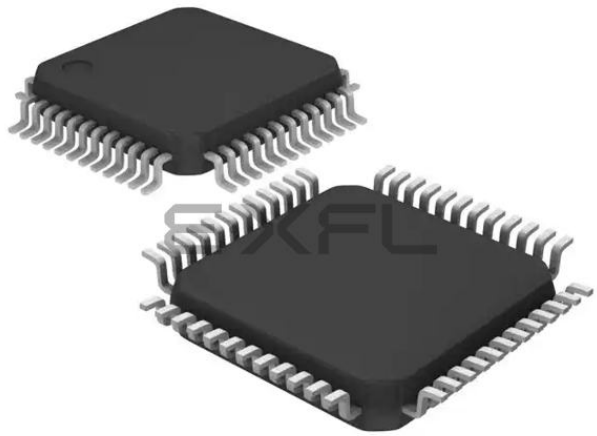


Welcome to E-XFL.COM

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 ~ 150°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/s9s12g48f0wlfr

Table 2-21. Block Register Map (G3) (continued)

Global Address Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x026D PPSJ	R	0	0	0	0	PPSJ3	PPSJ2	PPSJ1	PPSJ0
	W								
0x026E PIEJ	R	0	0	0	0	PIEJ3	PIEJ2	PIEJ1	PIEJ0
	W								
0x026F PIFJ	R	0	0	0	0	PIFJ3	PIFJ2	PIFJ1	PIFJ0
	W								
0x0270 PT0AD	R	0	0	0	0	PT0AD3	PT0AD2	PT0AD1	PT0AD0
	W								
0x0271 PT1AD	R	PT1AD7	PT1AD6	PT1AD5	PT1AD4	PT1AD3	PT1AD2	PT1AD1	PT1AD0
	W								
0x0272 PTI0AD	R	0	0	0	0	PTI0AD3	PTI0AD2	PTI0AD1	PTI0AD0
	W								
0x0273 PTI1AD	R	PTI1AD7	PTI1AD6	PTI1AD5	PTI1AD4	PTI1AD3	PTI1AD2	PTI1AD1	PTI1AD0
	W								
0x0274 DDR0AD	R	0	0	0	0	DDR0AD3	DDR0AD2	DDR0AD1	DDR0AD0
	W								
0x0275 DDR1AD	R	DDR1AD7	DDR1AD6	DDR1AD5	DDR1AD4	DDR1AD3	DDR1AD2	DDR1AD1	DDR1AD0
	W								
0x0276 Reserved	R	0	0	0	0	0	0	0	0
	W								
0x0277 Reserved	R	0	0	0	0	0	0	0	0
	W								
0x0278 PER0AD	R	0	0	0	0	PER0AD3	PER0AD2	PER0AD1	PER0AD0
	W								
0x0279 PER1AD	R	PER1AD7	PER1AD6	PER1AD5	PER1AD4	PER1AD3	PER1AD2	PER1AD1	PER1AD0
	W								
0x027A PPS0AD	R	0	0	0	0	PPS0AD3	PPS0AD2	PPS0AD1	PPS0AD0
	W								
0x027B PPS1AD	R	PPS1AD7	PPS1AD6	PPS1AD5	PPS1AD4	PPS1AD3	PPS1AD2	PPS1AD1	PPS1AD0
	W								

= Unimplemented or Reserved

2.5.2.3 Data Direction Register (DDRx)

This register defines whether the pin is used as an general-purpose input or an output.

If a peripheral module controls the pin the contents of the data direction register is ignored (Figure 2-64).

Independent of the pin usage with a peripheral module this register determines the source of data when reading the associated data register address (2.5.2.1/2-241).

NOTE

Due to internal synchronization circuits, it can take up to 2 bus clock cycles until the correct value is read on port data or port input registers, when changing the data direction register.

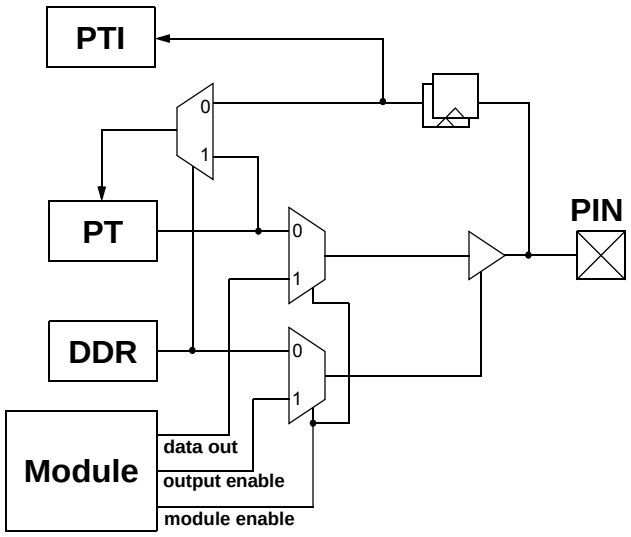


Figure 2-64. Illustration of I/O pin functionality

2.5.2.4 Pull Device Enable Register (PERx)

This register turns on a pullup or pulldown device on the related pins determined by the associated polarity select register (2.5.2.5/2-242).

