

Welcome to [E-XFL.COM](#)

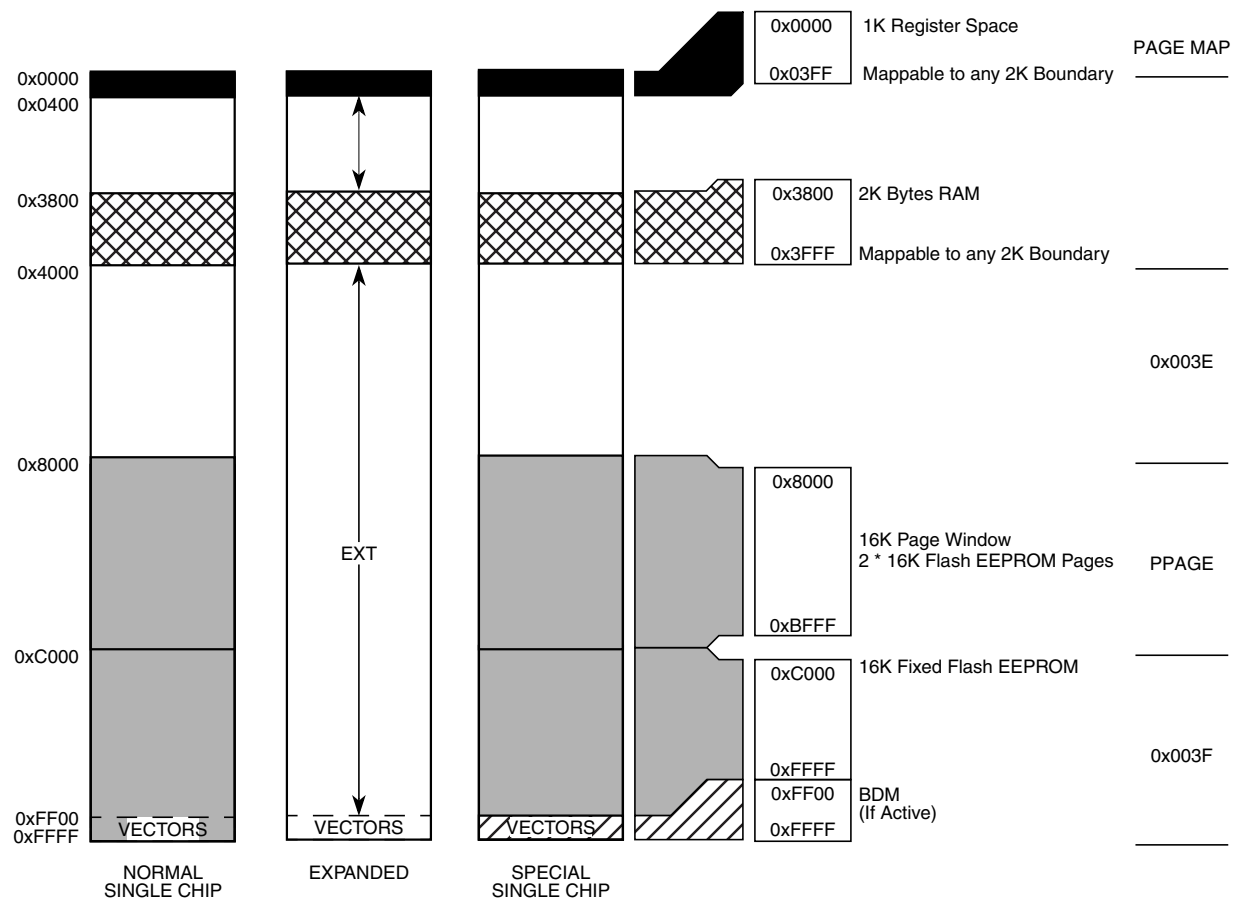
What is "[Embedded - Microcontrollers](#)"?

"[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	CANbus, EBI/EMI, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	35
Program Memory Size	96KB (96K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K 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	52-LQFP
Supplier Device Package	52-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12c96vpber



The figure shows a useful map, which is not the map out of reset. After reset the map is:
 0x0000–0x03FF: Register space
 0x0800–0x0FFF: 2K RAM

Flash erase sector size is 512 bytes

The flash page 0x003E is visible at 0x4000–0x7FFF in the memory map if ROMHM = 0.

In the figure ROMHM = 1 removing page 0x003E from 0x4000–0x7FFF.

Figure 1-5. MC9S12C32 and MC9S12GC32 User Configurable Memory Map

when mapping PWM channels to Port T in an 80QFP option, the associated PWM channels are then mapped to both Port P and Port T.

2.3.2.6.5 Port AD Pull Device Enable Register (PERAD)

Module Base + 0x0034

	7	6	5	4	3	2	1	0
R	PERAD7	PERAD6	PERAD5	PERAD4	PERAD3	PERAD2	PERAD1	PERAD0
W								
Reset	0	0	0	0	0	0	0	0

Figure 2-44. Port AD Pull Device Enable Register (PERAD)

Read: Anytime.

Write: Anytime.

Table 2-36. PERAD Field Descriptions

Field	Description
7–0 PERAD[7:0]	Pull Device Enable Port AD — This register configures whether a pull-up or a pull-down device is activated, if the port is used as input. This bit has no effect if the port is used as output. Out of reset no pull device is enabled. It is not possible to enable pull devices when a associated ATD channel is enabled simultaneously. 0 Pull-up or pull-down device is disabled. 1 Either a pull-up or pull-down device is enabled.

