NXP USA Inc. - S9S08AW32E5VPUE Datasheet





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 "<u>Embedded -</u> <u>Microcontrollers</u>"

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	54
Program Memory Size	32KB (32K 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 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s08aw32e5vpue

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong





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.



Chapter 3 Modes of Operation

After waking up from stop2, the PPDF bit in SPMSC2 is set. This flag may be used to direct user code to go to a stop2 recovery routine. PPDF remains set and the I/O pin states remain latched until a logic 1 is written to PPDACK in SPMSC2.

To maintain I/O state for pins that were configured as general-purpose I/O, the user must restore the contents of the I/O port registers, which have been saved in RAM, to the port registers before writing to the PPDACK bit. If the port registers are not restored from RAM before writing to PPDACK, then the register bits will assume their reset states when the I/O pin latches are opened and the I/O pins will switch to their reset states.

For pins that were configured as peripheral I/O, the user must reconfigure the peripheral module that interfaces to the pin before writing to the PPDACK bit. If the peripheral module is not enabled before writing to PPDACK, the pins will be controlled by their associated port control registers when the I/O latches are opened.

3.6.2 Stop3 Mode

Stop3 mode is entered by executing a STOP instruction under the conditions as shown in Table 3-1. The states of all of the internal registers and logic, RAM contents, and I/O pin states are maintained.

Stop3 can be exited by asserting $\overline{\text{RESET}}$, or by an interrupt from one of the following sources: the real-time interrupt (RTI), LVD, ADC, IRQ, or the KBI.

If stop3 is exited by means of the RESET pin, then the MCU is reset and operation will resume after taking the reset vector. Exit by means of one of the internal interrupt sources results in the MCU taking the appropriate interrupt vector.

3.6.3 Active BDM Enabled in Stop Mode

Entry into the active background mode from run mode is enabled if the ENBDM bit in BDCSCR is set. This register is described in Chapter 15, "Development Support" of this data sheet. If ENBDM is set when the CPU executes a STOP instruction, the system clocks to the background debug logic remain active when the MCU enters stop mode so background debug communication is still possible. In addition, the voltage regulator does not enter its low-power standby state but maintains full internal regulation. If the user attempts to enter stop2 with ENBDM set, the MCU will instead enter stop3.

Most background commands are not available in stop mode. The memory-access-with-status commands do not allow memory access, but they report an error indicating that the MCU is in either stop or wait mode. The BACKGROUND command can be used to wake the MCU from stop and enter active background mode if the ENBDM bit is set. After entering background debug mode, all background commands are available. Table 3-2 summarizes the behavior of the MCU in stop when entry into the background debug mode is enabled.



4.4.5 Access Errors

An access error occurs whenever the command execution protocol is violated.

Any of the following specific actions will cause the access error flag (FACCERR) in FSTAT to be set. FACCERR must be cleared by writing a 1 to FACCERR in FSTAT before any command can be processed.

- Writing to a FLASH address before the internal FLASH clock frequency has been set by writing to the FCDIV register
- Writing to a FLASH address while FCBEF is not set (A new command cannot be started until the command buffer is empty.)
- Writing a second time to a FLASH address before launching the previous command (There is only one write to FLASH for every command.)
- Writing a second time to FCMD before launching the previous command (There is only one write to FCMD for every command.)
- Writing to any FLASH control register other than FCMD after writing to a FLASH address
- Writing any command code other than the five allowed codes (\$05, \$20, \$25, \$40, or \$41) to FCMD
- Accessing (read or write) any FLASH control register other than the write to FSTAT (to clear FCBEF and launch the command) after writing the command to FCMD.
- The MCU enters stop mode while a program or erase command is in progress (The command is aborted.)
- Writing the byte program, burst program, or page erase command code (\$20, \$25, or \$40) with a background debug command while the MCU is secured (The background debug controller can only do blank check and mass erase commands when the MCU is secure.)
- Writing 0 to FCBEF to cancel a partial command

4.4.6 FLASH Block Protection

The block protection feature prevents the protected region of FLASH from program or erase changes. Block protection is controlled through the FLASH Protection Register (FPROT). When enabled, block protection begins at any 512 byte boundary below the last address of FLASH, \$FFFF. (see Section 4.6.4, "FLASH Protection Register (FPROT and NVPROT)").

