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, I²C, 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 ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12kg256mfue

Chapter 3
256 Kbyte ECC Flash Module (S12FTS256K2ECCV2)

Chapter 4
4 Kbyte EEPROM Module (S12EETS4KV2)

0x0020–0x002F DBG Map 1 of 1 (HCS12 Debug) (continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x002C	DBGCAL BKP0L	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x002D	DBGCBX BKP1X	R W	PAGSEL		EXTCMP					
0x002E	DBGCBH BKP1H	R W	Bit 15	14	13	12	11	10	9	Bit 8
0x002F	DBGCBL BKP1L	R W	Bit 7	6	5	4	3	2	1	Bit 0

0x0030–0x0031 MMC Map 4 of 4 (HCS12 Module Mapping Control)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0030	PPAGE	R W	0	0	PIX5	PIX4	PIX3	PIX2	PIX1	PIX0
0x0031	Reserved	R W	0	0	0	0	0	0	0	0

0x0032–0x0033 MEBI Map 3 of 3 (HCS12 Multiplexed External Bus Interface)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0032	PORTK	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x0033	DDRK	R W	Bit 7	6	5	4	3	2	1	Bit 0

0x0034–0x003F CRG (Clock and Reset Generator)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0034	SYNR	R W	0	0	SYN5	SYN4	SYN3	SYN2	SYN1	SYN0
0x0035	REFDV	R W	0	0	0	0	REFDV3	REFDV2	REFDV1	REFDV0
0x0036	CTFLG TEST ONLY	R W	TOUT7	TOUT6	TOUT5	TOUT4	TOUT3	TOUT2	TOUT1	TOUT0
0x0037	CRGFLG	R W	RTIF	PROF	0	LOCKIF	LOCK	TRACK	SCMIF	SCM
0x0038	CRGINT	R W	RTIE	0	0	LOCKIE	0	0	SCMIE	0
0x0039	CLKSEL	R W	PLLSEL	PSTP	SYSWAI	ROAWAI	PLLWAI	CWAI	RTIWAI	COPWAI
0x003A	PLLCTL	R W	CME	PLLON	AUTO	ACQ	0	PRE	PCE	SCME
0x003B	RTICTL	R W	0	RTR6	RTR5	RTR4	RTR3	RTR2	RTR1	RTR0

0x0040–0x006FTIM (Timer 16 Bit 8 Channels) (Sheet 3 of 3)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0067	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0068	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0069	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x006A	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x006B	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x006C	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x006D	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x006E	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x006F	Reserved	R	0	0	0	0	0	0	0	0
		W								

0x0070–0x007FReserved Space

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0070–0x007F	Reserved	R	0	0	0	0	0	0	0	0
		W								

0x0080–0x009F ATD0 (Analog to Digital Converter 10 Bit 8 Channel)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0080	ATD0CTL0	R	0	0	0	0	0	WRAP2	WRAP1	WRAP0
		W								
0x0081	ATD0CTL1	R	ETRIGSEL	0	0	0	0	ETRIGCH2	ETRIGCH1	ETRIGCH0
		W								
0x0082	ATD0CTL2	R	ADPU	AFFC	AWAI	ETRIGLE	ETRIGP	ETRIG	ASCIE	ASCIF
		W								
0x0083	ATD0CTL3	R	0	S8C	S4C	S2C	S1C	FIFO	FRZ1	FRZ0
		W								
0x0084	ATD0CTL4	R	SRES8	SMP1	SMP0	PRS4	PRS3	PRS2	PRS1	PRS0
		W								
0x0085	ATD0CTL5	R	DJM	DSGN	SCAN	MULT	0	CC	CB	CA
		W								
0x0086	ATD0STAT0	R	SCF	0	ETORF	FIFOR	0	CC2	CC1	CC0
		W								
0x0087	Reserved	R	0	0	0	0	0	0	0	0
		W								

1.6.2 Resets

Resets are a subset of the interrupts featured inTable 1-12. The different sources capable of generating a system reset are summarized in Table 1-13.

Table 1-13. Reset Summary

Reset	Priority	Source	Vector
Power-on Reset	1	CRG Module	0xFFFE, 0xFFFF
External Reset	1	RESET pin	0xFFFE, 0xFFFF
Low Voltage Reset	1	VREG Module	0xFFFE, 0xFFFF
Clock Monitor Reset	2	CRG Module	0xFFFC, 0xFFFD
COP Watchdog Reset	3	CRG Module	0xFFFA, 0xFFFB

1.6.2.1 Effects of Reset

When a reset occurs, MCU registers and control bits are changed to known start-up states. Refer to the respective module block description chapters for register reset states.

Refer to the PIM block description chapter for reset configurations of all peripheral module ports.

Refer to Table 1-5 for locations of the memories depending on the operating mode after reset.

The RAM array is not automatically initialized out of reset.