The pull device becomes active only if the pin is used as an input or as a wired-or output. Some peripheral module only allow certain configurations of pull devices to become active. Refer to Section 2.3, “PIM Routing - Functional description”.

2.5.2.5 Pin Polarity Select Register (PPSx)

This register selects either a pullup or pulldown device if enabled.

It becomes only active if the pin is used as an input. A pullup device can be activated if the pin is used as a wired-or output.

Table 3-3. ACMPS Register Field Descriptions

Field	Description
7 ACIF	ACMP Interrupt Flag— ACIF is set when a compare event occurs. Compare events are defined by ACMOD[1:0]. Writing a logic “1” to the bit field clears the flag. 0 Compare event has not occurred 1 Compare event has occurred
6 ACO	ACMP Output— Reading ACO returns the current value of the synchronized ACMP output. Refer to ACE description to account for initialization delay on this path.

3.7 Functional Description

The ACMP compares two analog input voltages applied to ACMPM and ACMPP. The comparator output is high when the voltage at the non-inverting input is greater than the voltage at the inverting input, and is low when the non-inverting input voltage is lower than the inverting input voltage.

The ACMP is enabled with register bit ACMPC[ACE]. When ACMPC[ACE] is set, the input pins are connected to low-pass filters. The comparator output is disconnected from the subsequent logic, which is held at its state for 63 bus clock cycles after setting ACMPC[ACE] to “1” to mask potential glitches. This initialization delay must be accounted for before the first comparison result can be expected.

The initial hold state after reset is zero, thus if input voltages are set to result in “true” result ($V_{ACMPP} > V_{ACMPM}$) before the initialization delay has passed, a flag will be set immediately after this.

Similarly the flag will also be set when disabling the ACMP, then re-enabling it with the inputs changing to produce an opposite result to the hold state before the end of the initialization delay.

By setting the ACMPC[ACICE] bit the gated comparator output can be connected to the synchronized timer input capture channel 5 (see Figure 3-1). This feature can be used to generate time stamps and timer interrupts on ACMP events.

The comparator output signal synchronized to the bus clock is used to read the comparator output status (ACMPS[ACO]) and to set the interrupt flag (ACMPS[ACIF]).

The condition causing the interrupt flag (ACMPS[ACIF]) to assert is selected with register bits ACMPC[ACMOD1:ACMOD0]. This includes any edge configuration, that is rising, or falling, or rising and falling (toggle) edges of the comparator output. Also flag setting can be disabled.

An interrupt will be generated if the interrupt enable bit (ACMPC[ACIE]) and the interrupt flag (ACMPS[ACIF]) are both set. ACMPS[ACIF] is cleared by writing a 1.

The raw comparator output signal ACMPO can be driven out on an external pin by setting the ACMPC[ACOPE] bit.

If the comparator register contents coincide with the SWI/BDM vector address then an SWI in user code could coincide with a DBG breakpoint. The CPU ensures that BDM requests have a higher priority than SWI requests. Returning from the BDM/SWI service routine care must be taken to avoid a repeated breakpoint at the same address.

Should a tagged or forced breakpoint coincide with a BGND in user code, then the instruction that follows the BGND instruction is the first instruction executed when normal program execution resumes.

NOTE

When program control returns from a tagged breakpoint using an RTI or BDM GO command without program counter modification it returns to the instruction whose tag generated the breakpoint. To avoid a repeated breakpoint at the same location reconfigure the DBG module in the SWI routine, if configured for an SWI breakpoint, or over the BDM interface by executing a TRACE command before the GO to increment the program flow past the tagged instruction.

8.5 Application Information

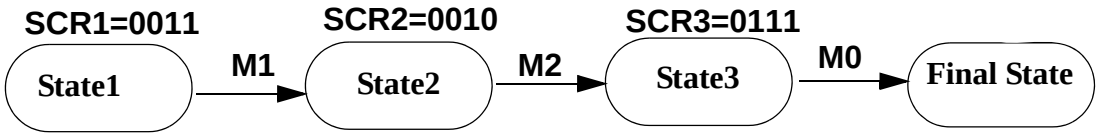
8.5.1 State Machine scenarios

Defining the state control registers as SCR1,SCR2, SCR3 and M0,M1,M2 as matches on channels 0,1,2 respectively. SCR encoding supported by S12SDBGV1 are shown in black. SCR encoding supported only in S12SDBGV2 are shown in red. For backwards compatibility the new scenarios use a 4th bit in each SCR register. Thus the existing encoding for SCRx[2:0] is not changed.

8.5.2 Scenario 1

A trigger is generated if a given sequence of 3 code events is executed.

