

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	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	I ² C, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	13
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	16-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	16-TSSOP
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08sh4mtgr

Section Number	Title	Page
6.3	Ganged Output	77
6.4	Pin Interrupts	78
6.4.1	Edge Only Sensitivity	78
6.4.2	Edge and Level Sensitivity	78
6.4.3	Pull-up/Pull-down Resistors	79
6.4.4	Pin Interrupt Initialization	79
6.5	Pin Behavior in Stop Modes	79
6.6	Parallel I/O and Pin Control Registers	79
6.6.1	Port A Registers	80
6.6.2	Port B Registers	85
6.6.3	Port C Registers	89

Chapter 7 Central Processor Unit (S08CPUV2)

7.1	Introduction	93
7.1.1	Features	93
7.2	Programmer's Model and CPU Registers	94
7.2.1	Accumulator (A)	94
7.2.2	Index Register (H:X)	94
7.2.3	Stack Pointer (SP)	95
7.2.4	Program Counter (PC)	95
7.2.5	Condition Code Register (CCR)	95
7.3	Addressing Modes	97
7.3.1	Inherent Addressing Mode (INH)	97
7.3.2	Relative Addressing Mode (REL)	97
7.3.3	Immediate Addressing Mode (IMM)	97
7.3.4	Direct Addressing Mode (DIR)	97
7.3.5	Extended Addressing Mode (EXT)	98
7.3.6	Indexed Addressing Mode	98
7.4	Special Operations	99
7.4.1	Reset Sequence	99
7.4.2	Interrupt Sequence	99
7.4.3	Wait Mode Operation	100
7.4.4	Stop Mode Operation	100
7.4.5	BGND Instruction	101
7.5	HCS08 Instruction Set Summary	102

Chapter 8 Analog Comparator 5-V (S08ACMPV2)

8.1	Introduction	113
8.1.1	ACMP Configuration Information	113
8.1.2	ACMP in Stop3 Mode	113

other than the most experienced programmers because it can lead to subtle program errors that are difficult to debug.

The interrupt service routine ends with a return-from-interrupt (RTI) instruction which restores the CCR, A, X, and PC registers to their pre-interrupt values by reading the previously saved information from the stack.

NOTE

For compatibility with M68HC08 devices, the H register is not automatically saved and restored. It is good programming practice to push H onto the stack at the start of the interrupt service routine (ISR) and restore it immediately before the RTI that is used to return from the ISR.

If more than one interrupt is pending when the I bit is cleared, the highest priority source is serviced first (see Table 5-2).

5.5.1 Interrupt Stack Frame

Figure 5-1 shows the contents and organization of a stack frame. Before the interrupt, the stack pointer (SP) points at the next available byte location on the stack. The current values of CPU registers are stored on the stack starting with the low-order byte of the program counter (PCL) and ending with the CCR. After stacking, the SP points at the next available location on the stack which is the address that is one less than the address where the CCR was saved. The PC value that is stacked is the address of the instruction in the main program that would have executed next if the interrupt had not occurred.

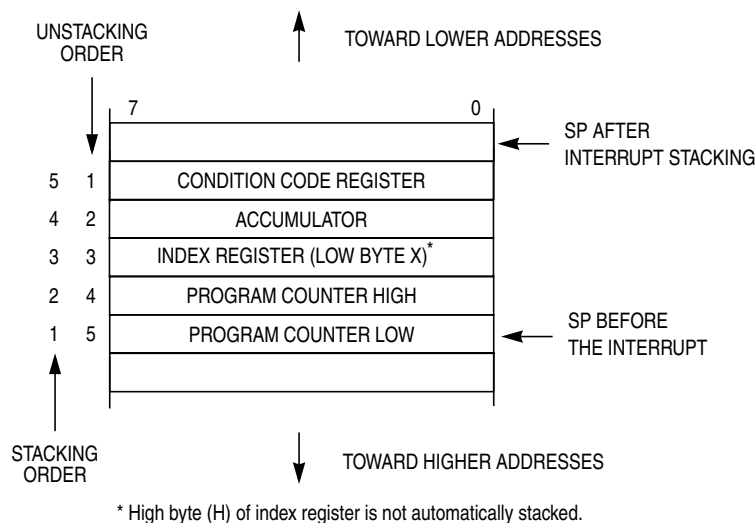
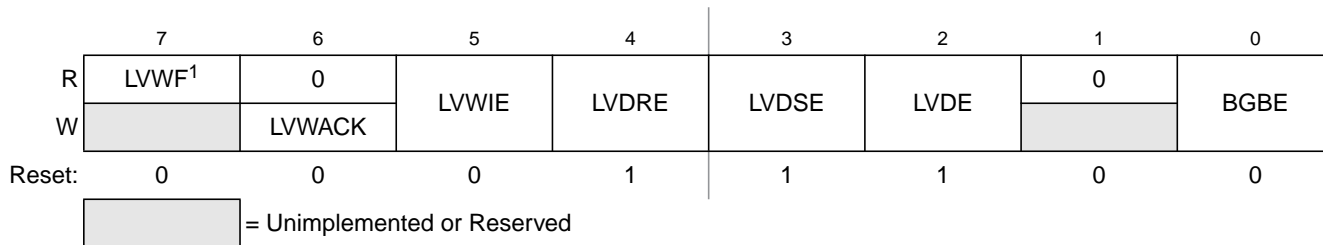


Figure 5-1. Interrupt Stack Frame

When an RTI instruction is executed, these values are recovered from the stack in reverse order. As part of the RTI sequence, the CPU fills the instruction pipeline by reading three bytes of program information, starting from the PC address recovered from the stack.

