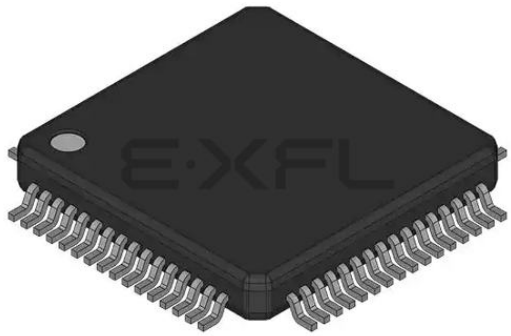


Welcome to E-XFL.COM

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	54
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	4K 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	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=s9s12g64f0clh

Address	Module	Size (Bytes)
0x000A–0x000B	MMC (Memory Map Control)	2
0x000C–0x000D	PIM (Port Integration Module)	2
0x000E–0x000F	Reserved	2
0x0010–0x0017	MMC (Memory Map Control)	8
0x0018–0x0019	Reserved	2
0x001A–0x001B	Device ID register	2
0x001C–0x001F	PIM (Port Integration Module)	4
0x0020–0x002F	DBG (Debug Module)	16
0x0030–0x0033	Reserved	4
0x0034–0x003F	CPMU (Clock and Power Management)	12
0x0040–0x006F	TIM (Timer Module <= 8 channels)	48
0x0070–0x009F	ADC (Analog to Digital Converter <= 16 channels)	48
0x00A0–0x00C7	PWM (Pulse-Width Modulator <= 8 channels)	40
0x00C8–0x00CF	SCI0 (Serial Communication Interface)	8
0x00D0–0x00D7	SCI1 (Serial Communication Interface)[1]	8
0x00D8–0x00DF	SPI0 (Serial Peripheral Interface)	8
0x00E0–0x00E7	Reserved	8
0x00E8–0x00EF	SCI2 (Serial Communication Interface)[2]	8
0x00F0–0x00F7	SPI1 (Serial Peripheral Interface)[3]	8
0x00F8–0x00FF	SPI2 (Serial Peripheral Interface)[4]	8
0x0100–0x0113	FTMRG control registers	20
0x0114–0x011F	Reserved	12
0x0120	INT (Interrupt Module)	1
0x0121–0x013F	Reserved	31
0x0140–0x017F	CAN[5]	64
0x0180–0x023F	Reserved	192
0x0240–0x025F	PIM (Port Integration Module)	32
0x0260–0x0261	ACMP (Analog Comparator)[6]	2
0x0262–0x0275	PIM (Port Integration Module)	20
0x0276	RVA (Reference Voltage Attenuator)[7]	1
0x0277–0x027F	PIM (Port Integration Module)	9
0x0280–0x02EF	Reserved	112
0x02F0–0x02FF	CPMU (Clock and Power Management)	16
0x0300–0x03BF	Reserved	192
0x03C0–0x03C7	DAC0 (Digital to Analog Converter)[8]	8

