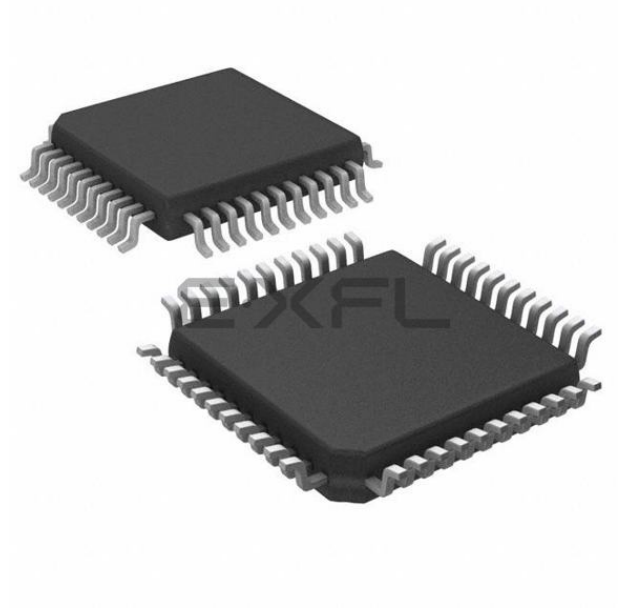


Welcome to E-XFL.COM

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	Obsolete
Core Processor	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	I²C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-QFP
Supplier Device Package	44-QFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08gt60cfber

I/O Pins

- All I/O pin states remain unchanged when the MCU enters stop3 mode.
- If the MCU is configured to go into stop2 mode, all I/O pins states are latched before entering stop.
- If the MCU is configured to go into stop1 mode, all I/O pins are forced to their default reset state upon entry into stop.

Memory

- All RAM and register contents are preserved while the MCU is in stop3 mode.
- All registers will be reset upon wake-up from stop2, but the contents of RAM are preserved and pin states remain latched until the PPDACK bit is written. The user may save any memory-mapped register data into RAM before entering stop2 and restore the data upon exit from stop2.
- All registers will be reset upon wake-up from stop1 and the contents of RAM are not preserved. The MCU must be initialized as upon reset. The contents of the FLASH memory are nonvolatile and are preserved in any of the stop modes.

ICG — In stop3 mode, the ICG enters its low-power standby state. Either the oscillator or the internal reference may be kept running when the ICG is in standby by setting the appropriate control bit. In both stop2 and stop1 modes, the ICG is turned off. Neither the oscillator nor the internal reference can be kept running in stop2 or stop1, even if enabled within the ICG module.

TPM — When the MCU enters stop mode, the clock to the TPM1 and TPM2 modules stop. The modules halt operation. If the MCU is configured to go into stop2 or stop1 mode, the TPM modules will be reset upon wake-up from stop and must be reinitialized.

ATD — When the MCU enters stop mode, the ATD will enter a low-power standby state**.** No conversion operation will occur while in stop. If the MCU is configured to go into stop2 or stop1 mode, the ATD will be reset upon wake-up from stop and must be reinitialized.

KBI — During stop3, the KBI pins that are enabled continue to function as interrupt sources that are capable of waking the MCU from stop3. The KBI is disabled in stop1 and stop2 and must be reinitialized after waking up from either of these modes.

SCI — When the MCU enters stop mode, the clocks to the SCI1 and SCI2 modules stop. The modules halt operation. If the MCU is configured to go into stop2 or stop1 mode, the SCI modules will be reset upon wake-up from stop and must be reinitialized.

SPI — When the MCU enters stop mode, the clocks to the SPI module stop. The module halts operation. If the MCU is configured to go into stop2 or stop1 mode, the SPI module will be reset upon wake-up from stop and must be reinitialized.

IIC — When the MCU enters stop mode, the clocks to the IIC module stops. The module halts operation. If the MCU is configured to go into stop2 or stop1 mode, the IIC module will be reset upon wake-up from stop and must be reinitialized.

Voltage Regulator — The voltage regulator enters a low-power standby state when the MCU enters any of the stop modes unless the LVD is enabled in stop mode or BDM is enabled.

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

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
$004F	Reserved	0	0	0	0	0	0	0	0
$0050	ATD1C	ATDPU	DJM	RES8	SGN	PRS			
$0051	ATD1SC	CCF	ATDIE	ATDCO	ATDCH				
$0052	ATD1RH	Bit 7	6	5	4	3	2	1	Bit 0
$0053	ATD1RL	Bit 7	6	5	4	3	2	1	Bit 0
$0054	ATD1PE	ATDPE7	ATDPE6	ATDPE5	ATDPE4	ATDPE3	ATDPE2	ATDPE1	ATDPE0
$0055–$0057	Reserved	—	—	—	—	—	—	—	—
$0058	IIC1A	ADDR							0
$0059	IIC1F	MULT		ICR					
$005A	IIC1C	IICEN	IICIE	MST	TX	TXAK	RSTA	0	0
$005B	IIC1S	TCF	IAAS	BUSY	ARBL	0	SRW	IICIF	RXAK
$005C	IIC1D	DATA							
$005D–$005F	Reserved	—	—	—	—	—	—	—	—
$0060	TPM2SC	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
$0061	TPM2CNTH	Bit 15	14	13	12	11	10	9	Bit 8
$0062	TPM2CNTL	Bit 7	6	5	4	3	2	1	Bit 0
$0063	TPM2MODH	Bit 15	14	13	12	11	10	9	Bit 8
$0064	TPM2MODL	Bit 7	6	5	4	3	2	1	Bit 0
$0065	TPM2C0SC	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
$0066	TPM2C0VH	Bit 15	14	13	12	11	10	9	Bit 8
$0067	TPM2C0VL	Bit 7	6	5	4	3	2	1	Bit 0
$0068	TPM2C1SC	CH1F	CH1IE	MS1B	MS1A	ELS1B	ELS1A	0	0
$0069	TPM2C1VH	Bit 15	14	13	12	11	10	9	Bit 8
$006A	TPM2C1VL	Bit 7	6	5	4	3	2	1	Bit 0
$006B	TPM2C2SC	CH2F	CH2IE	MS2B	MS2A	ELS2B	ELS2A	0	0
$006C	TPM2C2VH	Bit 15	14	13	12	11	10	9	Bit 8
$006D	TPM2C2VL	Bit 7	6	5	4	3	2	1	Bit 0
$006E	TPM2C3SC	CH3F	CH3IE	MS3B	MS3A	ELS3B	ELS3A	0	0
$006F	TPM2C3VH	Bit 15	14	13	12	11	10	9	Bit 8
$0070	TPM2C3VL	Bit 7	6	5	4	3	2	1	Bit 0
$0071	TPM2C4SC	CH4F	CH4IE	MS4B	MS4A	ELS4B	ELS4A	0	0
$0072	TPM2C4VH	Bit 15	14	13	12	11	10	9	Bit 8
$0073	TPM2C4VL	Bit 7	6	5	4	3	2	1	Bit 0
$0074–$007F	Reserved	—	—	—	—	—	—	—	—

Chapter 5 Resets, Interrupts, and System Configuration

5.1 Introduction

This section discusses basic reset and interrupt mechanisms and the various sources of reset and interrupts in the MC9S08GB/GT. Some interrupt sources from peripheral modules are discussed in greater detail within other sections of this data manual. This section gathers basic information about all reset and interrupt sources in one place for easy reference. A few reset and interrupt sources, including the computer operating properly (COP) watchdog and real-time interrupt (RTI), are not part of on-chip peripheral systems with their own sections but are part of the system control logic.

