

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, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	34
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K 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	44-LQFP
Supplier Device Package	44-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08aw60mfge



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. For your convenience, the page number designators have been linked to the appropriate location.

Revision Number	Revision Date	Description of Changes
1	1/2006	Initial external release.
2	12/2006	Includes KBI block changes; new V_{OL} / I_{OL} figures; R_{IDD} spec changes; SC part numbers with ICG trim modifications; addition of Temp Sensor to ADC. Resolved the stop IDD issues, added RTI figure, bandgap information, and incorporated electricals edits and any ProjectSync issues.

This product incorporates SuperFlash[®] technology licensed from SST.

Freescale[™] and the Freescale logo are trademarks of Freescale Semiconductor, Inc.
© Freescale Semiconductor, Inc., 2006. All rights reserved.

Table 3-4. Stop Mode Behavior (continued)

Peripheral	Mode	
	Stop2	Stop3
KBI	Off	Optionally On ³
RTI	Optionally On ⁴	Optionally On ⁴
SCI	Off	Standby
SPI	Off	Standby
TPM	Off	Standby
Voltage Regulator	Standby	Standby
I/O Pins	States Held	States Held

- ¹ Requires the asynchronous ADC clock and LVD to be enabled, else in standby.
- ² OSCSTEN set in ICSC1, else in standby. For high frequency range (RANGE in ICSC2 set) requires the LVD to also be enabled in stop3.
- ³ During stop3, KBI pins that are enabled continue to function as interrupt sources that are capable of waking the MCU from stop3.
- ⁴ This RTI can be enabled to run in stop2 or stop3 with the internal RTI clock source (RTICLK = 0, in SRTISC). The RTI also can be enabled to run in stop3 with the external clock source (RTICLK = 1 and OSCSTEN = 1).

Table 4-2. Direct-Page Register Summary (Sheet 1 of 3)

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
\$0000	PTAD	PTAD7	PTAD6	PTAD5	PTAD4	PTAD3	PTAD2	PTAD1	PTAD0
\$0001	PTADD	PTADD7	PTADD6	PTADD5	PTADD4	PTADD3	PTADD2	PTADD1	PTADD0
\$0002	PTBD	PTBD7	PTBD6	PTBD5	PTBD4	PTBD3	PTBD2	PTBD1	PTBD0
\$0003	PTBDD	PTBDD7	PTBDD6	PTBDD5	PTBDD4	PTBDD3	PTBDD2	PTBDD1	PTBDD0
\$0004	PTCD	0	PTCD6	PTCD5	PTCD4	PTCD3	PTCD2	PTCD1	PTCD0
\$0005	PTCDD	0	PTCDD6	PTCDD5	PTCDD4	PTCDD3	PTCDD2	PTCDD1	PTCDD0
\$0006	PTDD	PTDD7	PTDD6	PTDD5	PTDD4	PTDD3	PTDD2	PTDD1	PTDD0
\$0007	PTDDD	PTDDD7	PTDDD6	PTDDD5	PTDDD4	PTDDD3	PTDDD2	PTDDD1	PTDDD0
\$0008	PTED	PTED7	PTED6	PTED5	PTED4	PTED3	PTED2	PTED1	PTED0
\$0009	PTEDD	PTEDD7	PTEDD6	PTEDD5	PTEDD4	PTEDD3	PTEDD2	PTEDD1	PTEDD0
\$000A	PTFD	PTFD7	PTFD6	PTFD5	PTFD4	PTFD3	PTFD2	PTFD1	PTFD0
\$000B	PTFDD	PTFDD7	PTFDD6	PTFDD5	PTFDD4	PTFDD3	PTFDD2	PTFDD1	PTFDD0
\$000C	PTGD	0	PTGD6	PTGD5	PTGD4	PTGD3	PTGD2	PTGD1	PTGD0
\$000D	PTGDD	0	PTGDD6	PTGDD5	PTGDD4	PTGDD3	PTGDD2	PTGDD1	PTGDD0
\$000E– \$000F	Reserved	—	—	—	—	—	—	—	—
\$0010	ADC1SC1	COCO	AIEN	ADCO	ADCH				
\$0011	ADC1SC2	ADACT	ADTRG	ACFE	ACFGT	0	0	R	R
\$0012	ADC1RH	0	0	0	0	0	0	ADR9	ADR8
\$0013	ADC1RL	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
\$0014	ADC1CVH	0	0	0	0	0	0	ADCV9	ADCV8
\$0015	ADC1CVL	ADCV7	ADCV6	ADCV5	ADCV4	ADCV3	ADCV2	ADCV1	ADCV0
\$0016	ADC1CFG	ADLPC	ADIV		ADLSMP	MODE		ADICLK	
\$0017	APCTL1	ADPC7	ADPC6	ADPC5	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0
\$0018	APCTL2	ADPC15	ADPC14	ADPC13	ADPC12	ADPC11	ADPC10	ADPC9	ADPC8
\$0019	APCTL3	ADPC23	ADPC22	ADPC21	ADPC20	ADPC19	ADPC18	ADPC17	ADPC16
\$001A– \$001B	Reserved	—	—	—	—	—	—	—	—
\$001C	IRQSC	0	0	IRQEDG	IRQPE	IRQF	IRQACK	IRQIE	IRQMOD
\$001D	Reserved	—	—	—	—	—	—	—	—
\$001E	KBI1SC	KBEDG7	KBEDG6	KBEDG5	KBEDG4	KBF	KBACK	KBIE	KBIMOD
\$001F	KBI1PE	KBIPE7	KBIPE6	KBIPE5	KBIPE4	KBIPE3	KBIPE2	KBIPE1	KBIPE0
\$0020	TPM1SC	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
\$0021	TPM1CNTH	Bit 15	14	13	12	11	10	9	Bit 8
\$0022	TPM1CNTL	Bit 7	6	5	4	3	2	1	Bit 0
\$0023	TPM1MODH	Bit 15	14	13	12	11	10	9	Bit 8
\$0024	TPM1MODL	Bit 7	6	5	4	3	2	1	Bit 0
\$0025	TPM1C0SC	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
\$0026	TPM1C0VH	Bit 15	14	13	12	11	10	9	Bit 8
\$0027	TPM1C0VL	Bit 7	6	5	4	3	2	1	Bit 0

4.4.5 Access Errors

An access error occurs whenever the command execution protocol is violated.

Any of the following specific actions will cause the access error flag (FACCERR) in FSTAT to be set. FACCERR must be cleared by writing a 1 to FACCERR in FSTAT before any command can be processed.