2.3.2.6.6 Port AD Polarity Select Register (PPSAD)

Module Base + 0x0035

	7	6	5	4	3	2	1	0
R	PPSAD7	PPSAD6	PPSAD5	PPSAD4	PPSAD3	PPSAD2	PPSAD1	PPSAD0
W								
Reset	0	0	0	0	0	0	0	0

Figure 2-45. Port AD Polarity Select Register (PPSAD)

Read: Anytime.

Write: Anytime.

Table 2-37. PPSAD Field Descriptions

Field	Description
7–0 PPSAD[7:0]	Pull Select Port AD — This register selects whether a pull-down or a pull-up device is connected to the pin. 0 A pull-up device is connected to the associated port AD pin, if enabled by the associated bit in register PERAD and if the port is used as input. 1 A pull-down device is connected to the associated port AD pin, if enabled by the associated bit in register PERAD and if the port is used as input.

3.3.2.9 Program Page Index Register (PPAGE)

Module Base + 0x0030

Starting address location affected by INITRG register setting.

	7	6	5	4	3	2	1	0
R	0	0	PIX5	PIX4	PIX3	PIX2	PIX1	PIX0
W								
Reset ¹	—	—	—	—	—	—	—	—

1. The reset state of this register is controlled at chip integration. Please refer to the device overview section to determine the actual reset state of this register.

 = Unimplemented or Reserved

Figure 3-11. Program Page Index Register (PPAGE)

Read: Anytime

Write: Determined at chip integration. Generally it's: "write anytime in all modes;" on some devices it will be: "write only in special modes." Check specific device documentation to determine which applies.

Reset: Defined at chip integration as either 0x00 (paired with write in any mode) or 0x3C (paired with write only in special modes), see device overview chapter.

The HCS12 core architecture limits the physical address space available to 64K bytes. The program page index register allows for integrating up to 1M byte of FLASH or ROM into the system by using the six page index bits to page 16K byte blocks into the program page window located from 0x8000 to 0xBFFF as defined in [Table 3-14](#). CALL and RTC instructions have special access to read and write this register without using the address bus.

NOTE

Normal writes to this register take one cycle to go into effect. Writes to this register using the special access of the CALL and RTC instructions will be complete before the end of the associated instruction.

Table 3-13. MEMSIZ0 Field Descriptions

Field	Description
5:0 PIX[5:0]	Program Page Index Bits 5:0 — These page index bits are used to select which of the 64 FLASH or ROM array pages is to be accessed in the program page window as shown in Table 3-14 .

4.1.2 Modes of Operation

- Normal expanded wide mode
Ports A and B are configured as a 16-bit multiplexed address and data bus and port E provides bus control and status signals. This mode allows 16-bit external memory and peripheral devices to be interfaced to the system.
- Normal expanded narrow mode
Ports A and B are configured as a 16-bit address bus and port A is multiplexed with 8-bit data. Port E provides bus control and status signals. This mode allows 8-bit external memory and peripheral devices to be interfaced to the system.
- Normal single-chip mode
There is no external expansion bus in this mode. The processor program is executed from internal memory. Ports A, B, K, and most of E are available as general-purpose I/O.
- Special single-chip mode
This mode is generally used for debugging single-chip operation, boot-strapping, or security related operations. The active background mode is in control of CPU execution and BDM firmware is waiting for additional serial commands through the BKGD pin. There is no external expansion bus after reset in this mode.
- Emulation expanded wide mode
Developers use this mode for emulation systems in which the users target application is normal expanded wide mode.
- Emulation expanded narrow mode
Developers use this mode for emulation systems in which the users target application is normal expanded narrow mode.
- Special test mode
Ports A and B are configured as a 16-bit multiplexed address and data bus and port E provides bus control and status signals. In special test mode, the write protection of many control bits is lifted so that they can be thoroughly tested without needing to go through reset.
- Special peripheral mode
This mode is intended for Freescale Semiconductor factory testing of the system. The CPU is inactive and an external (tester) bus master drives address, data, and bus control signals.

4.2 External Signal Description

In typical implementations, the MEBI sub-block of the core interfaces directly with external system pins. Some pins may not be bonded out in all implementations.

Table 4-1 outlines the pin names and functions and gives a brief description of their operation reset state of these pins and associated pull-ups or pull-downs is dependent on the mode of operation and on the integration of this block at the chip level (chip dependent).

mode. Background debugging should not be used while the MCU is in special peripheral mode as internal bus conflicts between BDM and the external master can cause improper operation of both functions.

4.4.4 Internal Visibility

Internal visibility is available when the MCU is operating in expanded wide modes or emulation narrow mode. It is not available in single-chip, peripheral or normal expanded narrow modes. Internal visibility is enabled by setting the IVIS bit in the MODE register.

If an internal access is made while E, R/\overline{W} , and \overline{LSTRB} are configured as bus control outputs and internal visibility is off (IVIS=0), E will remain low for the cycle, R/\overline{W} will remain high, and address, data and the \overline{LSTRB} pins will remain at their previous state.

When internal visibility is enabled (IVIS=1), certain internal cycles will be blocked from going external. During cycles when the BDM is selected, R/\overline{W} will remain high, data will maintain its previous state, and address and \overline{LSTRB} pins will be updated with the internal value. During CPU no access cycles when the BDM is not driving, R/\overline{W} will remain high, and address, data and the \overline{LSTRB} pins will remain at their previous state.

NOTE

When the system is operating in a secure mode, internal visibility is not available (i.e., IVIS = 1 has no effect). Also, the IPIPE signals will not be visible, regardless of operating mode. IPIPE1–IPIPE0 will display 0es if they are enabled. In addition, the MOD bits in the MODE control register cannot be written.

4.4.5 Low-Power Options

The MEBI does not contain any user-controlled options for reducing power consumption. The operation of the MEBI in low-power modes is discussed in the following subsections.

