Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	12V1
Core Size	16-Bit
Speed	25MHz
Connectivity	CANbus, IrDA, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	86
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	3.13V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	100-LQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12g128f0cll

- Bus-off recovery by software intervention or automatically
- 16-bit time stamp of transmitted/received messages

1.3.12 Serial Communication Interface Module (SCI)

- Up to three SCI modules
- Full-duplex or single-wire operation
- Standard mark/space non-return-to-zero (NRZ) format
- Selectable IrDA 1.4 return-to-zero-inverted (RZI) format with programmable pulse widths
- 13-bit baud rate selection
- Programmable character length
- Programmable polarity for transmitter and receiver
- Active edge receive wakeup
- Break detect and transmit collision detect supporting LIN 1.3, 2.0, 2.1 and SAE J2602

1.3.13 Serial Peripheral Interface Module (SPI)

- Up to three SPI modules
- Configurable 8- or 16-bit data size
- Full-duplex or single-wire bidirectional
- Double-buffered transmit and receive
- Master or slave mode
- MSB-first or LSB-first shifting
- Serial clock phase and polarity options

1.3.14 Analog-to-Digital Converter Module (ADC)

Up to 16-channel, 10-bit/12-bit[1] analog-to-digital converter

  — 3 us conversion time
  — 8-/10[1]-bit resolution
  — Left or right justified result data
  — Wakeup from low power modes on analog comparison > or <= match
  — Continuous conversion mode
  — External triggers to initiate conversions via GPIO or peripheral outputs such as PWM or TIM
  — Multiple channel scans
  — Precision fixed voltage reference for ADC conversions
  —

- Pins can also be used as digital I/O including wakeup capability

1. 12-bit resolution only available on S12GA192 and S12GA240 devices.

Table 1-15. 32-Pin LQFP Pinout for S12G48 and S12G64

	Function <----lowest-----PRIORITY-----highest---->					Power Supply	Internal Pull Resistor	
Package Pin	Pin	2nd Func.	3rd Func.	4th Func	5th Func		CTRL	Reset State
30	PS7	API_EXTCLK	ECLK	PWM5	$\overline{SS0}$	V_{DDX}	PERS/PPSS	Up
31	PM0	RXD1	RXCAN	—	—	V_{DDX}	PERM/PPSM	Disabled
32	PM1	TXD1	TXCAN	—	—	V_{DDX}	PERM/PPSM	Disabled

[1] The regular I/O characteristics (see Section A.2, “I/O Characteristics”) apply if the EXTAL/XTAL function is disabled

2.3.9 Pins PM3-0

Table 2-13. Port M Pins PM3-0

PM3	• 64/100 LQFP: The SCI2 TXD signal is mapped to this pin when used with the SCI function. If the SCI2 TXD signal is enabled the I/O state will depend on the SCI2 configuration. • Signal priority: 64/100 LQFP: TXD2 > GPO
PM2	• 64/100 LQFP: The SCI2 RXD signal is mapped to this pin when used with the SCI function. If the SCI2 RXD signal is enabled the I/O state will be forced to be input. • Signal priority: 64/100 LQFP: RXD2 > GPO
PM1	• Except 20 TSSOP: The TXCAN signal is mapped to this pin when used with the CAN function. The enabled CAN forces the I/O state to be an output. • 32 LQFP: The SCI1 TXD signal is mapped to this pin when used with the SCI function. If the SCI1 TXD signal is enabled the I/O state will depend on the SCI1 configuration. • 48 LQFP: The SCI2 TXD signal is mapped to this pin when used with the SCI function. If the SCI2 TXD signal is enabled the I/O state will depend on the SCI2 configuration. • Signal priority: 32 LQFP: TXCAN > TXD1 > GPO 48 LQFP: TXCAN > TXD2 > GPO 64/100 LQFP: TXCAN > GPO
PM0	• Except 20 TSSOP: The RXCAN signal is mapped to this pin when used with the CAN function. The enabled CAN forces the I/O state to be an input. If CAN is active the selection of a pulldown device on the RXCAN input has no effect. • 32 LQFP: The SCI1 RXD signal is mapped to this pin when used with the SCI function. The enabled SCI1 RXD signal forces the I/O state to an input. • 48 LQFP: The SCI2 RXD signal is mapped to this pin when used with the SCI function. The enabled SCI2 RXD signal forces the I/O state to an input. • Signal priority: 32 LQFP: RXCAN > RXD1 > GPO 48 LQFP: RXCAN > RXD2 > GPO 64/100 LQFP: RXCAN > GPO

2.3.10 Pins PP7-0

Table 2-14. Port P Pins PP7-0

PP7-PP6	• 64/100 LQFP: The PWM channels 7 and 6 signal are mapped to these pins when used with the PWM function. The enabled PWM channel forces the I/O state to be an output. • 64/100 LQFP: Pin interrupts can be generated if enabled in input or output mode. • Signal priority: 64/100 LQFP: PWM > GPO
PP5-PP4	• 48/64/100 LQFP: The PWM channels 5 and 4 signal are mapped to these pins when used with the PWM function. The enabled PWM channel forces the I/O state to be an output. • 48/64/100 LQFP: Pin interrupts can be generated if enabled in input or output mode. • Signal priority: 48/64/100 LQFP: PWM > GPO