After exit from reset, FPROT is loaded with the contents of the NVPROT location which is in the nonvolatile register block of the FLASH memory. FPROT cannot be changed directly from application software so a runaway program cannot alter the block protection settings. Since NVPROT is within the last 512 bytes of FLASH, if any amount of memory is protected, NVPROT is itself protected and cannot be altered (intentionally or unintentionally) by the application software. FPROT can be written through background debug commands which allows a way to erase and reprogram a protected FLASH memory.

The block protection mechanism is illustrated below. The FPS bits are used as the upper bits of the last address of unprotected memory. This address is formed by concatenating FPS7:FPS1 with logic 1 bits as shown. For example, in order to protect the last 8192 bytes of memory (addresses \$E000 through \$FFFF), the FPS bits must be set to 1101 111 which results in the value \$DFFF as the last address of unprotected memory. In addition to programming the FPS bits to the appropriate value, FPDIS (bit 0 of NVPROT) must



the MCU secure. During development, whenever the FLASH is erased, it is good practice to immediately program the SEC00 bit to 0 in NVOPT so SEC01:SEC00 = 1:0. This would allow the MCU to remain unsecured after a subsequent reset.

The on-chip debug module cannot be enabled while the MCU is secure. The separate background debug controller can still be used for background memory access commands, but the MCU cannot enter active background mode except by holding BKGD/MS low at the rising edge of reset.

A user can choose to allow or disallow a security unlocking mechanism through an 8-byte backdoor security key. If the nonvolatile KEYEN bit in NVOPT/FOPT is 0, the backdoor key is disabled and there is no way to disengage security without completely erasing all FLASH locations. If KEYEN is 1, a secure user program can temporarily disengage security by:

- 1. Writing 1 to KEYACC in the FCNFG register. This makes the FLASH module interpret writes to the backdoor comparison key locations (NVBACKKEY through NVBACKKEY+7) as values to be compared against the key rather than as the first step in a FLASH program or erase command.
- 2. Writing the user-entered key values to the NVBACKKEY through NVBACKKEY+7 locations. These writes must be done in order starting with the value for NVBACKKEY and ending with NVBACKKEY+7. STHX should not be used for these writes because these writes cannot be done on adjacent bus cycles. User software normally would get the key codes from outside the MCU system through a communication interface such as a serial I/O.
- 3. Writing 0 to KEYACC in the FCNFG register. If the 8-byte key that was just written matches the key stored in the FLASH locations, SEC01:SEC00 are automatically changed to 1:0 and security will be disengaged until the next reset.

The security key can be written only from secure memory (either RAM or FLASH), so it cannot be entered through background commands without the cooperation of a secure user program.

The backdoor comparison key (NVBACKKEY through NVBACKKEY+7) is located in FLASH memory locations in the nonvolatile register space so users can program these locations exactly as they would program any other FLASH memory location. The nonvolatile registers are in the same 512-byte block of FLASH as the reset and interrupt vectors, so block protecting that space also block protects the backdoor comparison key. Block protects cannot be changed from user application programs, so if the vector space is block protected, the backdoor security key mechanism cannot permanently change the block protect, security settings, or the backdoor key.

Security can always be disengaged through the background debug interface by taking these steps:

- 1. Disable any block protections by writing FPROT. FPROT can be written only with background debug commands, not from application software.
- 2. Mass erase FLASH if necessary.
- 3. Blank check FLASH. Provided FLASH is completely erased, security is disengaged until the next reset.

To avoid returning to secure mode after the next reset, program NVOPT so SEC01:SEC00 = 1:0.



Chapter 6 Parallel Input/Output

6.7.4 Port B Pin Control Registers (PTBPE, PTBSE, PTBDS)

In addition to the I/O control, port B pins are controlled by the registers listed below.



Figure 6-16. Internal Pullup Enable for Port B (PTBPE)

Table 6-9. PTBPE Register Field Descriptions

Field	Description
7:0	Internal Pullup Enable for Port B Bits — Each of these control bits determines if the internal pullup device is
PTBPE[7:0]	enabled for the associated PTB pin. For port B 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 B bit n.
	1 Internal pullup device enabled for port B bit n.