4.4.5.1 Operation in Run Mode

The MEBI does not contain any options for reducing power in run mode; however, the external addresses are conditioned to reduce power in single-chip modes. Expanded bus modes will increase power consumption.

4.4.5.2 Operation in Wait Mode

The MEBI does not contain any options for reducing power in wait mode.

4.4.5.3 Operation in Stop Mode

The MEBI will cease to function after execution of a CPU STOP instruction.

Table 8-1. ATDCTL2 Field Descriptions (continued)

Field	Description
1 ASCIE	ATD Sequence Complete Interrupt Enable 0 ATD Sequence Complete interrupt requests are disabled. 1 ATD Interrupt will be requested whenever ASCIF = 1 is set.
0 ASCIF	ATD Sequence Complete Interrupt Flag — If ASCIE = 1 the ASCIF flag equals the SCF flag (see Section 8.3.2.7, “ATD Status Register 0 (ATDSTAT0)”), else ASCIF reads zero. Writes have no effect. 0 No ATD interrupt occurred 1 ATD sequence complete interrupt pending

Table 8-2. External Trigger Configurations

ETRIGLE	ETRIGP	External Trigger Sensitivity
0	0	Falling edge
0	1	Rising edge
1	0	Low level
1	1	High level

8.3.2.4 ATD Control Register 3 (ATDCTL3)

This register controls the conversion sequence length, FIFO for results registers and behavior in Freeze Mode. Writes to this register will abort current conversion sequence but will not start a new sequence.

Module Base + 0x0003

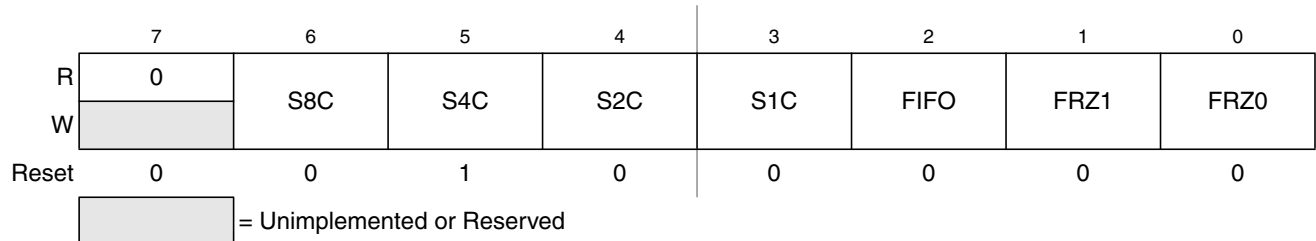


Figure 8-6. ATD Control Register 3 (ATDCTL3)

Read: Anytime

Write: Anytime

Table 8-3. ATDCTL3 Field Descriptions

Field	Description
6–3 S8C, S4C, S2C, S1C	Conversion Sequence Length — These bits control the number of conversions per sequence. Table 8-4 shows all combinations. At reset, S4C is set to 1 (sequence length is 4). This is to maintain software continuity to HC12 Family.

Table 8-8. Clock Prescaler Values

Prescale Value	Total Divisor Value	Maximum Bus Clock ⁽¹⁾	Minimum Bus Clock ⁽²⁾
00000	Divide by 2	4 MHz	1 MHz
00001	Divide by 4	8 MHz	2 MHz
00010	Divide by 6	12 MHz	3 MHz
00011	Divide by 8	16 MHz	4 MHz
00100	Divide by 10	20 MHz	5 MHz
00101	Divide by 12	24 MHz	6 MHz
00110	Divide by 14	28 MHz	7 MHz
00111	Divide by 16	32 MHz	8 MHz
01000	Divide by 18	36 MHz	9 MHz
01001	Divide by 20	40 MHz	10 MHz
01010	Divide by 22	44 MHz	11 MHz
01011	Divide by 24	48 MHz	12 MHz
01100	Divide by 26	52 MHz	13 MHz
01101	Divide by 28	56 MHz	14 MHz
01110	Divide by 30	60 MHz	15 MHz
01111	Divide by 32	64 MHz	16 MHz
10000	Divide by 34	68 MHz	17 MHz
10001	Divide by 36	72 MHz	18 MHz
10010	Divide by 38	76 MHz	19 MHz
10011	Divide by 40	80 MHz	20 MHz
10100	Divide by 42	84 MHz	21 MHz
10101	Divide by 44	88 MHz	22 MHz
10110	Divide by 46	92 MHz	23 MHz
10111	Divide by 48	96 MHz	24 MHz
11000	Divide by 50	100 MHz	25 MHz
11001	Divide by 52	104 MHz	26 MHz
11010	Divide by 54	108 MHz	27 MHz
11011	Divide by 56	112 MHz	28 MHz
11100	Divide by 58	116 MHz	29 MHz
11101	Divide by 60	120 MHz	30 MHz
11110	Divide by 62	124 MHz	31 MHz
11111	Divide by 64	128 MHz	32 MHz

1. Maximum ATD conversion clock frequency is 2 MHz. The maximum allowed bus clock frequency is shown in this column.

2. Minimum ATD conversion clock frequency is 500 kHz. The minimum allowed bus clock frequency is shown in this column.

The PLL filter can be manually or automatically configured into one of two possible operating modes:

- Acquisition mode
In acquisition mode, the filter can make large frequency corrections to the VCO. This mode is used at PLL start-up or when the PLL has suffered a severe noise hit and the VCO frequency is far off the desired frequency. When in acquisition mode, the TRACK status bit is cleared in the CRGFLG register.
- Tracking mode
In tracking mode, the filter makes only small corrections to the frequency of the VCO. PLL jitter is much lower in tracking mode, but the response to noise is also slower. The PLL enters tracking mode when the VCO frequency is nearly correct and the TRACK bit is set in the CRGFLG register.

