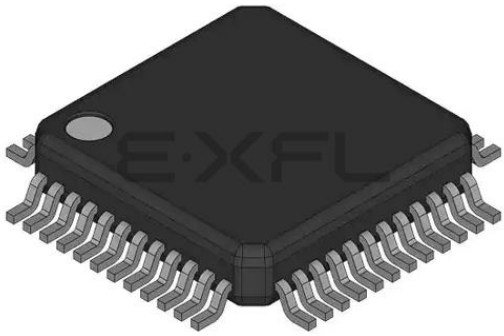


Welcome to E-XFL.COM

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	Obsolete
Core Processor	HCS12
Core Size	16-Bit
Speed	25MHz
Connectivity	EBI/EMI, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	31
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s12gc32vfae

0x0080–0x009F ATD (Analog-to-Digital Converter 10 Bit 8 Channel)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0080	ATDCTL0	Read:	0	0	0	0	0	0	0	0
		Write:								
0x0081	ATDCTL1	Read:	0	0	0	0	0	0	0	0
		Write:								
0x0082	ATDCTL2	Read:	ADPU	AFFC	AWAI	ETRIGLE	ETRIGP	ETRIG	ASCIE	ASCIF
		Write:								
0x0083	ATDCTL3	Read:	0	S8C	S4C	S2C	S1C	FIFO	FRZ1	FRZ0
		Write:								
0x0084	ATDCTL4	Read:	SRES8	SMP1	SMP0	PRS4	PRS3	PRS2	PRS1	PRS0
		Write:								
0x0085	ATDCTL5	Read:	DJM	DSGN	SCAN	MULT	0	CC	CB	CA
		Write:								
0x0086	ATDSTAT0	Read:	SCF	0	ETORF	FIFOR	0	CC2	CC1	CC0
		Write:								
0x0087	Reserved	Read:	0	0	0	0	0	0	0	0
		Write:								
0x0088	ATDTEST0	Read:	0	0	0	0	0	0	0	0
		Write:								
0x0089	ATDTEST1	Read:	0	0	0	0	0	0	0	SC
		Write:								
0x008A	Reserved	Read:	0	0	0	0	0	0	0	0
		Write:								
0x008B	ATDSTAT1	Read:	CCF7	CCF6	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0
		Write:								
0x008C	Reserved	Read:	0	0	0	0	0	0	0	0
		Write:								
0x008D	ATDDIEN	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
0x008E	Reserved	Read:	0	0	0	0	0	0	0	0
		Write:								
0x008F	PORTAD	Read:	Bit7	6	5	4	3	2	1	BIT 0
		Write:								
0x0090	ATDDR0H	Read:	Bit15	14	13	12	11	10	9	Bit8
		Write:								
0x0091	ATDDR0L	Read:	Bit7	Bit6	0	0	0	0	0	0
		Write:								
0x0092	ATDDR1H	Read:	Bit15	14	13	12	11	10	9	Bit8
		Write:								
0x0093	ATDDR1L	Read:	Bit7	Bit6	0	0	0	0	0	0
		Write:								
0x0094	ATDDR2H	Read:	Bit15	14	13	12	11	10	9	Bit8
		Write:								
0x0095	ATDDR2L	Read:	Bit7	Bit6	0	0	0	0	0	0
		Write:								

