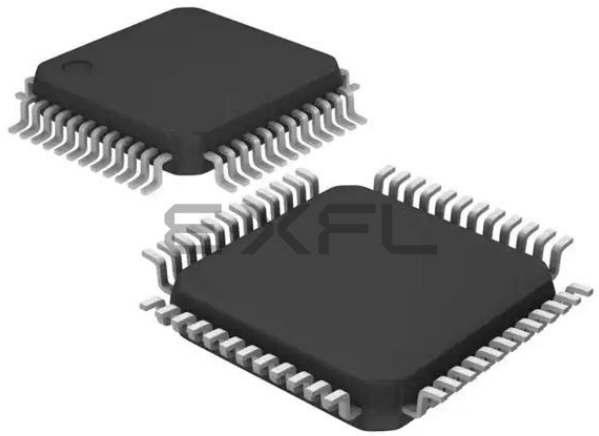


Welcome to E-XFL.COM

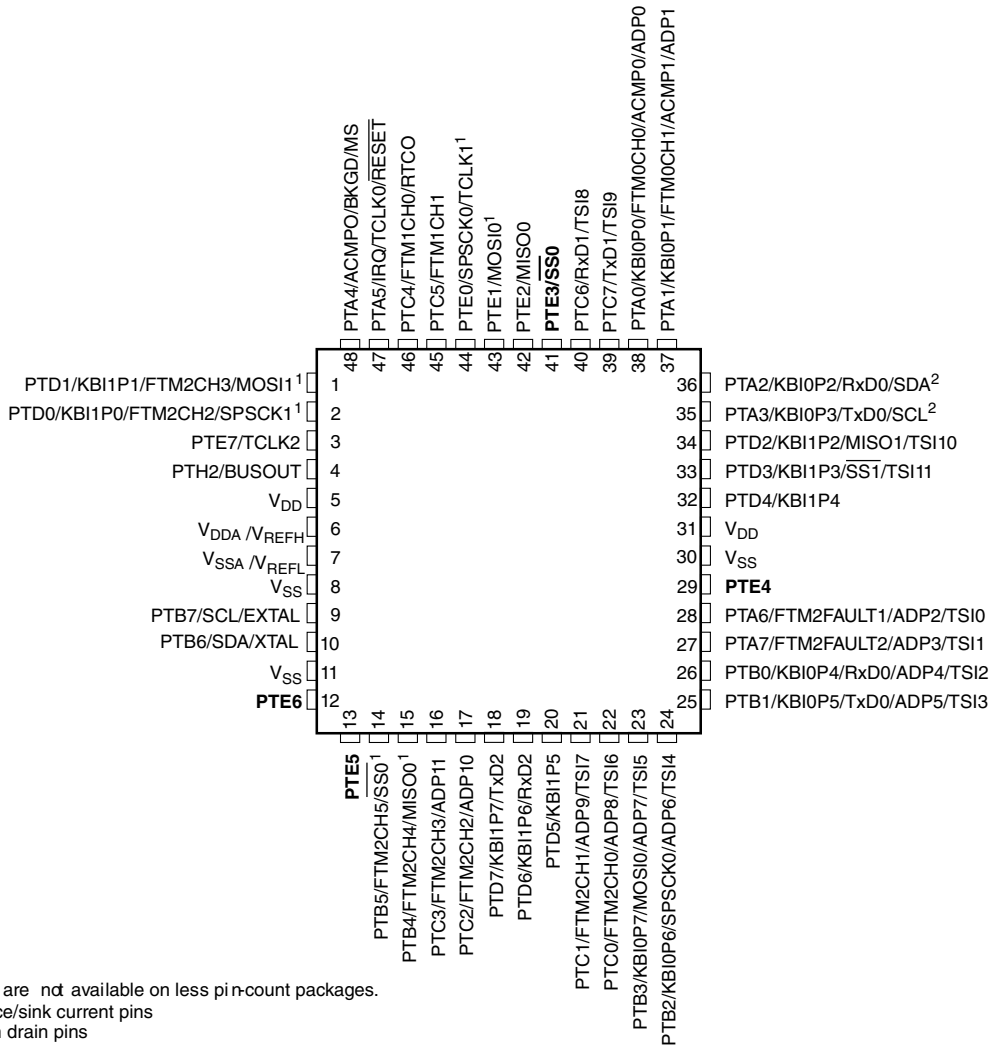
What is "Embedded - Microcontrollers"?

"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	Not For New Designs
Core Processor	S08
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	LVD, POR, PWM, WDT
Number of I/O	41
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 16x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08pt60vlf

12.4.3.3 Free running counter 341
12.4.3.4 Counter reset 342
12.4.4 Input capture mode 342
12.4.4.1 Filter for input capture mode 343
12.4.5 Output compare mode 344
12.4.6 Edge-aligned PWM (EPWM) mode 346
12.4.7 Center-aligned PWM (CPWM) mode 348
12.4.8 Combine mode 350
12.4.8.1 Asymmetrical PWM 357
12.4.9 Complementary mode 357
12.4.10 Update of the registers with write buffers 358
12.4.10.1 CNTINH:L registers 358
12.4.10.2 MODH:L registers 358
12.4.10.3 CnVH:L registers 359
12.4.11 PWM synchronization 360
12.4.11.1 Hardware trigger 360
12.4.11.2 Software trigger 361
12.4.11.3 Boundary cycle 362
12.4.11.4 MODH:L registers synchronization 363
12.4.11.5 CnVH:L registers synchronization 365
12.4.11.6 OUTMASK register synchronization 365
12.4.11.7 FTM counter synchronization 367
12.4.11.8 Summary of PWM synchronization 369
12.4.12 Deadtime insertion 371
12.4.12.1 Deadtime insertion corner cases 372
12.4.13 Output mask 373
12.4.14 Fault control 374
12.4.14.1 Automatic fault clearing 376
12.4.14.2 Manual fault clearing 377



