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

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
0x0066	**TPM2C0VH**	Bit 15	14	13	12	11	10	9	Bit 8
0x0067	**TPM2C0VL**	Bit 7	6	5	4	3	2	1	Bit 0
0x0068	**TPM2C1SC**	CH1F	CH1IE	MS1B	MS1A	ELS1B	ELS1A	0	0
0x0069	**TPM2C1VH**	Bit 15	14	13	12	11	10	9	Bit 8
0x006A	**TPM2C1VL**	Bit 7	6	5	4	3	2	1	Bit 0
0x006B	Reserved	Ñ	Ñ	Ñ	Ñ	Ñ	Ñ	Ñ	Ñ
0x006C	**RTCSC**	RTIF	RTCLKS		RTIE	RTCPS			
0x006D	**RTCCNT**	RTCCNT							
0x006E	**RTCMOD**	RTCMOD							
0x006F - 0x007F	Reserved	Ñ Ñ	Ñ Ñ	Ñ Ñ	Ñ Ñ	Ñ Ñ	Ñ Ñ	Ñ Ñ	Ñ Ñ

Table 7-2. Instruction Set Summary (Sheet 6 of 9)

Source Form	Operation	Address Mode	Object Code	Cycles	Cyc-by-Cyc Details	Affecton CCR	
						V 1 1 H	I N Z C

MOV *opr8a*,*opr8a*
MOV *opr8a*,X+
MOV *#opr8i*,*opr8a*
MOV ,X+,*opr8a*
	Move
	$(M)_{destination} \leftarrow (M)_{source}$
	In IX+/DIR and DIR/IX+ Modes, H:X ← (H:X) + $0001
		DIR/DIR
		DIR/IX+
		IMM/DIR
		IX+/DIR
			4E dd dd
			5E dd
			6E ii dd
			7E dd
				5
				5
				4
				5
					rpwpp
					rfwpp
					pwpp
					rfwpp
						0 1 1 Đ	Đ ↕ ↕ Đ

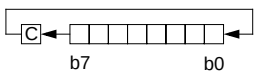
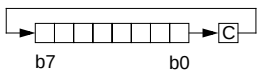
MUL	Unsigned multiply X:A ← (X) × (A)	INH	42	5	fffp	Đ 1 1 0	Đ Đ Đ 0
NEG *opr8a* NEGA NEGX NEG *oprx8*,X NEG ,X NEG *oprx8*,SP	Negate M ← Đ (M) = $00 Đ (M) (TwoÕs Complement) A ← Đ (A) = $00 Đ (A) X ← Đ (X) = $00 Đ (X) M ← Đ (M) = $00 Đ (M) M ← Đ (M) = $00 Đ (M) M ← Đ (M) = $00 Đ (M)	DIR INH INH IX1 IX SP1	30 dd 40 50 60 ff 70 9E 60 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	↕ 1 1 Đ	Đ ↕ ↕ ↕
NOP	No Operation Ñ Uses 1 Bus Cycle	INH	9D	1	p	Đ 1 1 Đ	Đ Đ Đ Đ
NSA	Nibble Swap Accumulator A ← (A[3:0]:A[7:4])	INH	62	1	p	Đ 1 1 Đ	Đ Đ Đ Đ
ORA *#opr8i* ORA *opr8a* ORA *opr16a* ORA *oprx16*,X ORA *oprx8*,X ORA ,X ORA *oprx16*,SP ORA *oprx8*,SP	Inclusive OR Accumulator and Memory A ← (A) | (M)	IMM DIR EXT IX2 IX1 IX SP2 SP1	AA ii BA dd CA hh ll DA ee ff EA ff FA 9E DA ee ff 9E EA ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	0 1 1 Đ	Đ ↕ ↕ Đ
PSHA	Push Accumulator onto Stack Push (A); SP ← (SP) Đ $0001	INH	87	2	sp	Đ 1 1 Đ	Đ Đ Đ Đ
PSHH	Push H (Index Register High) onto Stack Push (H); SP ← (SP) Đ $0001	INH	8B	2	sp	Đ 1 1 Đ	Đ Đ Đ Đ
PSHX	Push X (Index Register Low) onto Stack Push (X); SP ← (SP) Đ $0001	INH	89	2	sp	Đ 1 1 Đ	Đ Đ Đ Đ
PULA	Pull Accumulator from Stack SP ← (SP + $0001); Pull (A)	INH	86	3	ufp	Đ 1 1 Đ	Đ Đ Đ Đ
PULH	Pull H (Index Register High) from Stack SP ← (SP + $0001); Pull (H)	INH	8A	3	ufp	Đ 1 1 Đ	Đ Đ Đ Đ
PULX	Pull X (Index Register Low) from Stack SP ← (SP + $0001); Pull (X)	INH	88	3	ufp	Đ 1 1 Đ	Đ Đ Đ Đ
ROL *opr8a* ROLA ROLX ROL *oprx8*,X ROL ,X ROL *oprx8*,SP	Rotate Left through Carry 	DIR INH INH IX1 IX SP1	39 dd 49 59 69 ff 79 9E 69 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	↕ 1 1 Đ	Đ↕ ↕ ↕
ROR *opr8a* RORA RORX ROR *oprx8*,X ROR ,X ROR *oprx8*,SP	Rotate Right through Carry 	DIR INH INH IX1 IX SP1	36 dd 46 56 66 ff 76 9E 66 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	↕ 1 1 Đ	Đ ↕ ↕ ↕

