

Welcome to [E-XFL.COM](https://www.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	S08
Core Size	8-Bit
Speed	48MHz
Connectivity	I ² C, LINbus, SCI, SPI, USB
Peripherals	LVD, POR, PWM, WDT
Number of I/O	37
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-VFQFN Exposed Pad
Supplier Device Package	48-QFN-EP (7x7)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s08jm60cgt

Revision History

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify you have the latest information available, refer to:

<http://freescale.com/>

The following revision history table summarizes changes contained in this document.

Revision Number	Revision Date	Description of Changes
1	11/27/2007	Initial release
2	3/4/2008	Changed the location of R_S to connect to EXTAL in Figure 2-4 . Changed port rise and fall time in Table A-13 . Added DC injection current and RAM retention voltage in Table A-6 . Deleted note on 625 ns of item 17 in Table A-12 . Moved Bandgap Voltage Reference item from Table A-8 to Table A-6 . Added one paragraph on how to improve accuracy to Section 10.1.1.5 , “Temperature Sensor.”
3	1/21/2009	Changed the V_{TEMP25} from 1.396 mV to 1.396 V in Table A-10 . Complete the EMC data in Section A.15 , “EMC Performance.” Revised the Typo in Table 11-4 .
4	2/21/2010	Changed the location of R_S to connect to XTAL in Figure 2-4 .
5	10/2014	Updated t_{LPO} in Table A-13 .

This product incorporates SuperFlash® technology licensed from SST.

Freescall™ and the Freescall logo are trademarks of Freescall Semiconductor, Inc.
© Freescall Semiconductor, Inc., 2007-2014. All rights reserved.

6.3	Pin Control	82
6.3.1	Internal Pullup Enable	83
6.3.2	Output Slew Rate Control Enable	83
6.3.3	Output Drive Strength Select	83
6.4	Pin Behavior in Stop Modes	83
6.5	Parallel I/O and Pin Control Registers	83
6.5.1	Port A I/O Registers (PTAD and PTADD)	84
6.5.2	Port A Pin Control Registers (PTAPE, PTASE, PTADS)	84
6.5.3	Port B I/O Registers (PTBD and PTBDD)	86
6.5.4	Port B Pin Control Registers (PTBPE, PTBSE, PTBDS)	86
6.5.5	Port C I/O Registers (PTCD and PTCDD)	88
6.5.6	Port C Pin Control Registers (PTCPE, PTCSE, PTCDS)	88
6.5.7	Port D I/O Registers (PTDD and PTDDD)	90
6.5.8	Port D Pin Control Registers (PTDPE, PTDSE, PTDDS)	90
6.5.9	Port E I/O Registers (PTED and PTEDD)	92
6.5.10	Port E Pin Control Registers (PTEPE, PTESE, PTEDS)	92
6.5.11	Port F I/O Registers (PTFD and PTFDD)	94
6.5.12	Port F Pin Control Registers (PTFPE, PTFSE, PTFDS)	94
6.5.13	Port G I/O Registers (PTGD and PTGDD)	96
6.5.14	Port G Pin Control Registers (PTGPE, PTGSE, PTGDS)	96

Chapter 7

Central Processor Unit (S08CPUV2)

7.1	Introduction	99
7.1.1	Features	99
7.2	Programmer's Model and CPU Registers	100
7.2.1	Accumulator (A)	100
7.2.2	Index Register (H:X)	100
7.2.3	Stack Pointer (SP)	101
7.2.4	Program Counter (PC)	101
7.2.5	Condition Code Register (CCR)	101
7.3	Addressing Modes	103
7.3.1	Inherent Addressing Mode (INH)	103
7.3.2	Relative Addressing Mode (REL)	103
7.3.3	Immediate Addressing Mode (IMM)	103
7.3.4	Direct Addressing Mode (DIR)	103
7.3.5	Extended Addressing Mode (EXT)	104
7.3.6	Indexed Addressing Mode	104
7.4	Special Operations	105
7.4.1	Reset Sequence	105
7.4.2	Interrupt Sequence	105
7.4.3	Wait Mode Operation	106
7.4.4	Stop Mode Operation	106
7.4.5	BGND Instruction	107
7.5	HCS08 Instruction Set Summary	108

6.3.1 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 pullup device is disabled if the pin is configured as an output by the parallel I/O control logic or any shared peripheral function regardless of the state of the corresponding pullup enable register bit. The pullup device is also disabled if the pin is controlled by an analog function.

6.3.2 Output Slew Rate Control Enable

Slew rate control can be enabled for each port pin by setting the corresponding bit in one of the slew rate control registers (PTxSEn). When enabled, slew control limits the rate at which an output can transition in order to reduce EMC emissions. Slew rate control has no effect on pins which are configured as inputs.

