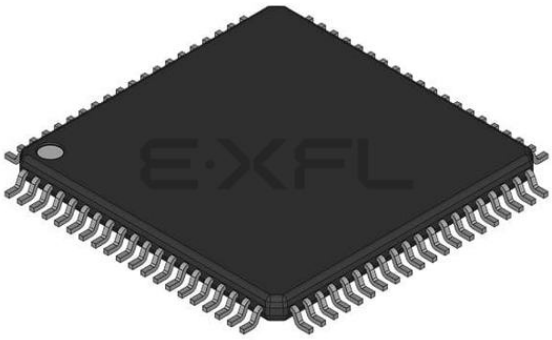

Welcome to E-XFL.COM

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	60
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 ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s12c64mfue

2.3.2.5.3 Port J Data Direction Register (DDRJ)

Module Base + 0x002A

	7	6	5	4	3	2	1	0
R	DDRJ7	DDRJ6	0	0	0	0	0	0
W								
Reset	0	0	—	—	—	—	—	—

= Unimplemented or Reserved

Figure 2-34. Port J Data Direction Register (DDRJ)

Read: Anytime.

Write: Anytime.

Table 2-28. DDRJ Field Descriptions

Field	Description
7–6 DDRJ[7:6]	**Data Direction Port J** — This register configures port pins J[7:6] as either input or output. DDRJ[7:6] — Data Direction Port J 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 PTJ or PTIJ registers, when changing the DDRJ register.

2.3.2.5.4 Port J Reduced Drive Register (RDRJ)

Module Base + 0x002B

	7	6	5	4	3	2	1	0
R	RDRJ7	RDRJ6	0	0	0	0	0	0
W								
Reset	0	0	—	—	—	—	—	—

= Unimplemented or Reserved

Figure 2-35. Port J Reduced Drive Register (RDRJ)

Read: Anytime.

Write: Anytime.

Table 2-29. RDRJ Field Descriptions

Field	Description
7–6 RDRJ[7:6]	**Reduced Drive Port J** — This register configures the drive strength of each port J 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.

2.4 Functional Description

Each pin can act as general purpose I/O. In addition the pin can act as an output from a peripheral module or an input to a peripheral module.

A set of configuration registers is common to all ports. All registers can be written at any time, however a specific configuration might not become active.

Example: Selecting a pull-up resistor. This resistor does not become active while the port is used as a push-pull output.

2.4.1 Registers

2.4.1.1 I/O Register

This register holds the value driven out to the pin if the port is used as a general purpose I/O. Writing to this register has only an effect on the pin if the port is used as general purpose output. When reading this address, the value of the pins are returned if the data direction register bits are set to 0.

If the data direction register bits are set to 1, the contents of the I/O register is returned. This is independent of any other configuration (Figure 2-46).

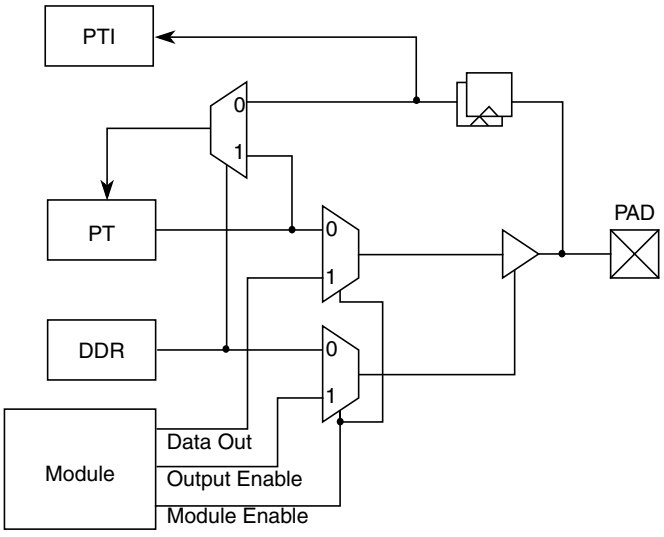

Figure 2-46. Illustration of I/O Pin Functionality

2.4.1.2 Input Register

This is a read-only register and always returns the value of the pin (Figure 2-46).

