



Welcome to [E-XFL.COM](#)

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

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

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

Details

Product Status	Active
Core Processor	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	I ² C, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	12
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	16-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	16-TSSOP
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s08sg32e1ctgr

Contents

Section Number	Title	Page
Chapter 1		
Device Overview		
1.1	Devices in the MC9S08SG32 Series	21
1.2	MCU Block Diagram	22
1.3	System Clock Distribution	24
Chapter 2		
Pins and Connections		
2.1	Device Pin Assignment	25
2.2	Recommended System Connections	27
2.2.1	Power	27
2.2.2	Oscillator (XOSC)	28
2.2.3	$\overline{\text{RESET}}$	28
2.2.4	Background / Mode Select (BKGD/MS)	29
2.2.5	General-Purpose I/O and Peripheral Ports	29
Chapter 3		
Modes of Operation		
3.1	Introduction	33
3.2	Features	33
3.3	Run Mode	33
3.4	Active Background Mode	33
3.5	Wait Mode	34
3.6	Stop Modes	34
3.6.1	Stop3 Mode	35
3.6.2	Stop2 Mode	36
3.6.3	On-Chip Peripheral Modules in Stop Modes	36
Chapter 4		
Memory		
4.1	MC9S08SG32 Series Memory Map	39
4.2	Reset and Interrupt Vector Assignments	40
4.3	Register Addresses and Bit Assignments	41
4.4	RAM	48
4.5	FLASH	48
4.5.1	Features	49
4.5.2	Program and Erase Times	49
4.5.3	Program and Erase Command Execution	50

Section Number	Title	Page
4.5.4	Burst Program Execution	51
4.5.5	Access Errors	53
4.5.6	FLASH Block Protection	53
4.5.7	Vector Redirection	54
4.6	Security	54
4.7	FLASH Registers and Control Bits	56
4.7.1	FLASH Clock Divider Register (FCDIV)	56
4.7.2	FLASH Options Register (FOPT and NVOPT)	57
4.7.3	FLASH Configuration Register (FCNFG)	58
4.7.4	FLASH Protection Register (FPROT and NVPROT)	58
4.7.5	FLASH Status Register (FSTAT)	59
4.7.6	FLASH Command Register (FCMD)	60

Chapter 5

Resets, Interrupts, and General System Control

5.1	Introduction	61
5.2	Features	61
5.3	MCU Reset	61
5.4	Computer Operating Properly (COP) Watchdog	62
5.5	Interrupts	63
5.5.1	Interrupt Stack Frame	64
5.5.2	Interrupt Vectors, Sources, and Local Masks	65
5.6	Low-Voltage Detect (LVD) System	67
5.6.1	Power-On Reset Operation	67
5.6.2	Low-Voltage Detection (LVD) Reset Operation	67
5.6.3	Low-Voltage Warning (LVW) Interrupt Operation	67
5.7	Reset, Interrupt, and System Control Registers and Control Bits	67
5.7.1	System Reset Status Register (SRS)	68
5.7.2	System Background Debug Force Reset Register (SBDFFR)	69
5.7.3	System Options Register 1 (SOPT1)	70
5.7.4	System Options Register 2 (SOPT2)	71
5.7.5	System Device Identification Register (SDIDH, SDIDL)	72
5.7.6	System Power Management Status and Control 1 Register (SPMSC1)	73
5.7.7	System Power Management Status and Control 2 Register (SPMSC2)	74

Chapter 6

Parallel Input/Output Control

6.1	Port Data and Data Direction	77
6.2	Pull-up, Slew Rate, and Drive Strength	78
6.3	Ganged Output	79
6.4	Pin Interrupts	80
6.4.1	Edge-Only Sensitivity	80

Section Number	Title	Page
11.3.1	ICS Control Register 1 (ICSC1)	176
11.3.2	ICS Control Register 2 (ICSC2)	177
11.3.3	ICS Trim Register (ICSTRM).....	178
11.3.4	ICS Status and Control (ICSSC).....	178
11.4	Functional Description	179
11.4.1	Operational Modes.....	179
11.4.2	Mode Switching	181
11.4.3	Bus Frequency Divider	182
11.4.4	Low Power Bit Usage	182
11.4.5	Internal Reference Clock	182
11.4.6	Optional External Reference Clock	182
11.4.7	Fixed Frequency Clock	183