6.3.3 Output Drive Strength Select

An output pin can be selected to have high output drive strength by setting the corresponding bit in one of the drive strength select registers (PTxDSn). When high drive is selected a pin is capable of sourcing and sinking greater current. Even though every I/O pin can be selected as high drive, the user must ensure that the total current source and sink limits for the chip are not exceeded. Drive strength selection is intended to affect the DC behavior of I/O pins. However, the AC behavior is also affected. High drive allows a pin to drive a greater load with the same switching speed as a low drive enabled pin into a smaller load. Because of this the EMC emissions may be affected by enabling pins as high drive.

6.4 Pin Behavior in Stop Modes

Depending on the stop mode, I/O functions differently as the result of executing a STOP instruction. An explanation of I/O behavior for the various stop modes follows:

- Stop2 mode is a partial power-down mode, whereby I/O latches are maintained in their state as before the STOP instruction was executed. CPU register status and the state of I/O registers must be saved in RAM before the STOP instruction is executed to place the MCU in stop2 mode. Upon recovery from stop2 mode, before accessing any I/O, the user must examine the state of the PPDF bit in the SPMSC2 register. If the PPDF bit is 0, I/O must be initialized as if a power on reset had occurred. If the PPDF bit is 1, I/O data previously stored in RAM, before the STOP instruction was executed, peripherals may require being initialized and restored to their pre-stop condition. The user must then write a 1 to the PPDACK bit in the SPMSC2 register. Access to I/O is now permitted again in the user's application program.
- In stop3 mode, all I/O is maintained because internal logic circuitry stays powered up. Upon recovery, normal I/O function is available to the user.

6.5 Parallel I/O and Pin Control Registers

This section provides information about the registers associated with the parallel I/O ports and pin control functions. These parallel I/O registers are located in page zero of the memory map and the pin control registers are located in the high page register section of memory.

6.5.9 Port E I/O Registers (PTED and PTEDD)

Port E parallel I/O function is controlled by the registers listed below.

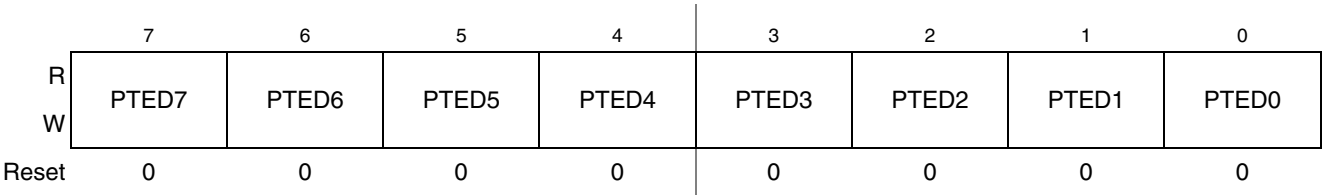


Figure 6-22. Port E Data Register (PTED)

Table 6-21. PTED Register Field Descriptions

Field	Description
7:0 PTED[7:0]	Port E Data Register Bits — For port E pins that are inputs, reads return the logic level on the pin. For port E pins that are configured as outputs, reads return the last value written to this register. Writes are latched into all bits of this register. For port E pins that are configured as outputs, the logic level is driven out the corresponding MCU pin. Reset forces PTED to all 0s, but these 0s are not driven out the corresponding pins because reset also configures all port pins as high-impedance inputs with pullups disabled.

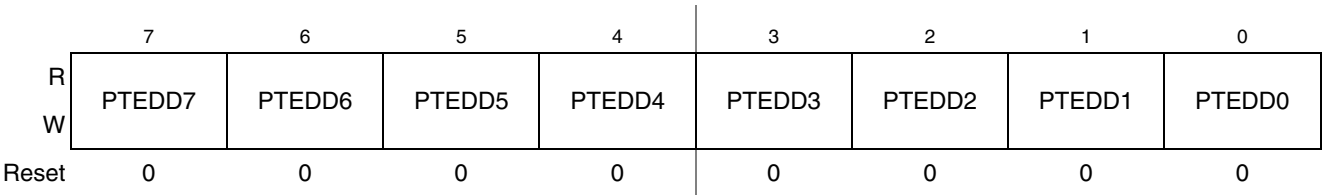


Figure 6-23. Data Direction for Port E (PTEDD)

Table 6-22. PTEDD Register Field Descriptions

Field	Description
7:0 PTEDD[7:0]	Data Direction for Port E Bits — These read/write bits control the direction of port E pins and what is read for PTED reads. 0 Input (output driver disabled) and reads return the pin value. 1 Output driver enabled for port E bit n and PTED reads return the contents of PTEDn.

6.5.10 Port E Pin Control Registers (PTEPE, PTESE, PTEDS)

In addition to the I/O control, port E pins are controlled by the registers listed below.

7.2 Programmer's Model and CPU Registers

Figure 7-1 shows the five CPU registers. CPU registers are not part of the memory map.

