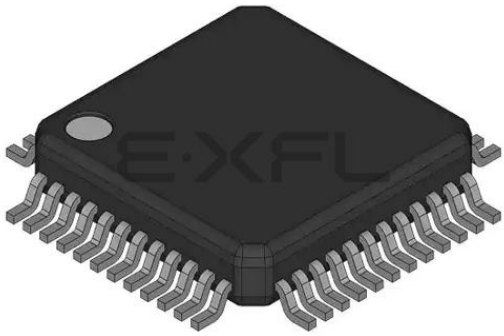


Welcome to E-XFL.COM

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	EBI/EMI, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	31
Program Memory Size	64KB (64K 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 ~ 85°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=mc9s12gc64cfae

1.5.3.3 Wait

This mode is entered by executing the CPU WAI instruction. In this mode the CPU will not execute instructions. The internal CPU signals (address and data bus) will be fully static. All peripherals stay active. For further power consumption reduction the peripherals can individually turn off their local clocks.

1.5.3.4 Run

Although this is not a low-power mode, unused peripheral modules should not be enabled in order to save power.

1.6 Resets and Interrupts

Consult the Exception Processing section of the CPU12 Reference Manual for information.

1.6.1 Vectors

Table 1-9 lists interrupt sources and vectors in default order of priority.

Table 1-9. Interrupt Vector Locations

Vector Address	Interrupt Source	CCR Mask	Local Enable	HPRIO Value to Elevate
0xFFFE, 0xFFFF	External reset, power on reset, or low voltage reset (see CRG flags register to determine reset source)	None	None	—
0xFFFC, 0xFFFD	Clock monitor fail reset	None	COPCTL (CME, FCME)	—
0xFFFA, 0xFFFB	COP failure reset	None	COP rate select	—
0xFFF8, 0xFFF9	Unimplemented instruction trap	None	None	—
0xFFF6, 0xFFF7	SWI	None	None	—
0xFFF4, 0xFFF5	XIRQ	X-Bit	None	—
0xFFF2, 0xFFF3	IRQ	I bit	INTCR (IRQEN)	0x00F2
0xFFF0, 0xFFF1	Real time Interrupt	I bit	CRGINT (RTIE)	0x00F0
0xFFEE, 0xFFEF	Standard timer channel 0	I bit	TIE (C0I)	0x00EE
0xFFEC, 0xFFED	Standard timer channel 1	I bit	TIE (C1I)	0x00EC
$FFEE, $FFEF	Reserved			
$FFEC, $FFED	Reserved			
0xFFEA, 0xFFEB	Standard timer channel 2	I bit	TIE (C2I)	0x00EA
0xFFE8, 0xFFE9	Standard timer channel 3	I bit	TIE (C3I)	0x00E8
0xFFE6, 0xFFE7	Standard timer channel 4	I bit	TIE (C4I)	0x00E6
0xFFE4, 0xFFE5	Standard timer channel 5	I bit	TIE (C5I)	0x00E4
0xFFE2, 0xFFE3	Standard timer channel 6	I bit	TIE (C6I)	0x00E2
0xFFE0, 0xFFE1	Standard timer channel 7	I bit	TIE (C7I)	0x00E0

Table 1-9. Interrupt Vector Locations (continued)

Vector Address	Interrupt Source	CCR Mask	Local Enable	HPRIO Value to Elevate
0xFFDE, 0xFFDF	Standard timer overflow	I bit	TMSK2 (TOI)	0x00DE
0xFFDC, 0xFFDD	Pulse accumulator A overflow	I bit	PACTL (PAOVI)	0x00DC
0xFFDA, 0xFFDB	Pulse accumulator input edge	I bit	PACTL (PAI)	0x00DA
0xFFD8, 0xFFD9	SPI	I bit	SPICR1 (SPIE, SPTIE)	0x00D8
0xFFD6, 0xFFD7	SCI	I bit	SCICR2 (TIE, TCIE, RIE, ILIE)	0x00D6
0xFFD4, 0xFFD5	Reserved			
0xFFD2, 0xFFD3	ATD	I bit	ATDCTL2 (ASCIE)	0x00D2
0xFFD0, 0xFFD1	Reserved			
0xFFCE, 0xFFCF	Port J	I bit	PIEP (PIEP7-6)	0x00CE
0xFFCC, 0xFFCD	Reserved			
0xFFCA, 0xFFCB	Reserved			
0xFFC8, 0xFFC9	Reserved			
0xFFC6, 0xFFC7	CRG PLL lock	I bit	PLLCR (LOCKIE)	0x00C6
0xFFC4, 0xFFC5	CRG self clock mode	I bit	PLLCR (SCMIE)	0x00C4
0xFFBA to 0xFFC3	Reserved			
0xFFB8, 0xFFB9	FLASH	I bit	FCNFG (CCIE, CBEIE)	0x00B8
0xFFB6, 0xFFB7	CAN wake-up(1)	I bit	CANRIER (WUPIE)	0x00B6
0xFFB4, 0xFFB5	CAN errors[1]	I bit	CANRIER (CSCIE, OVRIE)	0x00B4
0xFFB2, 0xFFB3	CAN receive[1]	I bit	CANRIER (RXFIE)	0x00B2
0xFFB0, 0xFFB1	CAN transmit[1]	I bit	CANTIER (TXEIE[2:0])	0x00B0
0xFF90 to 0xFFAF	Reserved			
0xFF8E, 0xFF8F	Port P	I bit	PIEP (PIEP7-0)	0x008E
0xFF8C, 0xFF8D	Reserved			
0xFF8C, 0xFF8D	PWM Emergency Shutdown	I bit	PWMSDN(PWMIE)	0x008C
0xFF8A, 0xFF8B	VREG LVI	I bit	CTRL0 (LVIE)	0x008A
0xFF80 to 0xFF89	Reserved			

