Freescale Semiconductor
Application Note

Document Number: AN3331
Rev.0, 02/2007

Migrating from the MC68HC908AZ60A to MC9S08DZ60

by: Gordon Borland
8-Bit Applications Engineering
Microcontroller Division
East Kilbride, Scotland

1 Introduction

Freescale Semiconductor’s MC9S08DZ60 represents a simple, low-cost, high-performance upgrade path from the MC68HC908AZ60A.

This document introduces features of the MC9S08DZ60, which can be used to enhance system design. This application note does not describe in detail how to use new features of the MC9S08DZ60, but highlights the differences between the two devices. Consult the specific MC9S08DZ60 reference manual and data sheet for details.

Contents

1 Introduction 1
2 MC9S08DZ60 Features. 4
3 Pinouts and Package Types 6
4 Low-Power Modes 9
5 Parallel Input/Output Control 10
6 CPU 14
7 Memory 17
8 Peripherals 21
9 Development Support Systems 40
10 Conclusion 45
11 Useful Links 45

NOTE

Excepting mask set errata documents, if any other Freescale document contains information conflicting with the device data sheet, the data sheet should be considered to have the most current and correct data.

The major differences between the MC9S08DZ60 and MC68HC908AZ60A:

- Lower price
- Bus speed
- Improved EMC
- Up to 2K EEPROM
- 4K RAM
- Flexible clock options
- Fast ADC with 10-bit resolution
- Reduced power consumption
- Non-intrusive on-chip debug system

1.1 Why MC9S08DZ60?

The MC9S08DZ60 is an evolutionary step from the MC68HC908AZ60A. The MC9S08DZ60 improves low-voltage/low-power performance without sacrificing CPU performance. MC9S08DZ60 MCUs are created by an advanced wafer-fabrication process that gives MC9S08DZ60 a normal operating voltage of 2.7 V to 5.5 V. As a result, MC9S08DZ60s are capable of higher bus speeds at lower operating voltages. The supply current is a good example of this. Operating at a bus frequency of 8 MHz, with 5 V VDD in the Run Idd for the MC9S08DZ60 is typically 7.5 mA. Under similar conditions (5 V VDD, bus frequency = 8.4 MHz) the MC68HC908AZ60A will typically draw 35 mA by comparison.

1.2 S08D and S08EN families

This document focuses on the differences between the MC9S08DZ60 and the MC68HC908AZ60A, but its content is also relevant to other devices in the S08D and S08EN families. The features and peripherals available on each of the family members varies. Table 1 shows the S08D and S08EN families and the feature set for each device. Please refer to this table to determine which sections of this document are relevant for each device in the S08D and S08EN families. For further details on the peripherals and features available on each S08D or S08EN family member, please consult the specific reference manual or data sheet.

Table 1. S08D and S08EN Families

Device	Flash	RAM	EEPROM	CAN	SCI	SPI	IIC	16-Bit Timer/PWM	10-Bit ADC	Comparator	Clock Type	Packages
9S08DZ60	60k	4k	2K	1	2	1	1	6ch 6ch 4ch	24ch 16ch 10ch	2	MCG	64LQFP, 48LQFP, 32LQFP
9S08DZ48	48k	3k	1.5K	1	2	1	1	6ch 6ch 4ch	24ch 16ch 10ch	2	MCG	64LQFP, 48LQFP, 32LQFP
9S08DZ32	32k	2k	1K	1	2	1	1	6ch 6cht 4ch	24ch 16ch 10ch	2	MCG	64LQFP, 48LQFP, 32LQFP
9S08DZ16	16k	1k	512b	1	2	1	1	6ch 4ch	16ch 10ch	2	MCG	48LQFP, 32LQFP
9S08DV60	60k	3k	—	1	2	1	1	6ch 6ch 4ch	16ch 16ch 10ch	2	MCG	64LQFP, 48LQFP, 32LQFP
9S08DV48	48k	2k	—	1	2	1	1	6ch 6ch 4ch	16ch 16ch 10ch	2	MCG	64LQFP, 48LQFP, 32LQFP
9S08DV32	32k	2k	—	1	2	1	1	6ch 6ch 4ch	16ch 16ch 10ch	2	MCG	64LQFP, 48LQFP, 32LQFP
9S08DV16	16k	1k	—	1	1	1	1	6ch 4ch	16ch 10ch	2	MCG	48LQFP, 32LQFP
9S08DN60	60k	2k	2K	—	1	1	1	6ch 6ch 4ch	16ch 16ch 10ch	2	MCG	64LQFP, 48LQFP, 32LQFP
9S08DN48	48k	2k	1.5K	—	1	1	1	6ch 6ch 4ch	16ch 16ch 10ch	2	MCG	64LQFP, 48LQFP, 32LQFP
9S08DN32	32k	1.5k	1K	—	1	1	1	6ch 6ch 4ch	16ch 16ch 10ch	2	MCG	64LQFP, 48LQFP, 32LQFP
9S08DN16	16k	1k	512b	—	1	1	1	6ch 4ch	16ch 10ch	2	MCG	48LQFP, 32LQFP
9S08EN32	32k	1k	—	—	1	1	—	4ch 4ch	12ch 10ch	1	MCG	48 LQFP, 32LQFP
9S08EN16	16k	512b	—	—	1	1	—	4ch 4ch	12ch 10ch	1	MCG	48 LQFP, 32LQFP

2 MC9S08DZ60 Features

- 8-bit HCS08 central processor unit (CPU)
  - 40 MHz HCS08 CPU (20 MHz bus)
  - HC08 instruction set with added BGND instruction
  - Support for up to 32 interrupt/reset sources
- On-chip memory
  - Flash read/program/erase over full operating voltage and temperature
    - MC9S08DZ60 = 60K
    - MC9S08DZ48 = 48K
    - MC9S08DZ32 = 32K
    - MC9S08DZ16 = 16K
  - Up to 2K EEPROM in-circuit programmable memory
    - 8-byte single-page or 4-byte dual-page erase sector
    - Program and erase while executing flash memory; erase abort
  - Up to 4K RAM
- Power-saving modes
  - Two very low-power stop modes
  - Reduced-power wait mode
  - Very low-power, real-time interrupt for use in run, wait, and stop
  - Internal voltage regulator for lower voltage CPU operation
- Clock source options
  - Oscillator (XOSC)
    - Loop-control pierce oscillator
    - Crystal or ceramic resonator range of 31.25 kHz to 38.4 kHz or 1 MHz to 16 MHz
  - Multi-purpose clock generator (MCG)
    - PLL and FLL modes
    - Internal reference clock with trim adjustment
    - External reference with oscillator/resonator options
- System protection
  - Watchdog computer operating properly (COP) reset with option to run from backup dedicated 1 kHz internal clock source or bus clock
  - Low-voltage detection with reset or interrupt and selectable trip points
  - Illegal opcode detection with reset
  - Illegal address detection with reset
  - Flash-block protect
  - Loss-of-lock protection

- Development support
  - Single-wire background debug interface
  - On-chip, in-circuit emulation (ICE) with real-time bus capture
- Analog-to-digital converter (ADC)
  - 24-channel, 10-bit resolution
  - 2.5 μs conversion time
  - Automatic compare function
  - 1.7 mV/°C temperature sensor
  - internal bandgap reference channel
- Analog comparator (ACMPx)
  - Two analog comparators with selectable interrupt on rising, falling, or either edge of comparator
  - Compare option to fixed internal bandgap reference voltage
- Controller area network (MSCAN)
  - CAN protocol —Version 2.0 A, B; standard and extended data frames
  - Support for remote frames
  - Five receive buffers with FIFO storage scheme
  - Flexible identifier acceptance filters programmable as: 2 × 32-bit, 4 × 16-bit, or 8 × 8-bit
- Serial communications interface (SCIx)
  - Two SCIs supporting LIN 2.0 protocol and SAE J2602 protocols
  - Master extended break generation
  - Slave extended break detection
  - Wakeup on active edge
- Serial peripheral interface (SPI)
  - Full-duplex or single-wire bidirectional
  - Double-buffered transmit and receive
  - Master or slave mode
  - MSB-first or LSB-first shifting
- Inter-integrated circuit (IIC)
  - Up to 100 kbps with maximum bus loading
  - Multi-master operation
  - Programmable slave address
  - General call address
  - Interrupt driven byte-by-byte data transfer
- Timer/pulse-width module (TPMx)
  - One 6-channel (TPM1) and one 2-channel (TPM2)
  - Selectable input capture, output compare, or buffered edge-aligned PWM on each channel

- Real-time counter (RTC)
  - — 8-bit modulus counter with binary- or decimal-based prescaler
  - — Real-time clock capabilities using external crystal and RTC for precise time base, time-of-day, calendar, or task-scheduling functions
  - — Free running on-chip low-power oscillator (1 kHz) for cyclic wake-up without external components

3 Pinouts and Package Types

The MC9S08DZ60 is available in 64-LQFP, 48-LQFP, and 32-LQFP packages. The MC68HC908AZ60A is available only in 64-QFP and 52-PLCC packages. Figure 1 and Figure 2 show the pin layouts for the 64-pin packages.

The MC9S08DZ60 has a different pinout from the MC68HC908AZ60A, making layout changes necessary. Table 2 provides a pinout comparison between the MC9S08DZ60A 64-LQFP package and the MC68HC908AZ60A 64-QFP package.

In addition to the pinout, changes in the pitch and footprints also require layout changes. The MC9S08DZ60 64-LQFP package has a pin pitch of 0.5 mm and an overall footprint of 12x12 mm. By contrast the MC68HC908AZ60A 64-QFP package has a pitch of 0.8 mm and an overall footprint of 17.2 x 17.2 mm.