- Writing to a FLASH address before the internal FLASH clock frequency has been set by writing to the FCDIV register
- Writing to a FLASH address while FCBEF is not set (A new command cannot be started until the command buffer is empty.)
- Writing a second time to a FLASH address before launching the previous command (There is only one write to FLASH for every command.)
- Writing a second time to FCMD before launching the previous command (There is only one write to FCMD for every command.)
- Writing to any FLASH control register other than FCMD after writing to a FLASH address
- Writing any command code other than the five allowed codes (\$05, \$20, \$25, \$40, or \$41) to FCMD
- Accessing (read or write) any FLASH control register other than the write to FSTAT (to clear FCBEF and launch the command) after writing the command to FCMD.
- The MCU enters stop mode while a program or erase command is in progress (The command is aborted.)
- Writing the byte program, burst program, or page erase command code (\$20, \$25, or \$40) with a background debug command while the MCU is secured (The background debug controller can only do blank check and mass erase commands when the MCU is secure.)
- Writing 0 to FCBEF to cancel a partial command

4.4.6 FLASH Block Protection

The block protection feature prevents the protected region of FLASH from program or erase changes. Block protection is controlled through the FLASH Protection Register (FPROT). When enabled, block protection begins at any 512 byte boundary below the last address of FLASH, \$FFFF. (see [Section 4.6.4, “FLASH Protection Register \(FPROT and NVPROT\)”](#)).

After exit from reset, FPROT is loaded with the contents of the NVPROT location which is in the nonvolatile register block of the FLASH memory. FPROT cannot be changed directly from application software so a runaway program cannot alter the block protection settings. Since NVPROT is within the last 512 bytes of FLASH, if any amount of memory is protected, NVPROT is itself protected and cannot be altered (intentionally or unintentionally) by the application software. FPROT can be written through background debug commands which allows a way to erase and reprogram a protected FLASH memory.

The block protection mechanism is illustrated below. The FPS bits are used as the upper bits of the last address of unprotected memory. This address is formed by concatenating FPS7:FPS1 with logic 1 bits as shown. For example, in order to protect the last 8192 bytes of memory (addresses \$E000 through \$FFFF), the FPS bits must be set to 1101 111 which results in the value \$DFFF as the last address of unprotected memory. In addition to programming the FPS bits to the appropriate value, FPDIS (bit 0 of NVPROT) must

Table 5-1. Vector Summary

Vector Priority	Vector Number	Address (High/Low)	Vector Name	Module	Source	Enable	Description	
Lower  Higher	26 through 31	\$FFC0/FFC1 through \$FFCA/FFCB	Unused Vector Space (available for user program)					
	25	\$FFCC/FFCD	Vrti	System control	RTIF	RTIE	Real-time interrupt	
	24	\$FFCE/FFCF	Viiic1	IIC1	IICIF	IICIE	IIC1	
	23	\$FFD0/FFD1	Vadc1	ADC1	COCO	AIEN	ADC1	
	22	\$FFD2/FFD3	Vkeyboard 1	KBI1	KBF	KBIE	KBI1 pins	
	21	\$FFD4/FFD5	Vsci2tx	SCI2	TDRE TC	TIE TCIE	SCI2 transmit	
	20	\$FFD6/FFD7	Vsci2rx	SCI2	IDLE RDRF	ILIE RIE	SCI2 receive	
	19	\$FFD8/FFD9	Vsci2err	SCI2	OR NF FE PF	ORIE NFIE FEIE PFIE	SCI2 error	
	18	\$FFDA/FFDB	Vsci1tx	SCI1	TDRE TC	TIE TCIE	SCI1 transmit	
	17	\$FFDC/FFDD	Vsci1rx	SCI1	IDLE RDRF	ILIE RIE	SCI1 receive	
	16	\$FFDE/FFDF	Vsci1err	SCI1	OR NF FE PF	ORIE NFIE FEIE PFIE	SCI1 error	
	15	\$FFE0/FFE1	Vspi1	SPI1	SPIF MODF SPTEF	SPIE SPIE SPTIE	SPI1	
	14	\$FFE2/FFE3	Vtpm2ovf	TPM2	TOF	TOIE	TPM2 overflow	
	13	\$FFE4/FFE5	Vtpm2ch1	TPM2	CH1F	CH1IE	TPM2 channel 1	
	12	\$FFE6/FFE7	Vtpm2ch0	TPM2	CH0F	CH0IE	TPM2 channel 0	
	11	\$FFE8/FFE9	Vtpm1ovf	TPM1	TOF	TOIE	TPM1 overflow	
	10	\$FFEA/FFEB	Vtpm1ch5	TPM1	CH5F	CH5IE	TPM1 channel 5	
	9	\$FFEC/FFED	Vtpm1ch4	TPM1	CH4F	CH4IE	TPM1 channel 4	
	8	\$FFEE/FFEF	Vtpm1ch3	TPM1	CH3F	CH3IE	TPM1 channel 3	
	7	\$FFF0/FFF1	Vtpm1ch2	TPM1	CH2F	CH2IE	TPM1 channel 2	
	6	\$FFF2/FFF3	Vtpm1ch1	TPM1	CH1F	CH1IE	TPM1 channel 1	
	5	\$FFF4/FFF5	Vtpm1ch0	TPM1	CH0F	CH0IE	TPM1 channel 0	
	4	\$FFF6/FFF7	Vicg	ICG	ICGIF (LOLS/LOCS)	LOLRE/LOCRE	ICG	
	3	\$FFF8/FFF9	Vlvd	System control	LVDF	LVDIE	Low-voltage detect	
	2	\$FFFA/FFFB	Virq	IRQ	IRQF	IRQIE	IRQ pin	
	1	\$FFFC/FFFD	Vswi	Core	SWI Instruction	—	Software interrupt	
	0	\$FFFE/FFFF	Vreset	System control	COP LVD RESET pin Illegal opcode	COPE LVDRE — —	Watchdog timer Low-voltage detect External pin Illegal opcode	

Port B pins are general-purpose I/O pins. Parallel I/O function is controlled by the port B data (PTBD) and data direction (PTBDD) registers which are located in page zero register space. The pin control registers, pullup enable (PTBPE), slew rate control (PTBSE), and drive strength select (PTBDS) are located in the high page registers. Refer to [Section 6.4, “Parallel I/O Control”](#) for more information about general-purpose I/O control and [Section 6.5, “Pin Control”](#) for more information about pin control.

