

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	HCS12
Core Size	16-Bit
Speed	25MHz
Connectivity	CANbus, 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 ~ 85°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/mc9s12c32cfae25



Appendix E Ordering Information686

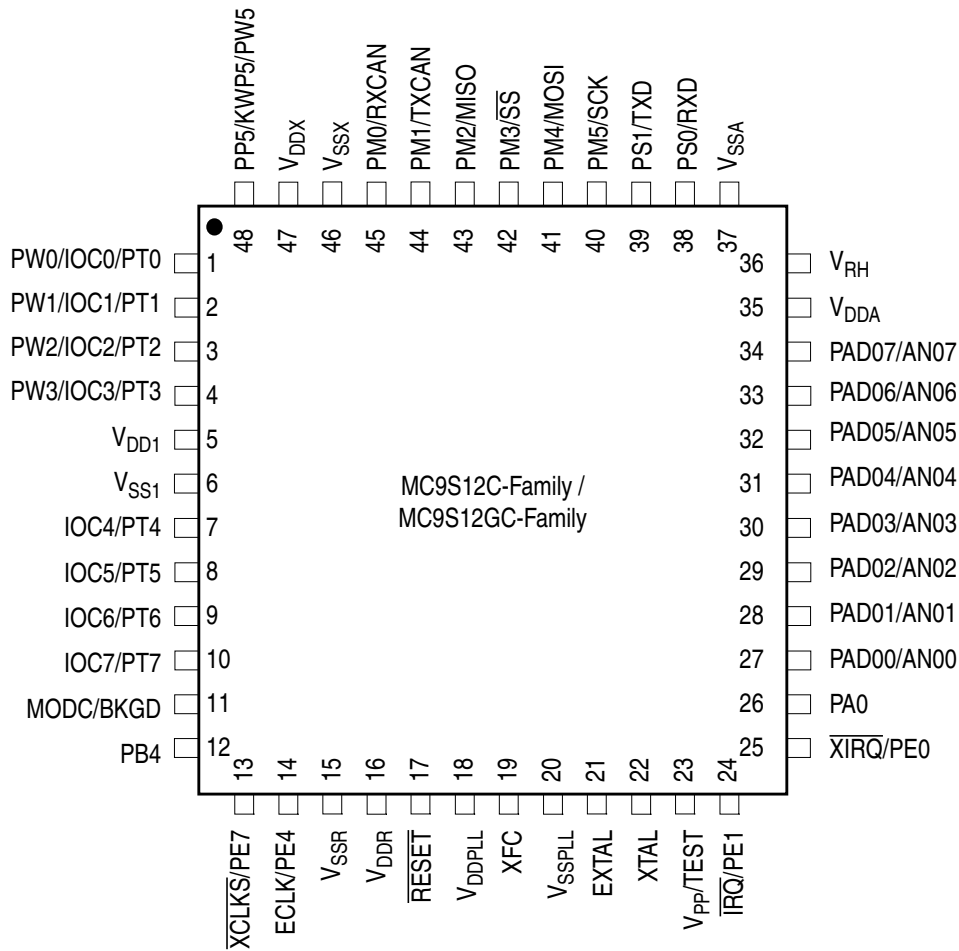


Figure 1-9. Pin Assignments in 48-Pin LQFP

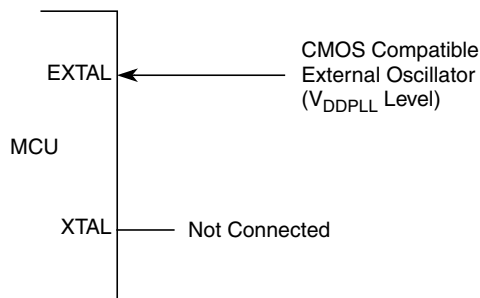


Figure 1-13. External Clock Connections (PE7 = 0)

1.3.4.9 PE6 / MODB / IPIPE1 — Port E I/O Pin 6

PE6 is a general purpose input or output pin. It is used as a MCU operating mode select pin during reset. The state of this pin is latched to the MODB bit at the rising edge of $\overline{\text{RESET}}$. This pin is shared with the instruction queue tracking signal IPIPE1. This pin is an input with a pull-down device which is only active when $\overline{\text{RESET}}$ is low. PE[6] is not available in the 48- / 52-pin package versions.

1.3.4.10 PE5 / MODA / IPIPE0 — Port E I/O Pin 5

PE5 is a general purpose input or output pin. It is used as a MCU operating mode select pin during reset. The state of this pin is latched to the MODA bit at the rising edge of $\overline{\text{RESET}}$. This pin is shared with the instruction queue tracking signal IPIPE0. This pin is an input with a pull-down device which is only active when $\overline{\text{RESET}}$ is low. This pin is not available in the 48- / 52-pin package versions.