Table 7-3. Opcode Map (Sheet 2 of 2)

Bit-Manipulation		Branch	Read-Modify-Write					Control		Register/Memory					
						9E60 6 NEG 3 SP1							9ED0 5 SUB 4 SP2	9EE0 4 SUB 3 SP1	
						9E61 6 CBEQ 4 SP1							9ED1 5 CMP 4 SP2	9EE1 4 CMP 3 SP1	
													9ED2 5 SBC 4 SP2	9EE2 4 SBC 3 SP1	
						9E63 6 COM 3 SP1							9ED3 5 CPX 4 SP2	9EE3 4 CPX 3 SP1	9EF3 6 CPHX 3 SP1
						9E64 6 LSR 3 SP1							9ED4 5 AND 4 SP2	9EE4 4 AND 3 SP1	
													9ED5 5 BIT 4 SP2	9EE5 4 BIT 3 SP1	
						9E66 6 ROR 3 SP1							9ED6 5 LDA 4 SP2	9EE6 4 LDA 3 SP1	
						9E67 6 ASR 3 SP1							9ED7 5 STA 4 SP2	9EE7 4 STA 3 SP1	
						9E68 6 LSL 3 SP1							9ED8 5 EOR 4 SP2	9EE8 4 EOR 3 SP1	
						9E69 6 ROL 3 SP1							9ED9 5 ADC 4 SP2	9EE9 4 ADC 3 SP1	
						9E6A 6 DEC 3 SP1							9EDA 5 ORA 4 SP2	9EEA 4 ORA 3 SP1	
						9E6B 8 DBNZ 4 SP1							9EDB 5 ADD 4 SP2	9EEB 4 ADD 3 SP1	
						9E6C 6 INC 3 SP1									
						9E6D 5 TST 3 SP1									
										9EAE 5 LDHX 2 IX	9EBE 6 LDHX 4 IX2	9ECE 5 LDHX 3 IX1	9EDE 5 LDX 4 SP2	9EEE 4 LDX 3 SP1	9EFE 5 LDHX 3 SP1
						9E6F 6 CLR 3 SP1							9EDF 5 STX 4 SP2	9EEF 4 STX 3 SP1	9EFF 5 STHX 3 SP1

INH	Inherent	REL	Relative	SP1	Stack Pointer, 8-Bit Offset
IMM	Immediate	IX	Indexed, No Offset	SP2	Stack Pointer, 16-Bit Offset
DIR	Direct	IX1	Indexed, 8-Bit Offset	IX+	Indexed, No Offset with Post Increment
EXT	Extended	IX2	Indexed, 16-Bit Offset		
DD	DIR to DIR	IMD	IMM to DIR	IX1+	Indexed, 1-Byte Offset with Post Increment
IX+D	IX+ to DIR	DIX+	DIR to IX+		

Note: All Sheet 2 Opcodes are Preceded by the Page 2 Prebyte (9E)

