



Welcome to [E-FL.COM](https://www.e-fl.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	20MHz
Connectivity	I <sup>2</sup> C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	4
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 4x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	8-DIP (0.300", 7.62mm)
Supplier Device Package	8-PDIP
Purchase URL	<a href="https://www.e-fl.com/product-detail/nxp-semiconductors/mc9s08qg44cpae">https://www.e-fl.com/product-detail/nxp-semiconductors/mc9s08qg44cpae</a>

# MC9S08QG8/4 Features

## 8-Bit HCS08 Central Processor Unit (CPU)

---

- 20-MHz HCS08 CPU (central processor unit)
- HC08 instruction set with added BGND instruction
- Background debugging system
- Breakpoint capability to allow single breakpoint setting during in-circuit debugging (plus two more breakpoints in on-chip debug module)
- Debug module containing two comparators and nine trigger modes. Eight deep FIFO for storing change-of-flow addresses and event-only data. Debug module supports both tag and force breakpoints
- Support for up to 32 interrupt/reset sources

## Memory Options

---

- FLASH read/program/erase over full operating voltage and temperature
- MC9S08QG8 — 8 Kbytes FLASH, 512 bytes RAM
- MC9S08QG4 — 4 Kbytes FLASH, 256 bytes RAM

## Power-Saving Modes

---

- Wait plus three stops

## Clock Source Options

---

- **ICS** — Internal clock source module containing a frequency-locked-loop (FLL) controlled by internal or external reference; precision trimming of internal reference allows 0.2% resolution and 2% deviation over temperature and voltage; supports bus frequencies from 1 MHz to 10 MHz
- **XOSC** — Low-power oscillator module with software selectable crystal or ceramic resonator range, 31.25 kHz to 38.4 kHz or 1 MHz to 16 MHz, and supports external clock source input up to 20 MHz

## System Protection

---

- Watchdog computer operating properly (COP) reset with option to run from dedicated 1-kHz internal clock source or bus clock
- Low-voltage detection with reset or interrupt
- Illegal opcode detection with reset
- Illegal address detection with reset
- FLASH block protect

## Peripherals

---

- **ADC** — 8-channel, 10-bit analog-to-digital converter with automatic compare function, asynchronous clock source, temperature sensor, and internal bandgap reference channel; ADC is hardware triggerable using the RTI counter
- **ACMP** — Analog comparator module with option to compare to internal reference; output can be optionally routed to TPM module
- **SCI** — Serial communications interface module with option for 13-bit break capabilities
- **SPI** — Serial peripheral interface module
- **IIC** — Inter-integrated circuit bus module
- **TPM** — 2-channel timer/pulse-width modulator; each channel can be used for input capture, output compare, buffered edge-aligned PWM, or buffered center-aligned PWM
- **MTIM** — 8-bit modulo timer module with 8-bit prescaler
- **KBI** — 8-pin keyboard interrupt module with software selectable polarity on edge or edge/level modes

## Input/Output

---

- 12 general-purpose input/output (I/O) pins, one input-only pin and one output-only pin; outputs 10 mA each, 60 mA max for package
- Software selectable pullups on ports when used as input
- Software selectable slew rate control and drive strength on ports when used as output
- Internal pullup on  $\overline{\text{RESET}}$  and  $\overline{\text{IRQ}}$  pins to reduce customer system cost

## Development Support

---

- Single-wire background debug interface
- On-chip, in-circuit emulation (ICE) with real-time bus capture

## Package Options

---

- 24-pin quad flat no lead (QFN) package
- 16-pin plastic dual in-line package (PDIP) — MC9S08QG8 only
- 16-pin quad flat no lead (QFN) package
- 16-pin thin shrink small outline package (TSSOP)
- 8-pin dual flat no lead (DFN) package
- 8-pin PDIP — MC9S08QG4 only
- 8-pin narrow body small outline integrated circuit (SOIC) package

# List of Chapters

Chapter	Title	Page
Chapter 1	Device Overview .....	19
Chapter 2	External Signal Description .....	23
Chapter 3	Modes of Operation .....	33
Chapter 4	Memory Map and Register Definition .....	39
Chapter 5	Resets, Interrupts, and General System Control.....	59
Chapter 6	Parallel Input/Output Control.....	77
Chapter 7	Central Processor Unit (S08CPUV2) .....	87
Chapter 8	Analog Comparator (S08ACMPV2) .....	107
Chapter 9	Analog-to-Digital Converter (S08ADC10V1).....	115
Chapter 10	Internal Clock Source (S08ICSV1).....	143
Chapter 11	Inter-Integrated Circuit (S08IICV1) .....	155
Chapter 12	Keyboard Interrupt (S08KBIV2).....	173
Chapter 13	Modulo Timer (S08MTIMV1).....	181
Chapter 14	Serial Communications Interface (S08SCIV3).....	191
Chapter 15	Serial Peripheral Interface (S08SPIV3) .....	211
Chapter 16	Timer/Pulse-Width Modulator (S08TPMV2) .....	227
Chapter 17	Development Support .....	243
Appendix A	Electrical Characteristics .....	265
Appendix B	Ordering Information and Mechanical Drawings.....	289

Section Number	Title	Page
8.1.4	Modes of Operation .....	109
8.1.5	Block Diagram .....	109
8.2	External Signal Description .....	111
8.3	Register Definition .....	111
8.3.1	ACMP Status and Control Register (ACMPSC) .....	112
8.4	Functional Description .....	113

