**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/product-detail/nxp-semiconductors/s9s12g64f0clhr |

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify you have the latest information available, refer to:

nxp.com/

A full list of family members and options is included in the appendices.

## 1.8.4 S12G48 and S12G64

### 1.8.4.1 Pinout 32-Pin LQFP

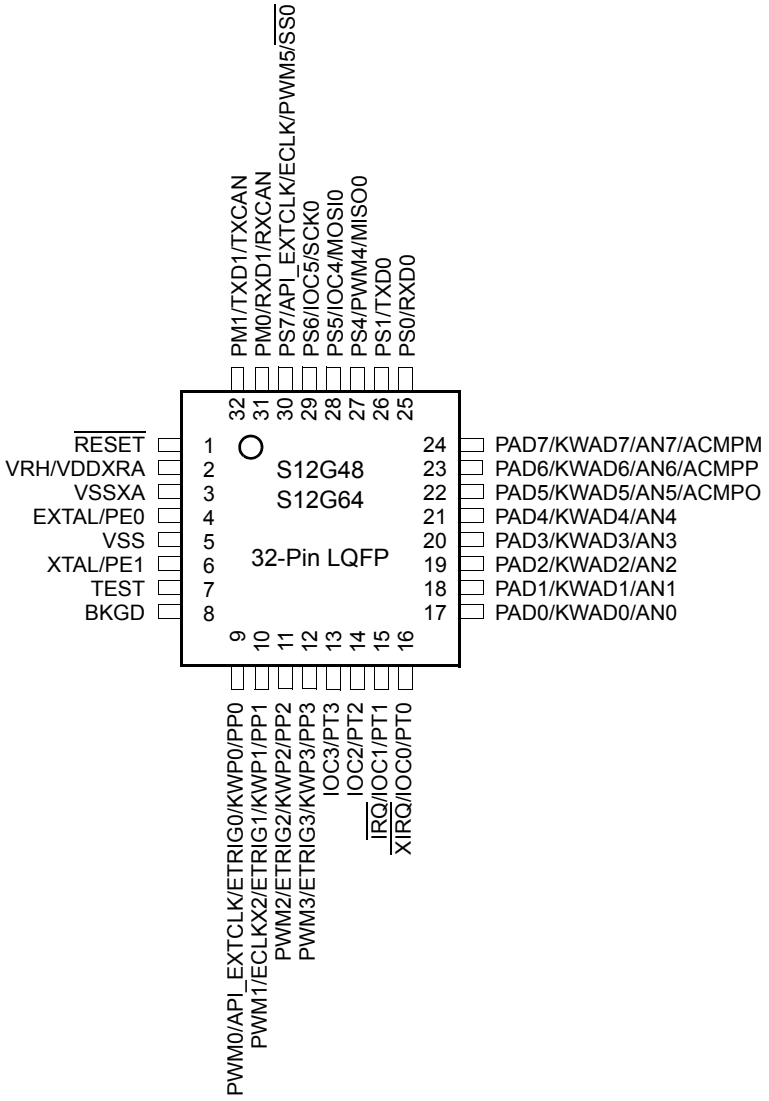



**Figure 1-10. 32-Pin LQFP Pinout for S12G48 and S12G64**

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

| | Function <----lowest-----PRIORITY-----highest----> | | | | | Power Supply | Internal Pull Resistor | |
|---|---|---|---|---|---|---|---|---|
| Package Pin | Pin | 2nd Func. | 3rd Func. | 4th Func | 5th Func | | CTRL | Reset State |
| 1 | RESET | — | — | — | — | $V_{DDX}$ | PULLUP | |

