Store Data)

In the A then event-only B trigger mode, the match condition for A must be met before the match condition for B is compared, after the A match has occurred, a trigger occurs each time B matches. When the match condition for A or B is met, the corresponding flag in DBGSC is set. The A then event-only B trigger mode is considered a begin-trigger type and BEGIN in DBGSC1 is ignored. TRGSEL in DBGSC1 applies only to the match condition for A. Please refer to Section 17.4.2.7, “Storage Memory,” for more information.

This trigger mode is incompatible with the detail capture mode so the detail capture mode will have priority. TRGSEL and BEGIN will not be ignored and this trigger mode will be the same as A then B.

#### 17.4.2.5.6 A and B (Full Mode)

In the A and B trigger mode, comparator A compares to the address bus and comparator B compares to the data bus. In the A and B trigger mode, if the match condition for A and B happen on the same bus cycle, both the A and B flags in the DBGSC register are set and a trigger occurs.

If TRGSEL = 1, only matches from comparator A are used to determine if the trigger condition is met and comparator B matches are ignored. If TRGSEL = 0, full-word data matches on an odd address boundary (misaligned access) do not work unless the access is to a RAM that manages misaligned accesses in a single clock cycle (which is typical of RAM modules used in HCS12 MCUs).

#### 17.4.2.5.7 A and Not B (Full Mode)

In the A and not B trigger mode, comparator A compares to the address bus and comparator B compares to the data bus. In the A and not B trigger mode, if the match condition for A and not B happen on the same bus cycle, both the A and B flags in DBGSC are set and a trigger occurs.

If TRGSEL = 1, only matches from comparator A are used to determine if the trigger condition is met and comparator B matches are ignored. As described in Section 17.4.2.5.6, “A and B (Full Mode),” full-word data compares on misaligned accesses will not match expected data (and thus will cause a trigger in this mode) unless the access is to a RAM that manages misaligned accesses in a single clock cycle.

Table 17-26. Breakpoint Setup

BEGIN	TRGSEL	DBGBRK	Type of Debug Run
0	0	0	Fill trace buffer until trigger address (no CPU breakpoint — keep running)
0	0	1	Fill trace buffer until trigger address, then a forced breakpoint request occurs
0	1	0	Fill trace buffer until trigger opcode is about to execute (no CPU breakpoint — keep running)
0	1	1	Fill trace buffer until trigger opcode about to execute, then a tagged breakpoint request occurs
1	0	0	Start trace buffer at trigger address (no CPU breakpoint — keep running)
1	0	1	Start trace buffer at trigger address, a forced breakpoint request occurs when trace buffer is full
1	1	0	Start trace buffer at trigger opcode (no CPU breakpoint — keep running)
1	1	1	Start trace buffer at trigger opcode, a forced breakpoint request occurs when trace buffer is full

### 17.4.3.2 Breakpoint Based on Comparator C

A breakpoint request to the CPU can be created if BKCEN in DBG2 is set. Breakpoints based on a successful comparator C match can be accomplished regardless of the mode of operation for comparator A or B, and do not affect the status of the ARM bit. TAGC in DBG2 is used to select either tagged or forced breakpoint requests for comparator C. Breakpoints based on comparator C are disabled in LOOP1 mode.

#### NOTE

Because breakpoints cannot be disabled when the DBG is armed, one must be careful to avoid an “infinite breakpoint loop” when using tagged-type C breakpoints while the DBG is armed. If BDM breakpoints are selected, executing a TRACE1 instruction before the GO instruction is the recommended way to avoid re-triggering a breakpoint if one does not wish to de-arm the DBG. If SWI breakpoints are selected, disarming the DBG in the SWI interrupt service routine is the recommended way to avoid re-triggering a breakpoint.

## 17.5 Resets

The DBG module is disabled after reset.

The DBG module cannot cause a MCU reset.

## 17.6 Interrupts

The DBG contains one interrupt source. If a breakpoint is requested and BDM in DBG2 is cleared, an SWI interrupt will be generated.

These register locations are not used (reserved). All unused registers and bits in this block return logic 0s when read. Writes to these registers have no effect.

These registers are not in the on-chip map in special peripheral mode.

### 19.3.2.6 Port E Data Register (PORTE)

Module Base + 0x0008

Starting address location affected by INITRG register setting.

	7	6	5	4	3	2	1	0
R	Bit 7	6	5	4	3	2	Bit 1	Bit 0
W								
Reset	0	0	0	0	0	0	u	u
Alternate Pin Function	NOACC	MODB or IPIPE1 or CLKTO	MODA or IPIPE0	ECLK	$\overline{\text{LSTRB}}$ or $\overline{\text{TAGLO}}$	R/ $\overline{\text{W}}$	$\overline{\text{IRQ}}$	$\overline{\text{XIRQ}}$

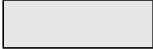
 = Unimplemented or Reserved      u = Unaffected by reset

Figure 19-10. Port E Data Register (PORTE)

Read: Anytime when register is in the map

Write: Anytime when register is in the map

Port E is associated with external bus control signals and interrupt inputs. These include mode select (MODB/IPIPE1, MODA/IPIPE0), E clock, size ( $\overline{\text{LSTRB}}$ / $\overline{\text{TAGLO}}$ ), read/write (R/ $\overline{\text{W}}$ ),  $\overline{\text{IRQ}}$ , and  $\overline{\text{XIRQ}}$ . When not used for one of these specific functions, port E pins 7:2 can be used as general-purpose I/O and pins 1:0 can be used as general-purpose input. The port E assignment register (PEAR) selects the function of each pin and DDRE determines whether each pin is an input or output when it is configured to be general-purpose I/O. DDRE also determines the source of data for a read of PORTE.