5.2 Features

Reset and interrupt features include:

- Multiple sources of reset for flexible system configuration and reliable operation:
  - Power-on detection (POR)
  - Low voltage detection (LVD) with enable
  - External $\overline{\text{RESET}}$ pin with enable
  - COP watchdog with enable and two timeout choices
  - Illegal opcode
  - Serial command from a background debug host
- Reset status register (SRS) to indicate source of most recent reset
- Separate interrupt vectors for each module (reduces polling overhead) (see Table 5-1)

5.3 MCU Reset

Resetting the MCU provides a way to start processing from a known set of initial conditions. During reset, most control and status registers are forced to initial values and the program counter is loaded from the reset vector ($FFFE:$FFFF). On-chip peripheral modules are disabled and I/O pins are initially configured as general-purpose high-impedance inputs with pullup devices disabled. The I bit in the condition code register (CCR) is set to block maskable interrupts so the user program has a chance to initialize the stack pointer (SP) and system control settings. SP is forced to $00FF at reset.

The MC9S08GB/GT has seven sources for reset:

- Power-on reset (POR)
- Low-voltage detect (LVD)
- Computer operating properly (COP) timer
- Illegal opcode detect
- Background debug forced reset
- The reset pin ($\overline{\text{RESET}}$)
- Clock generator loss of lock and loss of clock reset

7.2.2 Detailed Signal Descriptions

This section describes each pin signal in detail.

7.2.2.1 EXTAL— External Reference Clock / Oscillator Input

If the first write to the ICG control register 1 selected FLL engaged external or FLL bypassed modes, this signal is the analog external/reference clock or the input of the oscillator circuit. If the first write to the ICG control register 1 selected FLL engaged internal or self-clocked modes, this signal has no effect on the ICG.

7.2.2.2 XTAL— Oscillator Output

If the first write to the ICG control register 1 selected FLL engaged external or FLL bypassed modes using a crystal/resonator reference, this signal is the analog output of the oscillator amplifier circuit. In all other cases, this signal has no effect on the ICG.

7.2.3 External Clock Connections

If an external clock is used, then the pins are connected as shown below.

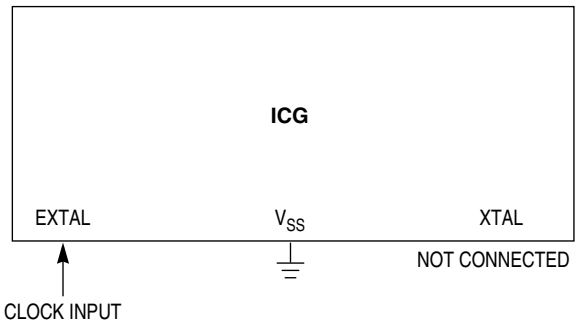


Figure 7-4. External Clock Connections

7.2.4 External Crystal/Resonator Connections

If an external crystal/resonator frequency reference is used, then the pins are connected as shown below. Recommended component values are listed in Appendix A, “Electrical Characteristics.”

ICGS1 then writing 1 to ICGIF (LOCRE = 0), or by a loss-of-clock induced reset (LOCRE = 1), or by any MCU reset.

If the ICG is in FEE, a loss of reference clock causes the ICG to enter SCM, and a loss of DCO clock causes the ICG to enter FBE mode. If the ICG is in FBE mode, a loss of reference clock will cause the ICG to enter SCM. In each case, the CLKST and CLKS bits will be automatically changed to reflect the new state.

A loss of clock will also cause a loss of lock when in FEE or FEI modes. Because the method of clearing the LOCS and LOLS bits is the same, this would only be an issue in the unlikely case that LOLRE = 1 and LOCRE = 0. In this case, the interrupt would be overridden by the reset for the loss of lock.

Table 7-2. Clock Monitoring

Mode	CLKS	REFST	ERCS	External Reference Clock Monitored?	DCO Clock Monitored?
Off	0X or 11	X	Forced Low	No	No
	10	0	Forced Low	No	No
	10	1	Real-Time(1)	Yes(1)	No
SCM (CLKST = 00)	0X	X	Forced Low	No	Yes(2)
	10	0	Forced High	No	Yes(2)
	10	1	Real-Time	Yes	Yes(2)
	11	X	Real-Time	Yes	Yes(2)
FEI (CLKST = 01)	0X	X	Forced Low	No	Yes
	11	X	Real-Time	Yes	Yes
FBE (CLKST = 10)	10	0	Forced High	No	No
	10	1	Real-Time	Yes	No
FEE (CLKST = 11)	11	X	Real-Time	Yes	Yes

1. If ENABLE is high (waiting for external crystal start-up after exiting stop).
2. DCO clock will not be monitored until DCOS = 1 upon entering SCM from off or FLL bypassed external mode.

7.3.8 Clock Mode Requirements

A clock mode is requested by writing to CLKS1:CLKS0 and the actual clock mode is indicated by CLKST1:CLKST0. Provided minimum conditions are met, the status shown in CLKST1:CLKST0 should be the same as the requested mode in CLKS1:CLKS0. Table 7-3 shows the relationship between CLKS, CLKST, and ICGOUT. It also shows the conditions for CLKS = CLKST or the reason CLKS ≠ CLKST.

NOTE

If a crystal will be used before the next reset, then be sure to set REFS = 1 and CLKS = 1x on the first write to the ICGC1 register. Failure to do so will result in “locking” REFS = 0 which will prevent the oscillator amplifier from being enabled until the next reset occurs.