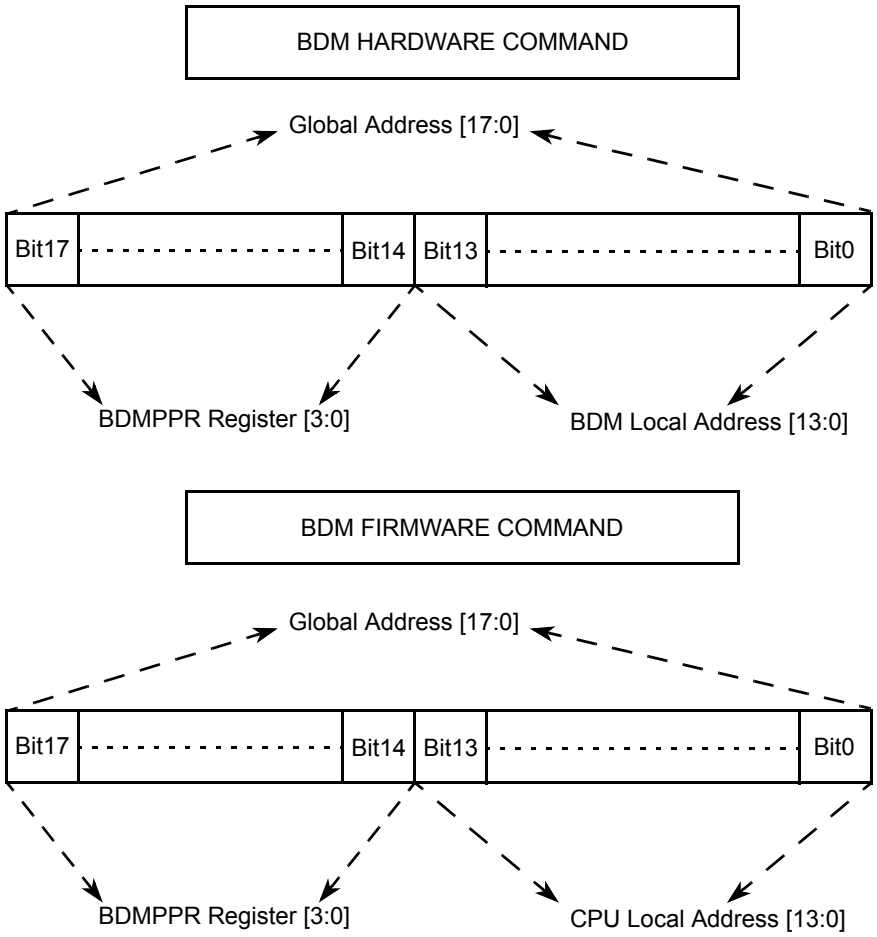

BDM HARDWARE COMMAND
Global Address [17:0]
Bit17
Bit14
Bit13
Bit0
BDMPPR Register [3:0]
BDM Local Address [13:0]
BDM FIRMWARE COMMAND
Global Address [17:0]
Bit17
Bit14
Bit13
Bit0
BDMPPR Register [3:0]
CPU Local Address [13:0]

Figure 5-10.

6.5 Initialization/Application Information

6.5.1 Initialization

After system reset, software should:

1. Initialize the interrupt vector base register if the interrupt vector table is not located at the default location (0xFF80–0xFFF9).
2. Enable I bit maskable interrupts by clearing the I bit in the CCR.
3. Enable the X bit maskable interrupt by clearing the X bit in the CCR.

6.5.2 Interrupt Nesting

The interrupt request scheme makes it possible to nest I bit maskable interrupt requests handled by the CPU.

- I bit maskable interrupt requests can be interrupted by an interrupt request with a higher priority.

I bit maskable interrupt requests cannot be interrupted by other I bit maskable interrupt requests per default. In order to make an interrupt service routine (ISR) interruptible, the ISR must explicitly clear the I bit in the CCR (CLI). After clearing the I bit, other I bit maskable interrupt requests can interrupt the current ISR.

An ISR of an interruptible I bit maskable interrupt request could basically look like this:

1. Service interrupt, that is clear interrupt flags, copy data, etc.
2. Clear I bit in the CCR by executing the instruction CLI (thus allowing other I bit maskable interrupt requests)
3. Process data
4. Return from interrupt by executing the instruction RTI

6.5.3 Wake Up from Stop or Wait Mode

6.5.3.1 CPU Wake Up from Stop or Wait Mode

Every I bit maskable interrupt request is capable of waking the MCU from stop or wait mode. To determine whether an I bit maskable interrupts is qualified to wake-up the CPU or not, the same conditions as in normal run mode are applied during stop or wait mode:

- If the I bit in the CCR is set, all I bit maskable interrupts are masked from waking-up the MCU.

Since there are no clocks running in stop mode, only interrupts which can be asserted asynchronously can wake-up the MCU from stop mode.

The X bit maskable interrupt request can wake up the MCU from stop or wait mode at anytime, even if the X bit in CCR is set[1].

1. The capability of the $\overline{\text{XIRQ}}$ pin to wake-up the MCU with the X bit set may not be available if, for example, the $\overline{\text{XIRQ}}$ pin is shared with other peripheral modules on the device. Please refer to the Device section of the MCU reference manual for details.

8.4.5.1.1 Storing with Begin Trigger Alignment

Storing with begin alignment, data is not stored in the Trace Buffer until the Final State is entered. Once the trigger condition is met the DBG module remains armed until 64 lines are stored in the Trace Buffer. If the trigger is at the address of the change-of-flow instruction the change of flow associated with the trigger is stored in the Trace Buffer. Using begin alignment together with tagging, if the tagged instruction is about to be executed then the trace is started. Upon completion of the tracing session the breakpoint is generated, thus the breakpoint does not occur at the tagged instruction boundary.

8.4.5.1.2 Storing with End Trigger Alignment

Storing with end alignment, data is stored in the Trace Buffer until the Final State is entered, at which point the DBG module becomes disarmed and no more data is stored. If the trigger is at the address of a change of flow instruction, the trigger event is not stored in the Trace Buffer. If all trace buffer lines have been used before a trigger event occurrs then the trace continues at the first line, overwriting the oldest entries.