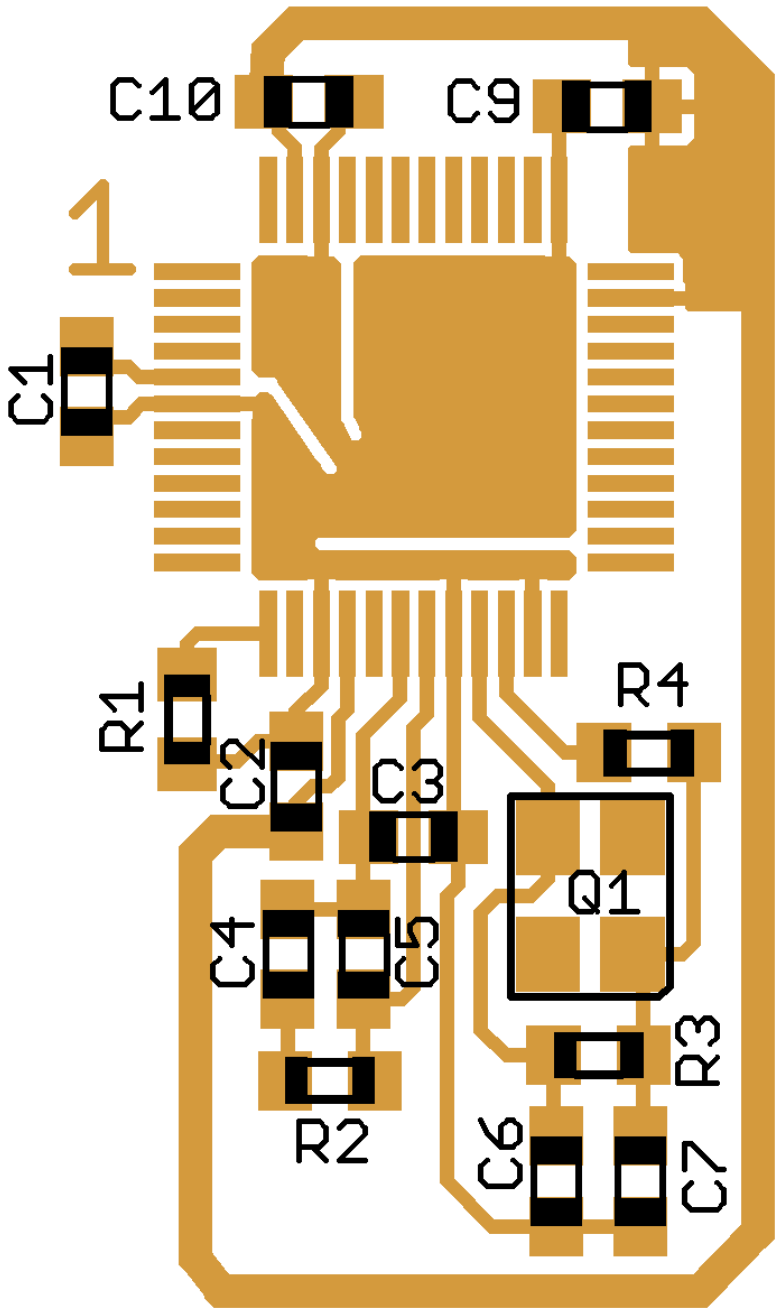


Figure 1-18. Recommended PCB Layout for 48 LQFP Pierce Oscillator

Table 4-1. External System Pins Associated With MEBI (continued)

Pin Name	Pin Functions	Description
PE4/ECLK	PE4	General-purpose I/O pin, see PORTE and DDRE registers.
	ECLK	Bus timing reference clock, can operate as a free-running clock at the system clock rate or to produce one low-high clock per visible access, with the high period stretched for slow accesses. ECLK is controlled by the NECLK bit in PEAR, the IVIS bit in MODE, and the ESTR bit in EBICTL.
PE3/$\overline{\text{LSTRB}}$/ $\overline{\text{TAGLO}}$	PE3	General-purpose I/O pin, see PORTE and DDRE registers.
	$\overline{\text{LSTRB}}$	Low strobe bar, 0 indicates valid data on D7–D0.
	SZ8	In special peripheral mode, this pin is an input indicating the size of the data transfer (0 = 16-bit; 1 = 8-bit).
	$\overline{\text{TAGLO}}$	In expanded wide mode or emulation narrow modes, when instruction tagging is on and low strobe is enabled, a 0 at the falling edge of E tags the low half of the instruction word being read into the instruction queue.
PE2/R/$\overline{\text{W}}$	PE2	General-purpose I/O pin, see PORTE and DDRE registers.
	R/$\overline{\text{W}}$	Read/write, indicates the direction of internal data transfers. This is an output except in special peripheral mode where it is an input.
PE1/$\overline{\text{IRQ}}$	PE1	General-purpose input-only pin, can be read even if $\overline{\text{IRQ}}$ enabled.
	$\overline{\text{IRQ}}$	Maskable interrupt request, can be level sensitive or edge sensitive.
PE0/$\overline{\text{XIRQ}}$	PE0	General-purpose input-only pin.
	$\overline{\text{XIRQ}}$	Non-maskable interrupt input.
PK7/$\overline{\text{ECS}}$	PK7	General-purpose I/O pin, see PORTK and DDRK registers.
	$\overline{\text{ECS}}$	Emulation chip select
PK6/$\overline{\text{XCS}}$	PK6	General-purpose I/O pin, see PORTK and DDRK registers.
	$\overline{\text{XCS}}$	External data chip select
PK5/X19 thru PK0/X14	PK5–PK0	General-purpose I/O pins, see PORTK and DDRK registers.
	X19–X14	Memory expansion addresses

Detailed descriptions of these pins can be found in the device overview chapter.

4.3 Memory Map and Register Definition

A summary of the registers associated with the MEBI sub-block is shown in Table 4-2. Detailed descriptions of the registers and bits are given in the subsections that follow. On most chips the registers are mappable. Therefore, the upper bits may not be all 0s as shown in the table and descriptions.

The DBG in DBG mode includes these distinctive features:

- Three comparators (A, B, and C)
  - Dual mode, comparators A and B used to compare addresses
  - Full mode, comparator A compares address and comparator B compares data
  - Can be used as trigger and/or breakpoint
  - Comparator C used in LOOP1 capture mode or as additional breakpoint
- Four capture modes
  - Normal mode, change-of-flow information is captured based on trigger specification
  - Loop1 mode, comparator C is dynamically updated to prevent redundant change-of-flow storage.
  - Detail mode, address and data for all cycles except program fetch (P) and free (f) cycles are stored in trace buffer
  - Profile mode, last instruction address executed by CPU is returned when trace buffer address is read
- Two types of breakpoint or debug triggers
  - Break just before a specific instruction will begin execution (tag)
  - Break on the first instruction boundary after a match occurs (force)
- BDM or SWI breakpoint
  - Enter BDM on breakpoint (BDM)
  - Execute SWI on breakpoint (SWI)
- Nine trigger modes for comparators A and B
  - A
  - A or B
  - A then B
  - A and B, where B is data (full mode)
  - A and not B, where B is data (full mode)
  - Event only B, store data
  - A then event only B, store data
  - Inside range, $A \leq address \leq B$
  - Outside range, address $< A$ or address $> B$
- Comparator C provides an additional tag or force breakpoint when capture mode is not configured in LOOP1 mode.
- Sixty-four word (16 bits wide) trace buffer for storing change-of-flow information, event only data and other bus information.
  - Source address of taken conditional branches (long, short, bit-conditional, and loop constructs)
  - Destination address of indexed JMP, JSR, and CALL instruction.
  - Destination address of RTI, RTS, and RTC instructions
  - Vector address of interrupts, except for SWI and BDM vectors

7.4.2.6.3 Detail Mode

In the detail mode, address and data for all cycles except program fetch (P) and free (f) cycles are stored in trace buffer. This mode is intended to supply additional information on indexed, indirect addressing modes where storing only the destination address would not provide all information required for a user to determine where his code was in error.

