

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	40
Program Memory Size	192KB (192K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	11K x 8
Voltage - Supply (Vcc/Vdd)	3.13V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12g192f0vlf

- 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-bit¹ analog-to-digital converter

- 3 μ s conversion time
- 8-/10¹-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 2-19. Block Register Map (G1) (continued)

Global Address Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x000A–0x000B Non-PIM Address Range	R W	Non-PIM Address Range							
0x000C PUCR	R W	0	BKPUE	0	PDPEE	PUPDE	PUPCE	PUPBE	PUPAE
0x000D Reserved	R W	0	0	0	0	0	0	0	0
0x000E–0x001B Non-PIM Address Range	R W	Non-PIM Address Range							
0x001C ECLKCTL	R W	NECLK	NCLKX2	DIV16	EDIV4	EDIV3	EDIV2	EDIV1	EDIV0
0x001D Reserved	R W	0	0	0	0	0	0	0	0
0x001E IRQCR	R W	IRQE	IRQEN	0	0	0	0	0	0
0x001F Reserved	R W	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0x0020–0x023F Non-PIM Address Range	R W	Non-PIM Address Range							
0x0240 PTT	R W	PTT7	PTT6	PTT5	PTT4	PTT3	PTT2	PTT1	PTT0
0x0241 PTIT	R W	PTIT7	PTIT6	PTIT5	PTIT4	PTIT3	PTIT2	PTIT1	PTIT0
0x0242 DDRT	R W	DDRT7	DDRT6	DDRT5	DDRT4	DDRT3	DDRT2	DDRT1	DDRT0
0x0243 Reserved	R W	0	0	0	0	0	0	0	0
0x0244 PERT	R W	PERT7	PERT6	PERT5	PERT4	PERT3	PERT2	PERT1	PERT0
0x0245 PPST	R W	PPST7	PPST6	PPST5	PPST4	PPST3	PPST2	PPST1	PPST0
		= Unimplemented or Reserved							

2.4.3.39 Port P Interrupt Enable Register (PIEP)

Read: Anytime

Address 0x025E (G1, G2)

Access: User read/write¹

	7	6	5	4	3	2	1	0
R	PIEP7	PIEP6	PIEP5	PIEP4	PIEP3	PIEP2	PIEP1	PIEP0
W								
Reset	0	0	0	0	0	0	0	0

Address 0x025E (G3)

Access: User read/write¹

	7	6	5	4	3	2	1	0
R	0	0	PIEP5	PIEP4	PIEP3	PIEP2	PIEP1	PIEP0
W								
Reset	0	0	0	0	0	0	0	0

Figure 2-40. Port P Interrupt Enable Register (PIEP)

¹ Read: Anytime
Write: Anytime

Table 2-66. PIEP Register Field Descriptions

Field	Description
7-0 PIEP	Port P interrupt enable— This bit enables or disables the edge sensitive pin interrupt on the associated pin. An interrupt can be generated if the pin is operating in input or output mode when in use with the general-purpose or related peripheral function. 1 Interrupt is enabled 0 Interrupt is disabled (interrupt flag masked)

2.4.3.40 Port P Interrupt Flag Register (PIFP)

Address 0x025F (G1, G2)

Access: User read/write¹

	7	6	5	4	3	2	1	0
R	PIFP7	PIFP6	PIFP5	PIFP4	PIFP3	PIFP2	PIFP1	PIFP0
W								
Reset	0	0	0	0	0	0	0	0

Address 0x025F (G3)

Access: User read/write¹

	7	6	5	4	3	2	1	0
R	0	0	PIFP5	PIFP4	PIFP3	PIFP2	PIFP1	PIFP0
W								
Reset	0	0	0	0	0	0	0	0

Figure 2-41. Port P Interrupt Flag Register (PIFP)

Chapter 4

Reference Voltage Attenuator (RVAV1)

Revision History

Rev. No. (Item No.)	Date (Submitted By)	Sections Affected	Substantial Change(s)
V00.05	09 Jun 2010		<ul style="list-style-type: none">Added appendix title in note to reference reduced ADC clockOrthographical corrections aligned to Freescale Publications Style Guide
V00.06	01 Jul 2010		<ul style="list-style-type: none">Aligned to S12 register guidelines
V01.00	18 Oct 2010		<ul style="list-style-type: none">Initial version

4.1 Introduction

The reference voltage attenuator (RVA) provides a circuit for reduction of the ADC reference voltage difference VRH-VSSA to gain more ADC resolution.