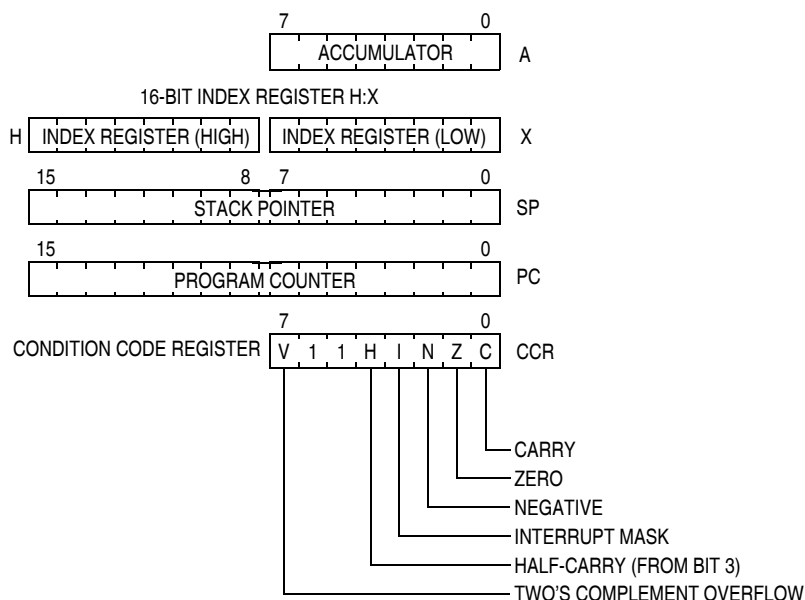


Figure 7-1. CPU Registers

7.2.1 Accumulator (A)

The A accumulator is a general-purpose 8-bit register. One operand input to the arithmetic logic unit (ALU) is connected to the accumulator and the ALU results are often stored into the A accumulator after arithmetic and logical operations. The accumulator can be loaded from memory using various addressing modes to specify the address where the loaded data comes from, or the contents of A can be stored to memory using various addressing modes to specify the address where data from A will be stored.

Reset has no effect on the contents of the A accumulator.

7.2.2 Index Register (H:X)

This 16-bit register is actually two separate 8-bit registers (H and X), which often work together as a 16-bit address pointer where H holds the upper byte of an address and X holds the lower byte of the address. All indexed addressing mode instructions use the full 16-bit value in H:X as an index reference pointer; however, for compatibility with the earlier M68HC05 Family, some instructions operate only on the low-order 8-bit half (X).

Many instructions treat X as a second general-purpose 8-bit register that can be used to hold 8-bit data values. X can be cleared, incremented, decremented, complemented, negated, shifted, or rotated. Transfer instructions allow data to be transferred from A or transferred to A where arithmetic and logical operations can then be performed.

For compatibility with the earlier M68HC05 Family, H is forced to 0x00 during reset. Reset has no effect on the contents of X.

Table 8-2. ACMP Status and Control Register Field Descriptions (continued)

Field	Description
2 ACOPE	Analog Comparator Output Pin Enable — ACOPE is used to enable the comparator output to be placed onto the external pin, ACMPO. 0 Analog comparator output not available on ACMPO 1 Analog comparator output is driven out on ACMPO
1:0 ACMOD	Analog Comparator Mode — ACMOD selects the type of compare event which sets ACF. 00 Encoding 0 — Comparator output falling edge 01 Encoding 1 — Comparator output rising edge 10 Encoding 2 — Comparator output falling edge 11 Encoding 3 — Comparator output rising or falling edge

8.4 Functional Description

The analog comparator can be used to compare two analog input voltages applied to ACMP+ and ACMP–; or it can be used to compare an analog input voltage applied to ACMP– with an internal bandgap reference voltage. ACBGS is used to select between the bandgap reference voltage or the ACMP+ pin as the input to the non-inverting input of the analog comparator. The comparator output is high when the non-inverting input is greater than the inverting input, and is low when the non-inverting input is less than the inverting input. ACMOD is used to select the condition which will cause ACF to be set. ACF can be set on a rising edge of the comparator output, a falling edge of the comparator output, or either a rising or a falling edge (toggle). The comparator output can be read directly through ACO. The comparator output can be driven onto the ACMPO pin using ACOPE.

Chapter 9

Keyboard Interrupt (S08KBIV2)

9.1 Introduction

The MC9S08JM60 series have one KBI module with eight keyboard interrupt inputs. See [Chapter 2, “Pins and Connections,”](#) for more information about the logic and hardware aspects of these pins.

NOTE

MC9S08JM60 series devices operate at a higher voltage range (2.7 V to 5.5 V) and do not include stop1 mode. Therefore, please disregard references to stop1.

If the MODE bits are changed, any data in ADCRH becomes invalid.