Prebyte (9E) and Opcode in Hexadecimal
Number of Bytes

9E60 6
NEG
3 SP1

HCS08 Cycles
Instruction Mnemonic
Addressing Mode

Table 10-3. IICF Field Descriptions

Field	Description
7Ð6 MULT	IIC Multiplier Factor . The MULT bits deÞne the multiplier factor, mul. This factor, along with the SCL divider, generates the IIC baud rate. The multiplier factor mul as deÞned by the MULT bits is provided below. 00 mul = 01 01 mul = 02 10 mul = 04 11 Reserved
5Ð0 ICR	IIC Clock Rate . The ICR bits are used to prescale the bus clock for bit rate selection. These bits and the MULT bits determine the IIC baud rate, the SDA hold time, the SCL Start hold time, and the SCL Stop hold time. Table 10-5 provides the SCL divider and hold values for corresponding values of the ICR. The SCL divider multiplied by multiplier factor mul generates IIC baud rate. $$\text{IIC baud rate} = \frac{\text{bus speed (Hz)}}{\text{mul} \times \text{SCLdivider}}$$ ***Eqn. 10-1*** SDA hold time is the delay from the falling edge of SCL (IIC clock) to the changing of SDA (IIC data). $$\text{SDA hold time} = \text{bus period (s)} \times \text{mul} \times \text{SDA hold value}$$ Eqn. 10-2 SCL start hold time is the delay from the falling edge of SDA (IIC data) while SCL is high (Start condition) to the falling edge of SCL (IIC clock). $$\text{SCL Start hold time} = \text{bus period (s)} \times \text{mul} \times \text{SCL Start hold value}$$ Eqn. 10-3 SCL stop hold time is the delay from the rising edge of SCL (IIC clock) to the rising edge of SDA SDA (IIC data) while SCL is high (Stop condition). $$\text{SCL Stop hold time} = \text{bus period (s)} \times \text{mul} \times \text{SCL Stop hold value}$$ Eqn. 10-4

For example, if the bus speed is 8 MHz, the table below shows the possible hold time values with different ICR and MULT selections to achieve an IIC baud rate of 100kbps.

Table 10-4. Hold Time Values for 8 MHz Bus Speed

MULT	ICR	Hold Times (µ s)		
		SDA	SCL Start	SCL Stop
0x2	0x00	3.500	3.000	5.500
0x1	0x07	2.500	4.000	5.250
0x1	0x0B	2.250	4.000	5.250
0x0	0x14	2.125	4.250	5.125
0x0	0x18	1.125	4.750	5.125

- CLKS bits are written to 00
- IREFS bit is written to 1
- RDIV bits are written to divide trimmed reference clock to be within the range of 31.25 kHz to 39.0625 kHz.

In FLL engaged internal mode, the ICSOUT clock is derived from the FLL clock, which is controlled by the internal reference clock. The FLL loop will lock the frequency to 1024 times the reference frequency, as selected by the RDIV bits. The ICSLCLK is available for BDC communications, and the internal reference clock is enabled.

11.4.1.2 FLL Engaged External (FEE)

The FLL engaged external (FEE) mode is entered when all the following conditions occur:

- CLKS bits are written to 00
- IREFS bit is written to 0
- RDIV bits are written to divide reference clock to be within the range of 31.25 kHz to 39.0625 kHz

In FLL engaged external mode, the ICSOUT clock is derived from the FLL clock which is controlled by the external reference clock.The FLL loop will lock the frequency to 1024 times the reference frequency, as selected by the RDIV bits. The ICSLCLK is available for BDC communications, and the external reference clock is enabled.

11.4.1.3 FLL Bypassed Internal (FBI)

The FLL bypassed internal (FBI) mode is entered when all the following conditions occur:

- CLKS bits are written to 01
- IREFS bit is written to 1.
- BDM mode is active or LP bit is written to 0

In FLL bypassed internal mode, the ICSOUT clock is derived from the internal reference clock. The FLL clock is controlled by the internal reference clock, and the FLL loop will lock the FLL frequency to 1024 times the reference frequency, as selected by the RDIV bits. The ICSLCLK will be available for BDC communications, and the internal reference clock is enabled.

11.4.1.4 FLL Bypassed Internal Low Power (FBILP)

The FLL bypassed internal low power (FBILP) mode is entered when all the following conditions occur:

- CLKS bits are written to 01
- IREFS bit is written to 1.
- BDM mode is not active and LP bit is written to 1

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

If EREFSTEN is set and the ERCLKEN bit is written to 1, the external reference clock will keep running during stop mode in order to provide a fast recovery upon exiting stop.

11.4.7 Fixed Frequency Clock

The ICS presents the divided FLL reference clock as ICSFFCLK for use as an additional clock source for peripheral modules. The ICS provides an output signal (ICSFFE) which indicates when the ICS is providing ICSOUT frequencies four times or greater than the divided FLL reference clock (ICSFFCLK). In FLL Engaged mode (FEI and FEE) this is always true and ICSFFE is always high. In ICS Bypass modes, ICSFFE will get asserted for the following combinations of BDIV and RDIV values:

- BDIV=00 (divide by 1), RDIV≥ 010
- BDIV=01 (divide by 2), RDIV≥ 011
- BDIV=10 (divide by 4), RDIV≥ 100
- BDIV=11 (divide by 8), RDIV≥ 101

ßag is set. If RDRF was already set indicating the receive data register (buffer) was already full, the overrun (OR) status ßag is set and the new data is lost. Because the SCI receiver is double-buffered, the program has one full character time after RDRF is set before the data in the receive data buffer must be read to avoid a receiver overrun.