1.8.7.3 Pinout 100-Pin LQFP

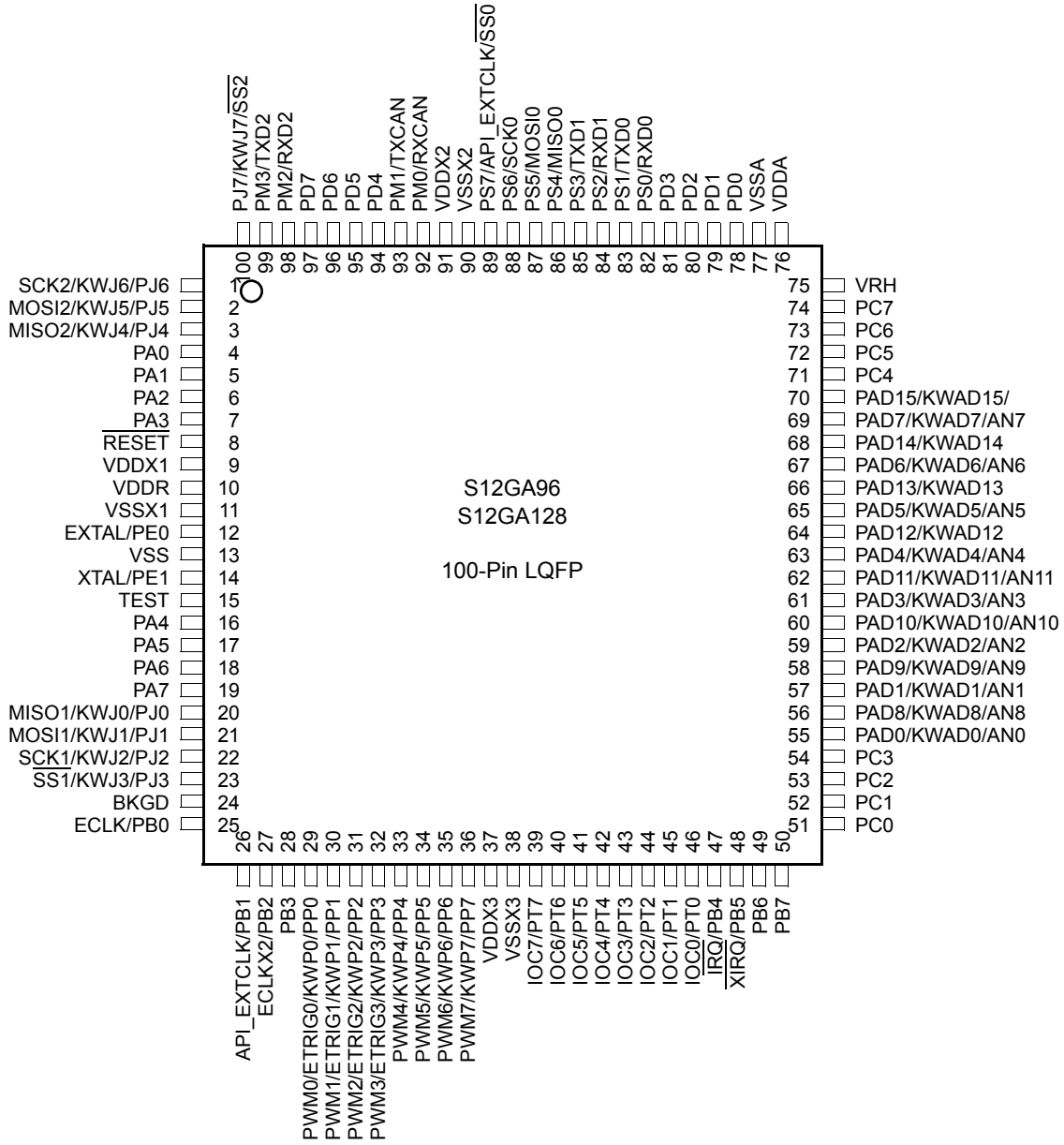


Figure 1-20. 100-Pin LQFP Pinout for S12GA96 and S12GA128

Table 8-18. State2 —Sequencer Next State Selection

SC[3:0]	Description (Unspecified matches have no effect)
0101	Match2 to Final State
0110	Match2 to State1..... Match0 to Final State
0111	Either Match0 or Match1 to Final State
1000	Reserved
1001	Reserved
1010	Reserved
1011	Reserved
1100	Either Match0 or Match1 to Final State.....Match2 to State3
1101	Reserved
1110	Reserved
1111	Either Match0 or Match1 to Final State.....Match2 to State1

The priorities described in Table 8-36 dictate that in the case of simultaneous matches, a match leading to final state has priority followed by the match on the lower channel number (0,1,2).

8.3.2.7.3 Debug State Control Register 3 (DBGSCR3)

Address: 0x0027

	7	6	5	4	3	2	1	0
R	0	0	0	0	SC3	SC2	SC1	SC0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 8-11. Debug State Control Register 3 (DBGSCR3)

Read: If COMRV[1:0] = 10

Write: If COMRV[1:0] = 10 and DBG is not armed.

This register is visible at 0x0027 only with COMRV[1:0] = 10. The state control register three selects the targeted next state whilst in State3. The matches refer to the match channels of the comparator match control logic as depicted in Figure 8-1 and described in Section 8.3.2.8.1, “Debug Comparator Control Register (DBGXCTL). Comparators must be enabled by setting the comparator enable bit in the associated DBGXCTL control register.

Table 8-19. DBGSCR3 Field Descriptions

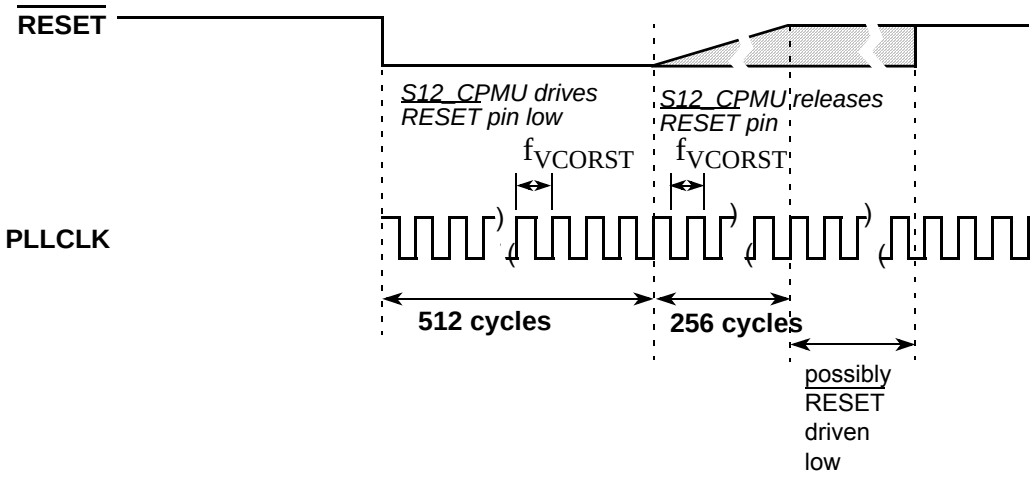
Field	Description
3–0 SC[3:0]	These bits select the targeted next state whilst in State3, based upon the match event.