2.4.1.3 Data Direction Register

This register defines whether the pin is used as an input or an output. If a peripheral module controls the pin the contents of the data direction register is ignored (Figure 2-46).

2.4.1.4 Reduced Drive Register

If the port is used as an output the register allows the configuration of the drive strength.

2.4.1.5 Pull Device Enable Register

This register turns on a pull-up or pull-down device. It becomes only active if the pin is used as an input or as a wired-or output.

2.4.1.6 Polarity Select Register

This register selects either a pull-up or pull-down device if enabled. It becomes only active if the pin is used as an input. A pull-up device can be activated if the pin is used as a wired-OR output.

2.4.2 Port Descriptions

2.4.2.1 Port T

This port is associated with the Standard Capture Timer. PWM output channels can be rerouted from port P to port pins T. In all modes, port T pins can be used for either general-purpose I/O, Standard Capture Timer I/O or as PWM channels module, if so configured by MODRR.

During reset, port T pins are configured as high-impedance inputs.

2.4.2.2 Port S

This port is associated with the serial SCI module. Port S pins PS[3:0] can be used either for general-purpose I/O, or with the SCI subsystem.

During reset, port S pins are configured as inputs with pull-up.

2.4.2.3 Port M

This port is associated with the MSCAN and SPI module. Port M pins PM[5:0] can be used either for general-purpose I/O, with the MSCAN or SPI subsystems.

During reset, port M pins are configured as inputs with pull-up.

2.4.2.4 Port AD

This port is associated with the ATD module. Port AD pins can be used either for general-purpose I/O, or for the ATD subsystem. There are 2 data port registers associated with the Port AD: PTAD[7:0], located in the PIM and PORTAD[7:0] located in the ATD.

To use PTAD[n] as a standard input port, the corresponding DDRD[n] must be cleared. To use PTAD[n] as a standard output port, the corresponding DDRD[n] must be set

NOTE: To use PORTAD[n], located in the ATD as an input port register, DDRD[n] must be cleared and ATDDIEN[n] must be set. *Please refer to ATD Block Guide for details*.

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.

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).

Table 5-3. ITEST Field Descriptions

Field	Description
7:0 INT[E:0]	**Interrupt TEST Bits** — These registers are used in special modes for testing the interrupt logic and priority independent of the system configuration. Each bit is used to force a specific interrupt vector by writing it to a logic 1 state. Bits are named INTE through INT0 to indicate vectors 0xFFxE through 0xFFx0. These bits can be written only in special modes and only with the WRTINT bit set (logic 1) in the interrupt test control register (ITCR). In addition, I interrupts must be masked using the I bit in the CCR. In this state, the interrupt input lines to the interrupt sub-block will be disconnected and interrupt requests will be generated only by this register. These bits can also be read in special modes to view that an interrupt requested by a system block (such as a peripheral block) has reached the INT module. There is a test register implemented for every eight interrupts in the overall system. All of the test registers share the same address and are individually selected using the value stored in the ADR[3:0] bits of the interrupt test control register (ITCR). **Note:** When ADR[3:0] have the value of 0x000F, only bits 2:0 in the ITEST register will be accessible. That is, vectors higher than 0xFFF4 cannot be tested using the test registers and bits 7:3 will always read as a logic 0. If ADR[3:0] point to an unimplemented test register, writes will have no effect and reads will always return a logic 0 value.

5.3.2.3 Highest Priority I Interrupt (Optional)

Module Base + 0x001F
Starting address location affected by INITRG register setting.

	7	6	5	4	3	2	1	0
R	PSEL7	PSEL6	PSEL5	PSEL4	PSEL3	PSEL2	PSEL1	0
W								
Reset	1	1	1	1	0	0	1	0

= Unimplemented or Reserved

Figure 5-4. Highest Priority I Interrupt Register (HPRIO)

Read: Anytime

Write: Only if I mask in CCR = 1

Table 5-4. HPRIO Field Descriptions

Field	Description
7:1 PSEL[7:1]	**Highest Priority I Interrupt Select Bits** — The state of these bits determines which I-bit maskable interrupt will be promoted to highest priority (of the I-bit maskable interrupts). To promote an interrupt, the user writes the least significant byte of the associated interrupt vector address to this register. If an unimplemented vector address or a non I-bit masked vector address (value higher than 0x00F2) is written, IRQ (0xFFF2) will be the default highest priority interrupt.