	7	6	5	4	3	2	1	0
R	0	0	0	0	ADR11	ADR10	ADR9	ADR8
W								
Reset:	0	0	0	0	0	0	0	0

Figure 10-5. Data Result High Register (ADCRH)

10.3.4 Data Result Low Register (ADCRL)

ADCRL contains the lower eight bits of the result of a 12-bit or 10-bit conversion, and all eight bits of an 8-bit conversion. This register is updated each time a conversion completes except when automatic compare is enabled and the compare condition is not met. In 12-bit and 10-bit mode, reading ADCRH prevents the ADC from transferring subsequent conversion results into the result registers until ADCRL is read. If ADCRL is not read until the after next conversion is completed, the intermediate conversion results are lost. In 8-bit mode, there is no interlocking with ADCRH. If the MODE bits are changed, any data in ADCRL becomes invalid.

	7	6	5	4	3	2	1	0
R	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
W								
Reset:	0	0	0	0	0	0	0	0

Figure 10-6. Data Result Low Register (ADCRL)

10.3.5 Compare Value High Register (ADCCVH)

In 12-bit mode, the ADCCVH register holds the upper four bits of the 12-bit compare value. When the compare function is enabled, these bits are compared to the upper four bits of the result following a conversion in 12-bit mode.

	7	6	5	4	3	2	1	0
R	0	0	0	0	ADCV11	ADCV10	ADCV9	ADCV8
W								
Reset:	0	0	0	0	0	0	0	0

Figure 10-7. Compare Value High Register (ADCCVH)

11.4.1.5 Repeated Start Signal

As shown in Figure 11-9, a repeated start signal is a start signal generated without first generating a stop signal to terminate the communication. This is used by the master to communicate with another slave or with the same slave in different mode (transmit/receive mode) without releasing the bus.

11.4.1.6 Arbitration Procedure

The IIC bus is a true multi-master bus that allows more than one master to be connected on it. If two or more masters try to control the bus at the same time, a clock synchronization procedure determines the bus clock, for which the low period is equal to the longest clock low period and the high is equal to the shortest one among the masters. The relative priority of the contending masters is determined by a data arbitration procedure, a bus master loses arbitration if it transmits logic 1 while another master transmits logic 0. The losing masters immediately switch over to slave receive mode and stop driving SDA output. In this case, the transition from master to slave mode does not generate a stop condition. Meanwhile, a status bit is set by hardware to indicate loss of arbitration.

11.4.1.7 Clock Synchronization

Because wire-AND logic is performed on the SCL line, a high-to-low transition on the SCL line affects all the devices connected on the bus. The devices start counting their low period and after a device's clock has gone low, it holds the SCL line low until the clock high state is reached. However, the change of low to high in this device clock may not change the state of the SCL line if another device clock is still within its low period. Therefore, synchronized clock SCL is held low by the device with the longest low period. Devices with shorter low periods enter a high wait state during this time (see Figure 11-10). When all devices concerned have counted off their low period, the synchronized clock SCL line is released and pulled high. There is then no difference between the device clocks and the state of the SCL line and all the devices start counting their high periods. The first device to complete its high period pulls the SCL line low again.

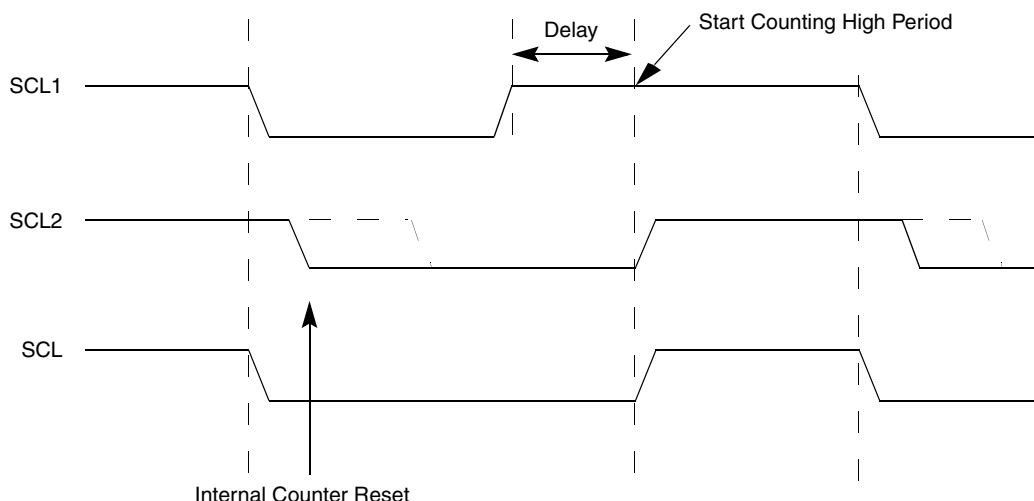


Figure 11-10. IIC Clock Synchronization

12.1.2 Modes of Operation

There are nine modes of operation for the MCG:

- FLL Engaged Internal (FEI)
- FLL Engaged External (FEE)
- FLL Bypassed Internal (FBI)
- FLL Bypassed External (FBE)
- PLL Engaged External (PEE)
- PLL Bypassed External (PBE)
- Bypassed Low Power Internal (BLPI)
- Bypassed Low Power External (BLPE)
- Stop

For details see [Section 12.4.1, “Operational Modes.”](#)

12.2 External Signal Description

There are no MCG signals that connect off chip.