1.3.4.11 PE4 / ECLK— Port E I/O Pin [4] / E-Clock Output

ECLK is the output connection for the internal bus clock. It is used to demultiplex the address and data in expanded modes and is used as a timing reference. ECLK frequency is equal to 1/2 the crystal frequency out of reset. The ECLK pin is initially configured as ECLK output with stretch in all expanded modes. The E clock output function depends upon the settings of the NECLK bit in the PEAR register, the IVIS bit in the MODE register and the ESTR bit in the EBICTL register. All clocks, including the E clock, are halted when the MCU is in stop mode. It is possible to configure the MCU to interface to slow external memory. ECLK can be stretched for such accesses. Reference the MISC register (EXSTR[1:0] bits) for more information. In normal expanded narrow mode, the E clock is available for use in external select decode logic or as a constant speed clock for use in the external application system. Alternatively PE4 can be used as a general purpose input or output pin.

1.3.4.12 PE3 / $\overline{\text{LSTRB}}$ — Port E I/O Pin [3] / Low-Byte Strobe ($\overline{\text{LSTRB}}$)

In all modes this pin can be used as a general-purpose I/O and is an input with an active pull-up out of reset. If the strobe function is required, it should be enabled by setting the LSTRE bit in the PEAR register. This signal is used in write operations. Therefore external low byte writes will not be possible until this function is enabled. This pin is also used as $\overline{\text{TAGLO}}$ in special expanded modes and is multiplexed with the $\overline{\text{LSTRB}}$ function. This pin is not available in the 48- / 52-pin package versions.

The security byte resides in a portion of the Flash array.

Check the Flash Block User Guide for more details on the security configuration.

1.5.2.2 Operation of the Secured Microcontroller

1.5.2.2.1 Normal Single Chip Mode

This will be the most common usage of the secured part. Everything will appear the same as if the part was not secured with the exception of BDM operation. The BDM operation will be blocked.

1.5.2.2.2 Executing from External Memory

The user may wish to execute from external space with a secured microcontroller. This is accomplished by resetting directly into expanded mode. The internal FLASH will be disabled. BDM operations will be blocked.

1.5.2.3 Unsecuring the Microcontroller

In order to unsecure the microcontroller, the internal FLASH must be erased. This can be done through an external program in expanded mode or via a sequence of BDM commands. Unsecuring is also possible via the Backdoor Key Access. Refer to Flash Block Guide for details.

Once the user has erased the FLASH, the part can be reset into special single chip mode. This invokes a program that verifies the erasure of the internal FLASH. Once this program completes, the user can erase and program the FLASH security bits to the unsecured state. This is generally done through the BDM, but the user could also change to expanded mode (by writing the mode bits through the BDM) and jumping to an external program (again through BDM commands). Note that if the part goes through a reset before the security bits are reprogrammed to the unsecure state, the part will be secured again.

1.5.3 Low-Power Modes

The microcontroller features three main low power modes. Consult the respective Block User Guide for information on the module behavior in stop, pseudo stop, and wait mode. An important source of information about the clock system is the Clock and Reset Generator User Guide (CRG).

1.5.3.1 Stop

Executing the CPU STOP instruction stops all clocks and the oscillator thus putting the chip in fully static mode. Wake up from this mode can be done via reset or external interrupts.

1.5.3.2 Pseudo Stop

This mode is entered by executing the CPU STOP instruction. In this mode the oscillator is still running and the real time interrupt (RTI) or watchdog (COP) sub module can stay active. Other peripherals are turned off. This mode consumes more current than the full stop mode, but the wake up time from this mode is significantly shorter.

2.3.2.3 Port M Registers

2.3.2.3.1 Port M I/O Register (PTM)

Module Base + 0x0010

	7	6	5	4	3	2	1	0
R	0	0	PTM5	PTM4	PTM3	PTM2	PTM1	PTM0
W								
MSCAN/ SPI	—	—	SCK	MOSI	\overline{SS}	MISO	TXCAN	RXCAN
Reset	0	0	0	0	0	0	0	0


 = Unimplemented or Reserved

Figure 2-17. Port M I/O Register (PTM)

Read: Anytime.

Write: Anytime.

If the data direction bits of the associated I/O pins are set to 1, a read returns the value of the port register, otherwise the value at the pins is read.

The SPI pin configurations (PM[5:2]) is determined by several status bits in the SPI module. *Please refer to the SPI Block User Guide for details.*

2.3.2.3.2 Port M Input Register (PTIM)

Module Base + 0x0011

	7	6	5	4	3	2	1	0
R	0	0	PTIM5	PTIM4	PTIM3	PTIM2	PTIM1	PTIM0
W								
Reset	—	—	—	—	—	—	—	—


 = Unimplemented or Reserved

Figure 2-18. Port M Input Register (PTIM)

Read: Anytime.