Chapter 12

Modulo Timer (S08MTIMV1)

12.1	Introduction	185
12.1.1	MTIM Configuration Information	185
12.1.2	Features	187
12.1.3	Modes of Operation	187
12.1.4	Block Diagram	188
12.2	External Signal Description	188
12.3	Register Definition	189
12.3.1	MTIM Status and Control Register (MTIMSC)	190
12.3.2	MTIM Clock Configuration Register (MTIMCLK)	191
12.3.3	MTIM Counter Register (MTIMCNT)	192
12.3.4	MTIM Modulo Register (MTIMMOD)	192
12.4	Functional Description	193
12.4.1	MTIM Operation Example	194

Chapter 13

Real-Time Counter (S08RTCV1)

13.1	Introduction	195
13.1.1	Features	197
13.1.2	Modes of Operation	197
13.1.3	Block Diagram	198
13.2	External Signal Description	198
13.3	Register Definition	198
13.3.1	RTC Status and Control Register (RTCSC)	199
13.3.2	RTC Counter Register (RTCCNT)	200
13.3.3	RTC Modulo Register (RTCMOD)	200
13.4	Functional Description	200
13.4.1	RTC Operation Example	201

Chapter 2

Pins and Connections

This section describes signals that connect to package pins. It includes pinout diagrams, recommended system connections, and detailed discussions of signals.

2.1 Device Pin Assignment

The following figures show the pin assignments for the MC9S08SG32 Series devices.

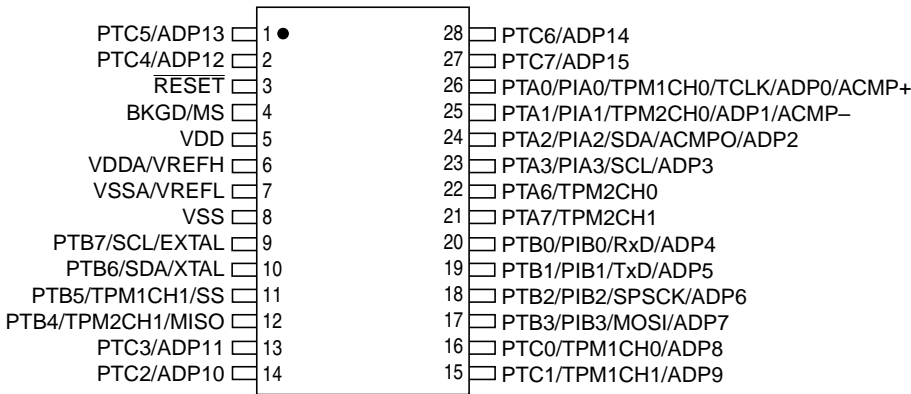


Figure 2-1. 28-Pin TSSOP

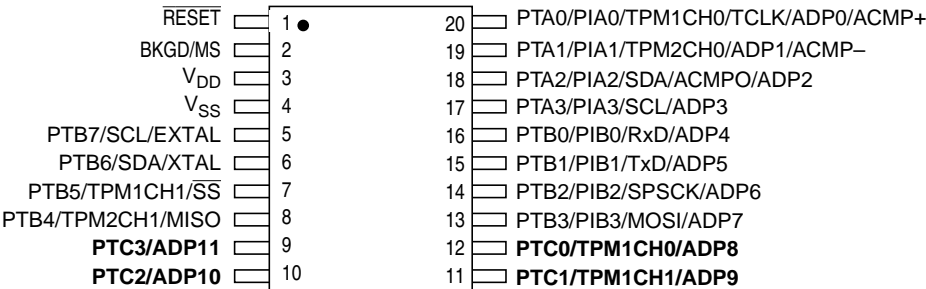


Figure 2-2. 20-Pin TSSOP¹

1. 20-Pin TSSOP package not available for the high-temperature rated devices.