2.4.1.3.5 Mass Erase Command

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

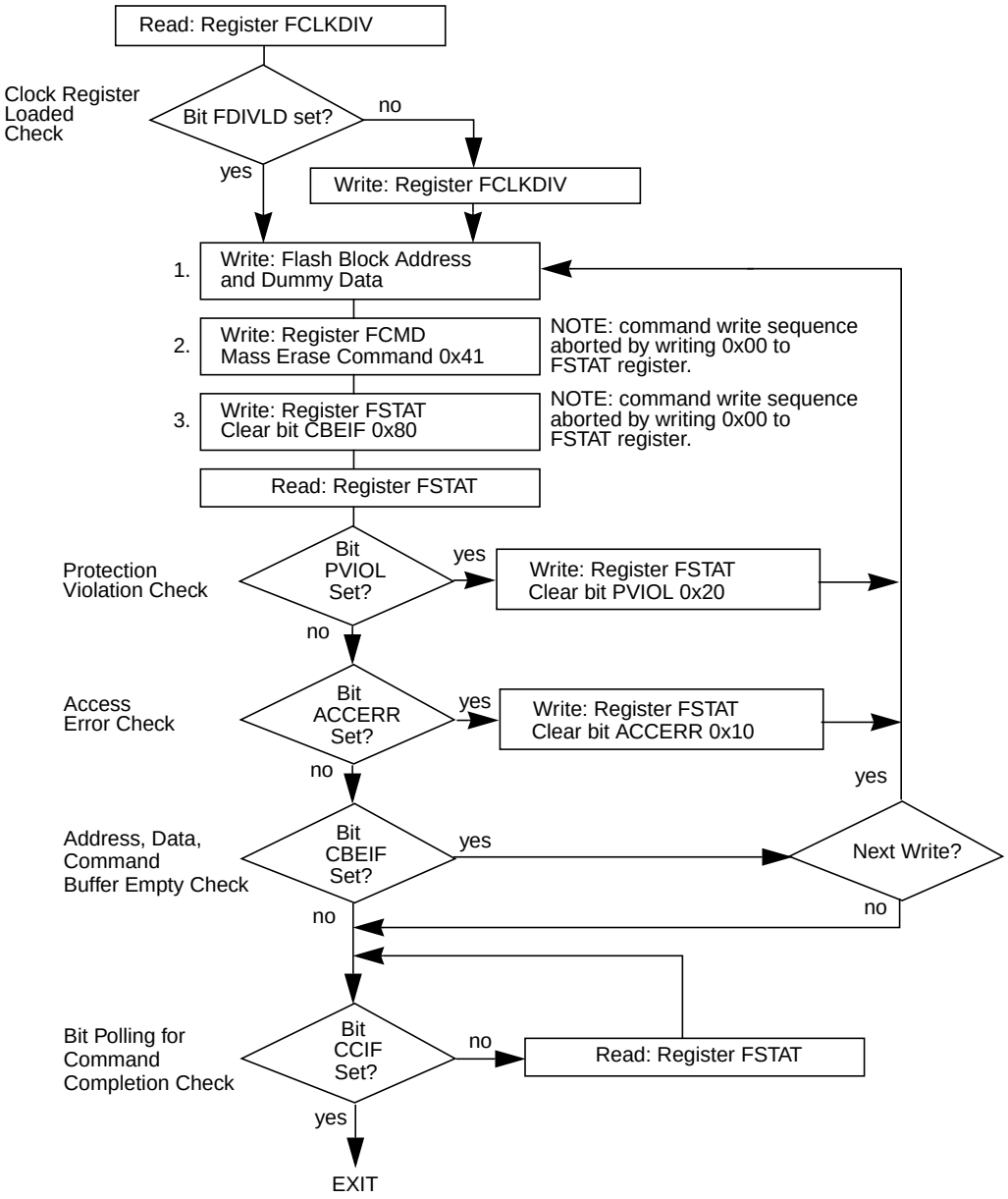

Figure 2-29. Example Mass Erase Command Flow

2.6 Flash Module Security

The Flash module provides the necessary security information to the MCU. After each reset, the Flash module determines the security state of the MCU as defined in Section 2.3.2.2, “Flash Security Register (FSEC)”.

The contents of the Flash security byte at 0xFF0F in the Flash configuration field must be changed directly by programming 0xFF0F when the MCU is unsecured and the higher address sector is unprotected. If the Flash security byte remains in a secured state, any reset will cause the MCU to initialize to a secure operating mode.

2.6.1 Unsecuring the MCU using Backdoor Key Access

The MCU may be unsecured by using the backdoor key access feature which requires knowledge of the contents of the backdoor keys (four 16-bit words programmed at addresses 0xFF00–0xFF07). If the KEYEN[1:0] bits are in the enabled state (see Section 2.3.2.2, “Flash Security Register (FSEC)”) and the KEYACC bit is set, a write to a backdoor key address in the Flash memory triggers a comparison between the written data and the backdoor key data stored in the Flash memory. If all four words of data are written to the correct addresses in the correct order and the data matches the backdoor keys stored in the Flash memory, the MCU will be unsecured. The data must be written to the backdoor keys sequentially starting with 0xFF00–0xFF01 and ending with 0xFF06–0xFF07. 0x0000 and 0xFFFF are not permitted as backdoor keys. While the KEYACC bit is set, reads of the Flash memory will return invalid data.