_	7	6	5	4	3	2	1	0
R W	PTBSE7	PTBSE6	PTBSE5	PTBSE4	PTBSE3	PTBSE2	PTBSE1	PTBSE0
Reset	0	0	0	0	0	0	0	0

Figure 6-17. Output Slew Rate Control Enable (PTBSE)

Table 6-10. PTBSE Register Field Descriptions

Field	Description
7:0	Output Slew Rate Control Enable for Port B Bits— Each of these control bits determine whether output slew
PTBSE[7:0]	rate control is enabled for the associated PTB pin. For port B pins that are configured as inputs, these bits have
	no effect.
	0 Output slew rate control disabled for port B bit n.
	1 Output slew rate control enabled for port B bit n.



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

MC9S08AW60 Data Sheet, Rev 2



Chapter 8 Internal Clock Generator (S08ICGV4)

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

MC9S08AW60 Data Sheet, Rev 2



Chapter 8 Internal Clock Generator (S08ICGV4)

entering off mode. If CLKS bits are set to 01 or 11 coming out of the Off state, the ICG enters this mode until ICGDCLK is stable as determined by the DCOS bit. After ICGDCLK is considered stable, the ICG automatically closes the loop by switching to FLL engaged (internal or external) as selected by the CLKS bits.



Figure 8-13. Detailed Frequency-Locked Loop Block Diagram

8.4.3 FLL Engaged, Internal Clock (FEI) Mode

FLL engaged internal (FEI) is entered when any of the following conditions occur:

- CLKS bits are written to 01
- The DCO clock stabilizes (DCOS = 1) while in SCM upon exiting the off state with CLKS = 01

In FLL engaged internal mode, the reference clock is derived from the internal reference clock ICGIRCLK, and the FLL loop will attempt to lock the ICGDCLK frequency to the desired value, as selected by the MFD bits.



Chapter 8 Internal Clock Generator (S08ICGV4)

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

Solving for N / R gives:

N / R = 8.38 MHz /(32 kHz * 64) = 4 ; we can choose N = 4 and R =1	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 (%0000000)

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	LOCRE	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

MC9S08AW60 Data Sheet, Rev 2



Chapter 11 Serial Communications Interface (S08SCIV2)

11.1 Introduction

The MC9S08AW60 Series includes two independent serial communications interface (SCI) modules which are sometimes called universal asynchronous receiver/transmitters (UARTs). Typically, these systems are used to connect to the RS232 serial input/output (I/O) port of a personal computer or workstation, but they can also be used to communicate with other embedded controllers.

A flexible, 13-bit, modulo-based baud rate generator supports a broad range of standard baud rates beyond 115.2 kbaud. Transmit and receive within the same SCI use a common baud rate, and each SCI module has a separate baud rate generator.

This SCI system offers many advanced features not commonly found on other asynchronous serial I/O peripherals on other embedded controllers. The receiver employs an advanced data sampling technique that ensures reliable communication and noise detection. Hardware parity, receiver wakeup, and double buffering on transmit and receive are also included.



Field	Description
1 RWU	 Receiver Wakeup Control — This bit can be written to 1 to place the SCI receiver in a standby state where it waits for automatic hardware detection of a selected wakeup condition. The wakeup condition is either an idle line between messages (WAKE = 0, idle-line wakeup), or a logic 1 in the most significant data bit in a character (WAKE = 1, address-mark wakeup). Application software sets RWU and (normally) a selected hardware condition automatically clears RWU. Refer to Section 11.3.3.2, "Receiver Wakeup Operation" for more details. 0 Normal SCI receiver operation. 1 SCI receiver in standby waiting for wakeup condition.
0 SBK	 Send Break — Writing a 1 and then a 0 to SBK queues a break character in the transmit data stream. Additional break characters of 10 or 11 bit times of logic 0 are queued as long as SBK = 1. Depending on the timing of the set and clear of SBK relative to the information currently being transmitted, a second break character may be queued before software clears SBK. Refer to Section 11.3.2.1, "Send Break and Queued Idle" for more details. 0 Normal transmitter operation. 1 Queue break character(s) to be sent.