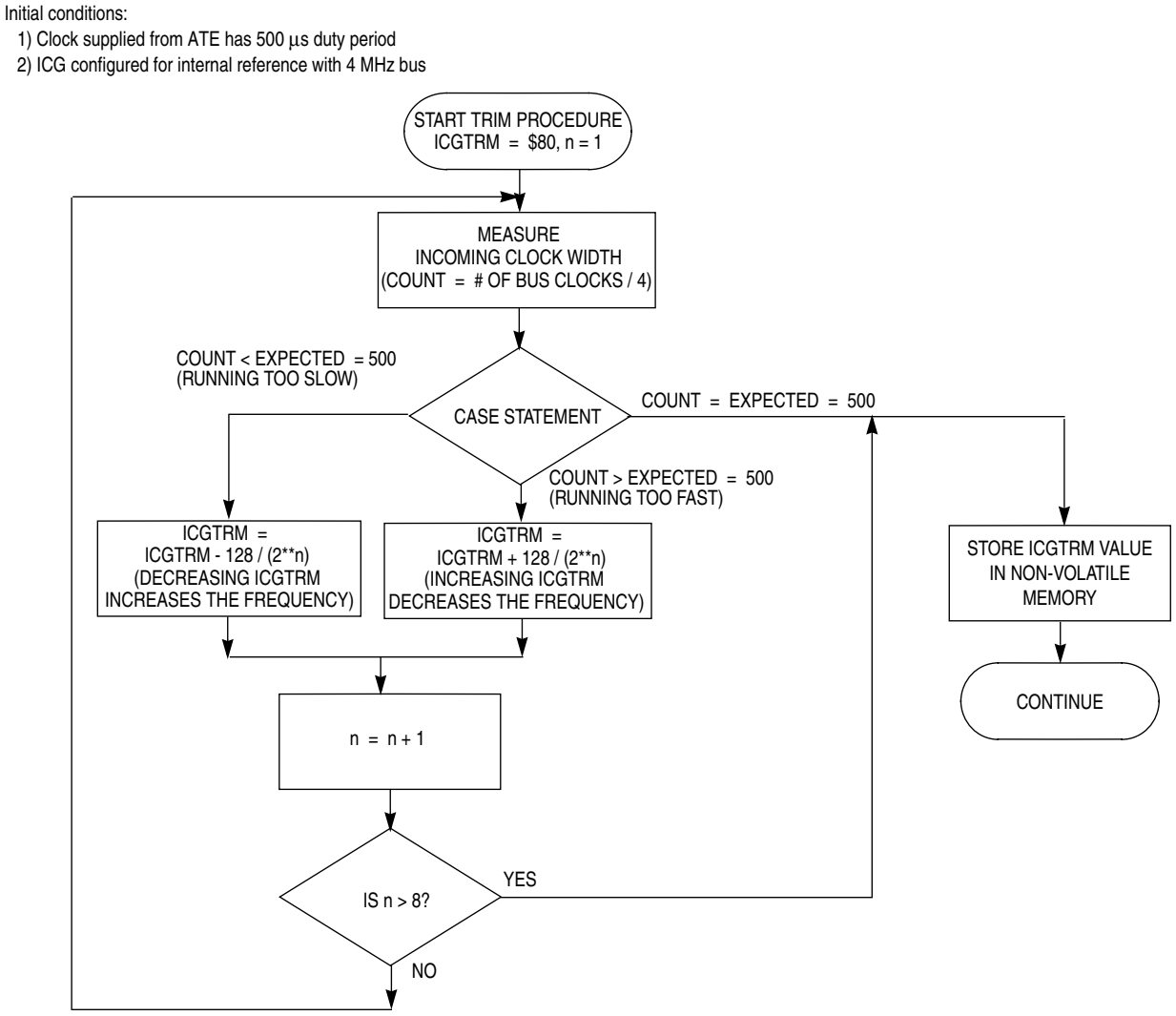


Figure 7-11. Trim Procedure

In this particular case, the MCU has been attached to a PCB and the entire assembly is undergoing final test with automated test equipment. A separate signal or message is provided to the MCU operating under user provided software control. The MCU initiates a trim procedure as outlined in Figure 7-11 while the tester supplies a precision reference signal.

If the intended bus frequency is near the maximum allowed for the device, it is recommended to trim using a reduction divisor (R) twice the final value. Once the trim procedure is complete, the reduction divisor can be restored. This will prevent accidental overshoot of the maximum clock frequency.

7.5 ICG Registers and Control Bits

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

ICGIF — 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 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 0 to ICGIF has no effect.

1 = An ICG interrupt request is pending.

0 = No ICG interrupt request is pending.

7.5.4 ICG Status Register 2 (ICGS2)

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	0	DCOS
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 7-16. ICG Status Register 2 (ICGS2)

DCOS — DCO Clock Stable

The DCOS bit is set when the DCO clock (ICG2DCLK) is stable, meaning the count error has not changed by more than n_{unlock} for two consecutive samples and the DCO clock is not static. This bit is used when exiting off state if CLKS = X1 to determine when to switch to the requested clock mode. It is also used in self-clocked mode to determine when to start monitoring the DCO clock. This bit is cleared upon entering the off state.

1 = DCO clock is stable.

0 = DCO clock is unstable.

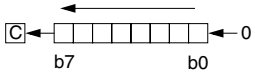
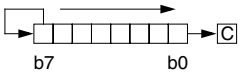
7.5.5 ICG Filter Registers (ICGFLTU, ICGFLTL)

	Bit 15	14	13	12	11	10	9	Bit 8
Read:	0	0	0	0	FLT			
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 7-17. ICG Upper Filter Register (ICGFLTU)