4.2 Features

The RVA has the following features:

- Attenuation of ADC reference voltage with low long-term drift

4.3 Block Diagram

The block diagram of the RVA module is shown below.

Refer to device overview section “ADC VRH/VRL Signal Connection” for connection of RVA to pins and ADC module.

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 set¹.

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.

Hardware commands are used to read and write target system memory locations and to enter active background debug mode, see [Section 7.4.3, “BDM Hardware Commands”](#). Target system memory includes all memory that is accessible by the CPU.

Firmware commands are used to read and write CPU resources and to exit from active background debug mode, see [Section 7.4.4, “Standard BDM Firmware Commands”](#). The CPU resources referred to are the accumulator (D), X index register (X), Y index register (Y), stack pointer (SP), and program counter (PC).

Hardware commands can be executed at any time and in any mode excluding a few exceptions as highlighted (see [Section 7.4.3, “BDM Hardware Commands”](#)) and in secure mode (see [Section 7.4.1, “Security”](#)). BDM firmware commands can only be executed when the system is not secure and is in active background debug mode (BDM).

7.4.1 Security

If the user resets into special single chip mode with the system secured, a secured mode BDM firmware lookup table is brought into the map overlapping a portion of the standard BDM firmware lookup table. The secure BDM firmware verifies that the on-chip Flash EEPROM are erased. This being the case, the UNSEC and ENBDM bit will get set. The BDM program jumps to the start of the standard BDM firmware and the secured mode BDM firmware is turned off and all BDM commands are allowed. If the Flash does not verify as erased, the BDM firmware sets the ENBDM bit, without asserting UNSEC, and the firmware enters a loop. This causes the BDM hardware commands to become enabled, but does not enable the firmware commands. This allows the BDM hardware to be used to erase the Flash.

BDM operation is not possible in any other mode than special single chip mode when the device is secured. The device can only be unsecured via BDM serial interface in special single chip mode. For more information regarding security, please see the S12S_9SEC Block Guide.

7.4.2 Enabling and Activating BDM

The system must be in active BDM to execute standard BDM firmware commands. BDM can be activated only after being enabled. BDM is enabled by setting the ENBDM bit in the BDM status (BDMSTS) register. The ENBDM bit is set by writing to the BDM status (BDMSTS) register, via the single-wire interface, using a hardware command such as WRITE_BD_BYTE.

After being enabled, BDM is activated by one of the following¹:

- Hardware BACKGROUND command
- CPU BGND instruction
- Breakpoint force or tag mechanism²

When BDM is activated, the CPU finishes executing the current instruction and then begins executing the firmware in the standard BDM firmware lookup table. When BDM is activated by a breakpoint, the type of breakpoint used determines if BDM becomes active before or after execution of the next instruction.

1. BDM is enabled and active immediately out of special single-chip reset.

2. This method is provided by the S12S_DBG module.

storage. The information bits indicate the size of access (word or byte) and the type of access (read or write).

When tracing in Detail Mode, all cycles are traced except those when the CPU is either in a free or opcode fetch cycle.

8.4.5.2.4 Compressed Pure PC Mode

In Compressed Pure PC Mode, the PC addresses of all executed opcodes, including illegal opcodes are stored. A compressed storage format is used to increase the effective depth of the trace buffer. This is achieved by storing the lower order bits each time and using 2 information bits to indicate if a 64 byte boundary has been crossed, in which case the full PC is stored.

Each Trace Buffer row consists of 2 information bits and 18 PC address bits

NOTE:

When tracing is terminated using forced breakpoints, latency in breakpoint generation means that opcodes following the opcode causing the breakpoint can be stored to the trace buffer. The number of opcodes is dependent on program flow. This can be avoided by using tagged breakpoints.

8.4.5.3 Trace Buffer Organization (Normal, Loop1, Detail modes)

ADRH, ADRM, ADRL denote address high, middle and low byte respectively. The numerical suffix refers to the tracing count. The information format for Loop1 and Normal modes is identical. In Detail mode, the address and data for each entry are stored on consecutive lines, thus the maximum number of entries is 32. In this case DBG CNT bits are incremented twice, once for the address line and once for the data line, on each trace buffer entry. In Detail mode CINF comprises of R/W and size access information (CRW and CSZ respectively).