When a program detects that the receive data register is full (RDRF = 1), it gets the data from the receive data register by reading SCID. The RDRF ßag is cleared automatically by a 2-step sequence which is normally satisÞed in the course of the userÕs program that handles receive data. Refer to Section 14.3.4, ÒInterrupts and Status FlagsÓ for more details about ßag clearing.

14.3.3.1 Data Sampling Technique

The SCI receiver uses a 16× baud rate clock for sampling. The receiver starts by taking logic level samples at 16 times the baud rate to search for a falling edge on the RxD serial data input pin. A falling edge is deÞned as a logic 0 sample after three consecutive logic 1 samples. The 16× baud rate clock is used to divide the bit time into 16 segments labeled RT1 through RT16. When a falling edge is located, three more samples are taken at RT3, RT5, and RT7 to make sure this was a real start bit and not merely noise. If at least two of these three samples are 0, the receiver assumes it is synchronized to a receive character.

The receiver then samples each bit time, including the start and stop bits, at RT8, RT9, and RT10 to determine the logic level for that bit. The logic level is interpreted to be that of the majority of the samples taken during the bit time. In the case of the start bit, the bit is assumed to be 0 if at least two of the samples at RT3, RT5, and RT7 are 0 even if one or all of the samples taken at RT8, RT9, and RT10 are 1s. If any sample in any bit time (including the start and stop bits) in a character frame fails to agree with the logic level for that bit, the noise ßag (NF) will be set when the received character is transferred to the receive data buffer.

The falling edge detection logic continuously looks for falling edges, and if an edge is detected, the sample clock is resynchronized to bit times. This improves the reliability of the receiver in the presence of noise or mismatched baud rates. It does not improve worst case analysis because some characters do not have any extra falling edges anywhere in the character frame.

In the case of a framing error, provided the received character was not a break character, the sampling logic that searches for a falling edge is Þlled with three logic 1 samples so that a new start bit can be detected almost immediately.

In the case of a framing error, the receiver is inhibited from receiving any new characters until the framing error ßag is cleared. The receive shift register continues to function, but a complete character cannot transfer to the receive data buffer if FE is still set.

14.3.3.2 Receiver Wakeup Operation

Receiver wakeup is a hardware mechanism that allows an SCI receiver to ignore the characters in a message that is intended for a different SCI receiver. In such a system, all receivers evaluate the Þrst character(s) of each message, and as soon as they determine the message is intended for a different receiver, they write logic 1 to the receiver wake up (RWU) control bit in SCIC2. When RWU bit is set, the status ßags associated with the receiver (with the exception of the idle bit, IDLE, when RWUID bit is set) are inhibited from setting, thus eliminating the software overhead for handling the unimportant message

15.3 Modes of Operation

15.3.1 SPI in Stop Modes

The SPI is disabled in all stop modes, regardless of the settings before executing the STOP instruction. During stop2 mode, the SPI module will be fully powered down. Upon wake-up from stop2 mode, the SPI module will be in the reset state. During stop3 mode, clocks to the SPI module are halted. No registers are affected. If stop3 is exited with a reset, the SPI will be put into its reset state. If stop3 is exited with an interrupt, the SPI continues from the state it was in when stop3 was entered.

15.4 Register DeÞnition

The SPI has Þve 8-bit registers to select SPI options, control baud rate, report SPI status, and for transmit/receive data.

Refer to the direct-page register summary in the Memory chapter of this data sheet for the absolute address assignments for all SPI registers. This section refers to registers and control bits only by their names, and a Freescale-provided equate or header Þle is used to translate these names into the appropriate absolute addresses.

15.4.1 SPI Control Register 1 (SPIC1)

This read/write register includes the SPI enable control, interrupt enables, and conÞguration options.