5.7.7 System Power Management Status and Control 1 Register (SPMSC1)

This high page register contains status and control bits to support the low voltage detect function, and to enable the bandgap voltage reference for use by the ADC module.



¹ LVWF will be set in the case when V_{Supply} transitions below the trip point or after reset and V_{Supply} is already below V_{LVW}

Figure 5-9. System Power Management Status and Control 1 Register (SPMSC1)

Table 5-10. SPMSC1 Register Field Descriptions

Field	Description
7 LVWF	Low-Voltage Warning Flag — The LVWF bit indicates the low voltage warning status. 0 Low voltage warning is not present. 1 Low voltage warning is present or was present.
6 LVWACK	Low-Voltage Warning Acknowledge — The LVWF bit indicates the low voltage warning status. Writing a 1 to LVWACK clears LVWF to a 0 if a low voltage warning is not present.
5 LVWIE	Low-Voltage Warning Interrupt Enable — This bit enables hardware interrupt requests for LVWF. 0 Hardware interrupt disabled (use polling). 1 Request a hardware interrupt when LVWF = 1.
4 LVDRE	Low-Voltage Detect Reset Enable — This write-once bit enables LVD events to generate a hardware reset (provided LVDE = 1). 0 LVD events do not generate hardware resets. 1 Force an MCU reset when an enabled low-voltage detect event occurs.
3 LVDSE	Low-Voltage Detect Stop Enable — Provided LVDE = 1, this control bit determines whether the low-voltage detect function operates when the MCU is in stop mode. 0 Low-voltage detect disabled during stop mode. 1 Low-voltage detect enabled during stop mode.
2 LVDE	Low-Voltage Detect Enable — This write-once bit enables low-voltage detect logic and qualifies the operation of other bits in this register. 0 LVD logic disabled. 1 LVD logic enabled.
0 BGBE	Bandgap Buffer Enable — This bit enables an internal buffer for the bandgap voltage reference for use by the ADC module on one of its internal channels or ACMP on its ACMP+ input. 0 Bandgap buffer disabled. 1 Bandgap buffer enabled.

6.6.2.3 Port B Pull Enable Register (PTBPE)

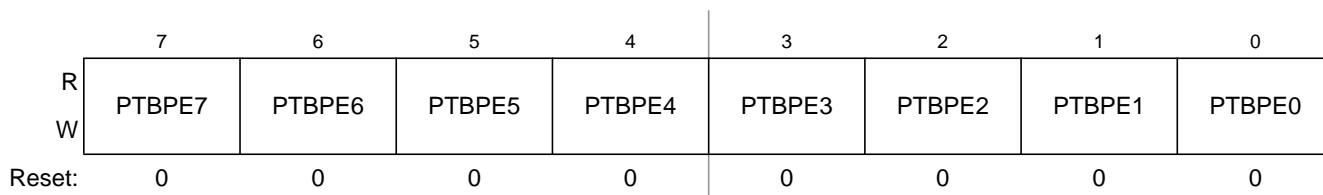


Figure 6-13. Internal Pull Enable for Port B Register (PTBPE)

Table 6-12. PTBPE Register Field Descriptions

Field	Description
7:0 PTBPE[7:0]	<p>Internal Pull Enable for Port B Bits — Each of these control bits determines if the internal pull-up or pull-down device is enabled for the associated PTB pin. For port B pins that are configured as outputs, these bits have no effect and the internal pull devices are disabled.</p> <p>0 Internal pull-up/pull-down device disabled for port B bit n. 1 Internal pull-up/pull-down device enabled for port B bit n.</p>

NOTE

Pull-down devices only apply when using pin interrupt functions, when corresponding edge select and pin select functions are configured.

6.6.2.4 Port B Slew Rate Enable Register (PTBSE)

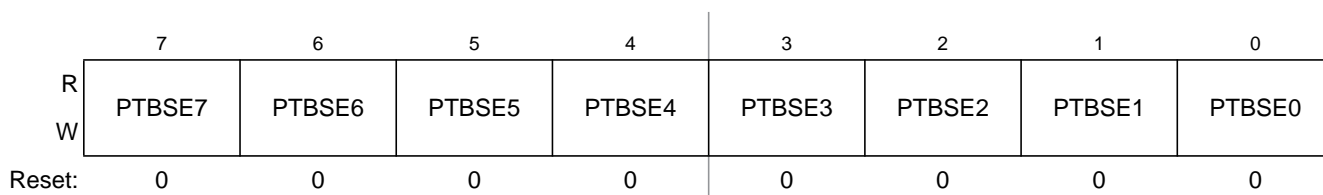


Figure 6-14. Slew Rate Enable for Port B Register (PTBSE)

Table 6-13. PTBSE Register Field Descriptions

Field	Description
7:0 PTBSE[7:0]	<p>Output Slew Rate Enable for Port B Bits — Each of these control bits determines if the output slew rate control is enabled for the associated PTB pin. For port B pins that are configured as inputs, these bits have no effect.</p> <p>0 Output slew rate control disabled for port B bit n. 1 Output slew rate control enabled for port B bit n.</p>