Single byte data accesses in Detail Mode are always stored to the low byte of the trace buffer (DATA1) and the high byte is cleared. When tracing word accesses, the byte at the lower address is always stored to trace buffer byte1 and the byte at the higher address is stored to byte0.

Table 8-37. Trace Buffer Organization (Normal, Loop1, Detail modes)

Mode	Entry Number	4-bits	8-bits	8-bits
		Field 2	Field 1	Field 0
Detail Mode	Entry 1	CINF1,ADRH1	ADRM1	ADRL1
		0	DATAH1	DATAL1
	Entry 2	CINF2,ADRH2	ADRM2	ADRL2
		0	DATAH2	DATAL2
Normal/Loop1 Modes	Entry 1	PCH1	PCM1	PCL1
	Entry 2	PCH2	PCM2	PCL2

0x003E

	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 10-14. Reserved Register (CPMUTEST1)

Read: Anytime

Write: Only in Special Mode

10.3.2.12 S12CPMU COP Timer Arm/Reset Register (CPMUARMCOP)

This register is used to restart the COP time-out period.

0x003F

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

Figure 10-15. S12CPMU CPMUARMCOP Register

Read: Always reads \$00

Write: Anytime

When the COP is disabled (CR[2:0] = “000”) writing to this register has no effect.

When the COP is enabled by setting CR[2:0] nonzero, the following applies:

Writing any value other than \$55 or \$AA causes a COP reset. To restart the COP time-out period write \$55 followed by a write of \$AA. These writes do not need to occur back-to-back, but the sequence (\$55, \$AA) must be completed prior to COP end of time-out period to avoid a COP reset. Sequences of \$55 writes are allowed. When the WCOP bit is set, \$55 and \$AA writes must be done in the last 25% of the selected time-out period; writing any value in the first 75% of the selected period will cause a COP reset.

10.3.2.13 Low Voltage Control Register (CPMULVCTL)

The CPMULVCTL register allows the configuration of the low-voltage detect features.

11.3.2.9 ATD Status Register 2 (ATDSTAT2)

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

Module Base + 0x000A

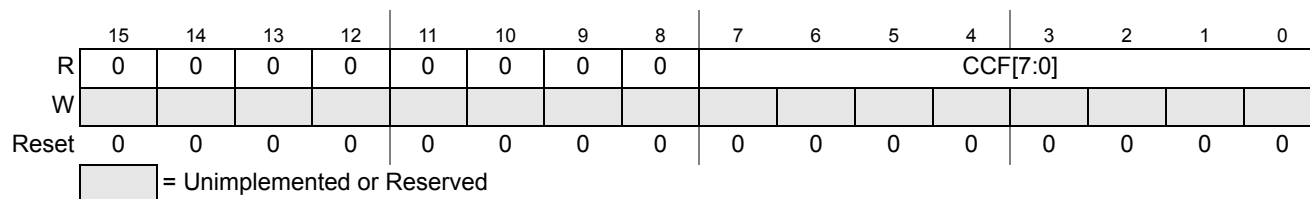


Figure 11-11. ATD Status Register 2 (ATDSTAT2)

Read: Anytime

Write: Anytime (for details see [Table 11-18](#) below)

Table 11-18. ATDSTAT2 Field Descriptions

Field	Description
7–0 CCF[7:0]	<p>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.</p> <p>If automatic compare of conversion results is enabled (CMPE[n]=1 in ATDCMPE), the conversion complete flag is only set if comparison with ATDDRn is true. If ACMPIE=1 a compare interrupt will be requested. In this case, as the ATDDRn 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.</p> <p>A flag CCF[n] is cleared when one of the following occurs:</p> <ul style="list-style-type: none"> 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 ATDDRn D) If AFFC=1 and CMPE[n]=1, write to result register ATDDRn <p>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.</p> <p>0 Conversion number n not completed or successfully compared</p> <p>1 If (CMPE[n]=0): Conversion number n has completed. Result is ready in ATDDRn.</p> <p>If (CMPE[n]=1): Compare for conversion result number n with compare value in ATDDRn, using compare operator CMPGT[n] is true. (No result available in ATDDRn)</p>

12.1.2 Modes of Operation

12.1.2.1 Conversion Modes

There is software programmable selection between performing **single** or **continuous conversion** on a **single channel** or **multiple channels**.