## Chapter 9

### Analog-to-Digital Converter (S08ADC10V1)

9.1	Introduction .....	115
9.1.1	Module Configurations .....	117
9.1.2	Features .....	119
9.1.3	Block Diagram .....	119
9.2	External Signal Description .....	120
9.2.1	Analog Power ( $V_{DDAD}$ ) .....	121
9.2.2	Analog Ground ( $V_{SSAD}$ ) .....	121
9.2.3	Voltage Reference High ( $V_{REFH}$ ) .....	121
9.2.4	Voltage Reference Low ( $V_{REFL}$ ) .....	121
9.2.5	Analog Channel Inputs (ADx) .....	121
9.3	Register Definition .....	121
9.3.1	Status and Control Register 1 (ADCSC1) .....	121
9.3.2	Status and Control Register 2 (ADCSC2) .....	123
9.3.3	Data Result High Register (ADCRH) .....	124
9.3.4	Data Result Low Register (ADCRL) .....	124
9.3.5	Compare Value High Register (ADCCVH) .....	125
9.3.6	Compare Value Low Register (ADCCVL) .....	125
9.3.7	Configuration Register (ADCCFG) .....	125
9.3.8	Pin Control 1 Register (APCTL1) .....	127
9.3.9	Pin Control 2 Register (APCTL2) .....	128
9.3.10	Pin Control 3 Register (APCTL3) .....	129
9.4	Functional Description .....	130
9.4.1	Clock Select and Divide Control .....	130
9.4.2	Input Select and Pin Control .....	131
9.4.3	Hardware Trigger .....	131
9.4.4	Conversion Control .....	131
9.4.5	Automatic Compare Function .....	134
9.4.6	MCU Wait Mode Operation .....	134
9.4.7	MCU Stop3 Mode Operation .....	134
9.4.8	MCU Stop1 and Stop2 Mode Operation .....	135
9.5	Initialization Information .....	135
9.5.1	ADC Module Initialization Example .....	135
9.6	Application Information .....	137

## 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 counter periodically. If the application program gets lost and fails to reset the COP counter before it times out, a system reset is generated to force the system back to a known starting point.

After any reset, the COPE becomes set in SOPT1 enabling the COP watchdog (see [Section 5.8.4, “System Options Register 1 \(SOPT1\)”](#) for additional information). If the COP watchdog is not used in an application, it can be disabled by clearing COPE. The COP counter 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 counter.

The COPCLKS bit in SOPT2 (see [Section 5.8.5, “System Options Register 2 \(SOPT2\)”](#) for additional information) selects the clock source used for the COP timer. The clock source options are either the bus clock or an internal 1-kHz clock source. With each clock source, there is an associated short and long time-out controlled by COPT in SOPT1. [Table 5-1](#) summarizes the control functions of the COPCLKS and COPT bits. The COP watchdog defaults to operation from the 1-kHz clock source and the associated long time-out ( $2^8$  cycles).

**Table 5-1. COP Configuration Options**

Control Bits		Clock Source	COP Overflow Count
COPCLKS	COPT		
0	0	~1 kHz	$2^5$ cycles (32 ms) <sup>1</sup>
0	1	~1 kHz	$2^8$ cycles (256 ms) <sup>1</sup>
1	0	Bus	$2^{13}$ cycles
1	1	Bus	$2^{18}$ cycles

<sup>1</sup> Values are shown in this column based on  $t_{RTI} = 1$  ms. See  $t_{RTI}$  in the appendix [Section A.8.1, “Control Timing,”](#) for the tolerance of this value.

Even if the application will use the reset default settings of COPE, COPCLKS, and COPT, the user must write to the write-once SOPT1 and SOPT2 registers during reset initialization to lock in the settings. That way, they cannot be changed accidentally if the application program gets lost. The initial writes to SOPT1 and SOPT2 will reset the COP counter.

The write to SRS that services (clears) the COP counter must 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.


In background debug mode, the COP counter will not increment.

When the bus clock source is selected, the COP counter does not increment while the system is in stop mode. The COP counter resumes as soon as the MCU exits stop mode.

## 5.8.4 System Options Register 1 (SOPT1)

This high page register is a write-once register so only the first write after reset is honored. It can be read at any time. Any subsequent attempt to write to SOPT1 (intentionally or unintentionally) is ignored to avoid accidental changes to these sensitive settings. SOPT1 must be written during the user reset initialization program to set the desired controls even if the desired settings are the same as the reset settings.

	7	6	5	4 <sup>1</sup>	3	2	1	0
R	COPE	COPT	STOPE		0	0	BKGDPE	RSTPE
W								
Reset:	1	1	0	1	0	0	1	u <sup>(2)</sup>
POR:	1	1	0	1	0	0	1	0
LVD:	1	1	0	1	0	0	1	0

 = Unimplemented or Reserved

**Figure 5-5. System Options Register 1 (SOPT1)**

<sup>1</sup> Bit 4 is reserved; writes will change the value but will have no effect on this MCU.

<sup>2</sup> u = unaffected

**Table 5-6. SOPT1 Register Field Descriptions**

Field	Description
7 COPE	<b>COP Watchdog Enable</b> — This write-once bit selects whether the COP watchdog is enabled. 0 COP watchdog timer disabled. 1 COP watchdog timer enabled (force reset on timeout).
6 COPT	<b>COP Watchdog Timeout</b> — This write-once bit selects the timeout period of the COP. COPT along with COPCLKS in SOPT2 defines the COP timeout period. 0 Short timeout period selected. 1 Long timeout period selected.
5 STOPE	<b>Stop Mode Enable</b> — This write-once bit is used to enable stop mode. If stop mode is disabled and a user program attempts to execute a STOP instruction, an illegal opcode reset is forced. 0 Stop mode disabled. 1 Stop mode enabled.
1 BKGDPE	<b>Background Debug Mode Pin Enable</b> — This write-once bit when set enables the PTA4/ACMPO/BKGD/MS pin to function as BKGD/MS. When clear, the pin functions as one of its output only alternative functions. This pin defaults to the BKGD/MS function following any MCU reset. 0 PTA4/ACMPO/BKGD/MS pin functions as PTA4 or ACMPO. 1 PTA4/ACMPO/BKGD/MS pin functions as BKGD/MS.
0 RSTPE	<b>RESET Pin Enable</b> — This write-once bit when set enables the PTA5/ $\overline{\text{IRQ}}$ /TCLK/ $\overline{\text{RESET}}$ pin to function as $\overline{\text{RESET}}$ . When clear, the pin functions as one of its input only alternative functions. This pin defaults to its input-only port function following an MCU POR. When RSTPE is set, an internal pullup device is enabled on $\overline{\text{RESET}}$ . 0 PTA5/ $\overline{\text{IRQ}}$ /TCLK/ $\overline{\text{RESET}}$ pin functions as PTA5, $\overline{\text{IRQ}}$ , or TCLK. 1 PTA5/ $\overline{\text{IRQ}}$ /TCLK/ $\overline{\text{RESET}}$ pin functions as $\overline{\text{RESET}}$ .