11.2.4 SCI Status Register 1 (SCIxS1)

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



Figure 11-8. SCI Status Register 1 (SCIxS1)

Table 11-5. SCI	xS1 Register	Field Descript	ions
-----------------	--------------	----------------	------

Field	Description
7 TDRE	Transmit Data Register Empty Flag — TDRE is set out of reset and when a transmit data value transfers from the transmit data buffer to the transmit shifter, leaving room for a new character in the buffer. To clear TDRE, read SCIxS1 with TDRE = 1 and then write to the SCI data register (SCIxD). 0 Transmit data register (buffer) full. 1 Transmit data register (buffer) empty.
6 TC	 Transmission Complete Flag — TC is set out of reset and when TDRE = 1 and no data, preamble, or break character is being transmitted. 0 Transmitter active (sending data, a preamble, or a break). 1 Transmitter idle (transmission activity complete). TC is cleared automatically by reading SCIxS1 with TC = 1 and then doing one of the following three things: Write to the SCI data register (SCIxD) to transmit new data Queue a preamble by changing TE from 0 to 1 Queue a break character by writing 1 to SBK in SCIxC2



13.4.1.4 STOP Signal

The master can terminate the communication by generating a STOP signal to free the bus. However, the master may generate a START signal followed by a calling command without generating a STOP signal first. This is called repeated START. A STOP signal is defined as a low-to-high transition of SDA while SCL at logical 1 (see Figure 13-8).

The master can generate a STOP even if the slave has generated an acknowledge at which point the slave must release the bus.

13.4.1.5 Repeated START Signal

As shown in Figure 13-8, a repeated START signal is a START signal generated without first generating a STOP signal to terminate the communication. This is used by the master to communicate with another slave or with the same slave in different mode (transmit/receive mode) without releasing the bus.

13.4.1.6 Arbitration Procedure

The IIC bus is a true multi-master bus that allows more than one master to be connected on it. If two or more masters try to control the bus at the same time, a clock synchronization procedure determines the bus clock, for which the low period is equal to the longest clock low period and the high is equal to the shortest one among the masters. The relative priority of the contending masters is determined by a data arbitration procedure, a bus master loses arbitration if it transmits logic 1 while another master transmits logic 0. The losing masters immediately switch over to slave receive mode and stop driving SDA output. In this case, the transition from master to slave mode does not generate a STOP condition. Meanwhile, a status bit is set by hardware to indicate loss of arbitration.

13.4.1.7 Clock Synchronization

Because wire-AND logic is performed on the SCL line, a high-to-low transition on the SCL line affects all the devices connected on the bus. The devices start counting their low period and after a device's clock has gone low, it holds the SCL line low until the clock high state is reached. However, the change of low to high in this device clock may not change the state of the SCL line if another device clock is still within its low period. Therefore, synchronized clock SCL is held low by the device with the longest low period. Devices with shorter low periods enter a high wait state during this time (see Figure 13-9). When all devices concerned have counted off their low period, the synchronized clock SCL line is released and pulled high. There is then no difference between the device clocks and the state of the SCL line and all the devices start counting their high periods. The first device to complete its high period pulls the SCL line low again.



Chapter 14 Analog-to-Digital Converter (S08ADC10V1)

14.1 Overview

The 10-bit analog-to-digital converter (ADC) is a successive approximation ADC designed for operation within an integrated microcontroller system-on-chip. The ADC module design supports up to 28 separate analog inputs (AD0-AD27). Only 18 (AD0-AD15, AD26, and AD27) of the possible inputs are implemented on the MC9S08AW60 Series of MCUs. These inputs are selected by the ADCH bits. Some inputs are shared with I/O pins as shown in Figure 14-1. All of the channel assignments of the ADC for the MC9S08AW60 Series devices are summarized in Table 14-1.

14.2 Channel Assignments

The ADC channel assignments for the MC9S08AW60 Series devices are shown in the table below. Channels that are unimplemented are internally connected to V_{REFL} . Reserved channels convert to an unknown value. Channels which are connected to an I/O pin have an associated pin control bit as shown.