	7	6	5	4	3	2	1	0
R W	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
Reset	0	0	0	0	0	1	0	0

Figure 15-5. SPI Control Register 1 (SPIC1)

Table 15-1. SPIC1 Field Descriptions

Field	Description
7 SPIE	SPI Interrupt Enable (for SPRF and MODF) Ñ This is the interrupt enable for SPI receive buffer full (SPRF) and mode fault (MODF) events. 0 Interrupts from SPRF and MODF inhibited (use polling) 1 When SPRF or MODF is 1, request a hardware interrupt
6 SPE	SPI System Enable Ñ Disabling the SPI halts any transfer that is in progress, clears data buffers, and initializes internal state machines. SPRF is cleared and SPTEF is set to indicate the SPI transmit data buffer is empty. 0 SPI system inactive 1 SPI system enabled
5 SPTIE	SPI Transmit Interrupt Enable Ñ This is the interrupt enable bit for SPI transmit buffer empty (SPTEF). 0 Interrupts from SPTEF inhibited (use polling) 1 When SPTEF is 1, hardware interrupt requested

When the TPM is conÞgured for center-aligned PWM (and ELSnB:ELSnA not = 0:0), the data direction for all channels in this TPM are overridden, the TPMxCHn pins are forced to be outputs controlled by the TPM, and the ELSnA bits control the polarity of each TPMxCHn output. If ELSnB:ELSnA=1:0, the corresponding TPMxCHn pin is cleared when the timer counter is counting up, and the channel value register matches the timer counter; the TPMxCHn pin is set when the timer counter is counting down, and the channel value register matches the timer counter. If ELSnA=1, the corresponding TPMxCHn pin is set when the timer counter is counting up and the channel value register matches the timer counter; the TPMxCHn pin is cleared when the timer counter is counting down and the channel value register matches the timer counter.

TPMxMODH:TPMxMODL = 0x0008
TPMxCnVH:TPMxCnVL = 0x0005

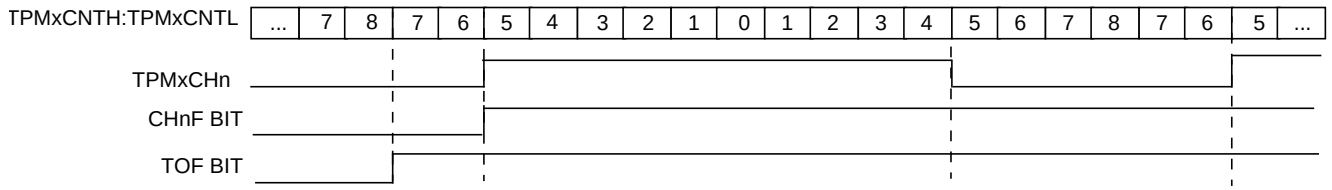


Figure 16-5. High-True Pulse of a Center-Aligned PWM

TPMxMODH:TPMxMODL = 0x0008
TPMxCnVH:TPMxCnVL = 0x0005

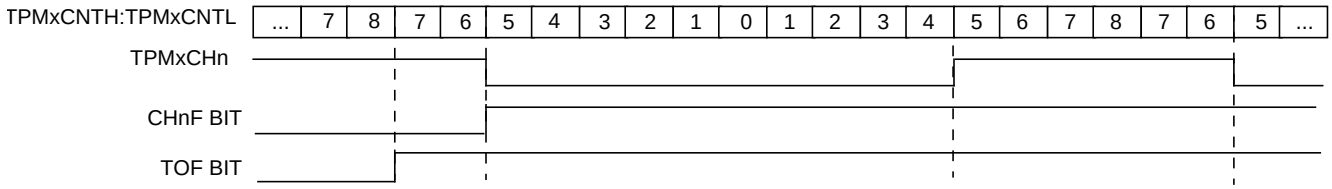


Figure 16-6. Low-True Pulse of a Center-Aligned PWM

Figure 17-4 shows the host receiving a logic 0 from the target HCS08 MCU. Because the host is asynchronous to the target MCU, there is a 0-to-1 cycle delay from the host-generated falling edge on BKGD to the start of the bit time as perceived by the target MCU. The host initiates the bit time but the target HCS08 Þnishes it. Because the target wants the host to receive a logic 0, it drives the BKGD pin low for 13 BDC clock cycles, then brießy drives it high to speed up the rising edge. The host samples the bit level about 10 cycles after starting the bit time.