Table 8-1. HCS08 Instruction Set Summary (Sheet 2 of 7)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Bus Cycles[1]
			V	H	I	N	Z	C				
ASL *opr8a* ASLA ASLX ASL *oprx8*,X ASL ,X ASL *oprx8*,SP	Arithmetic Shift Left (Same as LSL)			–	–				DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff	5 1 1 5 4 6
ASR *opr8a* ASRA ASRX ASR *oprx8*,X ASR ,X ASR *oprx8*,SP	Arithmetic Shift Right			–	–				DIR INH INH IX1 IX SP1	37 47 57 67 77 9E67	dd ff ff	5 1 1 5 4 6
BCC *rel*	Branch if Carry Bit Clear	Branch if (C) = 0	–	–	–	–	–	–	REL	24	rr	3
BCLR *n,opr8a*	Clear Bit n in Memory	Mn ← 0	–	–	–	–	–	–	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	11 13 15 17 19 1B 1D 1F	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BCS *rel*	Branch if Carry Bit Set (Same as BLO)	Branch if (C) = 1	–	–	–	–	–	–	REL	25	rr	3
BEQ *rel*	Branch if Equal	Branch if (Z) = 1	–	–	–	–	–	–	REL	27	rr	3
BGE *rel*	Branch if Greater Than or Equal To (Signed Operands)	Branch if (N ⊕ V) = 0	–	–	–	–	–	–	REL	90	rr	3
BGND	Enter Active Background if ENBDM = 1	Waits For and Processes BDM Commands Until GO, TRACE1, or TAGGO	–	–	–	–	–	–	INH	82		5+
BGT *rel*	Branch if Greater Than (Signed Operands)	Branch if (Z) | (N ⊕ V) = 0	–	–	–	–	–	–	REL	92	rr	3
BHCC *rel*	Branch if Half Carry Bit Clear	Branch if (H) = 0	–	–	–	–	–	–	REL	28	rr	3
BHCS *rel*	Branch if Half Carry Bit Set	Branch if (H) = 1	–	–	–	–	–	–	REL	29	rr	3
BHI *rel*	Branch if Higher	Branch if (C) | (Z) = 0	–	–	–	–	–	–	REL	22	rr	3
BHS *rel*	Branch if Higher or Same (Same as BCC)	Branch if (C) = 0	–	–	–	–	–	–	REL	24	rr	3
BIH *rel*	Branch if IRQ Pin High	Branch if IRQ pin = 1	–	–	–	–	–	–	REL	2F	rr	3
BIL *rel*	Branch if IRQ Pin Low	Branch if IRQ pin = 0	–	–	–	–	–	–	REL	2E	rr	3
BIT #*opr8i* BIT *opr8a* BIT *opr16a* BIT *oprx16*,X BIT *oprx8*,X BIT ,X BIT *oprx16*,SP BIT *oprx8*,SP	Bit Test	(A) & (M) (CCR Updated but Operands Not Changed)	0	–	–			–	IMM DIR EXT IX2 IX1 IX SP2 SP1	A5 B5 C5 D5 E5 F5 9ED5 9EE5	ii dd hh ll ee ff ff ee ff ff	2 3 4 4 3 3 5 4
BLE *rel*	Branch if Less Than or Equal To (Signed Operands)	Branch if (Z) | (N ⊕ V) = 1	–	–	–	–	–	–	REL	93	rr	3
BLO *rel*	Branch if Lower (Same as BCS)	Branch if (C) = 1	–	–	–	–	–	–	REL	25	rr	3
BLS *rel*	Branch if Lower or Same	Branch if (C) | (Z) = 1	–	–	–	–	–	–	REL	23	rr	3
BLT *rel*	Branch if Less Than (Signed Operands)	Branch if (N ⊕ V) = 1	–	–	–	–	–	–	REL	91	rr	3
BMC *rel*	Branch if Interrupt Mask Clear	Branch if (I) = 0	–	–	–	–	–	–	REL	2C	rr	3

9.5.2 KBI Pin Enable Register (KBI1PE)

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	KBIPE7	KBIPE6	KBIPE5	KBIPE4	KBIPE3	KBIPE2	KBIPE1	KBIPE0
Reset:	0	0	0	0	0	0	0	0

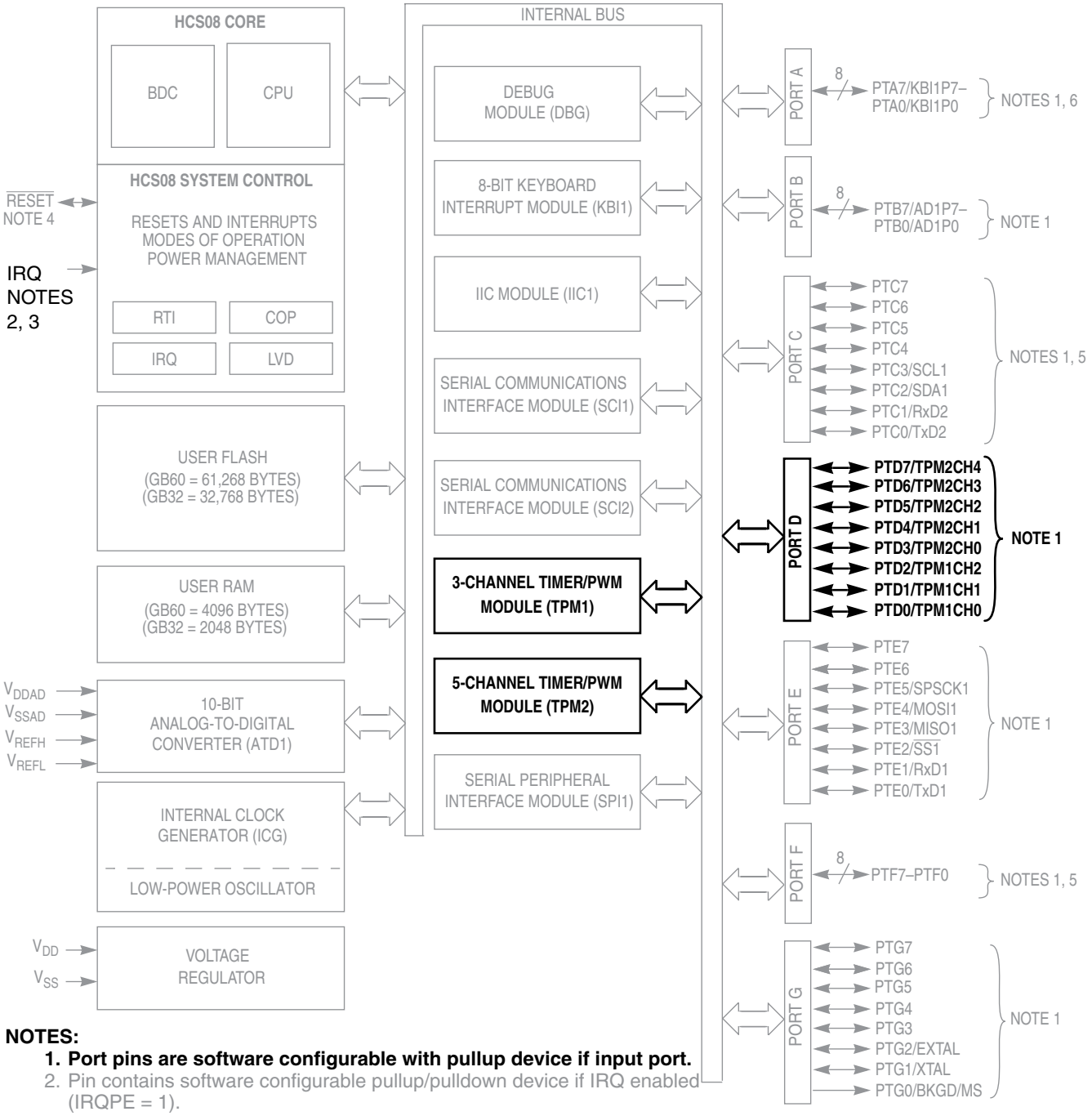
Figure 9-5. KBI Pin Enable Register (KBI1PE)

KBIPEn — Keyboard Pin Enable for KBI Port Bit n (n = 7–0)

Each of these read/write bits selects whether the associated KBI port pin is enabled as a keyboard interrupt input or functions as a general-purpose I/O pin.