Table 8-20. State3 — Sequencer Next State Selection

SC[3:0]	Description (Unspecified matches have no effect)
0000	Match0 to State1

The internal reset of the MCU remains asserted while the reset generator completes the 768 PLLCLK cycles long reset sequence. In case the RESET pin is externally driven low for more than these 768 PLLCLK cycles (External Reset), the internal reset remains asserted longer.

Figure 10-34. RESET Timing



10.5.2.1 Clock Monitor Reset

If the external oscillator is enabled (OSCE=1) in case of loss of oscillation or the oscillator frequency is below the failure assert frequency f_{CMFA} (see device electrical characteristics for values), the S12CPMU generates a Clock Monitor Reset.In Full Stop Mode the external oscillator and the clock monitor are disabled.

10.5.2.2 Computer Operating Properly Watchdog (COP) Reset

The COP (free running watchdog timer) enables the user to check that a program is running and sequencing properly. When the COP is being used, software is responsible for keeping the COP from timing out. If the COP times out it is an indication that the software is no longer being executed in the intended sequence; thus COP reset is generated.

The clock source for the COP is either ACLK, IRCCLK or OSCCLK depending on the setting of the COPOSCSEL0 and COPOSCSEL1 bit.
In Stop Mode with PSTP=1 (Pseudo Stop Mode), COPOSCSEL0=1 and COPOSCEL1=0 and PCE=1 the COP continues to run, else the COP counter halts in Stop Mode with COPOSCSEL1 =0.
In Pseudo Stop Mode and Full Stop Mode with COPOSCSEL1=1 the COP continues to run.

Table 10-28.gives an overview of the COP condition (run, static) in Stop Mode depending on legal configuration and status bit settings:

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0003	ATDCTL3	R	DJM	S8C	S4C	S2C	S1C	FIFO	FRZ1	FRZ0
		W								
0x0004	ATDCTL4	R	SMP2	SMP1	SMP0	PRS[4:0]				
		W								
0x0005	ATDCTL5	R	0	SC	SCAN	MULT	CD	CC	CB	CA
		W								
0x0006	ATDSTAT0	R	SCF	0	ETORF	FIFOR	CC3	CC2	CC1	CC0
		W								
0x0007	Unimple-mented	R	0	0	0	0	0	0	0	0
		W								
0x0008	ATDCMPEH	R	0	0	0	0	0	0	0	0
		W								
0x0009	ATDCMPEL	R	CMPE[7:0]							
		W								
0x000A	ATDSTAT2H	R	0	0	0	0	0	0	0	0
		W								
0x000B	ATDSTAT2L	R	CCF[7:0]							
		W								
0x000C	ATDDIENH	R	1	1	1	1	1	1	1	1
		W								
0x000D	ATDDIENL	R	IEN[7:0]							
		W								
0x000E	ATDCMPHTH	R	0	0	0	0	0	0	0	0
		W								
0x000F	ATDCMPHTL	R	CMPHT[7:0]							
		W								
0x0010	ATDDR0	R	See Section 12.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 12.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x0012	ATDDR1	R	See Section 12.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 12.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x0014	ATDDR2	R	See Section 12.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 12.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x0016	ATDDR3	R	See Section 12.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 12.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x0018	ATDDR4	R	See Section 12.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 12.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x001A	ATDDR5	R	See Section 12.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 12.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x001C	ATDDR6	R	See Section 12.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 12.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x001E	ATDDR7	R	See Section 12.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 12.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x0020 - 0x002F	Unimple-mented	R	0	0	0	0	0	0	0	0
		W								

= Unimplemented or Reserved

Figure 12-2. ADC12B8C Register Summary (Sheet 2 of 2)

edge or level sensitive with polarity control. Table 14-23 gives a brief description of the different combinations of control bits and their effect on the external trigger function.

In order to avoid maybe false trigger events please enable the external digital input via ATDDIEN register first and in the following enable the external trigger mode by bit ETRIGE.

Table 14-23. External Trigger Control Bits

ETRIGLE	ETRIGP	ETRIGE	SCAN	Description
X	X	0	0	Ignores external trigger. Performs one conversion sequence and stops.
X	X	0	1	Ignores external trigger. Performs continuous conversion sequences.
0	0	1	X	Trigger falling edge sensitive. Performs one conversion sequence per trigger.
0	1	1	X	Trigger rising edge sensitive. Performs one conversion sequence per trigger.
1	0	1	X	Trigger low level sensitive. Performs continuous conversions while trigger level is active.
1	1	1	X	Trigger high level sensitive. Performs continuous conversions while trigger level is active.

In either level or edge sensitive modes, the first conversion begins when the trigger is received.

Once ETRIGE is enabled a conversion must be triggered externally after writing to ATDCTL5 register.

During a conversion in edge sensitive mode, if additional trigger events are detected the overrun error flag ETORF is set.

If level sensitive mode is active and the external trigger de-asserts and later asserts again during a conversion sequence, this does not constitute an overrun. Therefore, the flag is not set. If the trigger is left active in level sensitive mode when a sequence is about to complete, another sequence will be triggered immediately.