The PLL can change the bandwidth or operational mode of the loop filter manually or automatically.

In automatic bandwidth control mode (AUTO = 1), the lock detector automatically switches between acquisition and tracking modes. Automatic bandwidth control mode also is used to determine when the PLL clock (PLLCLK) is safe to use as the source for the system and core clocks. If PLL LOCK interrupt requests are enabled, the software can wait for an interrupt request and then check the LOCK bit. If CPU interrupts are disabled, software can poll the LOCK bit continuously (during PLL start-up, usually) or at periodic intervals. In either case, only when the LOCK bit is set, is the PLLCLK clock safe to use as the source for the system and core clocks. If the PLL is selected as the source for the system and core clocks and the LOCK bit is clear, the PLL has suffered a severe noise hit and the software must take appropriate action, depending on the application.

The following conditions apply when the PLL is in automatic bandwidth control mode (AUTO = 1):

- The TRACK bit is a read-only indicator of the mode of the filter.
- The TRACK bit is set when the VCO frequency is within a certain tolerance, Δ_{trk} , and is clear when the VCO frequency is out of a certain tolerance, Δ_{unt} .
- The LOCK bit is a read-only indicator of the locked state of the PLL.
- The LOCK bit is set when the VCO frequency is within a certain tolerance, Δ_{Lock} , and is cleared when the VCO frequency is out of a certain tolerance, Δ_{unl} .
- CPU interrupts can occur if enabled (LOCKIE = 1) when the lock condition changes, toggling the LOCK bit.

The PLL can also operate in manual mode (AUTO = 0). Manual mode is used by systems that do not require an indicator of the lock condition for proper operation. Such systems typically operate well below the maximum system frequency (f_{sys}) and require fast start-up. The following conditions apply when in manual mode:

- ACQ is a writable control bit that controls the mode of the filter. Before turning on the PLL in manual mode, the ACQ bit should be asserted to configure the filter in acquisition mode.
- After turning on the PLL by setting the PLLON bit software must wait a given time (t_{acq}) before entering tracking mode (ACQ = 0).
- After entering tracking mode software must wait a given time (t_{al}) before selecting the PLLCLK as the source for system and core clocks (PLLSEL = 1).

Table 9-11. Outcome of Clock Loss in Wait Mode

CME	SCME	SCMIE	CRG Actions
0	X	X	Clock failure --> No action, clock loss not detected.
1	0	X	Clock failure --> CRG performs Clock Monitor Reset immediately
1	1	0	<p>Clock failure --></p> <p>Scenario 1: OSCCLK recovers prior to exiting Wait Mode.</p> <ul style="list-style-type: none"> – MCU remains in Wait Mode, – VREG enabled, – PLL enabled, – SCM activated, – Start Clock Quality Check, – Set SCMIF interrupt flag. <p><i>Some time later OSCCLK recovers.</i></p> <ul style="list-style-type: none"> – CM no longer indicates a failure, – 4096 OSCCLK cycles later Clock Quality Check indicates clock o.k., – SCM deactivated, – PLL disabled depending on PLLWAI, – VREG remains enabled (<i>never gets disabled in Wait Mode</i>). – MCU remains in Wait Mode. <p><i>Some time later either a wakeup interrupt occurs (no SCM interrupt)</i></p> <ul style="list-style-type: none"> – Exit Wait Mode using OSCCLK as system clock (SYSCLK), – Continue normal operation. <p><i>or an External Reset is applied.</i></p> <ul style="list-style-type: none"> – Exit Wait Mode using OSCCLK as system clock, – Start reset sequence. <p>Scenario 2: OSCCLK does not recover prior to exiting Wait Mode.</p> <ul style="list-style-type: none"> – MCU remains in Wait Mode, – VREG enabled, – PLL enabled, – SCM activated, – Start Clock Quality Check, – Set SCMIF interrupt flag, – Keep performing Clock Quality Checks (could continue infinitely) while in Wait Mode. <p><i>Some time later either a wakeup interrupt occurs (no SCM interrupt)</i></p> <ul style="list-style-type: none"> – Exit Wait Mode in SCM using PLL clock (f_{SCM}) as system clock, – Continue to perform additional Clock Quality Checks until OSCCLK is o.k. again. <p><i>or an External RESET is applied.</i></p> <ul style="list-style-type: none"> – Exit Wait Mode in SCM using PLL clock (f_{SCM}) as system clock, – Start reset sequence, – Continue to perform additional Clock Quality Checks until OSCCLK is o.k.again.

writes (0x0055 or 0x00AA) to the ARMCOP register must occur in the last 25% of the selected time-out period. A premature write the CRG will immediately generate a reset.

As soon as the reset sequence is completed the reset generator checks the reset condition. If no clock monitor failure is indicated and the latched state of the COP timeout is true, processing begins by fetching the COP vector.

9.5.3 Power-On Reset, Low Voltage Reset

The on-chip voltage regulator detects when V_{DD} to the MCU has reached a certain level and asserts power-on reset or low voltage reset or both. As soon as a power-on reset or low voltage reset is triggered the CRG performs a quality check on the incoming clock signal. As soon as clock quality check indicates a valid oscillator clock signal the reset sequence starts using the oscillator clock. If after 50 check windows the clock quality check indicated a non-valid oscillator clock the reset sequence starts using self-clock mode.

Figure 9-26 and Figure 9-27 show the power-up sequence for cases when the $\overline{\text{RESET}}$ pin is tied to V_{DD} and when the $\overline{\text{RESET}}$ pin is held low.