Write: Never, writes to this register have no effect.

Table 2-16. PTIM Field Descriptions

Field	Description
5–0 PTIM[5:0]	Port M Input Register — This register always reads back the status of the associated pins. This also can be used to detect overload or short circuit conditions on output pins.

3.1.1 Features

- Registers for mapping of address space for on-chip RAM, EEPROM, and FLASH (or ROM) memory blocks and associated registers
- Memory mapping control and selection based upon address decode and system operating mode
- Core address bus control
- Core data bus control and multiplexing
- Core security state decoding
- Emulation chip select signal generation ($\overline{\text{ECS}}$)
- External chip select signal generation ($\overline{\text{XCS}}$)
- Internal memory expansion
- External stretch and ROM mapping control functions via the MISC register
- Reserved registers for test purposes
- Configurable system memory options defined at integration of core into the system-on-a-chip (SoC).

3.1.2 Modes of Operation

Some of the registers operate differently depending on the mode of operation (i.e., normal expanded wide, special single chip, etc.). This is best understood from the register descriptions.

3.2 External Signal Description

All interfacing with the MMC sub-block is done within the core, it has no external signals.

3.3 Memory Map and Register Definition

A summary of the registers associated with the MMC sub-block is shown in [Figure 3-2](#). Detailed descriptions of the registers and bits are given in the subsections that follow.

3.3.1 Module Memory Map

Table 3-1. MMC Memory Map

Address Offset	Register	Access
0x0010	Initialization of Internal RAM Position Register (INITRM)	R/W
0x0011	Initialization of Internal Registers Position Register (INITRG)	R/W
0x0012	Initialization of Internal EEPROM Position Register (INITEE)	R/W
0x0013	Miscellaneous System Control Register (MISC)	R/W
0x0014	Reserved	—
⋮	⋮	—

During the execution of an RTC instruction, the CPU:

- Pulls the old PPAGE value from the stack
- Pulls the 16-bit return address from the stack and loads it into the PC
- Writes the old PPAGE value into the PPAGE register
- Refills the queue and resumes execution at the return address

This sequence is uninterruptable; an RTC can be executed from anywhere in memory, even from a different page of extended memory in the expansion window.

The CALL and RTC instructions behave like JSR and RTS, except they use more execution cycles. Therefore, routinely substituting CALL/RTC for JSR/RTS is not recommended. JSR and RTS can be used to access subroutines that are on the same page in expanded memory. However, a subroutine in expanded memory that can be called from other pages must be terminated with an RTC. And the RTC unstacks a PPAGE value. So any access to the subroutine, even from the same page, must use a CALL instruction so that the correct PPAGE value is in the stack.

3.4.3.2 Extended Address (XAB19:14) and $\overline{\text{ECS}}$ Signal Functionality

If the EMK bit in the MODE register is set (see MEBI block description chapter) the PIX5:0 values will be output on XAB19:14 respectively (port K bits 5:0) when the system is addressing within the physical program page window address space (0x8000–0xBFFF) and is in an expanded mode. When addressing anywhere else within the physical address space (outside of the paging space), the XAB19:14 signals will be assigned a constant value based upon the physical address space selected. In addition, the active-low emulation chip select signal, $\overline{\text{ECS}}$, will likewise function based upon the assigned memory allocation. In the cases of 48K byte and 64K byte allocated physical FLASH/ROM space, the operation of the $\overline{\text{ECS}}$ signal will additionally depend upon the state of the ROMHM bit (see [Section 3.3.2.4, “Miscellaneous System Control Register \(MISC\)”](#)) in the MISC register. [Table 3-18](#), [Table 3-19](#), [Table 3-20](#), and [Table 3-21](#) summarize the functionality of these signals based upon the allocated memory configuration. Again, this signal information is only available externally when the EMK bit is set and the system is in an expanded mode.