4.2 Reset and Interrupt Vector Assignments

Table 4-1 shows address assignments for reset and interrupt vectors. The vector names shown in this table are the labels used in the Freescale Semiconductor provided equate file for the MC9S08SG32 Series.

Table 4-1. Reset and Interrupt Vectors

Address (High/Low)	Vector	Vector Name
0xFFC0:0xFFC1	Reserved	—
0xFFC2:0xFFC3	ACMP	Vacmp
0xFFC4:0xFFC5	Reserved	—
0xFFC6:0xFFC7	Reserved	—
0xFFC8:0xFFC9	Reserved	—
0xFFCA:0xFFCB	MTIM Overflow	Vmtim
0xFFCC:0xFFCD	RTC	Vrtc
0xFFCE:0xFFCF	IIC	Viic
0xFFD0:0xFFD1	ADC Conversion	Vadc
0xFFD2:0xFFD3	Reserved	—
0xFFD4:0xFFD5	Port B Pin Interrupt	Vportb
0xFFD6:0xFFD7	Port A Pin Interrupt	Vporta
0xFFD8:0xFFD9	Reserved	—
0xFFDA:0xFFDB	SCI Transmit	Vscitx
0xFFDC:0xFFDD	SCI Receive	Vscirx
0xFFDE:0xFFDF	SCI Error	Vsc1err
0xFFE0:0xFFE1	SPI	Vspi
0xFFE2:0xFFE3	TPM2 Overflow	Vtpm2ovf
0xFFE4:0xFFE5	TPM2 Channel 1	Vtpm2ch1
0xFFE6:0xFFE7	TPM2 Channel 0	Vtpm2ch0
0xFFE8:0xFFE9	TPM1 Overflow	Vtpm1ovf
0xFFEA:0xFFEB	Reserved	—
0xFFEC:0xFFED	Reserved	—
0xFFEE:0xFFEF	Reserved	—
0xFFFF0:0xFFFF1	Reserved	—
0xFFFF2:0xFFFF3	TPM1 Channel 1	Vtpm1ch1
0xFFFF4:0xFFFF5	TPM1 Channel 0	Vtpm1ch0
0xFFFF6:0xFFFF7	Reserved	—
0xFFFF8:0xFFFF9	Low Voltage Detect	Vlvd
0xFFFFA:0xFFFFB	Reserved	—
0xFFFFC:0xFFFFD	SWI	Vswi
0xFFFFE:0xFFFFF	Reset	Vreset

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

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

Nonvolatile FLASH registers, shown in [Table 4-4](#), are located in the FLASH memory. These registers include an 8-byte backdoor key, NVBACKKEY, which can be used to gain access to secure memory resources. During reset events, the contents of NVPROT and NVOPT in the nonvolatile register area of the FLASH memory are transferred into corresponding FPROT and FOPT working registers in the high-page registers to control security and block protection options.

Table 4-4. Nonvolatile Register Summary

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
0xFFAE	NVFTRIM	0	0	0	0	0	0	0	FTRIM
0xFFAF	NVTRIM	TRIM							
0xFFB0 – 0xFFB7	NVBACKKEY	8-Byte Comparison Key							
0xFFB8 – 0xFFBC	Reserved	—	—	—	—	—	—	—	—
0xFFBD	NVPROT	FPS							FPDIS
0xFFBE	Reserved	—	—	—	—	—	—	—	—
0xFFBF	NVOPT	KEYEN	FNORED	0	0	0	0	SEC	

Provided the key enable (KEYEN) bit is 1, the 8-byte comparison key can be used to temporarily disengage memory security. This key mechanism can be accessed only through user code running in secure memory. (A security key cannot be entered directly through background debug commands.) This security key can be disabled completely by programming the KEYEN bit to 0. If the security key is disabled, the only way to disengage security is by mass erasing the FLASH if needed (normally through the background debug interface) and verifying that FLASH is blank. To avoid returning to secure mode after the next reset, program the security bits (SEC) to the unsecured state (1:0).