12.1.2.2 MCU Operating Modes

- **Stop Mode**
Entering Stop Mode aborts any conversion sequence in progress and if a sequence was aborted restarts it after exiting stop mode. This has the same effect/consequences as starting a conversion sequence with write to ATDCTL5. So after exiting from stop mode with a previously aborted sequence all flags are cleared etc.
- **Wait Mode**
ADC12B8C behaves same in Run and Wait Mode. For reduced power consumption continuous conversions should be aborted before entering Wait mode.
- **Freeze Mode**
In Freeze Mode the ADC12B8C will either continue or finish or stop converting according to the FRZ1 and FRZ0 bits. This is useful for debugging and emulation.

17.4.2.2 Analog Output Voltage Level Register (DACVOL)

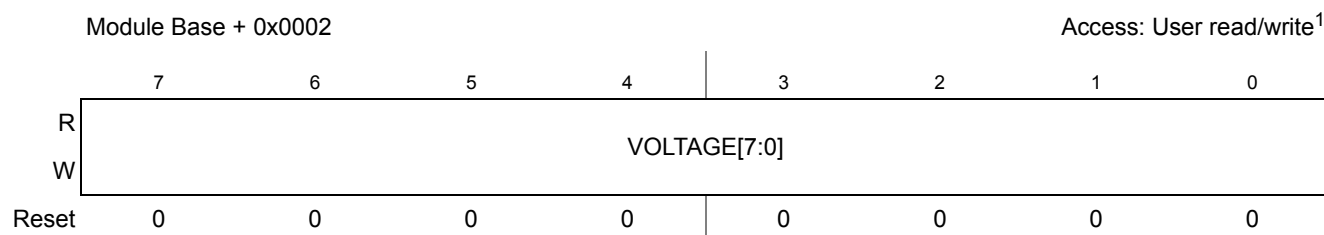


Figure 17-4. Analog Output Voltage Level Register (DACVOL)

¹ Read: Anytime
Write: Anytime

Table 17-4. DACVOL Field Description

Field	Description
7:0 VOLTAGE[7:0]	VOLTAGE — This register defines (together with the FVR bit) the analog output voltage. For more detail see Equation 17-1 and Equation 17-2 .

17.4.2.3 Reserved Register

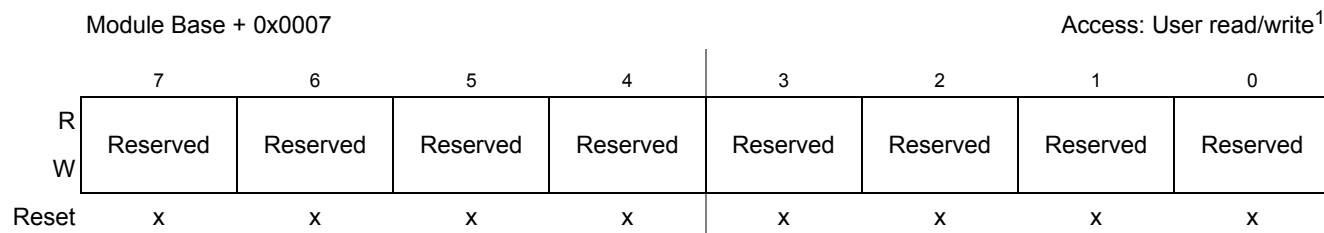


Figure 17-5. Reserved Registerfv_dac_8b5v_RESERVED

¹ Read: Anytime
Write: Only in special mode

17.5 Functional Description

17.5.1 Functional Overview

The DAC resistor network and the operational amplifier can be used together or stand alone. Following modes are supported:

Table 17-5. DAC Modes of Operation

DACM[2:0]		Description			
		Submodules		Output	
		DAC resistor network	Operational Amplifier	DACU	AMP
Off	000	disabled	disabled	disconnected	disconnected

18.2.2 TXCAN — CAN Transmitter Output Pin

TXCAN is the MSCAN transmitter output pin. The TXCAN output pin represents the logic level on the CAN bus:

- 0 = Dominant state
- 1 = Recessive state

18.2.3 CAN System

A typical CAN system with MSCAN is shown in [Figure 18-2](#). Each CAN station is connected physically to the CAN bus lines through a transceiver device. The transceiver is capable of driving the large current needed for the CAN bus and has current protection against defective CAN or defective stations.