Table 3-18. 0K Byte Physical FLASH/ROM Allocated

Address Space	Page Window Access	ROMHM	ECS	XAB19:14
0x0000–0x3FFF	N/A	N/A	1	0x3D
0x4000–0x7FFF	N/A	N/A	1	0x3E
0x8000–0xBFFF	N/A	N/A	0	PIX[5:0]
0xC000–0xFFFF	N/A	N/A	0	0x3F

Table 3-19. 16K Byte Physical FLASH/ROM Allocated

Address Space	Page Window Access	ROMHM	ECS	XAB19:14
0x0000–0x3FFF	N/A	N/A	1	0x3D
0x4000–0x7FFF	N/A	N/A	1	0x3E
0x8000–0xBFFF	N/A	N/A	1	PIX[5:0]
0xC000–0xFFFF	N/A	N/A	0	0x3F

Table 4-10. RDRIV Field Descriptions

Field	Description
7 RDRK	Reduced Drive of Port K 0 All port K output pins have full drive enabled. 1 All port K output pins have reduced drive enabled.
4 RDPE	Reduced Drive of Port E 0 All port E output pins have full drive enabled. 1 All port E output pins have reduced drive enabled.
1 RDPB	Reduced Drive of Port B 0 All port B output pins have full drive enabled. 1 All port B output pins have reduced drive enabled.
0 RDPA	Reduced Drive of Ports A 0 All port A output pins have full drive enabled. 1 All port A output pins have reduced drive enabled.

4.3.2.12 External Bus Interface Control Register (EBICTL)

Module Base + 0x000E

Starting address location affected by INITRG register setting.

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	ESTR
W								
Reset:								
Peripheral	0	0	0	0	0	0	0	0
All other modes	0	0	0	0	0	0	0	1

= Unimplemented or Reserved

Figure 4-16. External Bus Interface Control Register (EBICTL)

Read: Anytime (provided this register is in the map)

Write: Refer to individual bit descriptions below

The EBICTL register is used to control miscellaneous functions (i.e., stretching of external E clock).

This register is not in the on-chip memory map in expanded and special peripheral modes. Therefore, these accesses will be echoed externally.

Table 4-11. EBICTL Field Descriptions

Field	Description
0 ESTR	E Clock Stretches — This control bit determines whether the E clock behaves as a simple free-running clock or as a bus control signal that is active only for external bus cycles. Normal and Emulation: write once Special: write anytime 0 E never stretches (always free running). 1 E stretches high during stretched external accesses and remains low during non-visible internal accesses. This bit has no effect in single-chip modes.

5.4.1 Low-Power Modes

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

5.4.1.1 Operation in Run Mode

The INT does not contain any options for reducing power in run mode.

5.4.1.2 Operation in Wait Mode

Clocks to the INT can be shut off during system wait mode and the asynchronous interrupt path will be used to generate the wake-up signal upon recognition of a valid interrupt or any $\overline{\text{XIRQ}}$ request.

5.4.1.3 Operation in Stop Mode

Clocks to the INT can be shut off during system stop mode and the asynchronous interrupt path will be used to generate the wake-up signal upon recognition of a valid interrupt or any $\overline{\text{XIRQ}}$ request.

5.5 Resets

The INT supports three system reset exception request types: normal system reset or power-on-reset request, crystal monitor reset request, and COP watchdog reset request. The type of reset exception request must be decoded by the system and the proper request made to the core. The INT will then provide the service routine address for the type of reset requested.

5.6 Interrupts

As shown in the block diagram in [Figure 5-1](#), the INT contains a register block to provide interrupt status and control, an optional highest priority I interrupt (HPRIO) block, and a priority decoder to evaluate whether pending interrupts are valid and assess their priority.

5.6.1 Interrupt Registers

The INT registers are accessible only in special modes of operation and function as described in [Section 5.3.2.1, “Interrupt Test Control Register,”](#) and [Section 5.3.2.2, “Interrupt Test Registers,”](#) previously.

5.6.2 Highest Priority I-Bit Maskable Interrupt

When the optional HPRIO block is implemented, the user is allowed to promote a single I-bit maskable interrupt to be the highest priority I interrupt. The HPRIO evaluates all interrupt exception requests and passes the HPRIO vector to the priority decoder if the highest priority I interrupt is active. RTI replaces the promoted interrupt source.

Table 9-2. CRGFLG Field Descriptions (continued)

Field	Description
1 SCMIF	Self-Clock Mode Interrupt Flag — SCMIF is set to 1 when SCM status bit changes. This flag can only be cleared by writing a 1. Writing a 0 has no effect. If enabled (SCMIE=1), SCMIF causes an interrupt request. 0 No change in SCM bit. 1 SCM bit has changed.
0 SCM	Self-Clock Mode Status Bit — SCM reflects the current clocking mode. Writes have no effect. 0 MCU is operating normally with OSCCLK available. 1 MCU is operating in self-clock mode with OSCCLK in an unknown state. All clocks are derived from PLLCLK running at its minimum frequency f_{SCM} .

9.3.2.5 CRG Interrupt Enable Register (CRGINT)

This register enables CRG interrupt requests.