For more information on the mechanical dimensions of the package types, please refer to the MC9S08DZ60 and MC68HC908AZ60A reference manuals.

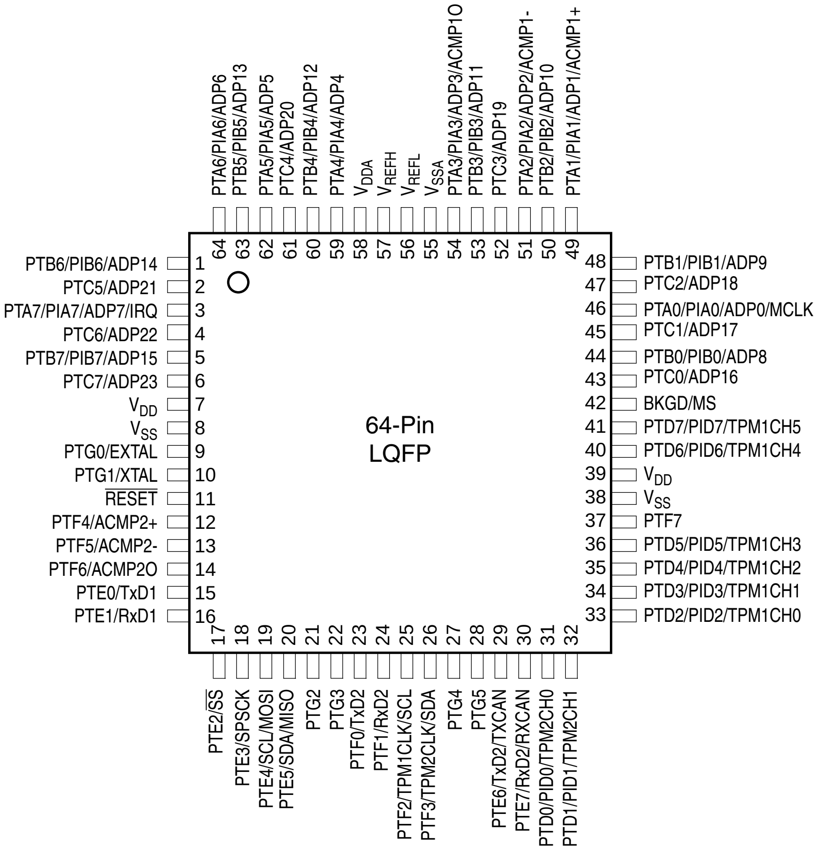


Figure 1. MC9S08DZ60 64-Pin LQFP

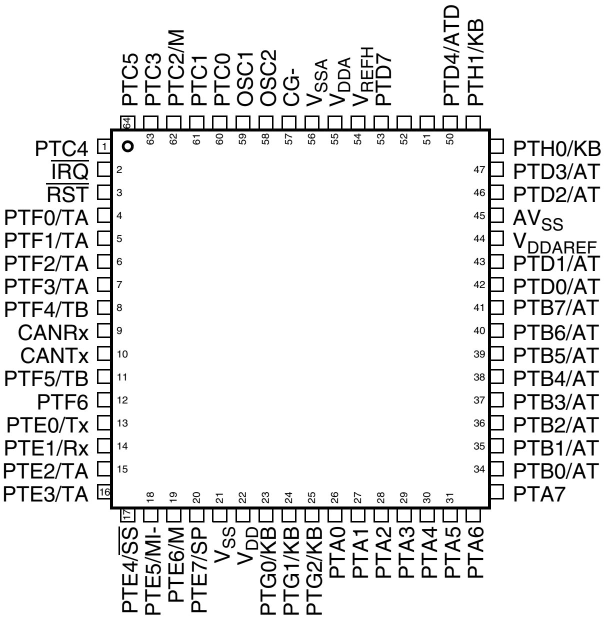


Figure 2. MC68HC908AZ60 64-Pin QFP

Table 2. MC9S08DZ60 64-QLFP and MC68HC908AZ60A Pinout Comparison (Sheet 1 of 2)

Pin #	MCHC08908AZ60A	MC9S08DZ60
1	PTC4	PTB6/PIB6/ADP14
2	$\overline{\text{IRQ}}$	PTC5/ADP21
3	$\overline{\text{RST}}$	PTA7/PIA7/ADP7/IRQ
4	PTF0/TACH2	PTC6/ADP22
5	PTF1/TACH3	PTB7/PIB7/ADP15
6	PTF2/TACH4	PTC7/ADP23
7	PTF3/TACH5	VDD
8	PTF4/TBCH0	VSS
9	CANRx	PTG0/EXTAL
10	CANTx	PTG1/XTAL
11	PTF5/TBCH1	$\overline{\text{RESET}}$
12	PTF6	PTF4/ACMP2+
13	PTE0/TxD	PTF5/ACMP2-
14	PTE1/RxD	PTF6/ACMP2O
15	PTE2/TACH0	PTE0/TxD1
16	PTE3/TACH1	PTE1/RxD1
17	$\overline{\text{PTE4/SS}}$	$\overline{\text{PTE2/SS}}$
18	PTE5/MISO	PTE3/SPSCK
19	PTE6/MOSI	PTE4/SCL/MOSI
20	PTE7/SPSCK	PTE5/SDA/MISO
21	VSS	PTG2
22	VDD	PTG3
23	PTG0/KBD0	PTF0/TxD2
24	PTG1/KBD1	PTF1/RxD2
25	PTG2/KBD2	PTF2/TPM1CLK/SCL
26	PTA0	PTF3/TPM2CLK/SDA
27	PTA1	PTG4
28	PTA2	PTG5
29	PTA3	PTE6/TxD2/TxCAN
30	PTA4	PTE7/RxD2/RxCAN
31	PTA5	PTD0/PID0/TPM2CH0
32	PTA6	PTD1/PID1/TPM2CH1
33	PTA7	PTD2/PID2/TPM1CH0
34	PTB0/ATD0	PTD3/PID3/TPM1CH1
35	PTB1/ATD1	PTD4/PID4/TPM1CH2
36	PTB2/ATD2	PTD5/PID5/TPM1CH3
37	PTB3/ATD3	PTF7
38	PTB4/ATD4	VSS
39	PTB5/ATD5	VDD

Table 2. MC9S08DZ60 64-QLFP and MC68HC908AZ60A Pinout Comparison (Sheet 2 of 2)

Pin #	MCHC08908AZ60A	MC9S08DZ60
40	PTB6/ATD6	PTD6/PID6/TPM1CH4
41	PTB7/ATD7	PTD7/PID7TPM1CH5
42	PTD0/ATD8	BKGD/MS
43	PTD1/ATD9	PTC0/ADP16
44	VDDAREF	PTB0/PIB0/ADP8
45	AVSS /VREFL	PTC1/ADP17
46	PTD2/ATD10	PTA0/PIA0/ADP0/MCLK
47	PTD3/ATD11	PTC2/ADP18
48	PTH0/KBD3	PTB1/PIB1/ADP9
49	PTH1/KBD4	PTA1/PAI1/ADP1/ACMP1+
50	PTD4/ATD12/TBCLK	PTB2/PIB2/ADP10
51	PTD5/ATD13	PTA2/PIA2/ADP2/ACMP1-
52	PTD6/ATD14/TACLK	PTC3/ADP19
53	PTD7	PTB3/PIB3/ADP11
54	VREFH	PTA3/PIA3/ADP3/ACMP10
55	VDDA	VSSA
56	VSSA	VREFL
57	CGMXFC	VREFH
58	OSC2	VDDA
59	OSC1	PTA4/PIA4/ADP4
60	PTC0	PTB4/PIB4/ADP12
61	PTC1	PTC4/ADP20
62	PTC2/MCLK	PTA5/PIA5/ADP5
63	PTC3	PTB5/PIB5/ADP13
64	PTC5	PTA6/PIA6/ADP6

4 Low-Power Modes

The MC9S08DZ60 has two low-power modes: stop3 and stop2. Unlike other devices in the HCS08 family, the MC9S08DZ60 does not support the low-power mode stop1.

Stop2 is a partial power-down mode in which most internal systems are turned off, but RAM remains powered. The registers are powered down in this mode, so they must be re-initialized. However, because RAM is powered in this mode, register values can be saved and restored. The internal voltage regulator is also turned off in this mode. The I/O pins remain latched in the state they were in upon entering stop2. Stop2 mode uses less power than stop3.

Stop3 is the same as the MC68HC908AZ60A stop mode. In this mode, the regulator is put into a loose regulation mode in which it consumes little current but can also support the static state of RAM and registers. One difference between stop3 and M68HC08 stop mode is how stop recovery is managed.

Stop recovery on the MC68HC908AZ60A is a timed event lasting 32 or 4096 oscillator cycles plus the time the oscillator source uses to start. On the MC9S08DZ60, stop recovery is based on the voltage regulator’s time to power up and return to full regulation.

Stop2 recovery on the MC9S08DZ60 always takes the reset vector regardless of the method waking the MCU from stop2. To distinguish a stop2 recovery from a normal reset, examine the status of the partial power down flag (PPDF) in the system power management status and control register 2 (SPMSC2). This flag is set upon waking up from stop2 and can direct user code to a stop2 recovery routine. PPDF remains set and the I/O pins remain latched until a 1 is written to the partial power down acknowledge (PPDACK) bit in SPMSC2.

