



Welcome to [E-XFL.COM](https://www.e-xfl.com)

What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Active
Core Processor	S08
Core Size	8-Bit
Speed	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 ~ 105°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/mc9s08aw60vfge



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; RI_{DD} 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.

Section Number	Title	Page
5.9.8	System Power Management Status and Control 1 Register (SPMSC1)	79
5.9.9	System Power Management Status and Control 2 Register (SPMSC2)	80

Chapter 6 Parallel Input/Output

6.1	Introduction	81
6.2	Features	81
6.3	Pin Descriptions	82
6.3.1	Port A	82
6.3.2	Port B	82
6.3.3	Port C	83
6.3.4	Port D	83
6.3.5	Port E	84
6.3.6	Port F	85
6.3.7	Port G	85
6.4	Parallel I/O Control	86
6.5	Pin Control	87
6.5.1	Internal Pullup Enable	87
6.5.2	Output Slew Rate Control Enable	87
6.5.3	Output Drive Strength Select	87
6.6	Pin Behavior in Stop Modes	88
6.7	Parallel I/O and Pin Control Registers	88
6.7.1	Port A I/O Registers (PTAD and PTADD)	88
6.7.2	Port A Pin Control Registers (PTAPE, PTASE, PTADS)	89
6.7.3	Port B I/O Registers (PTBD and PTBDD)	91
6.7.4	Port B Pin Control Registers (PTBPE, PTBSE, PTBDS)	92
6.7.5	Port C I/O Registers (PTCD and PTCDD)	94
6.7.6	Port C Pin Control Registers (PTCPE, PTCSE, PTCDS)	95
6.7.7	Port D I/O Registers (PTDD and PTDDD)	97
6.7.8	Port D Pin Control Registers (PTDPE, PTDSE, PTDDS)	98
6.7.9	Port E I/O Registers (PTED and PTEDD)	100
6.7.10	Port E Pin Control Registers (PTEPE, PTESE, PTEDS)	101
6.7.11	Port F I/O Registers (PTFD and PTFDD)	103
6.7.12	Port F Pin Control Registers (PTFPE, PTFSE, PTFDS)	104
6.7.13	Port G I/O Registers (PTGD and PTGDD)	106
6.7.14	Port G Pin Control Registers (PTGPE, PTGSE, PTGDS)	107

Chapter 7 Central Processor Unit (S08CPUV2)

7.1	Introduction	109
7.1.1	Features	109
7.2	Programmer's Model and CPU Registers	110

Table 2-1. Pin Sharing Priority

Lowest <- Pin Function Priority -> Highest			Reference ¹
Port Pins	Alternate Function	Alternate Function	
PTF3–PTF0	TPM1CH5– TPM1CH2		Chapter 10, “Timer/PWM (S08TPMV2)”
PTG4–PTG0	KBI1P4–KBI1P0		Chapter 9, “Keyboard Interrupt (S08KBIV1)”
PTG6–PTG5	EXTAL–XTAL		Chapter 8, “Internal Clock Generator (S08ICGV4)”

¹ See the listed chapter for information about modules that share these pins.

When an on-chip peripheral system is controlling a pin, data direction control bits still determine what is read from port data registers even though the peripheral module controls the pin direction by controlling the enable for the pin’s output buffer. See the [Chapter 6, “Parallel Input/Output”](#) chapter for more details.

Pullup enable bits for each input pin control whether on-chip pullup devices are enabled whenever the pin is acting as an input even if it is being controlled by an on-chip peripheral module. When the PTD7, PTD3, PTD2, and PTG4 pins are controlled by the KBI module and are configured for rising-edge/high-level sensitivity, the pullup enable control bits enable pulldown devices rather than pullup devices.

NOTE

When an alternative function is first enabled it is possible to get a spurious edge to the module, user software should clear out any associated flags before interrupts are enabled. [Table 2-1](#) illustrates the priority if multiple modules are enabled. The highest priority module will have control over the pin. Selecting a higher priority pin function with a lower priority function already enabled can cause spurious edges to the lower priority module. It is recommended that all modules that share a pin be disabled before enabling another module.

- Illegal opcode detect
- Background debug forced reset
- The reset pin ($\overline{\text{RESET}}$)
- Clock generator loss of lock and loss of clock reset

Each of these sources, with the exception of the background debug forced reset, has an associated bit in the system reset status register. Whenever the MCU enters reset, the internal clock generator (ICG) module switches to self-clocked mode with the frequency of $f_{\text{Self_reset}}$ selected. The reset pin is driven low for 34 bus cycles where the internal bus frequency is half the ICG frequency. After the 34 bus cycles are completed, the pin is released and will be pulled up by the internal pullup resistor, unless it is held low externally. After the pin is released, it is sampled after another 38 bus cycles to determine whether the reset pin is the cause of the MCU reset.

5.4 Computer Operating Properly (COP) Watchdog

The COP watchdog is intended to force a system reset when the application software fails to execute as expected. To prevent a system reset from the COP timer (when it is enabled), application software must reset the COP timer periodically. If the application program gets lost and fails to reset the COP before it times out, a system reset is generated to force the system back to a known starting point. The COP watchdog is enabled by the COPE bit in SOPT (see [Section 5.9.4, “System Options Register \(SOPT\)”](#) for additional information). The COP timer is reset by writing any value to the address of SRS. This write does not affect the data in the read-only SRS. Instead, the act of writing to this address is decoded and sends a reset signal to the COP timer.

After any reset, the COP timer is enabled. This provides a reliable way to detect code that is not executing as intended. If the COP watchdog is not used in an application, it can be disabled by clearing the COPE bit in the write-once SOPT register. Also, the COPT bit can be used to choose one of two timeout periods (2^{18} or 2^{13} cycles of the bus rate clock). Even if the application will use the reset default settings in COPE and COPT, the user should write to write-once SOPT during reset initialization to lock in the settings. That way, they cannot be changed accidentally if the application program gets lost.

The write to SRS that services (clears) the COP timer should not be placed in an interrupt service routine (ISR) because the ISR could continue to be executed periodically even if the main application program fails.

When the MCU is in active background mode, the COP timer is temporarily disabled.

5.5 Interrupts

Interrupts provide a way to save the current CPU status and registers, execute an interrupt service routine (ISR), and then restore the CPU status so processing resumes where it left off before the interrupt. Other than the software interrupt (SWI), which is a program instruction, interrupts are caused by hardware events such as an edge on the IRQ pin or a timer-overflow event. The debug module can also generate an SWI under certain circumstances.

If an event occurs in an enabled interrupt source, an associated read-only status flag will become set. The CPU will not respond until and unless the local interrupt enable is a logic 1 to enable the interrupt. The

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

	7	6	5	4	3	2	1 ¹	0
R	LVDF	0	LVDIE	LVDRE ⁽²⁾	LVDSE ⁽²⁾	LVDE ⁽²⁾		BGBE
W		LVDACK						
Reset	0	0	0	1	1	1	0	0

= Unimplemented or Reserved

¹ Bit 1 is a reserved bit that must always be written to 0.