6.6.3.3 Port C Pull Enable Register (PTCPE)

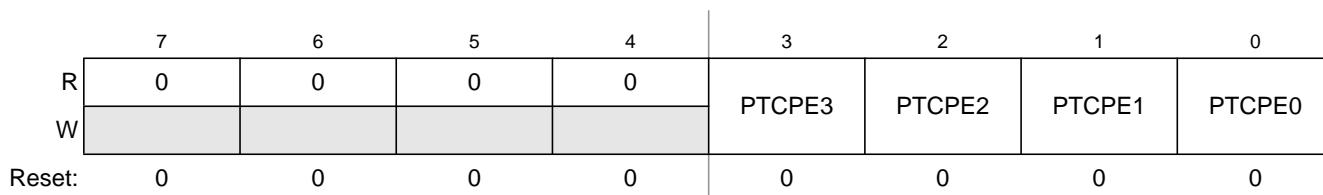


Figure 6-21. Internal Pull Enable for Port C Register (PTCPE)

Table 6-20. PTCPE Register Field Descriptions

Field	Description
3:0 PTCPE[3:0]	<p>Internal Pull Enable for Port C Bits — Each of these control bits determines if the internal pull-up device is enabled for the associated PTC pin. For port C pins that are configured as outputs, these bits have no effect and the internal pull devices are disabled.</p> <p>0 Internal pull-up device disabled for port C bit n. 1 Internal pull-up device enabled for port C bit n.</p>

6.6.3.4 Port C Slew Rate Enable Register (PTCSE)

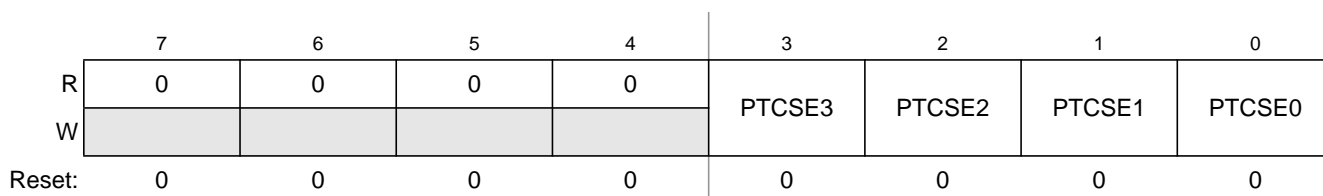


Figure 6-22. Slew Rate Enable for Port C Register (PTCSE)

Table 6-21. PTCSE Register Field Descriptions

Field	Description
3:0 PTCSE[3:0]	<p>Output Slew Rate Enable for Port C Bits — Each of these control bits determines if the output slew rate control is enabled for the associated PTC pin. For port C pins that are configured as inputs, these bits have no effect.</p> <p>0 Output slew rate control disabled for port C bit n. 1 Output slew rate control enabled for port C bit n.</p>

interrupt service routine, this would allow nesting of interrupts (which is not recommended because it leads to programs that are difficult to debug and maintain).

For compatibility with the earlier M68HC05 MCUs, the high-order half of the H:X index register pair (H) is not saved on the stack as part of the interrupt sequence. The user must use a PSHH instruction at the beginning of the service routine to save H and then use a PULH instruction just before the RTI that ends the interrupt service routine. It is not necessary to save H if you are certain that the interrupt service routine does not use any instructions or auto-increment addressing modes that might change the value of H.

The software interrupt (SWI) instruction is like a hardware interrupt except that it is not masked by the global I bit in the CCR and it is associated with an instruction opcode within the program so it is not asynchronous to program execution.

7.4.3 Wait Mode Operation

The WAIT instruction enables interrupts by clearing the I bit in the CCR. It then halts the clocks to the CPU to reduce overall power consumption while the CPU is waiting for the interrupt or reset event that will wake the CPU from wait mode. When an interrupt or reset event occurs, the CPU clocks will resume and the interrupt or reset event will be processed normally.

If a serial BACKGROUND command is issued to the MCU through the background debug interface while the CPU is in wait mode, CPU clocks will resume and the CPU will enter active background mode where other serial background commands can be processed. This ensures that a host development system can still gain access to a target MCU even if it is in wait mode.

7.4.4 Stop Mode Operation

Usually, all system clocks, including the crystal oscillator (when used), are halted during stop mode to minimize power consumption. In such systems, external circuitry is needed to control the time spent in stop mode and to issue a signal to wake up the target MCU when it is time to resume processing. Unlike the earlier M68HC05 and M68HC08 MCUs, the HCS08 can be configured to keep a minimum set of clocks running in stop mode. This optionally allows an internal periodic signal to wake the target MCU from stop mode.

When a host debug system is connected to the background debug pin (BKGD) and the ENBDM control bit has been set by a serial command through the background interface (or because the MCU was reset into active background mode), the oscillator is forced to remain active when the MCU enters stop mode. In this case, if a serial BACKGROUND command is issued to the MCU through the background debug interface while the CPU is in stop mode, CPU clocks will resume and the CPU will enter active background mode where other serial background commands can be processed. This ensures that a host development system can still gain access to a target MCU even if it is in stop mode.

Recovery from stop mode depends on the particular HCS08 and whether the oscillator was stopped in stop mode. Refer to the [Modes of Operation](#) chapter for more details.

8.2 External Signal Description

The ACMP has two analog input pins, ACMP+ and ACMP- and one digital output pin ACMPO. Each of these pins can accept an input voltage that varies across the full operating voltage range of the MCU. As shown in Figure 8-2, the ACMP- pin is connected to the inverting input of the comparator, and the ACMP+ pin is connected to the comparator non-inverting input if ACBGS is a 0. As shown in Figure 8-2, the ACMPO pin can be enabled to drive an external pin.

The signal properties of ACMP are shown in Table 8-1.