The user code stored in the Flash memory must have a method of receiving the backdoor key from an external stimulus. This external stimulus would typically be through one of the on-chip serial ports.

If the KEYEN[1:0] bits are in the enabled state (see Section 2.3.2.2, “Flash Security Register (FSEC)”), the MCU can be unsecured by the backdoor access sequence described below:

1. Set the KEYACC bit in the Flash configuration register (FCNFG).
2. Write the correct four 16-bit words to Flash addresses 0xFF00–0xFF07 sequentially starting with 0xFF00.
3. Clear the KEYACC bit.
4. If all four 16-bit words match the backdoor keys stored in Flash addresses 0xFF00–0xFF07, the MCU is unsecured and the SEC[1:0] bits in the FSEC register are forced to the unsecure state of 1:0.

The backdoor key access sequence is monitored by an internal security state machine. An illegal operation during the backdoor key access sequence will cause the security state machine to lock, leaving the MCU in the secured state. A reset of the MCU will cause the security state machine to exit the lock state and allow a new backdoor key access sequence to be attempted. The following operations during the backdoor key access sequence will lock the security state machine:

1. If any of the four 16-bit words does not match the backdoor keys programmed in the Flash array. Double bit faults detected while reading the backdoor keys from the Flash array are ignored.
2. If the four 16-bit words are written in the wrong sequence.
3. If more than four 16-bit words are written.
4. If any of the four 16-bit words written are 0x0000 or 0xFFFF.

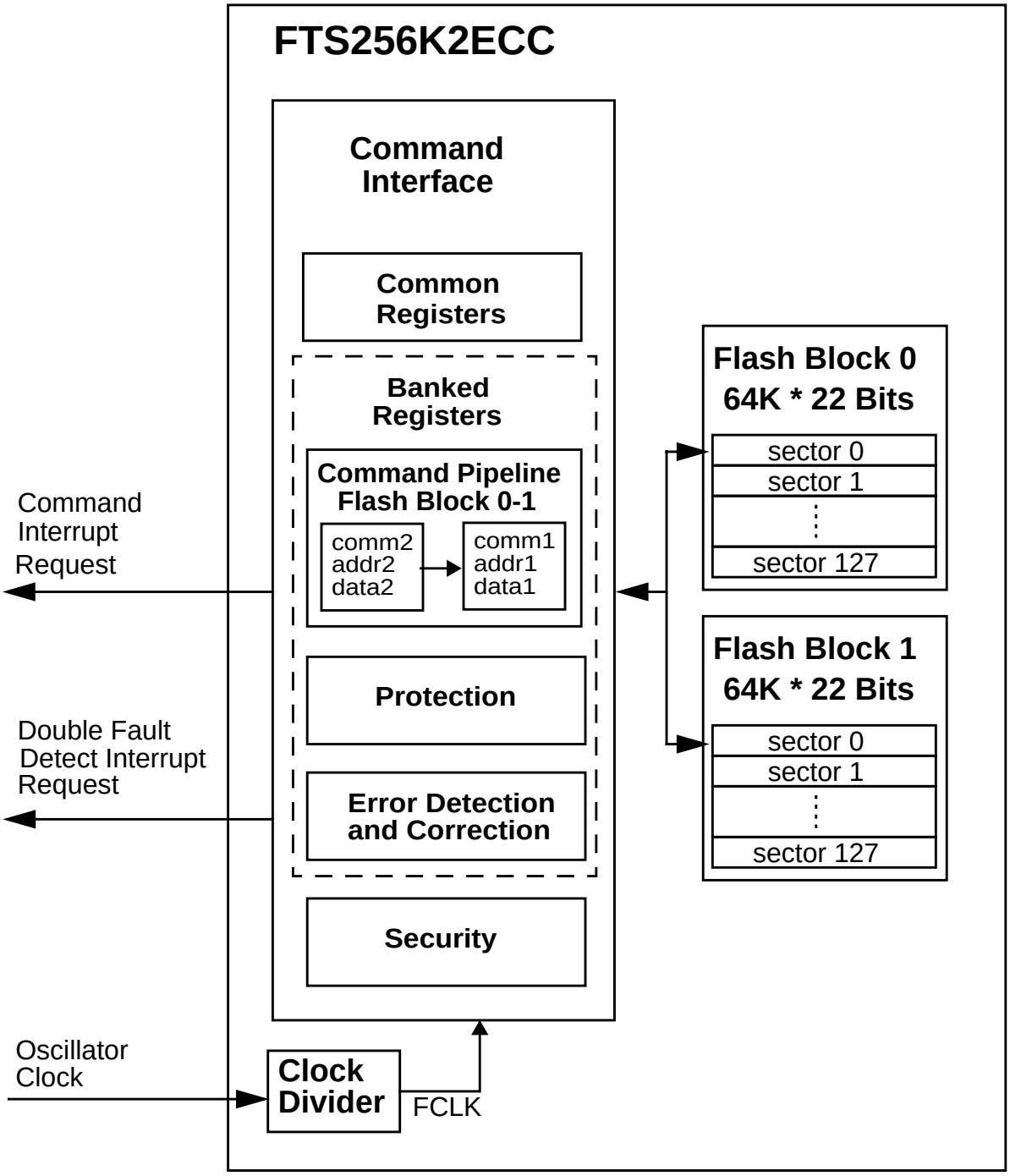

Figure 3-1. FTS256K2ECC Block Diagram

3.2 External Signal Description

The Flash module contains no signals that connect off-chip.