Pins in **bold** are not available on less pin-count packages.
1. High source/sink current pins
2. True open drain pins

Figure 2-2. MC9S08PT60 48-pin LQFP package

Table 3-1. Low power mode behavior (continued)

Mode	Run	Wait	Stop3
RAM	On	Standby	Standby
ADC	On	On	Optional on
ACMP	On	On	Optional on
TSI	On	On	Optional on
I/O	On	On	States held
SCI	On	On	Standby
SPI	On	On	Standby
IIC	On	On	Standby
FTM	On	On	Standby
MTIM	On	On	Standby
WDOG	On	On	Optional on
DBG	On	On	Standby
IPC	On	On	Standby
CRC	On	On	Standby
RTC	On	On	Optional on
LVD	On	On	Optional on

3.3 Low voltage detect (LVD) system

This device includes a system to protect against low voltage conditions in order to protect memory contents and control MCU system states during supply voltage variations. This system consists of a power-on reset (POR) circuit and an LVD circuit with a user selectable trip voltage, either high (V_{LVDH}) or low (V_{LVDL}). The LVD circuit is enabled when SPMSC1[LVDE] is set and the trip voltage is selected by SPMSC2[LVDV]. The LVD is disabled upon entering the stop modes unless the SPMSC1[LVDSE] bit is set or active BDM enabled (BDCSCR[ENBDM]=1). If SPMSC1[LVDSE] and SPMSC1[LVDE] are both set, the current consumption in stop3 with the LVD enabled will be greater.

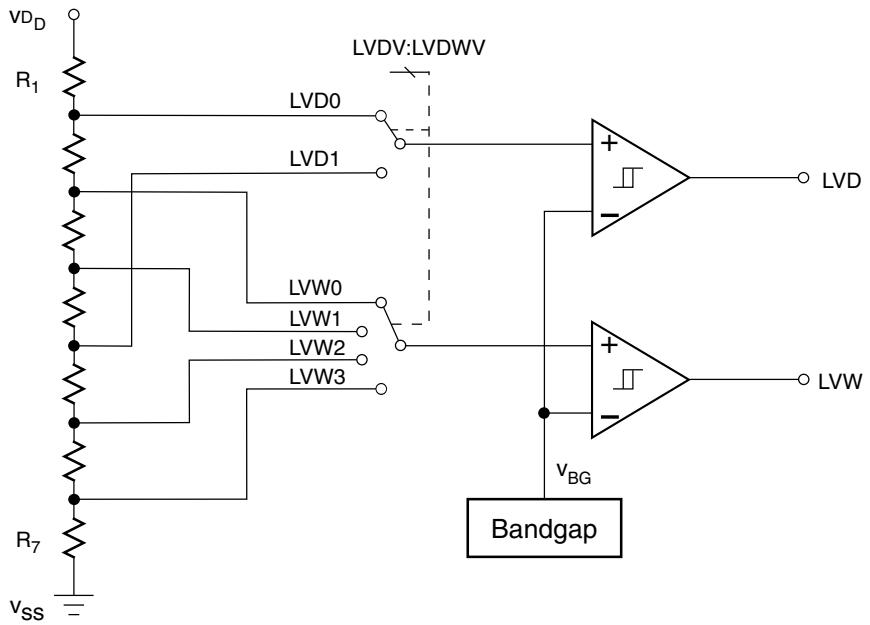


Figure 3-1. Low voltage detect (LVD) block diagram

3.3.1 Power-on reset (POR) operation

When power is initially applied to the MCU, or when the supply voltage drops below the V_{POR} level, the POR circuit will cause a reset condition. As the supply voltage rises, the LVD circuit will hold the chip in reset until the supply has risen above the V_{LVDL} level. Both the SRS[POR] and SRS[LVD] are set following a POR.

3.3.2 LVD reset operation

The LVD can be configured to generate a reset upon detection of a low voltage condition by setting SPMSC1[LVDRE] to 1. After an LVD reset has occurred, the LVD system will hold the MCU in reset until the supply voltage has risen above the level determined by LVDV. The SRS[LVD] bit is set following either an LVD reset or POR.

3.3.3 Low-voltage warning (LVW)

The LVD system has a low voltage warning flag to indicate that the supply voltage is approaching the LVD voltage. When a low voltage condition is detected and the LVD circuit is configured for interrupt operation (SPMSC1[LVDE] set, SPMSC1[LVWIE] set), SPMSC1[LVWF] will be set and LVW interrupt will occur. There are four user-selectable trip voltages for the LVW upon each LVDV configuration. The trip voltage is selected by SPMSC2[LVWV].

IPC_SC field descriptions (continued)

Field	Description
	IPMPS register. Writing IPM with PULIPM setting when IPCE is already set, the IPM will restore the value pulled from the IPMPS register, not the value written to the IPM register.

5.4.2 Interrupt Priority Mask Pseudo Stack Register (IPC_IPMPS)

This register is used to store the previous interrupt priority mask level temporarily when the currently active interrupt is executed.