### 6.4.2.1 Port A Internal Pullup Enable (PTAPE)

An internal pullup device can be enabled for each port pin by setting the corresponding bit in the pullup enable register (PTAPEn). The pullup device is disabled if the pin is configured as an output by the parallel I/O control logic or any shared peripheral function regardless of the state of the corresponding pullup enable register bit. The pullup device is also disabled if the pin is controlled by an analog function.

	7	6	5	4	3	2	1	0
R	0	0	PTAPE5	PTAPE4 <sup>1</sup>	PTAPE3	PTAPE2	PTAPE1	PTAPE0
W								
Reset:	0	0	0	0	0	0	0	0

**Figure 6-4. Internal Pullup Enable for Port A Register (PTAPE)**

<sup>1</sup> PTAPE4 has no effect on the output-only PTA4 pin.

**Table 6-3. PTAPE Register Field Descriptions**

Field	Description
5:0 PTAPE[5:0]	<b>Internal Pullup Enable for Port A Bits</b> — Each of these control bits determines if the internal pullup device is enabled for the associated PTA pin. For port A pins that are configured as outputs, these bits have no effect and the internal pullup devices are disabled. 0 Internal pullup device disabled for port A bit n. 1 Internal pullup device enabled for port A bit n.

### 6.4.2.2 Port A Slew Rate Enable (PTASE)

Slew rate control can be enabled for each port pin by setting the corresponding bit in the slew rate control register (PTASEn). When enabled, slew control limits the rate at which an output can transition to reduce EMC emissions. Slew rate control has no effect on pins which are configured as inputs.

	7	6	5	4	3	2	1	0
R	0	0	PTASE5 <sup>1</sup>	PTASE4	PTASE3	PTASE2	PTASE1	PTASE0
W								
Reset:	0	0	1	1	1	1	1	1

**Figure 6-6. Slew Rate Enable for Port A Register (PTASE)**

<sup>1</sup> PTASE5 has no effect on the input-only PTA5 pin.

**Table 6-4. PTASE Register Field Descriptions**

Field	Description
5:0 PTASE[5:0]	<b>Output Slew Rate Enable for Port A Bits</b> — Each of these control bits determines if the output slew rate control is enabled for the associated PTA pin. For port A pins that are configured as inputs, these bits have no effect. 0 Output slew rate control disabled for port A bit n. 1 Output slew rate control enabled for port A bit n.

## 7.2 Programmer's Model and CPU Registers

Figure 7-1 shows the five CPU registers. CPU registers are not part of the memory map.

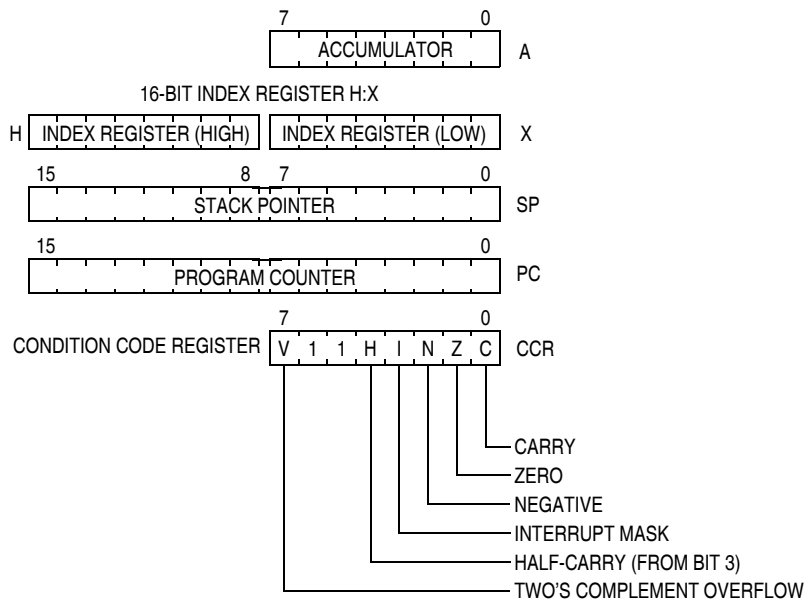


Figure 7-1. CPU Registers

### 7.2.1 Accumulator (A)

The A accumulator is a general-purpose 8-bit register. One operand input to the arithmetic logic unit (ALU) is connected to the accumulator and the ALU results are often stored into the A accumulator after arithmetic and logical operations. The accumulator can be loaded from memory using various addressing modes to specify the address where the loaded data comes from, or the contents of A can be stored to memory using various addressing modes to specify the address where data from A will be stored.

Reset has no effect on the contents of the A accumulator.

### 7.2.2 Index Register (H:X)

This 16-bit register is actually two separate 8-bit registers (H and X), which often work together as a 16-bit address pointer where H holds the upper byte of an address and X holds the lower byte of the address. All indexed addressing mode instructions use the full 16-bit value in H:X as an index reference pointer; however, for compatibility with the earlier M68HC05 Family, some instructions operate only on the low-order 8-bit half (X).

Many instructions treat X as a second general-purpose 8-bit register that can be used to hold 8-bit data values. X can be cleared, incremented, decremented, complemented, negated, shifted, or rotated. Transfer instructions allow data to be transferred from A or transferred to A where arithmetic and logical operations can then be performed.