5.4 Functional Description

The interrupt sub-block processes all exception requests made by the CPU. These exceptions include interrupt vector requests and reset vector requests. Each of these exception types and their overall priority level is discussed in the subsections below.

6.3.2 Register Descriptions

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0xFF00 Reserved	R	X	X	X	X	X	X	0	0
	W								
0xFF01 BDMSTS	R	ENBDM	BDMACT	ENTAG	SDV	TRACE	CLKSW	UNSEC	0
	W								
0xFF02 Reserved	R	X	X	X	X	X	X	X	X
	W								
0xFF03 Reserved	R	X	X	X	X	X	X	X	X
	W								
0xFF04 Reserved	R	X	X	X	X	X	X	X	X
	W								
0xFF05 Reserved	R	X	X	X	X	X	X	X	X
	W								
0xFF06 BDMCCR	R	CCR7	CCR6	CCR5	CCR4	CCR3	CCR2	CCR1	CCR0
	W								
0xFF07 BDMINR	R	0	REG14	REG13	REG12	REG11	0	0	0
	W								
0xFF08 Reserved	R	0	0	0	0	0	0	0	0
	W								
0xFF09 Reserved	R	0	0	0	0	0	0	0	0
	W								
0xFF0A Reserved	R	X	X	X	X	X	X	X	X
	W								
0xFF0B Reserved	R	X	X	X	X	X	X	X	X
	W								

= Unimplemented, Reserved
X = Indeterminate
= Implemented (do not alter)
0 = Always read zero

Figure 6-2. BDM Register Summary

7.4.2.3 Begin- and End-Trigger

The definitions of begin- and end-trigger as used in the DBG module are as follows:

- Begin-trigger: Storage in trace buffer occurs after the trigger and continues until 64 locations are filled.
- End-trigger: Storage in trace buffer occurs until the trigger, with the least recent data falling out of the trace buffer if more than 64 words are collected.

7.4.2.4 Arming the DBG Module

In DBG mode, arming occurs by setting DBGEN and ARM in DBGC1. The ARM bit in DBGC1 is cleared when the trigger condition is met in end-trigger mode or when the Trace Buffer is filled in begin-trigger mode. The TBC logic determines whether a trigger condition has been met based on the trigger mode and the trigger selection.

7.4.2.5 Trigger Modes

The DBG module supports nine trigger modes. The trigger modes are encoded as shown in Table 7-6. The trigger mode is used as a qualifier for either starting or ending the storing of data in the trace buffer. When the match condition is met, the appropriate flag A or B is set in DBGSC. Arming the DBG module clears the A, B, and C flags in DBGSC. In all trigger modes except for the event-only modes and DETAIL capture mode, change-of-flow addresses are stored in the trace buffer. In the event-only modes only the value on the data bus at the trigger event B will be stored. In DETAIL capture mode address and data for all cycles except program fetch (P) and free (f) cycles are stored in trace buffer.

7.4.2.5.1 A Only

In the A only trigger mode, if the match condition for A is met, the A flag in DBGSC is set and a trigger occurs.

7.4.2.5.2 A or B

In the A or B trigger mode, if the match condition for A or B is met, the corresponding flag in DBGSC is set and a trigger occurs.

7.4.2.5.3 A then B

In the A then B trigger mode, the match condition for A must be met before the match condition for B is compared. When the match condition for A or B is met, the corresponding flag in DBGSC is set. The trigger occurs only after A then B have matched.

NOTE

When tagging and using A then B, if addresses A and B are close together, then B may not complete the trigger sequence. This occurs when A and B are in the instruction queue at the same time. Basically the A trigger has not yet occurred, so the B instruction is not tagged. Generally, if address B is at

8.4.2.2 General-Purpose Digital Port Operation

The channel pins can be multiplexed between analog and digital data. As analog inputs, they are multiplexed and sampled to supply signals to the A/D converter. Alternatively they can be configured as digital I/O signals with the port I/O data being held in PORTAD.