Address: 3Eh base + 1h offset = 3Fh

Bit	7	6	5	4	3	2	1	0
Read	IPM3		IPM2		IPM1		IPM0	
Write								
Reset	0	0	0	0	0	0	0	0

IPC_IPMPS field descriptions

Field	Description
7–6 IPM3	Interrupt Priority Mask pseudo stack position 3 This field is the pseudo stack register for IPM3. The most recent information is stored in IPM3.
5–4 IPM2	Interrupt Priority Mask pseudo stack position 2 This field is the pseudo stack register for IPM2. The most recent information is stored in IPM2.
3–2 IPM1	Interrupt Priority Mask pseudo stack position 1 This field is the pseudo stack register for IPM1. The most recent information is stored in IPM1.
IPM0	Interrupt Priority Mask pseudo stack position 0 This field is the pseudo stack register for IPM0. The most recent information is stored in IPM0.

5.4.3 Interrupt Level Setting Registers n (IPC_ILRS*n*)

This set of registers (ILRS0-ILRS9) contains the user specified interrupt level for each interrupt source, and indicates the number of the register (ILRSn is ILRS0 through ILRS9).

Address: 3Eh base + 3012h offset + (1d × i), where i=0d to 9d

Bit	7	6	5	4	3	2	1	0
Read Write	ILRn3		ILRn2		ILRn1		ILRn0	
Reset	0	0	0	0	0	0	0	0

When a shared analog function is enabled for a pin, all digital pin functions are disabled. A read of the port data register returns a value of 0 for any bits that have shared analog functions enabled. In general, whenever a pin is shared with both an alternate digital function and an analog function, the analog function has priority such that if both of the digital and analog functions are enabled, the analog function controls the pin.

A write of valid data to a port data register must occur before setting the output enable bit of an associated port pin. This ensures that the pin will not be driven with an incorrect data value.

7.3 Internal pullup enable

An internal pullup device can be enabled for each port pin by setting the corresponding bit in one of the pullup enable registers (PTxPEn). The internal pullup device is disabled if the pin is configured as an output by the parallel I/O control logic, or by any shared peripheral function, regardless of the state of the corresponding pullup enable register bit. The internal pullup device is also disabled if the pin is controlled by an analog function.

NOTE

When configuring IIC to use "SDA(PTA2) and SCL(PTA3)" pins, and if an application uses internal pullups instead of external pullups, the internal pullups remain present setting when the pins are configured as outputs, but they are automatically disabled to save power when the output values are low.

7.4 Input glitch filter setting

A filter is implemented for each port pin that is configured as a digital input. It can be used as a simple low-pass filter to filter any glitch that is introduced from the pins of GPIO, IRQ,$\overline{\text{RESET}}$, and KBI. The glitch width threshold can be adjusted easily by setting registers PORT_IOFLTn and PORT_FCLKDIV between 1~4096 BUSCLKs (or 1~128 LPOCLKs). This configurable glitch filter can take the place of an on board external analog filter, and greatly improve the EMC performance.

Setting register PORT_IOFLTn can configure the filter of the whole port, etc. set PORT_IOFLT0[FLTA] will affect all PTAn pins.

- Writing a larger value slows down the ICSIRCLK frequency.
- Writing a smaller value to the ICSTRM register speeds up the ICSIRCLK frequency.

The TRIM bits affect the ICSOUT frequency if the ICS is in FLL engaged internal (FEI), FLL bypassed internal (FBI), or FLL bypassed internal low power (FBILP) mode.

Until ICSIRCLK is trimmed, programming low reference divider (BDIV) factors may result in ICSOUT frequencies that exceed the maximum chip-level frequency and violate the chip-level clock timing specifications.

If ICS_C1[IREFSTEN] is set and the ICS_C1[IRCLKEN] bit is written to 1, the internal reference clock keeps running during stop mode in order to provide a fast recovery upon exiting stop.

All MCU devices are factory programmed with a trim value in a factory reserved memory location (i.e. reserved nonvolatile information registers that can not be accessed by users). This value is uploaded to the ICS_C3 register and ICS_C4 register during any reset initialization. For finer precision, trim the internal oscillator in the application and set the ICS_C4[SCFTRIM] bit accordingly.

NOTE

Some tools like ProcessorExpert or USB Multilink may use flash memory location, such as 0xFF6F and/or 0xFF6E, to store the temporary trim value.

8.2.1.4 Fixed frequency clock (ICSFFCLK)

The ICS presents the divided FLL reference clock as ICSFFCLK for use as an additional clock source. ICSFFCLK frequency must be no more than 1/4 of the ICSOUT frequency to be valid. When ICSFFCLK is valid, ICS output signal (ICSFFE) gets asserted high. Because of this requirement, in bypass modes the ICSFFCLK is valid only in bypass external modes (FBE and FBELP) for the following combinations of BDIV, RDIV, and RANGE values:

- RANGE=1
- BDIV=000 (divide by 1), RDIV ≥ 010
- BDIV=001 (divide by 2), RDIV ≥ 011
- BDIV=010 (divide by 4), RDIV ≥ 100
- BDIV=011 (divide by 8), RDIV ≥ 101

- ICS_C1[IREFS] bit is written to 0
- BDM mode is not active and ICS_C2[LP] bit is written to 1

In FLL bypassed external low power mode, the ICSOUT clock is derived from the external reference clock source and the FLL is disabled. The ICSLCLK will be not available for BDC communications. The external reference clock source is enabled.

8.2.2.7 Stop (STOP)

In stop mode, the FLL is disabled and the internal clock source can be enabled or disabled. The BDC clock is not available and the ICS does not provide MCU clock source.