1. Not available on MC9S12GC Family members

2.3.2.4.3 Port P Data Direction Register (DDRP)

Module Base + 0x001A

	7	6	5	4	3	2	1	0
R W	DDRP7	DDRP6	DDRP5	DDRP4	DDRP3	DDRP2	DDRP1	DDRP0
Reset	0	0	0	0	0	0	0	0

Figure 2-26. Port P Data Direction Register (DDRP)

Read: Anytime.

Write: Anytime.

Table 2-22. DDRP Field Descriptions

Field	Description
7–0 DDRP[7:0]	**Data Direction Port P** — This register configures each port P pin as either input or output. 0 Associated pin is configured as input. 1 Associated pin is configured as output. **Note:** Due to internal synchronization circuits, it can take up to 2 bus cycles until the correct value is read on PTP or PTIP registers, when changing the DDRP register.

2.3.2.4.4 Port P Reduced Drive Register (RDRP)

Module Base + 0x001B

	7	6	5	4	3	2	1	0
R W	RDRP7	RDRP6	RDRP5	RDRP4	RDRP3	RDRP2	RDRP1	RDRP0
Reset	0	0	0	0	0	0	0	0

Figure 2-27. Port P Reduced Drive Register (RDRP)

Read: Anytime.

Write: Anytime.

Table 2-23. RDRP Field Descriptions

Field	Description
7–0 RDRP[7:0]	**Reduced Drive Port P** — This register configures the drive strength of each port P output pin as either full or reduced. If the port is used as input this bit is ignored. 0 Full drive strength at output. 1 Associated pin drives at about 1/3 of the full drive strength.

Table 2-39. Port Reset State Summary

Port	Reset States				
	Data Direction	Pull Mode	Reduced Drive	Wired-OR Mode	Interrupt
T	Input	Hi-z	Disabled	n/a	n/a
S	Input	Pull up	Disabled	Disabled	n/a
M	Input	Pull up	Disabled	Disabled	n/a
P	Input	Hi-z	Disabled	n/a	Disabled
J	Input	Hi-z	Disabled	n/a	Disabled
A	Refer to MEBI Block Guide for details.				
B					
E					
BKGD pin	Refer to BDM Block Guide for details.				

2.6 Interrupts

Port P and J generate a separate edge sensitive interrupt if enabled.

2.6.1 Interrupt Sources

Table 2-40. Port Integration Module Interrupt Sources

Interrupt Source	Interrupt Flag	Local Enable	Global (CCR) Mask
Port P	PIFP[7:0]	PIEP[7:0]	I Bit
Port J	PIFJ[7:6]	PIEJ[7:6]	I Bit

NOTE

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

2.6.2 Recovery from STOP

The PIM can generate wake-up interrupts from STOP on port P and J. For other sources of external interrupts please refer to the respective Block User Guide.

2.7 Application Information

It is not recommended to write PORTx and DDRx in a word access. When changing the register pins from inputs to outputs, the data may have extra transitions during the write access. Initialize the port data register before enabling the outputs.

Power consumption will increase the more the voltages on general purpose input pins deviate from the supply voltages towards mid-range because the digital input buffers operate in the linear region.

Table 4-8. MODC, MODB, and MODA Write Capability[1]

MODC	MODB	MODA	Mode	MODx Write Capability
0	0	0	Special single chip	MODC, MODB, and MODA write anytime but not to 110[2]
0	0	1	Emulation narrow	No write
0	1	0	Special test	MODC, MODB, and MODA write anytime but not to 110[2]
0	1	1	Emulation wide	No write
1	0	0	Normal single chip	MODC write never, MODB and MODA write once but not to 110
1	0	1	Normal expanded narrow	No write
1	1	0	Special peripheral	No write
1	1	1	Normal expanded wide	No write

1. No writes to the MOD bits are allowed while operating in a secure mode. For more details, refer to the device overview chapter.
2. If you are in a special single-chip or special test mode and you write to this register, changing to normal single-chip mode, then one allowed write to this register remains. If you write to normal expanded or emulation mode, then no writes remain.

4.3.2.10 Pull Control Register (PUCR)

Module Base + 0x000C
Starting address location affected by INITRG register setting.