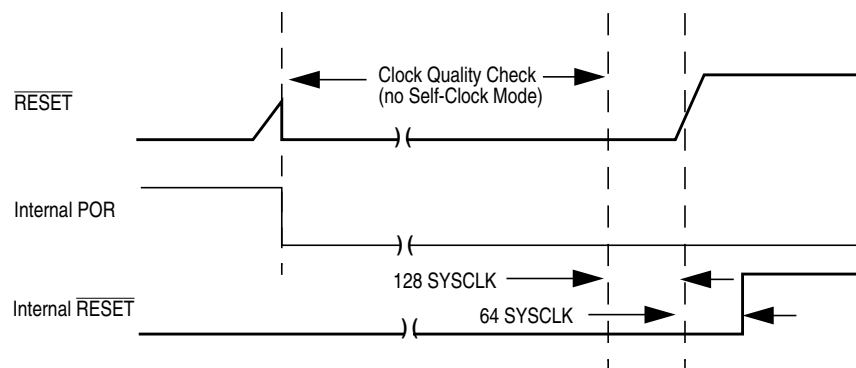


Figure 9-26. $\overline{\text{RESET}}$ Pin Tied to V_{DD} (by a Pull-Up Resistor)

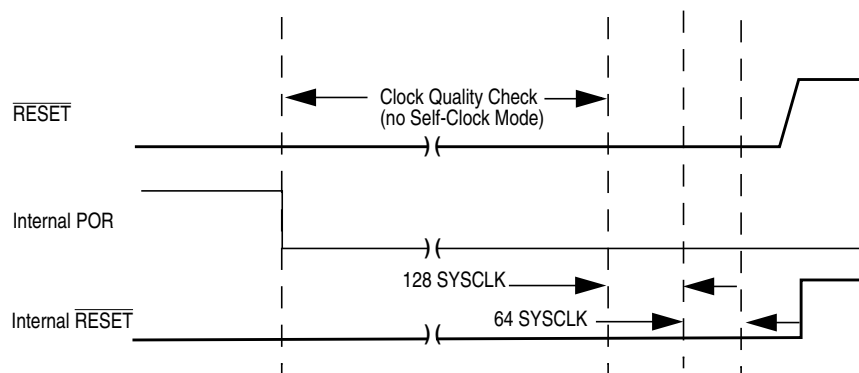


Figure 9-27. $\overline{\text{RESET}}$ Pin Held Low Externally

9.6 Interrupts

The interrupts/reset vectors requested by the CRG are listed in [Table 9-15](#). Refer to the device overview chapter for related vector addresses and priorities.

Table 9-15. CRG Interrupt Vectors

Interrupt Source	CCR Mask	Local Enable
Real-time interrupt	I bit	CRGINT (RTIE)
LOCK interrupt	I bit	CRGINT (LOCKIE)
SCM interrupt	I bit	CRGINT (SCMIE)

9.6.1 Real-Time Interrupt

The CRGV4 generates a real-time interrupt when the selected interrupt time period elapses. RTI interrupts are locally disabled by setting the RTIE bit to 0. The real-time interrupt flag (RTIF) is set to 1 when a timeout occurs, and is cleared to 0 by writing a 1 to the RTIF bit.

The RTI continues to run during pseudo-stop mode if the PRE bit is set to 1. This feature can be used for periodic wakeup from pseudo-stop if the RTI interrupt is enabled.

9.6.2 PLL Lock Interrupt

The CRGV4 generates a PLL lock interrupt when the LOCK condition of the PLL has changed, either from a locked state to an unlocked state or vice versa. Lock interrupts are locally disabled by setting the LOCKIE bit to 0. The PLL Lock interrupt flag (LOCKIF) is set to 1 when the LOCK condition has changed, and is cleared to 0 by writing a 1 to the LOCKIF bit.

9.6.3 Self-Clock Mode Interrupt

The CRGV4 generates a self-clock mode interrupt when the SCM condition of the system has changed, either entered or exited self-clock mode. SCM conditions can only change if the self-clock mode enable bit (SCME) is set to 1. SCM conditions are caused by a failing clock quality check after power-on reset (POR) or low voltage reset (LVR) or recovery from full stop mode (PSTP = 0) or clock monitor failure. For details on the clock quality check refer to [Section 9.4.4, “Clock Quality Checker.”](#) If the clock monitor is enabled (CME = 1) a loss of external clock will also cause a SCM condition (SCME = 1).

SCM interrupts are locally disabled by setting the SCMIE bit to 0. The SCM interrupt flag (SCMIF) is set to 1 when the SCM condition has changed, and is cleared to 0 by writing a 1 to the SCMIF bit.

13.3 Memory Map and Registers

This section provides a detailed description of all memory and registers.

13.3.1 Module Memory Map

The memory map for the SCI module is given below in [Figure 13-2](#). The Address listed for each register is the address offset. The total address for each register is the sum of the base address for the SCI module and the address offset for each register.

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000	SCIBDH	R	0	0	0	SBR12	SBR11	SBR10	SBR9	SBR8
		W								
0x0001	SCIBDL	R	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
		W								
0x0002	SCICR1	R	LOOPS	SCISWAI	RSRC	M	WAKE	ILT	PE	PT
		W								
0x0003	SCICR2	R	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
		W								
0x0004	SCISR1	R	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
		W								
0x0005	SCISR2	R	0	0	0	0	0	BRK13	TXDIR	RAF
		W								
0x0006	SCIDRH	R	R8	T8	0	0	0	0	0	0
		W								
0x0007	SCIDRL	R	R7	R6	R5	R4	R3	R2	R1	R0
		W	T7	T6	T5	T4	T3	T2	T1	T0


 = Unimplemented or Reserved

Figure 13-2. SCI Register Summary

13.3.2 Register Descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Writes to a reserved register location do not have any effect and reads of these locations return a zero. Details of register bit and field function follow the register diagrams, in bit order.