Table 8-1. Signal Properties

Signal	Function	I/O
ACMP-	Inverting analog input to the ACMP. (Minus input)	I
ACMP+	Non-inverting analog input to the ACMP. (Positive input)	I
ACMPO	Digital output of the ACMP.	O

8.3 Memory Map

8.3.1 Register Descriptions

The ACMP includes one register:

- An 8-bit status and control register

Refer to the direct-page register summary in the memory section of this data sheet for the absolute address assignments for all ACMP registers. This section refers to registers and control bits only by their names .

Some MCUs may have more than one ACMP, so register names include placeholder characters to identify which ACMP is being referenced.

8.3.1.1 ACMP Status and Control Register (ACMPSC)

ACMPSC contains the status flag and control bits which are used to enable and configure the ACMP.

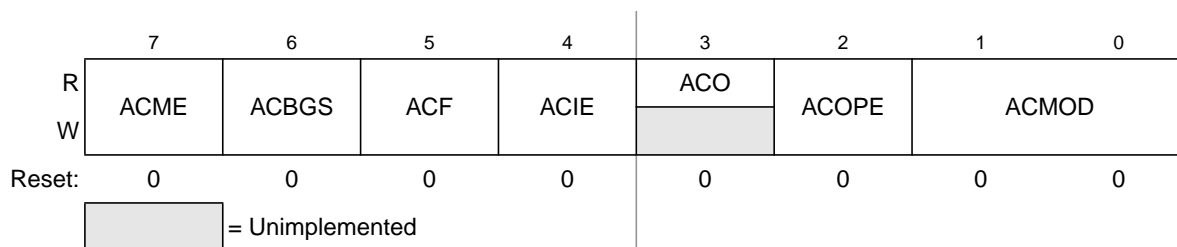


Figure 8-3. ACMP Status and Control Register

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

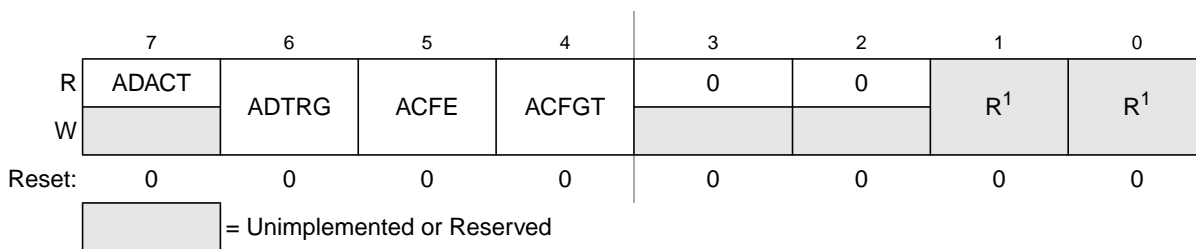
Field	Description
7 ACME	Analog Comparator Module Enable — ACME enables the ACMP module. 0 ACMP not enabled 1 ACMP is enabled
6 ACBGS	Analog Comparator Bandgap Select — ACBGS is used to select between the bandgap reference voltage or the ACMP+ pin as the input to the non-inverting input of the analog comparator. 0 External pin ACMP+ selected as non-inverting input to comparator 1 Internal reference select as non-inverting input to comparator Note: refer to this chapter introduction to verify if any other config bits are necessary to enable the bandgap reference in the chip level.
5 ACF	Analog Comparator Flag — ACF is set when a compare event occurs. Compare events are defined by ACMOD. ACF is cleared by writing a one to ACF. 0 Compare event has not occurred 1 Compare event has occurred
4 ACIE	Analog Comparator Interrupt Enable — ACIE enables the interrupt from the ACMP. When ACIE is set, an interrupt will be asserted when ACF is set. 0 Interrupt disabled 1 Interrupt enabled
3 ACO	Analog Comparator Output — Reading ACO will return the current value of the analog comparator output. ACO is reset to a 0 and will read as a 0 when the ACMP is disabled (ACME = 0).
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

Figure 9-4. Input Channel Select (continued)

ADCH	Input Select	ADCH	Input Select
01000	AD8	11000	AD24
01001	AD9	11001	AD25
01010	AD10	11010	AD26
01011	AD11	11011	AD27
01100	AD12	11100	Reserved
01101	AD13	11101	V _{REFH}
01110	AD14	11110	V _{REFL}
01111	AD15	11111	Module disabled

9.3.2 Status and Control Register 2 (ADCSC2)

The ADCSC2 register is used to control the compare function, conversion trigger and conversion active of the ADC module.



¹ Bits 1 and 0 are reserved bits that must always be written to 0.

Figure 9-5. Status and Control Register 2 (ADCSC2)

Table 9-4. ADCSC2 Register Field Descriptions

Field	Description
7 ADACT	Conversion Active — ADACT indicates that a conversion is in progress. ADACT is set when a conversion is initiated and cleared when a conversion is completed or aborted. 0 Conversion not in progress 1 Conversion in progress
6 ADTRG	Conversion Trigger Select — ADTRG is used to select the type of trigger to be used for initiating a conversion. Two types of trigger are selectable: software trigger and hardware trigger. When software trigger is selected, a conversion is initiated following a write to ADCSC1. When hardware trigger is selected, a conversion is initiated following the assertion of the ADHWT input. 0 Software trigger selected 1 Hardware trigger selected