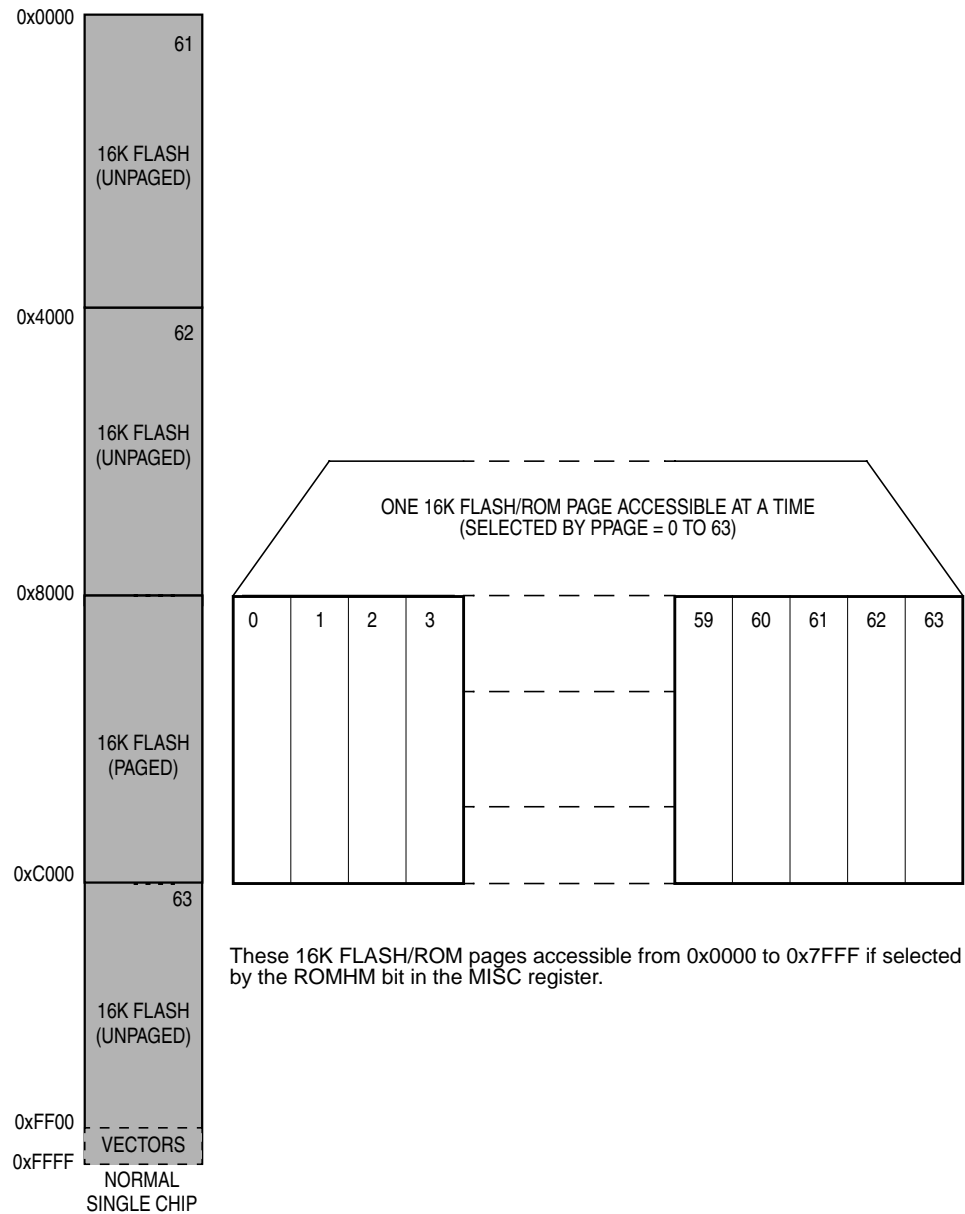
Some of these pins have software selectable pull resistors.  $\overline{\text{IRQ}}$  and  $\overline{\text{XIRQ}}$  can only be pulled up whereas the polarity of the PE7, PE4, PE3, and PE2 pull resistors are determined by chip integration. Please refer to the device overview chapter (Signal Property Summary) to determine the polarity of these resistors. A single control bit enables the pull devices for all of these pins when they are configured as inputs.

This register is not in the on-chip map in special peripheral mode or in expanded modes when the EME bit is set. Therefore, these accesses will be echoed externally.

#### NOTE

It is unwise to write PORTE and DDRE as a word access. If you are changing port E pins from being inputs to outputs, the data may have extra transitions during the write. It is best to initialize PORTE before enabling as outputs.

A graphical example of a memory paging for a system configured as 1M byte on-chip FLASH/ROM with 64K allocated physical space is given in Figure 20-12.



**Figure 20-12. Memory Paging Example: 1M Byte On-Chip FLASH/ROM, 64K Allocation**