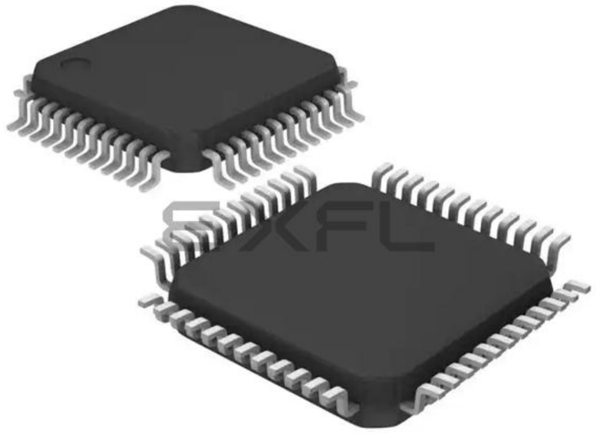


Welcome to E-XFL.COM

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	48KB (48K x 8)
Program Memory Type	FLASH
EEPROM Size	1.5K 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 ~ 125°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/s9s12g48f0mlf

Chapter 6
Interrupt Module (S12SINTV1)

Chapter 7
Background Debug Module (S12SBDMV1)

2.4.3.43 Port J Input Register (PTIJ)

Address 0x0269 (G1, G2) Access: User read only[1]

	7	6	5	4	3	2	1	0
R	PTIJ7	PTIJ6	PTIJ5	PTIJ4	PTIJ3	PTIJ2	PTIJ1	PTIJ0
W								
Reset	0	0	0	0	0	0	0	0

Address 0x0269 (G3) Access: User read only[1]

	7	6	5	4	3	2	1	0
R	0	0	0	0	PTIJ3	PTIJ2	PTIJ1	PTIJ0
W								
Reset	0	0	0	0	0	0	0	0

Figure 2-43. Port J Input Register (PTIJ)

[1] Read: Anytime
Write:Never

Table 2-69. PTIJ Register Field Descriptions

Field	Description
7-0 PTIJ	**Port J input data—** A read always returns the buffered input state of the associated pin. It can be used to detect overload or short circuit conditions on output pins.

2.4.3.44 Port J Data Direction Register (DDRJ)

Address 0x026A (G1, G2) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	DDRJ7	DDRJ6	DDRJ5	DDRJ4	DDRJ3	DDRJ2	DDRJ1	DDRJ0
Reset	0	0	0	0	0	0	0	0

Address 0x026A (G3) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	0	0	DDRJ3	DDRJ2	DDRJ1	DDRJ0
W								
Reset	0	0	0	0	0	0	0	0

Figure 2-44. Port J Data Direction Register (DDRJ)