Port B general-purpose I/O are shared with the ADC. Any pin enabled as an ADC input will have the general-purpose I/O function disabled. Refer to [Chapter 14, “Analog-to-Digital Converter \(S08ADC10V1\)”](#) for more information about using port B as analog inputs.

6.3.3 Port C

Port C	Bit 7	6	5	3	3	2	1	Bit 0
MCU Pin:		PTC6	PTC5/ RxD2	PTC4	PTC3/ TxD2	PTC2/ MCLK	PTC1/ SDA1	PTC0/ SCL1

Figure 6-3. Port C Pin Names

Port C pins are general-purpose I/O pins. Parallel I/O function is controlled by the port C data (PTCD) and data direction (PTCDD) registers which are located in page zero register space. The pin control registers, pullup enable (PTCPE), slew rate control (PTCSE), and drive strength select (PTCDS) are located in the high page registers. Refer to [Section 6.4, “Parallel I/O Control”](#) for more information about general-purpose I/O control and [Section 6.5, “Pin Control”](#) for more information about pin control.

Port C general-purpose I/O is shared with SCI2, IIC, and MCLK. When any of these shared functions is enabled, the direction, input or output, is controlled by the shared function and not by the data direction register of the parallel I/O port. Also, for pins which are configured as outputs by the shared function, the output data is controlled by the shared function and not by the port data register.

Refer to [Chapter 11, “Serial Communications Interface \(S08SCIV2\)”](#) for more information about using port C pins as SCI pins.

Refer to [Chapter 13, “Inter-Integrated Circuit \(S08IICV1\)”](#) for more information about using port C pins as IIC pins.

Refer to [Chapter 5, “Resets, Interrupts, and System Configuration”](#) for more information about using PTC2 as the MCLK pin.

6.3.4 Port D

Port D	Bit 7	6	5	4	3	2	1	Bit 0
MCU Pin:	PTD7/ AD1P15/ KBI1P7	PTD6/ AD1P14/ TPM1CLK	PTD5/ AD1P13/	PTD4/ AD1P12/ TPM2CLK	PTD3/ AD1P11/ KBI1P6	PTD2/ AD1P10/ KBI1P5	PTD1/ AD1P9	PTD0/ AD1P8

Figure 6-4. Port D Pin Names

Port D pins are general-purpose I/O pins. Parallel I/O function is controlled by the port D data (PTDD) and data direction (PTDDD) registers which are located in page zero register space. The pin control registers,

6.7.4 Port B Pin Control Registers (PTBPE, PTBSE, PTBDS)

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

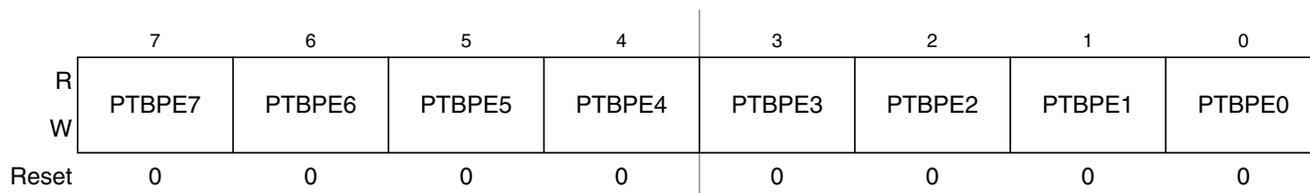


Figure 6-16. Internal Pullup Enable for Port B (PTBPE)

Table 6-9. PTBPE Register Field Descriptions

Field	Description
7:0 PTBPE[7:0]	<p>Internal Pullup Enable for Port B Bits — Each of these control bits determines if the internal pullup 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 pullup devices are disabled.</p> <p>0 Internal pullup device disabled for port B bit n. 1 Internal pullup device enabled for port B bit n.</p>

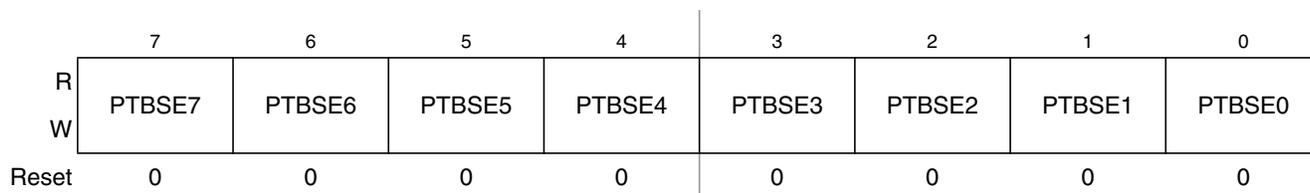


Figure 6-17. Output Slew Rate Control Enable (PTBSE)

Table 6-10. PTBSE Register Field Descriptions

Field	Description
7:0 PTBSE[7:0]	<p>Output Slew Rate Control Enable for Port B Bits— Each of these control bits determine whether 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>