Module Base + 0x0004

	7	6	5	4	3	2	1	0
R	RTIE	0	0	LOCKIE	0	0	SCMIE	0
W								
Reset	0	0	0	0	0	0	0	0
	= Unimplemented or Reserved							

Figure 9-8. CRG Interrupt Enable Register (CRGINT)

Read: anytime

Write: anytime

Table 9-3. CRGINT Field Descriptions

Field	Description
7 RTIE	Real-Time Interrupt Enable Bit 0 Interrupt requests from RTI are disabled. 1 Interrupt will be requested whenever RTIF is set.
4 LOCKIE	Lock Interrupt Enable Bit 0 LOCK interrupt requests are disabled. 1 Interrupt will be requested whenever LOCKIF is set.
1 SCMIE	Self-Clock Mode Interrupt Enable Bit 0 SCM interrupt requests are disabled. 1 Interrupt will be requested whenever SCMIF is set.

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

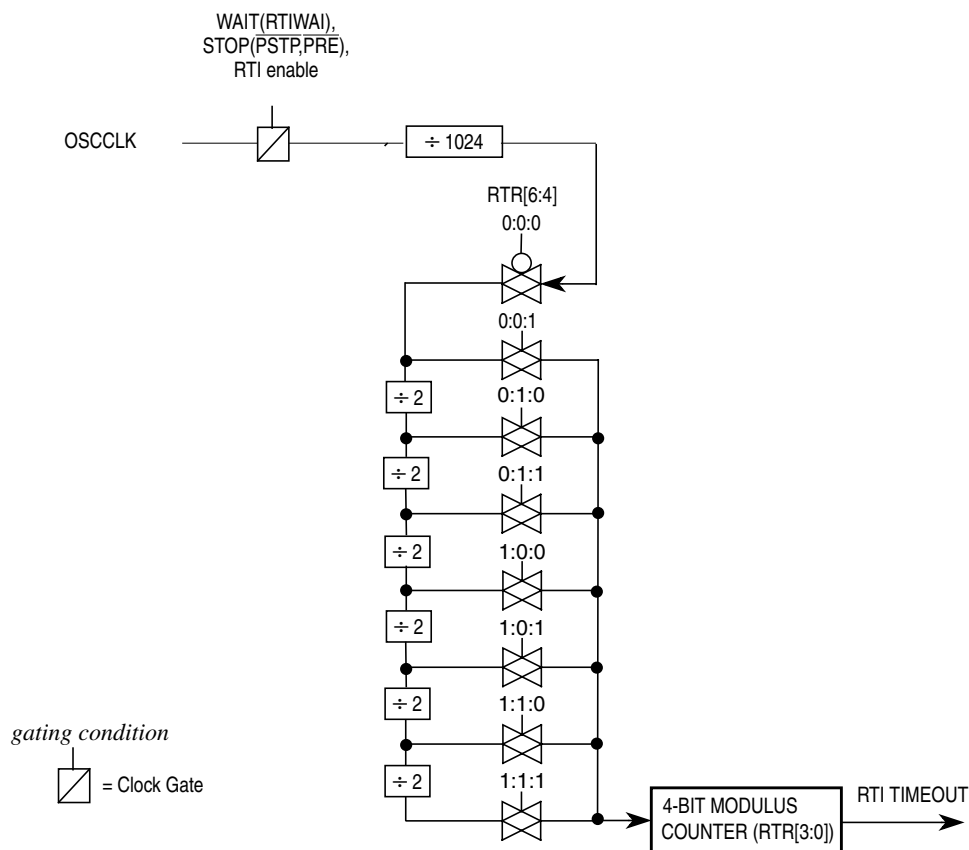


Figure 9-22. Clock Chain for RTI

9.4.7 Modes of Operation

9.4.7.1 Normal Mode

The CRGV4 block behaves as described within this specification in all normal modes.

9.4.7.2 Self-Clock Mode

The VCO has a minimum operating frequency, f_{SCM} . If the external clock frequency is not available due to a failure or due to long crystal start-up time, the bus clock and the core clock are derived from the VCO running at minimum operating frequency; this mode of operation is called self-clock mode. This requires $CME = 1$ and $SCME = 1$. If the MCU was clocked by the PLL clock prior to entering self-clock mode, the PLLSEL bit will be cleared. If the external clock signal has stabilized again, the CRG will automatically select OSCCLK to be the system clock and return to normal mode. See [Section 9.4.4, “Clock Quality Checker”](#) for more information on entering and leaving self-clock mode.

10.1.4 Modes of Operation

The following modes of operation are specific to the MSCAN. See [Section 10.4, “Functional Description,”](#) for details.

- Listen-Only Mode
- MSCAN Sleep Mode
- MSCAN Initialization Mode
- MSCAN Power Down Mode

10.2 External Signal Description

The MSCAN uses two external pins:

10.2.1 RXCAN — CAN Receiver Input Pin

RXCAN is the MSCAN receiver input pin.