8.4.5.2 Trace Modes

Four trace modes are available. The mode is selected using the TRCMOD bits in the DBGTCR register. Tracing is enabled using the TSOURCE bit in the DBGTCR register. The modes are described in the following subsections.

8.4.5.2.1 Normal Mode

In Normal Mode, change of flow (COF) program counter (PC) addresses are stored.

COF addresses are defined as follows:

- Source address of taken conditional branches (long, short, bit-conditional, and loop primitives)
- Destination address of indexed JMP, JSR, and CALL instruction
- Destination address of RTI, RTS, and RTC instructions
- Vector address of interrupts, except for BDM vectors

LBRA, BRA, BSR, BGND as well as non-indexed JMP, JSR, and CALL instructions are not classified as change of flow and are not stored in the trace buffer.

Stored information includes the full 18-bit address bus and information bits, which contains a source/destination bit to indicate whether the stored address was a source address or destination address.

NOTE

When a COF instruction with destination address is executed, the destination address is stored to the trace buffer on instruction completion, indicating the COF has taken place. If an interrupt occurs simultaneously then the next instruction carried out is actually from the interrupt service routine. The instruction at the destination address of the original program flow gets executed after the interrupt service routine.

- The Internal Reference Clock (IRC1M) provides a1MHz clock.

10.1.1 Features

The Pierce Oscillator (XOSCLCP) contains circuitry to dynamically control current gain in the output amplitude. This ensures a signal with low harmonic distortion, low power and good noise immunity.

- Supports quartz crystals or ceramic resonators from 4MHz to 16MHz.
- High noise immunity due to input hysteresis and spike filtering.
- Low RF emissions with peak-to-peak swing limited dynamically
- Transconductance (gm) sized for optimum start-up margin for typical crystals
- Dynamic gain control eliminates the need for external current limiting resistor
- Integrated resistor eliminates the need for external bias resistor.
- Low power consumption: Operates from internal 1.8V (nominal) supply, Amplitude control limits power

The Voltage Regulator (IVREG) has the following features:

- Input voltage range from 3.13V to 5.5V
- Low-voltage detect (LVD) with low-voltage interrupt (LVI)
- Power-on reset (POR)
- Low-voltage reset (LVR)

The Phase Locked Loop (PLL) has the following features:

- highly accurate and phase locked frequency multiplier
- Configurable internal filter for best stability and lock time.
- Frequency modulation for defined jitter and reduced emission
- Automatic frequency lock detector
- Interrupt request on entry or exit from locked condition
- Reference clock either external (crystal) or internal square wave (1MHz IRC1M) based.
- PLL stability is sufficient for LIN communication, even if using IRC1M as reference clock

The Internal Reference Clock (IRC1M) has the following features:

- Frequency trimming
  (A factory trim value for 1MHz is loaded from Flash Memory into the IRCTRIM register after reset, which can be overwritten by application if required)
- Temperature Coefficient (TC) trimming.
  (A factory trim value is loaded from Flash Memory into the IRCTRIM register to turned off TC trimming after reset. Application can trim the TC if required by overwriting the IRCTRIM register).
-

Other features of the S12CPMU include

- Clock monitor to detect loss of crystal

12.3.2.9 ATD Status Register 2 (ATDSTAT2)

This read-only register contains the Conversion Complete Flags CCF[7:0].

Module Base + 0x000A

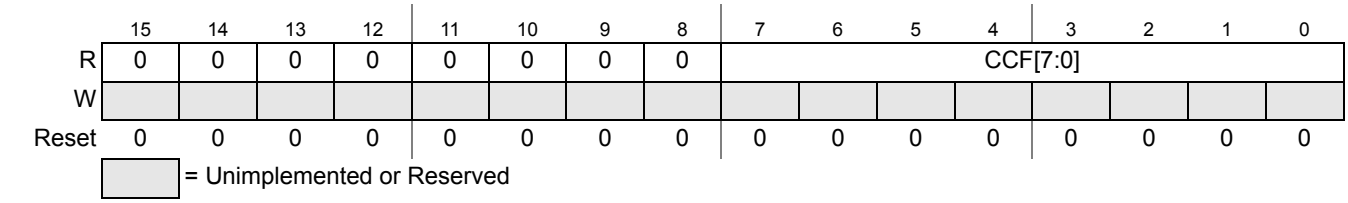


Figure 12-11. ATD Status Register 2 (ATDSTAT2)

Read: Anytime

Write: Anytime (for details see Table 12-18 below)

Table 12-18. ATDSTAT2 Field Descriptions

Field	Description
7–0 CCF[7:0]	**Conversion Complete Flag *n* (*n*= 7, 6, 5, 4, 3, 2, 1, 0) (*n conversion number, NOT channel number!)*—** A conversion complete flag is set at the end of each conversion in a sequence. The flags are associated with the conversion position in a sequence (and also the result register number). Therefore in non-fifo mode, CCF[4] is set when the fifth conversion in a sequence is complete and the result is available in result register ATDDR4; CCF[5] is set when the sixth conversion in a sequence is complete and the result is available in ATDDR5, and so forth. If automatic compare of conversion results is enabled (CMPE[*n*]=1 in ATDCMPE), the conversion complete flag is only set if comparison with ATDDR*n* is true. If ACMPIE=1 a compare interrupt will be requested. In this case, as the ATDDR*n* result register is used to hold the compare value, the result will not be stored there at the end of the conversion but is lost. A flag CCF[*n*] is cleared when one of the following occurs: A) Write to ATDCTL5 (a new conversion sequence is started) B) If AFFC=0, write “1” to CCF[*n*] C) If AFFC=1 and CMPE[*n*]=0, read of result register ATDDR*n* D) If AFFC=1 and CMPE[*n*]=1, write to result register ATDDR*n* In case of a concurrent set and clear on CCF[*n*]: The clearing by method A) will overwrite the set. The clearing by methods B) or C) or D) will be overwritten by the set. 0 Conversion number *n* not completed or successfully compared 1 If (CMPE[*n*]=0): Conversion number *n* has completed. Result is ready in ATDDR*n*. If (CMPE[*n*]=1): Compare for conversion result number *n* with compare value in ATDDR*n*, using compare operator CMPGT[*n*] is true. (No result available in ATDDR*n*)