Table 7-2. HCS08 Instruction Set Summary (Sheet 6 of 7)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Bus Cycles ¹
			V	H	I	N	Z	C				
ROR <i>opr8a</i> RORA RORX ROR <i>opr8,X</i> ROR <i>,X</i> ROR <i>opr8,SP</i>	Rotate Right through Carry		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	36 46 56 66 76 9E66	dd ff ff	5 1 1 5 4 6
RSP	Reset Stack Pointer	SP ← 0xFF (High Byte Not Affected)	–	–	–	–	–	–	INH	9C		1
RTI	Return from Interrupt	SP ← (SP) + 0x0001; Pull (CCR) SP ← (SP) + 0x0001; Pull (A) SP ← (SP) + 0x0001; Pull (X) SP ← (SP) + 0x0001; Pull (PCH) SP ← (SP) + 0x0001; Pull (PCL)	↕	↕	↕	↕	↕	↕	INH	80		9
RTS	Return from Subroutine	SP ← SP + 0x0001; Pull (PCH) SP ← SP + 0x0001; Pull (PCL)	–	–	–	–	–	–	INH	81		6
SBC <i>#opr8i</i> SBC <i>opr8a</i> SBC <i>opr16a</i> SBC <i>opr16,X</i> SBC <i>opr8,X</i> SBC <i>,X</i> SBC <i>opr16,SP</i> SBC <i>opr8,SP</i>	Subtract with Carry	A ← (A) – (M) – (C)	↕	–	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP2 SP1	A2 B2 C2 D2 E2 F2 9ED2 9EE2	ii dd hh ll ee ff ff ff ff ee ff ff	2 3 4 4 3 3 5 4
SEC	Set Carry Bit	C ← 1	–	–	–	–	–	1	INH	99		1
SEI	Set Interrupt Mask Bit	I ← 1	–	–	1	–	–	–	INH	9B		1
STA <i>opr8a</i> STA <i>opr16a</i> STA <i>opr16,X</i> STA <i>opr8,X</i> STA <i>,X</i> STA <i>opr16,SP</i> STA <i>opr8,SP</i>	Store Accumulator in Memory	M ← (A)	0	–	–	↕	↕	–	DIR EXT IX2 IX1 IX SP2 SP1	B7 C7 D7 E7 F7 9ED7 9EE7	dd hh ll ee ff ff ff ee ff ff	3 4 4 3 2 5 4
STHX <i>opr8a</i> STHX <i>opr16a</i> STHX <i>opr8,SP</i>	Store H:X (Index Reg.)	(M:M + 0x0001) ← (H:X)	0	–	–	↕	↕	–	DIR EXT SP1	35 96 9EFF	dd hh ll ff	4 5 5
STOP	Enable Interrupts: Stop Processing Refer to MCU Documentation	I bit ← 0; Stop Processing	–	–	0	–	–	–	INH	8E		2+
STX <i>opr8a</i> STX <i>opr16a</i> STX <i>opr16,X</i> STX <i>opr8,X</i> STX <i>,X</i> STX <i>opr16,SP</i> STX <i>opr8,SP</i>	Store X (Low 8 Bits of Index Register) in Memory	M ← (X)	0	–	–	↕	↕	–	DIR EXT IX2 IX1 IX SP2 SP1	BF CF DF EF FF 9EDF 9EEF	dd hh ll ee ff ff ff ee ff ff	3 4 4 3 2 5 4
SUB <i>#opr8i</i> SUB <i>opr8a</i> SUB <i>opr16a</i> SUB <i>opr16,X</i> SUB <i>opr8,X</i> SUB <i>,X</i> SUB <i>opr16,SP</i> SUB <i>opr8,SP</i>	Subtract	A ← (A) – (M)	↕	–	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP2 SP1	A0 B0 C0 D0 E0 F0 9ED0 9EE0	ii dd hh ll ee ff ff ff ee ff ff	2 3 4 4 3 3 5 4
SWI	Software Interrupt	PC ← (PC) + 0x0001 Push (PCL); SP ← (SP) – 0x0001 Push (PCH); SP ← (SP) – 0x0001 Push (X); SP ← (SP) – 0x0001 Push (A); SP ← (SP) – 0x0001 Push (CCR); SP ← (SP) – 0x0001 I ← 1; PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte	–	–	1	–	–	–	INH	83		11

8.3.3 ICG Status Register 1 (ICGS1)

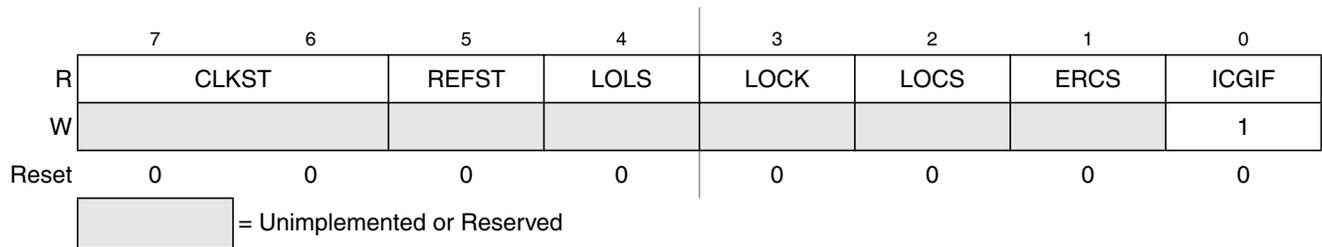


Figure 8-8. ICG Status Register 1 (ICGS1)

Table 8-3. ICGS1 Register Field Descriptions

Field	Description
7:6 CLKST	<p>Clock Mode Status — The CLKST bits indicate the current clock mode. The CLKST bits don't update immediately after a write to the CLKS bits due to internal synchronization between clock domains.</p> <p>00 Self-clocked 01 FLL engaged, internal reference 10 FLL bypassed, external reference 11 FLL engaged, external reference</p>
5 REFST	<p>Reference Clock Status — The REFST bit indicates which clock reference is currently selected by the Reference Select circuit.</p> <p>0 External Clock selected. 1 Crystal/Resonator selected.</p>
4 LOLS	<p>FLL Loss of Lock Status — The LOLS bit is an indication of FLL lock status.</p> <p>0 FLL has not unexpectedly lost lock since LOLS was last cleared. 1 FLL has unexpectedly lost lock since LOLS was last cleared, LOLRE determines action taken.</p>
3 LOCK	<p>FLL Lock Status — The LOCK bit indicates whether the FLL has acquired lock. The LOCK bit is cleared in off, self-clocked, and FLL bypassed modes.</p> <p>0 FLL is currently unlocked. 1 FLL is currently locked.</p>
2 LOCS	<p>Loss Of Clock Status — The LOCS bit is an indication of ICG loss of clock status.</p> <p>0 ICG has not lost clock since LOCS was last cleared. 1 ICG has lost clock since LOCS was last cleared, LOCRE determines action taken.</p>
1 ERCS	<p>External Reference Clock Status — The ERCS bit is an indication of whether or not the external reference clock (ICGERCLK) meets the minimum frequency requirement.</p> <p>0 External reference clock is not stable, frequency requirement is not met. 1 External reference clock is stable, frequency requirement is met.</p>
0 ICGIF	<p>ICG Interrupt Flag — The ICGIF read/write flag is set when an ICG interrupt request is pending. It is cleared by a reset or by reading the ICG status register when ICGIF is set and then writing a logic 1 to ICGIF. If another ICG interrupt occurs before the clearing sequence is complete, the sequence is reset so ICGIF would remain set after the clear sequence was completed for the earlier interrupt. Writing a logic 0 to ICGIF has no effect.</p> <p>0 No ICG interrupt request is pending. 1 An ICG interrupt request is pending.</p>