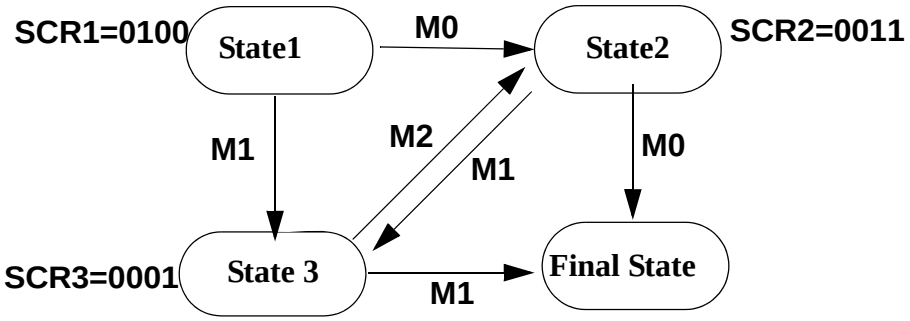
Figure 8-27. Scenario 1



Scenario 1 is possible with S12SDBGV1 SCR encoding

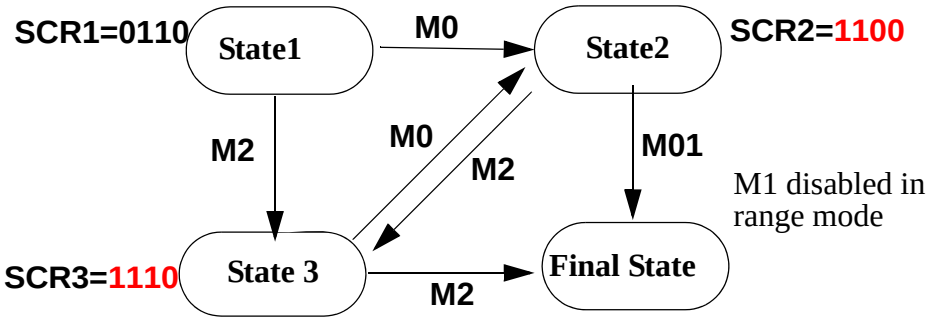
event B cause a trigger. Similarly 2 consecutive occurrences of event B without an intermediate event A cause a trigger. This is possible by using CompA and CompC to match on the same address as shown.

Figure 8-32. Scenario 4a



This scenario is currently not possible using 2 comparators only. S12SDBGV2 makes it possible with 2 comparators, State 3 allowing a M0 to return to state 2, whilst a M2 leads to final state as shown.

Figure 8-33. Scenario 4b (with 2 comparators)



The advantage of using only 2 channels is that now range comparisons can be included (channel0)

This however violates the S12SDBGV1 specification, which states that a match leading to final state always has priority in case of a simultaneous match, whilst priority is also given to the lowest channel number. For S12SDBG the corresponding CPU priority decoder is removed to support this, such that on simultaneous taghits, taghits pointing to final state have highest priority. If no taghit points to final state then the lowest channel number has priority. Thus with the above encoding from State3, the CPU and DBG would break on a simultaneous M0/M2.

Table 10-2. Reference Clock Frequency Selection if OSC_LCP is enabled

REFCLK Frequency Ranges (OSCE=1)	REFFRQ[1:0]
1MHz <= f_{REF} <= 2MHz	00
2MHz < f_{REF} <= 6MHz	01
6MHz < f_{REF} <= 12MHz	10
f_{REF} >12MHz	11

10.3.2.3 S12CPMU Post Divider Register (CPMUPOSTDIV)

The POSTDIV register controls the frequency ratio between the VCOCLK and the PLLCLK.

0x0036

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

= Unimplemented or Reserved

Figure 10-6. S12CPMU Post Divider Register (CPMUPOSTDIV)

Read: Anytime

Write: Anytime if PLLSEL=1. Else write has no effect.

If PLL is locked (LOCK=1) $$f_{PLL} = \frac{f_{VCO}}{(POSTDIV + 1)}$$

If PLL is not locked (LOCK=0) $$f_{PLL} = \frac{f_{VCO}}{4}$$

If PLL is selected (PLLSEL=1) $$f_{bus} = \frac{f_{PLL}}{2}$$

10.3.2.4 S12CPMU Flags Register (CPMUFLG)

This register provides S12CPMU status bits and flags.

Table 12-15. Analog Input Channel Select Coding

SC	CD	CC	CB	CA	Analog Input Channel
1	0	0	0	0	Internal_6,
	0	0	0	1	Internal_7
	0	0	1	0	Internal_0
	0	0	1	1	Internal_1
	0	1	0	0	VRH
	0	1	0	1	VRL
	0	1	1	0	(VRH+VRL) / 2
	0	1	1	1	Reserved
	1	0	0	0	Internal_2
	1	0	0	1	Internal_3
	1	0	1	0	Internal_4
	1	0	1	1	Internal_5
	1	1	X	X	Reserved

15.3.2.7 ATD Status Register 0 (ATDSTAT0)

This register contains the Sequence Complete Flag, overrun flags for external trigger and FIFO mode, and the conversion counter.