14.4.2.2 General-Purpose Digital Port Operation

Each ATD input pin can be switched between analog or digital input functionality. An analog multiplexer makes each ATD input pin selected as analog input available to the A/D converter.

The pad of the ATD input pin is always connected to the analog input channel of the analog mulitplexer.

Each pad input signal is buffered to the digital port register.

This buffer can be turned on or off with the ATDDIEN register for each ATD input pin.
This is important so that the buffer does not draw excess current when an ATD input pin is selected as analog input to the ADC12B12C.

16.3.2.4 ATD Control Register 3 (ATDCTL3)

Writes to this register will abort current conversion sequence.

Module Base + 0x0003

	7	6	5	4	3	2	1	0
R W	DJM	S8C	S4C	S2C	S1C	FIFO	FRZ1	FRZ0
Reset	0	0	1	0	0	0	0	0

= Unimplemented or Reserved

Figure 16-6. ATD Control Register 3 (ATDCTL3)

Read: Anytime

Write: Anytime

Table 16-8. ATDCTL3 Field Descriptions

Field	Description
7 DJM	**Result Register Data Justification** — Result data format is always unsigned. This bit controls justification of conversion data in the result registers. 0 Left justified data in the result registers. 1 Right justified data in the result registers. Table 16-9 gives example ATD results for an input signal range between 0 and 5.12 Volts.
6–3 S8C, S4C, S2C, S1C	**Conversion Sequence Length** — These bits control the number of conversions per sequence. Table 16-10 shows all combinations. At reset, S4C is set to 1 (sequence length is 4). This is to maintain software continuity to HC12 family.
2 FIFO	**Result Register FIFO Mode** — If this bit is zero (non-FIFO mode), the A/D conversion results map into the result registers based on the conversion sequence; the result of the first conversion appears in the first result register (ATDDR0), the second result in the second result register (ATDDR1), and so on. If this bit is one (FIFO mode) the conversion counter is not reset at the beginning or end of a conversion sequence; sequential conversion results are placed in consecutive result registers. In a continuously scanning conversion sequence, the result register counter will wrap around when it reaches the end of the result register file. The conversion counter value (CC3-0 in ATDSTAT0) can be used to determine where in the result register file, the current conversion result will be placed. Aborting a conversion or starting a new conversion clears the conversion counter even if FIFO=1. So the first result of a new conversion sequence, started by writing to ATDCTL5, will always be place in the first result register (ATDDDR0). Intended usage of FIFO mode is continuos conversion (SCAN=1) or triggered conversion (ETRIG=1). Which result registers hold valid data can be tracked using the conversion complete flags. Fast flag clear mode may be useful in a particular application to track valid data. If this bit is one, automatic compare of result registers is always disabled, that is ADC12B16C will behave as if ACMPIE and all CPME[*n*] were zero. 0 Conversion results are placed in the corresponding result register up to the selected sequence length. 1 Conversion results are placed in consecutive result registers (wrap around at end).
1–0 FRZ[1:0]	**Background Debug Freeze Enable** — When debugging an application, it is useful in many cases to have the ATD pause when a breakpoint (Freeze Mode) is encountered. These 2 bits determine how the ATD will respond to a breakpoint as shown in Table 16-11. Leakage onto the storage node and comparator reference capacitors may compromise the accuracy of an immediately frozen conversion depending on the length of the freeze period.

18.1.3 Features

The basic features of the MSCAN are as follows:

- Implementation of the CAN protocol — Version 2.0A/B
  - — Standard and extended data frames
  - — Zero to eight bytes data length
  - — Programmable bit rate up to 1 Mbps[1]
  - — Support for remote frames
- Five receive buffers with FIFO storage scheme
- Three transmit buffers with internal prioritization using a “local priority” concept
- Flexible maskable identifier filter supports two full-size (32-bit) extended identifier filters, or four 16-bit filters, or eight 8-bit filters
- Programmable wake-up functionality with integrated low-pass filter
- Programmable loopback mode supports self-test operation
- Programmable listen-only mode for monitoring of CAN bus
- Programmable bus-off recovery functionality
- Separate signalling and interrupt capabilities for all CAN receiver and transmitter error states (warning, error passive, bus-off)
- Programmable MSCAN clock source either bus clock or oscillator clock
- Internal timer for time-stamping of received and transmitted messages
- Three low-power modes: sleep, power down, and MSCAN enable
- Global initialization of configuration registers

18.1.4 Modes of Operation

For a description of the specific MSCAN modes and the module operation related to the system operating modes refer to Section 18.4.4, “Modes of Operation”.

18.2 External Signal Description

The MSCAN uses two external pins.

NOTE

On MCUs with an integrated CAN physical interface (transceiver) the MSCAN interface is connected internally to the transceiver interface. In these cases the external availability of signals TXCAN and RXCAN is optional.

18.2.1 RXCAN — CAN Receiver Input Pin

RXCAN is the MSCAN receiver input pin.