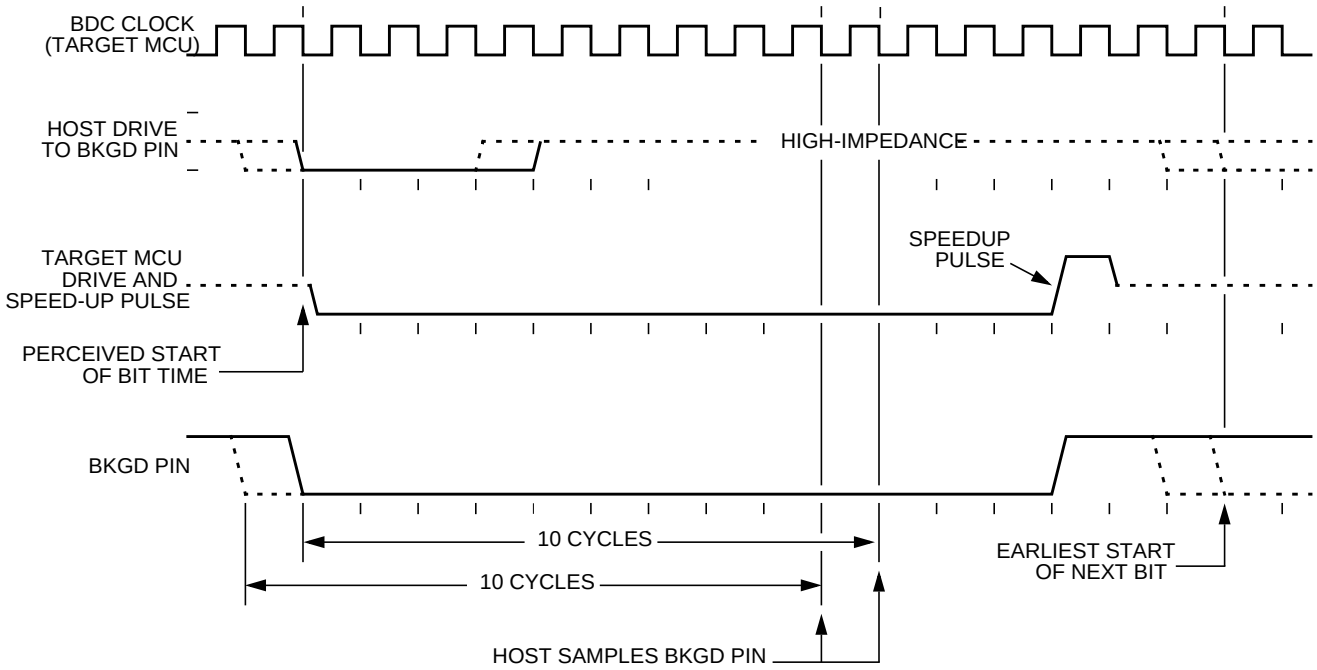


Figure 17-4. BDM Target-to-Host Serial Bit Timing (Logic 0)

The SYNC command is unlike other BDC commands because the host does not necessarily know the correct communications speed to use for BDC communications until after it has analyzed the response to the SYNC command.

To issue a SYNC command, the host:

- Drives the BKGD pin low for at least 128 cycles of the slowest possible BDC clock (The slowest clock is normally the reference oscillator/64 or the self-clocked rate/64.)
- Drives BKGD high for a brief speedup pulse to get a fast rise time (This speedup pulse is typically one cycle of the fastest clock in the system.)
- Removes all drive to the BKGD pin so it reverts to high impedance
- Monitors the BKGD pin for the sync response pulse

The target, upon detecting the SYNC request from the host (which is a much longer low time than would ever occur during normal BDC communications):

- Waits for BKGD to return to a logic high
- Delays 16 cycles to allow the host to stop driving the high speedup pulse
- Drives BKGD low for 128 BDC clock cycles
- Drives a 1-cycle high speedup pulse to force a fast rise time on BKGD
- Removes all drive to the BKGD pin so it reverts to high impedance

The host measures the low time of this 128-cycle sync response pulse and determines the correct speed for subsequent BDC communications. Typically, the host can determine the correct communication speed within a few percent of the actual target speed and the communication protocol can easily tolerate speed errors of several percent.

17.2.4 BDC Hardware Breakpoint

The BDC includes one relatively simple hardware breakpoint that compares the CPU address bus to a 16-bit match value in the BDCBKPT register. This breakpoint can generate a forced breakpoint or a tagged breakpoint. A forced breakpoint causes the CPU to enter active background mode at the Þrst instruction boundary following any access to the breakpoint address. The tagged breakpoint causes the instruction opcode at the breakpoint address to be tagged so that the CPU will enter active background mode rather than executing that instruction if and when it reaches the end of the instruction queue. This implies that tagged breakpoints can only be placed at the address of an instruction opcode while forced breakpoints can be set at any address.