Module Base + 0x0006

	7	6	5	4	3	2	1	0
R	SCF	0	ETORF	FIFOR	CC3	CC2	CC1	CC0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 15-9. ATD Status Register 0 (ATDSTAT0)

Read: Anytime

Write: Anytime (No effect on (CC3, CC2, CC1, CC0))

Table 15-16. ATDSTAT0 Field Descriptions

Field	Description
7 SCF	**Sequence Complete Flag** — This flag is set upon completion of a conversion sequence. If conversion sequences are continuously performed (SCAN=1), the flag is set after each one is completed. This flag is cleared when one of the following occurs: A) Write “1” to SCF B) Write to ATDCTL5 (a new conversion sequence is started) C) If AFFC=1 and a result register is read 0 Conversion sequence not completed 1 Conversion sequence has completed
5 ETORF	**External Trigger Overrun Flag** — While in edge sensitive mode (ETRIGLE=0), if additional active edges are detected while a conversion sequence is in process the overrun flag is set. This flag is cleared when one of the following occurs: A) Write “1” to ETORF B) Write to ATDCTL0,1,2,3,4, ATDCMPE or ATDCMPHT (a conversion sequence is aborted) C) Write to ATDCTL5 (a new conversion sequence is started) 0 No External trigger overrun error has occurred 1 External trigger overrun error has occurred
4 FIFOR	**Result Register Overrun Flag** — This bit indicates that a result register has been written to before its associated conversion complete flag (CCF) has been cleared. This flag is most useful when using the FIFO mode because the flag potentially indicates that result registers are out of sync with the input channels. However, it is also practical for non-FIFO modes, and indicates that a result register has been overwritten before it has been read (i.e. the old data has been lost). This flag is cleared when one of the following occurs: A) Write “1” to FIFOR B) Write to ATDCTL0,1,2,3,4, ATDCMPE or ATDCMPHT (a conversion sequence is aborted) C) Write to ATDCTL5 (a new conversion sequence is started) 0 No overrun has occurred 1 Overrun condition exists (result register has been written while associated CCFx flag was still set)

Table 18-3. CANCTL0 Register Field Descriptions (continued)

Field	Description
2 WUPE[3]	**Wake-Up Enable** — This configuration bit allows the MSCAN to restart from sleep mode or from power down mode (entered from sleep) when traffic on CAN is detected (see Section 18.4.5.5, “MSCAN Sleep Mode”). This bit must be configured before sleep mode entry for the selected function to take effect. 0 Wake-up disabled — The MSCAN ignores traffic on CAN 1 Wake-up enabled — The MSCAN is able to restart
1 SLPRQ[4]	**Sleep Mode Request** — This bit requests the MSCAN to enter sleep mode, which is an internal power saving mode (see Section 18.4.5.5, “MSCAN Sleep Mode”). The sleep mode request is serviced when the CAN bus is idle, i.e., the module is not receiving a message and all transmit buffers are empty. The module indicates entry to sleep mode by setting SLPAK = 1 (see Section 18.3.2.2, “MSCAN Control Register 1 (CANCTL1)”). SLPRQ cannot be set while the WUPIF flag is set (see Section 18.3.2.5, “MSCAN Receiver Flag Register (CANRFLG)”). Sleep mode will be active until SLPRQ is cleared by the CPU or, depending on the setting of WUPE, the MSCAN detects activity on the CAN bus and clears SLPRQ itself. 0 Running — The MSCAN functions normally 1 Sleep mode request — The MSCAN enters sleep mode when CAN bus idle
0 INITRQ[5,6]	**Initialization Mode Request** — When this bit is set by the CPU, the MSCAN skips to initialization mode (see Section 18.4.4.5, “MSCAN Initialization Mode”). Any ongoing transmission or reception is aborted and synchronization to the CAN bus is lost. The module indicates entry to initialization mode by setting INITAK = 1 (Section 18.3.2.2, “MSCAN Control Register 1 (CANCTL1)”). The following registers enter their hard reset state and restore their default values: CANCTL0[7], CANRFLG[8], CANRIER[9], CANTFLG, CANTIER, CANTARQ, CANTAAK, and CANTBSEL. The registers CANCTL1, CANBTR0, CANBTR1, CANIDAC, CANIDAR0-7, and CANIDMR0-7 can only be written by the CPU when the MSCAN is in initialization mode (INITRQ = 1 and INITAK = 1). The values of the error counters are not affected by initialization mode. When this bit is cleared by the CPU, the MSCAN restarts and then tries to synchronize to the CAN bus. If the MSCAN is not in bus-off state, it synchronizes after 11 consecutive recessive bits on the CAN bus; if the MSCAN is in bus-off state, it continues to wait for 128 occurrences of 11 consecutive recessive bits. Writing to other bits in CANCTL0, CANRFLG, CANRIER, CANTFLG, or CANTIER must be done only after initialization mode is exited, which is INITRQ = 0 and INITAK = 0. 0 Normal operation 1 MSCAN in initialization mode

1 See the Bosch CAN 2.0A/B specification for a detailed definition of transmitter and receiver states.

2 In order to protect from accidentally violating the CAN protocol, TXCAN is immediately forced to a recessive state when the CPU enters wait (CSWAI = 1) or stop mode (see Section 18.4.5.2, “Operation in Wait Mode” and Section 18.4.5.3, “Operation in Stop Mode”).