The analog/digital multiplex operation is performed in the pads. The pad is always connected to the analog inputs of the ATD10B8C. The pad signal is buffered to the digital port registers. This buffer can be turned on or off with the ATDDIEN register. This is important so that the buffer does not draw excess current when analog potentials are presented at its input.

8.4.2.3 Low-Power Modes

The ATD10B8C can be configured for lower MCU power consumption in three different ways:

1. Stop Mode: This halts A/D conversion. Exit from Stop mode will resume A/D conversion, But due to the recovery time the result of this conversion should be ignored.
2. Wait Mode with AWAI = 1: This halts A/D conversion. Exit from Wait mode will resume A/D conversion, but due to the recovery time the result of this conversion should be ignored.
3. Writing ADPU = 0 (Note that all ATD registers remain accessible.): This aborts any A/D conversion in progress.

NOTE

The reset value for the ADPU bit is zero. Therefore, when this module is reset, it is reset into the power down state.

8.5 Initialization/Application Information

8.5.1 Setting up and starting an A/D conversion

The following describes a typical setup procedure for starting A/D conversions. It is highly recommended to follow this procedure to avoid common mistakes.

Each step of the procedure will have a general remark and a typical example

8.5.1.1 Step 1

Power up the ATD and concurrently define other settings in ATDCTL2
Example: Write to ATDCTL2: ADPU=1 -> powers up the ATD, ASCIE=1 enable interrupt on finish of a conversion sequence.

8.5.1.2 Step 2

Wait for the ATD Recovery Time t_{REC} before you proceed with Step 3.

Example: Use the CPU in a branch loop to wait for a defined number of bus clocks.

10.3.2.10 MSCAN Transmitter Message Abort Acknowledge Register (CANTAAK)

The CANTAAK register indicates the successful abort of a queued message, if requested by the appropriate bits in the CANTARQ register.

Module Base + 0x0009

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	ABTAK2	ABTAK1	ABTAK0
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 10-13. MSCAN Transmitter Message Abort Acknowledge Register (CANTAAK)

NOTE

The CANTAAK register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK = 1).

Read: Anytime
Write: Unimplemented for ABTAKx flags

Table 10-14. CANTAAK Register Field Descriptions

Field	Description
2:0 ABTAK[2:0]	**Abort Acknowledge** — This flag acknowledges that a message was aborted due to a pending abort request from the CPU. After a particular message buffer is flagged empty, this flag can be used by the application software to identify whether the message was aborted successfully or was sent anyway. The ABTAKx flag is cleared whenever the corresponding TXE flag is cleared. 0 The message was not aborted. 1 The message was aborted.

Module Base + 0x000E

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

Figure 12-17. PWM Channel Counter Registers (PWMCNT2)

Module Base + 0x000F

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

Figure 12-18. PWM Channel Counter Registers (PWMCNT3)

Module Base + 0x00010

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

Figure 12-19. PWM Channel Counter Registers (PWMCNT4)

Module Base + 0x00011

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

Figure 12-20. PWM Channel Counter Registers (PWMCNT5)

Read: anytime

Write: anytime (any value written causes PWM counter to be reset to 0x0000).

12.3.2.13 PWM Channel Period Registers (PWMPERx)

There is a dedicated period register for each channel. The value in this register determines the period of the associated PWM channel.