[1] Read: Anytime
Write: Anytime

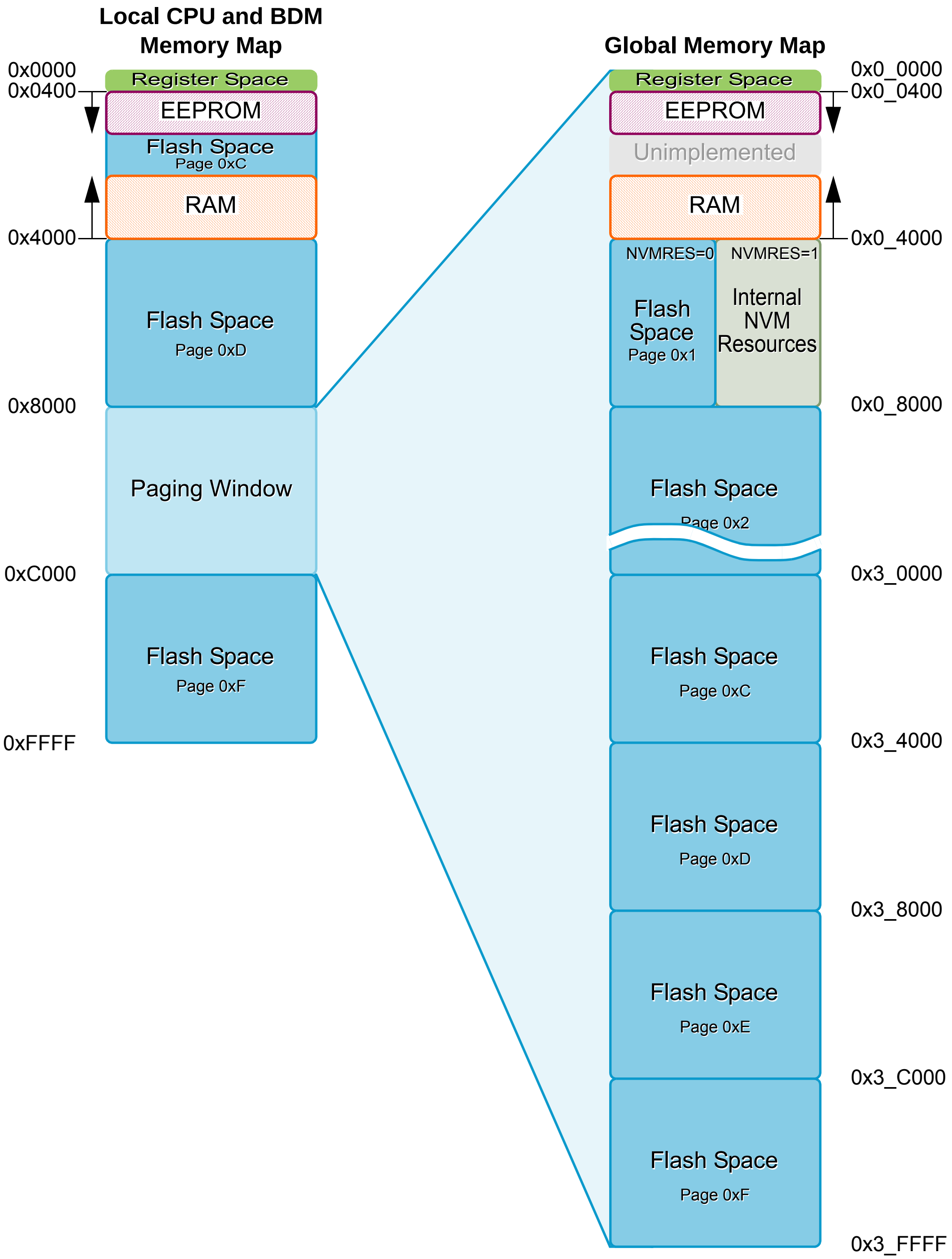


Figure 5-11. Local to Global Address Mapping

NOTE

> The ACK pulse does not provide a time out. This means for the GO_UNTIL command that it can not be distinguished if a stop or wait has been executed (command discarded and ACK not issued) or if the "UNTIL" condition (BDM active) is just not reached yet. Hence in any case where the ACK pulse of a command is not issued the possible pending command should be aborted before issuing a new command. See the handshake abort procedure described in Section 7.4.8, "Hardware Handshake Abort Procedure".

7.4.8 Hardware Handshake Abort Procedure

The abort procedure is based on the SYNC command. In order to abort a command, which had not issued the corresponding ACK pulse, the host controller should generate a low pulse in the BKGD pin by driving it low for at least 128 serial clock cycles and then driving it high for one serial clock cycle, providing a speedup pulse. By detecting this long low pulse in the BKGD pin, the target executes the SYNC protocol, see Section 7.4.9, "SYNC — Request Timed Reference Pulse", and assumes that the pending command and therefore the related ACK pulse, are being aborted. Therefore, after the SYNC protocol has been completed the host is free to issue new BDM commands. For BDM firmware READ or WRITE commands it can not be guaranteed that the pending command is aborted when issuing a SYNC before the corresponding ACK pulse. There is a short latency time from the time the READ or WRITE access begins until it is finished and the corresponding ACK pulse is issued. The latency time depends on the firmware READ or WRITE command that is issued and on the selected bus clock rate. When the SYNC command starts during this latency time the READ or WRITE command will not be aborted, but the corresponding ACK pulse will be aborted. A pending GO, TRACE1 or GO_UNTIL command can not be aborted. Only the corresponding ACK pulse can be aborted by the SYNC command.