1 = Bit n of KBI port enabled as a keyboard interrupt input

0 = Bit n of KBI port is a general-purpose I/O pin not associated with the KBI.



NOTES:

1. **Port pins are software configurable with pullup device if input port.**
2. Pin contains software configurable pullup/pulldown device if IRQ enabled (IRQPE = 1).
3. IRQ does not have a clamp diode to V_{DD}. IRQ should not be driven above V_{DD}.
4. Pin contains integrated pullup device.
5. High current drive
6. Pins PTA[7:4] contain both pullup and pulldown devices. Pulldown available when KBI enabled (KBIPn = 1).

Figure 10-1. Block Diagram Highlighting the TPM Modules

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 RxD1 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.

11.8 Additional SCI Functions

The following sections describe additional SCI functions.

11.8.1 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.

9-bit data mode typically is used in conjunction with parity to allow eight bits of data plus the parity in the ninth bit. Or it is used with address-mark wakeup so the ninth data bit can serve as the wakeup bit. In custom protocols, the ninth bit can also serve as a software-controlled marker.

11.9 Stop Mode Operation

During all stop modes, clocks to the SCI module are halted.

In stop1 and stop2 modes, all SCI register data is lost and must be re-initialized upon recovery from these two stop modes.

No SCI module registers are affected in stop3 mode.

RE — Receiver Enable

When the SCI receiver is off, the RxD1 pin reverts to being a general-purpose port I/O pin.
1 = Receiver on.
0 = Receiver off.

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 11.6.3, “Receiver Wakeup Operation,” for more details.
1 = SCI receiver in standby waiting for wakeup condition.
0 = Normal SCI receiver operation.

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 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 11.5.2, “Send Break and Queued Idle,” for more details.
1 = Queue break character(s) to be sent.
0 = Normal transmitter operation.

11.10.4 SCI x Status Register 1 (SCIxS1)

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

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
Write:								
Reset:	1	1	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 11-9. SCI x Status Register 1 (SCIxS1)

TDRE — 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 SCIxS1 with TDRE = 1 and then write to the SCI data register (SCIxD).
1 = Transmit data register (buffer) empty.
0 = Transmit data register (buffer) full.

12.1 Features

Features of the SPI module include:

- Master or slave mode operation
- Full-duplex or single-wire bidirectional option
- Programmable transmit bit rate
- Double-buffered transmit and receive
- Serial clock phase and polarity options
- Slave select output
- Selectable MSB-first or LSB-first shifting

12.2 Block Diagrams

This section includes block diagrams showing SPI system connections, the internal organization of the SPI module, and the SPI clock dividers that control the master mode bit rate.

12.2.1 SPI System Block Diagram

Figure 12-2 shows the SPI modules of two MCUs connected in a master-slave arrangement. The master device initiates all SPI data transfers. During a transfer, the master shifts data out (on the MOSI1 pin) to the slave while simultaneously shifting data in (on the MISO1 pin) from the slave. The transfer effectively exchanges the data that was in the SPI shift registers of the two SPI systems. The SPSCK1 signal is a clock output from the master and an input to the slave. The slave device must be selected by a low level on the slave select input ($\overline{SS1}$ pin). In this system, the master device has configured its $\overline{SS1}$ pin as an optional slave select output.

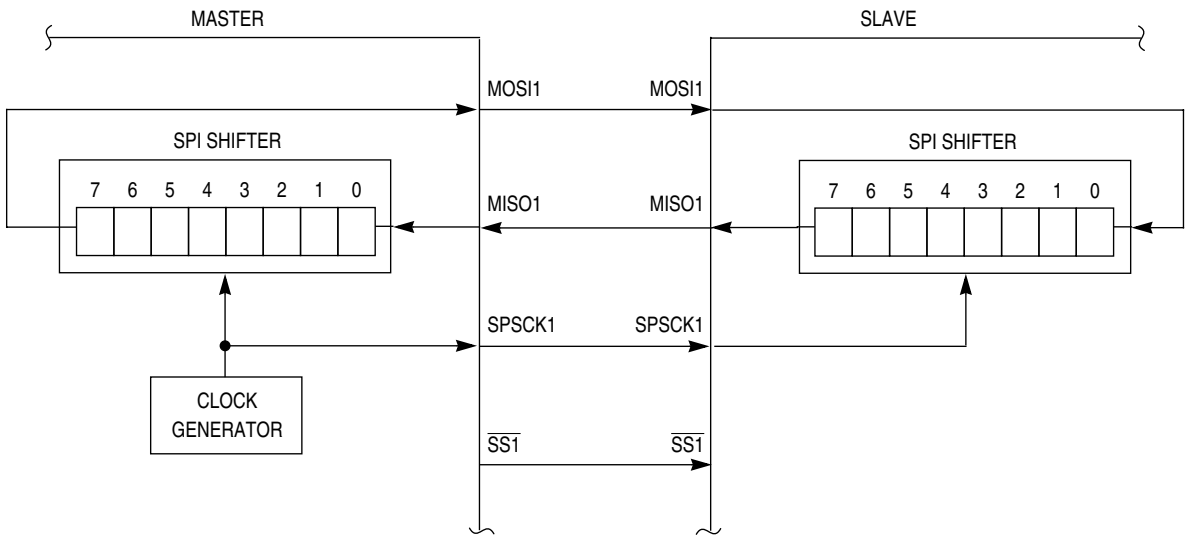
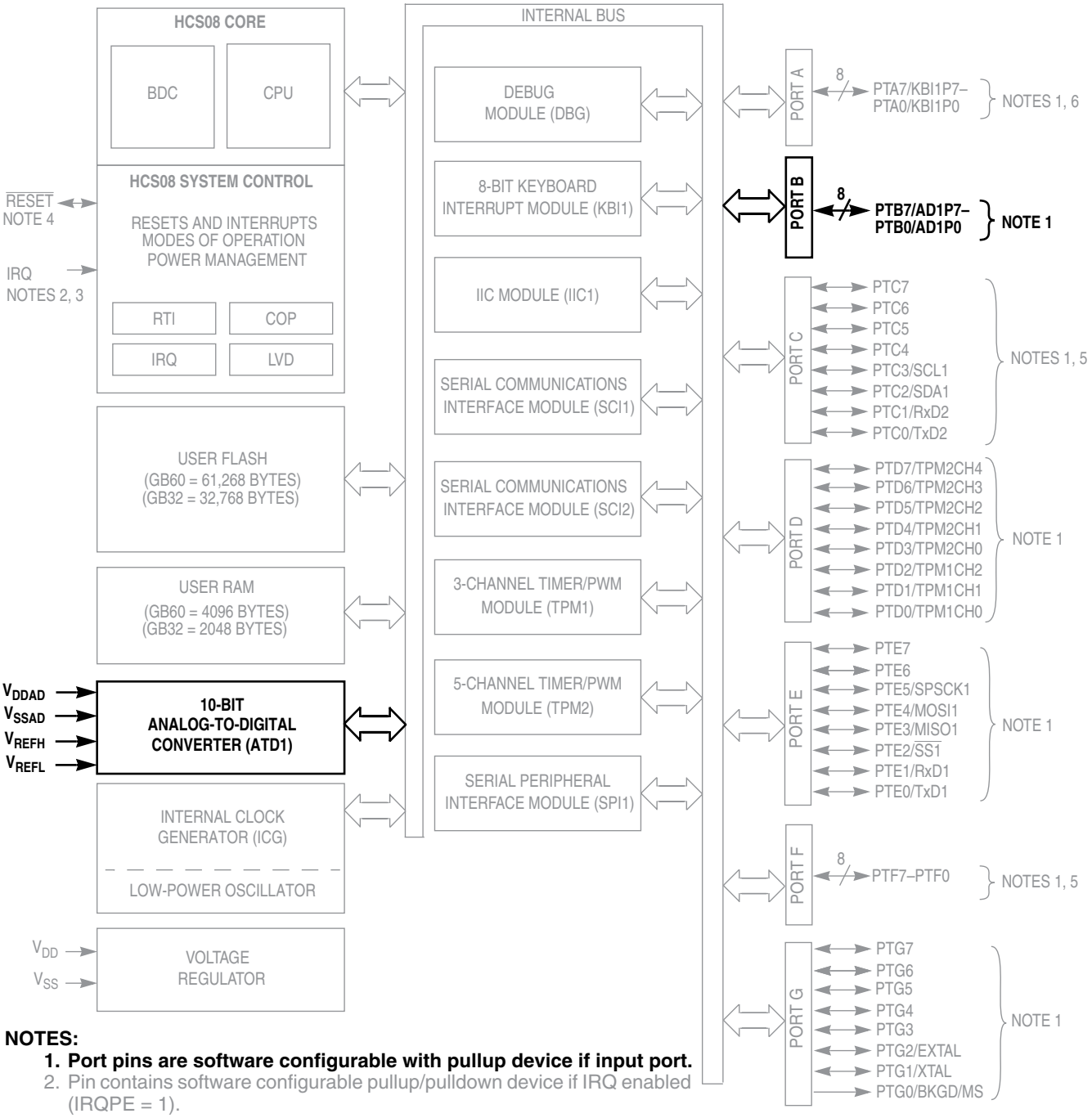