ADCH	Channel	Input	Pin Control	A
00000	AD0	PTB0/ADC1P0	ADPC0	10
00001	AD1	PTB1/ADC1P1	ADPC1	10
00010	AD2	PTB2/ADC1P2	ADPC2	10
00011	AD3	PTB3/ADC1P3	ADPC3	10
00100	AD4	PTB4/ADC1P4	ADPC4	10
00101	AD5	PTB5/ADC1P5	ADPC5	10
00110	AD6	PTB6/ADC1P6	ADPC6	10
00111	AD7	PTB7/ADC1P7	ADPC7	10
01000	AD8	PTD0/ADC1P8	ADPC8	1
01001	AD9	PTD1/ADC1P9	ADPC9	1
01010	AD10	PTD2/ADC1P10/ KBI1P5	ADPC10	1
01011	AD11	PTD3/ADC1P11/ KBI1P6	ADPC11	1
01100	AD12	PTD4/ADC1P12/ TPM2CLK	ADPC12	1
01101	AD13	PTD5/ADC1P13	ADPC13	1

	Table	14-1.	ADC	Channel	Assignment
--	-------	-------	-----	---------	------------

			1
ADCH	Channel	Input	Pin Control
10000	AD16	V _{REFL}	N/A
10001	AD17	V _{REFL}	N/A
10010	AD18	V _{REFL}	N/A
10011	AD19	V _{REFL}	N/A
10100	AD20	V _{REFL}	N/A
10101	AD21	V _{REFL}	N/A
10110	AD22	Reserved	N/A
10111	AD23	Reserved	N/A
11000	AD24	Reserved	N/A
11001	AD25	Reserved	N/A
11010	AD26	Temperature Sensor ¹	N/A
11011	AD27	Internal Bandgap	N/A
11100	—	Reserved	N/A
11101	V _{REFH}	V _{REFH}	N/A

ADCH	Channel	Input	Pin Control	ADCH	Channel	Input	Pin Contro
01110	AD14	PTD6/ADC1P14/ TPM1CLK	ADPC14	11110	V _{REFL}	V _{REFL}	N/A
01111	AD15	PTD7ADC1P15/ KBI1P7	ADPC15	11111	Module disabled	None	N/A

¹ For more information, see Section 14.2.3, "Temperature Sensor."

NOTE

Selecting the internal bandgap channel requires BGBE =1 in SPMSC1 see Section 5.9.8, "System Power Management Status and Control 1 Register (SPMSC1)." For value of bandgap voltage reference see Section Table A-7., "DC Characteristics."

14.2.1 Alternate Clock

The ADC module is capable of performing conversions using the MCU bus clock, the bus clock divided by two, the local asynchronous clock (ADACK) within the module, or the alternate clock, ALTCLK. The alternate clock for the MC9S08AW60 Series MCU devices is the external reference clock (ICGERCLK) from the internal clock generator (ICG) module.

Because ICGERCLK is active only while an external clock source is enabled, the ICG must be configured for either FBE or FEE mode (CLKS1 = 1). ICGERCLK must run at a frequency such that the ADC conversion clock (ADCK) runs at a frequency within its specified range (f_{ADCK}) after being divided down from the ALTCLK input as determined by the ADIV bits. For example, if the ADIV bits are set up to divide by four, then the minimum frequency for ALTCLK (ICGERCLK) is four times the minimum value for f_{ADCK} and the maximum frequency is four times the maximum value for f_{ADCK} . Because of the minimum frequency requirement, when an oscillator circuit is used it must be configured for high range operation (RANGE = 1).

ALTCLK is active while the MCU is in wait mode provided the conditions described above are met. This allows ALTCLK to be used as the conversion clock source for the ADC while the MCU is in wait mode.

ALTCLK cannot be used as the ADC conversion clock source while the MCU is in stop3.

14.2.2 Hardware Trigger

The ADC hardware trigger, ADHWT, is output from the real time interrupt (RTI) counter. The RTI counter can be clocked by either ICGERCLK or a nominal 1 kHz clock source within the RTI block. The 1-kHz clock source can be used with the MCU in run, wait, or stop3. With the ICG configured for either FBE or FEE mode, ICGERCLK can be used with the MCU in run or wait.

The period of the RTI is determined by the input clock frequency and the RTIS bits. When the ADC hardware trigger is enabled, a conversion is initiated upon an RTI counter overflow. The RTI counter is a free running counter that generates an overflow at the RTI rate determined by the RTIS bits.



14.3.1 Analog Power (V_{DDAD})

The ADC analog portion uses V_{DDAD} as its power connection. In some packages, V_{DDAD} is connected internally to V_{DD} . If externally available, connect the V_{DDAD} pin to the same voltage potential as V_{DD} . External filtering may be necessary to ensure clean V_{DDAD} for good results.