Although it is not recommended, the host could abort a pending BDM command by issuing a low pulse in the BKGD pin shorter than 128 serial clock cycles, which will not be interpreted as the SYNC command. The ACK is actually aborted when a negative edge is perceived by the target in the BKGD pin. The short abort pulse should have at least 4 clock cycles keeping the BKGD pin low, in order to allow the negative edge to be detected by the target. In this case, the target will not execute the SYNC protocol but the pending command will be aborted along with the ACK pulse. The potential problem with this abort procedure is when there is a conflict between the ACK pulse and the short abort pulse. In this case, the target may not perceive the abort pulse. The worst case is when the pending command is a read command (i.e., READ_BYTE). If the abort pulse is not perceived by the target the host will attempt to send a new command after the abort pulse was issued, while the target expects the host to retrieve the accessed memory byte. In this case, host and target will run out of synchronism. However, if the command to be aborted is not a read command the short abort pulse could be used. After a command is aborted the target assumes the next negative edge, after the abort pulse, is the first bit of a new BDM command.

NOTE

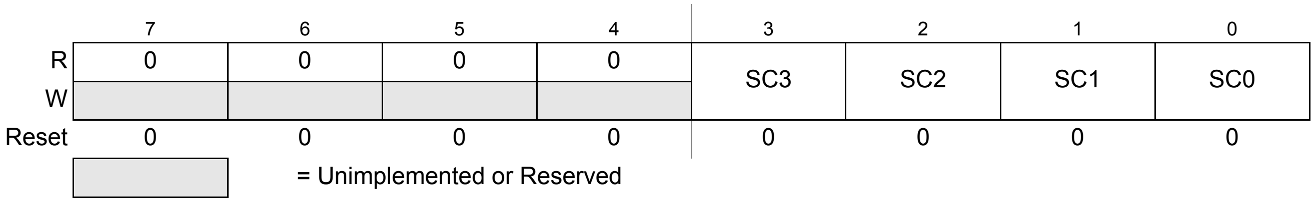
> The details about the short abort pulse are being provided only as a reference for the reader to better understand the BDM internal behavior. It is not recommended that this procedure be used in a real application.

Table 8-14. State Control Register Access Encoding

COMRV	Visible State Control Register
01	DBGSCR2
10	DBGSCR3
11	DBGMFR

8.3.2.7.1 Debug State Control Register 1 (DBGSCR1)

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 8-9. Debug State Control Register 1 (DBGSCR1)

Read: If COMRV[1:0] = 00

Write: If COMRV[1:0] = 00 and DBG is not armed.

This register is visible at 0x0027 only with COMRV[1:0] = 00. The state control register 1 selects the targeted next state whilst in State1. 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 8-15. DBGSCR1 Field Descriptions

Field	Description
3–0 SC[3:0]	These bits select the targeted next state whilst in State1, based upon the match event.