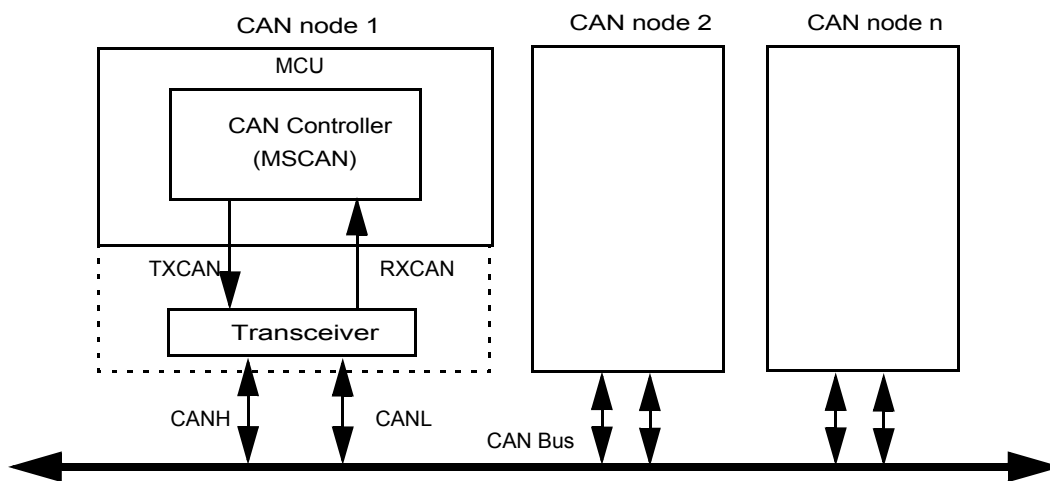


Figure 18-2. CAN System

18.3 Memory Map and Register Definition

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

18.3.1 Module Memory Map

[Figure 18-3](#) gives an overview on all registers and their individual bits in the MSCAN memory map. The *register address* results from the addition of *base address* and *address offset*. The *base address* is determined at the MCU level and can be found in the MCU memory map description. The *address offset* is defined at the module level.

The MSCAN occupies 64 bytes in the memory space. The base address of the MSCAN module is determined at the MCU level when the MCU is defined. The register decode map is fixed and begins at the first address of the module address offset.

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.

user-supplied keys match those stored in the Flash security bytes of the Flash configuration field (see [Table 25-4](#)). The Verify Backdoor Access Key command must not be executed from the Flash block containing the backdoor comparison key to avoid code runaway.

Table 25-52. Verify Backdoor Access Key Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x0C	Not required
001	Key 0	
010	Key 1	
011	Key 2	
100	Key 3	

Upon clearing CCIF to launch the Verify Backdoor Access Key command, the Memory Controller will check the FSEC KEYEN bits to verify that this command is enabled. If not enabled, the Memory Controller sets the ACCERR bit in the FSTAT register and terminates. If the command is enabled, the Memory Controller compares the key provided in FCCOB to the backdoor comparison key in the Flash configuration field with Key 0 compared to 0x3_FF00, etc. If the backdoor keys match, security will be released. If the backdoor keys do not match, security is not released and all future attempts to execute the Verify Backdoor Access Key command are aborted (set ACCERR) until a reset occurs. The CCIF flag is set after the Verify Backdoor Access Key operation has completed.

Table 25-53. Verify Backdoor Access Key Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 100 at command launch
		Set if an incorrect backdoor key is supplied
		Set if backdoor key access has not been enabled (KEYEN[1:0] != 10, see Section 25.3.2.2)
		Set if the backdoor key has mismatched since the last reset
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None

25.4.6.12 Set User Margin Level Command

The Set User Margin Level command causes the Memory Controller to set the margin level for future read operations of the P-Flash or EEPROM block.