For compatibility with the earlier M68HC05 Family, H is forced to 0x00 during reset. Reset has no effect on the contents of X.



### 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. . Instruction Set Summary (Sheet 3 of 9)

Source Form	Operation	Address Mode	Object Code	Cycles	Cyc-by-Cyc Details	Affect on CCR				
						VH	I	N	Z	C
BRA <i>rel</i>	Branch Always (if I = 1)	REL	20 rr	3	ppp	--	--	--	--	--
BRCLR <i>n,opr8a,rel</i>	Branch if Bit <i>n</i> in Memory Clear (if (Mn) = 0)	DIR (b0)	01 dd rr	5	rpppp	--	--	--	--	↑
		DIR (b1)	03 dd rr	5	rpppp					
		DIR (b2)	05 dd rr	5	rpppp					
		DIR (b3)	07 dd rr	5	rpppp					
		DIR (b4)	09 dd rr	5	rpppp					
		DIR (b5)	0B dd rr	5	rpppp					
		DIR (b6)	0D dd rr	5	rpppp					
		DIR (b7)	0F dd rr	5	rpppp					
BRN <i>rel</i>	Branch Never (if I = 0)	REL	21 rr	3	ppp	--	--	--	--	--
BRSET <i>n,opr8a,rel</i>	Branch if Bit <i>n</i> in Memory Set (if (Mn) = 1)	DIR (b0)	00 dd rr	5	rpppp	--	--	--	--	↑
		DIR (b1)	02 dd rr	5	rpppp					
		DIR (b2)	04 dd rr	5	rpppp					
		DIR (b3)	06 dd rr	5	rpppp					
		DIR (b4)	08 dd rr	5	rpppp					
		DIR (b5)	0A dd rr	5	rpppp					
		DIR (b6)	0C dd rr	5	rpppp					
		DIR (b7)	0E dd rr	5	rpppp					
BSET <i>n,opr8a</i>	Set Bit <i>n</i> in Memory (Mn ← 1)	DIR (b0)	10 dd	5	rfwpp	--	--	--	--	--
		DIR (b1)	12 dd	5	rfwpp					
		DIR (b2)	14 dd	5	rfwpp					
		DIR (b3)	16 dd	5	rfwpp					
		DIR (b4)	18 dd	5	rfwpp					
		DIR (b5)	1A dd	5	rfwpp					
		DIR (b6)	1C dd	5	rfwpp					
		DIR (b7)	1E dd	5	rfwpp					
BSR <i>rel</i>	Branch to Subroutine PC ← (PC) + \$0002 push (PCL); SP ← (SP) – \$0001 push (PCH); SP ← (SP) – \$0001 PC ← (PC) + <i>rel</i>	REL	AD rr	5	ssppp	--	--	--	--	--
CBEQ <i>opr8a,rel</i>	Compare and... Branch if (A) = (M)	DIR	31 dd rr	5	rpppp	--	--	--	--	--
CBEQA # <i>opr8i,rel</i>	Branch if (A) = (M)	IMM	41 ii rr	4	pppp					
CBEQX # <i>opr8i,rel</i>	Branch if (X) = (M)	IMM	51 ii rr	4	pppp					
CBEQ <i>opr8,X+,rel</i>	Branch if (A) = (M)	IX1+	61 ff rr	5	rpppp					
CBEQ <i>,X+,rel</i>	Branch if (A) = (M)	IX+	71 rr	5	rfppp					
CBEQ <i>opr8,SP,rel</i>	Branch if (A) = (M)	SP1	9E 61 ff rr	6	prpppp					
CLC	Clear Carry Bit (C ← 0)	INH	98	1	p	--	--	--	--	0
CLI	Clear Interrupt Mask Bit (I ← 0)	INH	9A	1	p	--	0	--	--	--
CLR <i>opr8a</i>	Clear M ← \$00	DIR	3F dd	5	rfwpp	0	--	0	1	--
CLRA	A ← \$00	INH	4F	1	p					
CLR X	X ← \$00	INH	5F	1	p					
CLRH	H ← \$00	INH	8C	1	p					
CLR <i>opr8,X</i>	M ← \$00	IX1	6F ff	5	rfwpp					
CLR <i>,X</i>	M ← \$00	IX	7F	4	rfwp					
CLR <i>opr8,SP</i>	M ← \$00	SP1	9E 6F ff	6	prfwpp					

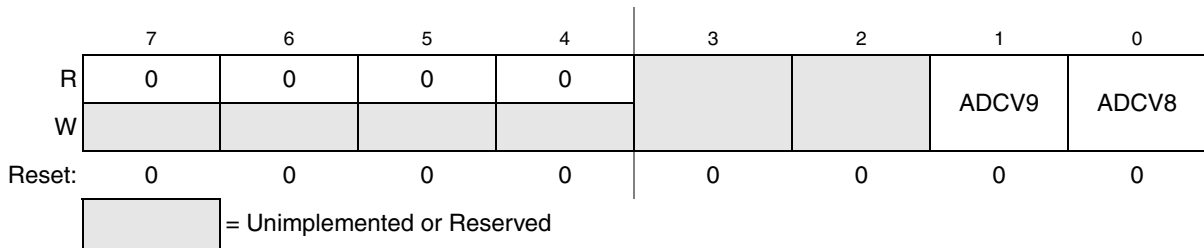
## 8.4 Functional Description

The analog comparator can be used to compare two analog input voltages applied to ACMP+ and ACMP–; or it can be used to compare an analog input voltage applied to ACMP– with an internal bandgap reference voltage. ACBGS is used to select between the bandgap reference voltage or the ACMP+ pin as the input to the non-inverting input of the analog comparator. The comparator output is high when the non-inverting input is greater than the inverting input, and is low when the non-inverting input is less than the inverting input. ACMOD is used to select the condition which will cause ACF to be set. ACF can be set on a rising edge of the comparator output, a falling edge of the comparator output, or either a rising or a falling edge (toggle). The comparator output can be read directly through ACO. The comparator output can be driven onto the ACMPO pin using ACOPE.