4.5.3 Program and Erase Command Execution

The steps for executing any of the commands are listed below. The FCDIV register must be initialized and any error flags cleared before beginning command execution. The command execution steps are:

1. Write a data value to an address in the FLASH array. The address and data information from this write is latched into the FLASH interface. This write is a required first step in any command sequence. For erase and blank check commands, the value of the data is not important. For page erase commands, the address may be any address in the 512-byte page of FLASH to be erased. For mass erase and blank check commands, the address can be any address in the FLASH memory. Whole pages of 512 bytes are the smallest block of FLASH that may be erased.

NOTE

Do not program any byte in the FLASH more than once after a successful erase operation. Reprogramming bits to a byte that is already programmed is not allowed without first erasing the page in which the byte resides or mass erasing the entire FLASH memory. Programming without first erasing may disturb data stored in the FLASH.

2. Write the command code for the desired command to FCMD. The five valid commands are blank check (0x05), byte program (0x20), burst program (0x25), page erase (0x40), and mass erase (0x41). The command code is latched into the command buffer.
3. Write a 1 to the FCBEF bit in FSTAT to clear FCBEF and launch the command (including its address and data information).

A partial command sequence can be aborted manually by writing a 0 to FCBEF any time after the write to the memory array and before writing the 1 that clears FCBEF and launches the complete command. Aborting a command in this way sets the FACCERR access error flag, which must be cleared before starting a new command.

A strictly monitored procedure must be obeyed or the command will not be accepted. This minimizes the possibility of any unintended changes to the FLASH memory contents. The command complete flag (FCCF) indicates when a command is complete. The command sequence must be completed by clearing FCBEF to launch the command. [Figure 4-2](#) is a flowchart for executing all of the commands except for burst programming. The FCDIV register must be initialized before using any FLASH commands. This must be done only once following a reset.

other than the most experienced programmers because it can lead to subtle program errors that are difficult to debug.

The interrupt service routine ends with a return-from-interrupt (RTI) instruction which restores the CCR, A, X, and PC registers to their pre-interrupt values by reading the previously saved information from the stack.

NOTE

For compatibility with M68HC08 devices, the H register is not automatically saved and restored. It is good programming practice to push H onto the stack at the start of the interrupt service routine (ISR) and restore it immediately before the RTI that is used to return from the ISR.

If more than one interrupt is pending when the I bit is cleared, the highest priority source is serviced first (see Table 5-2).

5.5.1 Interrupt Stack Frame

Figure 5-1 shows the contents and organization of a stack frame. Before the interrupt, the stack pointer (SP) points at the next available byte location on the stack. The current values of CPU registers are stored on the stack starting with the low-order byte of the program counter (PCL) and ending with the CCR. After stacking, the SP points at the next available location on the stack which is the address that is one less than the address where the CCR was saved. The PC value that is stacked is the address of the instruction in the main program that would have executed next if the interrupt had not occurred.

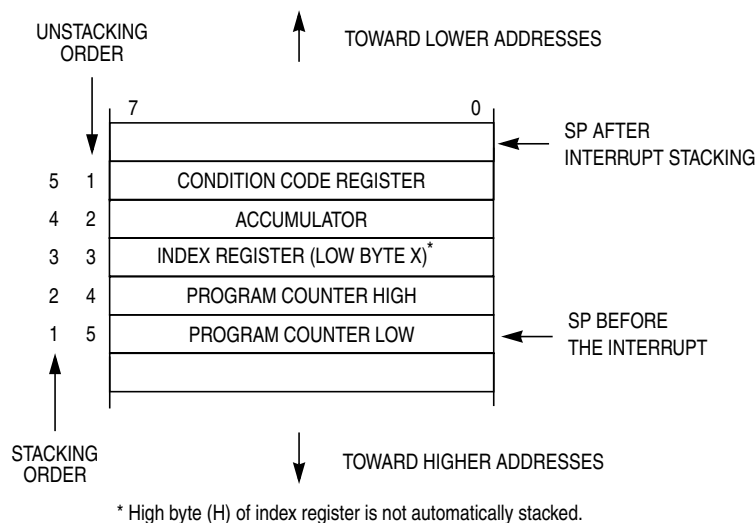


Figure 5-1. Interrupt Stack Frame

When an RTI instruction is executed, these values are recovered from the stack in reverse order. As part of the RTI sequence, the CPU fills the instruction pipeline by reading three bytes of program information, starting from the PC address recovered from the stack.