The period registers for each channel are double buffered so that if they change while the channel is enabled, the change will NOT take effect until one of the following occurs:

- The effective period ends
- The counter is written (counter resets to 0x0000)

Table 13-3. Loop Functions

LOOPS	RSRC	Function
0	x	Normal operation
1	0	Loop mode with Rx input internally connected to Tx output
1	1	Single-wire mode with Rx input connected to TXD

13.3.2.3 SCI Control Register 2 (SCICR2)

Module Base + 0x_0003

	7	6	5	4	3	2	1	0
R W	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
Reset	0	0	0	0	0	0	0	0

Figure 13-5. SCI Control Register 2 (SCICR2)

Read: Anytime

Write: Anytime

Table 13-4. SCICR2 Field Descriptions

Field	Description
7 TIE	**Transmitter Interrupt Enable Bit** — TIE enables the transmit data register empty flag, TDRE, to generate interrupt requests. 0 TDRE interrupt requests disabled 1 TDRE interrupt requests enabled
6 TCIE	**Transmission Complete Interrupt Enable Bit** — TCIE enables the transmission complete flag, TC, to generate interrupt requests. 0 TC interrupt requests disabled 1 TC interrupt requests enabled
5 RIE	**Receiver Full Interrupt Enable Bit** — RIE enables the receive data register full flag, RDRF, or the overrun flag, OR, to generate interrupt requests. 0 RDRF and OR interrupt requests disabled 1 RDRF and OR interrupt requests enabled
4 ILIE	**Idle Line Interrupt Enable Bit** — ILIE enables the idle line flag, IDLE, to generate interrupt requests. 0 IDLE interrupt requests disabled 1 IDLE interrupt requests enabled
3 TE	**Transmitter Enable Bit** — TE enables the SCI transmitter and configures the TXD pin as being controlled by the SCI. The TE bit can be used to queue an idle preamble. 0 Transmitter disabled 1 Transmitter enabled
2 RE	**Receiver Enable Bit** — RE enables the SCI receiver. 0 Receiver disabled 1 Receiver enabled

13.4 Functional Description

This section provides a complete functional description of the SCI block, detailing the operation of the design from the end user perspective in a number of subsections.

Figure 13-9 shows the structure of the SCI module. The SCI allows full duplex, asynchronous, NRZ serial communication between the CPU and remote devices, including other CPUs. The SCI transmitter and receiver operate independently, although they use the same baud rate generator. The CPU monitors the status of the SCI, writes the data to be transmitted, and processes received data.