7.4.2.6.4 Profile Mode

This mode is intended to allow a host computer to poll a running target and provide a histogram of program execution. Each read of the trace buffer address will return the address of the last instruction executed. The DBGCNT register is not incremented and the trace buffer does not get filled. The ARM bit is not used and all breakpoints and all other debug functions will be disabled.

7.4.2.7 Storage Memory

The storage memory is a 64 words deep by 16-bits wide dual port RAM array. The CPU accesses the RAM array through a single memory location window (DBGTBH:DBGTBL). The DBG module stores trace information in the RAM array in a circular buffer format. As data is read via the CPU, a pointer into the RAM will increment so that the next CPU read will receive fresh information. In all trigger modes except for event-only and detail capture mode, the data stored in the trace buffer will be change-of-flow addresses. change-of-flow addresses are defined as follows:

- Source address of conditional branches (long, short, BRSET, and loop constructs) taken
- Destination address of indexed JMP, JSR, and CALL instruction
- Destination address of RTI, RTS, and RTC instructions
- Vector address of interrupts except for SWI and BDM vectors

In the event-only trigger modes only the 16-bit data bus value corresponding to the event is stored. In the detail capture mode, address and then data are stored for all cycles except program fetch (P) and free (f) cycles.

7.4.2.8 Storing Data in Memory Storage Buffer

7.4.2.8.1 Storing with Begin-Trigger

Storing with begin-trigger can be used in all trigger modes. When DBG mode is enabled and armed in the begin-trigger mode, data is not stored in the trace buffer until the trigger condition is met. As soon as the trigger condition is met, the DBG module will remain armed until 64 words are stored in the trace buffer. If the trigger is at the address of the change-of-flow instruction the change-of-flow associated with the trigger event will be stored in the trace buffer.

7.4.2.8.2 Storing with End-Trigger

Storing with end-trigger cannot be used in event-only trigger modes. When DBG mode is enabled and armed in the end-trigger mode, data is stored in the trace buffer until the trigger condition is met. When the trigger condition is met, the DBG module will become de-armed and no more data will be stored. If

8.1.2.2 MCU Operating Modes

- **Stop Mode**

  Entering stop mode causes all clocks to halt and thus the system is placed in a minimum power standby mode. This aborts any conversion sequence in progress. During recovery from stop mode, there must be a minimum delay for the stop recovery time, t_{SR}, before initiating a new ATD conversion sequence.

- **Wait Mode**

  Entering wait mode the ATD conversion either continues or aborts for low power depending on the logical value of the AWAIT bit.

- **Freeze Mode**

  In freeze mode the ATD10B8C will behave according to the logical values of the FRZ1 and FRZ0 bits. This is useful for debugging and emulation.

8.1.3 Block Diagram

Figure 8-1 is a block diagram of the ATD.