10.2.2 TXCAN — CAN Transmitter Output Pin

TXCAN is the MSCAN transmitter output pin. The TXCAN output pin represents the logic level on the CAN bus:

- 0 = Dominant state
- 1 = Recessive state

10.2.3 CAN System

A typical CAN system with MSCAN is shown in [Figure 10-2](#). Each CAN station is connected physically to the CAN bus lines through a transceiver device. The transceiver is capable of driving the large current needed for the CAN bus and has current protection against defective CAN or defective stations.

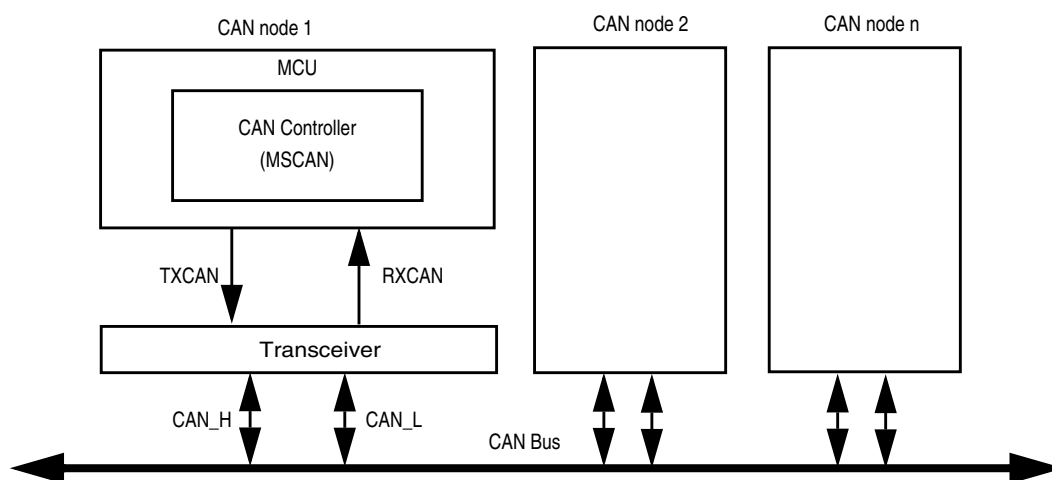


Figure 10-2. CAN System

Module Base + 0x00X1

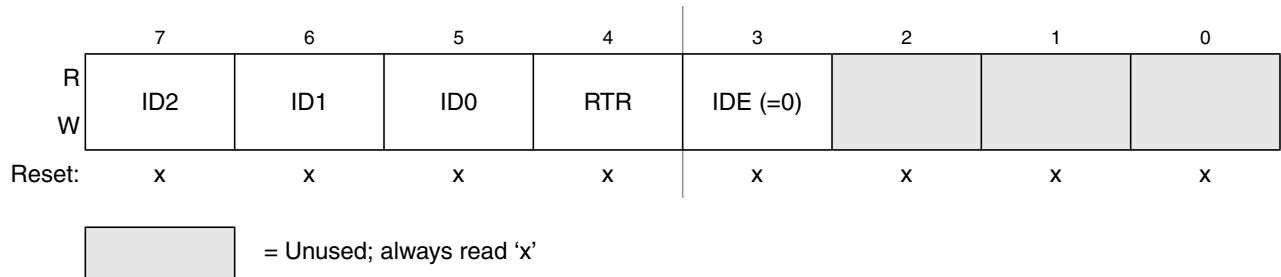


Figure 10-30. Identifier Register 1 — Standard Mapping

Table 10-29. IDR1 Register Field Descriptions

Field	Description
7:5 ID[2:0]	Standard Format Identifier — The identifiers consist of 11 bits (ID[10:0]) for the standard format. ID10 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number. See also ID bits in Table 10-28 .
4 RTR	Remote Transmission Request — This flag reflects the status of the Remote Transmission Request bit in the CAN frame. In the case of a receive buffer, it indicates the status of the received frame and supports the transmission of an answering frame in software. In the case of a transmit buffer, this flag defines the setting of the RTR bit to be sent. 0 Data frame 1 Remote frame
3 IDE	ID Extended — This flag indicates whether the extended or standard identifier format is applied in this buffer. In the case of a receive buffer, the flag is set as received and indicates to the CPU how to process the buffer identifier registers. In the case of a transmit buffer, the flag indicates to the MSCAN what type of identifier to send. 0 Standard format (11 bit) 1 Extended format (29 bit)