Table 8-16. State1 Sequencer Next State Selection

SC[3:0]	Description (Unspecified matches have no effect)
0000	Any match to Final State
0001	Match1 to State3
0010	Match2 to State2
0011	Match1 to State2
0100	Match0 to State2....... Match1 to State3
0101	Match1 to State3.........Match0 to Final State
0110	Match0 to State2....... Match2 to State3
0111	Either Match0 or Match1 to State2
1000	Reserved
1001	Match0 to State3

Table 10-20. Selectable Autonomous Periodical Interrupt Periods (continued)

APICLK	APIR[15:0]	Selected Period
0	0004	1.0 ms[1]
0	0005	1.2 ms[1]
0	.....	.....
0	FFFD	13106.8 ms[1]
0	FFFE	13107.0 ms[1]
0	FFFF	13107.2 ms[1]
1	0000	2 * Bus Clock period
1	0001	4 * Bus Clock period
1	0002	6 * Bus Clock period
1	0003	8 * Bus Clock period
1	0004	10 * Bus Clock period
1	0005	12 * Bus Clock period
1	.....	.....
1	FFFD	131068 * Bus Clock period
1	FFFE	131070 * Bus Clock period
1	FFFF	131072 * Bus Clock period

[1] When f_{ACLK} is trimmed to 10KHz.

10.3.2.17 Reserved Register CPMUTEST3

NOTE

This reserved register is designed for factory test purposes only, and is not intended for general user access. Writing to this register when in Special Mode can alter the S12CPMU's functionality.

0x02F6

	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 10-22. Reserved Register (CPMUTEST3)

Read: Anytime

Write: Only in Special Mode

12.3.2.9 ATD Status Register 2 (ATDSTAT2)

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

Module Base + 0x000A

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	CCF[7:0]							
W	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

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*)

14.3.2.9 ATD Status Register 2 (ATDSTAT2)

This read-only register contains the Conversion Complete Flags CCF[11:0].

Module Base + 0x000A

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	CCF[11:0]											
W	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 14-11. ATD Status Register 2 (ATDSTAT2)

Read: Anytime

Write: Anytime (for details see Table 14-18 below)

Table 14-18. ATDSTAT2 Field Descriptions

Field	Description
11–0 CCF[11:0]	**Conversion Complete Flag *n* (*n*= 11, 10, 9, 8, 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 15-9. Examples of ideal decimal ATD Results

Input Signal VRL = 0 Volts VRH = 5.12 Volts	8-Bit Codes (resolution=20mV)	10-Bit Codes (resolution=5mV)	Reserved
5.120 Volts	255	1023	Reserved
...	...	...	
0.022	1	4	
0.020	1	4	
0.018	1	4	
0.016	1	3	
0.014	1	3	
0.012	1	2	
0.010	1	2	
0.008	0	2	
0.006	0	1	
0.004	0	1	
0.003	0	1	
0.002	0	0	
0.000	0	0	

Table 15-10. Conversion Sequence Length Coding

S8C	S4C	S2C	S1C	Number of Conversions per Sequence
0	0	0	0	16
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

Table 15-11. ATD Behavior in Freeze Mode (Breakpoint)

FRZ1	FRZ0	Behavior in Freeze Mode
0	0	Continue conversion

16.3.2.12.2 Right Justified Result Data (DJM=1)

Module Base +
0x0010 = ATDDR0, 0x0012 = ATDDR1, 0x0014 = ATDDR2, 0x0016 = ATDDR3
0x0018 = ATDDR4, 0x001A = ATDDR5, 0x001C = ATDDR6, 0x001E = ATDDR7
0x0020 = ATDDR8, 0x0022 = ATDDR9, 0x0024 = ATDDR10, 0x0026 = ATDDR11
0x0028 = ATDDR12, 0x002A = ATDDR13, 0x002C = ATDDR14, 0x002E = ATDDR15

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	Result-Bit[11:0]											
W	▨	▨	▨	▨												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 16-15. Right justified ATD conversion result register (ATDDR*n*)

Table 16-22 shows how depending on the A/D resolution the conversion result is transferred to the ATD result registers for right justified data. Compare is always done using all 12 bits of both the conversion result and the compare value in ATDDRn.