3 The CPU has to make sure that the WUPE register and the WUPIE wake-up interrupt enable register (see Section 18.3.2.6, “MSCAN Receiver Interrupt Enable Register (CANRIER)) is enabled, if the recovery mechanism from stop or wait is required.

4 The CPU cannot clear SLPRQ before the MSCAN has entered sleep mode (SLPRQ = 1 and SLPAK = 1).

5 The CPU cannot clear INITRQ before the MSCAN has entered initialization mode (INITRQ = 1 and INITAK = 1).

6 In order to protect from accidentally violating the CAN protocol, TXCAN is immediately forced to a recessive state when the initialization mode is requested by the CPU. Thus, the recommended procedure is to bring the MSCAN into sleep mode (SLPRQ = 1 and SLPAK = 1) before requesting initialization mode.

7 Not including WUPE, INITRQ, and SLPRQ.

8 TSTAT1 and TSTAT0 are not affected by initialization mode.

9 RSTAT1 and RSTAT0 are not affected by initialization mode.

18.3.2.2 MSCAN Control Register 1 (CANCTL1)

The CANCTL1 register provides various control bits and handshake status information of the MSCAN module as described below.

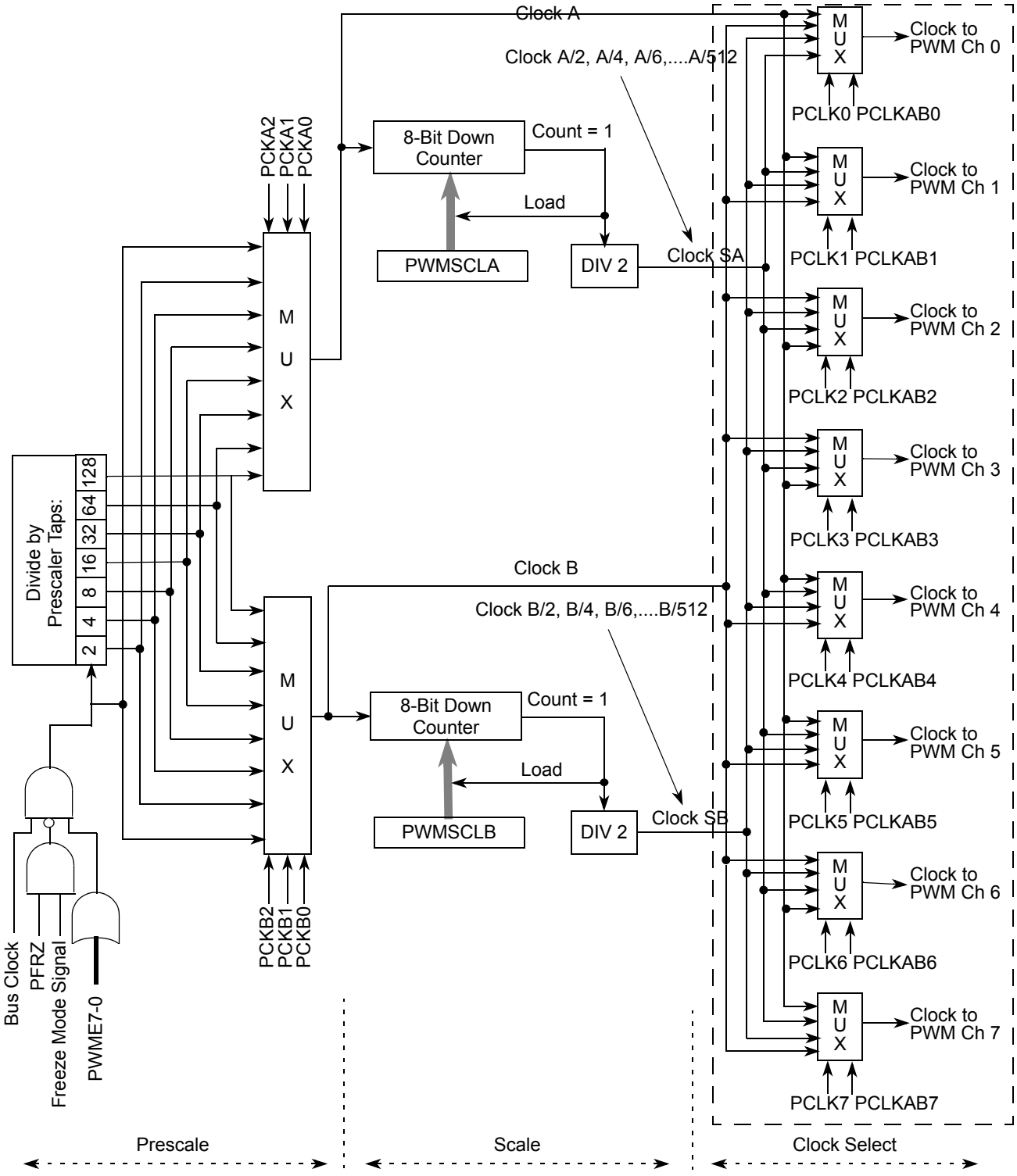


Figure 19-15. PWM Clock Select Block Diagram

20.1.4 Block Diagram

Figure 20-1 is a high level block diagram of the SCI module, showing the interaction of various function blocks.