3.3.2.2 Flash Security Register (FSEC)

The unbanked FSEC register holds all bits associated with the security of the MCU and Flash module.

Module Base + 0x0001

	7	6	5	4	3	2	1	0
R	KEYEN		RNV5	RNV4	RNV3	RNV2	SEC	
W								
Reset	F	F	F	F	F	F	F	F

= Unimplemented or Reserved

Figure 3-5. Flash Security Register (FSEC)

All bits in the FSEC register are readable but are not writable.

The FSEC register is loaded from the Flash Configuration Field at address 0xFF0F during the reset sequence, indicated by F in Figure 3-5. If the DFDIF flag in the FSTAT register is set while reading the security field location during the reset sequence, all bits in the FSEC register will be set to leave the module in a secured state with backdoor key access disabled.

Table 3-6. FSEC Field Descriptions

Field	Description
1-0 KEYEN[1:0]	**Backdoor Key Security Enable Bits** —The KEYEN[1:0] bits define the enabling of backdoor key access to the Flash module as shown in Table 3-7.
5-2 RNV[5:2]	**Reserved Nonvolatile Bits** — The RNV[5:2] bits must remain in the erased 1 state for future enhancements.
1-0 SEC[1:0]	**Flash Security Bits** — The SEC[1:0] bits define the security state of the MCU as shown in Table 3-8. If the Flash module is unsecured using backdoor key access, the SEC bits are forced to 1:0.

Table 3-7. Flash KEYEN States

KEYEN[1:0]	Status of Backdoor Key Access
00	DISABLED
01[1]	DISABLED
10	**ENABLED**
11	DISABLED

1 Preferred KEYEN state to disable Backdoor Key Access.

3.6 Flash Module Security

The Flash module provides the necessary security information to the MCU. After each reset, the Flash module determines the security state of the MCU as defined in Section 3.3.2.2, “Flash Security Register (FSEC)”.

The contents of the Flash security byte at 0xFF0F in the Flash configuration field must be changed directly by programming 0xFF0F when the MCU is unsecured and the higher address sector is unprotected. If the Flash security byte remains in a secured state, any reset will cause the MCU to initialize to a secure operating mode.

3.6.1 Unsecuring the MCU using Backdoor Key Access

The MCU may be unsecured by using the backdoor key access feature which requires knowledge of the contents of the backdoor keys (four 16-bit words programmed at addresses 0xFF00–0xFF07). If the KEYEN[1:0] bits are in the enabled state (see Section 3.3.2.2, “Flash Security Register (FSEC)”) and the KEYACC bit is set, a write to a backdoor key address in the Flash memory triggers a comparison between the written data and the backdoor key data stored in the Flash memory. If all four words of data are written to the correct addresses in the correct order and the data matches the backdoor keys stored in the Flash memory, the MCU will be unsecured. The data must be written to the backdoor keys sequentially starting with 0xFF00–0xFF01 and ending with 0xFF06–0xFF07. 0x0000 and 0xFFFF are not permitted as backdoor keys. While the KEYACC bit is set, reads of the Flash memory will return invalid data.

The user code stored in the Flash memory must have a method of receiving the backdoor key from an external stimulus. This external stimulus would typically be through one of the on-chip serial ports.

If the KEYEN[1:0] bits are in the enabled state (see Section 3.3.2.2, “Flash Security Register (FSEC)”), the MCU can be unsecured by the backdoor access sequence described below:

1. Set the KEYACC bit in the Flash configuration register (FCNFG).
2. Write the correct four 16-bit words to Flash addresses 0xFF00–0xFF07 sequentially starting with 0xFF00.
3. Clear the KEYACC bit.
4. If all four 16-bit words match the backdoor keys stored in Flash addresses 0xFF00–0xFF07, the MCU is unsecured and the SEC[1:0] bits in the FSEC register are forced to the unsecure state of 1:0.

The backdoor key access sequence is monitored by an internal security state machine. An illegal operation during the backdoor key access sequence will cause the security state machine to lock, leaving the MCU in the secured state. A reset of the MCU will cause the security state machine to exit the lock state and allow a new backdoor key access sequence to be attempted. The following operations during the backdoor key access sequence will lock the security state machine:

1. If any of the four 16-bit words does not match the backdoor keys programmed in the Flash array. Double bit faults detected while reading the backdoor keys from the Flash array are ignored.
2. If the four 16-bit words are written in the wrong sequence.
3. If more than four 16-bit words are written.
4. If any of the four 16-bit words written are 0x0000 or 0xFFFF.