Table 16-22. Conversion result mapping to ATDDRn

A/D resolution	DJM	conversion result mapping to ATDDR*n*
8-bit data	1	Result-Bit[7:0] = result, Result-Bit[11:8]=0000
10-bit data	1	Result-Bit[9:0] = result, Result-Bit[11:10]=00
12-bit data	1	Result-Bit[11:0] = result

Chapter 21
Serial Peripheral Interface (S12SPIV5)

Revision History

Revision Number	Date	Author	Summary of Changes
05.00	24 MAR 2005		Added 16-bit transfer width feature.

21.1 Introduction

The SPI module allows a duplex, synchronous, serial communication between the MCU and peripheral devices. Software can poll the SPI status flags or the SPI operation can be interrupt driven.

21.1.1 Glossary of Terms

SPI	Serial Peripheral Interface
$\overline{\text{SS}}$	Slave Select
SCK	Serial Clock
MOSI	Master Output, Slave Input
MISO	Master Input, Slave Output
MOMI	Master Output, Master Input
SISO	Slave Input, Slave Output

21.1.2 Features

The SPI includes these distinctive features:

- Master mode and slave mode
- Selectable 8 or 16-bit transfer width
- Bidirectional mode
- Slave select output
- Mode fault error flag with CPU interrupt capability