**Figure 9-7. Data Result Low Register (ADCRL)**

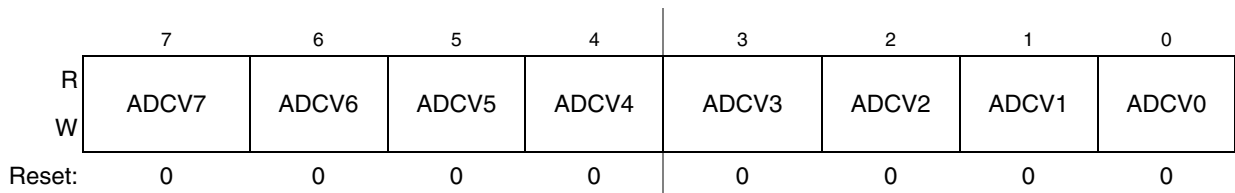
### 9.3.5 Compare Value High Register (ADCCVH)

This register holds the upper two bits of the 10-bit compare value. These bits are compared to the upper two bits of the result following a conversion in 10-bit mode when the compare function is enabled. In 8-bit operation, ADCCVH is not used during compare.


**Figure 9-8. Compare Value High Register (ADCCVH)**

### 9.3.6 Compare Value Low Register (ADCCVL)

This register holds the lower 8 bits of the 10-bit compare value, or all 8 bits of the 8-bit compare value. Bits ADCV7:ADCV0 are compared to the lower 8 bits of the result following a conversion in either 10-bit or 8-bit mode.


**Figure 9-9. Compare Value Low Register(ADCCVL)**

### 9.3.7 Configuration Register (ADCCFG)

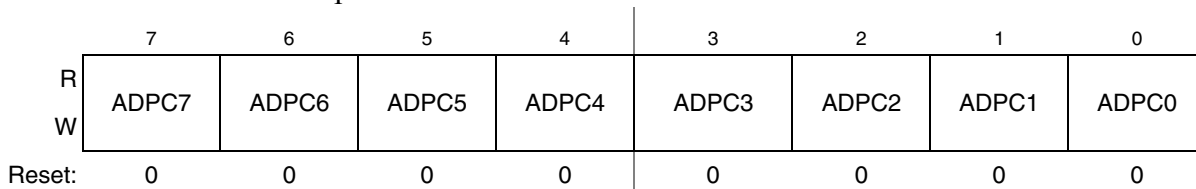
ADCCFG is used to select the mode of operation, clock source, clock divide, and configure for low power or long sample time.

**Table 9-8. Input Clock Select**

ADICLK	Selected Clock Source
00	Bus clock
01	Bus clock divided by 2
10	Alternate clock (ALTCLK)
11	Asynchronous clock (ADACK)

### 9.3.8 Pin Control 1 Register (APCTL1)

The pin control registers are used to disable the I/O port control of MCU pins used as analog inputs. APCTL1 is used to control the pins associated with channels 0–7 of the ADC module.


**Figure 9-11. Pin Control 1 Register (APCTL1)**
**Table 9-9. APCTL1 Register Field Descriptions**

Field	Description
7 ADPC7	<b>ADC Pin Control 7</b> — ADPC7 is used to control the pin associated with channel AD7. 0 AD7 pin I/O control enabled 1 AD7 pin I/O control disabled
6 ADPC6	<b>ADC Pin Control 6</b> — ADPC6 is used to control the pin associated with channel AD6. 0 AD6 pin I/O control enabled 1 AD6 pin I/O control disabled
5 ADPC5	<b>ADC Pin Control 5</b> — ADPC5 is used to control the pin associated with channel AD5. 0 AD5 pin I/O control enabled 1 AD5 pin I/O control disabled
4 ADPC4	<b>ADC Pin Control 4</b> — ADPC4 is used to control the pin associated with channel AD4. 0 AD4 pin I/O control enabled 1 AD4 pin I/O control disabled
3 ADPC3	<b>ADC Pin Control 3</b> — ADPC3 is used to control the pin associated with channel AD3. 0 AD3 pin I/O control enabled 1 AD3 pin I/O control disabled
2 ADPC2	<b>ADC Pin Control 2</b> — ADPC2 is used to control the pin associated with channel AD2. 0 AD2 pin I/O control enabled 1 AD2 pin I/O control disabled

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

### 9.6.1.2 Analog Reference Pins

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

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

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

### 9.6.1.3 Analog Input Pins

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

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

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

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

# Chapter 11

## Inter-Integrated Circuit (S08IICV1)

### 11.1 Introduction

The inter-integrated circuit (IIC) provides a method of communication between a number of devices. The interface is designed to operate up to 100 kbps with maximum bus loading and timing. The device is capable of operating at higher baud rates, up to a maximum of clock/20, with reduced bus loading. The maximum communication length and the number of devices that can be connected are limited by a maximum bus capacitance of 400 pF.

#### 11.1.1 Module Configuration

The IIC module pins, SDA and SCL can be repositioned under software control using IICPS in SOPT2 as as shown in [Table 11-1](#). IICPS in SOPT2 selects which general-purpose I/O ports are associated with IIC operation.

**Table 11-1. IIC Position Options**

IICPS in SOPT2	Port Pin for SDA	Port Pin for SCL
0 (default)	PTA2	PTA3
1	PTB6	PTB7

[Figure 11-1](#) is the MC9S08QG8/4 block diagram with the IIC block highlighted.

### 12.1.1 Features

The KBI features include:

- Up to eight keyboard interrupt pins with individual pin enable bits.
- Each keyboard interrupt pin is programmable as falling edge (or rising edge) only, or both falling edge and low level (or both rising edge and high level) interrupt sensitivity.
- One software enabled keyboard interrupt.
- Exit from low-power modes.

### 12.1.2 Modes of Operation

This section defines the KBI operation in wait, stop, and background debug modes.