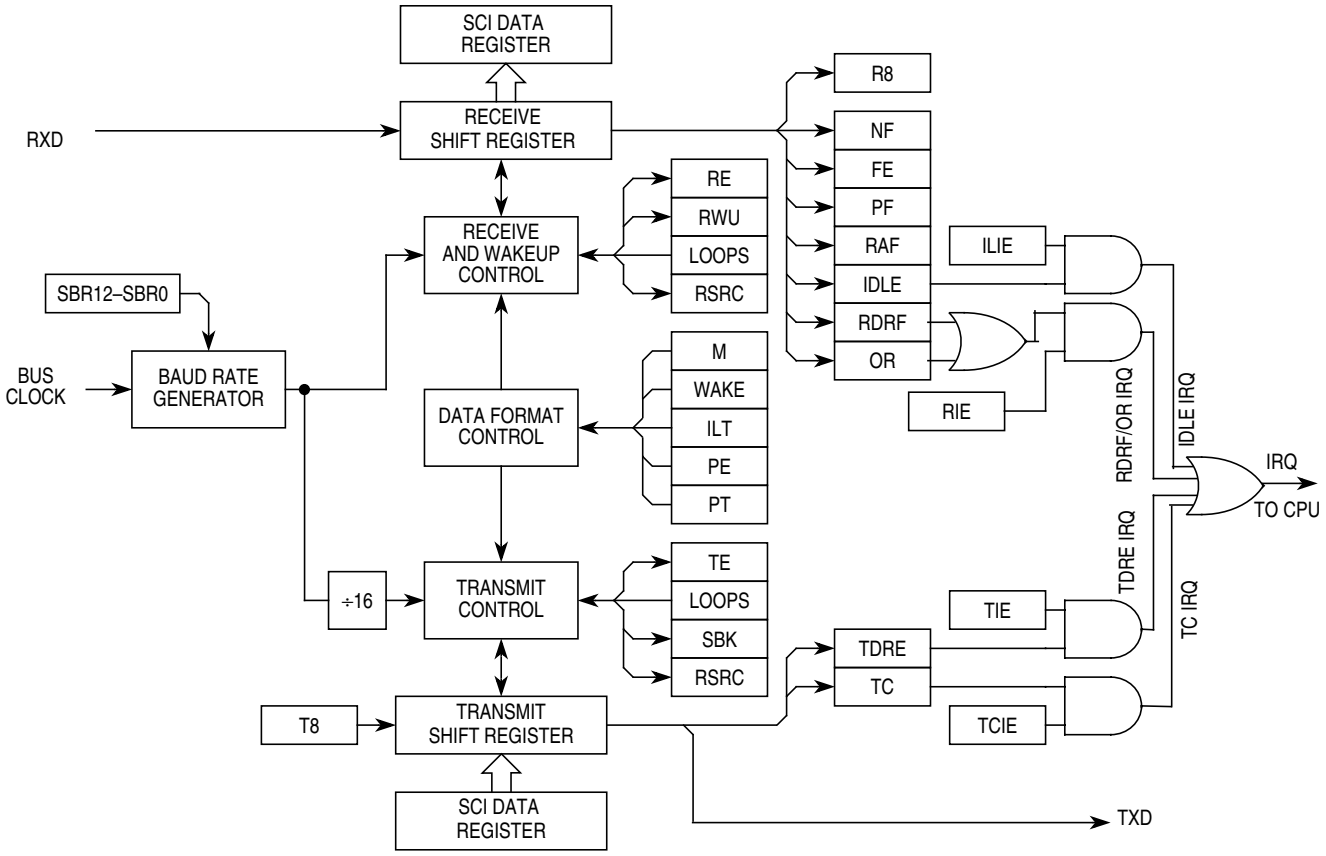

Figure 13-9. SCI Block Diagram

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.

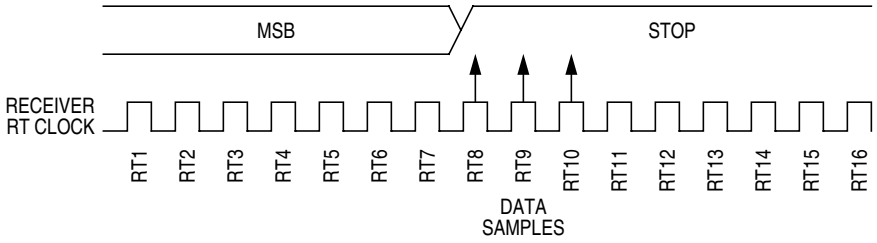

Figure 13-20. Slow Data

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.

Table 14-2. SPICR1 Field Descriptions

Field	Description
7 SPIE	**SPI Interrupt Enable Bit** — This bit enables SPI interrupt requests, if SPIF or MODF status flag is set. 0 SPI interrupts disabled. 1 SPI interrupts enabled.
6 SPE	**SPI System Enable Bit** — This bit enables the SPI system and dedicates the SPI port pins to SPI system functions. If SPE is cleared, SPI is disabled and forced into idle state, status bits in SPISR register are reset. 0 SPI disabled (lower power consumption). 1 SPI enabled, port pins are dedicated to SPI functions.
5 SPTIE	**SPI Transmit Interrupt Enable** — This bit enables SPI interrupt requests, if SPTEF flag is set. 0 SPTEF interrupt disabled. 1 SPTEF interrupt enabled.
4 MSTR	**SPI Master/Slave Mode Select Bit** — This bit selects, if the SPI operates in master or slave mode. Switching the SPI from master to slave or vice versa forces the SPI system into idle state. 0 SPI is in slave mode 1 SPI is in master mode
3 CPOL	**SPI Clock Polarity Bit** — This bit selects an inverted or non-inverted SPI clock. To transmit data between SPI modules, the SPI modules must have identical CPOL values. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 Active-high clocks selected. In idle state SCK is low. 1 Active-low clocks selected. In idle state SCK is high.
2 CPHA	**SPI Clock Phase Bit** — This bit is used to select the SPI clock format. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 Sampling of data occurs at odd edges (1,3,5,...,15) of the SCK clock 1 Sampling of data occurs at even edges (2,4,6,...,16) of the SCK clock
1 SSOE	**Slave Select Output Enable** — The $\overline{SS}$ output feature is enabled only in master mode, if MODFEN is set, by asserting the SSOE as shown in Table 14-3. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state.
0 LSBFE	**LSB-First Enable** — This bit does not affect the position of the MSB and LSB in the data register. Reads and writes of the data register always have the MSB in bit 7. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 Data is transferred most significant bit first. 1 Data is transferred least significant bit first.