1. Depending on the actual bit timing and the clock jitter of the PLL.

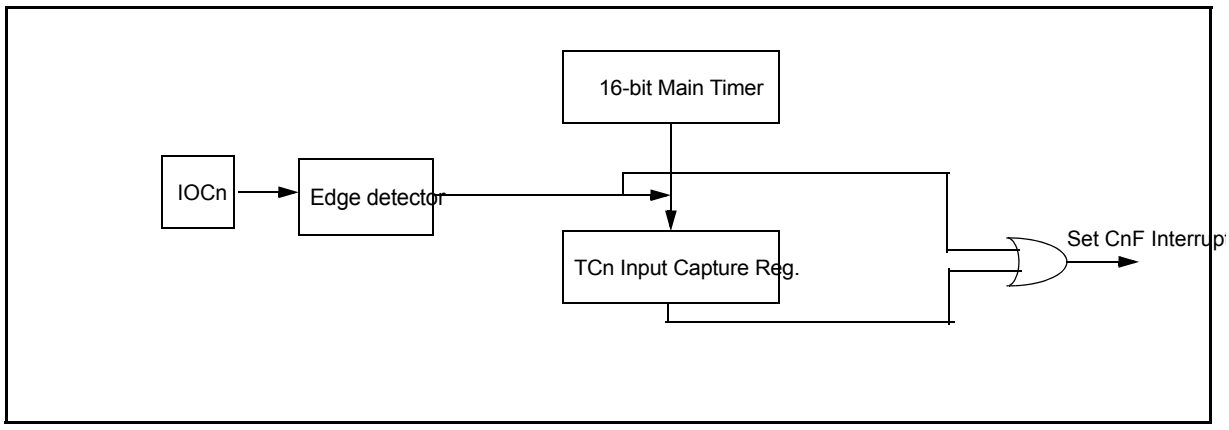


Figure 22-2. Interrupt Flag Setting

22.2 External Signal Description

The TIM16B6CV3 module has a selected number of external pins. Refer to device specification for exact number.

22.2.1 IOC5 - IOC0 — Input Capture and Output Compare Channel 5-0

Those pins serve as input capture or output compare for TIM16B6CV3 channel .

NOTE

For the description of interrupts see Section 22.6, “Interrupts”.

22.3 Memory Map and Register Definition

This section provides a detailed description of all memory and registers.

22.3.1 Module Memory Map

The memory map for the TIM16B6CV3 module is given below in Figure 22-3. The address listed for each register is the address offset. The total address for each register is the sum of the base address for the TIM16B6CV3 module and the address offset for each register.

22.3.2 Register Descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order.

Table 24-21. EEPROM Protection Address Range

DPS[4:0]	Global Address Range	Protected Size
00000	0x0_0400 – 0x0_041F	32 bytes
00001	0x0_0400 – 0x0_043F	64 bytes
00010	0x0_0400 – 0x0_045F	96 bytes
00011	0x0_0400 – 0x0_047F	128 bytes
00100	0x0_0400 – 0x0_049F	160 bytes
00101	0x0_0400 – 0x0_04BF	192 bytes
The Protection Size goes on enlarging in step of 32 bytes, for each DPS value increasing of one. . . .		
01111 - to - 11111	0x0_0400 – 0x0_05FF	512 bytes

24.3.2.11 Flash Common Command Object Register (FCCOB)

The FCCOB is an array of six words addressed via the CCOBIX index found in the FCCOBIX register. Byte wide reads and writes are allowed to the FCCOB register.

Offset Module Base + 0x000A

	7	6	5	4	3	2	1	0
R W	CCOB[15:8]							
Reset	0	0	0	0	0	0	0	0

Figure 24-15. Flash Common Command Object High Register (FCCOBHI)

Offset Module Base + 0x000B

	7	6	5	4	3	2	1	0
R W	CCOB[7:0]							
Reset	0	0	0	0	0	0	0	0

Figure 24-16. Flash Common Command Object Low Register (FCCOBLO)

24.3.2.11.1 FCCOB - NVM Command Mode

NVM command mode uses the indexed FCCOB register to provide a command code and its relevant parameters to the Memory Controller. The user first sets up all required FCCOB fields and then initiates

24.4.8 Wait Mode

The Flash module is not affected if the MCU enters wait mode. The Flash module can recover the MCU from wait via the CCIF interrupt (see Section 24.4.7, “Interrupts”).

24.4.9 Stop Mode

If a Flash command is active (CCIF = 0) when the MCU requests stop mode, the current Flash operation will be completed before the MCU is allowed to enter stop mode.

24.5 Security

The Flash module provides security information to the MCU. The Flash security state is defined by the SEC bits of the FSEC register (see Table 24-11). During reset, the Flash module initializes the FSEC register using data read from the security byte of the Flash configuration field at global address 0x3_FF0F. The security state out of reset can be permanently changed by programming the security byte assuming that the MCU is starting from a mode where the necessary P-Flash erase and program commands are available and that the upper region of the P-Flash is unprotected. If the Flash security byte is successfully programmed, its new value will take affect after the next MCU reset.