The breakpoint enable (BKPTEN) control bit in the BDC status and control register (BDCSCR) is used to enable the breakpoint logic (BKPTEN = 1). When BKPTEN = 0, its default value after reset, the breakpoint logic is disabled and no BDC breakpoints are requested regardless of the values in other BDC breakpoint registers and control bits. The force/tag select (FTS) control bit in BDCSCR is used to select forced (FTS = 1) or tagged (FTS = 0) type breakpoints.

The on-chip debug module (DBG) includes circuitry for two additional hardware breakpoints that are more ßexible than the simple breakpoint in the BDC module.

Appendix A
Electrical Characteristics

A.1 Introduction

This section contains electrical and timing speciÞcations for the MC9S08SG32 Series of microcontrollers available at the time of publication. The MC9S08SG32 Series includes both:

- Standard (STD)Ñ devices that are *standard-temperature* rated. Table rows marked with a ♦ indicate electrical characteristics that apply to these devices.
- AEC Grade 0 Ñ devices that are *high-temperature* rated. Table rows marked with a ♦ indicate electrical characteristics that apply to AEC Grade 0 devices.

A.2 Parameter ClassiÞcation

The electrical parameters shown in this supplement are guaranteed by various methods. To give the customer a better understanding, the following classiÞcation is used and the parameters are tagged accordingly in the tables where appropriate:

Table A-1. Parameter ClassiÞcations

P	Those parameters are guaranteed during production testing on each individual device.
C	Those parameters are achieved through the design characterization by measuring a statistically relevant sample size across process variations.
T	Those parameters are achieved by design characterization on a small sample size from typical devices under typical conditions unless otherwise noted. All values shown in the typical column are within this category.
D	Those parameters are derived mainly from simulations.

NOTE

The classiÞcation is shown in the column labeled ÒCÓ in the parameter tables where appropriate.

A.3 Absolute Maximum Ratings

Absolute maximum ratings are stress ratings only, and functional operation at the maxima is not guaranteed. Stress beyond the limits speciÞed in Table A-2 may affect device reliability or cause permanent damage to the device. For functional operating conditions, refer to the remaining tables in this section.

A.7 Supply Current Characteristics

This section includes information about power supply current in various operating modes.

Table A-7. Supply Current Characteristics

#	C	Parameter	Symbol	V_{DD} (V)	Typ[1]	Max[2]	Unit	Temp Rated	
								Standard	AEC Grade 0
1	C	Run supply current[3] measured at (CPU clock = 4 MHz, f_{Bus} = 2 MHz)	RI_{DD}	5	1.4	3	mA	♦	
	C			3	1.3	2.5	mA	♦	
2	P	Run supply current[3] measured at (CPU clock = 16 MHz, f_{Bus} = 8 MHz)	RI_{DD}	5	4.7	7.5	mA	♦	
	C			3	4.6	7	mA	♦	
3	C	Run supply current[4] measured at (CPU clock = 32 MHz, f_{Bus} = 16MHz)	RI_{DD}	5	8.9	10	mA	♦	
	C			3	8.7	9.6	mA	♦	
4		Stop3 mode supply current							
	C	Ð40°C (C,V, and M sufÞx)	$S3I_{DD}$	5	0.96	Ð	µA	♦	
	P	25°C (All parts)			1.3	Ð	µA	♦	
	P[5]	85°C (C sufÞx only)			16.9	35	µA	♦	
	P[5]	105°C (V sufÞx only)			37	90	µA	♦	
	P[5]	125°C (M sufÞx only)			84	150	µA	♦	
	C	Ð40°C (C,V, and M sufÞx)		3	0.85	Ð	µA	♦	
	P	25°C (All parts)			1.2	Ð	µA	♦	
	P[5]	85°C (C sufÞx only)			14.8	30	µA	♦	
	P[5]	105°C (V sufÞx only)			32.7	80	µA	♦	
	P[5]	125°C (M sufÞx only)			75	130	µA	♦	