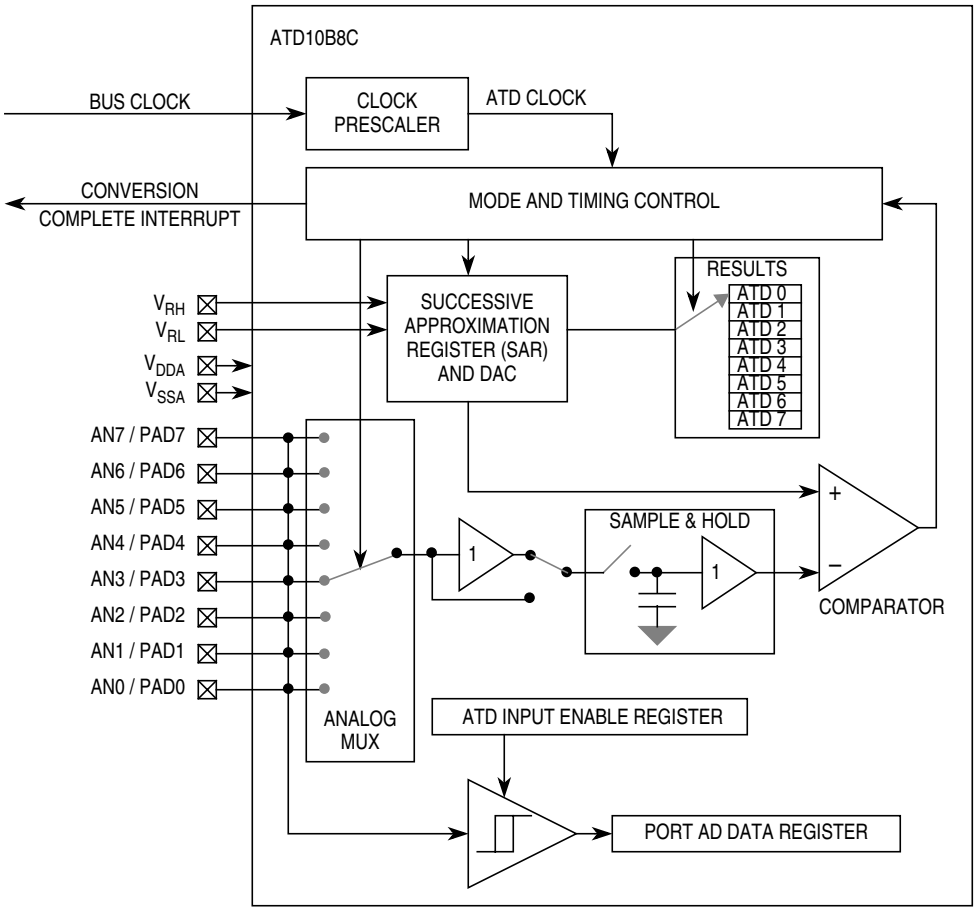


Figure 8-1. ATD10B8C Block Diagram

Chapter 9
Clocks and Reset Generator (CRGV4) Block Description

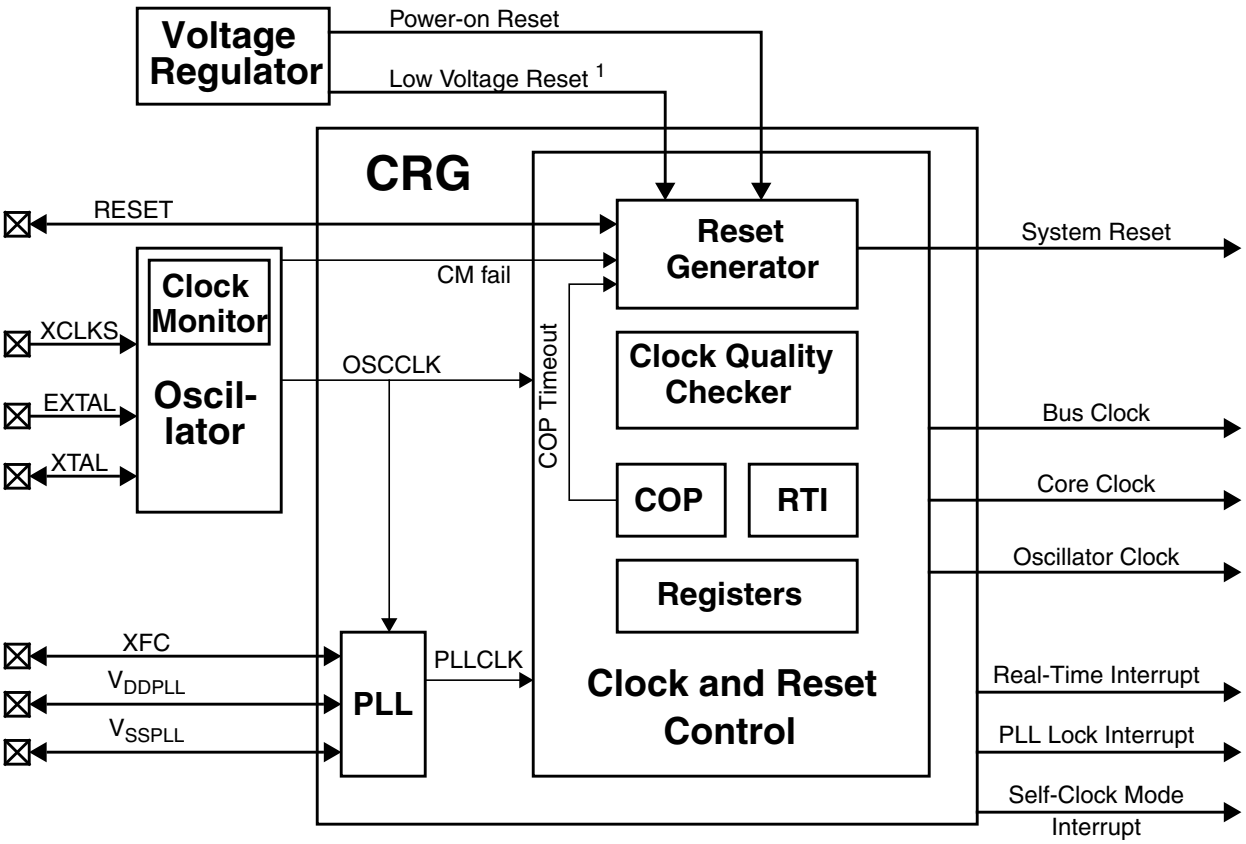
9.1 Introduction

This specification describes the function of the clocks and reset generator (CRGV4).

9.1.1 Features

The main features of this block are:

- Phase-locked loop (PLL) frequency multiplier
  - Reference divider
  - Automatic bandwidth control mode for low-jitter operation
  - Automatic frequency lock detector
  - CPU interrupt on entry or exit from locked condition
  - Self-clock mode in absence of reference clock
- System clock generator
  - Clock quality check
  - Clock switch for either oscillator- or PLL-based system clocks
  - User selectable disabling of clocks during wait mode for reduced power consumption
- Computer operating properly (COP) watchdog timer with time-out clear window
- System reset generation from the following possible sources:
  - Power-on reset
  - Low voltage reset
    Refer to the device overview section for availability of this feature.
  - COP reset
  - Loss of clock reset
  - External pin reset
- Real-time interrupt (RTI)



1 Refer to the device overview section for availability of the low-voltage reset feature.

Figure 9-1. CRG Block Diagram

9.2 External Signal Description

This section lists and describes the signals that connect off chip.

9.2.1 V_{DDPLL}, V_{SSPLL} — PLL Operating Voltage, PLL Ground

These pins provides operating voltage (V_{DDPLL}) and ground (V_{SSPLL}) for the PLL circuitry. This allows the supply voltage to the PLL to be independently bypassed. Even if PLL usage is not required V_{DDPLL} and V_{SSPLL} must be connected properly.

9.2.2 XFC — PLL Loop Filter Pin

A passive external loop filter must be placed on the XFC pin. The filter is a second-order, low-pass filter to eliminate the VCO input ripple. The value of the external filter network and the reference frequency determines the speed of the corrections and the stability of the PLL. Refer to the device overview chapter for calculation of PLL loop filter (XFC) components. If PLL usage is not required the XFC pin must be tied to V_{DDPLL}.