Table 13-5. SCISR1 Field Descriptions (continued)

Field	Description
5 RDRF	<p>Receive Data Register Full Flag — RDRF is set when the data in the receive shift register transfers to the SCI data register. Clear RDRF by reading SCI status register 1 (SCISR1) with RDRF set and then reading SCI data register low (SCIDRL).</p> <p>0 Data not available in SCI data register 1 Received data available in SCI data register</p>
4 IDLE	<p>Idle Line Flag — IDLE is set when 10 consecutive logic 1s (if M=0) or 11 consecutive logic 1s (if M=1) appear on the receiver input. Once the IDLE flag is cleared, a valid frame must again set the RDRF flag before an idle condition can set the IDLE flag. Clear IDLE by reading SCI status register 1 (SCISR1) with IDLE set and then reading SCI data register low (SCIDRL).</p> <p>0 Receiver input is either active now or has never become active since the IDLE flag was last cleared 1 Receiver input has become idle</p> <p>Note: When the receiver wakeup bit (RWU) is set, an idle line condition does not set the IDLE flag.</p>
3 OR	<p>Overrun Flag — OR is set when software fails to read the SCI data register before the receive shift register receives the next frame. The OR bit is set immediately after the stop bit has been completely received for the second frame. The data in the shift register is lost, but the data already in the SCI data registers is not affected. Clear OR by reading SCI status register 1 (SCISR1) with OR set and then reading SCI data register low (SCIDRL).</p> <p>0 No overrun 1 Overrun</p> <p>Note: OR flag may read back as set when RDRF flag is clear. This may happen if the following sequence of events occurs:</p> <ol style="list-style-type: none"> 1. After the first frame is received, read status register SCISR1 (returns RDRF set and OR flag clear); 2. Receive second frame without reading the first frame in the data register (the second frame is not received and OR flag is set); 3. Read data register SCIDRL (returns first frame and clears RDRF flag in the status register); 4. Read status register SCISR1 (returns RDRF clear and OR set). <p>Event 3 may be at exactly the same time as event 2 or any time after. When this happens, a dummy SCIDRL read following event 4 will be required to clear the OR flag if further frames are to be received.</p>
2 NF	<p>Noise Flag — NF is set when the SCI detects noise on the receiver input. NF bit is set during the same cycle as the RDRF flag but does not get set in the case of an overrun. Clear NF by reading SCI status register 1 (SCISR1), and then reading SCI data register low (SCIDRL).</p> <p>0 No noise 1 Noise</p>
1 FE	<p>Framing Error Flag — FE is set when a logic 0 is accepted as the stop bit. FE bit is set during the same cycle as the RDRF flag but does not get set in the case of an overrun. FE inhibits further data reception until it is cleared. Clear FE by reading SCI status register 1 (SCISR1) with FE set and then reading the SCI data register low (SCIDRL).</p> <p>0 No framing error 1 Framing error</p>
0 PF	<p>Parity Error Flag — PF is set when the parity enable bit (PE) is set and the parity of the received data does not match the parity type bit (PT). PF bit is set during the same cycle as the RDRF flag but does not get set in the case of an overrun. Clear PF by reading SCI status register 1 (SCISR1), and then reading SCI data register low (SCIDRL).</p> <p>0 No parity error 1 Parity error</p>

13.4.4.4 Framing Errors

If the data recovery logic does not detect a logic 1 where the stop bit should be in an incoming frame, it sets the framing error flag, FE, in SCI status register 1 (SCISR1). A break character also sets the FE flag because a break character has no stop bit. The FE flag is set at the same time that the RDRF flag is set.

13.4.4.5 Baud Rate Tolerance

A transmitting device may be operating at a baud rate below or above the receiver baud rate. Accumulated bit time misalignment can cause one of the three stop bit data samples (RT8, RT9, and RT10) to fall outside the actual stop bit. A noise error will occur if the RT8, RT9, and RT10 samples are not all the same logical values. A framing error will occur if the receiver clock is misaligned in such a way that the majority of the RT8, RT9, and RT10 stop bit samples are a logic zero.

As the receiver samples an incoming frame, it re-synchronizes the RT clock on any valid falling edge within the frame. Re synchronization within frames will correct a misalignment between transmitter bit times and receiver bit times.

13.4.4.5.1 Slow Data Tolerance

Figure 13-20 shows how much a slow received frame can be misaligned without causing a noise error or a framing error. The slow stop bit begins at RT8 instead of RT1 but arrives in time for the stop bit data samples at RT8, RT9, and RT10.

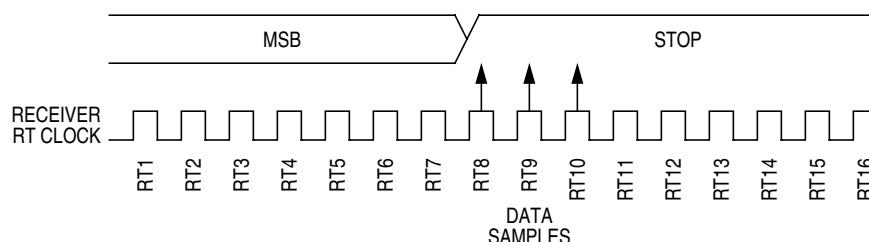


Figure 13-20. Slow Data

Let's take RTr as receiver RT clock and RTt as transmitter RT clock.

For an 8-bit data character, it takes the receiver 9 bit times x 16 RTr cycles + 7 RTr cycles = 151 RTr cycles to start data sampling of the stop bit.