9.4.1 KBI Status and Control Register (KBI1SC)

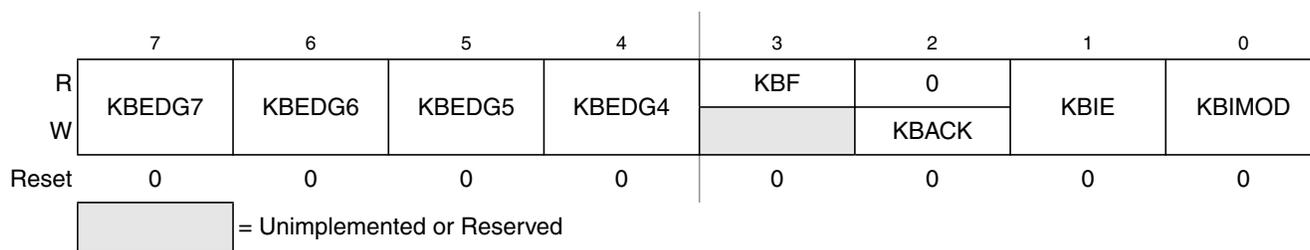


Figure 9-3. KBI Status and Control Register (KBI1SC)

Table 9-2. KBI1SC Register Field Descriptions

Field	Description
7:4 KBEDG[7:4]	Keyboard Edge Select for KBI Port Bits — Each of these read/write bits selects the polarity of the edges and/or levels that are recognized as trigger events on the corresponding KBI port pin when it is configured as a keyboard interrupt input (KBIPEn = 1). Also see the KBIMOD control bit, which determines whether the pin is sensitive to edges-only or edges and levels. 0 Falling edges/low levels 1 Rising edges/high levels
3 KBF	Keyboard Interrupt Flag — This read-only status flag is set whenever the selected edge event has been detected on any of the enabled KBI port pins. This flag is cleared by writing a 1 to the KBACK control bit. The flag will remain set if KBIMOD = 1 to select edge-and-level operation and any enabled KBI port pin remains at the asserted level. KBF can be used as a software pollable flag (KBIE = 0) or it can generate a hardware interrupt request to the CPU (KBIE = 1). 0 No KBI interrupt pending 1 KBI interrupt pending
2 KBACK	Keyboard Interrupt Acknowledge — This write-only bit (reads always return 0) is used to clear the KBF status flag by writing a 1 to KBACK. When KBIMOD = 1 to select edge-and-level operation and any enabled KBI port pin remains at the asserted level, KBF is being continuously set so writing 1 to KBACK does not clear the KBF flag.
1 KBIE	Keyboard Interrupt Enable — This read/write control bit determines whether hardware interrupts are generated when the KBF status flag equals 1. When KBIE = 0, no hardware interrupts are generated, but KBF can still be used for software polling. 0 KBF does not generate hardware interrupts (use polling) 1 KBI hardware interrupt requested when KBF = 1
KBIMOD	Keyboard Detection Mode — This read/write control bit selects either edge-only detection or edge-and-level detection. KBI port bits 3 through 0 can detect falling edges-only or falling edges and low levels. KBI port bits 7 through 4 can be configured to detect either: <ul style="list-style-type: none"> Rising edges-only or rising edges and high levels (KBEDGn = 1) Falling edges-only or falling edges and low levels (KBEDGn = 0) 0 Edge-only detection 1 Edge-and-level detection

10.2.1 Features

The TPM has the following features:

- Each TPM may be configured for buffered, center-aligned pulse-width modulation (CPWM) on all channels
- Clock sources independently selectable per TPM (multiple TPMs device)
- Selectable clock sources (device dependent): bus clock, fixed system clock, external pin
- Clock prescaler taps for divide by 1, 2, 4, 8, 16, 32, 64, or 128
- 16-bit free-running or up/down (CPWM) count operation
- 16-bit modulus register to control counter range
- Timer system enable
- One interrupt per channel plus a terminal count interrupt for each TPM module (multiple TPMs device)
- Channel features:
 - Each channel may be input capture, output compare, or buffered edge-aligned PWM
 - Rising-edge, falling-edge, or any-edge input capture trigger
 - Set, clear, or toggle output compare action
 - Selectable polarity on PWM outputs

10.2.2 Block Diagram

Figure 10-2 shows the structure of a TPM. Some MCUs include more than one TPM, with various numbers of channels.

Table 11-5. SC1xS1 Register Field Descriptions (continued)

Field	Description
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). In 8-bit mode, to clear RDRF, read SC1xS1 with RDRF = 1 and then read the SCI data register (SC1xD). In 9-bit mode, to clear RDRF, read SC1xS1 with RDRF = 1 and then read SC1xD and the SCI control 3 register (SC1xC3). SC1xD and SC1xC3 can be read in any order, but the flag is cleared only after both data registers are read.</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>
1 FE	<p>Framing Error Flag — FE is set at the same time as RDRF when the receiver detects a logic 0 where the stop bit was expected. This suggests the receiver was not properly aligned to a character frame. To clear FE, read SC1xS1 with FE = 1 and then read the SCI data register (SC1xD).</p> <p>0 No framing error detected. This does not guarantee the framing is correct. 1 Framing error.</p>
0 PF	<p>Parity Error Flag — PF is set at the same time as RDRF when parity is enabled (PE = 1) and the parity bit in the received character does not agree with the expected parity value. To clear PF, read SC1xS1 and then read the SCI data register (SC1xD).</p> <p>0 No parity error. 1 Parity error.</p>

systems with modems to determine when it is safe to turn off the modem. If the transmit complete interrupt enable (TCIE) bit is set, a hardware interrupt will be requested whenever $TC = 1$. Instead of hardware interrupts, software polling may be used to monitor the TDRE and TC status flags if the corresponding TIE or TCIE local interrupt masks are 0s.

When a program detects that the receive data register is full ($RDRF = 1$), it gets the data from the receive data register by reading SCIxD. The RDRF flag is cleared by reading SCIxS1 while $RDRF = 1$ and then reading SCIxD. If the SCI is configured to operate in 9-bit mode, an additional read to the SCIxC3 register is required to clear RDRF.

When polling is used, this sequence is naturally satisfied in the normal course of the user program. If hardware interrupts are used, SCIxS1 must be read in the interrupt service routine (ISR). Normally, this is done in the ISR anyway to check for receive errors, so the sequence is automatically satisfied.

The IDLE status flag includes logic that prevents it from getting set repeatedly when the RxD line remains idle for an extended period of time. IDLE is cleared by reading SCIxS1 while $IDLE = 1$ and then reading SCIxD. After IDLE has been cleared, it cannot become set again until the receiver has received at least one new character and has set RDRF.