**Table 2-16. Port AD Pins AD15-8**

| | |
|---|---|
| PAD15 | • 64/100 LQFP: The unbuffered analog output signal DACU0 of the DAC0 module is mapped to this pin if the DAC is operating in “unbuffered DAC” mode. If this pin is used with the DAC then the digital I/O function and pull device are disabled.<br>• 64/100 LQFP: If routing is inactive (PRR1[PRR1AN]=0) the ADC analog input channel signal AN15 and the related digital trigger input are mapped to this pin. The ADC function has no effect on the output state. Refer to NOTE/2-172 for input buffer control.<br>• 64/100 LQFP: Pin interrupts can be generated if enabled in digital input or output mode.<br>• Signal priority:<br>64/100 LQFP: DACU0 > GPO |
| PAD14 | • 64/100 LQFP: The non-inverting analog input signal AMPP0 of the DAC0 module is mapped to this pin if the DAC is operating in “unbuffered DAC with operational amplifier” or “operational amplifier only” mode. If this pin is used with the DAC then the digital input buffer is disabled.<br>• 64/100 LQFP: If routing is inactive (PRR1[PRR1AN]=0) the ADC analog input channel signal AN14 and the related digital trigger input are mapped to this pin. The ADC function has no effect on the output state. Refer to NOTE/2-172 for input buffer control.<br>• 64/100 LQFP: Pin interrupts can be generated if enabled in digital input or output mode.<br>• Signal priority:<br>64/100 LQFP: GPO |
| PAD13 | • 64/100 LQFP: The inverting analog input signal AMPM0 of the DAC0 module is mapped to this pin if the DAC is operating in “unbuffered DAC with operational amplifier” or “operational amplifier only” mode. If this pin is used with the DAC then the digital input buffer is disabled.<br>• 64/100 LQFP: If routing is inactive (PRR1[PRR1AN]=0) the ADC analog input channel signal AN13 and the related digital trigger input are mapped to this pin. The ADC function has no effect on the output state. Refer to NOTE/2-172 for input buffer control.<br>• 64/100 LQFP: Pin interrupts can be generated if enabled in digital input or output mode.<br>• Signal priority:<br>64/100 LQFP: GPO |
| PAD12 | • 64/100 LQFP: The ADC analog input channel signal AN12 and the related digital trigger input are mapped to this pin. The ADC function has no effect on the output state. Refer to NOTE/2-172 for input buffer control.<br>• 64/100 LQFP: Pin interrupts can be generated if enabled in digital input or output mode.<br>• Signal priority:<br>64/100 LQFP: GPO |

earlier. Synchronization between the host and target is established in this manner at the start of every bit time.

Figure 7-7 shows an external host transmitting a logic 1 and transmitting a logic 0 to the BKGD pin of a target system. The host is asynchronous to the target, so there is up to a one clock-cycle delay from the host-generated falling edge to where the target recognizes this edge as the beginning of the bit time. Ten target clock cycles later, the target senses the bit level on the BKGD pin. Internal glitch detect logic requires the pin be driven high no later that eight target clock cycles after the falling edge for a logic 1 transmission.

Since the host drives the high speedup pulses in these two cases, the rising edges look like digitally driven signals.