10.3.1 ICS Control Register 1 (ICSC1)

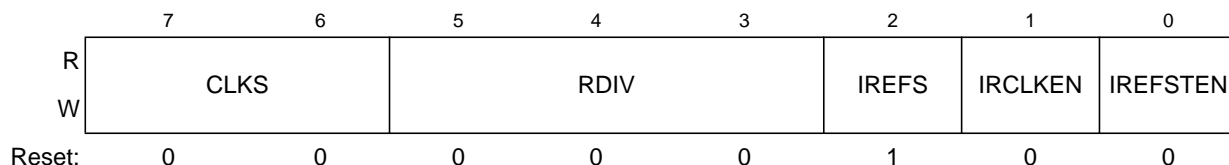


Figure 10-3. ICS Control Register 1 (ICSC1)

Table 10-2. ICS Control Register 1 Field Descriptions

Field	Description
7:6 CLKS	<p>Clock Source Select — Selects the clock source that controls the bus frequency. The actual bus frequency depends on the value of the BDIV bits.</p> <p>00 Output of FLL is selected. 01 Internal reference clock is selected. 10 External reference clock is selected. 11 Reserved, defaults to 00.</p>
5:3 RDIV	<p>Reference Divider — Selects the amount to divide down the FLL reference clock selected by the IREFS bits. Resulting frequency must be in the range 31.25 kHz to 39.0625 kHz.</p> <p>000 Encoding 0 — Divides reference clock by 1 (reset default) 001 Encoding 1 — Divides reference clock by 2 010 Encoding 2 — Divides reference clock by 4 011 Encoding 3 — Divides reference clock by 8 100 Encoding 4 — Divides reference clock by 16 101 Encoding 5 — Divides reference clock by 32 110 Encoding 6 — Divides reference clock by 64 111 Encoding 7 — Divides reference clock by 128</p>
2 IREFS	<p>Internal Reference Select — The IREFS bit selects the reference clock source for the FLL.</p> <p>1 Internal reference clock selected 0 External reference clock selected</p>
1 IRCLKEN	<p>Internal Reference Clock Enable — The IRCLKEN bit enables the internal reference clock for use as ICSIRCLK.</p> <p>1 ICSIRCLK active 0 ICSIRCLK inactive</p>
0 IREFSTEN	<p>Internal Reference Stop Enable — The IREFSTEN bit controls whether or not the internal reference clock remains enabled when the ICS enters stop mode.</p> <p>1 Internal reference clock stays enabled in stop if IRCLKEN is set or if ICS is in FEI, FBI, or FBILP mode before entering stop 0 Internal reference clock is disabled in stop</p>

11.4.2 10-bit Address

For 10-bit addressing, 0x11110 is used for the first 5 bits of the first address byte. Various combinations of read/write formats are possible within a transfer that includes 10-bit addressing.

11.4.2.1 Master-Transmitter Addresses a Slave-Receiver

The transfer direction is not changed (see Table 11-10). When a 10-bit address follows a start condition, each slave compares the first seven bits of the first byte of the slave address (11110XX) with its own address and tests whether the eighth bit (R/\overline{W} direction bit) is 0. More than one device can find a match and generate an acknowledge (A1). Then, each slave that finds a match compares the eight bits of the second byte of the slave address with its own address. Only one slave finds a match and generates an acknowledge (A2). The matching slave remains addressed by the master until it receives a stop condition (P) or a repeated start condition (Sr) followed by a different slave address.

S	Slave Address 1st 7 bits 11110 + AD10 + AD9	R/W 0	A1	Slave Address 2nd byte AD[8:1]	A2	Data	A	...	Data	A/A	P
---	--	----------	----	-----------------------------------	----	------	---	-----	------	-----	---

Table 11-10. Master-Transmitter Addresses Slave-Receiver with a 10-bit Address

After the master-transmitter has sent the first byte of the 10-bit address, the slave-receiver sees an IIC interrupt. Software must ensure the contents of IICD are ignored and not treated as valid data for this interrupt.

11.4.2.2 Master-Receiver Addresses a Slave-Transmitter

The transfer direction is changed after the second R/\overline{W} bit (see Table 11-11). Up to and including acknowledge bit A2, the procedure is the same as that described for a master-transmitter addressing a slave-receiver. After the repeated start condition (Sr), a matching slave remembers that it was addressed before. This slave then checks whether the first seven bits of the first byte of the slave address following Sr are the same as they were after the start condition (S) and tests whether the eighth (R/\overline{W}) bit is 1. If there is a match, the slave considers that it has been addressed as a transmitter and generates acknowledge A3. The slave-transmitter remains addressed until it receives a stop condition (P) or a repeated start condition (Sr) followed by a different slave address.

After a repeated start condition (Sr), all other slave devices also compare the first seven bits of the first byte of the slave address with their own addresses and test the eighth (R/\overline{W}) bit. However, none of them are addressed because $R/\overline{W} = 1$ (for 10-bit devices) or the 11110XX slave address (for 7-bit devices) does not match.

S	Slave Address 1st 7 bits 11110 + AD10 + AD9	R/W 0	A1	Slave Address 2nd byte AD[8:1]	A2	Sr	Slave Address 1st 7 bits 11110 + AD10 + AD9	R/W 1	A3	Data	A	...	Data	A	P
---	--	----------	----	-----------------------------------	----	----	--	----------	----	------	---	-----	------	---	---

Table 11-11. Master-Receiver Addresses a Slave-Transmitter with a 10-bit Address

After the master-receiver has sent the first byte of the 10-bit address, the slave-transmitter sees an IIC interrupt. Software must ensure the contents of IICD are ignored and not treated as valid data for this interrupt.