#### 12.1.2.1 KBI in Wait Mode

The KBI continues to operate in wait mode if enabled before executing the WAIT instruction. Therefore, an enabled KBI pin ( $KBPEx = 1$ ) can be used to bring the MCU out of wait mode if the KBI interrupt is enabled ( $KBIE = 1$ ).

#### 12.1.2.2 KBI in Stop Modes

The KBI operates asynchronously in stop3 mode if enabled before executing the STOP instruction. Therefore, an enabled KBI pin ( $KBPEx = 1$ ) can be used to bring the MCU out of stop3 mode if the KBI interrupt is enabled ( $KBIE = 1$ ).

During either stop1 or stop2 mode, the KBI is disabled. In some systems, the pins associated with the KBI may be sources of wakeup from stop1 or stop2, see the stop modes section in the [Modes of Operation](#) chapter. Upon wake-up from stop1 or stop2 mode, the KBI module will be in the reset state.

#### 12.1.2.3 KBI in Active Background Mode

When the microcontroller is in active background mode, the KBI will continue to operate normally.

### 12.1.3 Block Diagram

The block diagram for the keyboard interrupt module is shown [Figure 12-2](#).



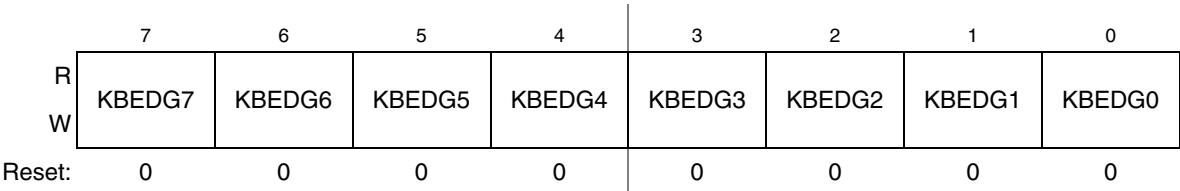


Figure 12-5. KBI Edge Select Register

Table 12-4. KBIES Register Field Descriptions

Field	Description
7:0 KBEDGn	<b>Keyboard Edge Selects</b> — Each of the KBEDGn bits selects the falling edge/low level or rising edge/high level function of the corresponding pin). 0 Falling edge/low level. 1 Rising edge/high level.

## 12.4 Functional Description

This on-chip peripheral module is called a keyboard interrupt (KBI) module because originally it was designed to simplify the connection and use of row-column matrices of keyboard switches. However, these inputs are also useful as extra external interrupt inputs and as an external means of waking the MCU from stop or wait low-power modes.

The KBI module allows up to eight pins to act as additional interrupt sources. Writing to the KBIPEn bits in the keyboard interrupt pin enable register (KBIPE) independently enables or disables each KBI pin. Each KBI pin can be configured as edge sensitive or edge and level sensitive based on the KBMOD bit in the keyboard interrupt status and control register (KBISC). Edge sensitive can be software programmed to be either falling or rising; the level can be either low or high. The polarity of the edge or edge and level sensitivity is selected using the KBEDGn bits in the keyboard interrupt edge select register (KBIES).

### 12.4.1 Edge Only Sensitivity

Synchronous logic is used to detect edges. A falling edge is detected when an enabled keyboard interrupt (KBIPEn=1) input signal is seen as a logic 1 (the deasserted level) during one bus cycle and then a logic 0 (the asserted level) during the next cycle. A rising edge is detected when the input signal is seen as a logic 0 (the deasserted level) during one bus cycle and then a logic 1 (the asserted level) during the next cycle. Before the first edge is detected, all enabled keyboard interrupt input signals must be at the deasserted logic levels. After any edge is detected, all enabled keyboard interrupt input signals must return to the deasserted level before any new edge can be detected.

A valid edge on an enabled KBI pin will set KBF in KBISC. If KBIE in KBISC is set, an interrupt request will be presented to the CPU. Clearing of KBF is accomplished by writing a 1 to KBACK in KBISC.

### 12.4.2 Edge and Level Sensitivity

A valid edge or level on an enabled KBI pin will set KBF in KBISC. If KBIE in KBISC is set, an interrupt request will be presented to the CPU. Clearing of KBF is accomplished by writing a 1 to KBACK in

Figure 14-4 shows the receiver portion of the SCI.