If the associated error was detected in the received character that caused RDRF to be set, the error flags — noise flag (NF), framing error (FE), and parity error flag (PF) — get set at the same time as RDRF. These flags are not set in overrun cases.

If RDRF was already set when a new character is ready to be transferred from the receive shifter to the receive data buffer, the overrun (OR) flag gets set instead and the data and any associated NF, FE, or PF condition is lost.

11.3.5 Additional SCI Functions

The following sections describe additional SCI functions.

11.3.5.1 8- and 9-Bit Data Modes

The SCI system (transmitter and receiver) can be configured to operate in 9-bit data mode by setting the M control bit in SCIxC1. In 9-bit mode, there is a ninth data bit to the left of the MSB of the SCI data register. For the transmit data buffer, this bit is stored in T8 in SCIxC3. For the receiver, the ninth bit is held in R8 in SCIxC3.

When transmitting 9-bit data, write to the T8 bit before writing to SCIxD for coherent writes to the transmit data buffer. If the bit value to be transmitted as the ninth bit of a new character is the same as for the previous character, it is not necessary to write to T8 again. When data is transferred from the transmit data buffer to the transmit shifter, the value in T8 is copied at the same time data is transferred from SCIxD to the shifter.

When receiving 9-bit data, clear the RDRF bit by reading both R8 and SCIxD. R8 and SCIxD can be read in either order.

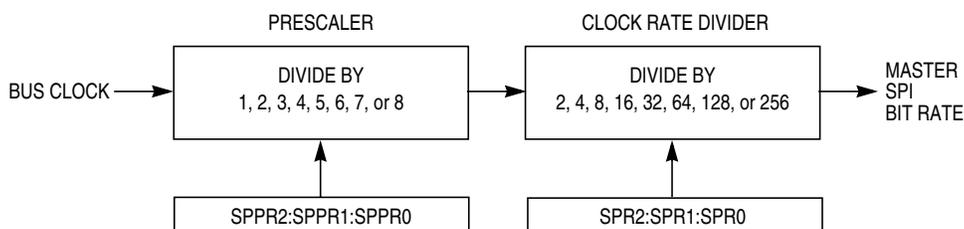


Figure 12-4. SPI Baud Rate Generation

12.1 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.

12.1.1 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.

12.1.2 MOSI — Master Data Out, Slave Data In

When the SPI is enabled as a master and SPI pin control zero ($SPC0$) is 0 (not bidirectional mode), this pin is the serial data output. When the SPI is enabled as a slave and $SPC0 = 0$, this pin is the serial data input. If $SPC0 = 1$ to select single-wire bidirectional mode, and master mode is selected, this pin becomes the bidirectional data I/O pin (MOMI). Also, the bidirectional mode output enable bit determines whether the pin acts as an input ($BIDIROE = 0$) or an output ($BIDIROE = 1$). If $SPC0 = 1$ and slave mode is selected, this pin is not used by the SPI and reverts to being a general-purpose port I/O pin.

12.1.3 MISO — Master Data In, Slave Data Out

When the SPI is enabled as a master and SPI pin control zero ($SPC0$) is 0 (not bidirectional mode), this pin is the serial data input. When the SPI is enabled as a slave and $SPC0 = 0$, this pin is the serial data output. If $SPC0 = 1$ to select single-wire bidirectional mode, and slave mode is selected, this pin becomes the bidirectional data I/O pin (SISO) and the bidirectional mode output enable bit determines whether the pin acts as an input ($BIDIROE = 0$) or an output ($BIDIROE = 1$). If $SPC0 = 1$ and master mode is selected, this pin is not used by the SPI and reverts to being a general-purpose port I/O pin.

12.1.4 \overline{SS} — Slave Select

When the SPI is enabled as a slave, this pin is the low-true slave select input. When the SPI is enabled as a master and mode fault enable is off ($MODFEN = 0$), this pin is not used by the SPI and reverts to being a general-purpose port I/O pin. When the SPI is enabled as a master and $MODFEN = 1$, the slave select output enable bit determines whether this pin acts as the mode fault input ($SSOE = 0$) or as the slave select output ($SSOE = 1$).

pin from a master and the MISO waveform applies to the MISO output from a slave. The \overline{SS} OUT waveform applies to the slave select output from a master (provided MODFEN and SSOE = 1). The master \overline{SS} output goes to active low one-half SPSCK cycle before the start of the transfer and goes back high at the end of the eighth bit time of the transfer. The \overline{SS} IN waveform applies to the slave select input of a slave.