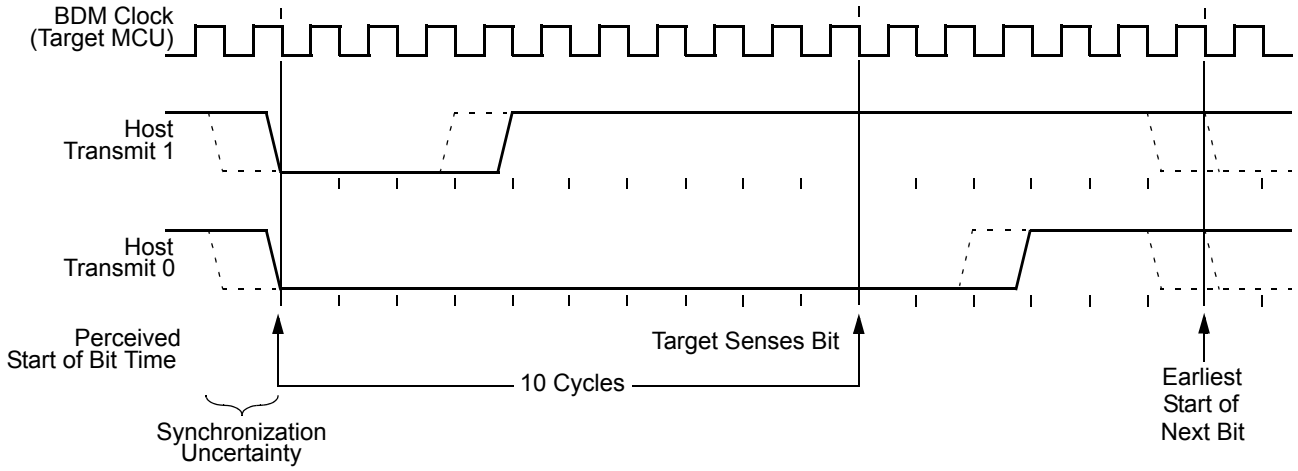



**Figure 7-7. BDM Host-to-Target Serial Bit Timing**

The receive cases are more complicated. Figure 7-8 shows the host receiving a logic 1 from the target system. Since the host is asynchronous to the target, there is up to one clock-cycle delay from the host-generated falling edge on BKGD to the perceived start of the bit time in the target. The host holds the BKGD pin low long enough for the target to recognize it (at least two target clock cycles). The host must release the low drive before the target drives a brief high speedup pulse seven target clock cycles after the perceived start of the bit time. The host should sample the bit level about 10 target clock cycles after it started the bit time.

## 11.1.2 Modes of Operation

### 11.1.2.1 Conversion Modes

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

### 11.1.2.2 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**
  ADC10B8C 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 ADC10B8C will either continue or finish or stop converting according to the FRZ1 and FRZ0 bits. This is useful for debugging and emulation.

### 17.5.4 Mode “Unbuffered DAC”

The “Unbuffered DAC” mode is selected by DACCNTL.DACM[2:0] = 0x4. During this mode the unbuffered analog voltage from the DAC resistor network output is available on the DACU output pin. The operational amplifier is disabled and the operational amplifier signals are disconnected from the AMP pins. For decoding of the control signals see Table 17-7.

### 17.5.5 Mode “Unbuffered DAC with Operational Amplifier”

The “Unbuffered DAC with Operational Amplifier” mode is selected by DACCTL.DACM[2:0] = 0x5. During this mode the DAC resistor network and the operational amplifier are enabled and usable independent from each other. The unbuffered analog voltage from the DAC resistor network output is available on the DACU output pin.

The operational amplifier is disconnected from the DAC resistor network. All required amplifier signals, AMP, AMPP and AMPM are available on the pins. The connection between the amplifier output and the negative amplifier input is open. For decoding of the control signals see Table 17-7.

### 17.5.6 Mode “Buffered DAC”

The “Buffered DAC” mode is selected by DACCTL.DACM[2:0] = 0x7. During this is mode the DAC resistor network and the operational amplifier are enabled. The analog output voltage from the DAC resistor network output is buffered by the operational amplifier and is available on the AMP output pin.

The DAC resistor network output is disconnected from the DACU pin. For the decoding of the control signals see Table 17-7.

### 17.5.7 Analog output voltage calculation

The DAC can provide an analog output voltage in two different voltage ranges:

- FVR = 0, reduced voltage range

  The DAC generates an analog output voltage inside the range from 0.1 x (VRH - VRL) + VRL to 0.9 x (VRH-VRL) + VRL with a resolution ((VRH-VRL) x 0.8) / 256, see equation below:

**analog output voltage = VOLATGE[7:0] x ((VRH-VRL) x 0.8) / 256) + 0.1 x (VRH-VRL) + VRL** ***Eqn. 17-1***

- FVR = 1, full voltage range

  The DAC generates an analog output voltage inside the range from VRL to VRH with a resolution (VRH-VRL) / 256, see equation below:

**analog output voltage = VOLTAGE[7:0] x (VRH-VRL) / 256 +VRL** ***Eqn. 17-2***

- The channel is disabled

In this way, the output of the PWM will always be either the old duty waveform or the new duty waveform, not some variation in between. If the channel is not enabled, then writes to the duty register will go directly to the latches as well as the buffer.