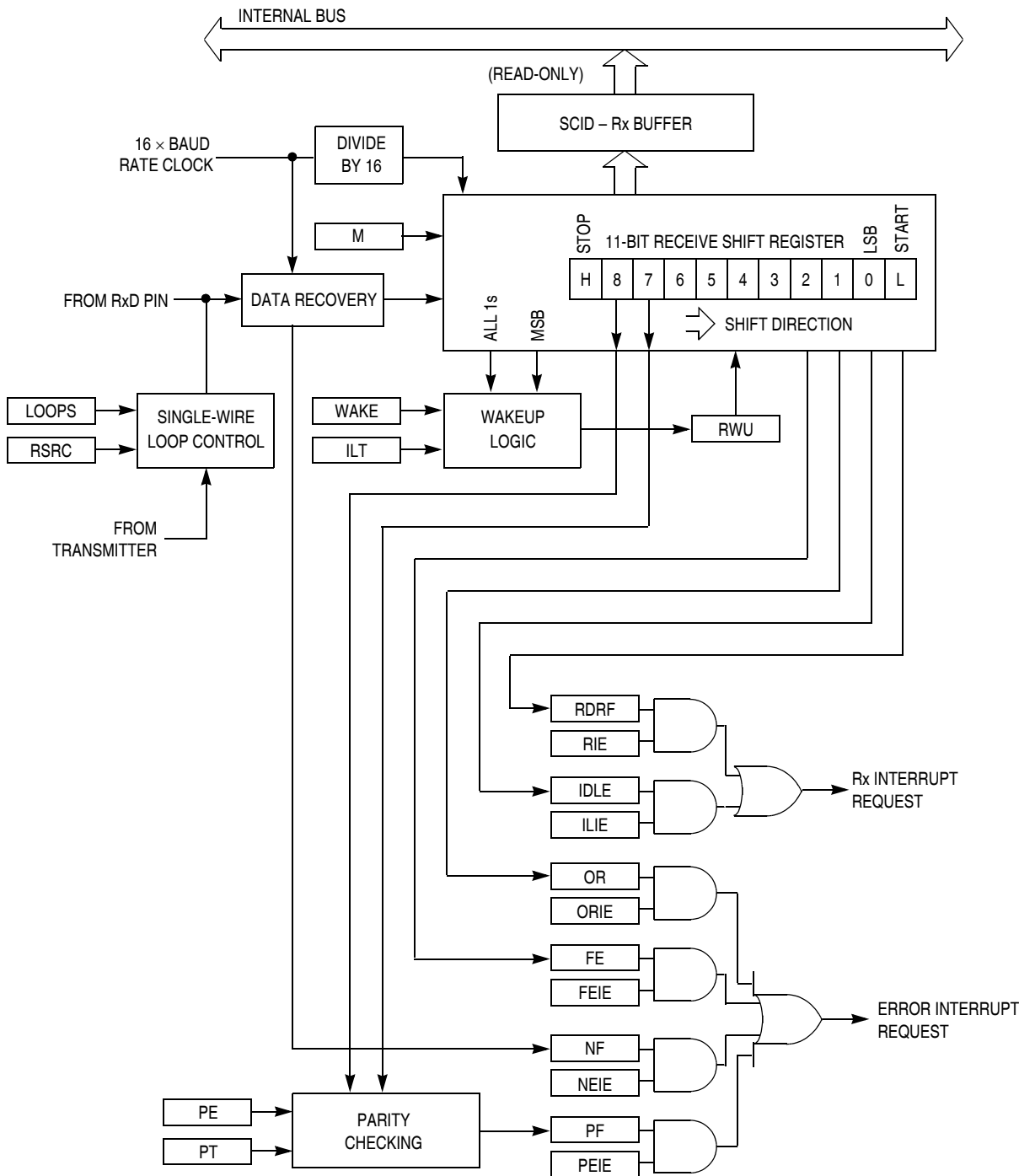


Figure 14-4. SCI Receiver Block Diagram

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

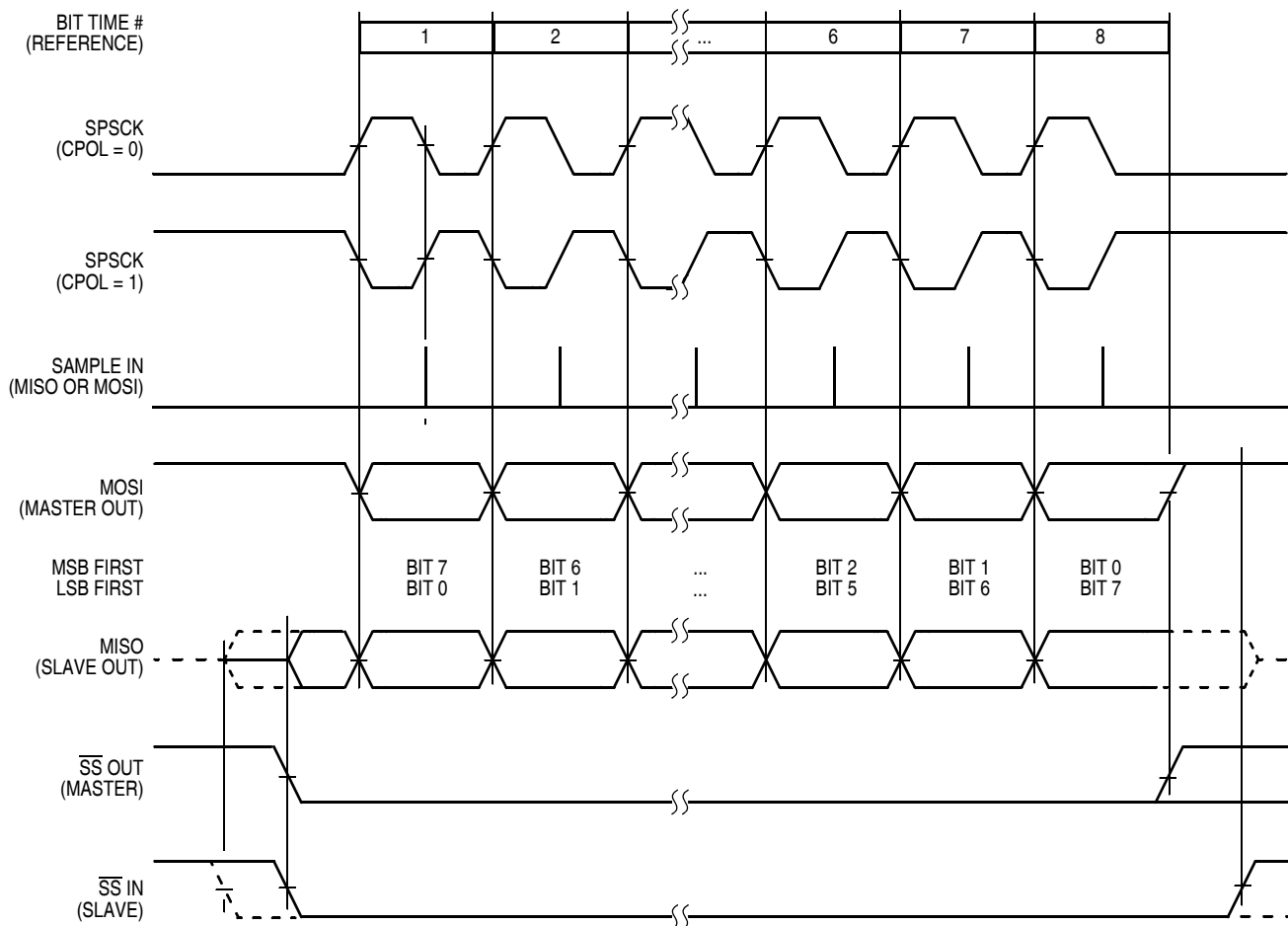


Figure 15-10. SPI Clock Formats (CPHA = 1)

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

Figure 15-11 shows the clock formats when CPHA = 0. At the top of the figure, the eight bit times are shown for reference with bit 1 starting as the slave is selected ( $\overline{SS}$  IN goes low), and bit 8 ends at the last SPSCCK edge. The MSB first and LSB first lines show the order of SPI data bits depending on the setting

All TPM channels are programmable independently as input capture, output compare, or buffered edge-aligned PWM channels.