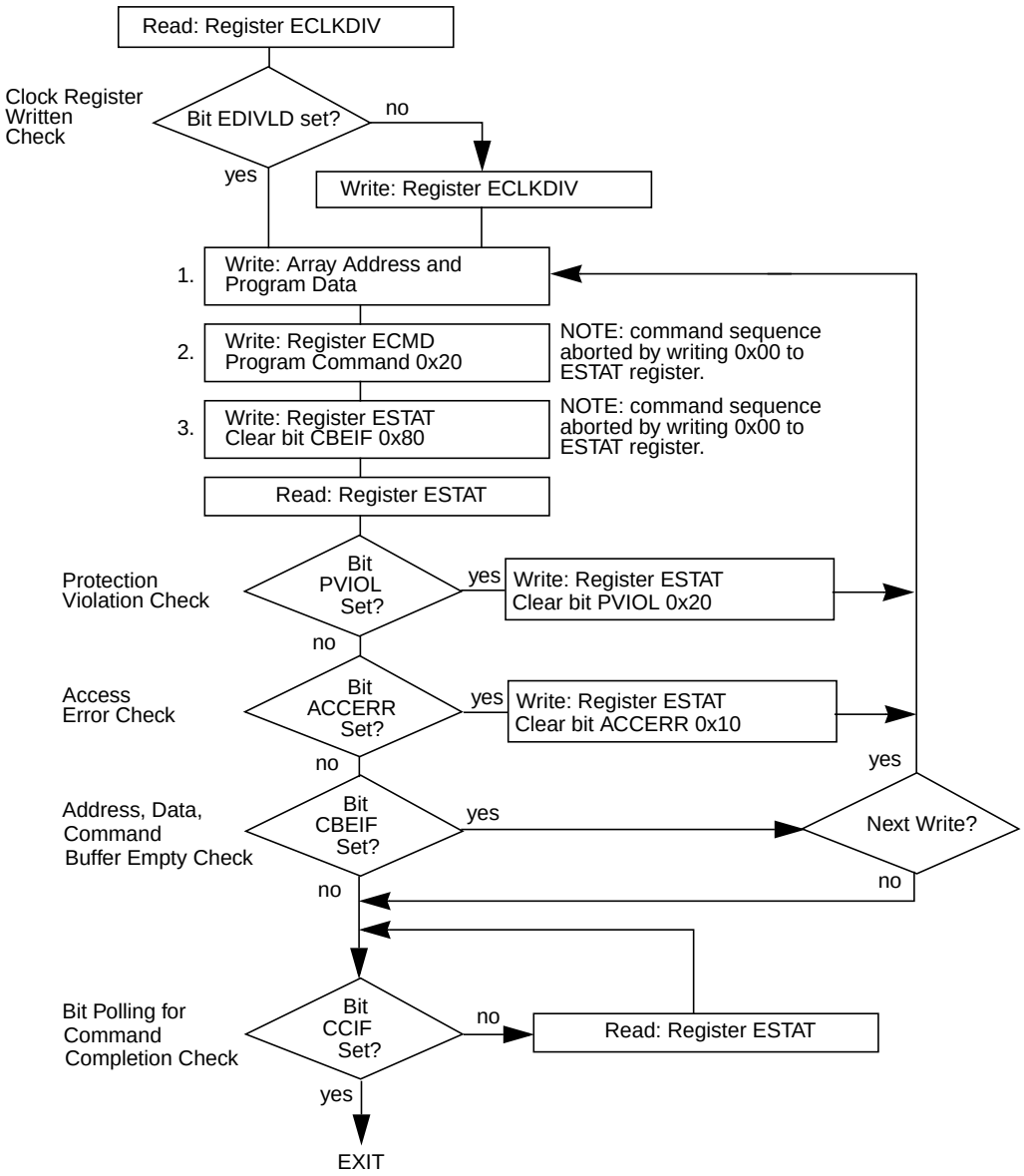

Figure 4-18. Example Program Command Flow

Table 5-1. Pin Functions and Priorities (Sheet 2 of 4)