9.4.5 Computer Operating Properly Watchdog (COP)

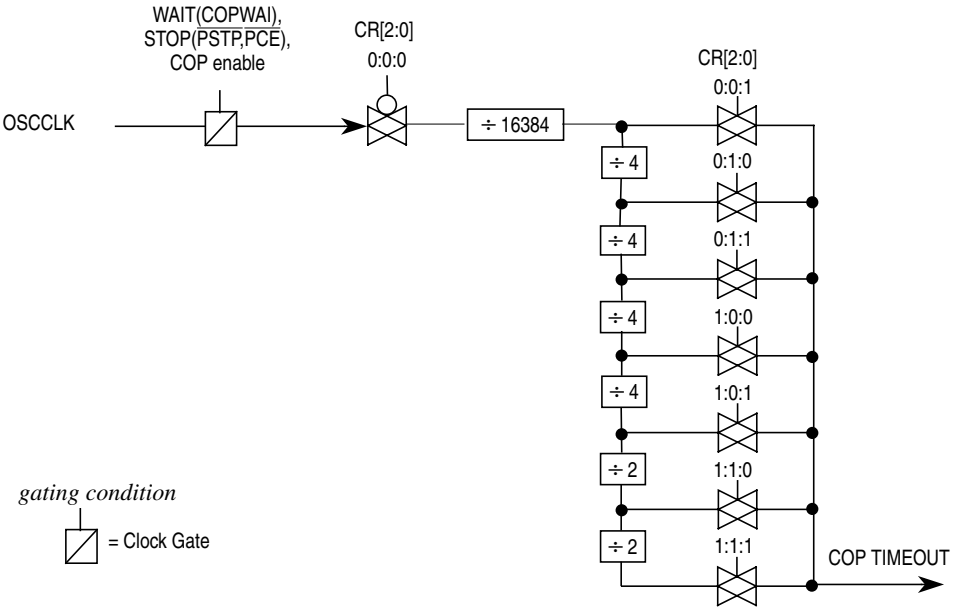


Figure 9-21. Clock Chain for COP

The COP (free running watchdog timer) enables the user to check that a program is running and sequencing properly. The COP is disabled out of reset. When the COP is being used, software is responsible for keeping the COP from timing out. If the COP times out it is an indication that the software is no longer being executed in the intended sequence; thus a system reset is initiated (see Section 9.5.2, “Computer Operating Properly Watchdog (COP) Reset).” The COP runs with a gated OSCCLK (see Section Figure 9-21., “Clock Chain for COP”). Three control bits in the COPCTL register allow selection of seven COP time-out periods.

When COP is enabled, the program must write 0x0055 and 0x00AA (in this order) to the ARMCOP register during the selected time-out period. As soon as this is done, the COP time-out period is restarted. If the program fails to do this and the COP times out, the part will reset. Also, if any value other than 0x0055 or 0x00AA is written, the part is immediately reset.

Windowed COP operation is enabled by setting WCOP in the COPCTL register. In this mode, writes to the ARMCOP register to clear the COP timer must occur in the last 25% of the selected time-out period. A premature write will immediately reset the part.

If PCE bit is set, the COP will continue to run in pseudo-stop mode.

9.4.6 Real-Time Interrupt (RTI)

The RTI can be used to generate a hardware interrupt at a fixed periodic rate. If enabled (by setting RTIE=1), this interrupt will occur at the rate selected by the RTICTL register. The RTI runs with a gated OSCCLK (see Section Figure 9-22., “Clock Chain for RTI”). At the end of the RTI time-out period the RTIF flag is set to 1 and a new RTI time-out period starts immediately.

A write to the RTICTL register restarts the RTI time-out period.

— a) the 14 most significant bits of the extended identifier plus the SRR and IDE bits of CAN 2.0B messages or

— b) the 11 bits of the standard identifier, the RTR and IDE bits of CAN 2.0A/B messages. Figure 10-40 shows how the first 32-bit filter bank (CANIDAR0–CANIDA3, CANIDMR0–3CANIDMR) produces filter 0 and 1 hits. Similarly, the second filter bank (CANIDAR4–CANIDAR7, CANIDMR4–CANIDMR7) produces filter 2 and 3 hits.

- Eight identifier acceptance filters, each to be applied to the first 8 bits of the identifier. This mode implements eight independent filters for the first 8 bits of a CAN 2.0A/B compliant standard identifier or a CAN 2.0B compliant extended identifier. Figure 10-41 shows how the first 32-bit filter bank (CANIDAR0–CANIDAR3, CANIDMR0–CANIDMR3) produces filter 0 to 3 hits. Similarly, the second filter bank (CANIDAR4–CANIDAR7, CANIDMR4–CANIDMR7) produces filter 4 to 7 hits.
- Closed filter. No CAN message is copied into the foreground buffer RxFG, and the RXF flag is never set.