7.3.6.7 SP-Relative, 16-Bit Offset (SP2)

This variation of indexed addressing uses the 16-bit value in the stack pointer (SP) plus a 16-bit offset included in the instruction as the address of the operand needed to complete the instruction.

7.4 Special Operations

The CPU performs a few special operations that are similar to instructions but do not have opcodes like other CPU instructions. In addition, a few instructions such as STOP and WAIT directly affect other MCU circuitry. This section provides additional information about these operations.

7.4.1 Reset Sequence

Reset can be caused by a power-on-reset (POR) event, internal conditions such as the COP (computer operating properly) watchdog, or by assertion of an external active-low reset pin. When a reset event occurs, the CPU immediately stops whatever it is doing (the MCU does not wait for an instruction boundary before responding to a reset event). For a more detailed discussion about how the MCU recognizes resets and determines the source, refer to the [Resets, Interrupts, and System Configuration](#) chapter.

The reset event is considered concluded when the sequence to determine whether the reset came from an internal source is done and when the reset pin is no longer asserted. At the conclusion of a reset event, the CPU performs a 6-cycle sequence to fetch the reset vector from 0xFFFFE and 0xFFFF and to fill the instruction queue in preparation for execution of the first program instruction.

7.4.2 Interrupt Sequence

When an interrupt is requested, the CPU completes the current instruction before responding to the interrupt. At this point, the program counter is pointing at the start of the next instruction, which is where the CPU should return after servicing the interrupt. The CPU responds to an interrupt by performing the same sequence of operations as for a software interrupt (SWI) instruction, except the address used for the vector fetch is determined by the highest priority interrupt that is pending when the interrupt sequence started.

The CPU sequence for an interrupt is:

1. Store the contents of PCL, PCH, X, A, and CCR on the stack, in that order.
2. Set the I bit in the CCR.
3. Fetch the high-order half of the interrupt vector.
4. Fetch the low-order half of the interrupt vector.
5. Delay for one free bus cycle.
6. Fetch three bytes of program information starting at the address indicated by the interrupt vector to fill the instruction queue in preparation for execution of the first instruction in the interrupt service routine.

After the CCR contents are pushed onto the stack, the I bit in the CCR is set to prevent other interrupts while in the interrupt service routine. Although it is possible to clear the I bit with an instruction in the

Table 9-10. APCTL1 Register Field Descriptions (continued)

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

9.3.9 Pin Control 2 Register (APCTL2)

The pin control registers disable the digital interface to the associated MCU pins used as analog inputs to reduce digital noise and improve conversion accuracy. APCTL2 controls channels 8–15 of the ADC module. This register is not implemented on MCUs that do not have associated external analog inputs. Consult the ADC channel assignment in the module introduction for information on availability of this register.