Table 14-4. SCIx2 Field Descriptions (continued)

Field	Description
3 TE	Transmitter Enable 0 Transmitter off. 1 Transmitter on. TE must be 1 in order to use the SCI transmitter. When TE = 1, the SCI forces the TxD pin to act as an output for the SCI system. When the SCI is configured for single-wire operation (LOOPS = RSRC = 1), TXDIR controls the direction of traffic on the single SCI communication line (TxD pin). TE also can be used to queue an idle character by writing TE = 0 then TE = 1 while a transmission is in progress. Refer to Section 14.3.2.1, “Send Break and Queued Idle” for more details. When TE is written to 0, the transmitter keeps control of the port TxD pin until any data, queued idle, or queued break character finishes transmitting before allowing the pin to revert to a general-purpose I/O pin.
2 RE	Receiver Enable — When the SCI receiver is off, the RxD pin reverts to being a general-purpose port I/O pin. If LOOPS = 1 the RxD pin reverts to being a general-purpose I/O pin even if RE = 1. 0 Receiver off. 1 Receiver on.
1 RWU	Receiver Wakeup Control — This bit can be written to 1 to place the SCI receiver in a standby state where it waits for automatic hardware detection of a selected wakeup condition. The wakeup condition is either an idle line between messages (WAKE = 0, idle-line wakeup), or a logic 1 in the most significant data bit in a character (WAKE = 1, address-mark wakeup). Application software sets RWU and (normally) a selected hardware condition automatically clears RWU. Refer to Section 14.3.3.2, “Receiver Wakeup Operation” for more details. 0 Normal SCI receiver operation. 1 SCI receiver in standby waiting for wakeup condition.
0 SBK	Send Break — Writing a 1 and then a 0 to SBK queues a break character in the transmit data stream. Additional break characters of 10 or 11 (13 or 14 if BRK13 = 1) bit times of logic 0 are queued as long as SBK = 1. Depending on the timing of the set and clear of SBK relative to the information currently being transmitted, a second break character may be queued before software clears SBK. Refer to Section 14.3.2.1, “Send Break and Queued Idle” for more details. 0 Normal transmitter operation. 1 Queue break character(s) to be sent.

14.2.4 SCI Status Register 1 (SClXS1)

This register has eight read-only status flags. Writes have no effect. Special software sequences (which do not involve writing to this register) are used to clear these status flags.

	7	6	5	4	3	2	1	0
R	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
W								
Reset	1	1	0	0	0	0	0	0


 = Unimplemented or Reserved

Figure 14-8. SCI Status Register 1 (SClXS1)

Table 14-5. SC1xS1 Field Descriptions

Field	Description
7 TDRE	<p>Transmit Data Register Empty Flag — TDRE is set out of reset and when a transmit data value transfers from the transmit data buffer to the transmit shifter, leaving room for a new character in the buffer. To clear TDRE, read SC1xS1 with TDRE = 1 and then write to the SCI data register (SC1xD).</p> <p>0 Transmit data register (buffer) full. 1 Transmit data register (buffer) empty.</p>
6 TC	<p>Transmission Complete Flag — TC is set out of reset and when TDRE = 1 and no data, preamble, or break character is being transmitted.</p> <p>0 Transmitter active (sending data, a preamble, or a break). 1 Transmitter idle (transmission activity complete).</p> <p>TC is cleared automatically by reading SC1xS1 with TC = 1 and then doing one of the following three things:</p> <ul style="list-style-type: none"> • Write to the SCI data register (SC1xD) to transmit new data • Queue a preamble by changing TE from 0 to 1 • Queue a break character by writing 1 to SBK in SC1xC2
5 RDRF	<p>Receive Data Register Full Flag — RDRF becomes set when a character transfers from the receive shifter into the receive data register (SC1xD). To clear RDRF, read SC1xS1 with RDRF = 1 and then read the SCI data register (SC1xD).</p> <p>0 Receive data register empty. 1 Receive data register full.</p>
4 IDLE	<p>Idle Line Flag — IDLE is set when the SCI receive line becomes idle for a full character time after a period of activity. When ILT = 0, the receiver starts counting idle bit times after the start bit. So if the receive character is all 1s, these bit times and the stop bit time count toward the full character time of logic high (10 or 11 bit times depending on the M control bit) needed for the receiver to detect an idle line. When ILT = 1, the receiver doesn't start counting idle bit times until after the stop bit. So the stop bit and any logic high bit times at the end of the previous character do not count toward the full character time of logic high needed for the receiver to detect an idle line.</p> <p>To clear IDLE, read SC1xS1 with IDLE = 1 and then read the SCI data register (SC1xD). After IDLE has been cleared, it cannot become set again until after a new character has been received and RDRF has been set. IDLE will get set only once even if the receive line remains idle for an extended period.</p> <p>0 No idle line detected. 1 Idle line was detected.</p>
3 OR	<p>Receiver Overrun Flag — OR is set when a new serial character is ready to be transferred to the receive data register (buffer), but the previously received character has not been read from SC1xD yet. In this case, the new character (and all associated error information) is lost because there is no room to move it into SC1xD. To clear OR, read SC1xS1 with OR = 1 and then read the SCI data register (SC1xD).</p> <p>0 No overrun. 1 Receive overrun (new SCI data lost).</p>
2 NF	<p>Noise Flag — The advanced sampling technique used in the receiver takes seven samples during the start bit and three samples in each data bit and the stop bit. If any of these samples disagrees with the rest of the samples within any bit time in the frame, the flag NF will be set at the same time as the flag RDRF gets set for the character. To clear NF, read SC1xS1 and then read the SCI data register (SC1xD).</p> <p>0 No noise detected. 1 Noise detected in the received character in SC1xD.</p>