Table 19-9. PWMCAE Field Descriptions

Note: Bits related to available channels have functional significance. Writing to unavailable bits has no effect. Read from unavailable bits return a zero

Field	Description
7–0 CAE[7:0]	Center Aligned Output Modes on Channels 7–0 0 Channels 7–0 operate in left aligned output mode. 1 Channels 7–0 operate in center aligned output mode.

19.3.2.6 PWM Control Register (PWMCTL)

The PWMCTL register provides for various control of the PWM module.

Module Base + 0x0005

	7	6	5	4	3	2	1	0
R	CON67	CON45	CON23	CON01	PSWAI	PFRZ	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 19-8. PWM Control Register (PWMCTL)

Read: Anytime

Write: Anytime

There are up to four control bits for concatenation, each of which is used to concatenate a pair of PWM channels into one 16-bit channel. If the corresponding channels do not exist on a particular derivative, then writes to these bits have no effect and reads will return zeroes. When channels 6 and 7are concatenated, channel 6 registers become the high order bytes of the double byte channel. When channels 4 and 5 are concatenated, channel 4 registers become the high order bytes of the double byte channel. When channels 2 and 3 are concatenated, channel 2 registers become the high order bytes of the double byte channel. When channels 0 and 1 are concatenated, channel 0 registers become the high order bytes of the double byte channel.

See Section 19.4.2.7, “PWM 16-Bit Functions” for a more detailed description of the concatenation PWM Function.

NOTE

Change these bits only when both corresponding channels are disabled.

19.4.2 PWM Channel Timers

The main part of the PWM module are the actual timers. Each of the timer channels has a counter, a period register and a duty register (each are 8-bit). The waveform output period is controlled by a match between the period register and the value in the counter. The duty is controlled by a match between the duty register and the counter value and causes the state of the output to change during the period. The starting polarity of the output is also selectable on a per channel basis. Shown below in Figure 19-16 is the block diagram for the PWM timer.