Figure 12-2. SPI System Connections



NOTES:

1. **Port pins are software configurable with pullup device if input port.**
2. Pin contains software configurable pullup/pulldown device if IRQ enabled (IRQPE = 1).
3. IRQ does not have a clamp diode to V_{DD}. IRQ should not be driven above V_{DD}.
4. Pin contains integrated pullup device.
5. High current drive
6. Pins PTA[7:4] contain both pullup and pulldown devices. Pulldown available when KBI enabled (KBIPn = 1).

Figure 14-1. MC9S08GBxx Block Diagram Highlighting ATD Block and Pins

14.3.3 Analog Input Multiplexer

The analog input multiplexer selects one of the eight external analog input channels to generate an analog sample. The analog input multiplexer includes negative stress protection circuitry which prevents cross-talk between channels when the applied input potentials are within specification. Only analog input signals within the potential range of V_{REFL} to V_{REFH} (ATD reference potentials) will result in valid ATD conversions.

14.3.4 ATD Module Accuracy Definitions

Figure 14-4 illustrates an ideal ATD transfer function. The horizontal axis represents the ATD input voltage in millivolts. The vertical axis the conversion result code. The ATD is specified with the following figures of merit:

- Number of bits (N) — The number of bits in the digitized output
- Resolution (LSB) — The resolution of the ATD is the step size of the ideal transfer function. This is also referred to as the ideal code width, or the difference between the transition voltages to a given code and to the next code. This unit, known as 1LSB, is equal to

$$1LSB = (V_{REFH} - V_{REFL}) / 2^N \qquad \textit{Eqn. 14-5}$$

- Inherent quantization error (E_Q) — This is the error caused by the division of the perfect ideal straight-line transfer function into the quantized ideal transfer function with 2^N steps. This error is ± 1/2 LSB.
- Differential non-linearity (DNL) — This is the difference between the current code width and the ideal code width (1LSB). The current code width is the difference in the transition voltages to the current code and to the next code. A negative DNL means the transfer function spends less time at the current code than ideal; a positive DNL, more. The DNL cannot be less than –1.0; a DNL of greater than 1.0 reduces the effective number of bits by 1.
- Integral non-linearity (INL) — This is the difference between the transition voltage to the current code and the transition to the corresponding code on the adjusted transfer curve. INL is a measure of how straight the line is (how far it deviates from a straight line). The adjusted ideal transition voltage is:

Eqn. 14-6

$$\text{Adjusted Ideal Trans. V} = \frac{(\text{Current Code} - 1/2)}{2^N} * ((V_{REFH} + E_{FS}) - (V_{REFL} + E_{ZS}))$$

- Zero scale error (E_{ZS}) — This is the difference between the transition voltage to the first valid code and the ideal transition to that code. Normally, it is defined as the difference between the actual and ideal transition to code $001, but in some cases the first transition may be to a higher code. The ideal transition to any code is:

Eqn. 14-7

$$\text{Ideal Transition V} = \frac{(\text{Current Code} - 1/2)}{2^N} *(V_{REFH} - V_{REFL})$$

Figure 15-4 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 finishes it. Because the target wants the host to receive a logic 0, it drives the BKGD pin low for 13 BDC clock cycles, then briefly drives it high to speed up the rising edge. The host samples the bit level about 10 cycles after starting the bit time.