Table 14-3. $\overline{SS}$ Input / Output Selection

MODFEN	SSOE	Master Mode	Slave Mode
0	0	$\overline{SS}$ not used by SPI	$\overline{SS}$ input
0	1	$\overline{SS}$ not used by SPI	$\overline{SS}$ input
1	0	$\overline{SS}$ input with MODF feature	$\overline{SS}$ input
1	1	$\overline{SS}$ is slave select output	$\overline{SS}$ input

14.4.7 Operation in Run Mode

In run mode with the SPI system enable (SPE) bit in the SPI control register clear, the SPI system is in a low-power, disabled state. SPI registers remain accessible, but clocks to the core of this module are disabled.

14.4.8 Operation in Wait Mode

SPI operation in wait mode depends upon the state of the SPISWAI bit in SPI Control Register 2.

- If SPISWAI is clear, the SPI operates normally when the CPU is in wait mode
- If SPISWAI is set, SPI clock generation ceases and the SPI module enters a power conservation state when the CPU is in wait mode.
  - If SPISWAI is set and the SPI is configured for master, any transmission and reception in progress stops at wait mode entry. The transmission and reception resumes when the SPI exits wait mode.
  - If SPISWAI is set and the SPI is configured as a slave, any transmission and reception in progress continues if the SCK continues to be driven from the master. This keeps the slave synchronized to the master and the SCK.

    If the master transmits several bytes while the slave is in wait mode, the slave will continue to send out bytes consistent with the operation mode at the start of wait mode (i.e. If the slave is currently sending its SPIDR to the master, it will continue to send the same byte. Else if the slave is currently sending the last received byte from the master, it will continue to send each previous master byte).

NOTE

Care must be taken when expecting data from a master while the slave is in wait or stop mode. Even though the shift register will continue to operate, the rest of the SPI is shut down (i.e. a SPIF interrupt will **not** be generated until exiting stop or wait mode). Also, the byte from the shift register will not be copied into the SPIDR register until after the slave SPI has exited wait or stop mode. A SPIF flag and SPIDR copy is only generated if wait mode is entered or exited during a tranmission. If the slave enters wait mode in idle mode and exits wait mode in idle mode, neither a SPIF nor a SPIDR copy will occur.

14.4.9 Operation in Stop Mode

Stop mode is dependent on the system. The SPI enters stop mode when the module clock is disabled (held high or low). If the SPI is in master mode and exchanging data when the CPU enters stop mode, the transmission is frozen until the CPU exits stop mode. After stop, data to and from the external SPI is exchanged correctly. In slave mode, the SPI will stay synchronized with the master.

The stop mode is not dependent on the SPISWAI bit.

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.

To change the Flash protection that will be loaded on reset, the upper sector of the Flash array must be unprotected, then the Flash protection byte located at Flash address 0xFF0D must be written to.

A protected Flash sector is disabled by FPHDIS while the size of the protected sector is defined by FPHS[1:0] in the FPROT register.

Trying to alter any of the protected areas will result in a protect violation error and the PVIOL flag will be set in the FSTAT register (see Section 17.3.2.6). A mass erase of the whole Flash array is only possible when protection is fully disabled by setting the FPOPEN and FPHDIS bits. An attempt to mass erase a Flash array while protection is enabled will set the PVIOL flag in the FSTAT register.