² This bit can be written only one time after reset. Additional writes are ignored.

Figure 5-10. System Power Management Status and Control 1 Register (SPMSC1)

Table 5-11. SPMSC1 Register Field Descriptions

Field	Description
7 LVDF	Low-Voltage Detect Flag — Provided LVDE = 1, this read-only status bit indicates a low-voltage detect event.
6 LVDACK	Low-Voltage Detect Acknowledge — This write-only bit is used to acknowledge low voltage detection errors (write 1 to clear LVDF). Reads always return 0.
5 LVDIE	Low-Voltage Detect Interrupt Enable — This read/write bit enables hardware interrupt requests for LVDF. 0 Hardware interrupt disabled (use polling). 1 Request a hardware interrupt when LVDF = 1.
4 LVDRE	Low-Voltage Detect Reset Enable — This read/write bit enables LVDF events to generate a hardware reset (provided LVDE = 1). 0 LVDF does not generate hardware resets. 1 Force an MCU reset when LVDF = 1.
3 LVDSE	Low-Voltage Detect Stop Enable — Provided LVDE = 1, this read/write bit determines whether the low-voltage detect function operates when the MCU is in stop mode. 0 Low-voltage detect disabled during stop mode. 1 Low-voltage detect enabled during stop mode.
2 LVDE	Low-Voltage Detect Enable — This read/write bit enables low-voltage detect logic and qualifies the operation of other bits in this register. 0 LVD logic disabled. 1 LVD logic enabled.
0 BGBE	Bandgap Buffer Enable — The BGBE bit is used to enable an internal buffer for the bandgap voltage reference for use by the ADC module on one of its internal channels. 0 Bandgap buffer disabled. 1 Bandgap buffer enabled.

Chapter 7

Central Processor Unit (S08CPUV2)

7.1 Introduction

This section provides summary information about the registers, addressing modes, and instruction set of the CPU of the HCS08 family. For a more detailed discussion, refer to the *HCS08 Family Reference Manual, volume 1*, Freescale Semiconductor document order number HCS08RMV1/D.

The HCS08 CPU is fully source- and object-code-compatible with the M68HC08 CPU. Several instructions and enhanced addressing modes were added to improve C compiler efficiency and to support a new background debug system which replaces the monitor mode of earlier M68HC08 microcontrollers (MCU).

7.1.1 Features

Features of the HCS08 CPU include:

- Object code fully upward-compatible with M68HC05 and M68HC08 Families
- All registers and memory are mapped to a single 64-Kbyte address space
- 16-bit stack pointer (any size stack anywhere in 64-Kbyte address space)
- 16-bit index register (H:X) with powerful indexed addressing modes
- 8-bit accumulator (A)
- Many instructions treat X as a second general-purpose 8-bit register
- Seven addressing modes:
 - Inherent — Operands in internal registers
 - Relative — 8-bit signed offset to branch destination
 - Immediate — Operand in next object code byte(s)
 - Direct — Operand in memory at 0x0000–0x00FF
 - Extended — Operand anywhere in 64-Kbyte address space
 - Indexed relative to H:X — Five submodes including auto increment
 - Indexed relative to SP — Improves C efficiency dramatically
- Memory-to-memory data move instructions with four address mode combinations
- Overflow, half-carry, negative, zero, and carry condition codes support conditional branching on the results of signed, unsigned, and binary-coded decimal (BCD) operations
- Efficient bit manipulation instructions
- Fast 8-bit by 8-bit multiply and 16-bit by 8-bit divide instructions
- STOP and WAIT instructions to invoke low-power operating modes

7.4.5 BGND Instruction

The BGND instruction is new to the HCS08 compared to the M68HC08. BGND would not be used in normal user programs because it forces the CPU to stop processing user instructions and enter the active background mode. The only way to resume execution of the user program is through reset or by a host debug system issuing a GO, TRACE1, or TAGGO serial command through the background debug interface.

Software-based breakpoints can be set by replacing an opcode at the desired breakpoint address with the BGND opcode. When the program reaches this breakpoint address, the CPU is forced to active background mode rather than continuing the user program.

Table 7-2. HCS08 Instruction Set Summary (Sheet 4 of 7)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Bus Cycles ¹
			V	H	I	N	Z	C				
CPX #opr8i CPX opr8a CPX opr16a CPX oprx16,X CPX oprx8,X CPX ,X CPX oprx16,SP CPX oprx8,SP	Compare X (Index Register Low) with Memory	$(X) - (M)$ (CCR Updated But Operands Not Changed)	↑	—	—	↑	↑	↑	IMM DIR EXT IX2 IX1 IX SP2 SP1	A3 B3 C3 D3 E3 F3 9ED3 9EE3	ii dd hh ll ee ff ff ff ff ff	2 3 4 4 3 3 5 4
DAA	Decimal Adjust Accumulator After ADD or ADC of BCD Values	$(A)_{10}$	U	—	—	↑	↑	↑	INH	72		1
DBNZ opr8a,rel DBNZ rel DBNZX rel DBNZ oprx8,X,rel DBNZ ,X,rel DBNZ oprx8,SP,rel	Decrement and Branch if Not Zero	Decrement A, X, or M Branch if (result) ≠ 0 DBNZX Affects X Not H	—	—	—	—	—	—	DIR INH INH IX1 IX SP1	3B 4B 5B 6B 7B 9E6B	dd rr rr rr rr rr rr rr	7 4 4 7 6 8
DEC opr8a DECA DECX DEC oprx8,X DEC ,X DEC oprx8,SP	Decrement	$M \leftarrow (M) - 0x01$ $A \leftarrow (A) - 0x01$ $X \leftarrow (X) - 0x01$ $M \leftarrow (M) - 0x01$ $M \leftarrow (M) - 0x01$ $M \leftarrow (M) - 0x01$	↑	—	—	↑	↑	—	DIR INH INH IX1 IX SP1	3A 4A 5A 6A 7A 9E6A	dd ff ff ff ff ff	5 1 1 5 4 6
DIV	Divide	$A \leftarrow (H:A) \div (X)$ H ← Remainder	—	—	—	—	↑	↑	INH	52		6
EOR #opr8i EOR opr8a EOR opr16a EOR oprx16,X EOR oprx8,X EOR ,X EOR oprx16,SP EOR oprx8,SP	Exclusive OR Memory with Accumulator	$A \leftarrow (A \oplus M)$	0	—	—	↑	↑	—	IMM DIR EXT IX2 IX1 IX SP2 SP1	A8 B8 C8 D8 E8 F8 9ED8 9EE8	ii dd hh ll ee ff ff ff ff ff	2 3 4 4 3 3 5 4
INC opr8a INCA INCX INC oprx8,X INC ,X INC oprx8,SP	Increment	$M \leftarrow (M) + 0x01$ $A \leftarrow (A) + 0x01$ $X \leftarrow (X) + 0x01$ $M \leftarrow (M) + 0x01$ $M \leftarrow (M) + 0x01$ $M \leftarrow (M) + 0x01$	↑	—	—	↑	↑	—	DIR INH INH IX1 IX SP1	3C 4C 5C 6C 7C 9E6C	dd ff ff ff ff ff	5 1 1 5 4 6
JMP opr8a JMP opr16a JMP oprx16,X JMP oprx8,X JMP ,X	Jump	$PC \leftarrow \text{Jump Address}$	—	—	—	—	—	—	DIR EXT IX2 IX1 IX	BC CC DC EC FC	dd hh ll ee ff ff ff	3 4 4 3 3
JSR opr8a JSR opr16a JSR oprx16,X JSR oprx8,X JSR ,X	Jump to Subroutine	$PC \leftarrow (PC) + n$ ($n = 1, 2, \text{ or } 3$) Push (PCL); $SP \leftarrow (SP) - 0x0001$ Push (PCH); $SP \leftarrow (SP) - 0x0001$ $PC \leftarrow \text{Unconditional Address}$	—	—	—	—	—	—	DIR EXT IX2 IX1 IX	BD CD DD ED FD	dd hh ll ee ff ff ff	5 6 6 5 5
LDA #opr8i LDA opr8a LDA opr16a LDA oprx16,X LDA oprx8,X LDA ,X LDA oprx16,SP LDA oprx8,SP	Load Accumulator from Memory	$A \leftarrow (M)$	0	—	—	↑	↑	—	IMM DIR EXT IX2 IX1 IX SP2 SP1	A6 B6 C6 D6 E6 F6 9ED6 9EE6	ii dd hh ll ee ff ff ff ff ff	2 3 4 4 3 3 5 4
LDHX #opr16i LDHX opr8a LDHX opr16a LDHX ,X LDHX oprx16,X LDHX oprx8,X LDHX oprx8,SP	Load Index Register (H:X) from Memory	$H:X \leftarrow (M:M + 0x0001)$	0	—	—	↑	↑	—	IMM DIR EXT IX IX2 IX1 SP1	45 55 32 9EAE 9EBE 9ECE 9EFE	jj kk dd ll hh ll ee ff ff ff ff	3 4 5 5 6 5 5