For more information on HCS08 stop modes, see AN2493, *MC9S08GB/GT Low-Power Modes*.

5 Parallel Input/Output Control

The MC9S08DZ60 input/output (I/O) control differs from that of the MC68HC908AZ60A. The MC9S08DZ60 (64-LQFP) has seven parallel I/O port, which include a total of 53 bidirectional I/O pins and one input-only pin. The MC68HC908AZ60A (64-QFP) has seven parallel ports, which provide 50 bidirectional I/O pins.

The main I/O registers such as the data registers and data-direction registers are located at the start of the direct memory page on both devices.

Associated with the parallel I/O ports on the MC9S08DZ60 is a set of registers in the high-page register space that operate independently of the parallel I/O registers. These registers are used to control pullups, slew rate, and drive strength for the pins.

Key enhancements of the MC9S08DZ60 I/O:

- An internal pullup device can be enabled for each port pin by setting the corresponding bit in the pullup enable (PTxPEn) register in the high-page register space. 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 an analog function controls the pin.
- Slew rate control can be enabled for each port pin by setting the corresponding bit in the slew rate control (PTxSEn) register located in the high-page register space. 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 that are configured as inputs.
- An output pin can be selected to have high output drive strength by setting the corresponding bit in the drive strength select (PTxDSn) register in the high page register space. When high drive is selected, a pin can source and sink greater current. Every I/O pin can be selected as high drive, but the total current source and sink limits for the MCU must not be exceeded. Drive strength selection is intended to affect the DC behavior of I/O pins; however, the AC behavior is also affected. High drive allows a pin to drive a greater load with the same switching speed as a low-drive-enabled pin into a smaller load. Because of this, enabling pins as high drive may affect EMC emissions.

The MC9S08DZ60 and MC68HC908AZ60A parallel I/O control registers are provided in Figure 3, Figure 4 and Figure 5.

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
0x0000	**PTAD**	PTAD7	PTAD6	PTAD5	PTAD4	PTAD3	PTAD2	PTAD1	PTAD0
0x0001	**PTADD**	PTADD7	PTADD6	PTADD5	PTADD4	PTADD3	PTADD2	PTADD1	PTADD0
0x0002	**PTBD**	PTBD7	PTBD6	PTBD5	PTBD4	PTBD3	PTBD2	PTBD1	PTBD0
0x0003	**PTBDD**	PTBDD7	PTBDD6	PTBDD5	PTBDD4	PTBDD3	PTBDD2	PTBDD1	PTBDD0
0x0004	**PTCD**	PTCD7	PTCD6	PTCD5	PTCD4	PTCD3	PTCD2	PTCD1	PTCD0
0x0005	**PTCDD**	PTCDD7	PTCDD6	PTCDD5	PTCDD4	PTCDD3	PTCDD2	PTCDD1	PTCDD0
0x0006	**PTDD**	PTDD7	PTDD6	PTDD5	PTDD4	PTDD3	PTDD2	PTDD1	PTDD0
0x0007	**PTDDD**	PTDDD7	PTDDD6	PTDDD5	PTDDD4	PTDDD3	PTDDD2	PTDDD1	PTDDD0
0x0008	**PTED**	PTED7	PTED6	PTED5	PTED4	PTED3	PTED2	PTED1	PTED0
0x0009	**PTEDD**	PTEDD7	PTEDD6	PTEDD5	PTEDD4	PTEDD3	PTEDD2	PTEDD1	PTEDD0
0x000A	**PTFD**	PTFD7	PTFD6	PTFD5	PTFD4	PTFD3	PTFD2	PTFD1	PTFD0
0x000B	**PTFDD**	PTFDD7	PTFDD6	PTFDD5	PTFDD4	PTFDD3	PTFDD2	PTFDD1	PTFDD0
0x000C	**PTGD**	0	0	PTGD5	PTGD4	PTGD3	PTGD2	PTGD1	PTGD0
0x000D	**PTGDD**	0	0	PTGDD5	PTGDD4	PTGDD3	PTGDD2	PTGDD1	PTGDD0

Figure 3. MC9S08DZ60 Parallel I/O Control Direct Page Registers

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
0x1840	**PTAPE**	PTAPE7	PTAPE6	PTAPE5	PTAPE4	PTAPE3	PTAPE2	PTAPE1	PTAPE0
0x1841	**PTASE**	PTASE7	PTASE6	PTASE5	PTASE4	PTASE3	PTASE2	PTASE1	PTASE0
0x1847	Reserved	—	—	—	—	—	—	—	—
0x1848	**PTBPE**	PTBPE7	PTBPE6	PTBPE5	PTBPE4	PTBPE3	PTBPE2	PTBPE1	PTBPE0
0x1849	**PTBSE**	PTBSE7	PTBSE6	PTBSE5	PTBSE4	PTBSE3	PTBSE2	PTBSE1	PTBSE0
0x184A	**PTBDS**	PTBDS7	PTBDS6	PTBDS5	PTBDS4	PTBDS3	PTBDS2	PTBDS1	PTBDS0
0x184B	Reserved	—	—	—	—	—	—	—	—
0x184C	**PTBSC**	0	0	0	0	PTBIF	PTBACK	PTBIE	PTBMOD
0x184D	**PTBPS**	PTBPS7	PTBPS6	PTBPS5	PTBPS4	PTBPS3	PTBPS2	PTBPS1	PTBPS0
0x184E	**PTBES**	PTBES7	PTBES6	PTBES5	PTBES4	PTBES3	PTBES2	PTBES1	PTBES0
0x184F	Reserved	—	—	—	—	—	—	—	—
0x1850	**PTCPE**	PTCPE7	PTCPE6	PTCPE5	PTCPE4	PTCPE3	PTCPE2	PTCPE1	PTCPE0
0x1851	**PTCSE**	PTCSE7	PTCSE6	PTCSE5	PTCSE4	PTCSE3	PTCSE2	PTCSE1	PTCSE0
0x1852	**PTCDS**	PTCDS7	PTCDS6	PTCDS5	PTCDS4	PTCDS3	PTCDS2	PTCDS1	PTCDS0
0x1853– 0x1857	Reserved	— —	— —	— —	— —	— —	— —	— —	— —
0x1858	**PTDPE**	PTDPE7	PTDPE6	PTDPE5	PTDPE4	PTDPE3	PTDPE2	PTDPE1	PTDPE0
0x1859	**PTDSE**	PTDSE7	PTDSE6	PTDSE5	PTDSE4	PTDSE3	PTDSE2	PTDSE1	PTDSE0
0x185A	**PTDDS**	PTDDS7	PTDDS6	PTDDS5	PTDDS4	PTDDS3	PTDDS2	PTDDS1	PTDDS0
0x185B	Reserved	—	—	—	—	—	—	—	—
0x185C	**PTDSC**	0	0	0	0	PTDIF	PTDACK	PTDIE	PTDMOD
0x185D	**PTDPS**	PTDPS7	PTDPS6	PTDPS5	PTDPS4	PTDPS3	PTDPS2	PTDPS1	PTDPS0
0x185E	**PTDES**	PTDES7	PTDES6	PTDES5	PTDES4	PTDES3	PTDES2	PTDES1	PTDES0
0x185F	Reserved	—	—	—	—	—	—	—	—
0x1860	**PTEPE**	PTEPE7	PTEPE6	PTEPE5	PTEPE4	PTEPE3	PTEPE2	PTEPE1	PTEPE0
0x1861	**PTESE**	PTESE7	PTESE6	PTESE5	PTESE4	PTESE3	PTESE2	PTESE1	PTESE0
0x1862	**PTEDS**	PTEDS7	PTEDS6	PTEDS5	PTEDS4	PTEDS3	PTEDS2	PTEDS1	PTEDS0
0x1863– 0x1867	Reserved	— —	— —	— —	— —	— —	— —	— —	— —
0x1868	**PTFPE**	PTFPE7	PTFPE6	PTFPE5	PTFPE4	PTFPE3	PTFPE2	PTFPE1	PTFPE0
0x1869	**PTFSE**	PTFSE7	PTFSE6	PTFSE5	PTFSE4	PTFSE3	PTFSE2	PTFSE1	PTFSE0
0x186A	**PTFDS**	PTFDS7	PTFDS6	PTFDS5	PTFDS4	PTFDS3	PTFDS2	PTFDS1	PTFDS0
0x186B– 0x186F	Reserved	— —	— —	— —	— —	— —	— —	— —	— —
0x1870	**PTGPE**	0	0	PTGPE5	PTGPE4	PTGPE3	PTGPE2	PTGPE1	PTGPE0
0x1871	**PTGSE**	0	0	PTGSE5	PTGSE4	PTGSE3	PTGSE2	PTGSE1	PTGSE0
0x1872	**PTGDS**	0	0	PTGDS5	PTGDS4	PTGDS3	PTGDS2	PTGDS1	PTGDS0

Figure 4. MC9S08DZ60 Parallel I/O Control High Page Registers

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
$0000	Port A Data Register (PTA)	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
$0001	Port B Data Register (PTB)	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
$0002	Port C Data Register (PTC)	0	0	PTC5	PTC4	PTC3	PTC2	PTC1	PTC0
$0003	Port D Data Register (PTD)	PTD7	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
$0004	Data Direction Register A (DDRA)	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
$0005	Data Direction Register B (DDRB)	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
$0006	Data Direction Register C (DDRC)	MCLKEN	0	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
$0007	Data Direction Register D (DDRD)	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
$0008	Port E Data Register (PTE)	PTE7	PTE6	PTE5	PTE4	PTE3	PTE2	PTE1	PTE0
$0009	Port F Data Register (PTF)	0	PTF6	PTF5	PTF4	PTF3	PTF2	PTF1	PTF0
$000A	Port G Data Register (PTG)	0	0	0	0	0	PTG2	PTG1	PTG0
$000B	Port H Data Register (PTH)	0	0	0	0	0	0	PTH1	PTH0
$000C	Data Direction Register E (DDRE)	DDRE7	DDRE6	DDRE5	DDRE4	DDRE3	DDRE2	DDRE1	DDRE0
$000D	Data Direction Register F (DDRF)	0	DDRF6	DDRF5	DDRF4	DDRF3	DDRF2	DDRF1	DDRF0
$000E	Data Direction Register G (DDRG)	0	0	0	0	0	DDRG2	DDRG1	DDRG0
$000F	Data Direction Register H (DDRH)	0	0	0	0	0	0	DDRH1	DDRH0

Figure 5. MC68HC908AZ60A Parallel I/O Control Registers

The MC68HC908AZ60A uses a keyboard interrupt (KBI) module to provide five independently maskable external interrupt pins on the 64-pin package. The MC9S08DZ60 incorporates the features of the KBI into the parallel input/output control logic. Port A, port B, and port D pins can be configured as external interrupt inputs and as an external means of waking the MCU from stop2 or wait low-power modes.