Table 16-3. TPMxSC Field Descriptions (continued)

Field	Description
4–3 CLKS[B:A]	Clock source selects. As shown in Table 16-4 , this 2-bit field is used to disable the TPM system or select one of three clock sources to drive the counter prescaler. The fixed system clock source is only meaningful in systems with a PLL-based or FLL-based system clock. When there is no PLL or FLL, the fixed-system clock source is the same as the bus rate clock. The external source is synchronized to the bus clock by TPM module, and the fixed system clock source (when a PLL or FLL is present) is synchronized to the bus clock by an on-chip synchronization circuit. When a PLL or FLL is present but not enabled, the fixed-system clock source is the same as the bus-rate clock.
2–0 PS[2:0]	Prescale factor select. This 3-bit field selects one of 8 division factors for the TPM clock input as shown in Table 16-5 . This prescaler is located after any clock source synchronization or clock source selection so it affects the clock source selected to drive the TPM system. The new prescale factor will affect the clock source on the next system clock cycle after the new value is updated into the register bits.

Table 16-4. TPM-Clock-Source Selection

CLKSB:CLKSA	TPM Clock Source to Prescaler Input
00	No clock selected (TPM counter disable)
01	Bus rate clock
10	Fixed system clock
11	External source

Table 16-5. Prescale Factor Selection

PS2:PS1:PS0	TPM Clock Source Divided-by
000	1
001	2
010	4
011	8
100	16
101	32
110	64
111	128

16.3.2 TPM-Counter Registers (TPMxCNTH:TPMxCNTL)

The two read-only TPM counter registers contain the high and low bytes of the value in the TPM counter. Reading either byte (TPMxCNTH or TPMxCNTL) latches the contents of both bytes into a buffer where they remain latched until the other half is read. This allows coherent 16-bit reads in either big-endian or little-endian order which makes this more friendly to various compiler implementations. The coherency mechanism is automatically restarted by an MCU reset or any write to the timer status/control register (TPMxSC).

(becomes unlatched) when the TPMxCnSC register is written (whether BDM mode is active or not). Any write to the channel registers will be ignored during the input capture mode.

When BDM is active, the coherency mechanism is frozen (unless reset by writing to TPMxCnSC register) such that the buffer latches remain in the state they were in when the BDM became active, even if one or both halves of the channel register are read while BDM is active. This assures that if the user was in the middle of reading a 16-bit register when BDM became active, it will read the appropriate value from the other half of the 16-bit value after returning to normal execution. The value read from the TPMxCnVH and TPMxCnVL registers in BDM mode is the value of these registers and not the value of their read buffer.

In output compare or PWM modes, writing to either byte (TPMxCnVH or TPMxCnVL) latches the value into a buffer. After both bytes are written, they are transferred as a coherent 16-bit value into the timer-channel registers according to the value of CLKS_B:CLKS_A bits and the selected mode, so:

- If (CLKS_B:CLKS_A = 0:0), then the registers are updated when the second byte is written.
- If (CLKS_B:CLKS_A not = 0:0 and in output compare mode) then the registers are updated after the second byte is written and on the next change of the TPM counter (end of the prescaler counting).
- If (CLKS_B:CLKS_A not = 0:0 and in EPWM or CPWM modes), then the registers are updated after the both bytes were written, and the TPM counter changes from (TPMxMODH:TPMxMODL - 1) to (TPMxMODH:TPMxMODL). If the TPM counter is a free-running counter then the update is made when the TPM counter changes from 0xFFFFE to 0xFFFF.

The latching mechanism may be manually reset by writing to the TPMxCnSC register (whether BDM mode is active or not). This latching mechanism allows coherent 16-bit writes in either big-endian or little-endian order which is friendly to various compiler implementations.

When BDM is active, the coherency mechanism is frozen such that the buffer latches remain in the state they were in when the BDM became active even if one or both halves of the channel register are written while BDM is active. Any write to the channel registers bypasses the buffer latches and directly write to the channel register while BDM is active. The values written to the channel register while BDM is active are used for PWM & output compare operation once normal execution resumes. Writes to the channel registers while BDM is active do not interfere with partial completion of a coherency sequence. After the coherency mechanism has been fully exercised, the channel registers are updated using the buffered values written (while BDM was not active) by the user.

16.4 Functional Description

All TPM functions are associated with a central 16-bit counter which allows flexible selection of the clock source and prescale factor. There is also a 16-bit modulo register associated with the main counter.

The CPWMS control bit chooses between center-aligned PWM operation for all channels in the TPM (CPWMS=1) or general purpose timing functions (CPWMS=0) where each channel can independently be configured to operate in input capture, output compare, or edge-aligned PWM mode. The CPWMS control bit is located in the main TPM status and control register because it affects all channels within the TPM and influences the way the main counter operates. (In CPWM mode, the counter changes to an up/down mode rather than the up-counting mode used for general purpose timer functions.)