With the misaligned character shown in Figure 13-20, the receiver counts 151 RTr cycles at the point when the count of the transmitting device is 9 bit times x 16 RTt cycles = 144 RTt cycles.

The maximum percent difference between the receiver count and the transmitter count of a slow 8-bit data character with no errors is:

$$((151 - 144) / 151) \times 100 = 4.63\%$$

For a 9-bit data character, it takes the receiver 10 bit times x 16 RTr cycles + 7 RTr cycles = 167 RTr cycles to start data sampling of the stop bit.

18.4.2 Operating Modes

18.4.2.1 Wait Mode

If the MCU enters wait mode while a Flash command is active ($CCIF = 0$), that command and any buffered command will be completed.

The Flash module can recover the MCU from wait mode if the interrupts are enabled (see [Section 18.4.5](#)).

18.4.2.2 Stop Mode

If the MCU enters stop mode while a Flash command is active ($CCIF = 0$), that command will be aborted and the data being programmed or erased is lost. The high voltage circuitry to the Flash array will be switched off when entering stop mode. $CCIF$ and $ACCERR$ flags will be set. Upon exit from stop mode, the $CBEIF$ flag will be set and any buffered command will not be executed. The $ACCERR$ flag must be cleared before returning to normal operation.

NOTE

As active Flash commands are immediately aborted when the MCU enters stop mode, it is strongly recommended that the user does not use the STOP instruction during program and erase execution.

18.4.2.3 Background Debug Mode

In background debug mode (BDM), the $FPROT$ register is writable. If the MCU is unsecured, then all Flash commands listed in [Table 18-16](#) can be executed. If the MCU is secured and is in special single chip mode, the only possible command to execute is mass erase.

18.4.3 Flash Module Security

The Flash module provides the necessary security information to the MCU. After each reset, the Flash module determines the security state of the MCU as defined in [Section 18.3.2.2](#), “Flash Security Register (FSEC)”.

The contents of the Flash security/options byte at address $0xFF0F$ in the Flash configuration field must be changed directly by programming address $0xFF0F$ when the device is unsecured and the higher address sector is unprotected. If the Flash security/options byte is left in the secure state, any reset will cause the MCU to return to the secure operating mode.

18.4.3.1 Unsecuring the MCU using Backdoor Key Access

The MCU may only be unsecured by using the backdoor key access feature which requires knowledge of the contents of the backdoor key (four 16-bit words programmed at addresses $0xFF00$ – $0xFF07$). If $KEYEN[1:0] = 1:0$ and the $KEYACC$ bit is set, a write to a backdoor key address in the Flash array triggers a comparison between the written data and the backdoor key data stored in the Flash array. If all four words of data are written to the correct addresses in the correct order and the data matches the backdoor key stored in the Flash array, the MCU will be unsecured. The data must be written to the backdoor key

19.4.1.3.4 Mass Erase Command

The mass erase operation will erase all addresses in a Flash array using an embedded algorithm.

An example flow to execute the mass erase operation is shown in [Figure 19-28](#). The mass erase command write sequence is as follows:

1. Write to a Flash array address to start the command write sequence for the mass erase command. The address and data written will be ignored.
2. Write the mass erase command, 0x41, to the FCMD register.
3. Clear the CBEIF flag in the FSTAT register by writing a 1 to CBEIF to launch the mass erase command.

If a Flash array to be erased contains any protected area, the PVIOL flag in the FSTAT register will set and the mass erase command will not launch. Once the mass erase command has successfully launched, the CCIF flag in the FSTAT register will set after the mass erase operation has completed unless a new command write sequence has been buffered.

addresses sequentially starting with 0xFF00–0xFF01 and ending with 0xFF06–0xFF07. The values 0x0000 and 0xFFFF are not permitted as keys. When the KEYACC bit is set, reads of the Flash array will return invalid data.

The user code stored in the Flash array must have a method of receiving the backdoor key from an external stimulus. This external stimulus would typically be through one of the on-chip serial ports.

If KEYEN[1:0] = 1:0 in the FSEC register, the MCU can be unsecured by the backdoor key access sequence described below:

1. Set the KEYACC bit in the FCNFG register
2. Write the correct four 16-bit words to Flash addresses 0xFF00–0xFF07 sequentially starting with 0xFF00
3. Clear the KEYACC bit in the FCNFG register
4. If all four 16-bit words match the backdoor key stored in Flash addresses 0xFF00–0xFF07, the MCU is unsecured and bits SEC[1:0] in the FSEC register are forced to the unsecure state of 1:0

The backdoor key access sequence is monitored by the internal security state machine. An illegal operation during the backdoor key access sequence will cause the security state machine to lock, leaving the MCU in the secured state. A reset of the MCU will cause the security state machine to exit the lock state and allow a new backdoor key access sequence to be attempted. The following illegal operations will lock the security state machine:

1. If any of the four 16-bit words does not match the backdoor key programmed in the Flash array
2. If the four 16-bit words are written in the wrong sequence
3. If more than four 16-bit words are written
4. If any of the four 16-bit words written are 0x0000 or 0xFFFF
5. If the KEYACC bit does not remain set while the four 16-bit words are written

After the backdoor key access sequence has been correctly matched, the MCU will be unsecured. The Flash security byte can be programmed to the unsecure state, if desired.

In the unsecure state, the user has full control of the contents of the four word backdoor key by programming bytes 0xFF00–0xFF07 of the Flash configuration field.