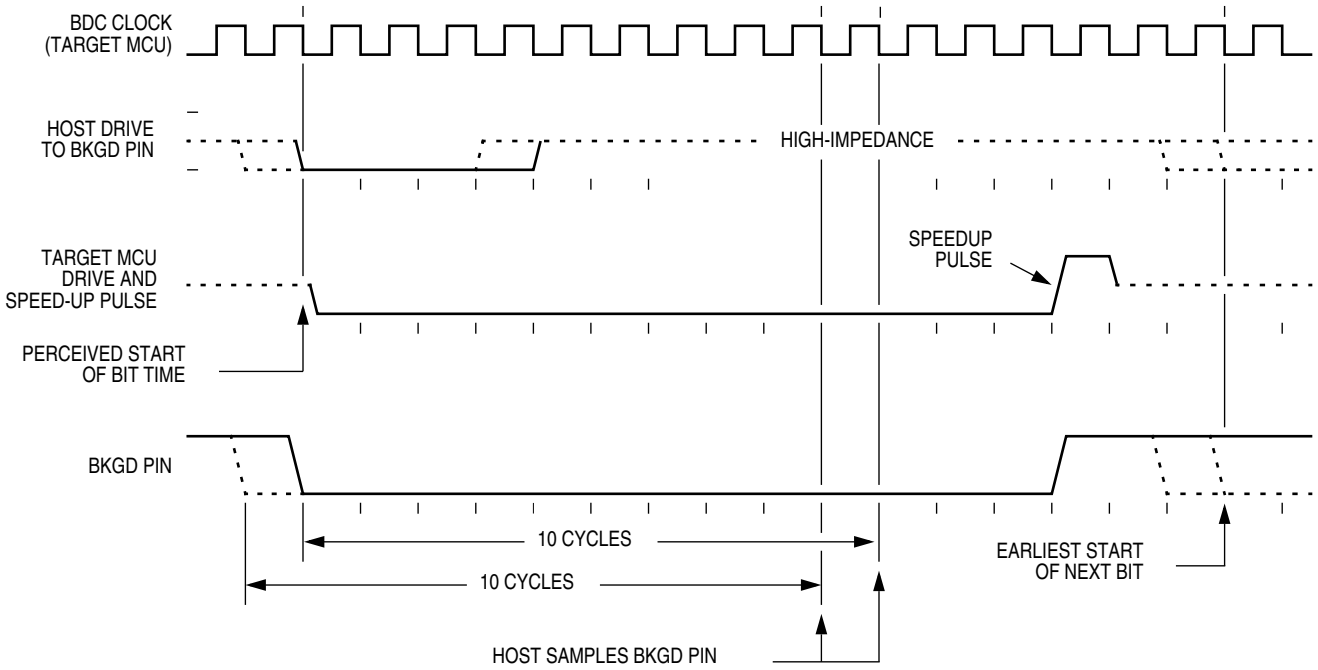


Figure 15-4. BDM Target-to-Host Serial Bit Timing (Logic 0)

15.3.3 BDC Commands

BDC commands are sent serially from a host computer to the BKGD pin of the target HCS08 MCU. All commands and data are sent MSB-first using a custom BDC communications protocol. Active background mode commands require that the target MCU is currently in the active background mode while non-intrusive commands may be issued at any time whether the target MCU is in active background mode or running a user application program.

Table 15-1 shows all HCS08 BDC commands, a shorthand description of their coding structure, and the meaning of each command.

Coding Structure Nomenclature

This nomenclature is used in Table 15-1 to describe the coding structure of the BDC commands.

		Commands begin with an 8-bit hexadecimal command code in the host-to-target direction (most significant bit first)
/	=	separates parts of the command
d	=	delay 16 target BDC clock cycles
AAAA	=	a 16-bit address in the host-to-target direction
RD	=	8 bits of read data in the target-to-host direction
WD	=	8 bits of write data in the host-to-target direction
RD16	=	16 bits of read data in the target-to-host direction
WD16	=	16 bits of write data in the host-to-target direction
SS	=	the contents of BDCSCR in the target-to-host direction (STATUS)
CC	=	8 bits of write data for BDCSCR in the host-to-target direction (CONTROL)
RBKP	=	16 bits of read data in the target-to-host direction (from BDCBKPT breakpoint register)
WBKP	=	16 bits of write data in the host-to-target direction (for BDCBKPT breakpoint register)

15.4 On-Chip Debug System (DBG)

Because HCS08 devices do not have external address and data buses, the most important functions of an in-circuit emulator have been built onto the chip with the MCU. The debug system consists of an 8-stage FIFO that can store address or data bus information, and a flexible trigger system to decide when to capture bus information and what information to capture. The system relies on the single-wire background debug system to access debug control registers and to read results out of the eight stage FIFO.

The debug module includes control and status registers that are accessible in the user's memory map. These registers are located in the high register space to avoid using valuable direct page memory space.

Most of the debug module's functions are used during development, and user programs rarely access any of the control and status registers for the debug module. The one exception is that the debug system can provide the means to implement a form of ROM patching. This topic is discussed in greater detail in Section 15.4.6, "Hardware Breakpoints."

15.4.1 Comparators A and B

Two 16-bit comparators (A and B) can optionally be qualified with the R/W signal and an opcode tracking circuit. Separate control bits allow you to ignore R/W for each comparator. The opcode tracking circuitry optionally allows you to specify that a trigger will occur only if the opcode at the specified address is actually executed as opposed to only being read from memory into the instruction queue. The comparators are also capable of magnitude comparisons to support the inside range and outside range trigger modes. Comparators are disabled temporarily during all BDC accesses.

The A comparator is always associated with the 16-bit CPU address. The B comparator compares to the CPU address or the 8-bit CPU data bus, depending on the trigger mode selected. Because the CPU data bus is separated into a read data bus and a write data bus, the RWAEN and RWA control bits have an additional purpose, in full address plus data comparisons they are used to decide which of these buses to use in the comparator B data bus comparisons. If RWAEN = 1 (enabled) and RWA = 0 (write), the CPU's write data bus is used. Otherwise, the CPU's read data bus is used.

The currently selected trigger mode determines what the debugger logic does when a comparator detects a qualified match condition. A match can cause:

- Generation of a breakpoint to the CPU
- Storage of data bus values into the FIFO
- Starting to store change-of-flow addresses into the FIFO (begin type trace)
- Stopping the storage of change-of-flow addresses into the FIFO (end type trace)

15.4.2 Bus Capture Information and FIFO Operation

The usual way to use the FIFO is to setup the trigger mode and other control options, then arm the debugger. When the FIFO has filled or the debugger has stopped storing data into the FIFO, you would read the information out of it in the order it was stored into the FIFO. Status bits indicate the number of words of valid information that are in the FIFO as data is stored into it. If a trace run is manually halted by writing 0 to ARM before the FIFO is full (CNT = 1:0:0:0), the information is shifted by one position and

CLKSW — Select Source for BDC Communications Clock

CLKSW defaults to 0, which selects the alternate BDC clock source.
1 = MCU bus clock.
0 = Alternate BDC clock source.

WS — Wait or Stop Status

When the target CPU is in wait or stop mode, most BDC commands cannot function. However, the BACKGROUND command can be used to force the target CPU out of wait or stop and into active background mode where all BDC commands work. Whenever the host forces the target MCU into active background mode, the host should issue a READ_STATUS command to check that BDMACT = 1 before attempting other BDC commands.
1 = Target CPU is in wait or stop mode, or a BACKGROUND command was used to change from wait or stop to active background mode.
0 = Target CPU is running user application code or in active background mode (was not in wait or stop mode when background became active).