Writing to the PTxPSn bits in the port interrupt pin select (PTxPS) register independently enables or disables each port pin. Each port can be configured as edge sensitive or edge and level sensitive based on the PTxMOD bit in the port interrupt status and control (PTxSC) register. Edge sensitivity can be software-programmed to be falling or rising; the level can be either low or high. The polarity of the edge or edge and level sensitivity is selected by the PTxESn bits in the port interrupt edge select (PTxES) register. Write a 1 to the PTxACK bit in PTxSC to clear interrupt flags. The PTxIE bit in PTxSC determines whether a port x interrupt is requested.

The MC9S08DZ60 registers for the port interrupt pins are shown in Figure 4. The MC68HC908AZ60A registers are shown in Figure 6.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
Keyboard Status and Control Register (KBSCR)	Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
	Write:	▨	▨	▨	▨	▨	ACKK		
	Reset:	0	0	0	0	0	0	0	0
Keyboard Interrupt Enable Register (KBIER)	Read:	0	0	0	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
	Write:	▨	▨	▨					
	Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented

Figure 6. MC68HC908AZ60A Keyboard Interrupt Registers

6 CPU

The CPU of the MC9S08DZ60 has a few enhancements over the MC68HC908AZ60A, but maintains backwards code compatibility. New opcodes improve C-code compiler efficiency. Many instructions also have slightly different cycle counts compared to the M68HC08 family.

6.1 New Opcodes

The instruction to enter the new background debug mode, BGND, has been added to the original M68HC08 opcode map. To improve C-code compiler efficiency, several addressing modes for manipulating the 16-bit H:X index register are added to the original three instructions. Ten new opcodes are added to the M68HC08 opcode map (see Table 3).

Table 3. New HCSO8 Opcodes

Instruction	Addressing Modes	New Opcodes
LDHX	EXT, IX, IX1, IX2, SPI	$32, $9EAE, $9ECE, $9EBE, $9EFE
CPHX	EXT, SP1	$3E, $9EF3
STHX	EXT, SP1	$96, $9EFF
BGD	INH	$82

6.2 Instruction Cycle Counts

Many instructions from the M68HC08 family have increased or decreased cycle counts by one or two cycles. For the instructions increased by one or two cycles, the increased bus speed performance of the MC9S08DZ60 will offset any cycle count increase. Because of these cycle changes, the MC9S08DZ60 is capable of faster bus speeds than the MC68HC908AZ60A.

Code written for the MC68HC908AZ60A that uses software delay loops may require rewriting for the MC9S08DZ60, depending on which instructions were used. For this reason, use the timer module to generate delays. Using the timer instead of software loops simplifies porting code from an MC68HC908AZ60A MCU to a MC9S08DZ60 MCU.

Table 4. HCS08 Instructions Reduced by One Cycle

Instruction	Address Modes	M68HC08 Cycles	HCS08 Cycles
DIV	INH	7	6
DAA	INH	7	1
TAP	INH	2	1
CLI	INH	2	1
SEI	INH	2	1

Table 5. HCS08 Instructions Reduced by Two Cycles

Instruction	Address Modes	M68HC08 Cycles	HCS08 Cycles
NSA	INH	3	1

Table 6. HCS08 Instructions Increased by One Cycle (Higher Speed of HCS08 Offsets Increase)

Instruction	Address Modes	M68HC08 Cycles	HCS08 Cycles
DBNZA	INH	3	4
DBNZX	INH	3	4
PULA	INH	2	3
PULX	INH	2	3
PULH	INH	2	3
STOP	INH	1	2
WAIT	INH	1	2
BSETn	INH	4	5
BCLRn	INH	4	5
CPHX	INH	4	5
NEG	DIR,IX,IX1,SP1	4,3,4,5	5,4,5,6
COM	DIR,IX,IX1,SP1	4,3,4,5	5,4,5,6
ASL	DIR,IX,IX1,SP1	4,3,4,5	5,4,5,6
ASR	DIR,IX,IX1,SP1	4,3,4,5	5,4,5,6
LSL	DIR,IX,IX1,SP1	4,3,4,5	5,4,5,6
LSR	DIR,IX,IX1,SP1	4,3,4,5	5,4,5,6
ROL	DIR,IX,IX1,SP1	4,3,4,5	5,4,5,6
ROR	DIR,IX,IX1,SP1	4,3,4,5	5,4,5,6
DEC	DIR,IX,IX1,SP1	4,3,4,5	5,4,5,6
INC	DIR,IX,IX1,SP1	4,3,4,5	5,4,5,6
TST	DIR,IX,IX1,SP1	3,2,3,4	4,3,4,5
JSR	DIR,EXT,IX	4,5,4	5,6,5
JMP	DIR,EXT,IX	2,3,2	3,4,3

Table 6. HCS08 Instructions Increased by One Cycle (Higher Speed of HCS08 Offsets Increase) (continued)

Instruction	Address Modes	M68HC08 Cycles	HCS08 Cycles
BSR	REL	4	5
CBEQ	IX+	4	5
ADC	IX	2	3
ADD	IX	2	3
AND	IX	2	3
BIT	IX	2	3
CMP	IX	2	3
CPX	IX	2	3
EOR	IX	2	3
LDA	IX	2	3
LDX	IX	2	3
ORA	IX	2	3
SBC	IX	2	3
SUB	IX	2	3

Table 7. HCS08 Instructions Increased by Two Cycles (Higher Speed of HCS08 Offsets Increase)

Instruction	Address Modes	M68HC08 Cycles	HCS08 Cycles
DBNZ	DIR,IX,IX1,SP1	3,2,3,4	5,4,5,6
CLR	DIR,IX,IX1,SP1	5,4,5,6	7,6,7,8
RTI	INH	7	9
RTS	INH	4	6
SWI	INH	9	11

6.3 Clocks

This section describes the differences between system clocks on the MC9S08DZ60 and the MC68HC908AZ60A. Section 8.2, “Multi-Purpose Clock Generator (MCG),” describes clock modules of the two MCUs.

On the MC68HC908AZ60A, the output of the primary clock source (whether it is an external crystal or PLL output) is divided by four to create the system bus clock. To obtain an 8 MHz bus clock, the oscillator must run at 32 MHz. The clock source on the HCS08 MCUs is divided by two instead of four; therefore, only a 16 MHz oscillator is required to obtain the same 8 MHz bus.

The M68HC08 CPU clock is equal to the bus clock. The HCS08 CPU clock is twice the speed of the bus clock. A typical HCS08 MCU has a maximum bus speed of 20 MHz and a maximum CPU speed of 40 MHz. The documented cycle times for the HCS08 CPU instructions remain referenced to bus cycles.

7 Memory

Figure 7 and Figure 8 show the memory maps of the MC9S08DZ60 and the MC68HC908AZ60A. Software should be modified to reflect these changes.