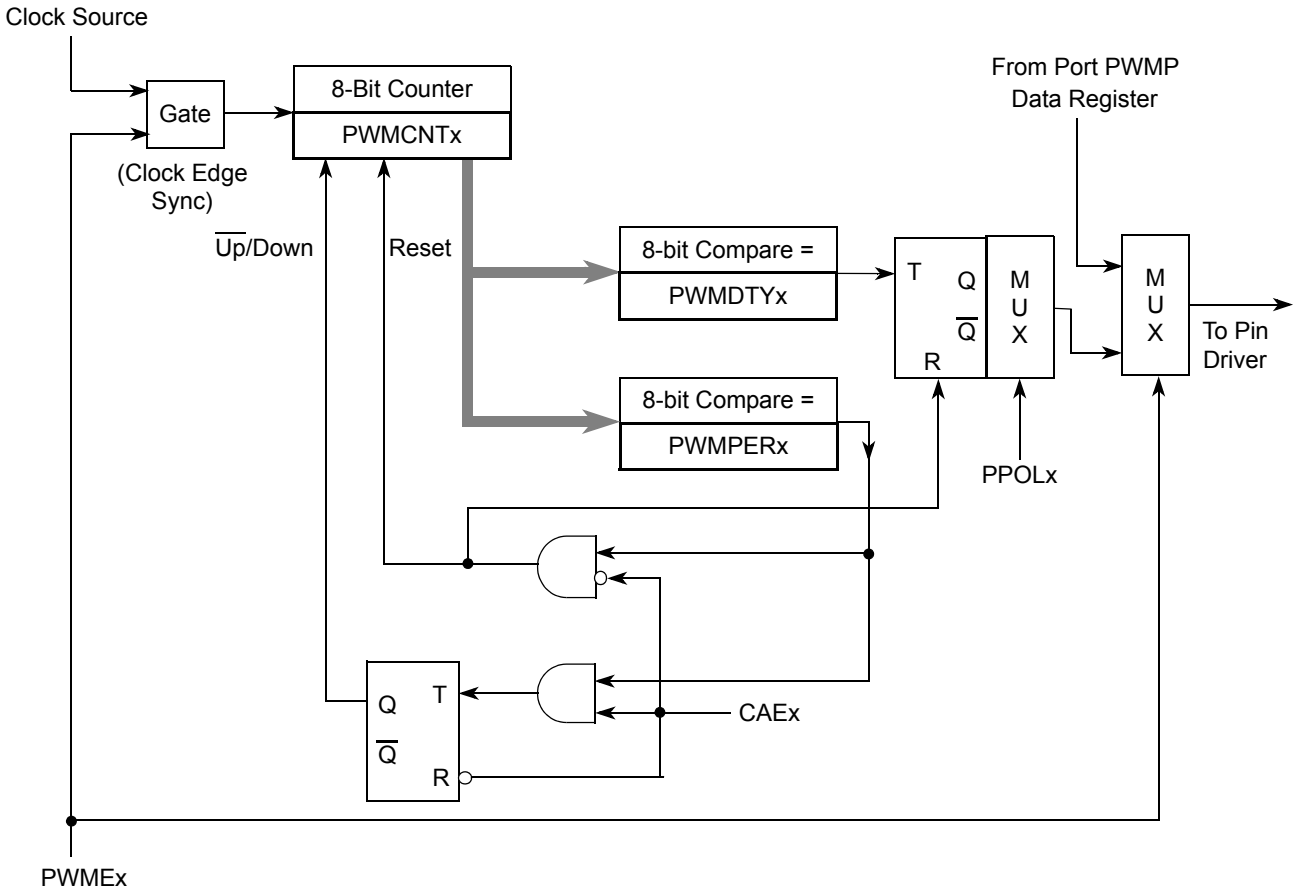


Figure 19-16. PWM Timer Channel Block Diagram

19.4.2.1 PWM Enable

Each PWM channel has an enable bit (PWMEx) to start its waveform output. When any of the PWMEx bits are set (PWMEx = 1), the associated PWM output signal is enabled immediately. However, the actual PWM waveform is not available on the associated PWM output until its clock source begins its next cycle due to the synchronization of PWMEx and the clock source. An exception to this is when channels are concatenated. Refer to Section 19.4.2.7, “PWM 16-Bit Functions” for more detail.

NOTE

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

Table 23-25. TIM16B8CV3 Interrupts

Interrupt	Offset	Vector	Priority	Source	Description
C[7:0]F	—	—	—	Timer Channel 7–0	Active high timer channel interrupts 7–0
PAOVI	—	—	—	Pulse Accumulator Input	Active high pulse accumulator input interrupt
PAOVF	—	—	—	Pulse Accumulator Overflow	Pulse accumulator overflow interrupt
TOF	—	—	—	Timer Overflow	Timer Overflow interrupt

The TIM16B8CV3 could use up to 11 interrupt vectors. The interrupt vector offsets and interrupt numbers are chip dependent.

23.6.1 Channel [7:0] Interrupt (C[7:0]F)

This active high outputs will be asserted by the module to request a timer channel 7 – 0 interrupt. The TIM block only generates the interrupt and does not service it. Only bits related to implemented channels are valid.

23.6.2 Pulse Accumulator Input Interrupt (PAOVI)

This active high output will be asserted by the module to request a timer pulse accumulator input interrupt. The TIM block only generates the interrupt and does not service it.

23.6.3 Pulse Accumulator Overflow Interrupt (PAOVF)

This active high output will be asserted by the module to request a timer pulse accumulator overflow interrupt. The TIM block only generates the interrupt and does not service it.

23.6.4 Timer Overflow Interrupt (TOF)

This active high output will be asserted by the module to request a timer overflow interrupt. The TIM block only generates the interrupt and does not service it.

Offset Module Base + 0x0004

	7	6	5	4	3	2	1	0
R	CCIE	0	0	IGNSF	0	0	FDFD	FSFD
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 24-9. Flash Configuration Register (FCNFG)

CCIE, IGNSF, FDFD, and FSFD bits are readable and writable while remaining bits read 0 and are not writable.

Table 24-13. FCNFG Field Descriptions

Field	Description
7 CCIE	**Command Complete Interrupt Enable** — The CCIE bit controls interrupt generation when a Flash command has completed. 0 Command complete interrupt disabled 1 An interrupt will be requested whenever the CCIF flag in the FSTAT register is set (see Section 24.3.2.7)
4 IGNSF	**Ignore Single Bit Fault** — The IGNSF controls single bit fault reporting in the FERSTAT register (see Section 24.3.2.8). 0 All single bit faults detected during array reads are reported 1 Single bit faults detected during array reads are not reported and the single bit fault interrupt will not be generated
1 FDFD	**Force Double Bit Fault Detect** — The FDFD bit allows the user to simulate a double bit fault during Flash array read operations and check the associated interrupt routine. The FDFD bit is cleared by writing a 0 to FDFD. 0 Flash array read operations will set the DFDIF flag in the FERSTAT register only if a double bit fault is detected 1 Any Flash array read operation will force the DFDIF flag in the FERSTAT register to be set (see Section 24.3.2.7) and an interrupt will be generated as long as the DFDIE interrupt enable in the FERCNFG register is set (see Section 24.3.2.6)
0 FSFD	**Force Single Bit Fault Detect** — The FSFD bit allows the user to simulate a single bit fault during Flash array read operations and check the associated interrupt routine. The FSFD bit is cleared by writing a 0 to FSFD. 0 Flash array read operations will set the SFDIF flag in the FERSTAT register only if a single bit fault is detected 1 Flash array read operation will force the SFDIF flag in the FERSTAT register to be set (see Section 24.3.2.7) and an interrupt will be generated as long as the SFDIE interrupt enable in the FERCNFG register is set (see Section 24.3.2.6)

24.3.2.6 Flash Error Configuration Register (FERCNFG)

The FERCNFG register enables the Flash error interrupts for the FERSTAT flags.

- VERNUM: Version number. The first version is number 0b_0001 with both 0b_0000 and 0b_1111 meaning ‘none’.

25.4.3 Internal NVM resource (NVMRES)

IFR is an internal NVM resource readable by CPU , when NVMRES is active. The IFR fields are shown in Table 25-5.

The NVMRES global address map is shown in Table 25-6.

25.4.4 Flash Command Operations

Flash command operations are used to modify Flash memory contents.

The next sections describe:

- How to write the FCLKDIV register that is used to generate a time base (FCLK) derived from BUSCLK for Flash program and erase command operations
- The command write sequence used to set Flash command parameters and launch execution
- Valid Flash commands available for execution, according to MCU functional mode and MCU security state.