14.3.2 Analog Ground (V_{SSAD})

The ADC analog portion uses V_{SSAD} as its ground connection. In some packages, V_{SSAD} is connected internally to V_{SS} . If externally available, connect the V_{SSAD} pin to the same voltage potential as V_{SS} .

14.3.3 Voltage Reference High (V_{REFH})

 V_{REFH} is the high reference voltage for the converter. In some packages, V_{REFH} is connected internally to V_{DDAD} . If externally available, 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}).

14.3.4 Voltage Reference Low (V_{REFL})

 V_{REFL} is the low reference voltage for the converter. In some packages, V_{REFL} is connected internally to V_{SSAD} . If externally available, connect the V_{REFL} pin to the same voltage potential as V_{SSAD} .

14.3.5 Analog Channel Inputs (ADx)

The ADC module supports up to 28 separate analog inputs. An input is selected for conversion through the ADCH channel select bits.

14.4 Register Definition

These memory mapped registers control and monitor operation of the ADC:

- Status and control register, ADC1SC1
- Status and control register, ADC1SC2
- Data result registers, ADC1RH and ADC1RL
- Compare value registers, ADC1CVH and ADC1CVL
- Configuration register, ADC1CFG
- Pin enable registers, APCTL1, APCTL2, APCTL3

14.4.1 Status and Control Register 1 (ADC1SC1)

This section describes the function of the ADC status and control register (ADC1SC1). Writing ADC1SC1 aborts the current conversion and initiates a new conversion (if the ADCH bits are equal to a value other than all 1s).



• Non-intrusive commands can be executed at any time even while the user's program is running. Non-intrusive commands allow a user to read or write MCU memory locations or access status and control registers within the background debug controller.

Typically, a relatively simple interface pod is used to translate commands from a host computer into commands for the custom serial interface to the single-wire background debug system. Depending on the development tool vendor, this interface pod may use a standard RS-232 serial port, a parallel printer port, or some other type of communications such as a universal serial bus (USB) to communicate between the host PC and the pod. The pod typically connects to the target system with ground, the BKGD pin, RESET, and sometimes V_{DD} . An open-drain connection to reset allows the host to force a target system reset, which is useful to regain control of a lost target system or to control startup of a target system before the on-chip nonvolatile memory has been programmed. Sometimes V_{DD} can be used to allow the pod to use power from the target system to avoid the need for a separate power supply. However, if the pod is powered separately, it can be connected to a running target system without forcing a target system reset or otherwise disturbing the running application program.



Figure 15-1. BDM Tool Connector

15.2.1 BKGD Pin Description

BKGD is the single-wire background debug interface pin. The primary function of this pin is for bidirectional serial communication of active background mode commands and data. During reset, this pin is used to select between starting in active background mode or starting the user's application program. This pin is also used to request a timed sync response pulse to allow a host development tool to determine the correct clock frequency for background debug serial communications.

BDC serial communications use a custom serial protocol first introduced on the M68HC12 family of microcontrollers. This protocol assumes the host knows the communication clock rate that is determined by the target BDC clock rate. All communication is initiated and controlled by the host that drives a high-to-low edge to signal the beginning of each bit time. Commands and data are sent most significant bit first (MSB first). For a detailed description of the communications protocol, refer to Section 15.2.2, "Communication Details."

If a host is attempting to communicate with a target MCU that has an unknown BDC clock rate, a SYNC command may be sent to the target MCU to request a timed sync response signal from which the host can determine the correct communication speed.

BKGD is a pseudo-open-drain pin and there is an on-chip pullup so no external pullup resistor is required. Unlike typical open-drain pins, the external RC time constant on this pin, which is influenced by external capacitance, plays almost no role in signal rise time. The custom protocol provides for brief, actively driven speedup pulses to force rapid rise times on this pin without risking harmful drive level conflicts. Refer to Section 15.2.2, "Communication Details," for more detail.



15.3 On-Chip Debug System (DBG)

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

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

Most of the debug module's functions are used during development, and user programs rarely access any of the control and status registers for the debug module. The one exception is that the debug system can provide the means to implement a form of ROM patching. This topic is discussed in greater detail in Section 15.3.6, "Hardware Breakpoints."

15.3.1 Comparators A and B

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

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

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

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

15.3.2 Bus Capture Information and FIFO Operation

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



Chapter 15 Development Support

15.4.3.7 Debug Control Register (DBGC)

This register can be read or written at any time.