	7	6	5	4	3	2	1	0
R	PUPKE	0	0	PUPEE	0	0	PUPBE	PUPAE
W								
Reset[1]	1	0	0	1	0	0	0	0

NOTES:
1. The default value of this parameter is shown. Please refer to the device overview chapter to determine the actual reset state of this register.

= Unimplemented or Reserved

Figure 4-14. Pull Control Register (PUCR)

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

Write: Anytime (provided this register is in the map).

This register is used to select pull resistors for the pins associated with the core ports. Pull resistors are assigned on a per-port basis and apply to any pin in the corresponding port that is currently configured as an input. The polarity of these pull resistors is determined by chip integration. Please refer to the device overview chapter to determine the polarity of these resistors.

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

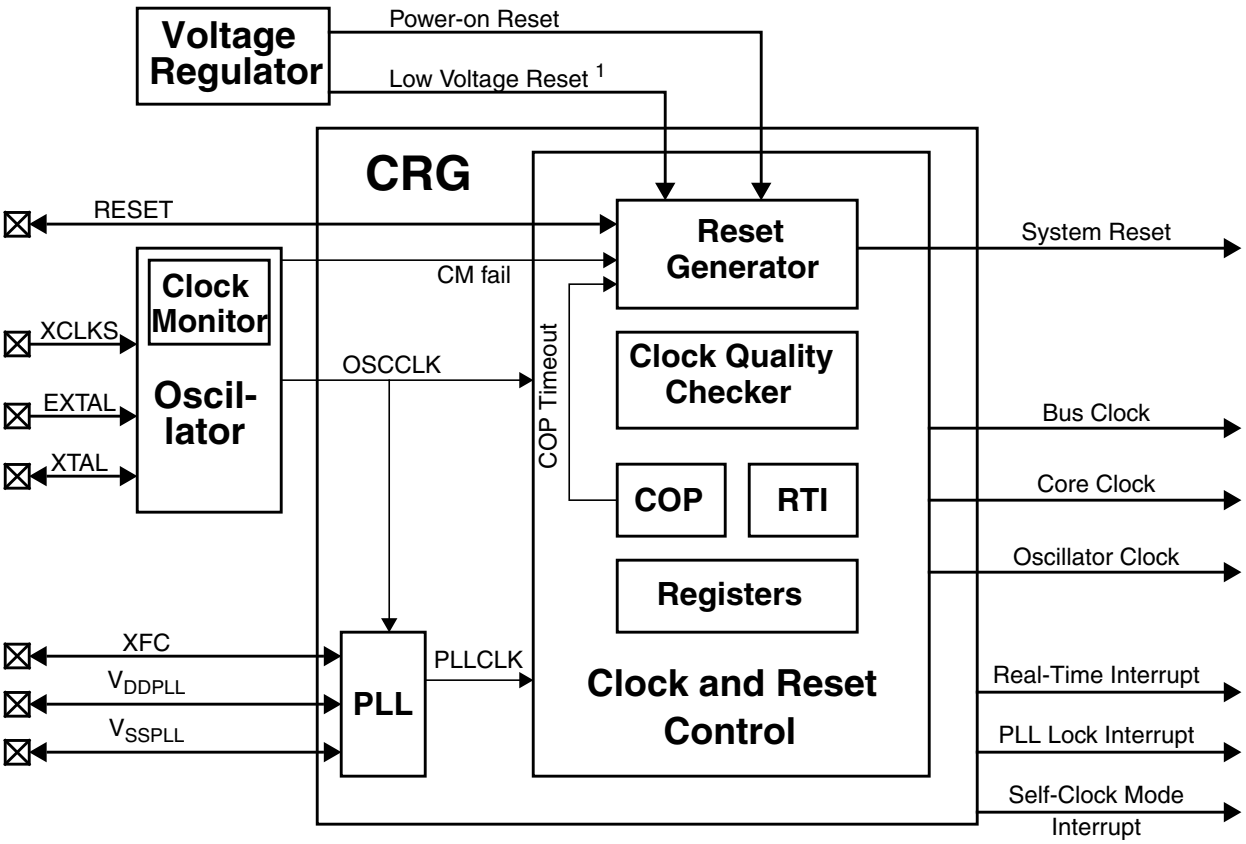
As shown in the block diagram in Figure 5-1, the INT contains a register block to provide interrupt status and control, an optional highest priority I interrupt (HPRIO) block, and a priority decoder to evaluate whether pending interrupts are valid and assess their priority.

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.



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}.