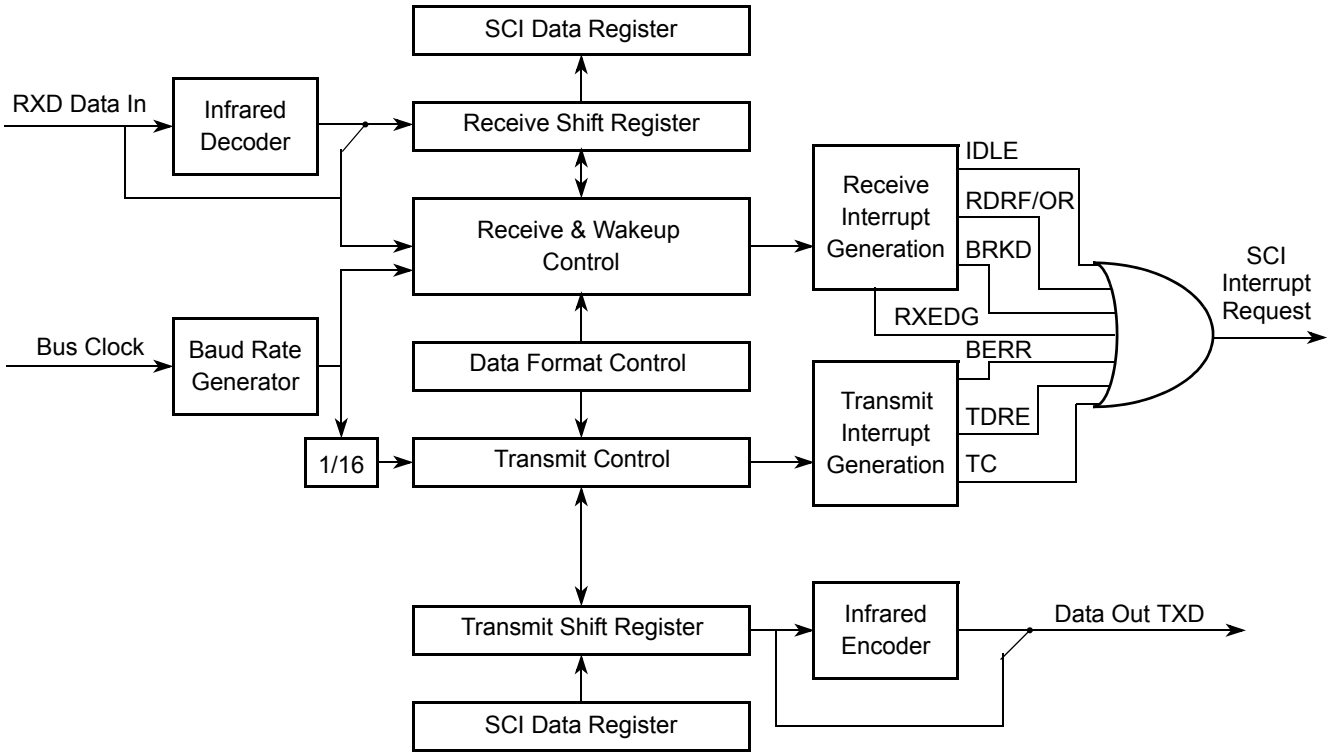


Figure 20-1. SCI Block Diagram

20.2 External Signal Description

The SCI module has a total of two external pins.

20.2.1 TXD — Transmit Pin

The TXD pin transmits SCI (standard or infrared) data. It will idle high in either mode and is high impedance anytime the transmitter is disabled.

20.2.2 RXD — Receive Pin

The RXD pin receives SCI (standard or infrared) data. An idle line is detected as a line high. This input is ignored when the receiver is disabled and should be terminated to a known voltage.

20.3 Memory Map and Register Definition

This section provides a detailed description of all the SCI registers.

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
$\overline{SS}$	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

Table 21-6. Example SPI Baud Rate Selection (25 MHz Bus Clock)