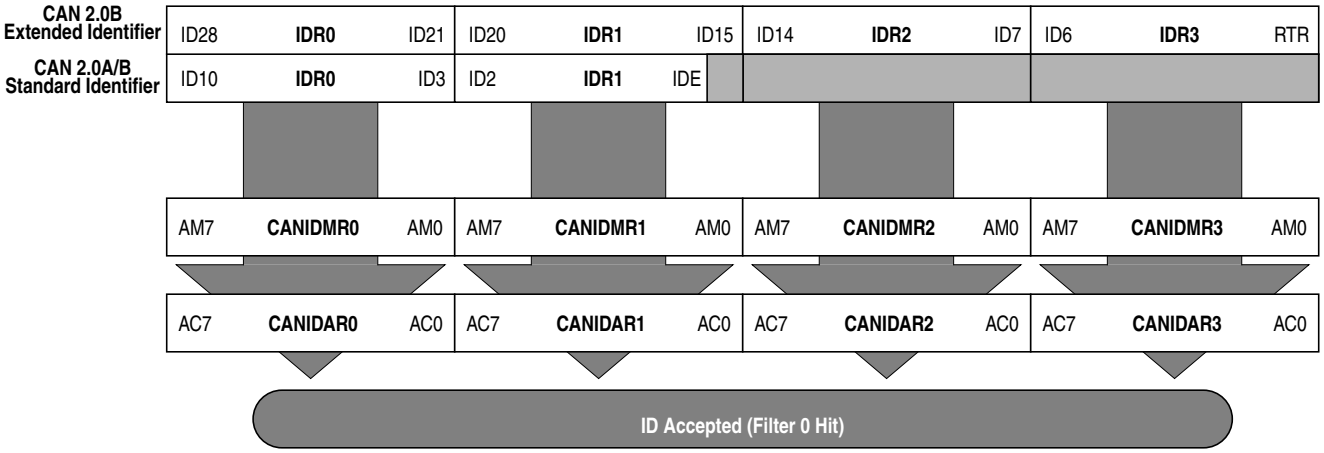


Figure 10-39. 32-bit Maskable Identifier Acceptance Filter

Table 12-10. PWMSDN Field Descriptions

Field	Description
7 PWMIF	**PWM Interrupt Flag** — Any change from passive to asserted (active) state or from active to passive state will be flagged by setting the PWMIF flag = 1. The flag is cleared by writing a logic 1 to it. Writing a 0 has no effect. 0 No change on PWM5IN input. 1 Change on PWM5IN input
6 PWMIE	**PWM Interrupt Enable** — If interrupt is enabled an interrupt to the CPU is asserted. 0 PWM interrupt is disabled. 1 PWM interrupt is enabled.
5 PWMRSTRT	**PWM Restart** — The PWM can only be restarted if the PWM channel input 5 is deasserted. After writing a logic 1 to the PWMRSTRT bit (trigger event) the PWM channels start running after the corresponding counter passes next “counter = 0” phase. Also, if the PWM5ENA bit is reset to 0, the PWM do not start before the counter passes 0x0000. The bit is always read as 0.
4 PWMLVL	**PWM Shutdown Output Level** — If active level as defined by the PWM5IN input, gets asserted all enabled PWM channels are immediately driven to the level defined by PWMLVL. 0 PWM outputs are forced to 0 1 PWM outputs are forced to 1.
2 PWM5IN	**PWM Channel 5 Input Status** — This reflects the current status of the PWM5 pin.
1 PWM5INL	**PWM Shutdown Active Input Level for Channel 5** — If the emergency shutdown feature is enabled (PWM5ENA = 1), this bit determines the active level of the PWM5 channel. 0 Active level is low 1 Active level is high
0 PWM5ENA	**PWM Emergency Shutdown Enable** — If this bit is logic 1 the pin associated with channel 5 is forced to input and the emergency shutdown feature is enabled. All the other bits in this register are meaningful only if PWM5ENA = 1. 0 PWM emergency feature disabled. 1 PWM emergency feature is enabled.

Chapter 13
Serial Communications Interface (S12SCIV2) Block Description

13.1 Introduction

This block guide provide an overview of serial communication interface (SCI) module. The SCI allows asynchronous serial communications with peripheral devices and other CPUs.

13.1.1 Glossary

IRQ — Interrupt Request

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

13.1.2 Features

The SCI includes these distinctive features:

- Full-duplex operation
- Standard mark/space non-return-to-zero (NRZ) format
- 13-bit baud rate selection
- Programmable 8-bit or 9-bit data format
- Separately enabled transmitter and receiver
- Programmable transmitter output parity
- Two receiver wake up methods:
  - Idle line wake-up
  - Address mark wake-up
- Interrupt-driven operation with eight flags:
  - Transmitter empty

13.3.2.6 SCI Data Registers (SCIDRH and SCIDRL)

Module Base + 0x_0006

	7	6	5	4	3	2	1	0
R	R8	T8	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

Module Base + 0x_0007

	7	6	5	4	3	2	1	0
R	R7	R6	R5	R4	R3	R2	R1	R0
W	T7	T6	T5	T4	T3	T2	T1	T0
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 13-8. SCI Data Registers (SCIDRH and SCIDRL)

Read: Anytime; reading accesses SCI receive data register

Write: Anytime; writing accesses SCI transmit data register; writing to R8 has no effect

Table 13-7. SCIDRH AND SCIDRL Field Descriptions

Field	Description
7 R8	**Received Bit 8** — R8 is the ninth data bit received when the SCI is configured for 9-bit data format (M = 1).
6 T8	**Transmit Bit 8** — T8 is the ninth data bit transmitted when the SCI is configured for 9-bit data format (M = 1).
7–0 R[7:0] T[7:0]	**Received Bits** — Received bits seven through zero for 9-bit or 8-bit data formats **Transmit Bits** — Transmit bits seven through zero for 9-bit or 8-bit formats

NOTE

If the value of T8 is the same as in the previous transmission, T8 does not have to be rewritten.The same value is transmitted until T8 is rewritten

In 8-bit data format, only SCI data register low (SCIDRL) needs to be accessed.

When transmitting in 9-bit data format and using 8-bit write instructions, write first to SCI data register high (SCIDRH), then SCIDRL.

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 18-17. RESERVED3

All bits read 0 and are not writable.

18.3.2.12 RESERVED4

This register is reserved for factory testing and is not accessible to the user.