Table 17-8. FPROT Field Descriptions

Field	Description
7 FPOPEN	**Protection Function for Program or Erase** — The FPOPEN bit is used to either select an address range to be protected using the FPHDIS and FPHS[1:0] bits or to select the same address range to be unprotected as shown in Table 17-9. 0 The FPHDIS bit allows a Flash address range to be unprotected 1 The FPHDIS bit allows a Flash address range to be protected
6 NV6	**Nonvolatile Flag Bit** — The NV6 bit should remain in the erased state for future enhancements.
5 FPHDIS	**Flash Protection Higher Address Range Disable** — The FPHDIS bit determines whether there is a protected/unprotected area in the higher space of the Flash address map. 0 Protection/unprotection enabled 1 Protection/unprotection disabled
4–3 FPHS[1:0]	**Flash Protection Higher Address Size** — The FPHS[1:0] bits determine the size of the protected/unprotected sector as shown in Table 17-10. The FPHS[1:0] bits can only be written to while the FPHDIS bit is set.
2–0 NV[2:0]	**Nonvolatile Flag Bits** — The NV[2:0] bits should remain in the erased state for future enhancements.

Table 17-9. Flash Protection Function

FPOPEN	FPHDIS	FPHS1	FPHS0	Function(1)
1	1	x	x	No protection
1	0	x	x	Protect high range
0	1	x	x	Full Flash array protected
0	0	x	x	Unprotected high range

1. For range sizes refer to Table 17-10.

Table 17-10. Flash Protection Higher Address Range

FPHS[1:0]	Address Range	Range Size
00	0xF800–0xFFFF	2 Kbytes
01	0xF000–0xFFFF	4 Kbytes
10	0xE000–0xFFFF	8 Kbytes
11	0xC000–0xFFFF	16 Kbytes

19.4.1.3.1 Erase Verify Command

The erase verify operation will verify that a Flash array is erased.

An example flow to execute the erase verify operation is shown in Figure 19-25. The erase verify command write sequence is as follows:

1. Write to a Flash array address to start the command write sequence for the erase verify command. The address and data written will be ignored.
2. Write the erase verify command, 0x05, to the FCMD register.
3. Clear the CBEIF flag in the FSTAT register by writing a 1 to CBEIF to launch the erase verify command.

After launching the erase verify command, the CCIF flag in the FSTAT register will set after the operation has completed unless a new command write sequence has been buffered. Upon completion of the erase verify operation, the BLANK flag in the FSTAT register will be set if all addresses in the Flash array are verified to be erased. If any address in the Flash array is not erased, the erase verify operation will terminate and the BLANK flag in the FSTAT register will remain clear.

Table A-5. Thermal Package Characteristics[1]

Num	C	Rating	Symbol	Min	Typ	Max	Unit
1	T	Thermal Resistance LQFP48, single layer PCB[2]	θ_{JA}	—	—	69	°C/W
2	T	Thermal Resistance LQFP48, double sided PCB with 2 internal planes[3]	θ_{JA}	—	—	53	°C/W
3	T	Junction to Board LQFP48	θ_{JB}	—	—	30	°C/W
4	T	Junction to Case LQFP48	θ_{JC}	—	—	20	°C/W
5	T	Junction to Package Top LQFP48	Ψ_{JT}	—	—	4	°C/W
6	T	Thermal Resistance LQFP52, single sided PCB	θ_{JA}	—	—	65	°C/W
7	T	Thermal Resistance LQFP52, double sided PCB with 2 internal planes	θ_{JA}	—	—	49	°C/W
8	T	Junction to Board LQFP52	θ_{JB}	—	—	31	°C/W
9	T	Junction to Case LQFP52	θ_{JC}	—	—	17	°C/W
10	T	Junction to Package Top LQFP52	Ψ_{JT}	—	—	3	°C/W
11	T	Thermal Resistance QFP 80, single sided PCB	θ_{JA}	—	—	52	°C/W
12	T	Thermal Resistance QFP 80, double sided PCB with 2 internal planes	θ_{JA}	—	—	42	°C/W
13	T	Junction to Board QFP80	θ_{JB}	—	—	28	°C/W
14	T	Junction to Case QFP80	θ_{JC}	—	—	18	°C/W
15	T	Junction to Package Top QFP80	Ψ_{JT}	—	—	4	°C/W

1. The values for thermal resistance are achieved by package simulations
2. PC Board according to EIA/JEDEC Standard 51-2
3. PC Board according to EIA/JEDEC Standard 51-7