SPPR2	SPPR1	SPPR0	SPR2	SPR1	SPR0	Baud Rate Divisor	Baud Rate
0	0	1	0	1	0	16	1.5625 Mbit/s
0	0	1	0	1	1	32	781.25 kbit/s
0	0	1	1	0	0	64	390.63 kbit/s
0	0	1	1	0	1	128	195.31 kbit/s
0	0	1	1	1	0	256	97.66 kbit/s
0	0	1	1	1	1	512	48.83 kbit/s
0	1	0	0	0	0	6	4.16667 Mbit/s
0	1	0	0	0	1	12	2.08333 Mbit/s
0	1	0	0	1	0	24	1.04167 Mbit/s
0	1	0	0	1	1	48	520.83 kbit/s
0	1	0	1	0	0	96	260.42 kbit/s
0	1	0	1	0	1	192	130.21 kbit/s
0	1	0	1	1	0	384	65.10 kbit/s
0	1	0	1	1	1	768	32.55 kbit/s
0	1	1	0	0	0	8	3.125 Mbit/s
0	1	1	0	0	1	16	1.5625 Mbit/s
0	1	1	0	1	0	32	781.25 kbit/s
0	1	1	0	1	1	64	390.63 kbit/s
0	1	1	1	0	0	128	195.31 kbit/s
0	1	1	1	0	1	256	97.66 kbit/s
0	1	1	1	1	0	512	48.83 kbit/s
0	1	1	1	1	1	1024	24.41 kbit/s
1	0	0	0	0	0	10	2.5 Mbit/s
1	0	0	0	0	1	20	1.25 Mbit/s
1	0	0	0	1	0	40	625 kbit/s
1	0	0	0	1	1	80	312.5 kbit/s
1	0	0	1	0	0	160	156.25 kbit/s
1	0	0	1	0	1	320	78.13 kbit/s
1	0	0	1	1	0	640	39.06 kbit/s
1	0	0	1	1	1	1280	19.53 kbit/s
1	0	1	0	0	0	12	2.08333 Mbit/s
1	0	1	0	0	1	24	1.04167 Mbit/s
1	0	1	0	1	0	48	520.83 kbit/s
1	0	1	0	1	1	96	260.42 kbit/s
1	0	1	1	0	0	192	130.21 kbit/s
1	0	1	1	0	1	384	65.10 kbit/s
1	0	1	1	1	0	768	32.55 kbit/s
1	0	1	1	1	1	1536	16.28 kbit/s
1	1	0	0	0	0	14	1.78571 Mbit/s

24.4.4.3 Valid Flash Module Commands

Table 24-25 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 24-25. 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.

24.4.4.4 P-Flash Commands

Table 24-26 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 24-26. 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.

24.4.6.2 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 24-31. Erase Verify Block Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x02	Flash block selection code [1:0]. See Table 24-32

Table 24-32. Flash block selection code description

Selection code[1:0]	Flash block to be verified
00	EEPROM
01	Invalid (ACCERR)
10	Invalid (ACCERR)
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 24-33. 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[1]
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read[2] or if blank check failed.
	MGSTAT0	Set if any non-correctable errors have been encountered during the read[2] or if blank check failed.

1 As defined by the memory map for FTMRG32K1.

2 As found in the memory map for FTMRG32K1.

24.4.6.3 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 25-11. Flash Security States

SEC[1:0]	Status of Security
00	SECURED
01	SECURED[1]
10	UNSECURED
11	SECURED

1 Preferred SEC state to set MCU to secured state.

The security function in the Flash module is described in Section 25.5.

25.3.2.3 Flash CCOB Index Register (FCCOBIX)

The FCCOBIX register is used to index the FCCOB register for Flash memory operations.

Offset Module Base + 0x0002

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

= Unimplemented or Reserved

Figure 25-7. FCCOB Index Register (FCCOBIX)

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

Table 25-12. 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 25.3.2.11 Flash Common Command Object Register (FCCOB),” for more details.

25.3.2.4 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 25-8. Flash Reserved0 Register (FRSV0)

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

Chapter 27
64 KByte Flash Module (S12FTMRG64K1V1)

Table 27-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V01.04	17 Jun 2010	27.4.6.1/27-950 27.4.6.2/27-951 27.4.6.3/27-951 27.4.6.14/27-961	Clarify Erase Verify Commands Descriptions related to the bits MGSTAT[1:0] of the register FSTAT.
V01.05	20 aug 2010	27.4.6.2/27-951 27.4.6.12/27-958 27.4.6.13/27-960	Updated description of the commands RD1BLK, MLOADU and MLOADF
Rev.1.27	31 Jan 2011	27.3.2.9/27-933	Updated description of protection on Section 27.3.2.9

27.1 Introduction

The FTMRG64K1 module implements the following:

- 64Kbytes of P-Flash (Program Flash) memory
- 2 Kbytes of EEPROM memory

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

CAUTION

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

The Flash memory may be read as bytes and aligned words. Read access time is one bus cycle for bytes and aligned words. For misaligned words access, the CPU has to perform twice the byte read access command. For Flash memory, an erased bit reads 1 and a programmed bit reads 0.