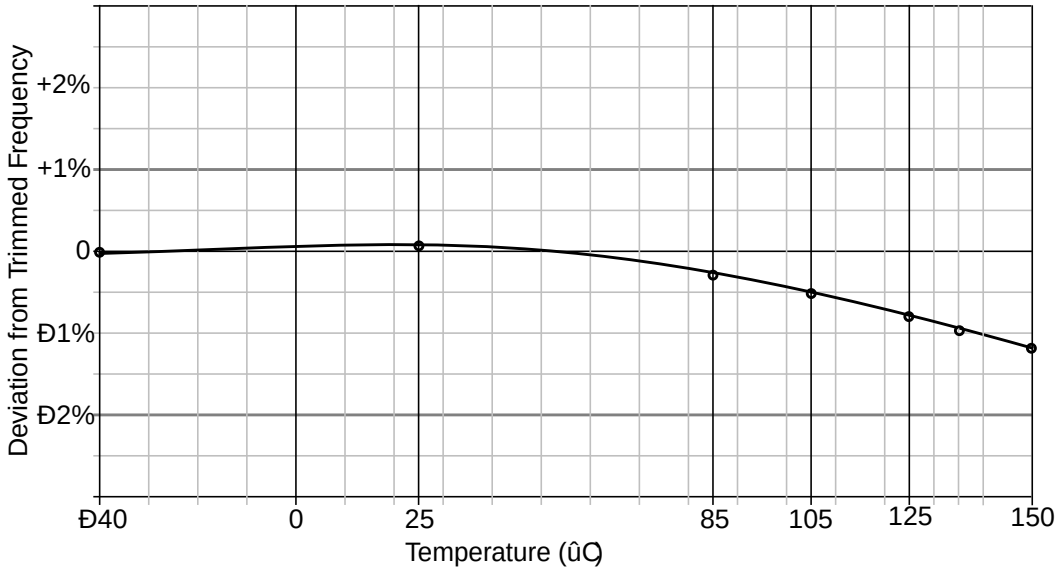


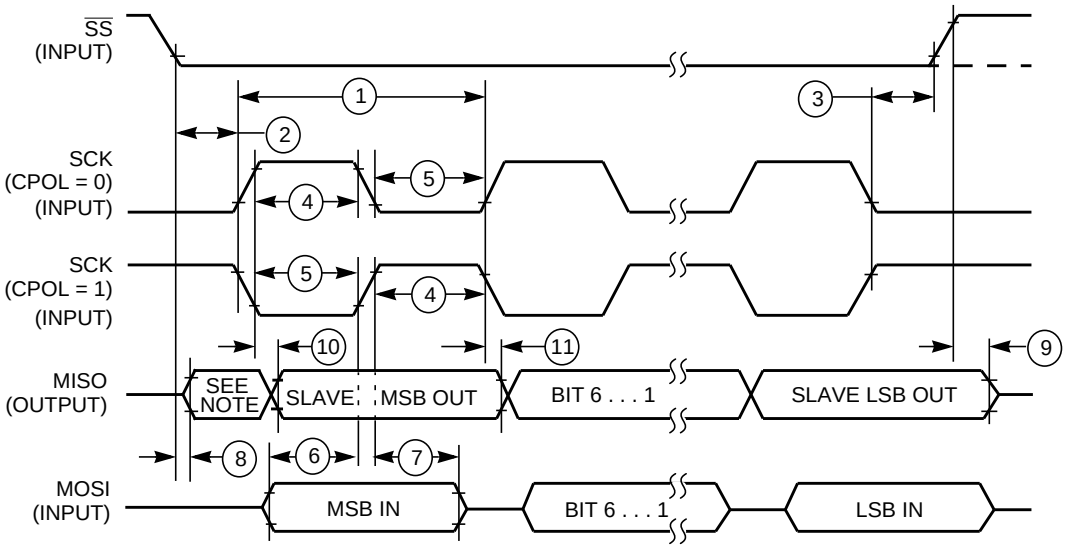
Figure A-8. Typical Frequency Deviation vs Temperature (ICS Trimmed to 16MHz bus@25ûC, 5V, FEI) [1]

A.10 Analog Comparator (ACMP) Electricals

Table A-10. Analog Comparator Electrical SpeciÞcations

#	C	Rating	Symbol	Min	Typical	Max	Unit	Temp Rated	
								Standard	AEC Grade 0
1	Ñ	Supply voltage	V_{DD}	2.7	Ñ	5.5	V	♦	
2	C/T	Supply current (active)	I_{DDAC}	Ñ	20	35	µA	♦	
3	D	Analog input voltage	V_{AIN}	V_{SS} Ð 0.3	Ñ	V_{DD}	V	♦	
4	D	Analog input offset voltage	V_{AIO}	Ñ	20	40	mV	♦	
5	D	Analog Comparator hysteresis	V_H	3.0	6.0	20.0	mV	♦	
6	D	Analog input leakage current	I_{ALKG}	Ñ	Ñ	1.0	µA	♦	
7	D	Analog Comparator initialization delay	t_{AINIT}	Ñ	Ñ	1.0	µs	♦	

1. Based on the average of several hundred units from a typical characterization lot.



NOTE:

1. Not defined but normally LSB of character just received

Figure A-17. SPI Slave Timing (CPHA = 1)