Module Base + 0x00X2

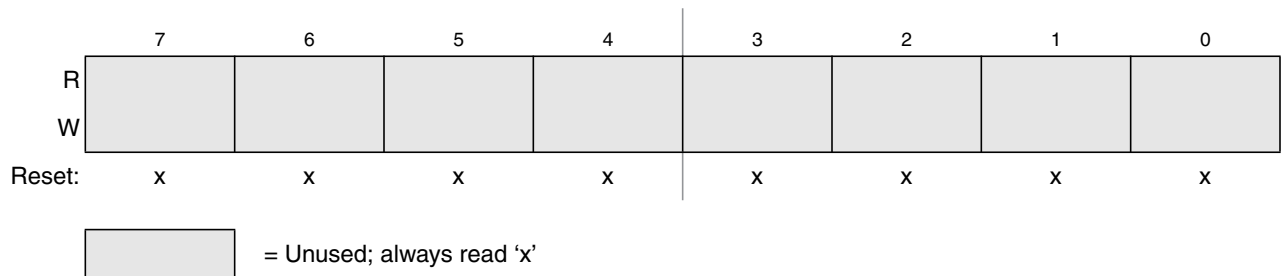


Figure 10-31. Identifier Register 2 — Standard Mapping

11.2 External Signal Description

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

11.2.1 V_{DDPLL} and V_{SSPLL} — PLL Operating Voltage, PLL Ground

These pins provide the operating voltage (V_{DDPLL}) and ground (V_{SSPLL}) for the OSCV2 circuitry. This allows the supply voltage to the OSCV2 to be independently bypassed.

11.2.2 EXTAL and XTAL — Clock/Crystal Source Pins

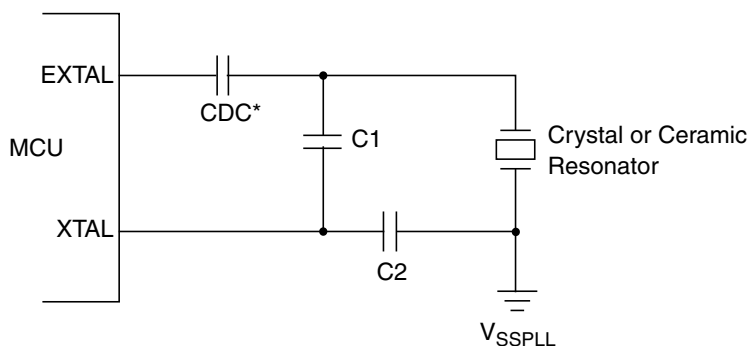
These pins provide the interface for either a crystal or a CMOS compatible clock to control the internal clock generator circuitry. EXTAL is the external clock input or the input to the crystal oscillator amplifier. XTAL is the output of the crystal oscillator amplifier. All the MCU internal system clocks are derived from the EXTAL input frequency. In full stop mode (PSTP = 0) the EXTAL pin is pulled down by an internal resistor of typical 200 k Ω .

NOTE

Freescall Semiconductor recommends an evaluation of the application board and chosen resonator or crystal by the resonator or crystal supplier.

The Crystal circuit is changed from standard.

The Colpitts circuit is not suited for overtone resonators and crystals.



* Due to the nature of a translated ground Colpitts oscillator a DC voltage bias is applied to the crystal.

Please contact the crystal manufacturer for crystal DC bias conditions and recommended capacitor value CDC.

Figure 11-1. Colpitts Oscillator Connections (XCLKS = 0)

NOTE

The Pierce circuit is not suited for overtone resonators and crystals without a careful component selection.

Module Base + 0x0005

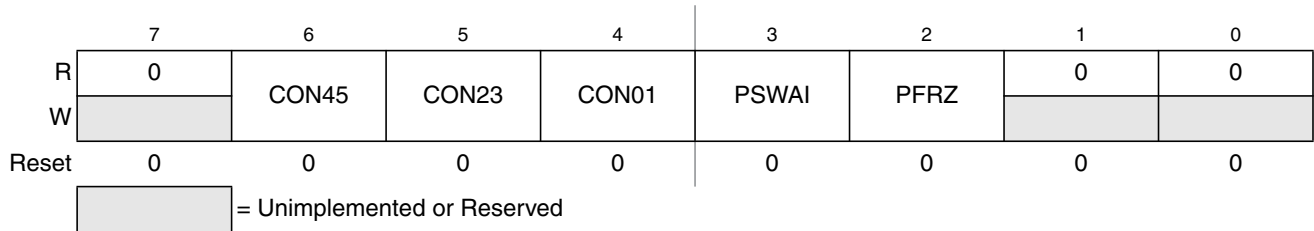


Figure 12-8. PWM Control Register (PWMCTL)

Read: anytime

Write: anytime

There are three control bits for concatenation, each of which is used to concatenate a pair of PWM channels into one 16-bit channel. When channels 4 and 5 are concatenated, channel 4 registers become the high-order bytes of the double-byte channel. When channels 2 and 3 are concatenated, channel 2 registers become the high-order bytes of the double-byte channel. When channels 0 and 1 are concatenated, channel 0 registers become the high-order bytes of the double-byte channel.