RTCPs and the RTCLKS[0] bit select the desired divide-by value. If a different value is written to RTCPs, the prescaler and RTCCNT counters are reset to 0x00. Table 13-6 shows different prescaler period values.

Table 13-6. Prescaler Period

RTCPs	1-kHz Internal Clock (RTCLKS = 00)	1-MHz External Clock (RTCLKS = 01)	32-kHz Internal Clock (RTCLKS = 10)	32-kHz Internal Clock (RTCLKS = 11)
0000	Off	Off	Off	Off
0001	8 ms	1.024 ms	250 μ s	32 ms
0010	32 ms	2.048 ms	1 ms	64 ms
0011	64 ms	4.096 ms	2 ms	128 ms
0100	128 ms	8.192 ms	4 ms	256 ms
0101	256 ms	16.4 ms	8 ms	512 ms
0110	512 ms	32.8 ms	16 ms	1.024 s
0111	1.024 s	65.5 ms	32 ms	2.048 s
1000	1 ms	1 ms	31.25 μ s	31.25 ms
1001	2 ms	2 ms	62.5 μ s	62.5 ms
1010	4 ms	5 ms	125 μ s	156.25 ms
1011	10 ms	10 ms	312.5 μ s	312.5 ms
1100	16 ms	20 ms	0.5 ms	0.625 s
1101	0.1 s	50 ms	3.125 ms	1.5625 s
1110	0.5 s	0.1 s	15.625 ms	3.125 s
1111	1 s	0.2 s	31.25 ms	6.25 s

The RTC modulo register (RTCMOD) allows the compare value to be set to any value from 0x00 to 0xFF. When the counter is active, the counter increments at the selected rate until the count matches the modulo value. When these values match, the counter resets to 0x00 and continues counting. The real-time interrupt flag (RTIF) is set when a match occurs. The flag sets on the transition from the modulo value to 0x00. Writing to RTCMOD resets the prescaler and the RTCCNT counters to 0x00.

The RTC allows for an interrupt to be generated when RTIF is set. To enable the real-time interrupt, set the real-time interrupt enable bit (RTIE) in RTCSC. RTIF is cleared by writing a 1 to RTIF.

13.4.1 RTC Operation Example

This section shows an example of the RTC operation as the counter reaches a matching value from the modulo register.

Writing 0 to TE does not immediately release the pin to be a general-purpose I/O pin. Any transmit activity that is in progress must first be completed. This includes data characters in progress, queued idle characters, and queued break characters.

14.3.2.1 Send Break and Queued Idle

The SBK control bit in SCIxC2 is used to send break characters which were originally used to gain the attention of old teletype receivers. Break characters are a full character time of logic 0 (10 bit times including the start and stop bits). A longer break of 13 bit times can be enabled by setting BRK13 = 1. Normally, a program would wait for TDRE to become set to indicate the last character of a message has moved to the transmit shifter, then write 1 and then write 0 to the SBK bit. This action queues a break character to be sent as soon as the shifter is available. If SBK is still 1 when the queued break moves into the shifter (synchronized to the baud rate clock), an additional break character is queued. If the receiving device is another Freescale Semiconductor SCI, the break characters will be received as 0s in all eight data bits and a framing error (FE = 1) occurs.

When idle-line wakeup is used, a full character time of idle (logic 1) is needed between messages to wake up any sleeping receivers. Normally, a program would wait for TDRE to become set to indicate the last character of a message has moved to the transmit shifter, then write 0 and then write 1 to the TE bit. This action queues an idle character to be sent as soon as the shifter is available. As long as the character in the shifter does not finish while TE = 0, the SCI transmitter never actually releases control of the TxD pin. If there is a possibility of the shifter finishing while TE = 0, set the general-purpose I/O controls so the pin that is shared with TxD is an output driving a logic 1. This ensures that the TxD line will look like a normal idle line even if the SCI loses control of the port pin between writing 0 and then 1 to TE.

The length of the break character is affected by the BRK13 and M bits as shown below.

Table 14-8. Break Character Length

BRK13	M	Break Character Length
0	0	10 bit times
0	1	11 bit times
1	0	13 bit times
1	1	14 bit times

14.3.3 Receiver Functional Description

In this section, the receiver block diagram ([Figure 14-3](#)) is used as a guide for the overall receiver functional description. Next, the data sampling technique used to reconstruct receiver data is described in more detail. Finally, two variations of the receiver wakeup function are explained.

The receiver input is inverted by setting RXINV = 1. The receiver is enabled by setting the RE bit in SCIxC2. Character frames consist of a start bit of logic 0, eight (or nine) data bits (LSB first), and a stop bit of logic 1. For information about 9-bit data mode, refer to [Section 14.3.5.1, “8- and 9-Bit Data Modes.”](#) For the remainder of this discussion, we assume the SCI is configured for normal 8-bit data mode.