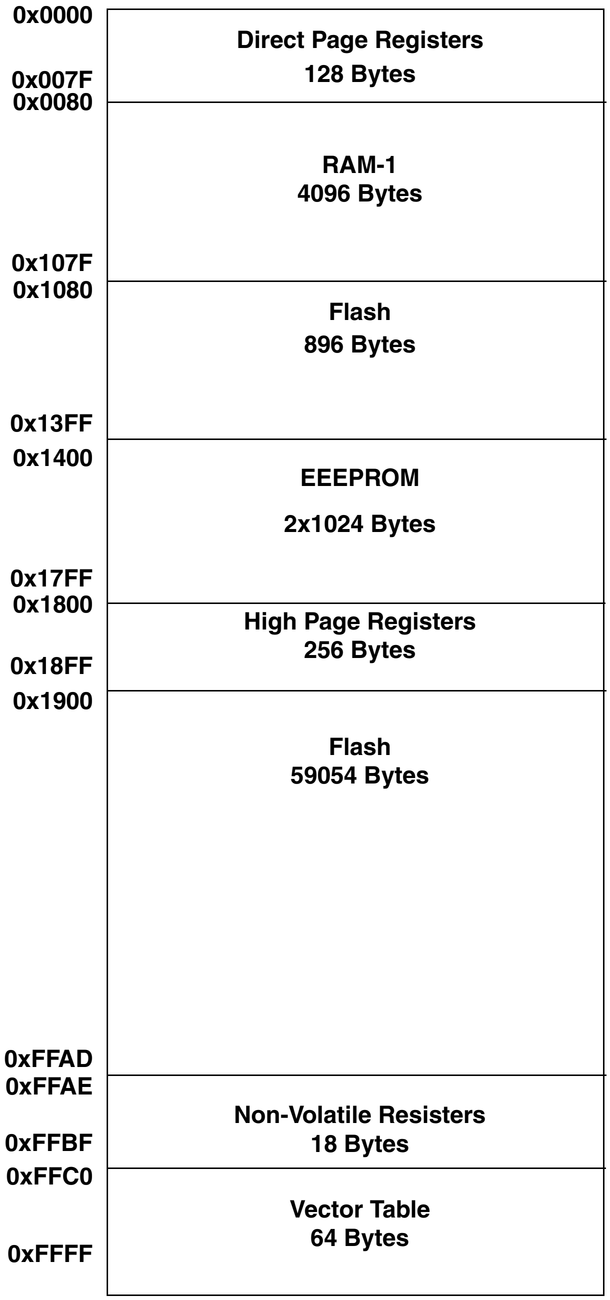


Figure 7. MC9S08DZ60 Memory Map

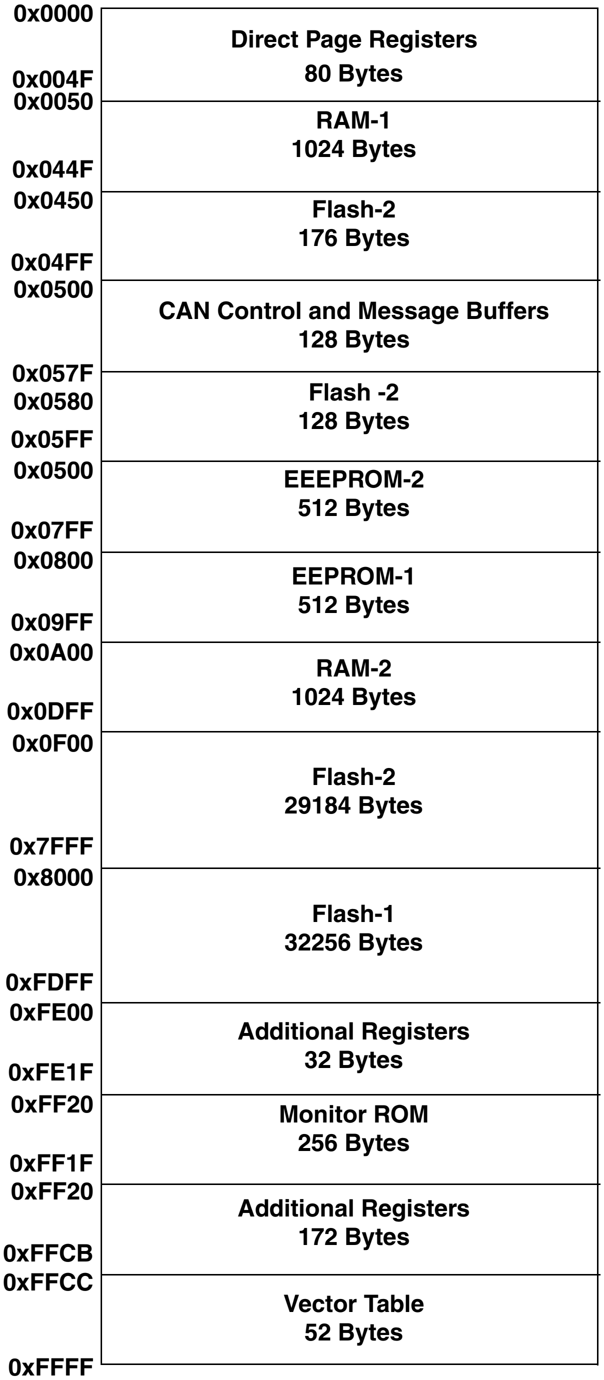


Figure 8. MC68HC908AZ60A Memory Map

7.1 Monitor ROM

All MC68HC908AZ60As, whether they are ROM-based or flash-based, have some on-chip ROM. In monitor mode, a small (approximately 200 byte) software routine resides in monitor ROM. These monitor ROMs sometimes include test firmware and/or flash programming routines.

As discussed in Section 4, “Low-Power Modes ,” monitor mode is eliminated on the MC9S08DZ60 and replaced with the unintrusive background debug module, eliminating the need for on-chip ROM.

7.2 Flash

The MC9S08DZ60 has a single non-continuous flash memory array of 60032 bytes. There are 896 bytes of flash memory from 0x1080 to 0x13FF and 59136 bytes from 0x1900 to 0xFFFF.

The MC9S08DZ60 flash is programmed byte by byte. It features a burst programming mode that can be used to program sequential bytes of data in less time than would normally be required if they were programmed individually.

The MC9S08DZ60 flash memory can be erased by sector (768 bytes) or mass erased.

The MC68HC908AZ60A has two flash memory arrays. Flash-1 is a continuous array of memory consisting of 32256 byes from 0x8000 to 0xFDFF. Flash-2 is a non-continuous array of memory consisting of 29488 bytes. There are 176 bytes of flash-2 memory from 0x0450 to 0x04FF, 128 bytes off flash-2 memory from 0x0580 to 0x05FF and 29184 bytes of flash-2 memory from 0x0E00 to 0xFDFF.

The MC68HC908AZ60A flash is row programmable (64 bytes). You can program a single byte in a row, but the programming voltage is applied to all bytes in the row. The MC68HC908AZ60A flash does not have a burst programming mode.

The MC68HC908AZ60A flash is erased by block (128 bytes) or mass erased.

The MC9S08DZ60 has a simplified command interface (CI) for programming, erasing, and blank-checking the flash array. The CI makes modifying the flash array easier from within the MCU application for tasks such as using a block of flash as EEPROM or upgrading the application firmware in the field.

The MC68HC908AZ60A flash interface is less automated and you must generate several delays between setting register bits to latch data and enable the programming voltage. In comparison, programming one byte of flash on a MC9S08DZ60 requires six steps, none of which requires a time delay. Programming one byte of flash on an MC68HC908AZ60A requires 13 steps, four of which require a time delay.

Key advantages of the MC9S08DZ60 CI:

- The MC9S08DZ60 CI automatically manages all timing delays. Set the flash and EEPROM clock divider (FCDIV) register to generate the specified flash and EEPROM clock frequency from the bus frequency. On the MC68HC908AZ60A, you must calculate all timing delays.
- The MC9S08DZ60 can program a byte of flash in only 20 μs. The MC68HC908AZ60A requires 30 μs.
- The MC9S08DZ60 CI has a blank check feature to verify that the entire array is blank. The MC68HC908AZ60A does not have this feature.
- The MC9S08DZ60 CI has an error check feature that sets if the flash command procedure is not obeyed. MC68HC908AZ60A does not have this feature.
- The MC9S08DZ60 background debug mode provides a convenient high-speed download to flash over a single-wire interface. The MC68HC908AZ60A requires monitor mode, which is more intrusive and slower.

7.3 EEPROM

On the MC9S08DZ60 there are 2048 bytes of paged EEPROM located in two 1024 byte arrays. Half of the EEPROM is in the memory map. The EEPROM page select (EPGSEL) bit in the flash memory and EEPROM configuration (FCNFG) register selects which array half can be accessed in foreground, but the other half cannot be accessed in background. Two mapping mode options can be selected to configure the 8-byte EEPROM sectors: 4-byte mode and 8-byte mode. Each mode is selected by the EPGMOD bit in the FOPT register.

In 4-byte sector mode (EPGMOD = 0), each 8-byte sector splits four bytes on foreground and four bytes on background but on the same addresses. The EPGSEL bit selects which four bytes can be accessed. During a sector erase, the entire 8-byte sector (four bytes in foreground and four bytes in background) is erased.

In 8-byte sector mode (EPGMOD = 1), each 8-byte sector is in a single page. The EPGSEL bit selects which sectors are on background. During a sector erase, the entire 8-byte sector in foreground is erased.

The MC9S08DZ60 EEPROM is programmed byte by byte. It features a burst programming mode to program sequential bytes of data in less time than normally required if individually programmed.

The MC9S08DZ60 EEPROM can be mass erased, or erased by sector, 4-byte or 8-byte depending how it is configured.

On the MC68HC908AZ60A there are 1024 bytes of available EEPROM, which are in two 512 byte arrays in the memory map. Each 512 byte EEPROM block is composed of four 128 byte pages which are byte, block, or bulk erasable.

The MC68HC908AZ60A EEPROM is byte programmable. It does not have a burst-programming mode.

On the MC9S08DZ60, the flash memory simplified command interface (CI) is shared by EEPROM for programming, erasing, and blank checking the EEPROM array with the same benefits and advantages as outlined previously.

Though the flash memory and EEPROM share the same command interface, it is not possible to program both at the same time.

Register	7	6	5	4	3	2	1	0
FICDIV	DIVLD	PRDIV8	DIV					
FOPT	KEYEN	FNORED	EPGMOD	0	0	0	SEC	
FTSTMOD	0	MRDS		0	0	0	0	0
FCNFG		EPGSEL	KEYACC	ECCDIS				
FPROT	EPS	FPS						
FSTAT	FCBEF	FCCF	FPVIOL	FACCERR		FBLANK		
FCMD	FCMID							

Figure 9. EEPROM and Flash Register Set

8 Peripherals

8.1 Keyboard Interrupt (KBI)

Unlike the MC68HC908AZ60A, the MC9S08DZ60 does not have a dedicated KBI module. Instead, the MC9S08DZ60 incorporates the features of the KBI into the parallel input/output control logic.