**NOTE**

Reads of this register return the most recent value written. Reads do not necessarily return the value of the currently active duty due to the double buffering scheme.

See Section 19.4.2.3, “PWM Period and Duty” for more information.

**NOTE**

Depending on the polarity bit, the duty registers will contain the count of either the high time or the low time. If the polarity bit is one, the output starts high and then goes low when the duty count is reached, so the duty registers contain a count of the high time. If the polarity bit is zero, the output starts low and then goes high when the duty count is reached, so the duty registers contain a count of the low time.

To calculate the output duty cycle (high time as a% of period) for a particular channel:

- Polarity = 0 (PPOL x =0)

  Duty Cycle = [(PWMPERx-PWMDTYx)/PWMPERx] * 100%
- Polarity = 1 (PPOLx = 1)

  Duty Cycle = [PWMDTYx / PWMPERx] * 100%

For boundary case programming values, please refer to Section 19.4.2.8, “PWM Boundary Cases”.

Module Base + 0x001C = PWMDTY0, 0x001D = PWMDTY1, 0x001E = PWMDTY2, 0x001F = PWMDTY3
Module Base + 0x0020 = PWMDTY4, 0x0021 = PWMDTY5, 0x0022 = PWMDTY6, 0x0023 = PWMDTY7

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Figure 19-14. PWM Channel Duty Registers (PWMDTYx)**

[1] This register is available only when the corresponding channel exists and is reserved if that channel does not exist. Writes to a reserved register have no functional effect. Reads from a reserved register return zeroes.

Read: Anytime

Write: Anytime

# Chapter 20
# Serial Communication Interface (S12SCIV5)

**Table 20-1. Revision History**

| Version Number | Revision Date | Effective Date | Author | Description of Changes |
|---|---|---|---|---|
| 05.03 | 12/25/2008 | | | remove redundancy comments in Figure1-2 |
| 05.04 | 08/05/2009 | | | fix typo, SCIBDL reset value be 0x04, not 0x00 |
| 05.05 | 06/03/2010 | | | fix typo, Table 20-4,SCICR1 Even parity should be PT=0<br>fix typo, on page 20-674,should be BKDIF,not BLDIF |

## 20.1 Introduction

This block guide provides an overview of the serial communication interface (SCI) module.

The SCI allows asynchronous serial communications with peripheral devices and other CPUs.

### 20.1.1 Glossary

IR: InfraRed

IrDA: Infrared Design Associate

IRQ: Interrupt Request

LIN: Local Interconnect Network

LSB: Least Significant Bit

MSB: Most Significant Bit

NRZ: Non-Return-to-Zero

RZI: Return-to-Zero-Inverted

RXD: Receive Pin

SCI : Serial Communication Interface

TXD: Transmit Pin

### 24.4.6.11 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 24-10). 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 Table 24-4). The Verify Backdoor Access Key command must not be executed from the Flash block containing the backdoor comparison key to avoid code runaway.

**Table 24-50. 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 24-51. 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 24.3.2.2) |
| | | Set if the backdoor key has mismatched since the last reset |
| | FPVIOL | None |
| | MGSTAT1 | None |
| | MGSTAT0 | None |

### 24.4.6.12 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.

#### 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.

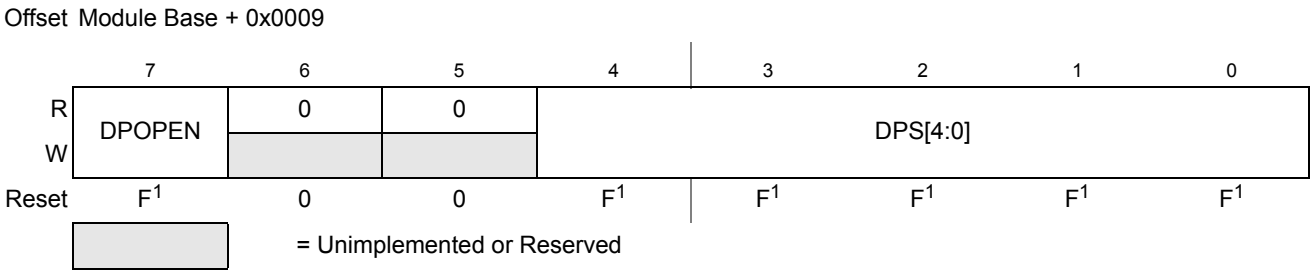



**Figure 25-15. EEPROM Protection Register (EEPROT)**

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.