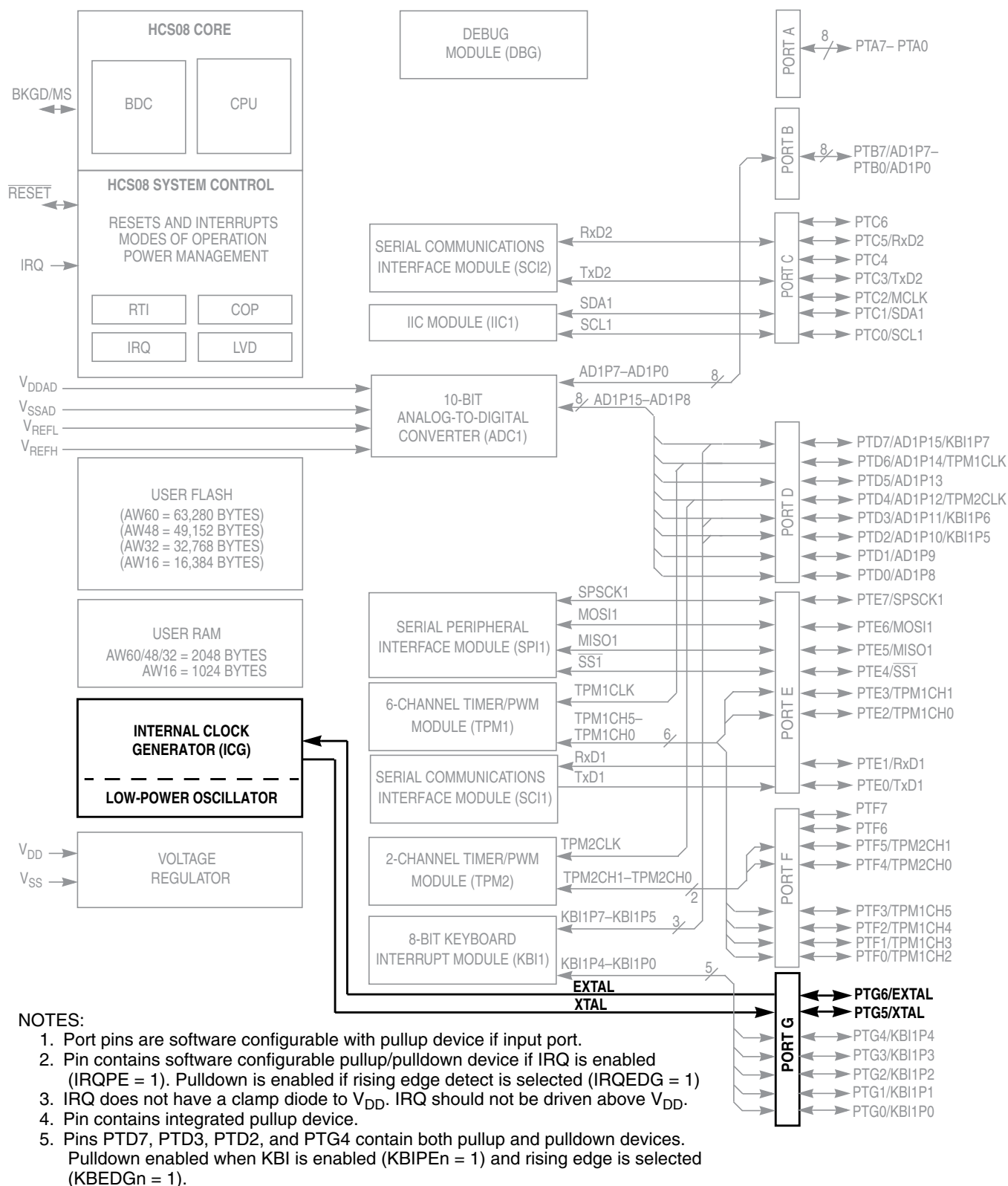


Figure 8-2. Block Diagram Highlighting ICG Module

- Digitally-controlled oscillator (DCO) preserves previous frequency settings, allowing fast frequency lock when recovering from stop3 mode
- DCO will maintain operating frequency during a loss or removal of reference clock
- Post-FLL divider selects 1 of 8 bus rate divisors (/1 through /128)
- Separate self-clocked source for real-time interrupt
- Trimmable internal clock source supports SCI communications without additional external components
- Automatic FLL engagement after lock is acquired
- External oscillator selectable for low power or high gain

8.1.2 Modes of Operation

This is a high-level description only. Detailed descriptions of operating modes are contained in [Section 8.4, “Functional Description.”](#)