The average chip-junction temperature (T_J) in °C can be obtained from:

$$T_J = T_A + (P_D \times \theta_{JA}) \quad \text{Eqn. A-1}$$

where:

T_A = Ambient temperature, °C

θ_{JA} = Package thermal resistance, junction-to-ambient, °C/W

$P_D = P_{int} + P_{I/O}$

$P_{int} = I_{DD} \times V_{DD}$, Watts — chip internal power

$P_{I/O}$ = Power dissipation on input and output pins — user determined

For most applications, $P_{I/O} \ll P_{int}$ and can be neglected. An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

$$P_D = K \div (T_J + 273^\circ\text{C}) \quad \text{Eqn. A-2}$$

Solving [Equation A-1](#) and [Equation A-2](#) for K gives:

$$K = P_D \times (T_A + 273^\circ\text{C}) + \theta_{JA} \times (P_D)^2 \quad \text{Eqn. A-3}$$

where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A . Using this value of K, the values of P_D and T_J can be obtained by solving [Equation A-1](#) and [Equation A-2](#) iteratively for any value of T_A .

- ¹ Typical values are based on characterization data at 25°C. See Figure A-5 through Figure A-7 for typical curves across voltage/temperature.
- ² Max values in this column apply for the full operating temperature range of the device unless otherwise noted.
- ³ All modules except ADC active, ICS configured for FBE, and does not include any dc loads on port pins.
- ⁴ All modules except ADC active, ICS configured for FEI, and does not include any dc loads on port pins.
- ⁵ Stop currents are tested in production for 25°C on all parts. Tests at other temperatures depend upon the part number suffix and maturity of the product. Freescale may eliminate a test insertion at a particular temperature from the production test flow once sufficient data has been collected and is approved.
- ⁶ Most customers are expected to find that auto-wakeup from stop2 or stop3 can be used instead of the higher current wait mode.
- ⁷ Values given under the following conditions: low range operation (RANGE = 0) with a 32.768kHz crystal and low power mode (HGO = 0).

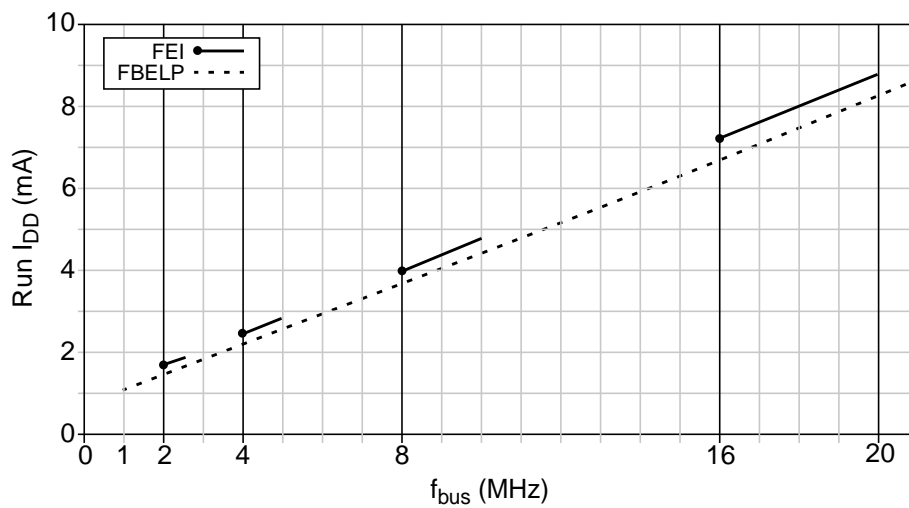


Figure A-5. Typical Run I_{DD} vs. Bus Frequency ($V_{DD} = 5V$)

A.12.3 SPI

Table A-15 and Figure A-14 through Figure A-17 describe the timing requirements for the SPI system.

Table A-15. SPI Electrical Characteristic

Num ¹	C	Rating ²	Symbol	Min	Max	Unit	
1	D	Cycle time	Master	t_{SCK}	2	2048	t_{cyc}
			Slave	t_{SCK}	4	—	t_{cyc}
2	D	Enable lead time	Master	t_{Lead}	—	1/2	t_{SCK}
			Slave	t_{Lead}	1/2	—	t_{SCK}
3	D	Enable lag time	Master	t_{Lag}	—	1/2	t_{SCK}
			Slave	t_{Lag}	1/2	—	t_{SCK}
4	D	Clock (SPSCK) high time Master and Slave	t_{SCKH}	$1/2 t_{SCK} - 25$	—	ns	
5	D	Clock (SPSCK) low time Master and Slave	t_{SCKL}	$1/2 t_{SCK} - 25$	—	ns	
6	D	Data setup time (inputs)	Master	$t_{SI(M)}$	30	—	ns
			Slave	$t_{SI(S)}$	30	—	ns
7	D	Data hold time (inputs)	Master	$t_{HI(M)}$	30	—	ns
			Slave	$t_{HI(S)}$	30	—	ns
8	D	Access time, slave ³	t_A	0	40	ns	
9	D	Disable time, slave ⁴	t_{dis}	—	40	ns	
10	D	Data setup time (outputs)	Master	t_{SO}	25	—	ns
			Slave	t_{SO}	25	—	ns
11	D	Data hold time (outputs)	Master	t_{HO}	-10	—	ns
			Slave	t_{HO}	-10	—	ns
12	D	Operating frequency	Master	f_{op}	$f_{Bus}/2048$	5^5	MHz
			Slave	f_{op}	dc	$f_{Bus}/4$	

¹ Refer to Figure A-14 through Figure A-17.

² All timing is shown with respect to 20% V_{DD} and 70% V_{DD} , unless noted; 100 pF load on all SPI pins. All timing assumes slew rate control disabled and high drive strength enabled for SPI output pins.

³ Time to data active from high-impedance state.

⁴ Hold time to high-impedance state.

⁵ Maximum baud rate must be limited to 5 MHz due to input filter characteristics.