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	Clock failure --> Scenario 1: OSCCLK **recovers** prior to exiting Wait Mode. – MCU remains in Wait Mode, – VREG enabled, – PLL enabled, – SCM activated, – Start Clock Quality Check, – Set SCMIF interrupt flag. *Some time later OSCCLK recovers.* – 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 *(never gets disabled in Wait Mode)*. – MCU remains in Wait Mode. *Some time later either a wakeup interrupt occurs (no SCM interrupt)* – Exit Wait Mode using OSCCLK as system clock (SYSCLK), – Continue normal operation. *or an External Reset is applied.* – Exit Wait Mode using OSCCLK as system clock, – Start reset sequence. Scenario 2: OSCCLK **does not recover** prior to exiting Wait Mode. – 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. *Some time later either a wakeup interrupt occurs (no SCM interrupt)* – 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. or an External RESET is applied. – 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.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 CANCTL0	R	RXFRM	RXACT	CSWAI	SYNCH	TIME	WUPE	SLPRQ	INITRQ
	W								
0x0001 CANCTL1	R	CANE	CLKSRC	LOOPB	LISTEN		WUPM	SLPAK	INITAK
	W								
0x0002 CANBTR0	R	SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
	W								
0x0003 CANBTR1	R	SAMP	TSEG22	TSEG21	TSEG20	TSEG13	TSEG12	TSEG11	TSEG10
	W								
0x0004 CANRFLG	R	WUPIF	CSCIF	RSTAT1	RSTAT0	TSTAT1	TSTAT0	OVRIF	RXF
	W								
0x0005 CANRIER	R	WUPIE	CSCIE	RSTATE1	RSTATE0	TSTATE1	TSTATE0	OVRIE	RXFIE
	W								
0x0006 CANTFLG	R	0	0	0	0	0	TXE2	TXE1	TXE0
	W								
0x0007 CANTIER	R	0	0	0	0	0	TXEIE2	TXEIE1	TXEIE0
	W								
0x0008 CANTARQ	R	0	0	0	0	0	ABTRQ2	ABTRQ1	ABTRQ0
	W								
0x0009 CANTAAK	R	0	0	0	0	0	ABTAK2	ABTAK1	ABTAK0
	W								
0x000A CANTBSEL	R	0	0	0	0	0	TX2	TX1	TX0
	W								
0x000B CANIDAC	R	0	0	IDAM1	IDAM0	0	IDHIT2	IDHIT1	IDHIT0
	W								
0x000C–0x000D Reserved	R	0	0	0	0	0	0	0	0
	W								
0x000E CANRXERR	R	RXERR7	RXERR6	RXERR5	RXERR4	RXERR3	RXERR2	RXERR1	RXERR0
	W								
0x000F CANTXERR	R	TXERR7	TXERR6	TXERR5	TXERR4	TXERR3	TXERR2	TXERR1	TXERR0
	W								

= Unimplemented or Reserved u = Unaffected

Figure 10-3. MSCAN Register Summary

Table 12-12 is used to summarize which channels are used to set the various control bits when in 16-bit mode.

Table 12-12. 16-bit Concatenation Mode Summary

CONxx	PWMEx	PPOLx	PCLKx	CAEx	PWMx Output
CON45	PWME5	PPOL5	PCLK5	CAE5	PWM5
CON23	PWME3	PPOL3	PCLK3	CAE3	PWM3
CON01	PWME1	PPOL1	PCLK1	CAE1	PWM1

12.4.2.8 PWM Boundary Cases

Table 12-13 summarizes the boundary conditions for the PWM regardless of the output mode (left aligned or center aligned) and 8-bit (normal) or 16-bit (concatenation):

Table 12-13. PWM Boundary Cases

PWMDTYx	PWMPERx	PPOLx	PWMx Output
0x0000 (indicates no duty)	>0x0000	1	Always Low
0x0000 (indicates no duty)	>0x0000	0	Always High
XX	0x0000(1) (indicates no period)	1	Always High
XX	0x0000[1] (indicates no period)	0	Always Low
>= PWMPERx	XX	1	Always High
>= PWMPERx	XX	0	Always Low

1. Counter = 0x0000 and does not count.

12.5 Resets

The reset state of each individual bit is listed within the register description section (see Section 12.3, “Memory Map and Register Definition,” which details the registers and their bit-fields. All special functions or modes which are initialized during or just following reset are described within this section.

- The 8-bit up/down counter is configured as an up counter out of reset.
- All the channels are disabled and all the counters don’t count.

12.6 Interrupts

The PWM8B6CV1 module has only one interrupt which is generated at the time of emergency shutdown, if the corresponding enable bit (PWMIE) is set. This bit is the enable for the interrupt. The interrupt flag PWMIF is set whenever the input level of the PWM5 channel changes while PWM5ENA=1 or when PWMENA is being asserted while the level at PWM5 is active.

A description of the registers involved and affected due to this interrupt is explained in Section 12.3.2.15, “PWM Shutdown Register (PWMSDN).”

writing another byte to the Transmitter buffer (SCIDRH/SCIDRL), while the shift register is still shifting out the first byte.

To initiate an SCI transmission:

1. Configure the SCI:
   a) Select a baud rate. Write this value to the SCI baud registers (SCIBDH/L) to begin the baud rate generator. Remember that the baud rate generator is disabled when the baud rate is zero. Writing to the SCIBDH has no effect without also writing to SCIBDL.
   b) Write to SCICR1 to configure word length, parity, and other configuration bits (LOOPS,RSRC,M,WAKE,ILT,PE,PT).
   c) Enable the transmitter, interrupts, receive, and wake up as required, by writing to the SCICR2 register bits (TIE,TCIE,RIE,ILIE,TE,RE,RWU,SBK). A preamble or idle character will now be shifted out of the transmitter shift register.
2. Transmit Procedure for Each Byte:
   a. Poll the TDRE flag by reading the SCISR1 or responding to the TDRE interrupt. Keep in mind that the TDRE bit resets to one.
   d) If the TDRE flag is set, write the data to be transmitted to SCIDRH/L, where the ninth bit is written to the T8 bit in SCIDRH if the SCI is in 9-bit data format. A new transmission will not result until the TDRE flag has been cleared.
3. Repeat step 2 for each subsequent transmission.