The following subsections describe these security-related subjects:

- Unsecuring the MCU using Backdoor Key Access
- Unsecuring the MCU in Special Single Chip Mode using BDM
- Mode and Security Effects on Flash Command Availability

24.5.1 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 0x3_FF00-0x3_FF07). If the KEYEN[1:0] bits are in the enabled state (see Section 24.3.2.2), the Verify Backdoor Access Key command (see Section 24.4.6.11) allows the user to present four prospective keys for comparison to the keys stored in the Flash memory via the Memory Controller. If the keys presented in the Verify Backdoor Access Key command match the backdoor keys stored in the Flash memory, the SEC bits in the FSEC register (see Table 24-11) will be changed to unsecure the MCU. Key values of 0x0000 and 0xFFFF are not permitted as backdoor keys. While the Verify Backdoor Access Key command is active, P-Flash memory and EEPROM memory will not be available for read access and will return invalid data.

25.3.2.9.1 P-Flash Protection Restrictions

The general guideline is that P-Flash protection can only be added and not removed. Table 25-21 specifies all valid transitions between P-Flash protection scenarios. Any attempt to write an invalid scenario to the FPROT register will be ignored. The contents of the FPROT register reflect the active protection scenario. See the FPHS and FPLS bit descriptions for additional restrictions.

Table 25-21. P-Flash Protection Scenario Transitions

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

1 Allowed transitions marked with X, see Figure 25-14 for a definition of the scenarios.

25.3.2.10 EEPROM Protection Register (EEPROT)

The EEPROT register defines which EEPROM sectors are protected against program and erase operations.

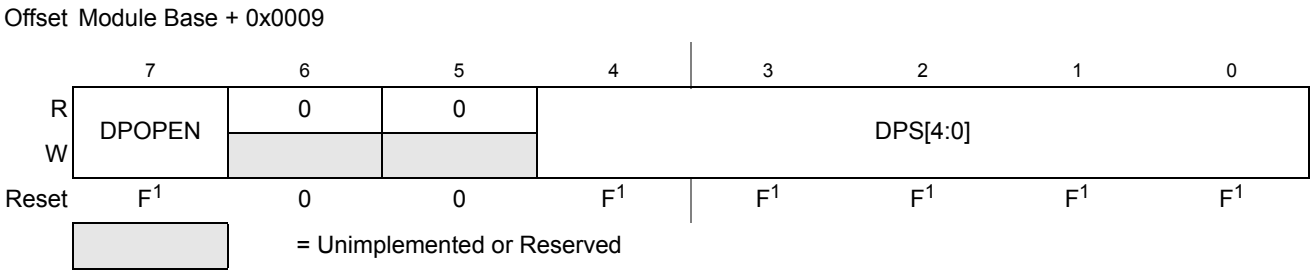


Figure 25-15. EEPROM Protection Register (EEPROT)

1 Loaded from IFR Flash configuration field, during reset sequence.

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

CCOBIX bits are readable and writable while remaining bits read 0 and are not writable.

Table 29-12. FCCOBIX Field Descriptions

Field	Description
2–0 CCOBIX[1:0]	**Common Command Register Index**— The CCOBIX bits are used to select which word of the FCCOB register array is being read or written to. See 29.3.2.11 Flash Common Command Object Register (FCCOB)," for more details.

29.3.2.4 Flash Reserved0 Register (FRSV0)

This Flash register is reserved for factory testing.

Offset 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 29-8. Flash Reserved0 Register (FRSV0)

All bits in the FRSV0 register read 0 and are not writable.

29.3.2.5 Flash Configuration Register (FCNFG)

The FCNFG register enables the Flash command complete interrupt and forces ECC faults on Flash array read access from the CPU.

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 29-9. Flash Configuration Register (FCNFG)

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

Table 29-43. Program Once Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 101 at command launch
		Set if command not available in current mode (see Table 29-27)
		Set if an invalid phrase index is supplied
		Set if the requested phrase has already been programmed[1]
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

1 If a Program Once phrase is initially programmed to 0xFFFF_FFFF_FFFF_FFFF, the Program Once command will be allowed to execute again on that same phrase.

29.4.6.7 Erase All Blocks Command

The Erase All Blocks operation will erase the entire P-Flash and EEPROM memory space.

Table 29-44. Erase All Blocks Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x08	Not required

Upon clearing CCIF to launch the Erase All Blocks command, the Memory Controller will erase the entire Flash memory space and verify that it is erased. If the Memory Controller verifies that the entire Flash memory space was properly erased, security will be released. During the execution of this command (CCIF=0) the user must not write to any Flash module register. The CCIF flag will set after the Erase All Blocks operation has completed.

Table 29-45. Erase All Blocks Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 000 at command launch
		Set if command not available in current mode (see Table 29-27)
	FPVIOL	Set if any area of the P-Flash or EEPROM memory is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

29.4.6.8 Erase Flash Block Command

The Erase Flash Block operation will erase all addresses in a P-Flash or EEPROM block.

29.4.7 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 29-66. 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.

29.4.7.1 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 29.3.2.5, “Flash Configuration Register (FCNFG)”, Section 29.3.2.6, “Flash Error Configuration Register (FERCNFG)”, Section 29.3.2.7, “Flash Status Register (FSTAT)”, and Section 29.3.2.8, “Flash Error Status Register (FERSTAT)”.