The security as defined in the Flash security/options byte at address 0xFF0F is not changed by using the backdoor key access sequence to unsecure. The backdoor key stored in addresses 0xFF00–0xFF07 is unaffected by the backdoor key access sequence. After the next reset sequence, the security state of the Flash module is determined by the Flash security/options byte at address 0xFF0F. The backdoor key access sequence has no effect on the program and erase protection defined in the FPROT register.

It is not possible to unsecure the MCU in special single chip mode by executing the backdoor key access sequence in background debug mode.

21.3.2 Register Descriptions

The Flash module contains a set of 16 control and status registers located between module base + 0x0000 and 0x000F. A summary of the Flash module registers is given in Figure 21-3. Detailed descriptions of each register bit are provided.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000	R	FDIVLD	PRDIV8	FDIV5	FDIV4	FDIV3	FDIV2	FDIV1	FDIV0
FCLKDIV	W								
0x0001	R	KEYEN1	KEYEN0	NV5	NV4	NV3	NV2	SEC1	SEC0
FSEC	W								
0x0002	R	0	0	0	0	0	0	0	0
RESERVED1 ⁽¹⁾	W								
0x0003	R	CBEIE	CCIE	KEYACC	0	0	0	0	0
FCNFG	W								
0x0004	R	FPOPEN	NV6	FPHDIS	FPHS1	FPHS0	FPLDIS	FPLS1	FPLS0
FPROT	W								
0x0005	R	CBEIF	CCIF	PVIOL	ACCERR	0	BLANK	FAIL	DONE
FSTAT	W								
0x0006	R	0	CMDB6	CMDB5	0	0	CMDB2	0	CMDB0
FCMD	W								
0x0007	R	0	0	0	0	0	0	0	0
RESERVED2 ¹	W								
0x0008	R	FABHI							
FADDRHI ¹	W								
0x0009	R	FABLO							
FADDRLO ¹	W								
0x000A	R	FDHI							
FDATAHI ¹	W								
0x000B	R	FDLO							
FDATALO ¹	W								
0x000C	R	0	0	0	0	0	0	0	0
RESERVED3 ¹	W								
0x000D	R	0	0	0	0	0	0	0	0
RESERVED4 ¹	W								
0x000E	R	0	0	0	0	0	0	0	0
RESERVED5 ¹	W								
0x000F	R	0	0	0	0	0	0	0	0
RESERVED6 ¹	W								

= Unimplemented or Reserved

Figure 21-3. Flash Register Summary

1. Intended for factory test purposes only.

Table 21-13. FSTAT Field Descriptions

Field	Description
5 PVIOL	Protection Violation — The PVIOL flag indicates an attempt was made to program or erase an address in a protected Flash array memory area. The PVIOL flag is cleared by writing a 1 to PVIOL. Writing a 0 to the PVIOL flag has no effect on PVIOL. While PVIOL is set, it is not possible to launch another command. 0 No protection violation detected 1 Protection violation has occurred
4 ACCERR	Access Error — The ACCERR flag indicates an illegal access to the Flash array caused by either a violation of the command write sequence, issuing an illegal command (illegal combination of the CMDBx bits in the FCMD register) or the execution of a CPU STOP instruction while a command is executing (CCIF=0). The ACCERR flag is cleared by writing a 1 to ACCERR. Writing a 0 to the ACCERR flag has no effect on ACCERR. While ACCERR is set, it is not possible to launch another command. 0 No access error detected 1 Access error has occurred
2 BLANK	Flash Array Has Been Verified as Erased — The BLANK flag indicates that an erase verify command has checked the Flash array and found it to be erased. The BLANK flag is cleared by hardware when CBEIF is cleared as part of a new valid command write sequence. Writing to the BLANK flag has no effect on BLANK. 0 If an erase verify command has been requested, and the CCIF flag is set, then a 0 in BLANK indicates the array is not erased 1 Flash array verifies as erased
1 FAIL	Flag Indicating a Failed Flash Operation — In special modes, the FAIL flag will set if the erase verify operation fails (Flash array verified as not erased). Writing a 0 to the FAIL flag has no effect on FAIL. The FAIL flag is cleared by writing a 1 to FAIL. While FAIL is set, it is not possible to launch another command. 0 Flash operation completed without error 1 Flash operation failed
0 DONE	Flag Indicating a Failed Operation is not Active — In special modes, the DONE flag will clear if a program, erase, or erase verify operation is active. 0 Flash operation is active 1 Flash operation is not active

21.3.2.7 Flash Command Register (FCMD)

The FCMD register defines the Flash commands.

Module Base + 0x0006

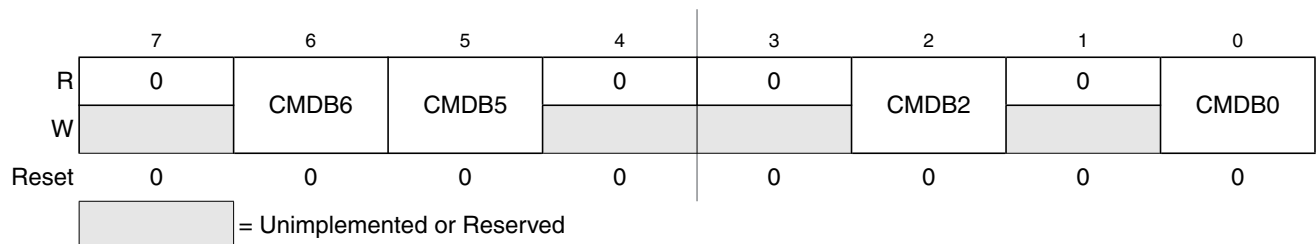


Figure 21-11. Flash Command Register (FCMD)

Bits CMDB6, CMDB5, CMDB2, and CMDB0 are readable and writable during a command write sequence while bits 7, 4, 3, and 1 read 0 and are not writable.