- Mode 1 — Off
The output clock, ICGOUT, is static. This mode may be entered when the STOP instruction is executed.
- Mode 2 — Self-clocked (SCM)
Default mode of operation that is entered immediately after reset. The ICG’s FLL is open loop and the digitally controlled oscillator (DCO) is free running at a frequency set by the filter bits.
- Mode 3 — FLL engaged internal (FEI)
In this mode, the ICG’s FLL is used to create frequencies that are programmable multiples of the internal reference clock.
 - FLL engaged internal unlocked is a transition state that occurs while the FLL is attempting to lock. The FLL DCO frequency is off target and the FLL is adjusting the DCO to match the target frequency.
 - FLL engaged internal locked is a state that occurs when the FLL detects that the DCO is locked to a multiple of the internal reference.
- Mode 4 — FLL bypassed external (FBE)
In this mode, the ICG is configured to bypass the FLL and use an external clock as the clock source.
- Mode 5 — FLL engaged external (FEE)
The ICG’s FLL is used to generate frequencies that are programmable multiples of the external clock reference.
 - FLL engaged external unlocked is a transition state that occurs while the FLL is attempting to lock. The FLL DCO frequency is off target and the FLL is adjusting the DCO to match the target frequency.
 - FLL engaged external locked is a state which occurs when the FLL detects that the DCO is locked to a multiple of the internal reference.

selected, this pin is not used by the ICG. The oscillator is capable of being configured to provide a higher amplitude output for improved noise immunity. This mode of operation is selected by $HGO = 1$.

8.2.3 External Clock Connections

If an external clock is used, then the pins are connected as shown [Figure 8-4](#).

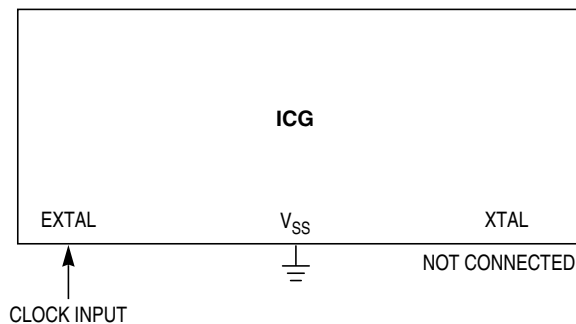


Figure 8-4. External Clock Connections

8.2.4 External Crystal/Resonator Connections

If an external crystal/resonator frequency reference is used, then the pins are connected as shown in [Figure 8-5](#). Recommended component values are listed in the [Electrical Characteristics](#) chapter.

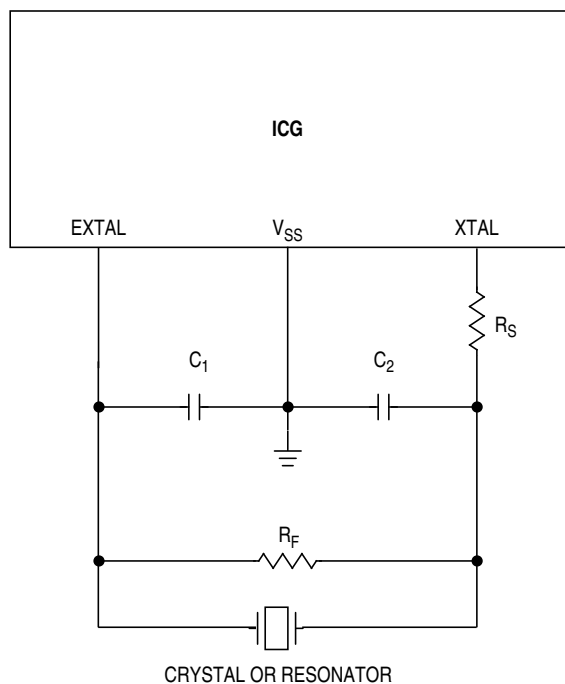


Figure 8-5. External Frequency Reference Connection

Table 8-12. MFD and RFD Decode Table

101	14	101	+32
110	16	110	+64
111	18	111	+128

8.5.2 Example #1: External Crystal = 32 kHz, Bus Frequency = 4.19 MHz

In this example, the FLL will be used (in FEE mode) to multiply the external 32 kHz oscillator up to 8.38 MHz to achieve 4.19 MHz bus frequency.

After the MCU is released from reset, the ICG is in self-clocked mode (SCM) and supplies approximately 8 MHz on ICGOUT, which corresponds to a 4 MHz bus frequency (f_{Bus}).

The clock scheme will be FLL engaged, external (FEE). So

$$f_{ICGOUT} = f_{ext} * P * N / R ; P = 64, f_{ext} = 32 \text{ kHz} \quad \text{Eqn. 8-1}$$

Solving for N / R gives:

$$N / R = 8.38 \text{ MHz} / (32 \text{ kHz} * 64) = 4 ; \text{ we can choose } N = 4 \text{ and } R = 1 \quad \text{Eqn. 8-2}$$

The values needed in each register to set up the desired operation are:

ICGC1 = \$38 (%00111000)

Bit 7	HGO	0	Configures oscillator for low power
Bit 6	RANGE	0	Configures oscillator for low-frequency range; FLL prescale factor is 64
Bit 5	REFS	1	Oscillator using crystal or resonator is requested
Bits 4:3	CLKS	11	FLL engaged, external reference clock mode
Bit 2	OSCSTEN	0	Oscillator disabled
Bit 1	LOCD	0	Loss-of-clock detection enabled
Bit 0		0	Unimplemented or reserved, always reads zero

ICGC2 = \$00 (%00000000)

Bit 7	LOLRE	0	Generates an interrupt request on loss of lock
Bits 6:4	MFD	000	Sets the MFD multiplication factor to 4
Bit 3	LOCRES	0	Generates an interrupt request on loss of clock
Bits 2:0	RFD	000	Sets the RFD division factor to +1

ICGS1 = \$xx

This is read only except for clearing interrupt flag

ICGS2 = \$xx

This is read only; should read DCOS = 1 before performing any time critical tasks

ICGFLTLU/L = \$xx

Only needed in self-clocked mode; FLT will be adjusted by loop to give 8.38 MHz DCO clock
 Bits 15:12 unused 0000

8.5.3 Example #2: External Crystal = 4 MHz, Bus Frequency = 20 MHz

In this example, the FLL will be used (in FEE mode) to multiply the external 4 MHz oscillator up to 40-MHz to achieve 20 MHz bus frequency.

After the MCU is released from reset, the ICG is in self-clocked mode (SCM) and supplies approximately 8 MHz on ICGOUT which corresponds to a 4 MHz bus frequency (f_{Bus}).