Port	Pin Name	Pin Function and Priority	Description	Pull Mode after Reset	Pin Function after Reset
Port M	PM7	TXCAN4	MSCAN4 transmit pin	Hi-Z	GPIO
		GPIO	General-purpose I/O		
	PM6	RXCAN4	MSCAN4 receive pin		
		GPIO	General-purpose I/O		
	PM5	TXCAN0	MSCAN0 transmit pin		
		TXCAN4	MSCAN4 transmit pin		
		SCK0	Serial Peripheral Interface 0 serial clock pin		
		GPIO	General-purpose I/O		
	PM4	RXCAN0	MSCAN0 receive pin		
		RXCAN4	MSCAN4 receive pin		
		MOSI0	Serial Peripheral Interface 0 master out/slave in pin		
		GPIO	General-purpose I/O		
	PM3	TXCAN1	MSCAN1 transmit pin		
		TXCAN0	MSCAN0 transmit pin		
		$\overline{SS0}$[1]	Serial Peripheral Interface 0 slave select output in master mode, input for slave mode or master mode.		
		GPIO	General-purpose I/O		
	PM2	RXCAN1	MSCAN1 receive pin		
		RXCAN0	MSCAN0 receive pin		
		MISO0[1]	Serial Peripheral Interface 0 master in/slave out pin		
		GPIO	General-purpose I/O		
	PM1	TXCAN0	MSCAN0 transmit pin		
		GPIO	General-purpose I/O		
	PM0	RXCAN0	MSCAN0 receive pin		
		GPIO	General-purpose I/O		

Table 5-17. MODRR Field Descriptions (continued)

Field	Description
3–2 MODRR[3:2]	**CAN4 Routing Bits** — See Table 5-19.
1–0 MODRR[1:0]	**CAN0 Routing Bits** — See Table 5-18.

Table 5-18. CAN0 Routing

MODRR[1]	MODRR[0]	RXCAN0	TXCAN0
0	0	PM0	PM1
0	1	PM2	PM3
1	0	PM4	PM5
1	1	Reserved	

Table 5-19. CAN4 Routing

MODRR[3]	MODRR[2]	RXCAN4	TXCAN4
0	0	PJ6	PJ7
0	1	PM4	PM5
1	0	PM6	PM7
1	1	Reserved	

Table 5-20. SPI0 Routing

MODRR[4]	MISO0	MOSI0	SCK0	SS0
0	PS4	PS5	PS6	PS7
1	PM2	PM4	PM5	PM3

Table 5-21. SPI1 Routing

MODRR[5]	MISO1	MOSI1	SCK1	SS1
0	PP0	PP1	PP2	PP3
1	PH0	PH1	PH2	PH3

Table 5-22. SPI2 Routing

MODRR[6]	MISO2	MOSI2	SCK2	SS2
0	PP4	PP5	PP6	PP7
1	PH4	PH5	PH6	PH7

Table 8-19. Special Channel Select Coding

SC	CC	CB	CA	Analog Input Channel
1	0	X	X	Reserved
1	1	0	0	V_{RH}
1	1	0	1	V_{RL}
1	1	1	0	$(V_{RH}+V_{RL}) / 2$
1	1	1	1	Reserved

8.3.2.10 ATD Status Register 1 (ATDSTAT1)

This read-only register contains the conversion complete flags.

Module Base + 0x000B

	7	6	5	4	3	2	1	0
R	CCF7	CCF6	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 8-12. ATD Status Register 1 (ATDSTAT1)

Read: Anytime

Write: Anytime, no effect

Table 8-20. ATDSTAT1 Field Descriptions

Field	Description
7–0 CCF[7:0]	**Conversion Complete Flag x (x = 7, 6, 5, 4, 3, 2, 1, 0)** — A conversion complete flag is set at the end of each conversion in a conversion sequence. The flags are associated with the conversion position in a sequence (and also the result register number). Therefore, CCF0 is set when the first conversion in a sequence is complete and the result is available in result register ATDDR0; CCF1 is set when the second conversion in a sequence is complete and the result is available in ATDDR1, and so forth. A flag CCFx (x = 7, 6, 5, 4, 3, 2,1, 70) is cleared when ADPU=1 and one of the following occurs: A) Write to ATDCTL5 (a new conversion sequence is started) B) If AFFC=0 and read of ATDSTAT1 followed by read of result register ATDDRx C) If AFFC=1 and read of result register ATDDRx In case of a concurrent set and clear on CCFx: The clearing by method A) will overwrite the set. The clearing by methods B) or C) will be overwritten by the set. 0 Conversion number x not completed 1 Conversion number x has completed, result ready in ATDDRx

NOTE

Register Address = Base Address + Address Offset, where the Base Address is defined at the MCU level and the Address Offset is defined at the module level.

13.3.2 Register Descriptions

This section describes in detail all the registers and register bits in the PWM module.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 PWME	R	PWME7	PWME6	PWME5	PWME4	PWME3	PWME2	PWME1	PWME0
	W								
0x0001 PWMPOL	R	PPOL7	PPOL6	PPOL5	PPOL4	PPOL3	PPOL2	PPOL1	PPOL0
	W								
0x0002 PWMCLK	R	PCLK7	PCLKL6	PCLK5	PCLK4	PCLK3	PCLK2	PCLK1	PCLK0
	W								
0x0003 PWMPRCLK	R	0	PCKB2	PCKB1	PCKB0	0	PCKA2	PCKA1	PCKA0
	W								
0x0004 PWMCAE	R	CAE7	CAE6	CAE5	CAE4	CAE3	CAE2	CAE1	CAE0
	W								
0x0005 PWMCTL	R	CON67	CON45	CON23	CON01	PSWAI	PFRZ	0	0
	W								
0x0006 PWMTST[1]	R	0	0	0	0	0	0	0	0
	W								
0x0007 PWMPRSC[1]	R	0	0	0	0	0	0	0	0
	W								
0x0008 PWMSCLA	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x0009 PWMSCLB	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x000A PWMSCNTA[1]	R	0	0	0	0	0	0	0	0
	W								