Stop mode is entered whenever the MCU enters a stop state. In this mode, all ICS clock signals are static except in the following cases:

- ICSIRCLK will be active in stop mode when all of the following conditions occur:
  - ICS_C1[IRCLKEN] bit is written to 1
  - ICS_C1[IREFSTEN] bit is written to 1
- OSCOUT will be active in stop mode when all of the following conditions occur:
  - ICS_OSCSC[OSCEN] bit is written to 1
  - ICS_OSCSC[OSCSTEN] bit is written to 1

NOTE

The DCO frequency changes from the pre-stop value to its reset value and the FLL need to re-acquire the lock before the frequency is stable. Timing sensitive operations must wait for the FLL acquisition time, t_{Aquire}, before executing.

```
LDA     $F03B
```

This instruction uses extended addressing because $F03B is above the zero page. In most assemblers, the programmer does not need to specify whether an instruction is direct or extended. The assembler automatically selects the shortest form of the instruction.

10.3.6 Indexed Addressing Mode

Indexed addressing mode has seven variations, including five that use the 16-bit H:X index register pair and two that use the stack pointer as the base reference.

10.3.6.1 Indexed, No Offset (IX)

Instructions using the indexed, no offset addressing mode are one-byte instructions that can access data with variable addresses. The X (Index register low byte) register contains the low byte of the conditional address of the operand and the H (Index register high byte) register contains the high byte of the address.

Indexed, no offset instructions can move a pointer through a table or hold the address of a frequently used RAM or input/output (I/O) location.

10.3.6.2 Indexed, No Offset with Post Increment (IX+)

Instructions using the indexed, no offset with post increment addressing mode are two-byte instructions that address the operands and then increment the Index register (H:X). The X (Index register low byte) register contains the low byte of the conditional address of the operand and the H (Index register high byte) register contains the high byte of the address. This addressing mode is usually used for table searches. MOV and CBEQ instructions use this addressing mode as well.

10.3.6.3 Indexed, 8-Bit Offset (IX1)

Indexed with 8-bit offset instructions are two-byte instructions that can access data with a variable address. The CPU adds the unsigned bytes in the H:X register to the unsigned byte immediately following the opcode. The sum is the effective address.

Indexed, 8-bit offset instructions are useful in selecting the k-th element in an n-element table. The table can begin anywhere and can extend as far as the address map allows. The k value would typically be in H:X, and the address of the beginning of the table would be

in the byte following the opcode. Using H:X in this way, this addressing mode is limited to the first 256 addresses in memory. Tables can be located anywhere in the address map when H:X is used as the base address, and the byte following the opcode is the offset.

10.3.6.4 Indexed, 8-Bit Offset with Post Increment (IX1+)

Indexed, 8-bit offset with post-increment instructions are three-byte instructions that access the operands with variable addresses, then increment H:X. The CPU adds the unsigned bytes in the H:X register to the byte immediately following the opcode. The sum is the effective address. This addressing mode is generally used for table searches. This addressing mode is used for CBEQ instruction.

10.3.6.5 Indexed, 16-Bit Offset (IX2)

Indexed, 16-bit offset instructions are three-byte instructions that can access data with variable addresses at any location in memory. The CPU adds the unsigned contents of H:X to the 16-bit unsigned word formed by the two bytes following the opcode. The sum is the effective address of the operand. The first byte after the opcode is the most significant byte of the 16-bit offset; the second byte is the least significant byte of the 16-bit offset. As with direct and extended addressing, most assemblers determine the shortest form of indexed addressing.

Indexed, 16-bit offset instructions are useful in selecting the k-th element in an n-element table. The table can begin anywhere and can extend as far as the address map allows. The k value would typically be in H:X, and the address of the beginning of the table would be in the bytes following the opcode.

10.3.6.6 SP-Relative, 8-Bit Offset (SP1)

Stack pointer, 8-bit offset instructions are three-byte instructions that address operands in much the same way as indexed 8-bit offset instructions, except that the 8-bit offset is added to the value of the stack pointer instead of the index register.

The stack pointer, 8-bit offset addressing mode permits easy addressing of data on the stack. The CPU adds the unsigned byte in the 16-bit stack pointer (SP) register to the unsigned byte following the opcode. The sum is the effective address of the operand. If interrupts are disabled, this addressing mode allows the stack pointer to be used as a second "index" register.

12.3.20 External Trigger (FTMx_EXTTRIG)

This register indicates when a channel trigger was generated, enables the generation of a trigger when the FTM counter is equal to its initial value, and selects which channels are used in the generation of the channel triggers. Several FTM channels can be selected to generate multiple triggers in one PWM period.

Channels 6 and 7 are not used to generate channel triggers.

Address: Base address + 23h offset

Bit	7	6	5	4	3	2	1	0
Read Write	TRIGF	INITTRIGEN	CH1TRIG	CH0TRIG	CH5TRIG	CH4TRIG	CH3TRIG	CH2TRIG
Reset	0	0	0	0	0	0	0	0

FTMx_EXTTRIG field descriptions

Field	Description
7 TRIGF	Channel Trigger Flag Set by hardware when a channel trigger is generated. Clear TRIGF by reading EXTTRIG while TRIGF is set and then writing a 0 to TRIGF. Writing a 1 to TRIGF has no effect. If another channel trigger is generated before the clearing sequence is completed, the sequence is reset so TRIGF remains set after the clear sequence is completed for the earlier TRIGF. 0 No channel trigger was generated. 1 A channel trigger was generated.
6 INITTRIGEN	Initialization Trigger Enable Enables the generation of the trigger when the FTM counter is equal to its initial value. 0 The generation of initialization trigger is disabled. 1 The generation of initialization trigger is enabled.
5 CH1TRIG	Channel 1 Trigger Enable Enables the generation of the channel trigger when the FTM counter is equal to the CV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
4 CH0TRIG	Channel 0 Trigger Enable Enables the generation of the channel trigger when the FTM counter is equal to the CV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
3 CH5TRIG	Channel 5 Trigger Enable Enables the generation of the channel trigger when the FTM counter is equal to the CV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.