25.4.4.1 Writing the FCLKDIV Register

Prior to issuing any Flash program or erase command after a reset, the user is required to write the FCLKDIV register to divide BUSCLK down to a target FCLK of 1 MHz. Table 25-8 shows recommended values for the FDIV field based on BUSCLK frequency.

NOTE

Programming or erasing the Flash memory cannot be performed if the bus clock runs at less than 0.8 MHz. Setting FDIV too high can destroy the Flash memory due to overstress. Setting FDIV too low can result in incomplete programming or erasure of the Flash memory cells.

When 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. If the FCLKDIV register has not been written, any Flash program or erase command loaded during a command write sequence will not execute and the ACCERR bit in the FSTAT register will set.

25.4.4.2 Command Write Sequence

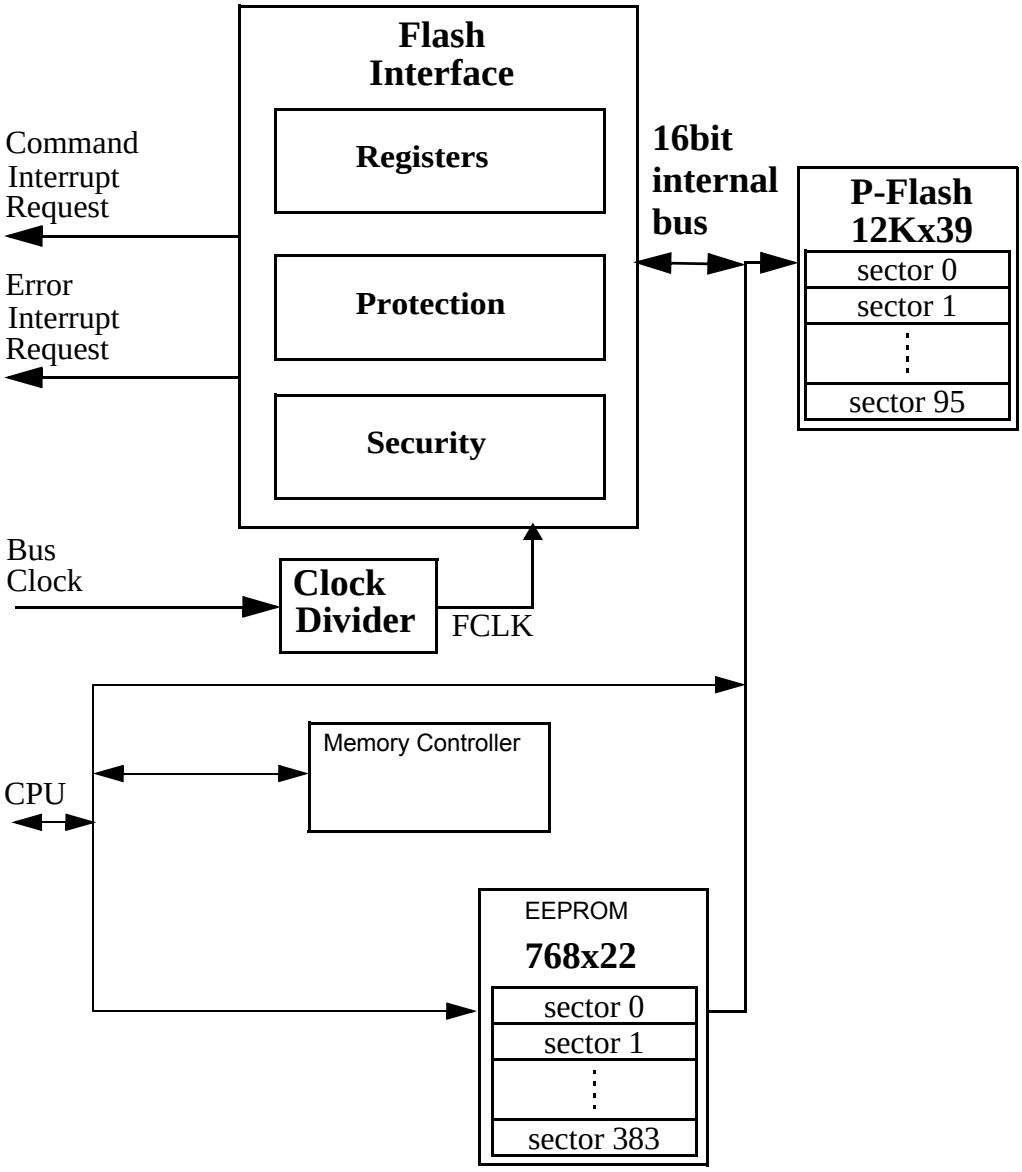
The Memory Controller will launch all valid Flash commands entered using a command write sequence.

Before launching a command, the ACCERR and FPVIOL bits in the FSTAT register must be clear (see Section 25.3.2.7) and the CCIF flag should be tested to determine the status of the current command write sequence. If CCIF is 0, the previous command write sequence is still active, a new command write sequence cannot be started, and all writes to the FCCOB register are ignored.

26.1.3 Block Diagram

The block diagram of the Flash module is shown in Figure 26-1.