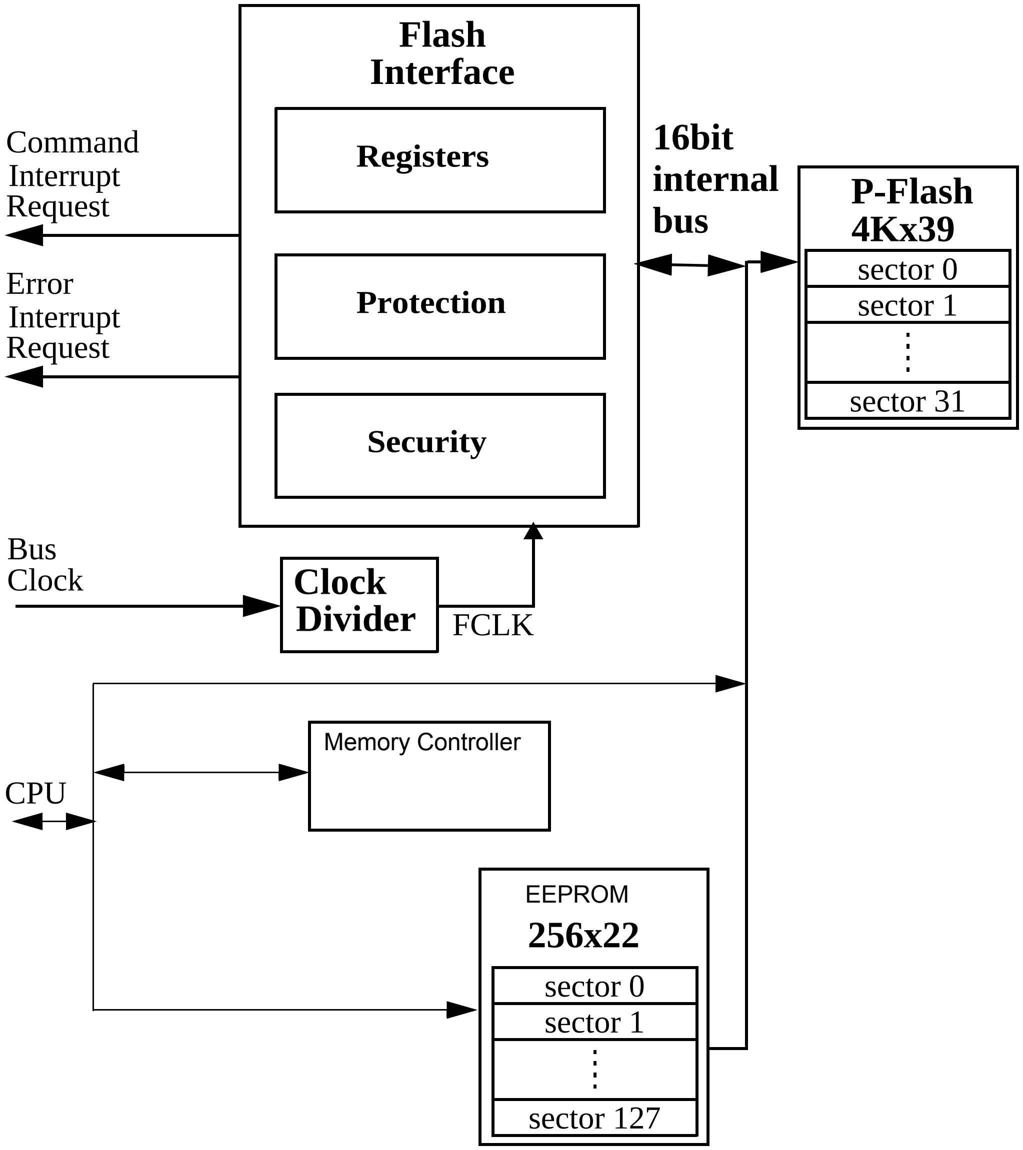


Figure 24-1. FTMRG16K1 Block Diagram

24.2 External Signal Description

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

CAUTION

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

Table 28-7. 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 28-8 shows recommended values for FDIV[5:0] based on the BUSCLK frequency. Please refer to Section 28.4.4, "Flash Command Operations," for more information.

Table 28-8. FDIV values for various BUSCLK Frequencies

BUSCLK Frequency (MHz)		FDIV[5:0]	BUSCLK Frequency (MHz)		FDIV[5:0]
MIN[1]	MAX[2]		MIN[1]	MAX[2]	
1.0	1.6	0x00	16.6	17.6	0x10
1.6	2.6	0x01	17.6	18.6	0x11
2.6	3.6	0x02	18.6	19.6	0x12
3.6	4.6	0x03	19.6	20.6	0x13
4.6	5.6	0x04	20.6	21.6	0x14
5.6	6.6	0x05	21.6	22.6	0x15
6.6	7.6	0x06	22.6	23.6	0x16
7.6	8.6	0x07	23.6	24.6	0x17
8.6	9.6	0x08	24.6	25.6	0x18
9.6	10.6	0x09			
10.6	11.6	0x0A			
11.6	12.6	0x0B			
12.6	13.6	0x0C			
13.6	14.6	0x0D			
14.6	15.6	0x0E			
15.6	16.6	0x0F			

[1] BUSCLK is Greater Than this value.

[2] BUSCLK is Less Than or Equal to this value.

The reserved nonvolatile information register accessed by the Program Once command cannot be erased and any attempt to program one of these phrases a second time will not be allowed. Valid phrase index values for the Program Once command range from 0x0000 to 0x0007. During execution of the Program Once command, any attempt to read addresses within P-Flash will return invalid data.

Table 28-43. 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 28-27)
		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.

28.4.6.7 Erase All Blocks Command

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

Table 28-44. 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 28-45. 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 28-27)
	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[1]
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation[1]

[1] As found in the memory map for FTMRG96K1.

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

Table 30-12. 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 30.3.2.11 Flash Common Command Object Register (FCCOB),” for more details.

30.3.2.4 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 30-8. Flash Reserved0 Register (FRSV0)

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

30.3.2.5 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 30-9. Flash Configuration Register (FCNFG)

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

Table 30-43. 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 30-27)
		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.

30.4.6.7 Erase All Blocks Command

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

Table 30-44. 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 30-45. 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 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

30.4.6.8 Erase Flash Block Command

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

NOTE

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

30.4.7.1 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 30.3.2.5, “Flash Configuration Register (FCNFG)”, Section 30.3.2.6, “Flash Error Configuration Register (FERCNFG)”, Section 30.3.2.7, “Flash Status Register (FSTAT)”, and Section 30.3.2.8, “Flash Error Status Register (FERSTAT)”.