During reset initialization software, the clock scheme will be set to FLL engaged, external (FEE). So

$$f_{ICGOUT} = f_{ext} * P * N / R ; P = 1, f_{ext} = 4.00 \text{ MHz} \quad \text{Eqn. 8-3}$$

Solving for N / R gives:

$$N / R = 40 \text{ MHz} / (4 \text{ MHz} * 1) = 10 ; \text{ We can choose } N = 10 \text{ and } R = 1 \quad \text{Eqn. 8-4}$$

The values needed in each register to set up the desired operation are:

ICGC1 = \$78 (%01111000)

Bit 7	HGO	0	Configures oscillator for low power
Bit 6	RANGE	1	Configures oscillator for high-frequency range; FLL prescale factor is 1
Bit 5	REFS	1	Requests an oscillator
Bits 4:3	CLKS	11	FLL engaged, external reference clock mode
Bit 2	OSCSTEN	0	Disables the oscillator
Bit 1	LOCD	0	Loss-of-clock detection enabled
Bit 0		0	Unimplemented or reserved, always reads zero

ICGC2 = \$30 (%00110000)

Bit 7	LOLRE	0	Generates an interrupt request on loss of lock
Bit 6:4	MFD	011	Sets the MFD multiplication factor to 10
Bit 3	LOCRES	0	Generates an interrupt request on loss of clock
Bit 2:0	RFD	000	Sets the RFD division factor to ÷1

ICGS1 = \$xx

This is read only except for clearing interrupt flag

ICGS2 = \$xx

This is read only. Should read DCOS before performing any time critical tasks

ICGFLTLU/L = \$xx

Not used in this example

ICGTRM

Not used in this example

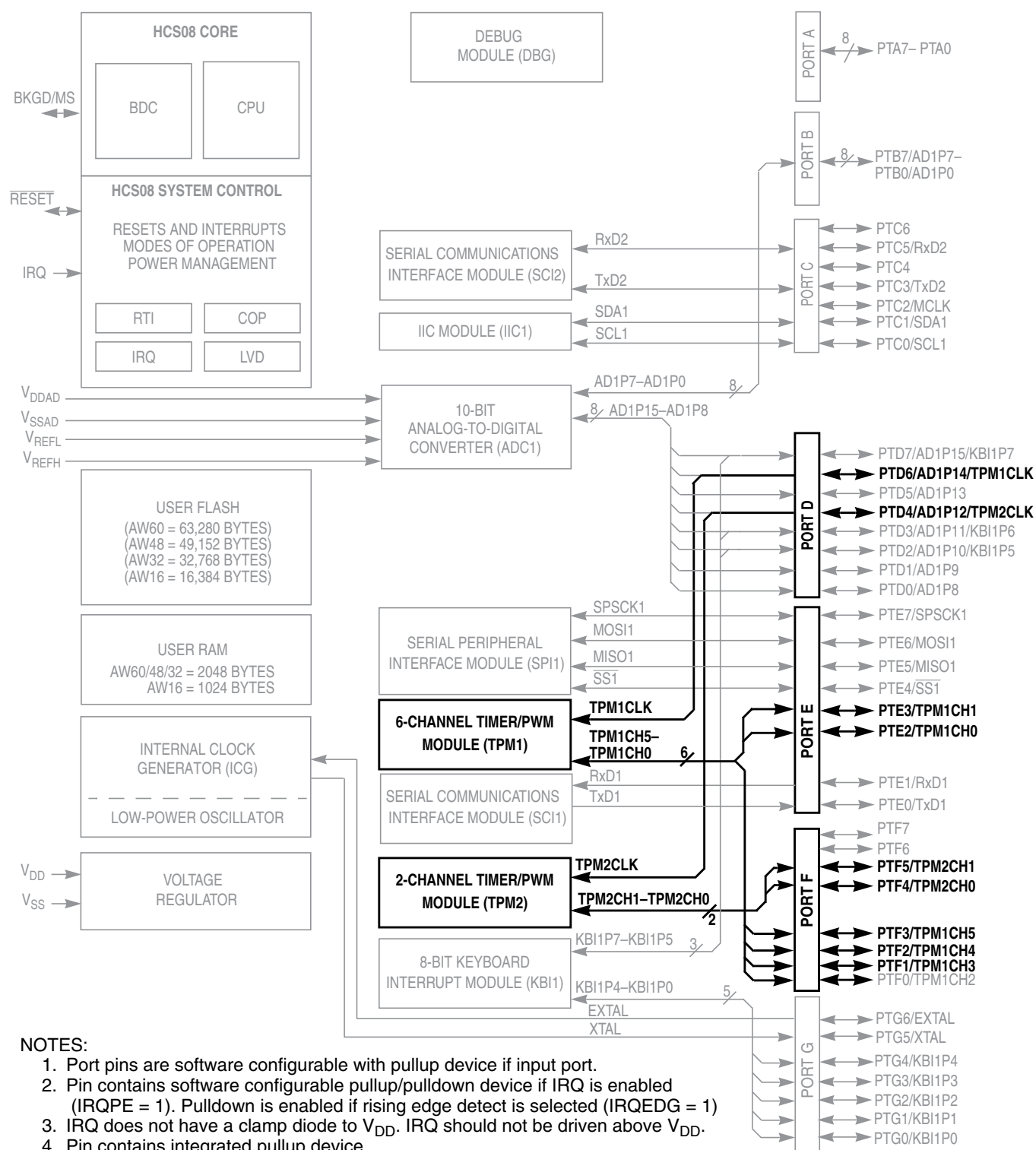


Figure 10-1. Block Diagram Highlighting the TPM Module

Freescall-provided equate or header file is used to translate these names into the appropriate absolute addresses.

Some MCU systems have more than one TPM, so register names include placeholder characters to identify which TPM and which channel is being referenced. For example, TPMxCnSC refers to timer (TPM) x, channel n and TPM1C2SC is the status and control register for timer 1, channel 2.

10.4.1 Timer x Status and Control Register (TPMxSC)

TPMxSC contains the overflow status flag and control bits that are used to configure the interrupt enable, TPM configuration, clock source, and prescale divisor. These controls relate to all channels within this timer module.

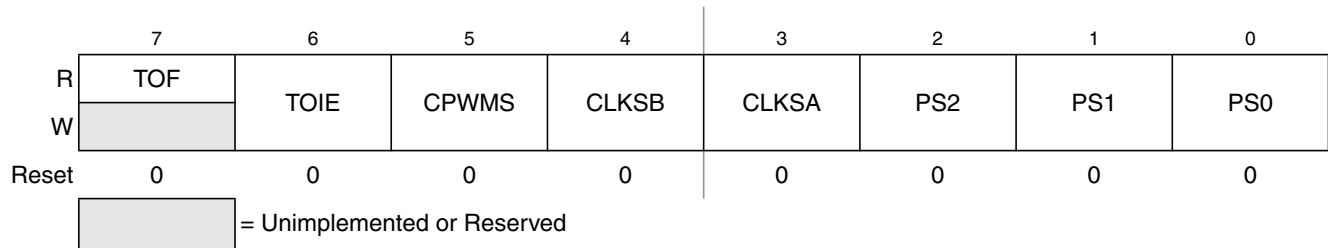


Figure 10-3. Timer x Status and Control Register (TPMxSC)