**Table 25-61. Erase Verify EEPROM Section Command Error Handling**

| Register | Error Bit | Error Condition |
|---|---|---|
| FSTAT | ACCERR | Set if CCOBIX[2:0] != 010 at command launch |
| | | Set if command not available in current mode (see Table 25-27) |
| | | Set if an invalid global address [17:0] is supplied |
| | | Set if a misaligned word address is supplied (global address [0] != 0) |
| | | Set if the requested section breaches the end of the EEPROM block |
| | 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. |

### 25.4.6.15 Program EEPROM Command

The Program EEPROM operation programs one to four previously erased words in the EEPROM block. The Program EEPROM operation will confirm that the targeted location(s) were successfully programmed upon completion.

**CAUTION**

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

**Table 25-62. Program EEPROM Command FCCOB Requirements**

| CCOBIX[2:0] | FCCOB Parameters | |
|---|---|---|
| 000 | 0x11 | Global address [17:16] to identify the EEPROM block |
| 001 | Global address [15:0] of word to be programmed | |
| 010 | Word 0 program value | |
| 011 | Word 1 program value, if desired | |
| 100 | Word 2 program value, if desired | |
| 101 | Word 3 program value, if desired | |

Upon clearing CCIF to launch the Program EEPROM command, the user-supplied words will be transferred to the Memory Controller and be programmed if the area is unprotected. The CCOBIX index value at Program EEPROM command launch determines how many words will be programmed in the EEPROM block. The CCIF flag is set when the operation has completed.

A summary of the Flash module registers is given in Figure 26-4 with detailed descriptions in the following subsections.

| Address & Name | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 0x0000 FCLKDIV | R | FDIVLD | FDIVLCK | FDIV5 | FDIV4 | FDIV3 | FDIV2 | FDIV1 | FDIV0 |
| | W | | | | | | | | |
| 0x0001 FSEC | R | KEYEN1 | KEYEN0 | RNV5 | RNV4 | RNV3 | RNV2 | SEC1 | SEC0 |
| | W | | | | | | | | |
| 0x0002 FCCOBIX | R | 0 | 0 | 0 | 0 | 0 | CCOBIX2 | CCOBIX1 | CCOBIX0 |
| | W | | | | | | | | |
| 0x0003 FRSV0 | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | W | | | | | | | | |
| 0x0004 FCNFG | R | CCIE | 0 | 0 | IGNSF | 0 | 0 | FDFD | FSFD |
| | W | | | | | | | | |
| 0x0005 FERCNFG | R | 0 | 0 | 0 | 0 | 0 | 0 | DFDIE | SFDIE |
| | W | | | | | | | | |
| 0x0006 FSTAT | R | CCIF | 0 | ACCERR | FPVIOL | MGBUSY | RSVD | MGSTAT1 | MGSTAT0 |
| | W | | | | | | | | |
| 0x0007 FERSTAT | R | 0 | 0 | 0 | 0 | 0 | 0 | DFDIF | SFDIF |
| | W | | | | | | | | |
| 0x0008 FPROT | R | FPOPEN | RNV6 | FPHDIS | FPHS1 | FPHS0 | FPLDIS | FPLS1 | FPLS0 |
| | W | | | | | | | | |
| 0x0009 EEPROT | R | DPOPEN | 0 | DPS5 | DPS4 | DPS3 | DPS2 | DPS1 | DPS0 |
| | W | | | | | | | | |
| 0x000A FCCOBHI | R | CCOB15 | CCOB14 | CCOB13 | CCOB12 | CCOB11 | CCOB10 | CCOB9 | CCOB8 |
| | W | | | | | | | | |
| 0x000B FCCOBLO | R | CCOB7 | CCOB6 | CCOB5 | CCOB4 | CCOB3 | CCOB2 | CCOB1 | CCOB0 |
| | W | | | | | | | | |
| 0x000C FRSV1 | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | W | | | | | | | | |