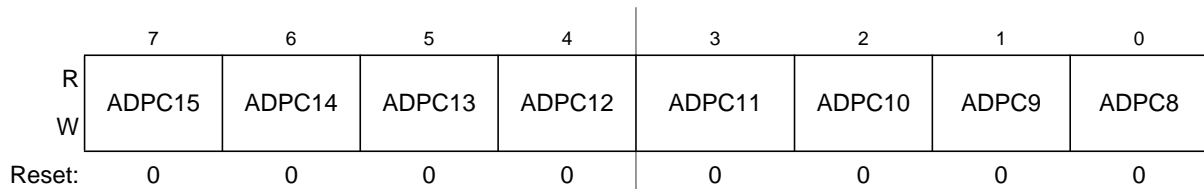
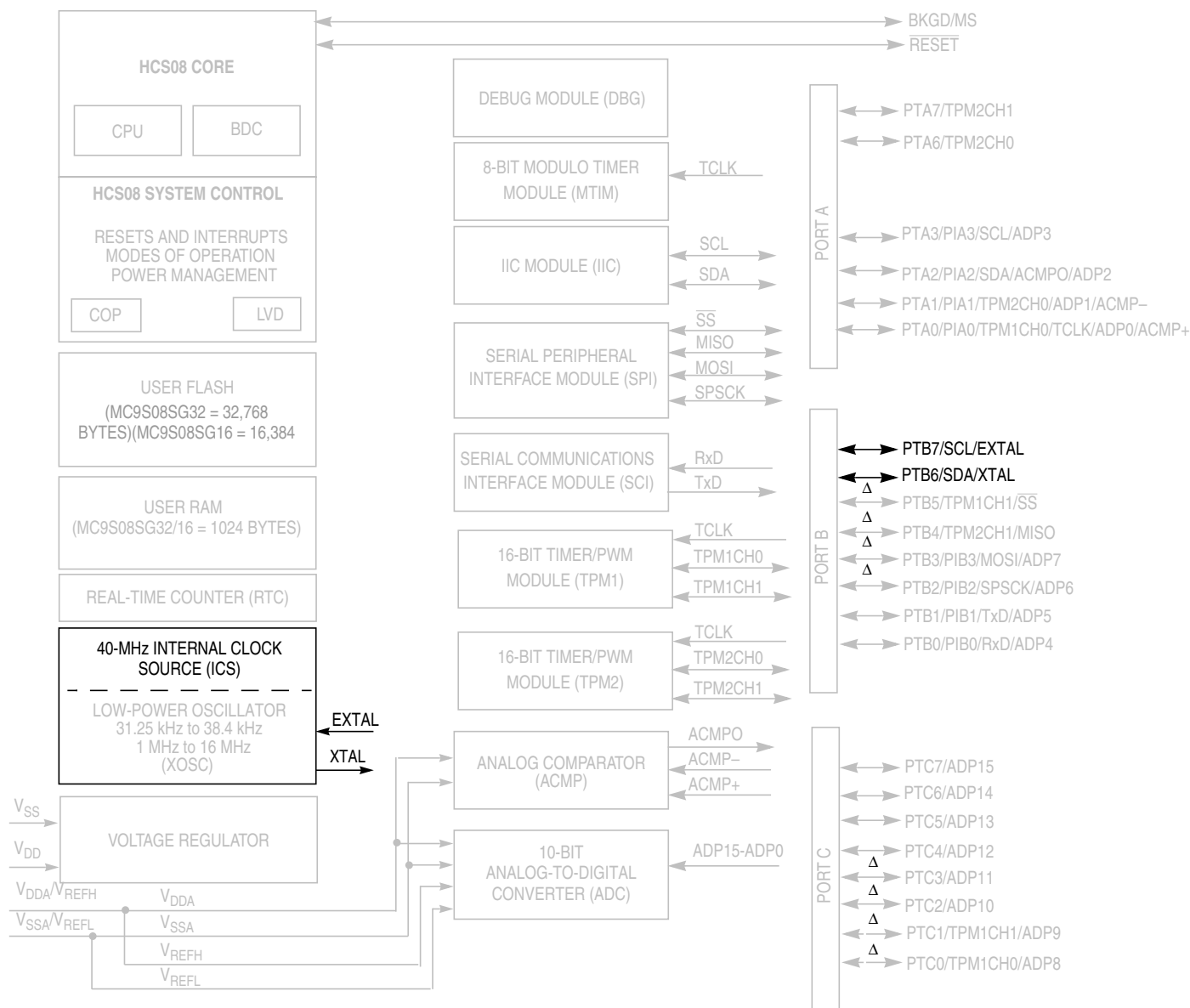


Figure 9-11. Pin Control 2 Register (APCTL2)