Table 10-1. TPMxSC Register Field Descriptions

Field	Description
7 TOF	Timer Overflow Flag — This flag is set when the TPM counter changes to 0x0000 after reaching the modulo value programmed in the TPM counter modulo registers. When the TPM is configured for CPWM, TOF is set after the counter has reached the value in the modulo register, at the transition to the next lower count value. Clear TOF by reading the TPM status and control register when TOF is set and then writing a 0 to TOF. If another TPM overflow occurs before the clearing sequence is complete, the sequence is reset so TOF would remain set after the clear sequence was completed for the earlier TOF. Reset clears TOF. Writing a 1 to TOF has no effect. 0 TPM counter has not reached modulo value or overflow 1 TPM counter has overflowed
6 TOIE	Timer Overflow Interrupt Enable — This read/write bit enables TPM overflow interrupts. If TOIE is set, an interrupt is generated when TOF equals 1. Reset clears TOIE. 0 TOF interrupts inhibited (use software polling) 1 TOF interrupts enabled
5 CPWMS	Center-Aligned PWM Select — This read/write bit selects CPWM operating mode. Reset clears this bit so the TPM operates in up-counting mode for input capture, output compare, and edge-aligned PWM functions. Setting CPWMS reconfigures the TPM to operate in up-/down-counting mode for CPWM functions. Reset clears CPWMS. 0 All TPMx channels operate as input capture, output compare, or edge-aligned PWM mode as selected by the MSnB:MSnA control bits in each channel's status and control register 1 All TPMx channels operate in center-aligned PWM mode
4:3 CLKS[B:A]	Clock Source Select — As shown in Table 10-2, this 2-bit field is used to disable the TPM system or select one of three clock sources to drive the counter prescaler. The external source and the XCLK are synchronized to the bus clock by an on-chip synchronization circuit.
2:0 PS[2:0]	Prescale Divisor Select — This 3-bit field selects one of eight divisors for the TPM clock input as shown in Table 10-3. This prescaler is located after any clock source synchronization or clock source selection, so it affects whatever clock source is selected to drive the TPM system.