**Figure 26-4. FTMRG48K1 Register Summary**

**Table 26-15. FSTAT Field Descriptions**

| Field | Description |
|---|---|
| 7<br>CCIF | **Command Complete Interrupt Flag** — The CCIF flag indicates that a Flash command has completed. The CCIF flag is cleared by writing a 1 to CCIF to launch a command and CCIF will stay low until command completion or command violation.<br>0 Flash command in progress<br>1 Flash command has completed |
| 5<br>ACCERR | **Flash Access Error Flag** — The ACCERR bit indicates an illegal access has occurred to the Flash memory caused by either a violation of the command write sequence (see Section 26.4.4.2) or issuing an illegal Flash command. While ACCERR is set, the CCIF flag cannot be cleared to launch a command. The ACCERR bit is cleared by writing a 1 to ACCERR. Writing a 0 to the ACCERR bit has no effect on ACCERR.<br>0 No access error detected<br>1 Access error detected |
| 4<br>FPVIOL | **Flash Protection Violation Flag** —The FPVIOL bit indicates an attempt was made to program or erase an address in a protected area of P-Flash or EEPROM memory during a command write sequence. The FPVIOL bit is cleared by writing a 1 to FPVIOL. Writing a 0 to the FPVIOL bit has no effect on FPVIOL. While FPVIOL is set, it is not possible to launch a command or start a command write sequence.<br>0 No protection violation detected<br>1 Protection violation detected |
| 3<br>MGBUSY | **Memory Controller Busy Flag** — The MGBUSY flag reflects the active state of the Memory Controller.<br>0 Memory Controller is idle<br>1 Memory Controller is busy executing a Flash command (CCIF = 0) |
| 2<br>RSVD | **Reserved Bit** — This bit is reserved and always reads 0. |
| 1–0<br>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 26.4.6, “Flash Command Description,” and Section 26.6, “Initialization” for details. |

### 26.3.2.8 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 26-12. Flash Error Status Register (FERSTAT)**

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

**Table 26-25. FOPT Field Descriptions**

| Field | Description |
|---|---|
| 7–0<br>NV[7:0] | **Nonvolatile Bits** — The NV[7:0] bits are available as nonvolatile bits. Refer to the device user guide for proper use of the NV bits. |

## 26.3.2.17 Flash Reserved5 Register (FRSV5)

This Flash register is reserved for factory testing.

Offset Module Base + 0x0011

| | 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 26-23. Flash Reserved5 Register (FRSV5)**

All bits in the FRSV5 register read 0 and are not writable.

## 26.3.2.18 Flash Reserved6 Register (FRSV6)

This Flash register is reserved for factory testing.

Offset Module Base + 0x0012

| | 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 26-24. Flash Reserved6 Register (FRSV6)**

All bits in the FRSV6 register read 0 and are not writable.

## 26.3.2.19 Flash Reserved7 Register (FRSV7)

This Flash register is reserved for factory testing.

**Table 28-28. P-Flash Commands**

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

### 28.4.4.5 EEPROM Commands

Table 28-29 summarizes the valid EEPROM commands along with the effects of the commands on the EEPROM block.

**Table 28-29. EEPROM Commands**

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

### 29.4.4.3 Valid Flash Module Commands

Table 29-27 present the valid Flash commands, as enabled by the combination of the functional MCU mode (Normal SingleChip NS, Special Singlechip SS) with the MCU security state (Unsecured, Secured).

Special Singlechip mode is selected by input mmc_ss_mode_ts2 asserted. MCU Secured state is selected by input mmc_secure input asserted.