Table continues on the next page...

FTMx_FLTFILTER field descriptions (continued)

Field	Description
	The fault filter is disabled when the value is zero. **NOTE:** Writing to this field has immediate effect and must be done only when the fault control or the fault input is disabled. Failure to do so could result in a missing fault detection.

12.3.25 Fault Input Control (FTMx_FLTCTRL)

This register selects the fault inputs and enables the fault input filter.

Address: Base address + 29h offset

Bit	7	6	5	4	3	2	1	0
Read Write	FFLTR3EN	FFLTR2EN	FFLTR1EN	FFLTR0EN	FAULT3EN	FAULT2EN	FAULT1EN	FAULT0EN
Reset	0	0	0	0	0	0	0	0

FTMx_FLTCTRL field descriptions

Field	Description
7 FFLTR3EN	Fault Input 3 Filter Enable Enables the filter for the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input filter is disabled. 1 Fault input filter is enabled.
6 FFLTR2EN	Fault Input 2 Filter Enable Enables the filter for the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input filter is disabled. 1 Fault input filter is enabled.
5 FFLTR1EN	Fault Input 1 Filter Enable Enables the filter for the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input filter is disabled. 1 Fault input filter is enabled.
4 FFLTR0EN	Fault Input 0 Filter Enable Enables the filter for the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input filter is disabled. 1 Fault input filter is enabled.

Table continues on the next page...

- Loop mode
- Single-wire mode

15.1.3 Block diagram

The following figure shows the transmitter portion of the SCI.

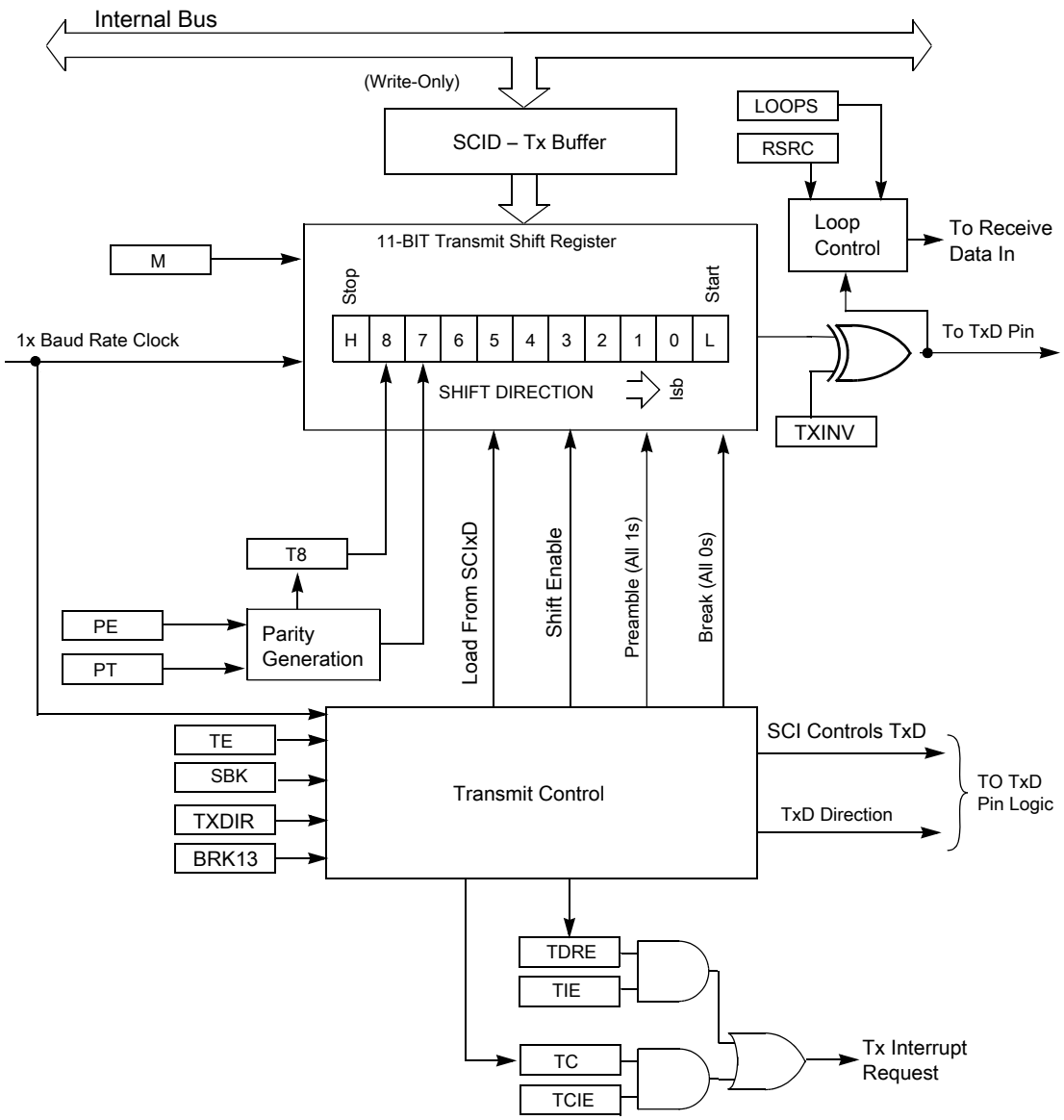


Figure 15-1. SCI transmitter block diagram

The following figure shows the receiver portion of the SCI.