Reference [Section 12.4.2.7, “PWM 16-Bit Functions,”](#) for a more detailed description of the concatenation PWM function.

NOTE

Change these bits only when both corresponding channels are disabled.

Table 14-7. Example SPI Baud Rate Selection (25 MHz Bus Clock)

SPPR2	SPPR1	SPPR0	SPR2	SPR1	SPR0	Baud Rate Divisor	Baud Rate
0	0	0	0	0	0	2	12.5 MHz
0	0	0	0	0	1	4	6.25 MHz
0	0	0	0	1	0	8	3.125 MHz
0	0	0	0	1	1	16	1.5625 MHz
0	0	0	1	0	0	32	781.25 kHz
0	0	0	1	0	1	64	390.63 kHz
0	0	0	1	1	0	128	195.31 kHz
0	0	0	1	1	1	256	97.66 kHz
0	0	1	0	0	0	4	6.25 MHz
0	0	1	0	0	1	8	3.125 MHz
0	0	1	0	1	0	16	1.5625 MHz
0	0	1	0	1	1	32	781.25 kHz
0	0	1	1	0	0	64	390.63 kHz
0	0	1	1	0	1	128	195.31 kHz
0	0	1	1	1	0	256	97.66 kHz
0	0	1	1	1	1	512	48.83 kHz
0	1	0	0	0	0	6	4.16667 MHz
0	1	0	0	0	1	12	2.08333 MHz
0	1	0	0	1	0	24	1.04167 MHz
0	1	0	0	1	1	48	520.83 kHz
0	1	0	1	0	0	96	260.42 kHz
0	1	0	1	0	1	192	130.21 kHz
0	1	0	1	1	0	384	65.10 kHz
0	1	0	1	1	1	768	32.55 kHz
0	1	1	0	0	0	8	3.125 MHz
0	1	1	0	0	1	16	1.5625 MHz
0	1	1	0	1	0	32	781.25 kHz
0	1	1	0	1	1	64	390.63 kHz
0	1	1	1	0	0	128	195.31 kHz
0	1	1	1	0	1	256	97.66 kHz
0	1	1	1	1	0	512	48.83 kHz
0	1	1	1	1	1	1024	24.41 kHz
1	0	0	0	0	0	10	2.5 MHz
1	0	0	0	0	1	20	1.25 MHz
1	0	0	0	1	0	40	625 kHz
1	0	0	0	1	1	80	312.5 kHz
1	0	0	1	0	0	160	156.25 kHz
1	0	0	1	0	1	320	78.13 kHz
1	0	0	1	1	0	640	39.06 kHz

14.3.2.4 SPI Status Register (SPISR)

Module Base 0x0003

	7	6	5	4	3	2	1	0
R	SPIF	0	SPTEF	MODF	0	0	0	0
W								
Reset	0	0	1	0	0	0	0	0

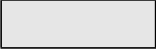
 = Unimplemented or Reserved

Figure 14-6. SPI Status Register (SPISR)

Read: anytime

Write: has no effect

Table 14-8. SPISR Field Descriptions

Field	Description
7 SPIF	SPIF Interrupt Flag — This bit is set after a received data byte has been transferred into the SPI Data Register. This bit is cleared by reading the SPISR register (with SPIF set) followed by a read access to the SPI Data Register. 0 Transfer not yet complete 1 New data copied to SPIDR
5 SPTEF	SPI Transmit Empty Interrupt Flag — If set, this bit indicates that the transmit data register is empty. To clear this bit and place data into the transmit data register, SPISR has to be read with SPTEF = 1, followed by a write to SPIDR. Any write to the SPI Data Register without reading SPTEF = 1, is effectively ignored. 0 SPI Data register not empty 1 SPI Data register empty
4 MODF	Mode Fault Flag — This bit is set if the \overline{SS} input becomes low while the SPI is configured as a master and mode fault detection is enabled, MODFEN bit of SPICR2 register is set. Refer to MODFEN bit description in Section 14.3.2.2, “SPI Control Register 2 (SPICR2)” . The flag is cleared automatically by a read of the SPI Status Register (with MODF set) followed by a write to the SPI Control Register 1. 0 Mode fault has not occurred. 1 Mode fault has occurred.

14.3.2.5 SPI Data Register (SPIDR)

Module Base 0x0005

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


 = Unimplemented or Reserved

Figure 14-7. SPI Data Register (SPIDR)

Read: anytime; normally read only after SPIF is set

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.

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.

21.4.1.3.4 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 21-25](#). 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.