Figure 26-1. FTMRG48K1 Block Diagram



26.2 External Signal Description

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

Table 26-28. P-Flash Commands

FCMD	Command	Function on P-Flash Memory
0x02	Erase Verify Block	Verify that a P-Flash block is erased.
0x03	Erase Verify P-Flash Section	Verify that a given number of words starting at the address provided are erased.
0x04	Read Once	Read a dedicated 64 byte field in the nonvolatile information register in P-Flash block that was previously programmed using the Program Once command.
0x06	Program P-Flash	Program a phrase in a P-Flash block.
0x07	Program Once	Program a dedicated 64 byte field in the nonvolatile information register in P-Flash block that is allowed to be programmed only once.
0x08	Erase All Blocks	Erase all P-Flash (and EEPROM) blocks. An erase of all Flash blocks is only possible when the FPLDIS, FPHDIS, and FPOPEN bits in the FPROT register and the DPOPEN bit in the EEPROT register are set prior to launching the command.
0x09	Erase Flash Block	Erase a P-Flash (or EEPROM) block. An erase of the full P-Flash block is only possible when FPLDIS, FPHDIS and FPOPEN bits in the FPROT register are set prior to launching the command.
0x0A	Erase P-Flash Sector	Erase all bytes in a P-Flash sector.
0x0B	Unsecure Flash	Supports a method of releasing MCU security by erasing all P-Flash (and EEPROM) blocks and verifying that all P-Flash (and EEPROM) blocks are erased.
0x0C	Verify Backdoor Access Key	Supports a method of releasing MCU security by verifying a set of security keys.
0x0D	Set User Margin Level	Specifies a user margin read level for all P-Flash blocks.
0x0E	Set Field Margin Level	Specifies a field margin read level for all P-Flash blocks (special modes only).

26.4.4.5 EEPROM Commands

Table 26-29 summarizes the valid EEPROM commands along with the effects of the commands on the EEPROM block.

Table 26-29. EEPROM Commands

FCMD	Command	Function on EEPROM Memory
0x01	Erase Verify All Blocks	Verify that all EEPROM (and P-Flash) blocks are erased.
0x02	Erase Verify Block	Verify that the EEPROM block is erased.

27.4.7 Interrupts

The Flash module can generate an interrupt when a Flash command operation has completed or when a Flash command operation has detected an ECC fault.

Table 27-66. Flash Interrupt Sources

Interrupt Source	Interrupt Flag	Local Enable	Global (CCR) Mask
Flash Command Complete	CCIF (FSTAT register)	CCIE (FCNFG register)	I Bit
ECC Double Bit Fault on Flash Read	DFDIF (FERSTAT register)	DFDIE (FERCNFG register)	I Bit
ECC Single Bit Fault on Flash Read	SFDIF (FERSTAT register)	SFDIE (FERCNFG register)	I Bit

NOTE

Vector addresses and their relative interrupt priority are determined at the MCU level.

27.4.7.1 Description of Flash Interrupt Operation

The Flash module uses the CCIF flag in combination with the CCIE interrupt enable bit to generate the Flash command interrupt request. The Flash module uses the DFDIF and SFDIF flags in combination with the DFDIE and SFDIE interrupt enable bits to generate the Flash error interrupt request. For a detailed description of the register bits involved, refer to Section 27.3.2.5, “Flash Configuration Register (FCNFG)”, Section 27.3.2.6, “Flash Error Configuration Register (FERCNFG)”, Section 27.3.2.7, “Flash Status Register (FSTAT)”, and Section 27.3.2.8, “Flash Error Status Register (FERSTAT)”.

The logic used for generating the Flash module interrupts is shown in Figure 27-27.

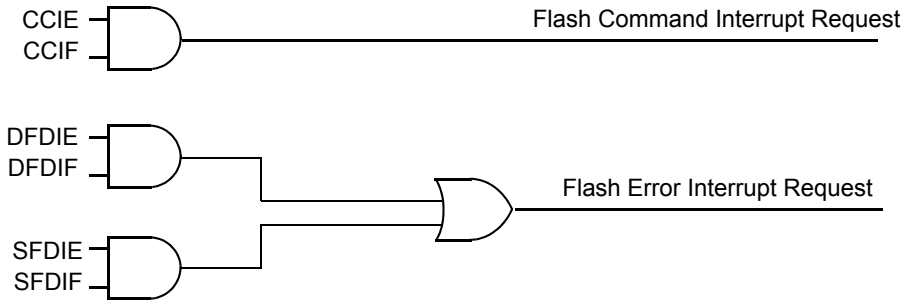


Figure 27-27. Flash Module Interrupts Implementation

The user code stored in the P-Flash memory must have a method of receiving the backdoor keys 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 27.3.2.2), the MCU can be unsecured by the backdoor key access sequence described below:

1. Follow the command sequence for the Verify Backdoor Access Key command as explained in Section 27.4.6.11
2. If the Verify Backdoor Access Key command is successful, the MCU is unsecured and the SEC[1:0] bits in the FSEC register are forced to the unsecure state of 10

The Verify Backdoor Access Key command is monitored by the Memory Controller and an illegal key will prohibit future use of the Verify Backdoor Access Key command. A reset of the MCU is the only method to re-enable the Verify Backdoor Access Key command. The security as defined in the Flash security byte (0x3_FF0F) is not changed by using the Verify Backdoor Access Key command sequence. The backdoor keys stored in addresses 0x3_FF00-0x3_FF07 are unaffected by the Verify Backdoor Access Key command sequence. The Verify Backdoor Access Key command sequence has no effect on the program and erase protections defined in the Flash protection register, FPROT.

After the backdoor keys have been correctly matched, the MCU will be unsecured. After the MCU is unsecured, the sector containing the Flash security byte can be erased and the Flash security byte can be reprogrammed to the unsecure state, if desired. In the unsecure state, the user has full control of the contents of the backdoor keys by programming addresses 0x3_FF00-0x3_FF07 in the Flash configuration field.