After receiving the stop bit into the receive shifter, and provided the receive data register is not already full, the data character is transferred to the receive data register and the receive data register full (RDRF)

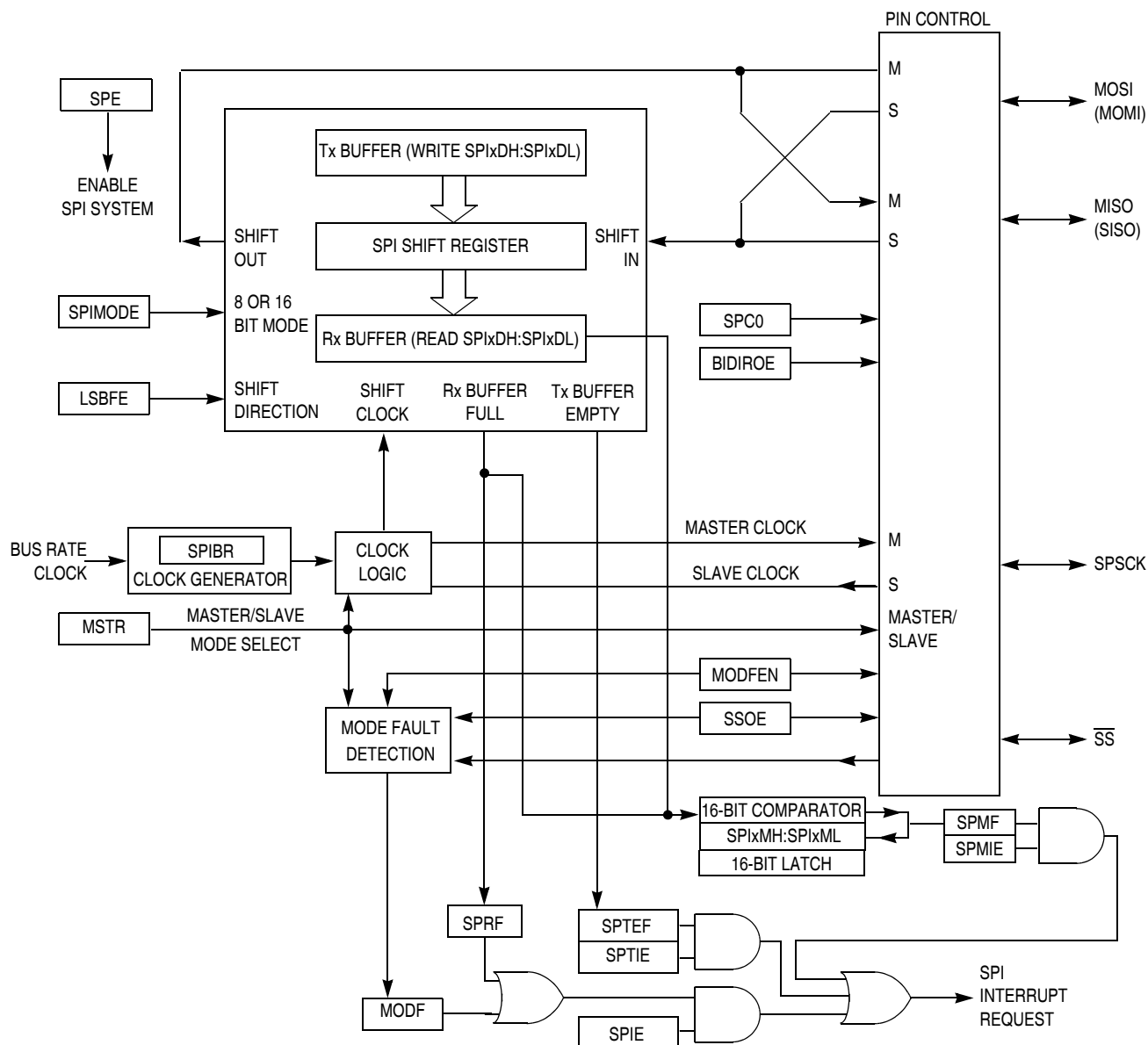


Figure 15-4. SPI Module Block Diagram

15.2 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 being general-purpose port I/O pins that are not controlled by the SPI.

15.2.1 SPCK — SPI Serial Clock

When the SPI is enabled as a slave, this pin is the serial clock input. When the SPI is enabled as a master, this pin is the serial clock output.

The TPM channels are programmable independently as input capture, output compare, or edge-aligned PWM channels. Alternately, the TPM can be configured to produce CPWM outputs on all channels. When the TPM is configured for CPWMs, the counter operates as an up/down counter; input capture, output compare, and EPWM functions are not practical.

If a channel is configured as input capture, an internal pullup device may be enabled for that channel. The details of how a module interacts with pin controls depends upon the chip implementation because the I/O pins and associated general purpose I/O controls are not part of the module. Refer to the discussion of the I/O port logic in a full-chip specification.

Because center-aligned PWMs are usually used to drive 3-phase AC-induction motors and brushless DC motors, they are typically used in sets of three or six channels.