In the external SPI system, simply connect all SPSCK pins to each other, all MISO pins together, and all MOSI pins together. Peripheral devices often use slightly different names for these pins.

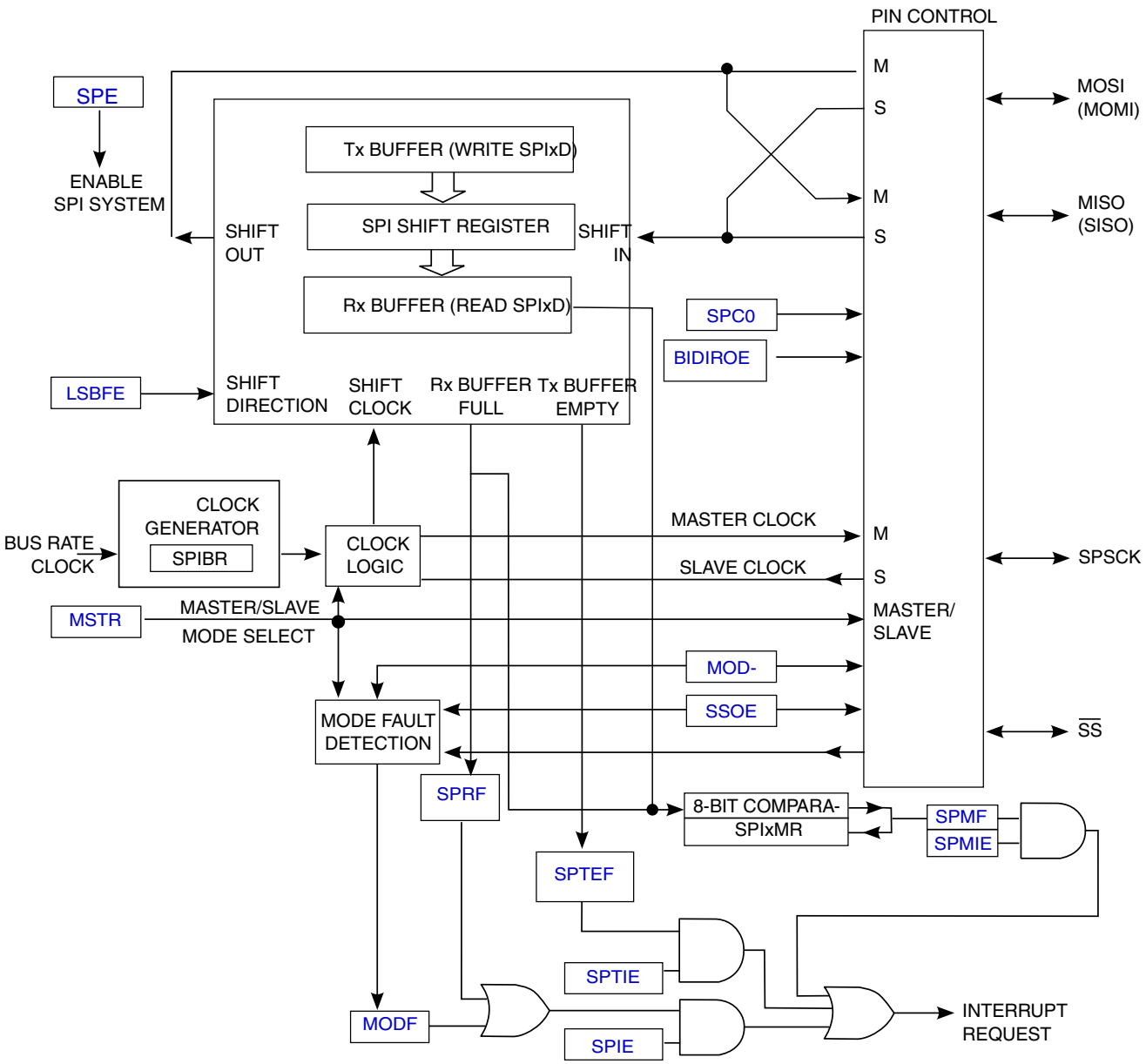


Figure 16-2. SPI Module Block Diagram without FIFO

16.2 External Signal Description

The SPI optionally shares four port pins. The function of these pins depends on the settings of SPI control bits. When the SPI is disabled (SPE = 0), these four pins revert to other functions that are not controlled by the SPI (based on chip configuration).

SPI1_S field descriptions (continued)