WSF — Wait or Stop Failure Status

This status bit is set if a memory access command failed due to the target CPU executing a wait or stop instruction at or about the same time. The usual recovery strategy is to issue a BACKGROUND command to get out of wait or stop mode into active background mode, repeat the command that failed, then return to the user program. (Typically, the host would restore CPU registers and stack values and re-execute the wait or stop instruction.)
1 = Memory access command failed because the CPU entered wait or stop mode.
0 = Memory access did not conflict with a wait or stop instruction.

DVF — Data Valid Failure Status

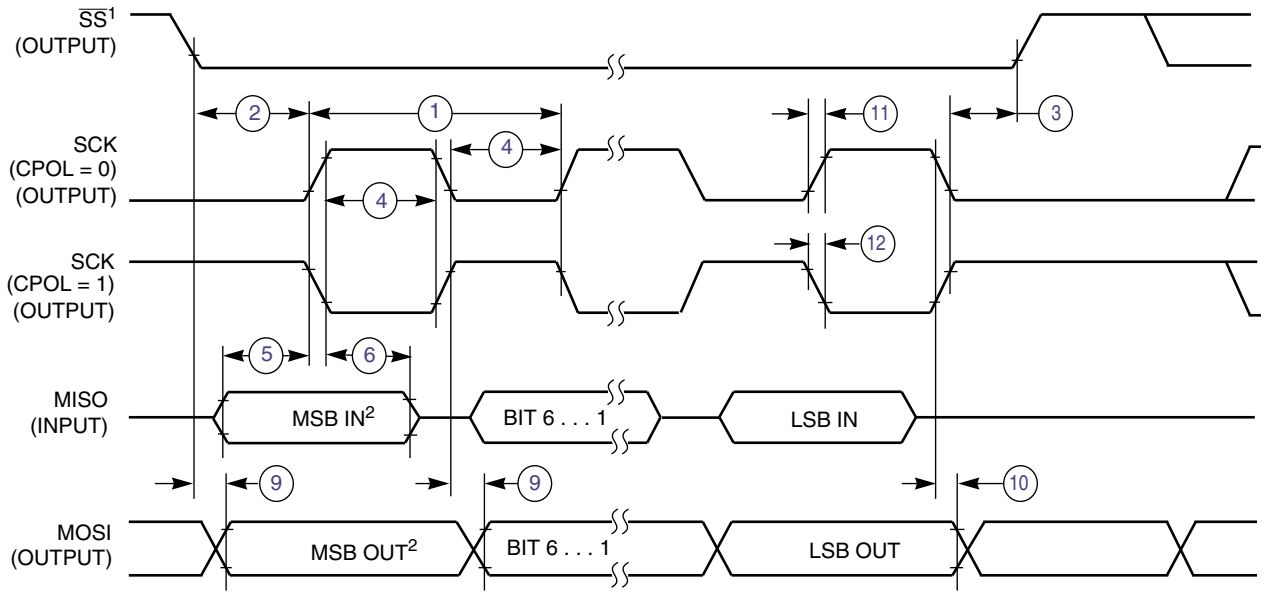
This status bit is not used in the MC9S08GB/GT because it does not have any slow access memory.
1 = Memory access command failed because CPU was not finished with a slow memory access.
0 = Memory access did not conflict with a slow memory access.

15.5.1.2 BDC Breakpoint Match Register (BDCBKPT)

This 16-bit register holds the address for the hardware breakpoint in the BDC. The BKPTEN and FTS control bits in BDCSCR are used to enable and configure the breakpoint logic. Dedicated serial BDC commands (READ_BKPT and WRITE_BKPT) are used to read and write the BDCBKPT register but is not accessible to user programs because it is not located in the normal memory map of the MCU. Breakpoints are normally set while the target MCU is in active background mode before running the user application program. For additional information about setup and use of the hardware breakpoint logic in the BDC, refer to Section 15.3.4, “BDC Hardware Breakpoint.”

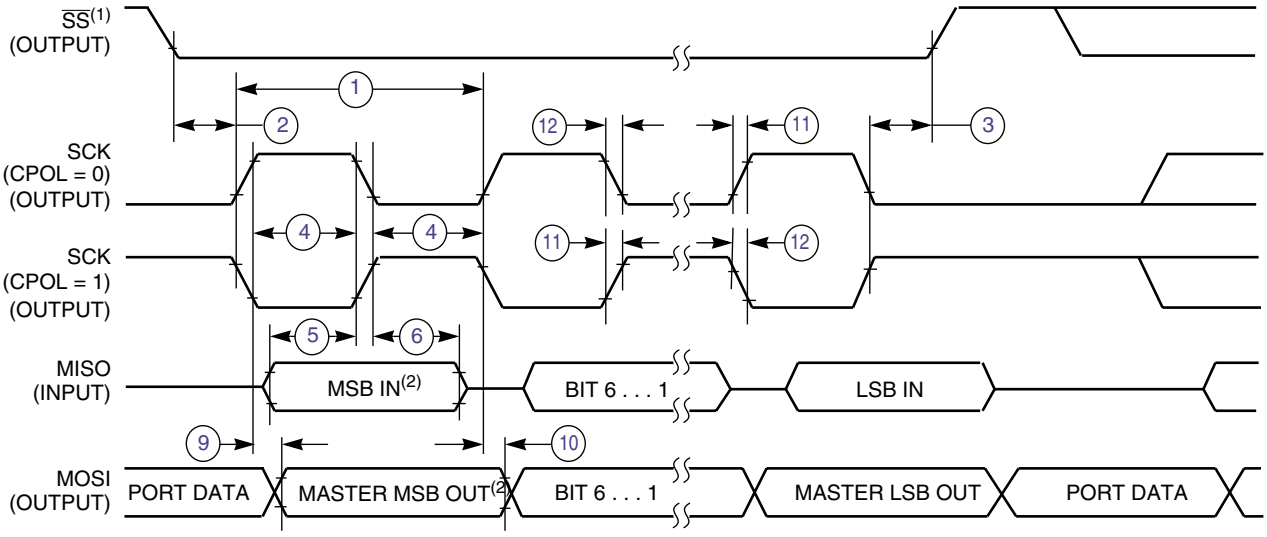
15.5.2 System Background Debug Force Reset Register (SBDFR)

This register contains a single write-only control bit. A serial active background mode command such as WRITE_BYTE must be used to write to SBDFR. Attempts to write this register from a user program are ignored. Reads always return $00.



NOTES:
1. $\overline{SS}$ output mode (DDS7 = 1, SSOE = 1).
2. LSBF = 0. For LSBF = 1, bit order is LSB, bit 1, ..., bit 6, MSB.

Figure A-16. SPI Master Timing (CPHA = 0)



NOTES:
1. $\overline{SS}$ output mode (DDS7 = 1, SSOE = 1).
2. LSBF = 0. For LSBF = 1, bit order is LSB, bit 1, ..., bit 6, MSB.

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