16.2 Signal Description

Table 16-1 shows the user-accessible signals for the TPM. The number of channels may be varied from one to eight. When an external clock is included, it can be shared with the same pin as any TPM channel; however, it could be connected to a separate input pin. Refer to the I/O pin descriptions in full-chip specification for the specific chip implementation.

Table 16-1. Signal Properties

Name	Function
EXTCLK ¹	External clock source which may be selected to drive the TPM counter.
TPMxCHn ²	I/O pin associated with TPM channel n

¹ When preset, this signal can share any channel pin; however depending upon full-chip implementation, this signal could be connected to a separate external pin.

² n=channel number (1 to 8)

Refer to documentation for the full-chip for details about reset states, port connections, and whether there is any pullup device on these pins.

TPM channel pins can be associated with general purpose I/O pins and have passive pullup devices which can be enabled with a control bit when the TPM or general purpose I/O controls have configured the associated pin as an input. When no TPM function is enabled to use a corresponding pin, the pin reverts to being controlled by general purpose I/O controls, including the port-data and data-direction registers. Immediately after reset, no TPM functions are enabled, so all associated pins revert to general purpose I/O control.

16.2.1 Detailed Signal Descriptions

This section describes each user-accessible pin signal in detail. Although Table 16-1 grouped all channel pins together, any TPM pin can be shared with the external clock source signal. Since I/O pin logic is not part of the TPM, refer to full-chip documentation for a specific derivative for more details about the interaction of TPM pin functions and general purpose I/O controls including port data, data direction, and pullup controls.

	7	6	5	4	3	2	1	0
R	Bit 15	14	13	12	11	10	9	Bit 8
W								
Reset	0	0	0	0	0	0	0	0

Figure 16-10. TPM Counter Modulo Register High (TPMxMODH)

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

Figure 16-11. TPM Counter Modulo Register Low (TPMxMODL)

Reset the TPM counter before writing to the TPM modulo registers to avoid confusion about when the first counter overflow will occur.

16.3.4 TPM Channel n Status and Control Register (TPMxCnSC)

TPMxCnSC contains the channel-interrupt-status flag and control bits used to configure the interrupt enable, channel configuration, and pin function.

	7	6	5	4	3	2	1	0
R	CHnF	CHnIE	MSnB	MSnA	ELSnB	ELSnA	0	0
W	0							
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 16-12. TPM Channel n Status and Control Register (TPMxCnSC)

Table 17-1. USBVREN Configuration

USBVREN	3.3-V Regulator	VDD Supply Voltage Range
0	External 3.3-V Regulator (as input to V _{USB33} pin)	V _{USB33} ≤ V _{DD} Supply Voltage
1	Internal 3.3-V Regulator (no external supply connected to V _{USB33} pin)	3.9 V ≤ V _{DD} Supply Voltage ≤ 5.5V

Table 17-12. ERRSTAT Field Descriptions (continued)

Field	Description
1 CRC5	CRC5 Interrupt Enable — Setting this bit will enable CRC5 interrupts. 0 Interrupt disabled 1 Interrupt enabled
0 PIDERR	PIDERR Interrupt Enable — Setting this bit will enable PIDERR interrupts. 0 Interrupt disabled 1 Interrupt enabled

17.3.9 Status Register (STAT)

The STAT reports the transaction status within the USB module. When the MCU receives a TOKDNE interrupt, the STAT is read to determine the status of the previous endpoint communication. The data in the status register is valid only when the TOKDNEF interrupt flag is asserted. The STAT register is actually a read window into a status FIFO maintained by the USB module. When the USB module uses a BD, it updates the status register. If another USB transaction is performed before the TOKDNE interrupt is serviced, the USB module will store the status of the next transaction in the STAT FIFO. Thus, the STAT register is actually a four byte FIFO which allows the microcontroller to process one transaction while the serial interface engine (SIE) is processing the next. Clearing the TOKDNEF bit in the INTSTAT register causes the SIE to update the STAT register with the contents of the next STAT value. If the next data in the STAT FIFO holding register is valid, the SIE will immediately reassert the TOKDNE interrupt.

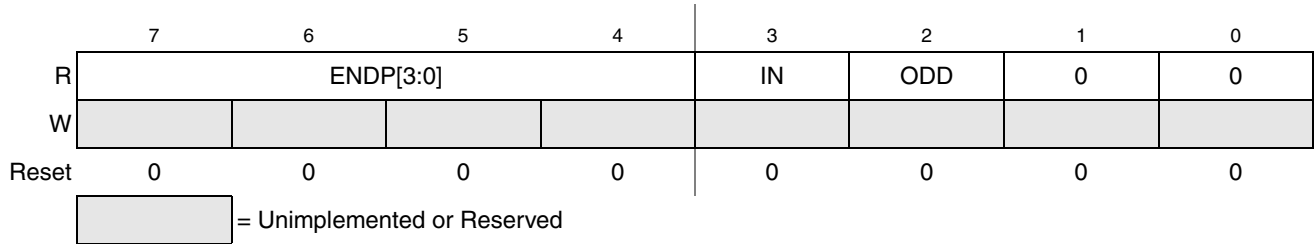


Figure 17-12. Status Register (STAT)