Field	Description
5 SPTEF	SPI Transmit Buffer Empty Flag (when FIFO is not supported or not enabled) or SPI transmit FIFO empty flag (when FIFO is supported and enabled) When the FIFO is not supported or not enabled (FIFOMODE is not present or is 0): This bit is set when the transmit data buffer is empty. SPTEF is cleared by reading the S register with SPTEF set and then writing a data value to the transmit buffer at DH:DL. The S register must be read with SPTEF set to 1 before writing data to the DH:DL register; otherwise, the DH:DL write is ignored. SPTEF is automatically set when all data from the transmit buffer transfers into the transmit shift register. For an idle SPI, data written to DH:DL is transferred to the shifter almost immediately so that SPTEF is set within two bus cycles, allowing a second set of data to be queued into the transmit buffer. After completion of the transfer of the data in the shift register, the queued data from the transmit buffer automatically moves to the shifter, and SPTEF is set to indicate that room exists for new data in the transmit buffer. If no new data is waiting in the transmit buffer, SPTEF simply remains set and no data moves from the buffer to the shifter. When the FIFO is supported and enabled (FIFOMODE is 1): This bit provides the status of the FIFO rather than of an 8-bit or a 16-bit buffer. This bit is set when the transmit FIFO is empty. SPTEF is cleared by writing a data value to the transmit FIFO at DH:DL. SPTEF is automatically set when all data from the transmit FIFO transfers into the transmit shift register. For an idle SPI, data written to the DH:DL register is transferred to the shifter almost immediately, so that SPTEF is set within two bus cycles. A second write of data to the DH:DL register clears this SPTEF flag. After completion of the transfer of the data in the shift register, the queued data from the transmit FIFO automatically moves to the shifter, and SPTEF will be set only when all data written to the transmit FIFO has been transfered to the shifter. If no new data is waiting in the transmit FIFO, SPTEF simply remains set and no data moves from the buffer to the shifter. 0 SPI transmit buffer not empty (when FIFOMODE is not present or is 0) or SPI FIFO not empty (when FIFOMODE is 1) 1 SPI transmit buffer empty (when FIFOMODE is not present or is 0) or SPI FIFO empty (when FIFOMODE is 1)
4 MODF	Master Mode Fault Flag MODF is set if the SPI is configured as a master and the slave select input goes low, indicating some other SPI device is also configured as a master. The $\overline{SS}$ pin acts as a mode fault error input only when C1[MSTR] is 1, C2[MODFEN] is 1, and C1[SSOE] is 0; otherwise, MODF will never be set. MODF is cleared by reading MODF while it is 1 and then writing to the SPI Control Register 1 (C1). 0 No mode fault error 1 Mode fault error detected
3 RNFULLF	Receive FIFO nearly full flag This flag is set when more than three 16-bit words or six 8-bit bytes of data remain in the receive FIFO, provided C3[4] is 0, or when more than two 16-bit words or four 8-bit bytes of data remain in the receive FIFO, provided C3[4] is 1. It has no function if FIFOMODE is not present or is 0. 0 Receive FIFO has received less than 48 bits (when C3[4] is 0) or less than 32 bits (when C3[4] is 1) 1 Receive FIFO has received data of an amount equal to or greater than 48 bits (when C3[4] is 0) or 32 bits (when C3[4] is 1)
2 TNEAREF	Transmit FIFO nearly empty flag This flag is set when only one 16-bit word or two 8-bit bytes of data remain in the transmit FIFO, provided C3[5] is 0, or when only two 16-bit words or four 8-bit bytes of data remain in the transmit FIFO, provided C3[5] is 1. If FIFOMODE is not enabled, ignore this bit. **NOTE:** At an initial POR, the values of TNEAREF and RFIFOEF are 0. However, the status (S) register and both TX and RX FIFOs are reset due to a change of SPIMODE, FIFOMODE or SPE. If this

Table continues on the next page...

SPIx_C1=0x54(%01010100)				
Bit 7	SPIE	=	0	Disables receive and mode fault interrupts
Bit 6	SPE	=	1	Enables the SPI system
Bit 5	SPTIE	=	0	Disables SPI transmit interrupts
Bit 4	MSTR	=	1	Sets the SPI module as a master SPI device
Bit 3	CPOL	=	0	Configures SPI clock as active-high
Bit 2	CPHA	=	1	First edge on SPSCK at start of first data transfer cycle
Bit 1	SSOE	=	0	Determines SS pin function when mode fault enabled
Bit 0	LSBFE	=	0	SPI serial data transfers start with most significant bit

SPIx_C2 = 0xC0(%11000000)				
Bit 7	SPMIE	=	1	SPI hardware match interrupt enabled
Bit 6	SPIMODE	=	1	Configures SPI for 16-bit mode
Bit 5		=	0	Reserved
Bit 4	MODFEN	=	0	Disables mode fault function
Bit 3	BIDIROE	=	0	SPI data I/O pin acts as input
Bit 2		=	0	Reserved
Bit 1	SPISWAI	=	0	SPI clocks operate in wait mode
Bit 0	SPC0	=	0	uses separate pins for data input and output

SPIx_BR = 0x00(%00000000)				
Bit 7		=	0	Reserved
Bit 6:4		=	000	Sets prescale divisor to 1
Bit 3:0		=	0000	Sets baud rate divisor to 2

SPIx_S = 0x00(%00000000)				
Bit 7	SPRF	=	0	Flag is set when receive data buffer is full
Bit 6	SPMF	=	0	Flag is set when SPIx_MH/ML = receive data buffer
Bit 5	SPTEF	=	0	Flag is set when transmit data buffer is empty
Bit 4	MODF	=	0	Mode fault flag for master mode
Bit 3:0		=	0	FIFOMODE is not enabled

SPIx_MH = 0xXX
In 16-bit mode, this register holds bits 8–15 of the hardware match buffer. In 8-bit mode, writes to this register will be ignored.

SPIx_ML = 0xXX
Holds bits 0–7 of the hardware match buffer.