The logic used for generating the Flash module interrupts is shown in Figure 30-27.

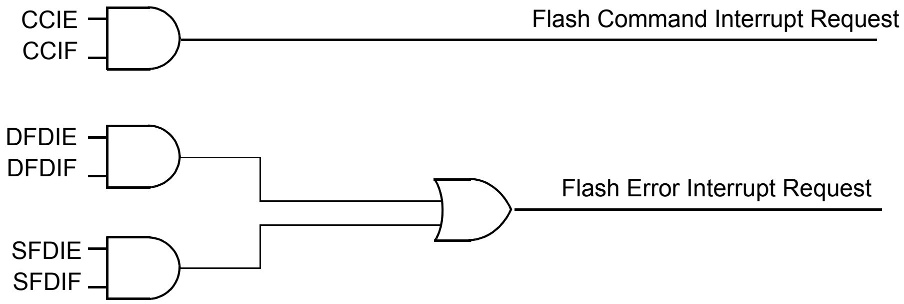


Figure 30-27. Flash Module Interrupts Implementation

30.4.8 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 30.4.7, “Interrupts”).

30.4.9 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.

31.5 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 31-11). 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

31.5.1 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 31.3.2.2), the Verify Backdoor Access Key command (see Section 31.4.6.11) 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 31-11) 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.

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 31.3.2.2), 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 31.4.6.11
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

All VDDX pins are internally connected by metal.

All VSSX pins are internally connected by metal.

VDDA, VDDX and VSSA, VSSX are connected by diodes for ESD protection.

NOTE

> In the following context V_{DD35} is used for either VDDA, VDDR, and VDDX; V_{SS35} is used for either VSSA and VSSX unless otherwise noted.
>
> I_{DD35} denotes the sum of the currents flowing into the VDDA, VDDX and VDDR pins.

A.1.3 Pins

There are four groups of functional pins.

A.1.3.1 I/O Pins

The I/O pins have a level in the range of 3.13V to 5.5V. This class of pins is comprised of all port I/O pins, the analog inputs, BKGD and the $\overline{\text{RESET}}$ pins. Some functionality may be disabled.

A.1.3.2 Analog Reference

This group consists of the VRH pin.

A.1.3.3 Oscillator

The pins EXTAL, XTAL dedicated to the oscillator have a nominal 1.8V level.

A.1.3.4 TEST

This pin is used for production testing only. The TEST pin must be tied to ground in all applications.

A.1.4 Current Injection

Power supply must maintain regulation within operating V_{DD35} or V_{DD} range during instantaneous and operating maximum current conditions. If positive injection current ($V_{in} > V_{DD35}$) is greater than I_{DD35}, the injection current may flow out of V_{DD35} and could result in external power supply going out of regulation. Ensure external V_{DD35} load will shunt current greater than maximum injection current. This will be the greatest risk when the MCU is not consuming power; e.g., if no system clock is present, or if clock rate is very low which would reduce overall power consumption.

A.1.5 Absolute Maximum Ratings

Absolute maximum ratings are stress ratings only. A functional operation under or outside those maxima is not guaranteed. Stress beyond those limits may affect the reliability or cause permanent damage of the device.

A.4.3.1 ADC Accuracy Definitions

For the following definitions see also Figure A-1.
Differential non-linearity (DNL) is defined as the difference between two adjacent switching steps.

$$\mathrm{DNL}(i) = \frac{V_i - V_{i-1}}{1\mathrm{LSB}} - 1$$

The integral non-linearity (INL) is defined as the sum of all DNLs:

$$\mathrm{INL}(n) = \sum_{i=1}^{n} \mathrm{DNL}(i) = \frac{V_n - V_0}{1\mathrm{LSB}} - n$$