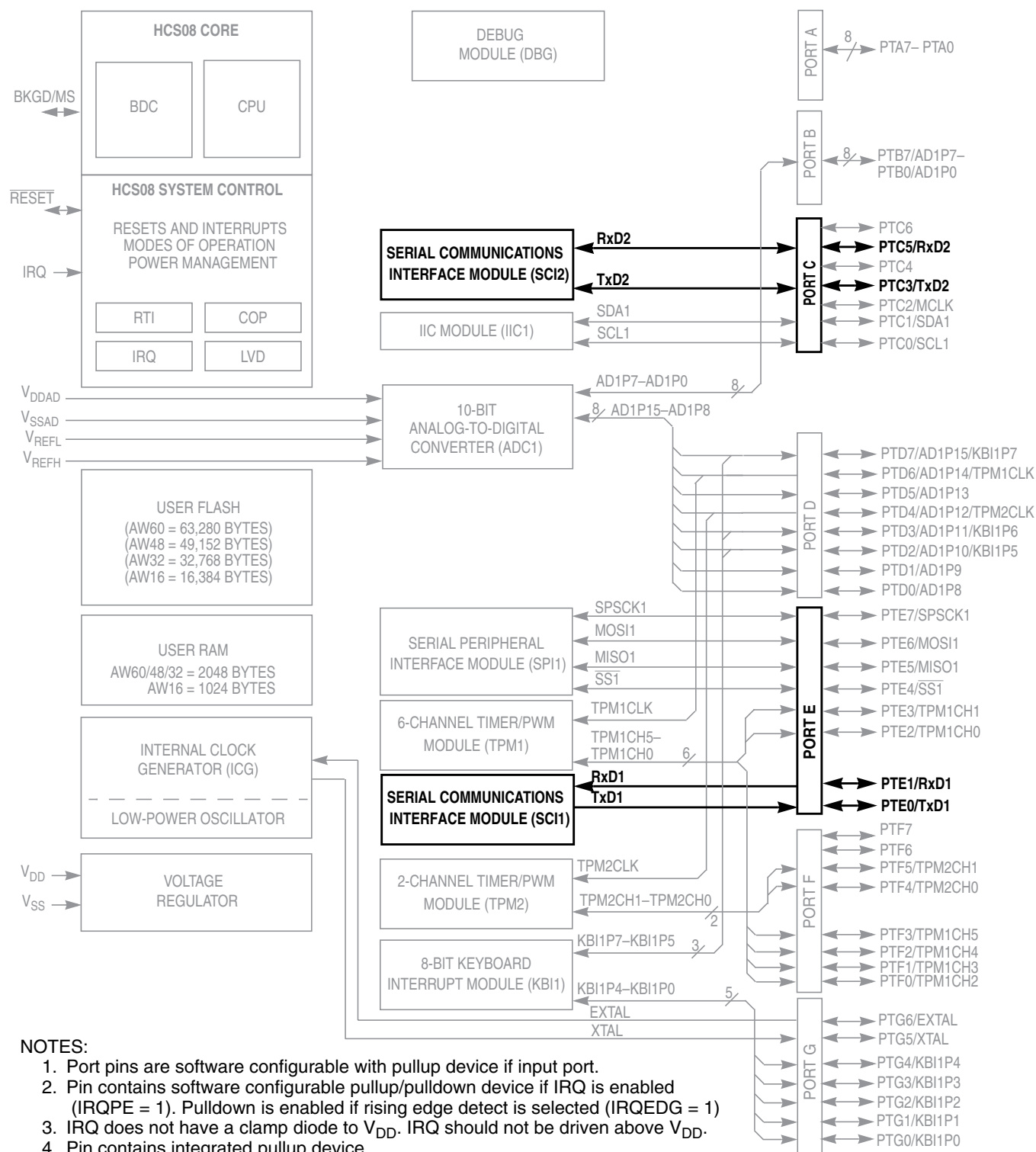


Figure 11-1. Block Diagram Highlighting the SCI Modules

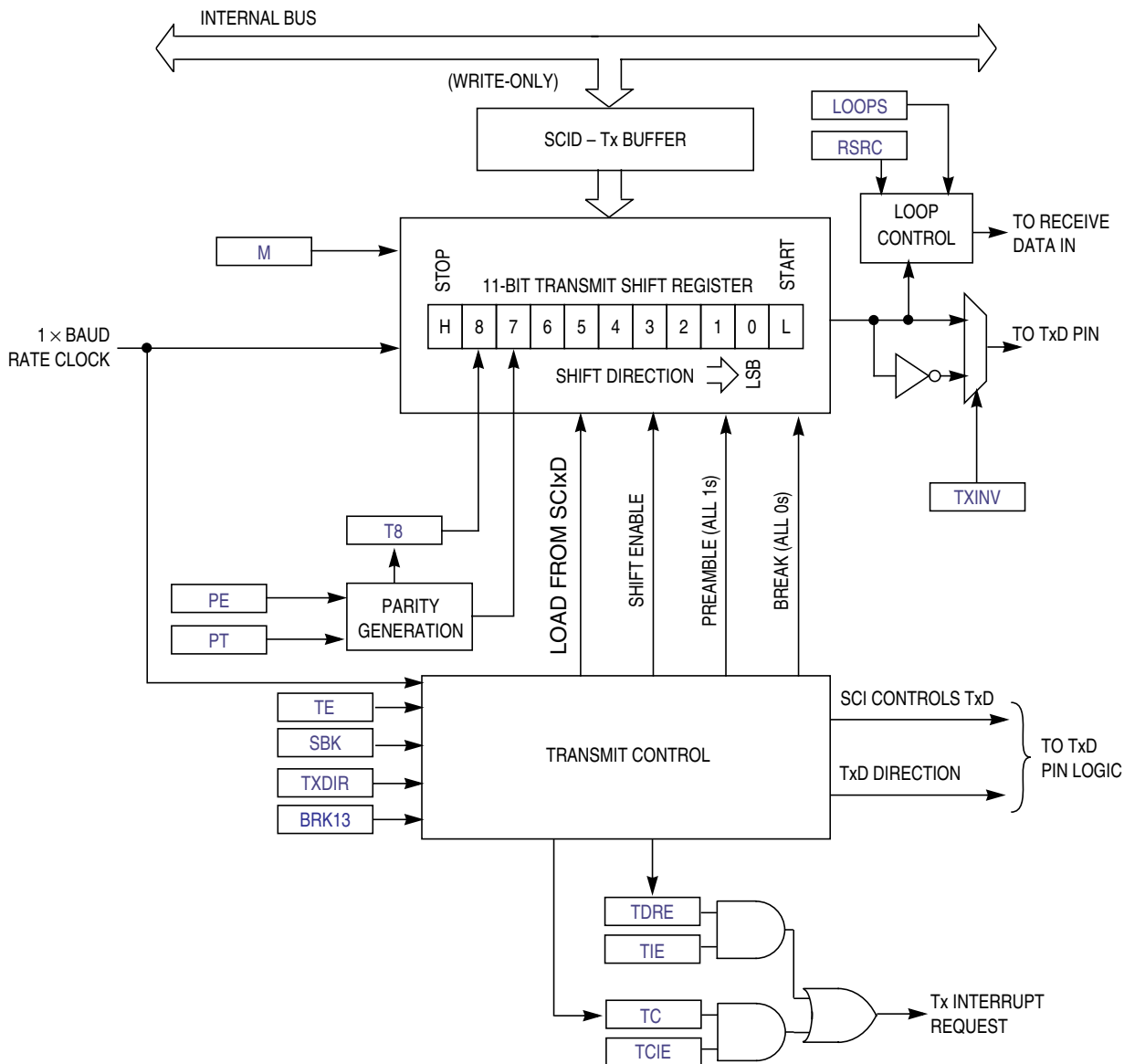


Figure 11-2. SCI Transmitter Block Diagram

13.3.4 IIC Status Register (IIC1S)

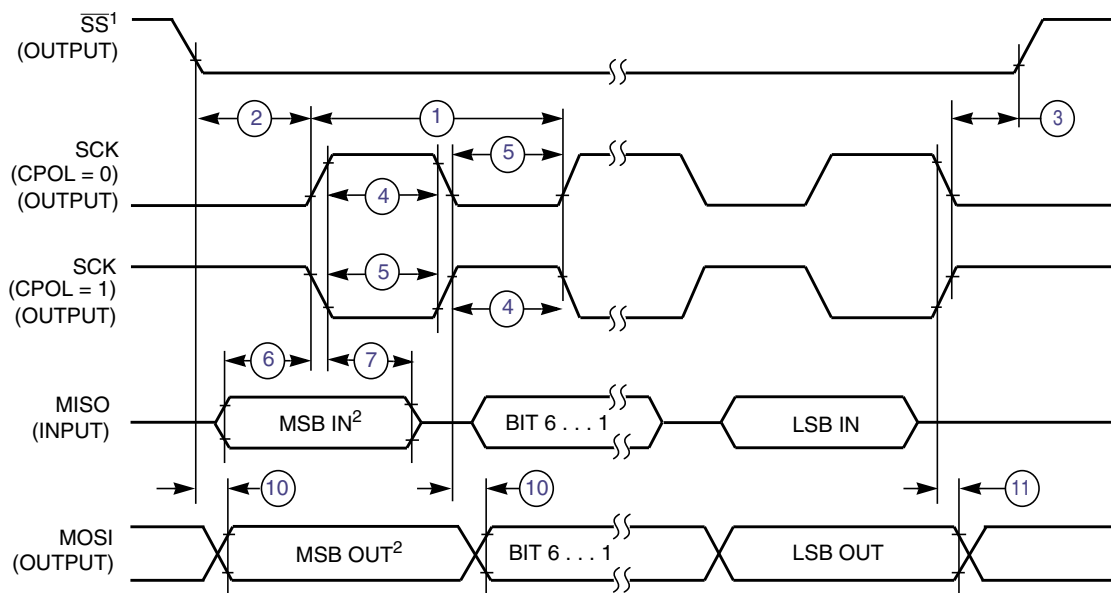
	7	6	5	4	3	2	1	0
R	TCF		BUSY	ARBL	0	SRW	IICIF	RXAK
W		IAAS						
Reset	1	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 13-6. IIC Status Register (IIC1S)