Table 9-11. APCTL2 Register Field Descriptions

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



NOTE

- PTC7-PTC0 and PTA7-PTA6 are not available on 16-pin Packages
 - PTC7-PTC4 and PTA7-PTA6 are not available on 20-pin Packages
 - For the 16-pin and 20-pin packages: V_{DDA}/V_{REFH} and V_{SSA}/V_{REFL} are double bonded to V_{DD} and V_{SS} respectively.
- Δ = Pin can be enabled as part of the ganged output drive feature

Figure 11-1. MC9S08SG32 Series Block Diagram Highlighting ICS Block and Pins



15.5.2 SPI Interrupts

There are three flag bits, two interrupt mask bits, and one interrupt vector associated with the SPI system. The SPI interrupt enable mask (SPIE) enables interrupts from the SPI receiver full flag (SPRF) and mode fault flag (MODF). The SPI transmit interrupt enable mask (SPTIE) enables interrupts from the SPI transmit buffer empty flag (SPTEF). When one of the flag bits is set, and the associated interrupt mask bit is set, a hardware interrupt request is sent to the CPU. If the interrupt mask bits are cleared, software can poll the associated flag bits instead of using interrupts. The SPI interrupt service routine (ISR) should check the flag bits to determine what event caused the interrupt. The service routine should also clear the flag bit(s) before returning from the ISR (usually near the beginning of the ISR).

15.5.3 Mode Fault Detection

A mode fault occurs and the mode fault flag (MODF) becomes set when a master SPI device detects an error on the \overline{SS} pin (provided the \overline{SS} pin is configured as the mode fault input signal). The \overline{SS} pin is configured to be the mode fault input signal when $MSTR = 1$, mode fault enable is set ($MODFEN = 1$), and slave select output enable is clear ($SSOE = 0$).

The mode fault detection feature can be used in a system where more than one SPI device might become a master at the same time. The error is detected when a master's \overline{SS} pin is low, indicating that some other SPI device is trying to address this master as if it were a slave. This could indicate a harmful output driver conflict, so the mode fault logic is designed to disable all SPI output drivers when such an error is detected.

When a mode fault is detected, MODF is set and MSTR is cleared to change the SPI configuration back to slave mode. The output drivers on the SPSCCK, MOSI, and MISO (if not bidirectional mode) are disabled.

MODF is cleared by reading it while it is set, then writing to the SPI control register 1 (SPIC1). User software should verify the error condition has been corrected before changing the SPI back to master mode.

Chapter 16

Timer Pulse-Width Modulator (S08TPMV3)

16.1 Introduction

The TPM uses one input/output (I/O) pin per channel, TPMxCHn where x is the TPM number (for example, 1 or 2) and n is the channel number (for example, 0–1). The TPM shares its I/O pins with general-purpose I/O port pins (refer to the [Pins and Connections](#) chapter for more information).

All MC9S08SG32 Series MCUs have two TPM modules.

[Figure 16-1](#) shows the MC9S08SG32 Series block diagram with the TPM modules highlighted.

16.1.1 TPM Configuration Information

The external clock for the MTIM module, TCLK, is selected by setting CLKS = 1:1 or 1:0 in MTIMCLK, which selects the TCLK pin input. The TCLK input can be enabled as external clock inputs to both the MTIM and TPM modules simultaneously.

16.1.2 TPM Pin Repositioning

The TPM modules pins, TPM1CHx and TPM2CHx can be repositioned under software control using TxCHnPS bits in SOPT2 as shown in [Table 16-1](#).

Table 16-1. TPM Position Options

TxCHxPS in SOPT2	Port Pin for TPM2CH1	Port Pin for TPM2CH0	Port Pin for TPM1CH1	Port Pin for TPM1CH0
0 (default)	PTB4	PTA1	PTB5	PTA0
1	PTA7	PTA6	PTC1	PTC0

BDM mode returns the latched value of TPMxCNTH:L from the read buffer instead of the frozen TPM counter value.