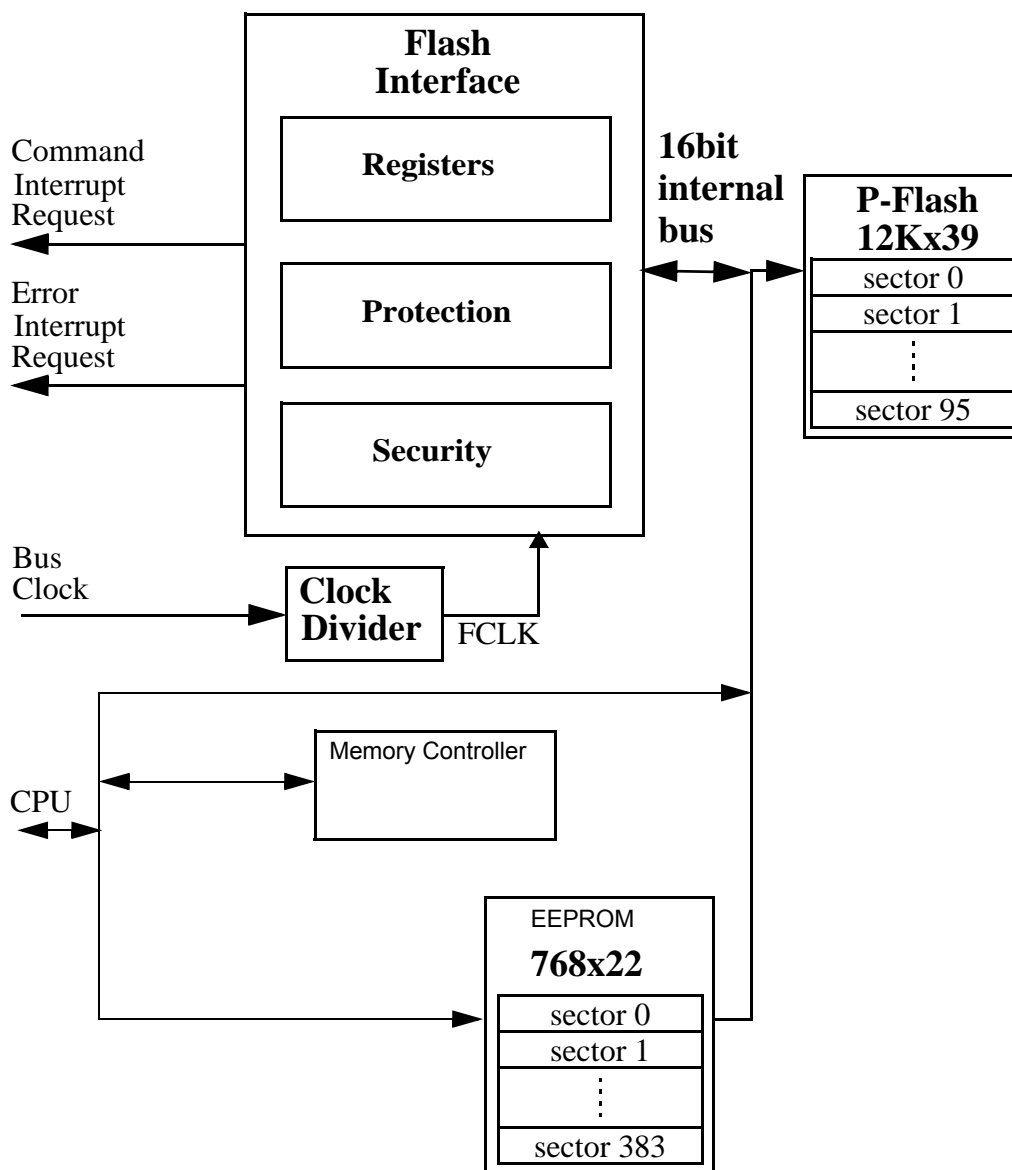
Table 25-54. Set User Margin Level Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x0D	Flash block selection code [1:0]. See Table 25-34