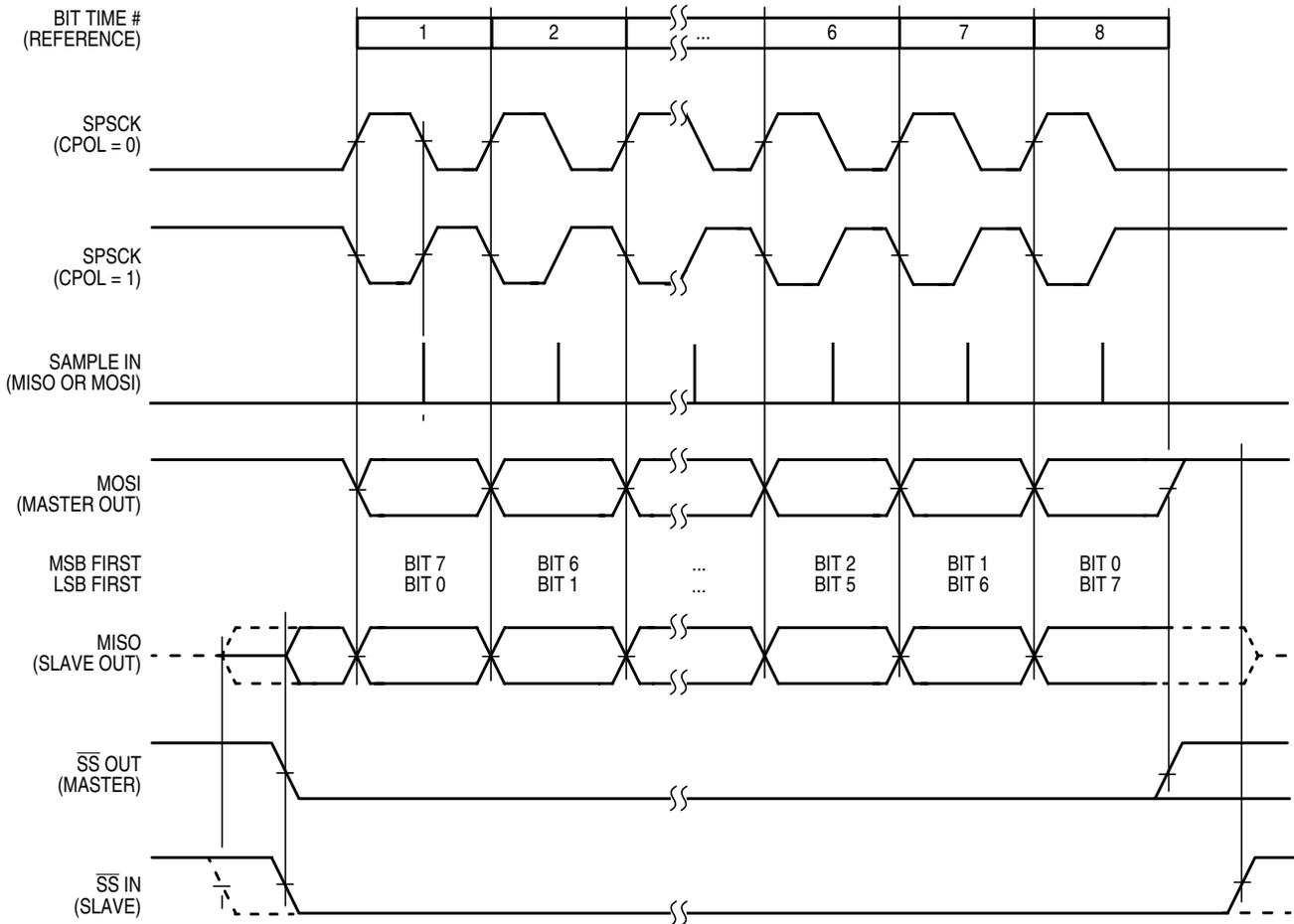


Figure 12-10. SPI Clock Formats (CPHA = 1)

When CPHA = 1, the slave begins to drive its MISO output when \overline{SS} goes to active low, but the data is not defined until the first SPSCK edge. The first SPSCK edge shifts the first bit of data from the shifter onto the MOSI output of the master and the MISO output of the slave. The next SPSCK edge causes both the master and the slave to sample the data bit values on their MISO and MOSI inputs, respectively. At the third SPSCK edge, the SPI shifter shifts one bit position which shifts in the bit value that was just sampled, and shifts the second data bit value out the other end of the shifter to the MOSI and MISO outputs of the master and slave, respectively. When CHPA = 1, the slave's \overline{SS} input is not required to go to its inactive high level between transfers.

Figure 12-11 shows the clock formats when CPHA = 0. At the top of the figure, the eight bit times are shown for reference with bit 1 starting as the slave is selected (\overline{SS} IN goes low), and bit 8 ends at the last SPSCK edge. The MSB first and LSB first lines show the order of SPI data bits depending on the setting

Table 14-9. APCTL1 Register Field Descriptions (continued)

Field	Description
1 ADPC1	ADC Pin Control 1 — ADPC1 is used to control the pin associated with channel AD1. 0 AD1 pin I/O control enabled 1 AD1 pin I/O control disabled
0 ADPC0	ADC Pin Control 0 — ADPC0 is used to control the pin associated with channel AD0. 0 AD0 pin I/O control enabled 1 AD0 pin I/O control disabled

14.4.9 Pin Control 2 Register (APCTL2)

APCTL2 is used to control channels 8–15 of the ADC module.

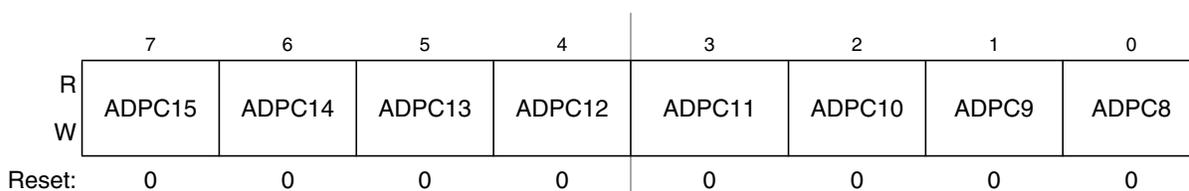


Figure 14-12. Pin Control 2 Register (APCTL2)

Table 14-10. APCTL2 Register Field Descriptions

Field	Description
7 ADPC15	ADC Pin Control 15 — ADPC15 is used to control the pin associated with channel AD15. 0 AD15 pin I/O control enabled 1 AD15 pin I/O control disabled
6 ADPC14	ADC Pin Control 14 — ADPC14 is used to control the pin associated with channel AD14. 0 AD14 pin I/O control enabled 1 AD14 pin I/O control disabled
5 ADPC13	ADC Pin Control 13 — ADPC13 is used to control the pin associated with channel AD13. 0 AD13 pin I/O control enabled 1 AD13 pin I/O control disabled
4 ADPC12	ADC Pin Control 12 — ADPC12 is used to control the pin associated with channel AD12. 0 AD12 pin I/O control enabled 1 AD12 pin I/O control disabled
3 ADPC11	ADC Pin Control 11 — ADPC11 is used to control the pin associated with channel AD11. 0 AD11 pin I/O control enabled 1 AD11 pin I/O control disabled
2 ADPC10	ADC Pin Control 10 — ADPC10 is used to control the pin associated with channel AD10. 0 AD10 pin I/O control enabled 1 AD10 pin I/O control disabled

In cases where separate power supplies are used for analog and digital power, the ground connection between these supplies must be at the V_{SSAD} pin. This should be the only ground connection between these supplies if possible. The V_{SSAD} pin makes a good single point ground location.

14.7.1.2 Analog Reference Pins

In addition to the analog supplies, the ADC module has connections for two reference voltage inputs. The high reference is V_{REFH} , which may be shared on the same pin as V_{DDAD} on some devices. The low reference is V_{REFL} , which may be shared on the same pin as V_{SSAD} on some devices.

When available on a separate pin, V_{REFH} may be connected to the same potential as V_{DDAD} , or may be driven by an external source that is between the minimum V_{DDAD} spec and the V_{DDAD} potential (V_{REFH} must never exceed V_{DDAD}). When available on a separate pin, V_{REFL} must be connected to the same voltage potential as V_{SSAD} . Both V_{REFH} and V_{REFL} must be routed carefully for maximum noise immunity and bypass capacitors placed as near as possible to the package.