Figure 15-7. Debug Control Register (DBGC)

Table 15-4	. DBGC	Register	Field	Descriptions
------------	--------	----------	-------	--------------

Field	Description
7 DBGEN	 Debug Module Enable — Used to enable the debug module. DBGEN cannot be set to 1 if the MCU is secure. 0 DBG disabled 1 DBG enabled
6 ARM	 Arm Control — Controls whether the debugger is comparing and storing information in the FIFO. A write is used to set this bit (and ARMF) and completion of a debug run automatically clears it. Any debug run can be manually stopped by writing 0 to ARM or to DBGEN. 0 Debugger not armed 1 Debugger armed
5 TAG	Tag/Force Select — Controls whether break requests to the CPU will be tag or force type requests. IfBRKEN = 0, this bit has no meaning or effect.0 CPU breaks requested as force type requests1 CPU breaks requested as tag type requests
4 BRKEN	 Break Enable — Controls whether a trigger event will generate a break request to the CPU. Trigger events can cause information to be stored in the FIFO without generating a break request to the CPU. For an end trace, CPU break requests are issued to the CPU when the comparator(s) and R/W meet the trigger requirements. For a begin trace, CPU break requests are issued when the FIFO becomes full. TRGSEL does not affect the timing of CPU break requests. 0 CPU break requests not enabled 1 Triggers cause a break request to the CPU
3 RWA	 R/W Comparison Value for Comparator A — When RWAEN = 1, this bit determines whether a read or a write access qualifies comparator A. When RWAEN = 0, RWA and the R/W signal do not affect comparator A. 0 Comparator A can only match on a write cycle 1 Comparator A can only match on a read cycle
2 RWAEN	 Enable R/W for Comparator A — Controls whether the level of R/W is considered for a comparator A match. 0 R/W is not used in comparison A 1 R/W is used in comparison A
1 RWB	 R/W Comparison Value for Comparator B — When RWBEN = 1, this bit determines whether a read or a write access qualifies comparator B. When RWBEN = 0, RWB and the R/W signal do not affect comparator B. 0 Comparator B can match only on a write cycle 1 Comparator B can match only on a read cycle
0 RWBEN	 Enable R/W for Comparator B — Controls whether the level of R/W is considered for a comparator B match. 0 R/W is not used in comparison B 1 R/W is used in comparison B



Appendix A Electrical Characteristics and Timing Specifications

The conducted susceptibility is determined by injecting the transient susceptibility signal on each pin of the microcontroller. The transient waveform and injection methodology is based on IEC 61000-4-4 (EFT/B). The transient voltage required to cause performance degradation on any pin in the tested configuration is greater than or equal to the reported levels unless otherwise indicated by footnotes below the table.

Parameter	Symbol	Conditions	f _{OSC} /f _{BUS}	Result	Amplitude ¹ (Min)	Unit
				A	±0 ±2.0 ²	
Conducted susceptibility, electrical	V _{CS_EFT}	$V_{DD} = 5.5V$ $T_A = +25^{\circ}C$ package type	32768 Hz crystal 2 MHz Bus	В	±2.5	kV
fast transient/burst (EFT/B)				С	±3.0	
		64 QFP		D	>±3.0	

Table A-18.	Conducted	Susceptibility
-------------	-----------	----------------

¹ Data based on qualification test results. Not tested in production.

² The RESET pin is susceptible to the minimum applied transient of 220 V. All other pins have a result of A up to a minimum of 2000V.

The susceptibility performance classification is described in Table A-19.

Result	Performance Criteria				
A	No failure	The MCU performs as designed during and after exposure.			
В	Self-recovering failure	The MCU does not perform as designed during exposure. The MCU returns automatically to normal operation after exposure is removed.			
С	Soft failure	The MCU does not perform as designed during exposure. The MCU does not return to normal operation until exposure is removed and the RESET pin is asserted.			
D	Hard failure	The MCU does not perform as designed during exposure. The MCU does not return to normal operation until exposure is removed and the power to the MCU is cycled.			
E	Damage	The MCU does not perform as designed during and after exposure. The MCU cannot be returned to proper operation due to physical damage or other permanent performance degradation.			

Table A-19. Susceptibility Performance Classification