26.1.3 Block Diagram

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

Figure 26-1. FTMRG48K1 Block Diagram



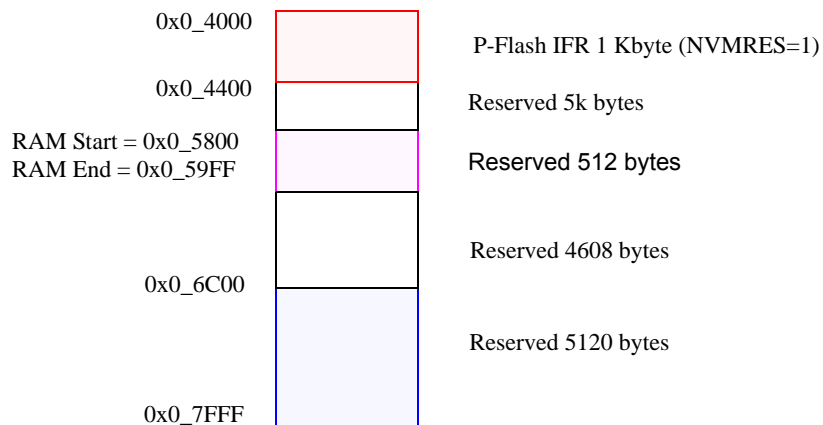
26.2 External Signal Description

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

Table 26-6. Memory Controller Resource Fields (NVMRES¹=1)

Global Address	Size (Bytes)	Description
0x0_4000 – 0x040FFF	256	P-Flash IFR (see Table 26-5)
0x0_4100 – 0x0_41FF	256	Reserved.
0x0_4200 – 0x0_57FF		Reserved
0x0_5800 – 0x0_59FF	512	Reserved
0x0_5A00 – 0x0_5FFF	1,536	Reserved
0x0_6000 – 0x0_6BFF	3,072	Reserved
0x0_6C00 – 0x0_7FFF	5,120	Reserved

¹ NVMRES - See [Section 26.4.3](#) for NVMRES (NVM Resource) detail.

**Figure 26-3. Memory Controller Resource Memory Map (NVMRES=1)**

26.3.2 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 Flash command execution (for more detail, see Caution note in [Section 26.3](#)).

29.4.6.2 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 29-33. Erase Verify Block Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x02	Flash block selection code [1:0]. See Table 29-34

Table 29-34. Flash block selection code description

Selection code[1:0]	Flash block to be verified
00	EEPROM
01	Invalid (ACCERR)
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 29-35. Erase Verify Block Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 000 at command launch
		Set if an invalid FlashBlockSelectionCode[1:0] is supplied
	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 or if blank check failed.

29.4.6.3 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.

Table 30-24. FCCOB - NVM Command Mode (Typical Usage)

CCOBIX[2:0]	Byte	FCCOB Parameter Fields (NVM Command Mode)
010	HI	Data 0 [15:8]
	LO	Data 0 [7:0]
011	HI	Data 1 [15:8]
	LO	Data 1 [7:0]
100	HI	Data 2 [15:8]
	LO	Data 2 [7:0]
101	HI	Data 3 [15:8]
	LO	Data 3 [7:0]

30.3.2.12 Flash Reserved1 Register (FRSV1)

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 30-18. Flash Reserved1 Register (FRSV1)

All bits in the FRSV1 register read 0 and are not writable.

30.3.2.13 Flash Reserved2 Register (FRSV2)

This Flash register is reserved for factory testing.

Offset Module Base + 0x000D

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


 = Unimplemented or Reserved

Figure 30-19. Flash Reserved2 Register (FRSV2)

All bits in the FRSV2 register read 0 and are not writable.

30.3.2.14 Flash Reserved3 Register (FRSV3)

This Flash register is reserved for factory testing.

Table 30-49. Erase P-Flash Sector 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 global address [17:16] is supplied see Table 30-3)
		Set if a misaligned phrase address is supplied (global address [2:0] != 000)
	FPVIOL	Set if the selected P-Flash sector 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

30.4.6.10 Unsecure Flash Command

The Unsecure Flash command will erase the entire P-Flash and EEPROM memory space and, if the erase is successful, will release security.

Table 30-50. Unsecure Flash Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x0B	Not required

Upon clearing CCIF to launch the Unsecure Flash command, the Memory Controller will erase the entire P-Flash and EEPROM 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. If the erase verify is not successful, the Unsecure Flash operation sets MGSTAT1 and terminates without changing the security state. During the execution of this command (CCIF=0) the user must not write to any Flash module register. The CCIF flag is set after the Unsecure Flash operation has completed.

Table 30-51. Unsecure Flash 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 30-27)
	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

30.4.6.11 Verify Backdoor Access Key Command

The Verify Backdoor Access Key command will only execute if it is enabled by the KEYEN bits in the FSEC register (see [Table 30-10](#)). The Verify Backdoor Access Key command releases security if user-supplied keys match those stored in the Flash security bytes of the Flash configuration field (see