- This read coherency mechanism is cleared in TPM v3 in BDM mode if there is a write to TPMxSC, TPMxCNTH or TPMxCNTL. Instead, in these conditions the TPM v2 does not clear this read coherency mechanism.
3. Read of TPMxCnVH:L registers ([Section 16.3.5, “TPM Channel Value Registers \(TPMxCnVH:TPMxCnVL\)”](#))
 - In TPM v3, any read of TPMxCnVH:L registers during BDM mode returns the value of the TPMxCnVH:L register. In TPM v2, if only one byte of the TPMxCnVH:L registers was read before the BDM mode became active, then any read of TPMxCnVH:L registers during BDM mode returns the latched value of TPMxCNTH:L from the read buffer instead of the value in the TPMxCnVH:L registers.
 - This read coherency mechanism is cleared in TPM v3 in BDM mode if there is a write to TPMxCnSC. Instead, in this condition the TPM v2 does not clear this read coherency mechanism.
 4. Write to TPMxCnVH:L registers
 - Input Capture Mode ([Section 16.4.2.1, “Input Capture Mode”](#))

In this mode the TPM v3 does not allow the writes to TPMxCnVH:L registers. Instead, the TPM v2 allows these writes.
 - Output Compare Mode ([Section 16.4.2.2, “Output Compare Mode”](#))

In this mode and if (CLKSB:CLKSA not = 0:0), the TPM v3 updates the TPMxCnVH:L registers with the value of their write buffer at the next change of the TPM counter (end of the prescaler counting) after the second byte is written. Instead, the TPM v2 always updates these registers when their second byte is written.

The following procedure can be used in the TPM v3 to verify if the TPMxCnVH:L registers were updated with the new value that was written to these registers (value in their write buffer).

```

...
write the new value to TPMxCnVH:L;
read TPMxCnVH and TPMxCnVL registers;
while (the read value of TPMxCnVH:L is different from the new value written to
TPMxCnVH:L)
begin
  read again TPMxCnVH and TPMxCnVL;
end
...

```

In this point, the TPMxCnVH:L registers were updated, so the program can continue and, for example, write to TPMxC0SC without cancelling the previous write to TPMxCnVH:L registers.
 - Edge-Aligned PWM ([Section 16.4.2.3, “Edge-Aligned PWM Mode”](#))

In this mode and if (CLKSB:CLKSA not = 00), the TPM v3 updates the TPMxCnVH:L registers with the value of their write buffer after that the both bytes were written and when the

This device contains circuitry protecting against damage due to high static voltage or electrical fields; however, it is advised that normal precautions be taken to avoid application of any voltages higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (for instance, either V_{SS} or V_{DD}) or the programmable pull-up resistor associated with the pin is enabled.

Table A-2. Absolute Maximum Ratings

#	Rating	Symbol	Value	Unit	Temp Rated	
					Standard	AEC Grade 0
1	Supply voltage	V_{DD}	−0.3 to +5.8	V	◆	◆
2	Maximum current into V_{DD}	I_{DD}	120	mA	◆	◆
3	Digital input voltage	V_{In}	−0.3 to $V_{DD} + 0.3$	V	◆	◆
4	Instantaneous maximum current Single pin limit (applies to all port pins) ^{1, 2, 3}	I_D	± 25	mA	◆	◆
5	Storage temperature range	T_{stg}	−55 to 150	°C	◆	◆

¹ Input must be current limited to the value specified. To determine the value of the required current-limiting resistor, calculate resistance values for positive (V_{DD}) and negative (V_{SS}) clamp voltages, then use the larger of the two resistance values.

² All functional non-supply pins except \overline{RESET} are internally clamped to V_{SS} and V_{DD} .

³ Power supply must maintain regulation within operating V_{DD} range during instantaneous and operating maximum current conditions. If positive injection current ($V_{In} > V_{DD}$) is greater than I_{DD} , the injection current may flow out of V_{DD} and could result in external power supply going out of regulation. Ensure external V_{DD} load will shunt current greater than maximum injection current. This will be the greatest risk when the MCU is not consuming power. Examples are: if no system clock is present, or if the clock rate is very low (which would reduce overall power consumption).