Module Base + 0x000D

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

= Unimplemented or Reserved

Figure 18-18. RESERVED4

All bits read 0 and are not writable.

18.3.2.13 RESERVED5

This register is reserved for factory testing and is not accessible to the user.

Module Base + 0x000E

	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 18-19. RESERVED5

All bits read 0 and are not writable.

18.3.2.14 RESERVED6

This register is reserved for factory testing and is not accessible to the user.

Table 19-5. FSEC Field Descriptions

Field	Description
7–6 KEYEN[1:0]	**Backdoor Key Security Enable Bits** — The KEYEN[1:0] bits define the enabling of the backdoor key access to the Flash module as shown in Table 19-6.
5–2 NV[5:2]	**Nonvolatile Flag Bits** — The NV[5:2] bits are available to the user as nonvolatile flags.
1–0 SEC[1:0]	**Flash Security Bits** — The SEC[1:0] bits define the security state of the MCU as shown in Table 19-7. If the Flash module is unsecured using backdoor key access, the SEC[1:0] bits are forced to 1:0.

Table 19-6. Flash KEYEN States

KEYEN[1:0]	Status of Backdoor Key Access
00	DISABLED
01(1)	DISABLED
10	ENABLED
11	DISABLED

1. Preferred KEYEN state to disable Backdoor Key Access.

Table 19-7. Flash Security States

SEC[1:0]	Status of Security
00	Secured
01(1)	Secured
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 19.4.3, “Flash Module Security”.

19.3.2.3 RESERVED1

This register is reserved for factory testing and is not accessible to the user.

Module Base + 0x0002

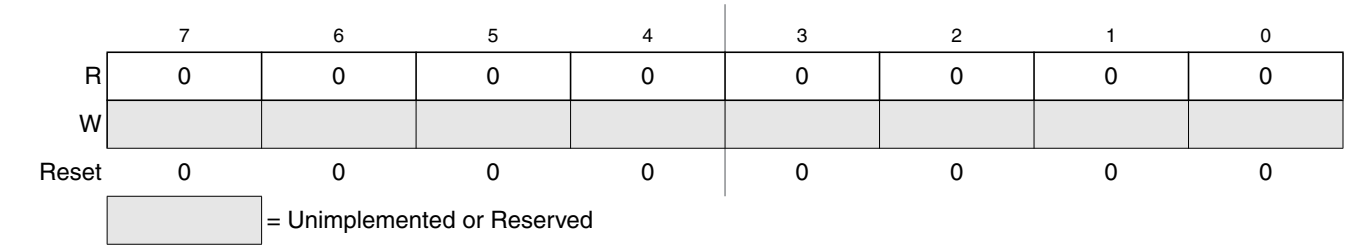
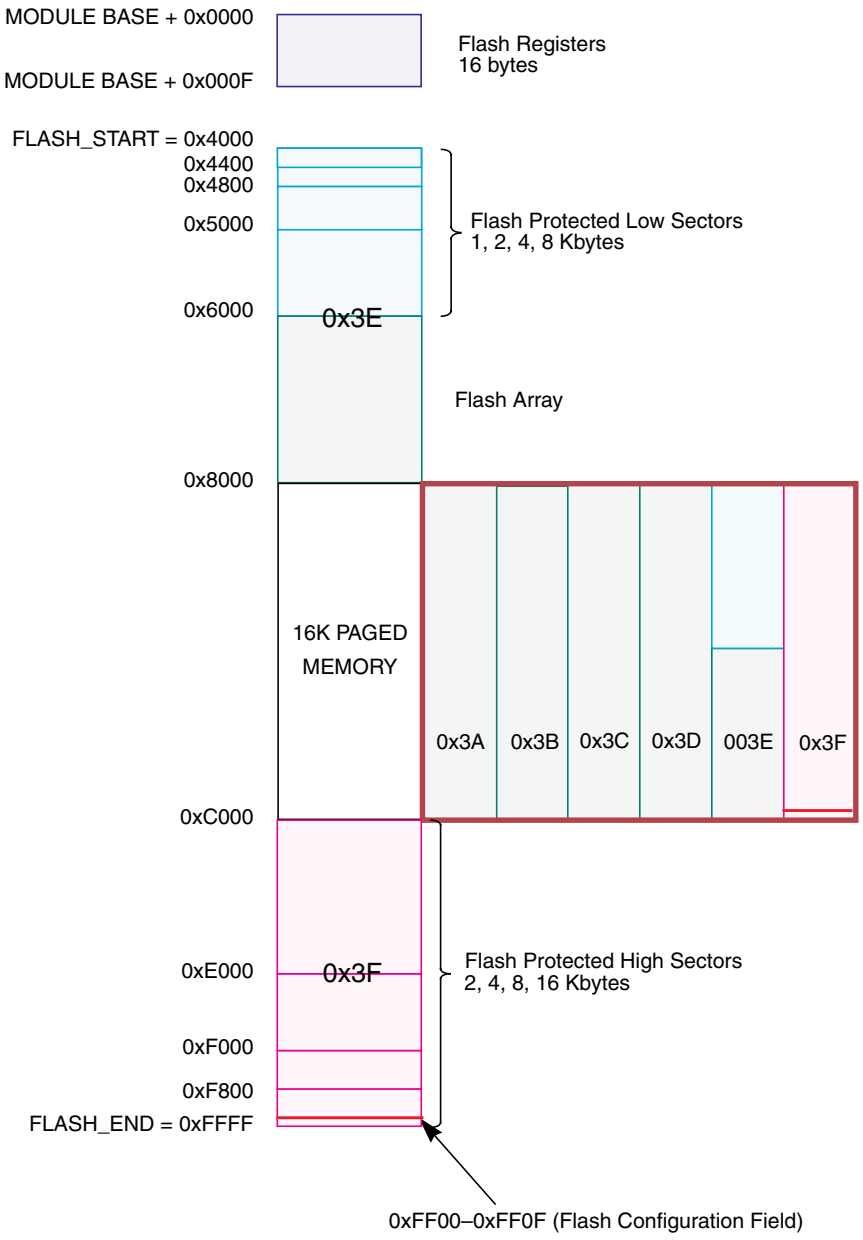


Figure 19-8. RESERVED1

All bits read 0 and are not writable.