Table 19-13. Total conversion time vs. control conditions

Conversion type	ADICLK	ADLSMP	Max total conversion time
Single or first continuous 8-bit	0x, 10	0	20 ADCK cycles + 5 bus clock cycles
Single or first continuous 10-bit or 12-bit	0x, 10	0	23 ADCK cycles + 5 bus clock cycles
Single or first continuous 8-bit	0x, 10	1	40 ADCK cycles + 5 bus clock cycles
Single or first continuous 10-bit or 12-bit	0x, 10	1	43 ADCK cycles + 5 bus clock cycles
Single or first continuous 8-bit	11	0	5 µs + 20 ADCK + 5 bus clock cycles
Single or first continuous 10-bit or 12-bit	11	0	5 µs + 23 ADCK + 5 bus clock cycles
Single or first continuous 8-bit	11	1	5 µs + 40 ADCK + 5 bus clock cycles
Single or first continuous 10-bit or 12-bit	11	1	5 µs + 43 ADCK + 5 bus clock cycles
Subsequent continuous 8-bit; $f_{BUS} > f_{ADCK}$	xx	0	17 ADCK cycles
Subsequent continuous 10-bit or 12-bit; $f_{BUS} > f_{ADCK}$	xx	0	20 ADCK cycles
Subsequent continuous 8-bit; $f_{BUS} > f_{ADCK}/11$	xx	1	37 ADCK cycles
Subsequent continuous 10-bit or 12-bit; $f_{BUS} > f_{ADCK}/11$	xx	1	40 ADCK cycles

The maximum total conversion time is determined by the selected clock source and the divide ratio. The clock source is selectable by the ADC_SC3[ADICLK] bits, and the divide ratio is specified by the ADC_SC3[ADIV] bits. For example, in 10-bit mode, with the bus clock selected as the input clock source, the input clock divide-by-1 ratio selected, and a bus frequency of 8 MHz, then the conversion time for a single conversion as given below:

$$\text{Conversion time} = \frac{23\,\text{ADCK Cyc}}{8\,\text{MHz}/1} + \frac{5\,\text{bus Cyc}}{8\,\text{MHz}} = 3.5\,\mu\text{s}$$

The number of bus cycles at 8 MHz is:

$$\text{Bus cycles} = 3.5\mu\text{s} \times 8\text{MHz} = 28$$

Note

The ADCK frequency must be between f_{ADCK} minimum and f_{ADCK} maximum to meet ADC specifications.

DBG_CBL field descriptions

Field	Description
CB[7:0]	Comparator B Low The Comparator B Low compare bits control whether Comparator B will compare the address bus bits [7:0] to a logic 1 or logic 0. 0 Compare corresponding address bit to a logic 0. 1 Compare corresponding address bit to a logic 1.

25.3.5 Debug Comparator C High Register (DBG_CCH)

NOTE

All the bits in this register reset to 0 in POR or non-end-run reset. The bits are undefined in end-run reset. In the case of an end-trace to reset where DBGEN = 1 and BEGIN = 0, the bits in this register do not change after reset.

Address: 3010h base + 4h offset = 3014h

Bit	7	6	5	4	3	2	1	0
Read Write	CC[15:8]							
Reset	0	0	0	0	0	0	0	0

DBG_CCH field descriptions

Field	Description
CC[15:8]	Comparator C High Compare Bits The Comparator C High compare bits control whether Comparator C will compare the address bus bits [15:8] to a logic 1 or logic 0. 0 Compare corresponding address bit to a logic 0. 1 Compare corresponding address bit to a logic 1.

DBG_FH field descriptions

Field	Description
F[15:8]	FIFO High Data Bits The FIFO High data bits provide access to bits [15:8] of data in the FIFO. This register is not used in event only modes and will read a $00 for valid FIFO words.

25.3.8 Debug FIFO Low Register (DBG_FL)

NOTE

All the bits in this register reset to 0 in POR or non-end-run reset. The bits are undefined in end-run reset. In the case of an end-trace to reset where DBGEN = 1 and BEGIN = 0, the bits in this register do not change after reset.

Address: 3010h base + 7h offset = 3017h

Bit	7	6	5	4	3	2	1	0
Read	F[7:0]							
Write								
Reset	0	0	0	0	0	0	0	0

DBG_FL field descriptions

Field	Description
F[7:0]	FIFO Low Data Bits The FIFO Low data bits contain the least significant byte of data in the FIFO. When reading FIFO words, read DBGFX and DBGFH before reading DBGFL because reading DBGFL causes the FIFO pointers to advance to the next FIFO location. In event-only modes, there is no useful information in DBGFX and DBGFH so it is not necessary to read them before reading DBGFL.

25.3.12 Debug FIFO Extended Information Register (DBG_FX)

NOTE

All the bits in this register reset to 0 in POR or non-end-run reset. The bits are undefined in end-run reset. In the case of an end-trace to reset where DBGEN = 1 and BEGIN = 0, the bits in this register do not change after reset.

Address: 3010h base + Bh offset = 301Bh

Bit	7	6	5	4	3	2	1	0
Read	PPACC	0						Bit16
Write								
Reset	0	0	0	0	0	0	0	0

DBG_FX field descriptions

Field	Description
7 PPACC	PPAGE Access Indicator Bit This bit indicates whether the captured information in the current FIFO word is associated with an extended access through the PPAGE mechanism or not. This is indicated by the internal signal mmu_ppage_sel which is 1 when the access is through the PPAGE mechanism. 0 The information in the corresponding FIFO word is event-only data or an unpaged 17-bit CPU address with bit-16 = 0. 1 The information in the corresponding FIFO word is a 17-bit flash address with PPAGE[2:0] in the three most significant bits and CPU address[13:0] in the 14 least significant bits.
6–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 Bit16	Extended Address Bit 16 This bit is the most significant bit of the 17-bit core address.

25.3.13 Debug Control Register (DBG_C)

Address: 3010h base + Ch offset = 301Ch

Bit	7	6	5	4	3	2	1	0
Read	DBGEN	ARM	TAG	BRKEN	0			LOOP1
Write								
Reset	1	1	0	0	0	0	0	0