NOTE

The TDRE flag is set when the shift register is loaded with the next data to be transmitted from SCIDRH/L, which happens, generally speaking, a little over half-way through the stop bit of the previous frame. Specifically, this transfer occurs 9/16ths of a bit time AFTER the start of the stop bit of the previous frame.

Writing the TE bit from 0 to a 1 automatically loads the transmit shift register with a preamble of 10 logic 1s (if M = 0) or 11 logic 1s (if M = 1). After the preamble shifts out, control logic transfers the data from the SCI data register into the transmit shift register. A logic 0 start bit automatically goes into the least significant bit position of the transmit shift register. A logic 1 stop bit goes into the most significant bit position.

Hardware supports odd or even parity. When parity is enabled, the most significant bit (msb) of the data character is the parity bit.

The transmit data register empty flag, TDRE, in SCI status register 1 (SCISR1) becomes set when the SCI data register transfers a byte to the transmit shift register. The TDRE flag indicates that the SCI data register can accept new data from the internal data bus. If the transmit interrupt enable bit, TIE, in SCI control register 2 (SCICR2) is also set, the TDRE flag generates a transmitter interrupt request.

When the transmit shift register is not transmitting a frame, the **Tx output** signal goes to the idle condition, logic 1. If at any time software clears the TE bit in SCI control register 2 (SCICR2), the transmitter enable signal goes low and the transmit signal goes idle.

Chapter 14
Serial Peripheral Interface (SPIV3) Block Description

14.1 Introduction

The SPI module allows a duplex, synchronous, serial communication between the MCU and peripheral devices. Software can poll the SPI status flags or the SPI operation can be interrupt driven.

14.1.1 Features

The SPIV3 includes these distinctive features:

- Master mode and slave mode
- Bidirectional mode
- Slave select output
- Mode fault error flag with CPU interrupt capability
- Double-buffered data register
- Serial clock with programmable polarity and phase
- Control of SPI operation during wait mode

14.1.2 Modes of Operation

The SPI functions in three modes, run, wait, and stop.

- Run Mode
  This is the basic mode of operation.
- Wait Mode
  SPI operation in wait mode is a configurable low power mode, controlled by the SPISWAI bit located in the SPICR2 register. In wait mode, if the SPISWAI bit is clear, the SPI operates like in Run Mode. If the SPISWAI bit is set, the SPI goes into a power conservative state, with the SPI clock generation turned off. If the SPI is configured as a master, any transmission in progress stops, but is resumed after CPU goes into Run Mode. If the SPI is configured as a slave, reception and transmission of a byte continues, so that the slave stays synchronized to the master.
- Stop Mode
  The SPI is inactive in stop mode for reduced power consumption. If the SPI is configured as a master, any transmission in progress stops, but is resumed after CPU goes into run mode. If the SPI is configured as a slave, reception and transmission of a byte continues, so that the slave stays synchronized to the master.

This is a high level description only, detailed descriptions of operating modes are contained in Section 14.4, “Functional Description.”

14.1.3 Block Diagram

Figure 14-1 gives an overview on the SPI architecture. The main parts of the SPI are status, control, and data registers, shifter logic, baud rate generator, master/slave control logic, and port control logic.