= Unimplemented or Reserved

Figure 13-2. PWM Register Summary (Sheet 1 of 3)

Module Base + 0x0009

	7	6	5	4	3	2	1	0
R W	OM3	OL3	OM2	OL2	OM1	OL1	OM0	OL0
Reset	0	0	0	0	0	0	0	0

Figure 14-15. Timer Control Register 2 (TCTL2)

Read: Anytime

Write: Anytime

Table 14-9. TCTL1/TCTL2 Field Descriptions

Field	Description
7:0 OMx	**Output Mode** — These eight pairs of control bits are encoded to specify the output action to be taken as a result of a successful OCx compare. When either OMx or OLx is 1, the pin associated with OCx becomes an output tied to OCx. **Note:** To enable output action by OMx bits on timer port, the corresponding bit in OC7M should be cleared.
7:0 OLx	**Output Level** — These eight pairs of control bits are encoded to specify the output action to be taken as a result of a successful OCx compare. When either OMx or OLx is 1, the pin associated with OCx becomes an output tied to OCx. **Note:** To enable output action by OLx bits on timer port, the corresponding bit in OC7M should be cleared.

Table 14-10. Compare Result Output Action

OMx	OLx	Action
0	0	Timer disconnected from output pin logic
0	1	Toggle OCx output line
1	0	Clear OCx output line to zero
1	1	Set OCx output line to one

To operate the 16-bit pulse accumulator independently of input capture or output compare 7 and 0 respectively the user must set the corresponding bits IOSx = 1, OMx = 0 and OLx = 0. OC7M7 in the OC7M register must also be cleared.

To enable output action using the OM7 and OL7 bits on the timer port,the corresponding bit OC7M7 in the OC7M register must also be cleared. The settings for these bits can be seen in Table 14-11

Table 14-11. The OC7 and OCx event priority

OC7M7=0				OC7M7=1			
OC7Mx=1		OC7Mx=0		OC7Mx=1		OC7Mx=0	
TC7=TCx	TC7>TCx	TC7=TCx	TC7>TCx	TC7=TCx	TC7>TCx	TC7=TCx	TC7>TCx
IOCx=OC7Dx IOC7=OM7/OL7	IOCx=OC7Dx +OMx/OLx IOC7=OM7/OL7	IOCx=OMx/OLx IOC7=OM7/OL7		IOCx=OC7Dx IOC7=OC7D7	IOCx=OC7Dx +OMx/OLx IOC7=OC7D7	IOCx=OMx/OLx IOC7=OC7D7	

If a read command is issued but the data is not retrieved within 512 serial clock cycles, a soft-reset will occur causing the command to be disregarded. The data is not available for retrieval after the time-out has occurred. This is the expected behavior if the handshake protocol is not enabled. However, consider the behavior where the BDC is running in a frequency much greater than the CPU frequency. In this case, the command could time out before the data is ready to be retrieved. In order to allow the data to be retrieved even with a large clock frequency mismatch (between BDC and CPU) when the hardware handshake protocol is enabled, the time out between a read command and the data retrieval is disabled. Therefore, the host could wait for more then 512 serial clock cycles and continue to be able to retrieve the data from an issued read command. However, as soon as the handshake pulse (ACK pulse) is issued, the time-out feature is re-activated, meaning that the target will time out after 512 clock cycles. Therefore, the host needs to retrieve the data within a 512 serial clock cycles time frame after the ACK pulse had been issued. After that period, the read command is discarded and the data is no longer available for retrieval. Any falling edge of the BKGD pin after the time-out period is considered to be a new command or a SYNC request.

Note that whenever a partially issued command, or partially retrieved data, has occurred the time out in the serial communication is active. This means that if a time frame higher than 512 serial clock cycles is observed between two consecutive negative edges and the command being issued or data being retrieved is not complete, a soft-reset will occur causing the partially received command or data retrieved to be disregarded. The next falling edge of the BKGD pin, after a soft-reset has occurred, is considered by the target as the start of a new BDM command, or the start of a SYNC request pulse.

16.4.13 Operation in Wait Mode

The BDM cannot be used in wait mode if the system disables the clocks to the BDM.

There is a clearing mechanism associated with the WAIT instruction when the clocks to the BDM (CPU core platform) are disabled. As the clocks restart from wait mode, the BDM receives a soft reset (clearing any command in progress) and the ACK function will be disabled. This is a change from previous BDM modules.

16.4.14 Operation in Stop Mode

The BDM is completely shutdown in stop mode.

There is a clearing mechanism associated with the STOP instruction. STOP must be enabled and the part must go into stop mode for this to occur. As the clocks restart from stop mode, the BDM receives a soft reset (clearing any command in progress) and the ACK function will be disabled. This is a change from previous BDM modules.