Table 17-13. STAT Field Descriptions

Field	Description
7–4 ENDP[3:0]	Endpoint Number — These four bits encode the endpoint address that received or transmitted the previous token. This allows the microcontroller to determine which BDT entry was updated by the last USB transaction. 0000 Endpoint 0 0001 Endpoint 1 0010 Endpoint 2 0011 Endpoint 3 0100 Endpoint 4 0101 Endpoint 5 0110 Endpoint 6

2. Create BDT entries for Endpoint 0 OUT, and set the DTS and OWN bits to 1.
3. Wait for interrupt TOKDNE.
4. Read STAT register.
 - The status register must show Endpoint 0, RX. If it does not, then assert the EPSTALL bit in the endpoint control register.
5. Read Endpoint 0 OUT BD.
 - Verify that the token type is a SETUP token. If it is not, then assert the EPSTALL bit in the endpoint control register.
6. Decode and process the setup packet.
 - If the direction field in the setup packet indicates an OUT transfer, then process the out data phase to receive exactly the number of bytes specified in the wLength field of the setup packet.
 - If the direction field in the setup packet indicates an IN transfer, then process the in data phase to deliver no more than the number of bytes specified in the wLength field. Note that it is common for the host to request more bytes than it needs, expecting the device to only send as much as it needs to.
7. After processing the data phase (if there was one), create a zero-byte status phase transaction.
 - This is accomplished for an OUT data phase (IN status phase) by setting the BC to 0 in the next BD, while also setting OWN=1. For an IN data phase (OUT status phase), the host will send a zero-byte packet to the device.
 - Firmware can verify completion of the data phase by verifying the received token in the BD on receipt of the TOKDNE interrupt. If the data phase was of type IN, then the status phase token will be OUT. If the data phase was of type OUT, then the status phase token will be IN.

17.4.4.3 Endpoint 0 Exception Conditions

The USB includes a number of error checking and recovery mechanisms to ensure reliable data transfer. One such exception occurs when the host sends a SETUP packet to a device, and the host never receives the acknowledge handshake from the device. In this case, the host will retry the SETUP packet.

Endpoint 0 request handlers on the device must be aware of the possibility that after receiving a correct SETUP packet, they could receive another SETUP packet before the data phase actually begins.

17.4.5 Start of Frame Processing

The USB host allocates time in 1.0 ms chunks called “Frames” for the purposes of packet scheduling. The USB host starts each frame with a broadcast token called SOF (start of frame) that includes an 11-bit sequence number. The TOKSOF interrupt is used to notify firmware when an SOF token was received.

Firmware can read the current frame number from the FRMNUML/FRMNUMH registers.

In general, the SOF interrupt is only monitored by devices using isochronous endpoints to help ensure that the device and host remain synchronized.

a resume from low-power suspend. This will trigger an asynchronous interrupt to wake the CPU from stop3 mode and resume clocks to the USB module.

NOTE

As a precaution, after LPRESF is set, firmware must check the state of the USB bus to see if the K-state was a result of a transient event and not a true host-initiated resume. If this is the case, then the device can drop back into stop3 if necessary. To do this, the RESUME interrupt can be enabled in conjunction with the USBRESMEN feature. Then, after LPRESF is set, and a K-state is still detected approximately 2.5 μ s after clocks have restarted, firmware can check that the RESUMEF interrupt has triggered, indicating resume signaling from the host.

17.4.6.2.2 USB Reset Signaling

Reset can wake a device from the suspend state.

17.4.6.2.3 Remote Wakeup

The USB device can send a resume event to the host by writing to the CRESUME bit. Firmware must first set the bit for the time period required by the USB Specification Rev. 2.0 (Section 7.1.7.7) and then clear it to 0.

17.4.7 Resets

The module supports multiple types of resets. The first is a bus reset generated by the USB Host, the second is a module reset generated by the MCU.

17.4.7.1 USB Bus Reset

At any time, the USB host may issue a reset to one or all of the devices attached to the bus. A USB reset is defined as a period of single ended zero (SE0) on the cable for greater than 2.5 μ s. When the device detects reset signaling, it resets itself to the unconfigured state, and sets its USB address zero. The USB host uses reset signaling to force one or all connected devices into a known state prior to commencing enumeration.

The USB module responds to reset signaling by asserting the USBRST interrupt in the INTSTAT register. Software is required to service this interrupt to ensure correct operation of the USB.

17.4.7.2 USB Module Reset

USB module resets are initiated on-chip. During a module reset, the USB module is configured in the default mode. The USB module can also be forced into its reset state by setting the USBRESET bit in the USBCTL0 register. The default mode includes the following settings:

- Interrupts masked.
- USB clock enabled
- USB voltage regulator disabled