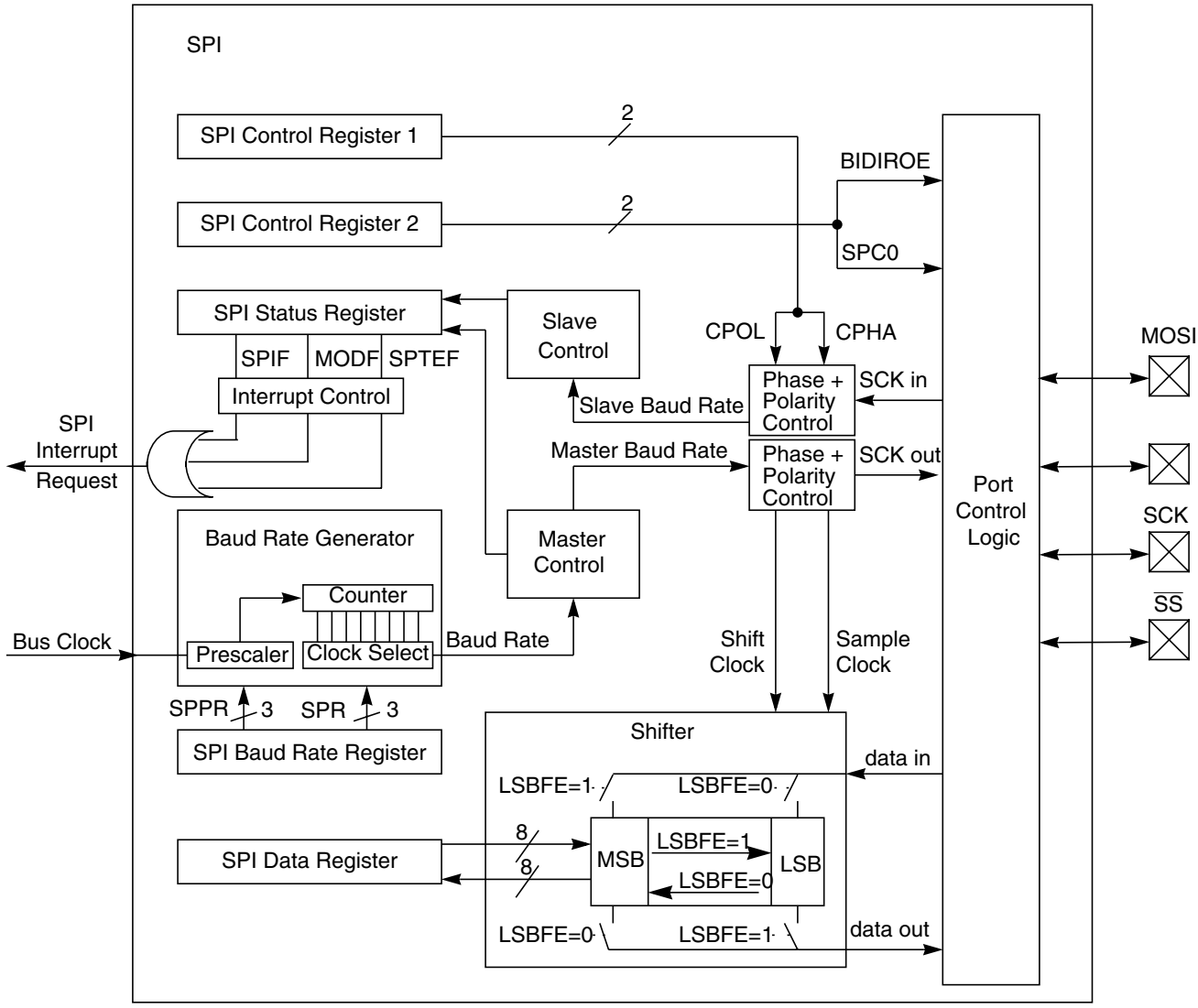


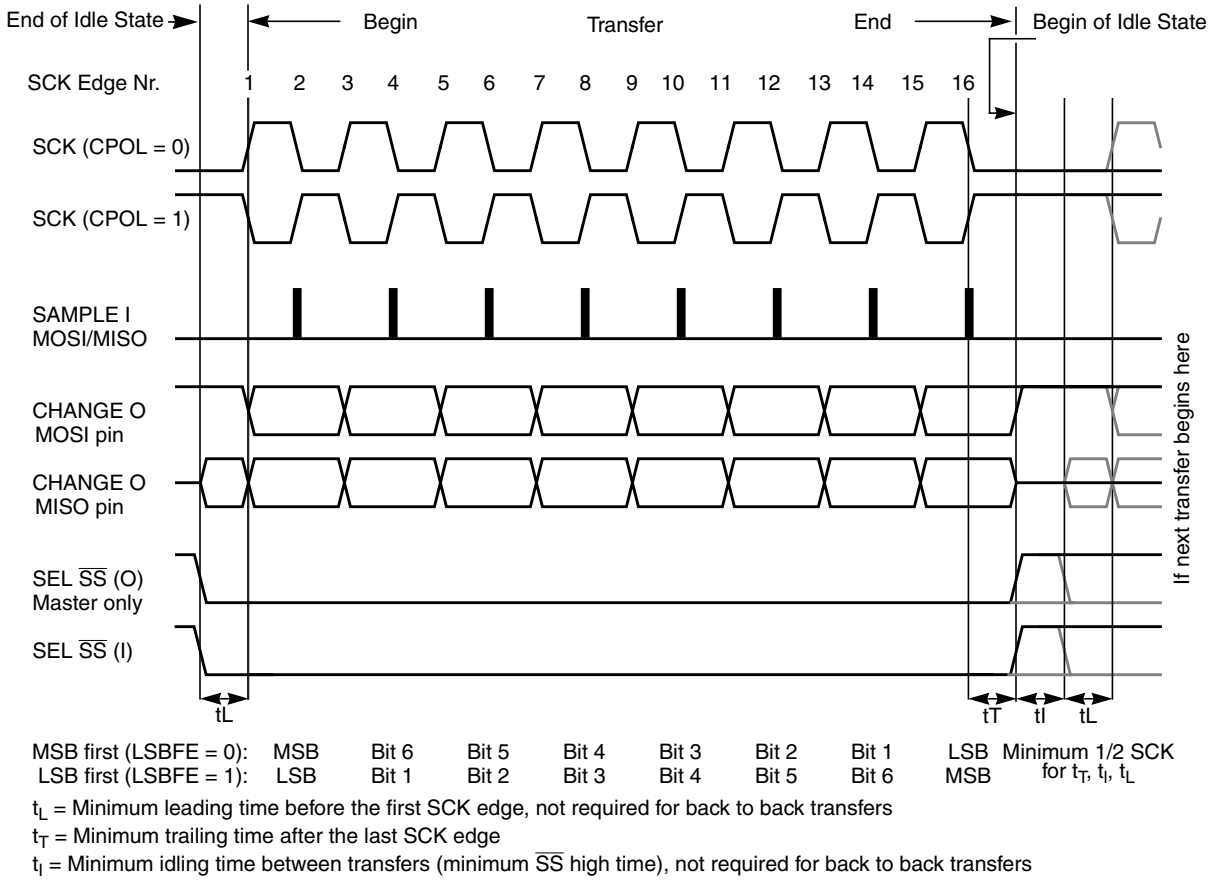
Figure 14-1. SPI Block Diagram

14.2 External Signal Description

This section lists the name and description of all ports including inputs and outputs that do, or may, connect off chip. The SPIV3 module has a total of four external pins.

14.2.1 MOSI — Master Out/Slave In Pin

This pin is used to transmit data out of the SPI module when it is configured as a master and receive data when it is configured as slave.



t_L = Minimum leading time before the first SCK edge, not required for back to back transfers
t_T = Minimum trailing time after the last SCK edge
t_I = Minimum idling time between transfers (minimum $\overline{SS}$ high time), not required for back to back transfers

Figure 14-10. SPI Clock Format 1 (CPHA = 1)

14.4.4 SPI Baud Rate Generation

Baud rate generation consists of a series of divider stages. Six bits in the SPI Baud Rate register (SPPR2, SPPR1, SPPR0, SPR2, SPR1, and SPR0) determine the divisor to the SPI module clock which results in the SPI baud rate.

The SPI clock rate is determined by the product of the value in the baud rate preselection bits (SPPR2–SPPR0) and the value in the baud rate selection bits (SPR2–SPR0). The module clock divisor equation is shown in Figure 14-11

When all bits are clear (the default condition), the SPI module clock is divided by 2. When the selection bits (SPR2–SPR0) are 001 and the preselection bits (SPPR2–SPPR0) are 000, the module clock divisor becomes 4. When the selection bits are 010, the module clock divisor becomes 8 etc.

When the preselection bits are 001, the divisor determined by the selection bits is multiplied by 2. When the preselection bits are 010, the divisor is multiplied by 3, etc. See Table 14-7 for baud rate calculations for all bit conditions, based on a 25-MHz bus clock. The two sets of selects allows the clock to be divided by a non-power of two to achieve other baud rates such as divide by 6, divide by 10, etc.

Table 15-7. TSCR1 Field Descriptions (continued)