Table 13-5. IIC1S Register Field Descriptions

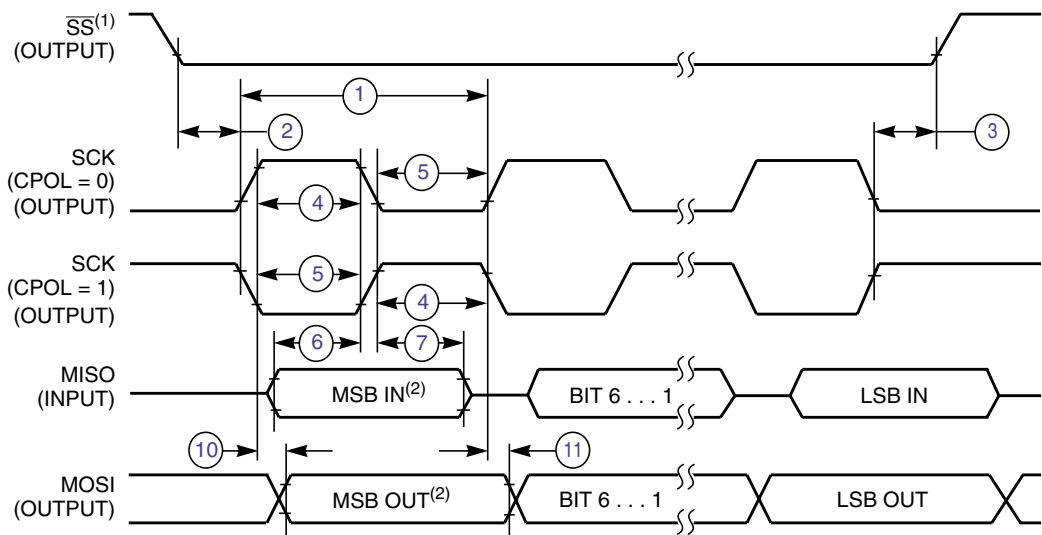
Field	Description
7 TCF	Transfer Complete Flag — This bit is set on the completion of a byte transfer. Note that this bit is only valid during or immediately following a transfer to the IIC module or from the IIC module. The TCF bit is cleared by reading the IIC1D register in receive mode or writing to the IIC1D in transmit mode. 0 Transfer in progress. 1 Transfer complete.
6 IAAS	Addressed as a Slave — The IAAS bit is set when the calling address matches the programmed slave address. Writing the IIC1C register clears this bit. 0 Not addressed. 1 Addressed as a slave.
5 BUSY	Bus Busy — The BUSY bit indicates the status of the bus regardless of slave or master mode. The BUSY bit is set when a START signal is detected and cleared when a STOP signal is detected. 0 Bus is idle. 1 Bus is busy.
4 ARBL	Arbitration Lost — This bit is set by hardware when the arbitration procedure is lost. The ARBL bit must be cleared by software, by writing a one to it. 0 Standard bus operation. 1 Loss of arbitration.
2 SRW	Slave Read/Write — When addressed as a slave the SRW bit indicates the value of the R/W command bit of the calling address sent to the master. 0 Slave receive, master writing to slave. 1 Slave transmit, master reading from slave.
1 IICIF	IIC Interrupt Flag — The IICIF bit is set when an interrupt is pending. This bit must be cleared by software, by writing a one to it in the interrupt routine. One of the following events can set the IICIF bit: <ul style="list-style-type: none"> One byte transfer completes Match of slave address to calling address Arbitration lost 0 No interrupt pending. 1 Interrupt pending.
0 RXAK	Receive Acknowledge — When the RXAK bit is low, it indicates an acknowledge signal has been received after the completion of one byte of data transmission on the bus. If the RXAK bit is high it means that no acknowledge signal is detected. 0 Acknowledge received. 1 No acknowledge received.



NOTES:

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

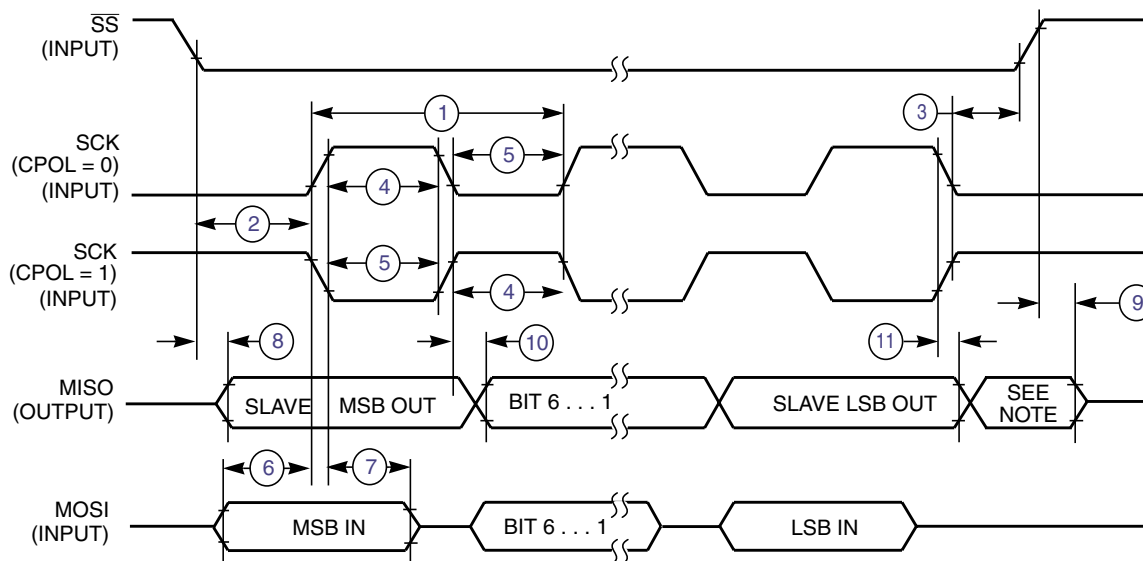
Figure A-16. SPI Master Timing (CPHA = 0)



NOTES:

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

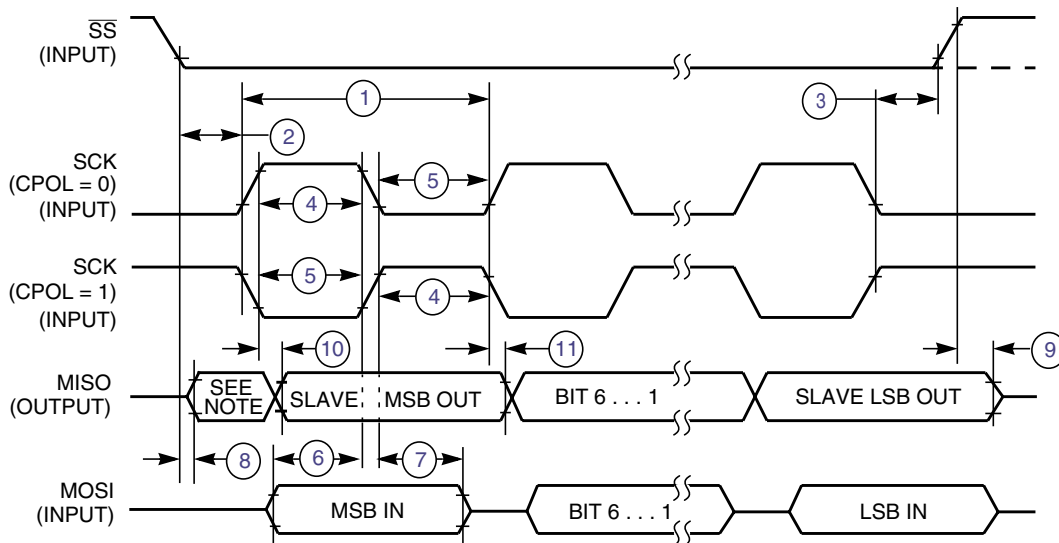
Figure A-17. SPI Master Timing (CPHA = 1)



NOTE:

1. Not defined but normally MSB of character just received

Figure A-18. SPI Slave Timing (CPHA = 0)



NOTE:

1. Not defined but normally LSB of character just received

Figure A-19. SPI Slave Timing (CPHA = 1)