The logic used for generating the Flash module interrupts is shown in Figure 29-27.

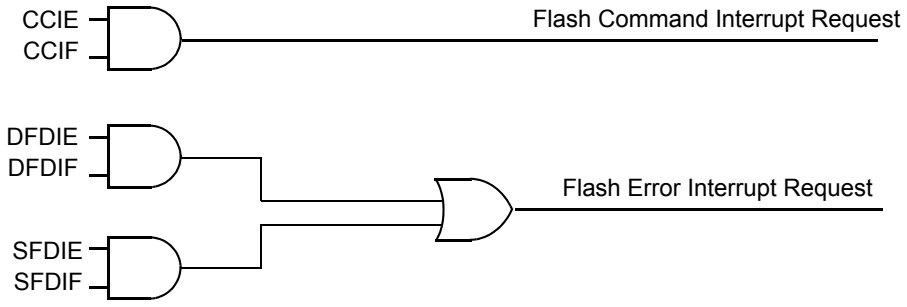


Figure 29-27. Flash Module Interrupts Implementation

Table 30-15. FSTAT Field Descriptions (continued)

Field	Description
3 MGBUSY	**Memory Controller Busy Flag** — The MGBUSY flag reflects the active state of the Memory Controller. 0 Memory Controller is idle 1 Memory Controller is busy executing a Flash command (CCIF = 0)
2 RSVD	**Reserved Bit** — This bit is reserved and always reads 0.
1–0 MGSTAT[1:0]	**Memory Controller Command Completion Status Flag** — One or more MGSTAT flag bits are set if an error is detected during execution of a Flash command or during the Flash reset sequence. See Section 30.4.6, “Flash Command Description,” and Section 30.6, “Initialization” for details.

30.3.2.8 Flash Error Status Register (FERSTAT)

The FERSTAT register reflects the error status of internal Flash operations.

Offset Module Base + 0x0007

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

= Unimplemented or Reserved

Figure 30-12. Flash Error Status Register (FERSTAT)

All flags in the FERSTAT register are readable and only writable to clear the flag.

Table 30-16. FERSTAT Field Descriptions

Field	Description
1 DFDIF	**Double Bit Fault Detect Interrupt Flag** — The setting of the DFDIF flag indicates that a double bit fault was detected in the stored parity and data bits during a Flash array read operation or that a Flash array read operation returning invalid data was attempted on a Flash block that was under a Flash command operation.[1] The DFDIF flag is cleared by writing a 1 to DFDIF. Writing a 0 to DFDIF has no effect on DFDIF.[2] 0 No double bit fault detected 1 Double bit fault detected or a Flash array read operation returning invalid data was attempted while command running
0 SFDIF	**Single Bit Fault Detect Interrupt Flag** — With the IGNSF bit in the FCNFG register clear, the SFDIF flag indicates that a single bit fault was detected in the stored parity and data bits during a Flash array read operation or that a Flash array read operation returning invalid data was attempted on a Flash block that was under a Flash command operation.[1] The SFDIF flag is cleared by writing a 1 to SFDIF. Writing a 0 to SFDIF has no effect on SFDIF. 0 No single bit fault detected 1 Single bit fault detected and corrected or a Flash array read operation returning invalid data was attempted while command running

1 The single bit fault and double bit fault flags are mutually exclusive for parity errors (an ECC fault occurrence can be either single fault or double fault but never both). A simultaneous access collision (Flash array read operation returning invalid data attempted while command running) is indicated when both SFDIF and DFDIF flags are high.

2 There is a one cycle delay in storing the ECC DFDIF and SFDIF fault flags in this register. At least one NOP is required after a flash memory read before checking FERSTAT for the occurrence of ECC errors.

30.3.2.9 P-Flash Protection Register (FPROT)

The FPROT register defines which P-Flash sectors are protected against program and erase operations.

Offset Module Base + 0x0008

	7	6	5	4	3	2	1	0
R	FPOPEN	RNV6	FPHDIS	FPHS[1:0]		FPLDIS	FPLS[1:0]	
W								
Reset	F[1]	F[1]	F[1]	F[1]	F[1]	F[1]	F[1]	F[1]

= Unimplemented or Reserved

Figure 30-13. Flash Protection Register (FPROT)

[1] Loaded from IFR Flash configuration field, during reset sequence.

The (unreserved) bits of the FPROT register are writable with the restriction that the size of the protected region can only be increased (see Section 30.3.2.9.1, “P-Flash Protection Restrictions,” and Table 30-21).

During the reset sequence, the FPROT register is loaded with the contents of the P-Flash protection byte in the Flash configuration field at global address 0x3_FF0C located in P-Flash memory (see Table 30-4) as indicated by reset condition ‘F’ in Figure 30-13. To change the P-Flash protection that will be loaded during the reset sequence, the upper sector of the P-Flash memory must be unprotected, then the P-Flash protection byte must be reprogrammed. If a double bit fault is detected while reading the P-Flash phrase containing the P-Flash protection byte during the reset sequence, the FPOPEN bit will be cleared and remaining bits in the FPROT register will be set to leave the P-Flash memory fully protected.