## 16.2 External Signal Description

When any pin associated with the timer is configured as a timer input, a passive pullup can be enabled. After reset, the TPM modules are disabled and all pins default to general-purpose inputs with the passive pullups disabled.

### 16.2.1 External TPM Clock Sources

When control bits CLKSB:CLKSA in the timer status and control register are set to 1:1, the prescaler and consequently the 16-bit counter for TPM are driven by an external clock source, TPMxCLK, connected to an I/O pin. A synchronizer is needed between the external clock and the rest of the TPM. This synchronizer is clocked by the bus clock so the frequency of the external source must be less than one-half the frequency of the bus rate clock. The upper frequency limit for this external clock source is specified to be one-fourth the bus frequency to conservatively accommodate duty cycle and phase-locked loop (PLL) or frequency-locked loop (FLL) frequency jitter effects.

On some devices the external clock input is shared with one of the TPM channels. When a TPM channel is shared as the external clock input, the associated TPM channel cannot use the pin. (The channel can still be used in output compare mode as a software timer.) Also, if one of the TPM channels is used as the external clock input, the corresponding ELSnB:ELSnA control bits must be set to 0:0 so the channel is not trying to use the same pin.

### 16.2.2 TPMCHn — TPM Channel n I/O Pins

Each TPM channel is associated with an I/O pin on the MCU. The function of this pin depends on the configuration of the channel. In some cases, no pin function is needed so the pin reverts to being controlled by general-purpose I/O controls. When a timer has control of a port pin, the port data and data direction registers do not affect the related pin(s). See the [Pins and Connections](#) chapter for additional information about shared pin functions.

## 16.3 Register Definition

The TPM includes:

- An 8-bit status and control register (TPMSC)
- A 16-bit counter (TPMCNTH:TPMCNTL)
- A 16-bit modulo register (TPMMODH:TPMMODL)

Each timer channel has:

- An 8-bit status and control register (TPMCnSC)
- A 16-bit channel value register (TPMCnVH:TPMCnVL)

Refer to the direct-page register summary in the [Memory](#) chapter of this data sheet for the absolute address assignments for all TPM registers. This section refers to registers and control bits only by their names. A

the host must perform  $((8 - \text{CNT}) - 1)$  dummy reads of the FIFO to advance it to the first significant entry in the FIFO.

In most trigger modes, the information stored in the FIFO consists of 16-bit change-of-flow addresses. In these cases, read DBGFH then DBGFL to get one coherent word of information out of the FIFO. Reading DBGFL (the low-order byte of the FIFO data port) causes the FIFO to shift so the next word of information is available at the FIFO data port. In the event-only trigger modes (see [Section 17.3.5, “Trigger Modes”](#)), 8-bit data information is stored into the FIFO. In these cases, the high-order half of the FIFO (DBGFH) is not used and data is read out of the FIFO by simply reading DBGFL. Each time DBGFL is read, the FIFO is shifted so the next data value is available through the FIFO data port at DBGFL.

In trigger modes where the FIFO is storing change-of-flow addresses, there is a delay between CPU addresses and the input side of the FIFO. Because of this delay, if the trigger event itself is a change-of-flow address or a change-of-flow address appears during the next two bus cycles after a trigger event starts the FIFO, it will not be saved into the FIFO. In the case of an end-trace, if the trigger event is a change-of-flow, it will be saved as the last change-of-flow entry for that debug run.

The FIFO can also be used to generate a profile of executed instruction addresses when the debugger is not armed. When  $\text{ARM} = 0$ , reading DBGFL causes the address of the most-recently fetched opcode to be saved in the FIFO. To use the profiling feature, a host debugger would read addresses out of the FIFO by reading DBGFH then DBGFL at regular periodic intervals. The first eight values would be discarded because they correspond to the eight DBGFL reads needed to initially fill the FIFO. Additional periodic reads of DBGFH and DBGFL return delayed information about executed instructions so the host debugger can develop a profile of executed instruction addresses.

### 17.3.3 Change-of-Flow Information

To minimize the amount of information stored in the FIFO, only information related to instructions that cause a change to the normal sequential execution of instructions is stored. With knowledge of the source and object code program stored in the target system, an external debugger system can reconstruct the path of execution through many instructions from the change-of-flow information stored in the FIFO.

For conditional branch instructions where the branch is taken (branch condition was true), the source address is stored (the address of the conditional branch opcode). Because BRA and BRN instructions are not conditional, these events do not cause change-of-flow information to be stored in the FIFO.

Indirect JMP and JSR instructions use the current contents of the H:X index register pair to determine the destination address, so the debug system stores the run-time destination address for any indirect JMP or JSR. For interrupts, RTI, or RTS, the destination address is stored in the FIFO as change-of-flow information.

### 17.3.4 Tag vs. Force Breakpoints and Triggers

Tagging is a term that refers to identifying an instruction opcode as it is fetched into the instruction queue, but not taking any other action until and unless that instruction is actually executed by the CPU. This distinction is important because any change-of-flow from a jump, branch, subroutine call, or interrupt causes some instructions that have been fetched into the instruction queue to be thrown away without being executed.