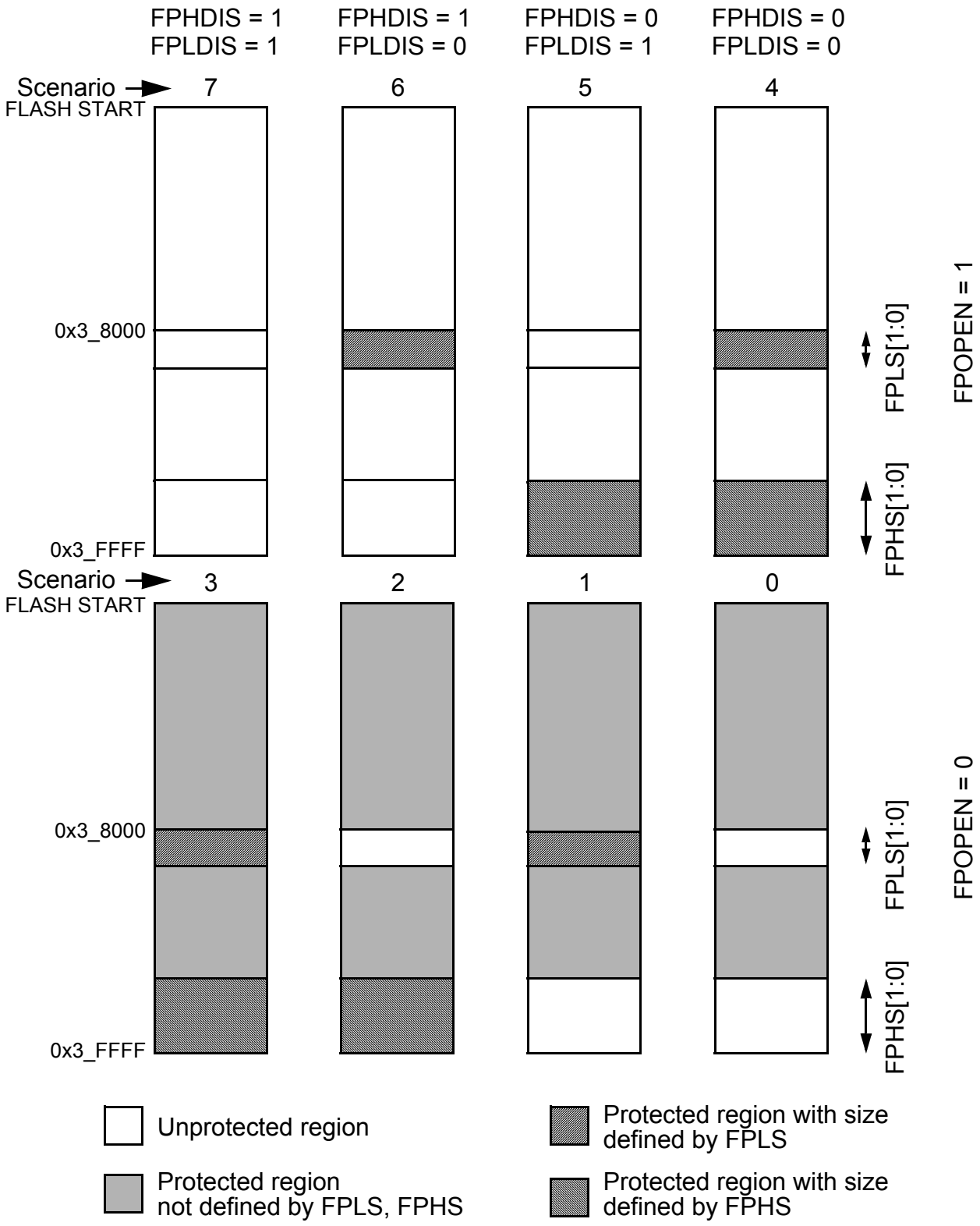


Figure 30-14. P-Flash Protection Scenarios

31.3.2.11 Flash Common Command Object Register (FCCOB)

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

Offset Module Base + 0x000A

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

Figure 31-16. Flash Common Command Object High Register (FCCOBHI)

Offset Module Base + 0x000B

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

Figure 31-17. Flash Common Command Object Low Register (FCCOBLO)

31.3.2.11.1 FCCOB - NVM Command Mode

NVM command mode uses the indexed FCCOB register to provide a command code and its relevant parameters to the Memory Controller. The user first sets up all required FCCOB fields and then initiates the command's execution by writing a 1 to the CCIF bit in the FSTAT register (a 1 written by the user clears the CCIF command completion flag to 0). When the user clears the CCIF bit in the FSTAT register all FCCOB parameter fields are locked and cannot be changed by the user until the command completes (as evidenced by the Memory Controller returning CCIF to 1). Some commands return information to the FCCOB register array.

The generic format for the FCCOB parameter fields in NVM command mode is shown in Table 31-24. The return values are available for reading after the CCIF flag in the FSTAT register has been returned to 1 by the Memory Controller. Writes to the unimplemented parameter fields (CCOBIX = 110 and CCOBIX = 111) are ignored with reads from these fields returning 0x0000.

Table 31-24 shows the generic Flash command format. The high byte of the first word in the CCOB array contains the command code, followed by the parameters for this specific Flash command. For details on the FCCOB settings required by each command, see the Flash command descriptions in Section 31.4.6.

Table 31-24. FCCOB - NVM Command Mode (Typical Usage)

CCOBIX[2:0]	Byte	FCCOB Parameter Fields (NVM Command Mode)
000	HI	FCMD[7:0] defining Flash command
	LO	6'h0, Global address [17:16]
001	HI	Global address [15:8]
	LO	Global address [7:0]