Note: 0x3A–0x3F correspond to the PPAGE register content

Figure 20-4. Flash Memory Map

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 20-19. RESERVED3

All bits read 0 and are not writable.

20.3.2.12 RESERVED4

This register is reserved for factory testing and is not accessible to the user.

Module Base + 0x000D

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

= Unimplemented or Reserved

Figure 20-20. RESERVED4

All bits read 0 and are not writable.

20.3.2.13 RESERVED5

This register is reserved for factory testing and is not accessible to the user.

Module Base + 0x000E

	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 20-21. RESERVED5

All bits read 0 and are not writable.

20.3.2.14 RESERVED6

This register is reserved for factory testing and is not accessible to the user.

Table 21-12. Flash Protection Scenario Transitions

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

1. Allowed transitions marked with X.

21.3.2.6 Flash Status Register (FSTAT)

The FSTAT register defines the status of the Flash command controller and the results of command execution.

Module Base + 0x0005

	7	6	5	4	3	2	1	0
R	CBEIF	CCIF	PVIOL	ACCERR	0	BLANK	FAIL	DONE
W								
Reset	1	1	0	0	0	0	0	1

= Unimplemented or Reserved

Figure 21-10. Flash Status Register (FSTAT)

In normal modes, bits CBEIF, PVIOL, and ACCERR are readable and writable, bits CCIF and BLANK are readable and not writable, remaining bits, including FAIL and DONE, read 0 and are not writable. In special modes, FAIL is readable and writable while DONE is readable but not writable. FAIL must be clear in special modes when starting a command write sequence.

Table 21-13. FSTAT Field Descriptions

Field	Description
7 CBEIF	**Command Buffer Empty Interrupt Flag** — The CBEIF flag indicates that the address, data and command buffers are empty so that a new command write sequence can be started. The CBEIF flag is cleared by writing a 1 to CBEIF. Writing a 0 to the CBEIF flag has no effect on CBEIF. Writing a 0 to CBEIF after writing an aligned word to the Flash address space but before CBEIF is cleared will abort a command write sequence and cause the ACCERR flag in the FSTAT register to be set. Writing a 0 to CBEIF outside of a command write sequence will not set the ACCERR flag. The CBEIF flag is used together with the CBEIE bit in the FCNFG register to generate an interrupt request (see Figure 21-26). 0 Buffers are full 1 Buffers are ready to accept a new command
6 CCIF	**Command Complete Interrupt Flag** — The CCIF flag indicates that there are no more commands pending. The CCIF flag is cleared when CBEIF is clear and sets automatically upon completion of all active and pending commands. The CCIF flag does not set when an active commands completes and a pending command is fetched from the command buffer. Writing to the CCIF flag has no effect. The CCIF flag is used together with the CCIE bit in the FCNFG register to generate an interrupt request (see Figure 21-26). 0 Command in progress 1 All commands are completed

21.4.4 Flash Reset Sequence

On each reset, the Flash module executes a reset sequence to hold CPU activity while loading the following registers from the Flash array memory according to Table 21-1:

- FPROT — Flash Protection Register (see Section 21.3.2.5)
- FSEC — Flash Security Register (see Section 21.3.2.2)

21.4.4.1 Reset While Flash Command Active

If a reset occurs while any Flash command is in progress, that command will be immediately aborted. The state of the word being programmed or the sector/array being erased is not guaranteed.

21.4.5 Interrupts

The Flash module can generate an interrupt when all Flash commands have completed execution or the Flash address, data, and command buffers are empty.

Table 21-17. Flash Interrupt Sources

Interrupt Source	Interrupt Flag	Local Enable	Global (CCR) Mask
Flash Address, Data, and Command Buffers are empty	CBEIF (FSTAT register)	CBEIE	I Bit
All Flash commands have completed execution	CCIF (FSTAT register)	CCIE	I Bit

NOTE

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

21.4.5.1 Description of Interrupt Operation

Figure 21-26 shows the logic used for generating interrupts.

The Flash module uses the CBEIF and CCIF flags in combination with the enable bits CBIE and CCIE to discriminate for the generation of interrupts.

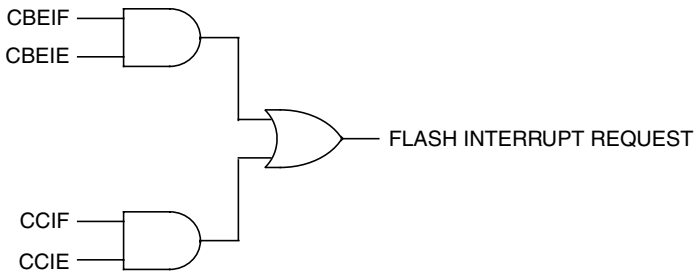


Figure 21-26. Flash Interrupt Implementation

For a detailed description of these register bits, refer to Section 21.3.2.4, “Flash Configuration Register (FCNFG)” and Section 21.3.2.6, “Flash Status Register (FSTAT)”.