Key differences between the MC68HC908AZ60A KBI and the pin interrupts of the MC9S08DZ60 are:

- The MC9S08DZ60 has up to 24 external interrupt pins. The MC68HC908AZ60A has five KBI pins.
- External interrupt pins are present on every MC9S08DZ60 package type. The MC68HC908AZ60A KBI module is available only on the 64 QFP package.
- The MC9S08DZ60 external interrupt pins can be configured for falling edge, falling edge and level, rising edge, or rising edge and level sensitivity. The MC68HC908AZ60A KBI provides falling edge or falling edge and level sensitivity

You can configure port A, port B, and port D pins as external interrupt inputs and as an external means of waking the MCU from stop3 or wait modes.

Writing to the PTxPSn bits in the port interrupt pin select (PTxPS) register independently enables or disables each port pin.

Each port can be configured as edge-sensitive or edge-and-level sensitive based on the PTxMOD bit in the port interrupt status and control (PTxSC) register.

Edge sensitivity can be software programmed to be falling or rising; the level can be low or high. The polarity of the edge or edge and level sensitivity is selected using the PTxESn bits in the port interrupt edge select (PTxES) register.

The PTxES register interacts with the pullup enable register PTxPE. If a pullup is enabled, changing the polarity of the edge or edge-and-level sensitivity in PTxES will determine whether a pullup or pulldown device is selected by the corresponding bit in PTxPE. If falling edge or falling edge and low-level sensitivity is selected, PTxPE will enable a pullup. If rising edge or rising edge and high-level sensitivity is selected, PTXPE will enable a pulldown instead.

Write a 1 to the PTxACK bit in PTxSC to clear interrupt flags. The PTxIE bit in PTxSC determines whether a port x interrupt is requested.

The MC9S08DZ60 registers for the port A interrupt pins are shown in Figure 10. The MC9S08DZ60 registers for Ports B and D interrupt pins have the same format. The MC68HC908AZ60A registers are shown in Figure 11.

Register Acronym		7	6	5	4	3	2	1	0
PTASC	R	0	0	0	0	PTAIF	0	PTAIE	PTAMOD
	W	▨	▨	▨	▨	▨	PTAACK		
PTAPS	R	PTAPS7	PTAPS6	PTAPS5	PTAPS4	PTAPS3	PTAPS2	PTAPS1	PTAPS0
	W								
PTAES	R	PTAES7	PTAES6	PTAES5	PTAES4	PTAES3	PTAES2	PTAES1	PTAES0
	W								

▨ = Unimplemented or Reserved

Figure 10. MC9S08DZ60 Port A Pin Registers

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
Keyboard Status and Control Register (KBSCR)	Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
	Write:	▨	▨	▨	▨	▨	ACKK		
	Reset:	0	0	0	0	0	0	0	0
Keyboard Interrupt Enable Register (KBIER)	Read:	0	0	0	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
	Write:	▨	▨	▨					
	Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented

Figure 11. MC68HC908AZ60A Keyboard Interrupt Registers

8.2 Multi-Purpose Clock Generator (MCG)

The multi-purpose clock generator (MCG) is a new clock module designed for the MC9S08DZ60 and provides several clock source choices for the MCU. Similar to the MC68HC908AZ60A clock-generator module (CGM), it has a phase-locked loop (PLL) circuit for generating multiple frequencies. However, the MC9S08DZ60 MCG also has a frequency-locked loop (FLL) circuit and an internal clock source, which is available as a reference clock for both the PLL and FLL.

Key enhancements of MC9S08DZ60 MCG:

- The MC9S08DZ60 MCG has eight clock modes
  - — FLL-engaged internal (FEI)
  - — FLL-engaged external (FEE)
  - — FLL-bypassed internal reference (FBI)
  - — FLL-bypassed external reference (FBE)

- — PLL-engaged external (PEE)
- — PLL-bypassed external reference (PBE)
- — FLL-bypassed internal reference low-power (BLPI)
- — FLL-bypassed external reference low-power (BLPE)

- The MC9S08DZ60 MCG has a FLL and a PLL sourced by an internal or an external reference clock. The MC68HC908AZ60A CGM has a PLL and no internal oscillator.
- The MC9S08DZ60 MCG does not require external components. The MC68HC908AZ60A CGM requires a dedicated pin for the PLL filter components.
- The MC9S08DZ60 MCG can generate a 20 MHz bus clock. The MC68HC908AZ60A CGM can generate an 8 MHz bus clock.
- The MC9S08DZ60 MCG has a VCO divider and a bus divider to generate multiple bus frequencies. The VCO divider is used only in PLL-engaged modes and is selectable in integer steps of four from 4 through 40. FLL-engaged modes have a fixed multiplication factor of 1024. The bus divider is available in all clock modes and is selectable from 1, 2, 4, or 8. The MC68HC908AZ60A CGM has a multiplication factor equal to any integer from 1 to 15.
- The MC9S08DZ60 MCG uses a reference divider to ensure the reference clock falls within the input frequency ranges for the FLL and PLL. The FLL has an input range of 31.25 kHz to 39.0625 kHz. The PLL’s input reference range is 1 MHz to 2 MHz. The MC68HC908AZ60A CGM can select an appropriate VCO frequency range for its PLL.
- The MC9S08DZ60 MCU allows switching between the available clock modes. The MC9S08DZ60 MCG automates this process so you select the new clock source and reference clock, ensuring the reference frequency remains within the range required by the PLL or FLL. The MCG makes the switch when the new clock source is stable. Status bits are available to determine the current status of the MCG.
- The MC9S08DZ60 MCG and the MC68HC908AZ60A CGM have clock monitor circuits to detect a lock loss and force an interrupt. The HCS08 MCG also allows for a reset when a loss of external clock is detected.

The MCG and CGM registers are provided in Figure 12 and Figure 13.

Address	Register	7	6	5	4	3	2	1	0
0x0048	MCGC1	CLKS		RDIV			IREFS	IRCLKEN	IREFSTEN
0x0049	MCGC2	BDIV		RANGE	HGO	LP	EREFS	ERCLKEN	EREFSTEN
0x004A	MCGTRM	TRIM							
0x004B	MCGSC	LOLS	LOCK	PLLST	IREFST	CLKST		CSCINIT	FTRIM
0x004C	MCGC3	LOLIE	PLLS	CME	0	VDIV			
0x004D	MCGT	0	0	0	0	0	0	0	0

Figure 12. MC9S08DZ60 MCG Register Set

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
PLL Control Register (PCTL)	Read:	PLLIE	PLLF	PLLON	BCS	1	1	1	1
	Write:		▨			▨	▨	▨	▨
	Reset:	0	0	1	0	1	1	1	1
PLL Bandwidth Control Register (PBWC)	Read:	AUTO	LOCK	ACQ	XLD	0	0	0	0
	Write:		▨			▨	▨	▨	▨
	Reset:	0	0	0	0	0	0	0	0
PLL Programming Register (PPG)	Read:	MUL7	MUL6	MUL5	MUL4	VRS7	VRS6	VRS5	VRS4
	Write:								
	Reset:	0	1	1	0	0	1	1	0

▨ = Unimplemented

Figure 13. MC68HC908AZ60A CGM Register Set

8.3 Analog Comparator (ACMP)

The analog comparator (ACMP) is new for the HCS08 family of MCUs. No equivalent peripheral exists on any variant of the MC68HC908AZ60A.

The ACMP provides a circuit for comparing two analog voltages or for comparing one analog voltage to an internal reference.

Key features of the ACMP:

- Full rail-to-rail operation.
- Selectable interrupt on rising edge, falling edge, or rising or falling edges of comparator output.
- Option to compare to fixed internal bandgap reference voltage.
- Option to allow comparator output to be visible on a pin.
- The comparator output is a digital signal, which is high when the non-inverting input of the ACMP is greater than the inverting input, and is low when the non-inverting ACMP input is less than the inverting input.

For more information on the ACMP, please see the MC9S08DZ60 reference manual.

8.4 Analog-to-Digital Converter (ADC)

The MC9S08DZ60 features an ADC module designed for the HCS08 family. The MC9S08DZ60 ADC module uses linear successive approximation register architecture with sample and hold, but with the added resolution of 10-bit operation.

Table 8. Key Differences Between MC9S08DZ60 and MC68HC908AZ60A

	MC9S08DZ60	MC68HC908AZ60A
Resolution	10-bit or 8-bit	8-bit only
Conversion clock frequency	Up to 8 MHz	Up to 1 MHz
Temperature sensor	Yes	—
Hardware trigger	Yes	—
Automatic compare function	Yes	—
Operating modes	Run, wait, stop3	Run, wait
Bandgap voltage reference	Yes	—
Clock sources	4	2
ADC pin control	Yes	—
Single conversion time	2.8 μS (8-bit) 3.2 μS (10-bit)	17 μS (8-bit)
Clock prescalers	4 (divide by 1, 2, 4, or 8)	5 (divide by 1, 2, 4, 8, or 16)

The MC9S08DZ60 hardware trigger is the output from the real-time counter (RTC). When enabled, the hardware trigger initiates a conversion on an RTC overflow.