27.5.2 Unsecuring the MCU in Special Single Chip Mode using BDM

A secured MCU can be unsecured in special single chip mode by using the following method to erase the P-Flash and EEPROM memory:

1. Reset the MCU into special single chip mode
2. Delay while the BDM executes the Erase Verify All Blocks command write sequence to check if the P-Flash and EEPROM memories are erased
3. Send BDM commands to disable protection in the P-Flash and EEPROM memory
4. Execute the Erase All Blocks command write sequence to erase the P-Flash and EEPROM memory. Alternatively the Unsecure Flash command can be executed, if so the steps 5 and 6 below are skeeped.
5. After the CCIF flag sets to indicate that the Erase All Blocks operation has completed, reset the MCU into special single chip mode
6. Delay while the BDM executes the Erase Verify All Blocks command write sequence to verify that the P-Flash and EEPROM memory are erased

If the P-Flash and EEPROM memory are verified as erased, the MCU will be unsecured. All BDM commands will now be enabled and the Flash security byte may be programmed to the unsecure state by continuing with the following steps:

7. Send BDM commands to execute the Program P-Flash command write sequence to program the Flash security byte to the unsecured state

30.3.2 Register Descriptions

The Flash module contains a set of 20 control and status registers located between Flash module base + 0x0000 and 0x0013.

In the case of the writable registers, the write accesses are forbidden during Fash command execution (for more detail, see Caution note in Section 30.3).

A summary of the Flash module registers is given in Figure 30-4 with detailed descriptions in the following subsections.

Address & Name		7	6	5	4	3	2	1	0
0x0000 FCLKDIV	R	FDIVLD	FDIVLCK	FDIV5	FDIV4	FDIV3	FDIV2	FDIV1	FDIV0
	W								
0x0001 FSEC	R	KEYEN1	KEYEN0	RNV5	RNV4	RNV3	RNV2	SEC1	SEC0
	W								
0x0002 FCCOBIX	R	0	0	0	0	0	CCOBIX2	CCOBIX1	CCOBIX0
	W								
0x0003 FRSV0	R	0	0	0	0	0	0	0	0
	W								
0x0004 FCNFG	R	CCIE	0	0	IGNSF	0	0	FDFD	FSFD
	W								
0x0005 FERCNFG	R	0	0	0	0	0	0	DFDIE	SFDIE
	W								
0x0006 FSTAT	R	CCIF	0	ACCERR	FPVIOL	MGBUSY	RSVD	MGSTAT1	MGSTAT0
	W								
0x0007 FERSTAT	R	0	0	0	0	0	0	DFDIF	SFDIF
	W								
0x0008 FPROT	R	FPOPEN	RNV6	FPHDIS	FPHS1	FPHS0	FPLDIS	FPLS1	FPLS0
	W								
0x0009 EEPROT	R	DPOPEN	DPS6	DPS5	DPS4	DPS3	DPS2	DPS1	DPS0
	W								

Figure 30-4. FTMRG192K2 Register Summary

Table 30-57. Set Field Margin Level Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x0E	Flash block selection code [1:0]. See Table 30-34
001	Margin level setting.	

Upon clearing CCIF to launch the Set Field Margin Level command, the Memory Controller will set the field margin level for the targeted block and then set the CCIF flag.

NOTE

When the EEPROM block is targeted, the EEPROM field margin levels are applied only to the EEPROM reads. However, when the P-Flash block is targeted, the P-Flash field margin levels are applied to both P-Flash and EEPROM reads. It is not possible to apply field margin levels to the P-Flash block only.

Valid margin level settings for the Set Field Margin Level command are defined in Table 30-58.

Table 30-58. Valid Set Field Margin Level Settings

CCOB (CCOBIX=001)	Level Description
0x0000	Return to Normal Level
0x0001	User Margin-1 Level[1]
0x0002	User Margin-0 Level[2]
0x0003	Field Margin-1 Level[1]
0x0004	Field Margin-0 Level[2]

1 Read margin to the erased state

2 Read margin to the programmed state

Table 30-59. Set Field Margin Level Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 001 at command launch.
		Set if command not available in current mode (see Table 30-27).
		Set if an invalid margin level setting is supplied.
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None

Appendix B
Detailed Register Address Map

Revision History

Version Number	Revision Date	Description of Changes
Rev 0.05	30-Aug-2010	• Updated ADCCTL2 register in Appendix B, “Detailed Register Address Map”. • Updated CPMUOSC register in Appendix B, “Detailed Register Address Map”.
Rev 0.06	18-Oct-2010	• Updated ADC registers in Appendix B, “Detailed Register Address Map”.
Rev 0.07	9-Nov-2010	• Updated CPMU registers in Appendix B, “Detailed Register Address Map”.
Rev 0.08	4-Dec-2010	• Updated PIM registers in Appendix B, “Detailed Register Address Map”.
Rev 0.09	24-Apr-2012	• Typos and formatting

B.1 Detailed Register Map

The following tables show the detailed register map of the MC9S12G-Family.

NOTE

This is a summary of all register bits implemented on MC9S12G devices. Each member of the MC9S12G-Family implements the subset of registers, which is associated with its feature set (see Table 1-1).

0x0000–0x0009 Port Integration Module (PIM) Map 1 of 6

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0000	PORTA	R	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA 0
		W								
0x0001	PORTB	R	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
		W								
0x0002	DDRA	R	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		W								
0x0003	DDRB	R	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		W								
0x0004	PORTC	R	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
		W								
0x0005	PORTD	R	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
		W								
0x0006	DDRC	R	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		W								
0x0007	DDRD	R	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
		W								
0x0008	PORTE	R	0	0	0	0	0	0	PE1	PE0
		W								
0x0009	DDRE	R	0	0	0	0	0	0	DDRE1	DDRE0
		W								