**Table 29-27. Flash Commands by Mode and Security State**

| FCMD | Command | Unsecured | | Secured | |
|---|---|---|---|---|---|
| | | NS[1] | SS[2] | NS[3] | SS[4] |
| 0x01 | Erase Verify All Blocks | * | * | * | * |
| 0x02 | Erase Verify Block | * | * | * | * |
| 0x03 | Erase Verify P-Flash Section | * | * | * | |
| 0x04 | Read Once | * | * | * | |
| 0x06 | Program P-Flash | * | * | * | |
| 0x07 | Program Once | * | * | * | |
| 0x08 | Erase All Blocks | | * | | * |
| 0x09 | Erase Flash Block | * | * | * | |
| 0x0A | Erase P-Flash Sector | * | * | * | |
| 0x0B | Unsecure Flash | | * | | * |
| 0x0C | Verify Backdoor Access Key | * | | * | |
| 0x0D | Set User Margin Level | * | * | * | |
| 0x0E | Set Field Margin Level | | * | | |
| 0x10 | Erase Verify EEPROM Section | * | * | * | |
| 0x11 | Program EEPROM | * | * | * | |
| 0x12 | Erase EEPROM Sector | * | * | * | |

1 Unsecured Normal Single Chip mode
2 Unsecured Special Single Chip mode.
3 Secured Normal Single Chip mode.
4 Secured Special Single Chip mode.

### 29.4.4.4 P-Flash Commands

Table 29-28 summarizes the valid P-Flash commands along with the effects of the commands on the P-Flash block and other resources within the Flash module.

**Table 29-28. P-Flash Commands**

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

### 29.4.8 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 29.4.7, “Interrupts”).

### 29.4.9 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.

## 29.5 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 29-11). 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

### 29.5.1 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 29.3.2.2), the Verify Backdoor Access Key command (see Section 29.4.6.11) 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 29-11) 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.

**Table 31-57. Set Field Margin Level Command FCCOB Requirements**

| CCOBIX[2:0] | FCCOB Parameters | |
|---|---|---|
| 000 | 0x0E | Flash block selection code [1:0]. See Table 31-34 |
| 001 | Margin level setting. | |

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

**NOTE**

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

Valid margin level settings for the Set Field Margin Level command are defined in Table 31-58.

**Table 31-58. Valid Set Field Margin Level Settings**

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

1 Read margin to the erased state

2 Read margin to the programmed state

**Table 31-59. Set Field Margin Level Command Error Handling**

| Register | Error Bit | Error Condition |
|---|---|---|
| FSTAT | ACCERR | Set if CCOBIX[2:0] != 001 at command launch. |
| | | Set if command not available in current mode (see Table 31-27). |
| | | Set if an invalid margin level setting is supplied. |
| | FPVIOL | None |
| | MGSTAT1 | None |
| | MGSTAT0 | None |

**Table A-18. Pseudo Stop Current Characteristics**

| Conditions are: $V_{DDX}$=5V, $V_{DDR}$=5V, $V_{DDA}$=5V, RTI and COP and API enabled, see Table A-12. | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Num** | **C** | **Rating** | **Symbol** | **Min** | **Typ** | **Max** | **Unit** |
| S12GN16, S12GN32 | | | | | | | |
| 1 | C | -40°C | $I_{DDPS}$ | | 155 | | µA |
| 2 | C | 25°C | $I_{DDPS}$ | | 165 | | µA |
| 3 | C | 150°C | $I_{DDPS}$ | | 265 | | µA |
| 4 | C | 160°C | $I_{DDPS}$ | | 295 | | µA |
| S12GN48, S12G48, S12G64 | | | | | | | |
| 5 | C | -40°C | $I_{DDPS}$ | | 160 | | µA |
| 6 | C | 25°C | $I_{DDPS}$ | | 170 | | µA |
| 7 | C | 150°C | $I_{DDPS}$ | | 285 | | µA |
| S12G96, S12G128 | | | | | | | |
| 8 | C | -40°C | $I_{DDPS}$ | | 165 | | µA |
| 9 | C | 25°C | $I_{DDPS}$ | | 175 | | µA |
| 10 | C | 150°C | $I_{DDPS}$ | | 320 | | µA |
| S12G192, S12GA192, S12G240, S12GA240 | | | | | | | |
| 11 | C | -40°C | $I_{DDPS}$ | | 175 | | µA |
| 12 | C | 25°C | $I_{DDPS}$ | | 185 | | µA |
| 13 | C | 150°C | $I_{DDPS}$ | | 430 | | µA |

## A.4 ADC Characteristics

This section describes the characteristics of the analog-to-digital converter.