Field	Description
5 TSFRZ	**Timer Stops While in Freeze Mode** 0 Allows the timer counter to continue running while in freeze mode. 1 Disables the timer counter whenever the MCU is in freeze mode. This is useful for emulation. TSFRZ does not stop the pulse accumulator.
4 TFFCA	**Timer Fast Flag Clear All** 0 Allows the timer flag clearing to function normally. 1 For TFLG1(0x000E), a read from an input capture or a write to the output compare channel (0x0010–0x001F) causes the corresponding channel flag, CnF, to be cleared. For TFLG2 (0x000F), any access to the TCNT register (0x0004, 0x0005) clears the TOF flag. Any access to the PACNT registers (0x0022, 0x0023) clears the PAOVF and PAIF flags in the PAFLG register (0x0021). This has the advantage of eliminating software overhead in a separate clear sequence. Extra care is required to avoid accidental flag clearing due to unintended accesses.

15.3.2.7 Timer Toggle On Overflow Register 1 (TTOV)

Module Base + 0x0007

	7	6	5	4	3	2	1	0
R W	TOV7	TOV6	TOV5	TOV4	TOV3	TOV2	TOV1	TOV0
Reset	0	0	0	0	0	0	0	0

Figure 15-13. Timer Toggle On Overflow Register 1 (TTOV)

Read: Anytime

Write: Anytime

Table 15-8. TTOV Field Descriptions

Field	Description
7:0 TOV[7:0]	**Toggle On Overflow Bits** — TOVx toggles output compare pin on overflow. This feature only takes effect when in output compare mode. When set, it takes precedence over forced output compare but not channel 7 override events. 0 Toggle output compare pin on overflow feature disabled. 1 Toggle output compare pin on overflow feature enabled.

15.3.2.8 Timer Control Register 1/Timer Control Register 2 (TCTL1/TCTL2)

Module Base + 0x0008

	7	6	5	4	3	2	1	0
R W	OM7	OL7	OM6	OL6	OM5	OL5	OM4	OL4
Reset	0	0	0	0	0	0	0	0

Figure 15-14. Timer Control Register 1 (TCTL1)

Note: in Table 15-11, the IOS7 and IOSx should be set to 1

IOSx is the register TIOS bit x,

OC7Mx is the register OC7M bit x,

TCx is timer Input Capture/Output Compare register,

IOCx is channel x,

OMx/OLx is the register TCTL1/TCTL2,

OC7Dx is the register OC7D bit x.