19.3.2.8 Port E Assignment Register (PEAR)

Module Base + 0x000A
Starting address location affected by INITRG register setting.

	7	6	5	4	3	2	1	0
R	NOACCE	0	PIPOE	NECLK	LSTRE	RDWE	0	0
W								
Reset								
Special Single Chip	0	0	0	0	0	0	0	0
Special Test	0	0	1	0	1	1	0	0
Peripheral	0	0	0	0	0	0	0	0
Emulation Expanded Narrow	1	0	1	0	1	1	0	0
Emulation Expanded Wide	1	0	1	0	1	1	0	0
Normal Single Chip	0	0	0	1	0	0	0	0
Normal Expanded Narrow	0	0	0	0	0	0	0	0
Normal Expanded Wide	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 19-12. Port E Assignment Register (PEAR)

Read: Anytime (provided this register is in the map).

Write: Each bit has specific write conditions. Please refer to the descriptions of each bit on the following pages.

Port E serves as general-purpose I/O or as system and bus control signals. The PEAR register is used to choose between the general-purpose I/O function and the alternate control functions. When an alternate control function is selected, the associated DDRE bits are overridden.

The reset condition of this register depends on the mode of operation because bus control signals are needed immediately after reset in some modes. In normal single-chip mode, no external bus control signals are needed so all of port E is configured for general-purpose I/O. In normal expanded modes, only the E clock is configured for its alternate bus control function and the other bits of port E are configured for general-purpose I/O. As the reset vector is located in external memory, the E clock is required for this access. R/$\overline{W}$ is only needed by the system when there are external writable resources. If the normal expanded system needs any other bus control signals, PEAR would need to be written before any access that needed the additional signals. In special test and emulation modes, IPIPE1, IPIPE0, E, $\overline{LSTRB}$, and R/$\overline{W}$ are configured out of reset as bus control signals.

This register is not in the on-chip memory map in expanded and special peripheral modes. Therefore, these accesses will be echoed externally.

19.3.2.9 Mode Register (MODE)

Module Base + 0x000B
Starting address location affected by INITRG register setting.

	7	6	5	4	3	2	1	0
R	MODC	MODB	MODA	0	IVIS	0	EMK	EME
W								
Reset								
Special Single Chip	0	0	0	0	0	0	0	0
Emulation Expanded Narrow	0	0	1	0	1	0	1	1
Special Test	0	1	0	0	1	0	0	0
Emulation Expanded Wide	0	1	1	0	1	0	1	1
Normal Single Chip	1	0	0	0	0	0	0	0
Normal Expanded Narrow	1	0	1	0	0	0	0	0
Peripheral	1	1	0	0	0	0	0	0
Normal Expanded Wide	1	1	1	0	0	0	0	0

= Unimplemented or Reserved

Figure 19-13. Mode Register (MODE)

Read: Anytime (provided this register is in the map).

Write: Each bit has specific write conditions. Please refer to the descriptions of each bit on the following pages.

The MODE register is used to establish the operating mode and other miscellaneous functions (i.e., internal visibility and emulation of port E and K).

In special peripheral mode, this register is not accessible but it is reset as shown to system configuration features. Changes to bits in the MODE register are delayed one cycle after the write.

This register is not in the on-chip memory map in expanded and special peripheral modes. Therefore, these accesses will be echoed externally.

20.3.2.9 Program Page Index Register (PPAGE)

Module Base + 0x0030
Starting address location affected by INITRG register setting.

	7	6	5	4	3	2	1	0
R	0	0	PIX5	PIX4	PIX3	PIX2	PIX1	PIX0
W								
Reset[1]	—	—	—	—	—	—	—	—

1. The reset state of this register is controlled at chip integration. Please refer to the device overview section to determine the actual reset state of this register.

= Unimplemented or Reserved

Figure 20-11. Program Page Index Register (PPAGE)

Read: Anytime

Write: Determined at chip integration. Generally it's: "write anytime in all modes;" on some devices it will be: "write only in special modes." Check specific device documentation to determine which applies.

Reset: Defined at chip integration as either 0x00 (paired with write in any mode) or 0x3C (paired with write only in special modes), see device overview chapter.

The HCS12 core architecture limits the physical address space available to 64K bytes. The program page index register allows for integrating up to 1M byte of FLASH or ROM into the system by using the six page index bits to page 16K byte blocks into the program page window located from 0x8000 to 0xBFFF as defined in Table 20-14. CALL and RTC instructions have special access to read and write this register without using the address bus.

NOTE

Normal writes to this register take one cycle to go into effect. Writes to this register using the special access of the CALL and RTC instructions will be complete before the end of the associated instruction.

Table 20-13. MEMSIZ0 Field Descriptions

Field	Description
5:0 PIX[5:0]	**Program Page Index Bits 5:0** — These page index bits are used to select which of the 64 FLASH or ROM array pages is to be accessed in the program page window as shown in Table 20-14.