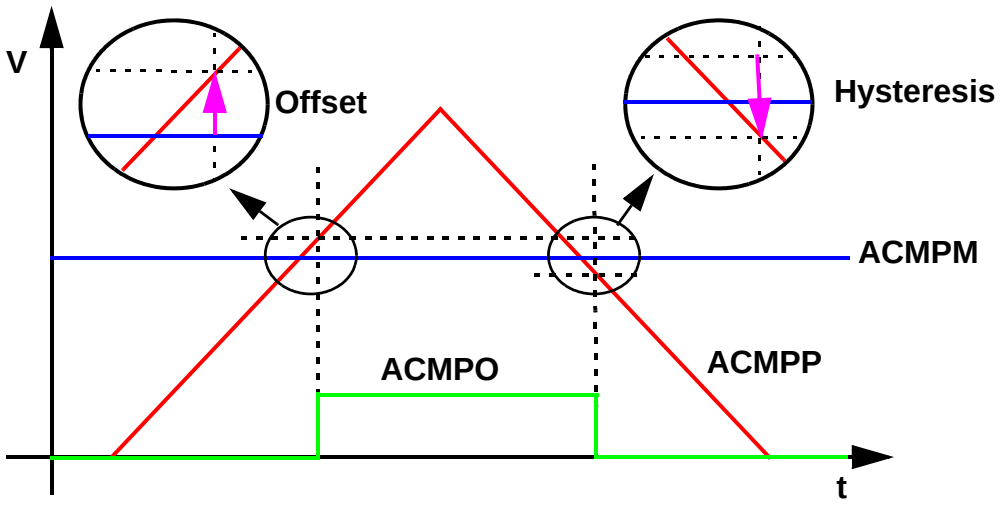
### A.4.1 ADC Operating Characteristics

The Table A-19 and Table A-20 show conditions under which the ADC operates.

The following constraints exist to obtain full-scale, full range results:

$$V_{SSA} \le V_{RL} \le V_{IN} \le V_{RH} \le V_{DDA}.$$

Figure A-3. Input Offset and Hysteresis




## A.6 DAC Characteristics

This section describes the electrical characteristics of the digital to analog converter.

Table A-33. Static Electrical Characteristics

| Characteristics noted under conditions 3.13V <= VDDA <= 5.5V>, -40°C < Tj < 150°C >, VRH=VDDA, VRL=VSSA unless otherwise noted. Typical values noted reflect the approximate parameter mean at $T_A$ = 25°C under nominal conditions unless otherwise noted. | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Num** | **C** | **Ratings** | **Symbol** | **Min** | **Typ** | **Max** | **Unit** |
| 1 | <br>D<br>P<br>P | Supply Current<br>buffer disabled<br>buffer enabled FVR=0 DRIVE=1<br>buffer enabled FVR=1 DRIVE=0 | $I_{buf}$ | <br>-<br>-<br>- | <br>-<br>365<br>215 | <br>5<br>800<br>800 | µA |
| 2 | <br>D<br>P | Reference current<br>reference disabled<br>reference enabled | $I_{ref}$ | - | <br>-<br>50 | <br>1<br>150 | µA |
| 3 | D | Resolution | | 8 | | | bit |
| 4 | C | Relative Accuracy @ amplifier output | INL | -0.5 | | +0.5 | LSB |
| 5 | P | Differential Nonlinearity @ amplifier output | DNL | -0.5 | | +0.5 | LSB |
| 6 | D | DAC Range A (FVR bit = 1) | $V_{out}$ | 0...255/256(VRH-VRL)+VRL | | | V |
| 7 | D | DAC Range B (FVR bit = 0 | $V_{out}$ | 32...287/320(VRH-VRL)+VRL | | | V |