IOCx = OC7Dx+ OMx/OLx, means that both OC7 event and OCx event will change channel x value.

18.4.1.3.3 Sector Erase Command

The sector erase operation will erase all addresses in a 512 byte sector of the Flash array using an embedded algorithm.

An example flow to execute the sector erase operation is shown in Figure 18-24. The sector erase command write sequence is as follows:

1. Write to a Flash array address to start the command write sequence for the sector erase command. The Flash address written determines the sector to be erased while MCU address bits [8:0] and the data written are ignored.
2. Write the sector erase command, 0x40, to the FCMD register.
3. Clear the CBEIF flag in the FSTAT register by writing a 1 to CBEIF to launch the sector erase command.

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

20.3.2.1 Flash Clock Divider Register (FCLKDIV)

The FCLKDIV register is used to control timed events in program and erase algorithms.

Module Base + 0x0000

	7	6	5	4	3	2	1	0
R	FDIVLD	PRDIV8	FDIV5	FDIV4	FDIV3	FDIV2	FDIV1	FDIV0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 20-6. Flash Clock Divider Register (FCLKDIV)

All bits in the FCLKDIV register are readable, bits 6–0 are write once and bit 7 is not writable.

Table 20-4. FCLKDIV Field Descriptions

Field	Description
7 FDIVLD	**Clock Divider Loaded** 0 FCLKDIV register has not been written 1 FCLKDIV register has been written to since the last reset
6 PRDIV8	**Enable Prescalar by 8** 0 The oscillator clock is directly fed into the Flash clock divider 1 The oscillator clock is divided by 8 before feeding into the Flash clock divider
5–0 FDIV[5:0]	**Clock Divider Bits** — The combination of PRDIV8 and FDIV[5:0] must divide the oscillator clock down to a frequency of 150 kHz – 200 kHz. The maximum divide ratio is 512. Refer to Section 20.4.1.1, “Writing the FCLKDIV Register” for more information.

20.3.2.2 Flash Security Register (FSEC)

The FSEC register holds all bits associated with the security of the MCU and Flash module.

Module Base + 0x0001

	7	6	5	4	3	2	1	0
R	KEYEN1	KEYEN0	NV5	NV4	NV3	NV2	SEC1	SEC0
W								
Reset	F	F	F	F	F	F	F	F

= Unimplemented or Reserved

Figure 20-7. Flash Security Register (FSEC)

All bits in the FSEC register are readable but not writable.

The FSEC register is loaded from the Flash configuration field at 0xFF0F during the reset sequence, indicated by F in Figure 20-7.

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

= Unimplemented or Reserved

Figure 21-3. Flash Register Summary

1. Intended for factory test purposes only.

Table A-18. NVM Timing Characteristics

Conditions are shown in Table A-4 unless otherwise noted							
Num	**C**	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	D	External Oscillator Clock	f_{NVMOSC}	0.5	—	50(1)	MHz
2	D	Bus frequency for Programming or Erase Operations	f_{NVMBUS}	1	—		MHz
3	D	Operating Frequency	f_{NVMOP}	150	—	200	kHz
4	P	Single Word Programming Time	t_{swpgm}	46(2)	—	74.5(3)	µs
5	D	Flash Burst Programming consecutive word	t_{bwpgm}	20.4[2]	—	31[3]	µs
6	D	Flash Burst Programming Time for 32 Word row	t_{brpgm}	678.4[2]	—	1035.5[3]	µs
6	D	Flash Burst Programming Time for 64 Word row	t_{brpgm}	1331.2[2]	—	2027.5[3]	µs
7	P	Sector Erase Time	t_{era}	20(4)	—	26.7[3]	ms
8	P	Mass Erase Time	t_{mass}	100[4]	—	133[3]	ms
9	D	Blank Check Time Flash per block	t_{check}	11(5)	—	32778(6)	(7)t_{cyc}
9	D	Blank Check Time Flash per block	t_{check}	11(8)	—	65546(9)	[7]t_{cyc}

1. Restrictions for oscillator in crystal mode apply!
2. Minimum Programming times are achieved under maximum NVM operating frequency f_{NVMOP} and maximum bus frequency f_{bus}.
3. Maximum Erase and Programming times are achieved under particular combinations of f_{NVMOP} and bus frequency f bus. Refer to formulae in Sections A.3.1.1 - A.3.1.4 for guidance.
4. Minimum Erase times are achieved under maximum NVM operating frequency f_{NVMOP}
5. Minimum time, if first word in the array is not blank (512 byte sector size).
6. Maximum time to complete check on an erased block (512 byte sector size)
7. Where t_{cyc} is the system bus clock period.
8. Minimum time, if first word in the array is not blank (1024 byte sector size)
9. Maximum time to complete check on an erased block (1024 byte sector size).