A temperature sensor is included in the MC9S08DZ60 ADC module. The temperature sensor output is attached to one of the ADC analog input channels. To calculate the temperature, read the temperature sensor channel, convert the reading into a voltage, and apply the formula:

$$\text{Temp} = 25 - ((\text{VTEMP} - \text{VTEMP25}) / m) \quad \textit{Eqn. 1}$$

Where:

- VTEMP is the voltage of the temperature sensor channel at the ambient temperature
- VTEMP25 the voltage of the temperature sensor channel at 25 °C
- m is the hot or cold voltage versus temperature slope in V/°C

Values for VTEMP25 and m are available from the ADC electricals table of the MC9S08DZ60 data sheet. If VTEMP is greater than VTEMP25 the cold slope value is applied in Equation 1. If VTEMP is less than VTEMP25 the hot slope value is applied in Equation 1.

The automatic compare function of the MC9S08DZ60 can be configured to check for an upper or lower limit. If the compare condition is met, the conversion complete flag (COCO) is set and an interrupt will occur if the ADC interrupt is enabled. If the compare condition is not met, the conversion complete flag is not set. The automatic compare function can monitor a voltage on a channel in either wait or stop3 mode. When the compare condition is met, the resulting ADC interrupt (if enabled) can wake the MCU.

The MC9S08DZ60 ADC has an internal channel connected to a constant voltage source, allowing software compensation for voltage drift of the MCU power supply.

The MC9S08DZ60 ADC can choose from four clock sources: bus clock, bus clock divided by two, alternate clock (ALTCLK), or an asynchronous clock (ADACK). ALTCLK is the external MCG clock reference. It remains active while the MCU is in wait mode, allowing the ADC to perform conversions. ADACK is generated from an asynchronous clock source within the ADC module. When selected as the

ADC clock source, ADACK remains active in wait or stop3 modes and allows conversions in these modes for lower noise operation.

The MC9S08DZ60 ADC pin control feature is used to disable the I/O port control of the pins used as analog inputs. The pin control registers (APCTL3, APCTL2, and APCTL1) give each ADC pin a control bit in addition to the input channel select, allowing pins used by the ADC to be reserved. This also masks pins that are not to be used as ADC inputs so they cannot accidentally be selected by the ADC input channel select.

The MC9S08DZ60 ADC requires 20 ADC clock cycles plus five bus cycles for a single 8-bit conversion. A single 10-bit conversion will require 23 ADC clock cycles and five bus cycles. Thus an MC9S08DZ60 running with a 16 MHz bus clock and an 8 MHz ADC clock performs a single 8-bit conversion in 2.8 µs and a single 10-bit conversion in 3.2 µs. By comparison, the MC68HC908AZ60A ADC requires 17 ADC clock cycles for a single 8-bit conversion, which will take 17.0 µS to complete, assuming the maximum conversion clock frequency of 1 MHz.

The MC9S08DZ60 ADC has only four clock prescalers compared to five on the MC68HC908AZ60A. The MC9S08DZ60 compensates for this by having four possible clock sources. In particular, the combination of clock sources, bus clock, and bus clock divided by 2 effectively increases the prescaler values available on the MC9S08DZ60 ADC to five (divide by 1, 2, 4, 8, or 16), matching the MC68HC908AZ60A.

The ADC registers for the MC9S08DZ60 and MC68HC908AZ60A are provided in Figure 14 and Figure 15, respectively.

Address	Register	7	6	5	4	3	2	1	0
0x0010	ADCSC1	COCO	AIEN	ADCO	ADCH				
0x0011	ADCSC2	ADACT	ADTRG	ACFE	ACFGT	—	—	—	—
0x0012	ADCRH	0	0	0	0	0	0	ADR9	ADR8
0x0013	ADCRL	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
0x0014	ADCCVH	0	0	0	0	0	0	ADCV9	ADCV8
0x0015	ADCCVL	ADCV7	ADCV6	ADCV5	ADCV4	ADCV3	ADCV2	ADCV1	ADCV0
0x0016	ADCCFG	ADLPC	ADIV		ADLSMP	MODE		ADICLK	
0x0017	APCTL1	ADPC7	ADPC6	ADPC5	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0
0x0018	APCTL2	ADPC15	ADPC14	ADPC13	ADPC12	ADPC11	ADPC10	ADPC9	ADPC8
0x0019	APCTL3	ADPC23	ADPC22	ADPC21	ADPC20	ADPC19	ADPC18	ADPC17	ADPC16

Figure 14. MC9S08DZ60 ADC Registers

Address	Register		7	6	5	4	3	2	1	0
0x0038	ADSCR	R:	COCO	AIEN	ADCO	ADCH4	ADCH3	ADCH2	ADCH1	ADCH04
		W:	R							
0x0039	ADR	R:	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
		W:	R	R	R	R	R	R	R	R
0x003A	ADICLk	R:	ADIV2	ADIV1	ADIV0	ADICLK	0	0	0	0
		W:					R	R	R	R

Figure 15. MC68HC908AZ60A ADC Registers

8.5 Inter-Integrated Circuit (IIC)

The inter-integrated circuit (IIC) on the MC9S08DZ60 communicates among many devices. No equivalent peripheral exists on any variant of the MC68HC908AZ60A.

The IIC 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 connected devices is limited by a maximum bus capacitance of 400 pF.

Key features of the IIC:

- Compatible with IIC bus standard
- Multi-master operation
- Software programmable for one of 64 different serial clock frequencies
- Software selectable acknowledge bit
- Interrupt driven byte-by-byte data transfer
- Arbitration lost interrupt with automatic mode switching from master to slave
- Calling address identification interrupt
- Start and stop signal generation/detection
- Repeated start signal generation
- Acknowledge bit generation/detection
- Bus busy detection
- General call recognition
- 10-bit address extension

For more information on the IIC, please see the MC9S08DZ60 reference manual.

8.6 Freescale Controller Area Network (MSCAN)

The MC9S08DZ60 MSCAN is an extended version of the MC68HC08 MSCAN. The MC9S08DZ60 and the MC68HC908AZ60A MSCANs support the CAN Protocol Specification, Version 2, A and B. Both MSCANs:

- Support remote frame handling
- Use triple-buffered transmit storage scheme with internal prioritization
- Have a programmable wakeup functionality with low-pass filter
- Have a programmable loop-back mode
- Have separate signalling and interrupts capabilities for all CAN receiver and transmitter error states
- Support receive and transmit time stamping
- Have a low-power sleep mode
- Have a programmable clock source: either CPU bus clock or crystal oscillator output
- Have a flexible maskable identifier filter

The MC9S08DZ60 MSCAN's additional features:

- Listen-only mode for monitoring the CAN bus
- Programmable bus-off recovery functionality
- Internal timer for time stamping

Table 9 displays the differences between the MC68HC908AZ60A and MC9S08DZ60 MSCAN implementation.

Table 9. Differences Between the MC68HC908AZ60A and MC9S08DZ60 MSCAN Implementation

Enhancements	MC68HC908AZ60A	MC9S08DZ60
Number of receive buffers	Two (one foreground and one background buffer)	Five (one foreground and four background buffer)
Number of identifier filters	One 32-bit filter, two 16-bit filters, or four 8-bit filters	Two 32-bit filters, four 16-bit filters, or eight 8-bit filters
Time stamp storage	Timer link to timer module with software storage	Automatic storage in transmit or receive buffer with using MSCAN internal 16-bit timer
Transmit buffers memory map	Direct access to all three transmit buffers	The three transmit buffers are paged (single memory address space) and accessible via transmit buffer selection flag
Number of control registers	9	13
Memory space for MSCAN module	128	64

8.7 Serial Peripheral Interface (SPI)

The MC9S08DZ60 SPI module is based on the M68HC08 family SPI module; however, several enhancements were made to improve functionality. The MC9S08DZ60 SPI can perform most functions of the MC68HC908AZ60A SPI.

Key enhancements of the MC9S08DZ60 SPI:

- The MC9S08DZ60 SPI has eight selectable baud rate prescalers in addition to eight selectable baud rates. The MC68HC908AZ60A SPI has four selectable baud rates.
- The MC9S08DZ60 has a double-buffered receiver; the MC68HC908AZ60A does not.
- The MC9S08DZ60 SPI has a bidirectional mode; the MC68HC908AZ60A SPI does not.
- The MC9S08DZ60 SPI can be configured for LSB- or MSB-first operation. The MC68HC908AZ60A SPI always sends the MSB first.
- The MC9S08DZ60 SPI can be configured to automatically shut down in wait mode to conserve power; the MC68HC908AZ60A SPI cannot.
- When configured as a master, the MC9S08DZ60 SPI can use the $\overline{\text{SS}}$ pin as an automatic slave select output, a master mode fault detect input, or a general-purpose I/O pin. On the MC68HC908AZ60A family, the $\overline{\text{SS}}$ pin can be used only as a master mode fault detect input or a general-purpose I/O pin.

Functions of the MC68HC908AZ60A SPI eliminated on the MC9S08DZ60 SPI:

- The MC68HC908AZ60A family supports wired-OR-mode; the MC9S08DZ60 family does not.
- The MC68HC908AZ60A family has an interruptible error flag for receiver overruns. (Receiver overruns are when a new byte is received before the previous byte has been read.) The MC9S08DZ60 family has a double-buffered receiver and therefore does not require a flag or interrupt for this condition.
- The MC68HC908AZ60A family has a separate interrupt enable bit (ERRIE) for two error conditions: receiver overflow (OVRF) or master mode fault detect (MODF). The MC9S08DZ60 family uses the SPI interrupt enable bit (SPIE) to enable both the receive buffer full (SPRF) and MODF interrupts.

Figure 16 and Figure 17 show the register sets for the MC9S08DZ60 and MC68HC908AZ60A SPI modules.