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

Table 30-17. FPROT Field Descriptions

Field	Description
7 FPOPEN	**Flash Protection Operation Enable** — The FPOPEN bit determines the protection function for program or erase operations as shown in Table 30-18 for the P-Flash block. 0 When FPOPEN is clear, the FPHDIS and FPLDIS bits define unprotected address ranges as specified by the corresponding FPHS and FPLS bits 1 When FPOPEN is set, the FPHDIS and FPLDIS bits enable protection for the address range specified by the corresponding FPHS and FPLS bits
6 RNV[6]	**Reserved Nonvolatile Bit** — The RNV bit should remain in the erased state for future enhancements.
5 FPHDIS	**Flash Protection Higher Address Range Disable** — The FPHDIS bit determines whether there is a protected/unprotected area in a specific region of the P-Flash memory ending with global address 0x3_FFFF. 0 Protection/Unprotection enabled 1 Protection/Unprotection disabled
4–3 FPHS[1:0]	**Flash Protection Higher Address Size** — The FPHS bits determine the size of the protected/unprotected area in P-Flash memory as shown inTable 30-19. The FPHS bits can only be written to while the FPHDIS bit is set.

30.4.6.2 Erase Verify Block Command

The Erase Verify Block command allows the user to verify that an entire P-Flash or EEPROM block has been erased. The FCCOB FlashBlockSelectionCode[1:0]bits determine which block must be verified.

Table 30-33. Erase Verify Block Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x02	Flash block selection code [1:0]. See Table 30-34

Table 30-34. Flash block selection code description

Selection code[1:0]	Flash block to be verified
00	EEPROM
01	P-Flash
10	P-Flash
11	P-Flash

Upon clearing CCIF to launch the Erase Verify Block command, the Memory Controller will verify that the selected P-Flash or EEPROM block is erased. The CCIF flag will set after the Erase Verify Block operation has completed.If the block is not erased, it means blank check failed, both MGSTAT bits will be set.

Table 30-35. Erase Verify Block Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 000 at command launch.
	FPVIOL	None.
	MGSTAT1	Set if any errors have been encountered during the read or if blank check failed.
	MGSTAT0	Set if any non-correctable errors have been encountered during the read[1] or if blank check failed.

30.4.6.3 Erase Verify P-Flash Section Command

The Erase Verify P-Flash Section command will verify that a section of code in the P-Flash memory is erased. The Erase Verify P-Flash Section command defines the starting point of the code to be verified and the number of phrases.

Chapter 31
240 KByte Flash Module (S12FTMRG240K2V1)

Table 31-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V01.06	23 Jun 2010	31.4.6.2/31-1159 31.4.6.12/31-1166 31.4.6.13/31-1167	Updated description of the commands RD1BLK, MLOADU and MLOADF
V01.07	20 aug 2010	31.4.6.2/31-1159 31.4.6.12/31-1166 31.4.6.13/31-1167	Updated description of the commands RD1BLK, MLOADU and MLOADF
Rev.1.27	31 Jan 2011	31.3.2.9/31-1142	Updated description of protection on Section 31.3.2.9

31.1 Introduction

The FTMRG240K2 module implements the following:

- 240Kbytes of P-Flash (Program Flash) memory
- 4Kbytes of EEPROM memory

The Flash memory is ideal for single-supply applications allowing for field reprogramming without requiring external high voltage sources for program or erase operations. The Flash module includes a memory controller that executes commands to modify Flash memory contents. The user interface to the memory controller consists of the indexed Flash Common Command Object (FCCOB) register which is written to with the command, global address, data, and any required command parameters. The memory controller must complete the execution of a command before the FCCOB register can be written to with a new command.

CAUTION

A Flash word or phrase must be in the erased state before being programmed. Cumulative programming of bits within a Flash word or phrase is not allowed.

Offset Module Base + 0x0000

	7	6	5	4	3	2	1	0
R	FDIVLD	FDIVLCK	FDIV[5:0]					
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 31-5. Flash Clock Divider Register (FCLKDIV)

All bits in the FCLKDIV register are readable, bit 7 is not writable, bit 6 is write-once-hi and controls the writability of the FDIV field in normal mode. In special mode, bits 6-0 are writable any number of times but bit 7 remains unwritable.

CAUTION

The FCLKDIV register should never be written while a Flash command is executing (CCIF=0).

Table 31-7. FCLKDIV Field Descriptions

Field	Description
7 FDIVLD	Clock Divider Loaded 0 FCLKDIV register has not been written since the last reset 1 FCLKDIV register has been written since the last reset
6 FDIVLCK	Clock Divider Locked 0 FDIV field is open for writing 1 FDIV value is locked and cannot be changed. Once the lock bit is set high, only reset can clear this bit and restore writability to the FDIV field in normal mode.
5–0 FDIV[5:0]	**Clock Divider Bits** — FDIV[5:0] must be set to effectively divide BUSCLK down to 1 MHz to control timed events during Flash program and erase algorithms. Table 31-8 shows recommended values for FDIV[5:0] based on the BUSCLK frequency. Please refer to Section 31.4.4, “Flash Command Operations,” for more information.