AC current in the form of current spikes required to supply charge to the capacitor array at each successive approximation step is drawn through the V_{REFH} and V_{REFL} loop. The best external component to meet this current demand is a 0.1 μF capacitor with good high frequency characteristics. This capacitor is connected between V_{REFH} and V_{REFL} and must be placed as near as possible to the package pins. Resistance in the path is not recommended because the current will cause a voltage drop which could result in conversion errors. Inductance in this path must be minimum (parasitic only).

14.7.1.3 Analog Input Pins

The external analog inputs are typically shared with digital I/O pins on MCU devices. The pin I/O control is disabled by setting the appropriate control bit in one of the pin control registers. Conversions can be performed on inputs without the associated pin control register bit set. It is recommended that the pin control register bit always be set when using a pin as an analog input. This avoids problems with contention because the output buffer will be in its high impedance state and the pullup is disabled. Also, the input buffer draws dc current when its input is not at either V_{DD} or V_{SS} . Setting the pin control register bits for all pins used as analog inputs should be done to achieve lowest operating current.

Empirical data shows that capacitors on the analog inputs improve performance in the presence of noise or when the source impedance is high. Use of 0.01 μF capacitors with good high-frequency characteristics is sufficient. These capacitors are not necessary in all cases, but when used they must be placed as near as possible to the package pins and be referenced to V_{SSA} .

For proper conversion, the input voltage must fall between V_{REFH} and V_{REFL} . If the input is equal to or exceeds V_{REFH} , the converter circuit converts the signal to \$3FF (full scale 10-bit representation) or \$FF (full scale 8-bit representation). If the input is equal to or less than V_{REFL} , the converter circuit converts it to \$000. Input voltages between V_{REFH} and V_{REFL} are straight-line linear conversions. There will be a brief current associated with V_{REFL} when the sampling capacitor is charging. The input is sampled for 3.5 cycles of the ADCK source when ADLSMP is low, or 23.5 cycles when ADLSMP is high.

For minimal loss of accuracy due to current injection, pins adjacent to the analog input pins should not be transitioning during conversions.

- Non-intrusive commands can be executed at any time even while the user's program is running. Non-intrusive commands allow a user to read or write MCU memory locations or access status and control registers within the background debug controller.

Typically, a relatively simple interface pod is used to translate commands from a host computer into commands for the custom serial interface to the single-wire background debug system. Depending on the development tool vendor, this interface pod may use a standard RS-232 serial port, a parallel printer port, or some other type of communications such as a universal serial bus (USB) to communicate between the host PC and the pod. The pod typically connects to the target system with ground, the BKGD pin, $\overline{\text{RESET}}$, and sometimes V_{DD} . An open-drain connection to reset allows the host to force a target system reset, which is useful to regain control of a lost target system or to control startup of a target system before the on-chip nonvolatile memory has been programmed. Sometimes V_{DD} can be used to allow the pod to use power from the target system to avoid the need for a separate power supply. However, if the pod is powered separately, it can be connected to a running target system without forcing a target system reset or otherwise disturbing the running application program.

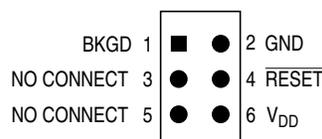


Figure 15-1. BDM Tool Connector

15.2.1 BKGD Pin Description

BKGD is the single-wire background debug interface pin. The primary function of this pin is for bidirectional serial communication of active background mode commands and data. During reset, this pin is used to select between starting in active background mode or starting the user's application program. This pin is also used to request a timed sync response pulse to allow a host development tool to determine the correct clock frequency for background debug serial communications.

BDC serial communications use a custom serial protocol first introduced on the M68HC12 family of microcontrollers. This protocol assumes the host knows the communication clock rate that is determined by the target BDC clock rate. All communication is initiated and controlled by the host that drives a high-to-low edge to signal the beginning of each bit time. Commands and data are sent most significant bit first (MSB first). For a detailed description of the communications protocol, refer to [Section 15.2.2, "Communication Details."](#)

If a host is attempting to communicate with a target MCU that has an unknown BDC clock rate, a SYNC command may be sent to the target MCU to request a timed sync response signal from which the host can determine the correct communication speed.

BKGD is a pseudo-open-drain pin and there is an on-chip pullup so no external pullup resistor is required. Unlike typical open-drain pins, the external RC time constant on this pin, which is influenced by external capacitance, plays almost no role in signal rise time. The custom protocol provides for brief, actively driven speedup pulses to force rapid rise times on this pin without risking harmful drive level conflicts. Refer to [Section 15.2.2, "Communication Details,"](#) for more detail.

The conducted susceptibility is determined by injecting the transient susceptibility signal on each pin of the microcontroller. The transient waveform and injection methodology is based on IEC 61000-4-4 (EFT/B). The transient voltage required to cause performance degradation on any pin in the tested configuration is greater than or equal to the reported levels unless otherwise indicated by footnotes below the table.

Table A-18. Conducted Susceptibility

Parameter	Symbol	Conditions	f_{osc}/f_{BUS}	Result	Amplitude ¹ (Min)	Unit
Conducted susceptibility, electrical fast transient/burst (EFT/B)	V_{CS_EFT}	$V_{DD} = 5.5V$ $T_A = +25^{\circ}C$ package type 64 QFP	32768 Hz crystal 2 MHz Bus	A	± 0 $\pm 2.0^2$	kV
				B	± 2.5	
				C	± 3.0	
				D	$> \pm 3.0$	

¹ Data based on qualification test results. Not tested in production.

² The \overline{RESET} pin is susceptible to the minimum applied transient of 220 V. All other pins have a result of A up to a minimum of 2000V.

The susceptibility performance classification is described in [Table A-19](#).

Table A-19. Susceptibility Performance Classification

Result	Performance Criteria	
A	No failure	The MCU performs as designed during and after exposure.
B	Self-recovering failure	The MCU does not perform as designed during exposure. The MCU returns automatically to normal operation after exposure is removed.
C	Soft failure	The MCU does not perform as designed during exposure. The MCU does not return to normal operation until exposure is removed and the \overline{RESET} pin is asserted.
D	Hard failure	The MCU does not perform as designed during exposure. The MCU does not return to normal operation until exposure is removed and the power to the MCU is cycled.
E	Damage	The MCU does not perform as designed during and after exposure. The MCU cannot be returned to proper operation due to physical damage or other permanent performance degradation.