Register	7	6	5	4	3	2	1	0
SPIC1	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
SPIC2	0	0	0	MODFEN	BIDIROE	0	SPISWAI	SPSCO
SPIBR	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
SPIS	SPRF	0	SPTEF	MODF	0	0	0	0
SPID	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Figure 16. MC9S08DZ60 SPI Registers

Addr	Register Name	R/W	Bit 7	6	5	4	3	2	1	Bit 0
$0010	SPI Control Register (SPCR)	Read:	SPRIE	R	SPMSTR	CPOL	CPHA	SPWOM	SPE	SPTIE
		Write:								
		Reset:	0	0	1	0	1	0	0	0
$0011	SPI Status and Control Register (SPSCR)	Read:	SPRF	ERRIE	OVRF	MODF	SPTE	MODFEN	SPR1	SPR0
		Write:	▨		▨	▨	▨			
		Reset:	0	0	0	0	1	0	0	0
$0012	SPI Data Register (SPDR)	Read:	R7	R6	R5	R4	R3	R2	R1	R0
		Write:	T7	T6	T5	T4	T3	T2	T1	T0
		Reset:	Unaffected by Reset							

R = Reserved　▨ = Unimplemented

Figure 17. MC68HC908AZ60A SPI Registers

The most significant differences between the SPI modules are the baud-rate generation and the double-buffered receiver. The MC68HC908AZ60A SPI limits the baud-rate generation to four baud rate choices based from the internal bus clock. The four choices are the bus clock divided by 2, 8, 32, or 128.

The MC9S08DZ60 has more choices because of the use of a baud-rate prescaler. The prescaler can be one of eight integer values (from 1 to 8). The baud rate is then selected from one of eight divisor values: 2, 4, 8, 16, 32, 64, 128, or 256. The resulting baud rate is determined by the equation:

baud rate = bus clock ÷ prescaler ÷ divisor *Eqn. 2*

This equation results in 64 possible MC9S08DZ60 baud rates for a given bus frequency. Several possible values are repeated; for example, the two following combinations result in the same baud-rate value:

- prescaler = 1, divisor = 16
- prescaler = 2, divisor = 8

8.8 Serial Communications Interface (SCI)

The MC9S08DZ60 SCI module was derived from the HCS12 SCI and therefore has several changes from the MC68HC908AZ60A SCI.

Key enhancements of the MC9S08DZ60 SCI:

- The MC9S08DZ60 SCI has a 13-bit baud rate register, which can be set to any integer value from 1 to 8191. The MC68HC908AZ60A SCI has a 2-bit prescaler value that can be set to divide by 1, 3, 4, or 13 and a 3-bit baud rate selector that can be set to divide by 1, 2, 4, 8, 16, 32, 64, or 128 for a total of 32 combinations.
- The maximum baud rate of the MC9S08DZ60 SCI is the bus frequency divided by 16. For the MC68HC908AZ60A SCI, it is the bus frequency divided by 64.
- The MC9S08DZ60 SCI offers a single-wire mode; the MC68HC908AZ60A SCI does not.

- The MC9S08DZ60 SCI can be configured to stop automatically when wait mode is entered. The MC68HC908AZ60A SCI must be turned off by software if it is not needed in wait mode.
- The MC9S08DZ60 SCI offers selectable receiver input polarity by setting a single bit. The MC68HC908AZ60A SCI does not have this function.
- The MC9S08DZ60 SCI can be configured to generate an interrupt when an active edge occurs on the RxD pin. The MC68HC908AZ60A SCI does not have this function.
- The MC9S08DZ60 SCI has a LIN break detect circuit that will set the LIN break detect interrupt flag (LBKDIF) while inhibiting the receive data register full (RDRF) and framing error (FE) flags when a LIN break character is detected. The MC68HC908AZ60A cannot distinguish between a LIN break character and a shorter ordinary break character. It always sets the break (BKF), SCI receiver full (SCRF), and framing error (FE) flags.

The SCI registers for the MC9S08DZ60 and MC68HC908AZ60A are provided in Figure 18 and Figure 19, respectively.

Address	Register	7	6	5	4	3	2	1	0
0x0038	SCI1BDH	LBKDIE	REXEDGIE	0	SBR12	SBR11	SBR10	SBR9	SBR8
0x0039	SCI1BDL	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
0x003A	SCI1C1	LOOPS	SCISWAI	RSRC	M	WAKE	ILT	PE	PT
0x003B	SCI1C2	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
0x003C	SCI1S1	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
0x003D	SCI1S2	LBKDIF	RXEDGIF	0	RXINV	RWUID	BRK13	LBKDE	RAF
0x003E	SCI1S3	R8	T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE
0x003Ff	SCI1D	BIT 7	6	5	4	3	2	1	0

Figure 18. MC9S08DZ60 SCI Registers

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
SCI Control Register 1 (SCC1)	Read:	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
	Write:								
	Reset:	0	0	0	0	0	0	0	0
SCI Control Register 2 (SCC2)	Read:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
	Write:								
	Reset:	0	0	0	0	0	0	0	0
SCI Control Register 3 (SCC3)	Read:	R8	T8	R	R	ORIE	NEIE	FEIE	PEIE
	Write:	▨							
	Reset:	U	U	0	0	0	0	0	0
SCI Status Register 1 (SCS1)	Read:	SCTE	TC	SCRF	IDLE	OR	NF	FE	PE
	Write:	▨	▨	▨	▨	▨	▨	▨	▨
	Reset:	1	1	0	0	0	0	0	0
SCI Status Register 2 (SCS2)	Read:	0	0	0	0	0	0	BKF	RPF
	Write:	▨	▨	▨	▨	▨	▨	▨	▨
	Reset:	0	0	0	0	0	0	0	0
SCI Data Register (SCDR)	Read:	R7	R6	R5	R4	R3	R2	R1	R0
	Write:	T7	T6	T5	T4	T3	T2	T1	T0
	Reset:	Unaffected by Reset							
SCI Baud Rate Register (SCBR)	Read:	0	0	SCP1	SCP0	R	SCR2	SCR1	SCR0
	Write:	▨	▨						
	Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented U = Unaffected R = Reserved

Figure 19. MC68HC908AZ60A SCI Registers

8.9 Real-Time Counter (RTC)

The MC68HC908AZ60A does not have an equivalent to the real-time counter (RTC) module of the MC9S08DZ60; however, the programmable interrupt timer (PIT) module fulfils much of the same functionality. Both modules can generate periodic interrupts over a wide time range using prescalers, counters, and modulo registers.

Key enhancements of the RTC module:

- The RTC can use three different clock sources: an external clock, a 32 kHz internal clock, or a 1 kHz low-power clock. The PIT uses the MCU bus clock as its clock source.
- The RTC continues to run during wait, stop3, and stop2 modes. The PIT operates in wait mode; is inactive in stop mode.
- The RTC uses a prescaler block with binary and decimal selectable values giving 28 different prescaler choices (see Figure 20). The PIT has seven prescaler values: 1, 2, 4, 8, 32, and 64.

	RTCPS															
RTCLKS[0]	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	OFF	2^3	2^5	2^6	2^7	2^8	2^9	2^{10}	1	2	2^2	10	2^4	10^2	5x10^2	10^3
1	OFF	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}	10^3	2x10^3	5x10^3	10^4	2x10^4	5x10^4	10^5	2x10^5

Figure 20. MC9S08DZ60 RTC Prescaler Divide-By Values

Features of the PIT not implemented on the RTC:

- The PIT modulo and counter are 16-bit registers; the RTC modulo and counter are 8-bit. The smaller size of the RTC registers is offset by the extra prescalers and available clock sources. Taking advantage of the RTC prescalers and clock sources, a periodic interrupt within a range of intervals from 8 ms to almost 27 minutes can be generated. The PIT can generate periodic interrupts from 8 ms to a maximum of 8.4 s.
- The PIT has a stop bit which when set halts the PIT counter until it is cleared. The RTC has no equivalent feature.
- The PIT has a reset bit that clears the PIT counter and prescaler. There is no equivalent bit in the RTC, but writing to the RTC modulo register (RTCMOD) or changing the RTC clock source clears the RTC counter and prescaler.

The RTC registers for the MC9S08DZ60 and the PIT registers for the MC68HC908AZ60A are provided in Figure 21 and Figure 22, respectively.

Address	Register	7	6	5	4	3	2	1	0
0x006C	RTCSC	RTIF	RTCLKS		RTIE	RTCPS			
0x006D	RTCCNT	RTCCNT							
0x006E	RTCMOD	RTCMOD							

Figure 21. MC9S08DZ60 RTC Registers

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
PIT Status and Control Register (PSC)	Read:	POF	POIE	PSTOP	0	0	PPS2	PPS1	PPS0
	Write:	0			PRST	▨			
	Reset:	0	0	1	0	0	0	0	0
PIT Counter Register High (PCNTH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
	Write:	▨	▨	▨	▨	▨	▨	▨	▨
	Reset:	0	0	0	0	0	0	0	0
PIT Counter Register Low (PCNTL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
	Write:	▨	▨	▨	▨	▨	▨	▨	▨
	Reset:	0	0	0	0	0	0	0	0
PIT Counter Modulo Register High (PMODH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
	Write:								
	Reset:	1	1	1	1	1	1	1	1
PIT Counter Modulo Register Low (PMODL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
	Write:								
	Reset:	1	1	1	1	1	1	1	1

▨ =Unimplemented

Figure 22. MC68HC908AZ60A PIT Registers

8.10 Computer Operating Properly (COP)

The MC9S08DZ60 COP control bits are incorporated into the system options registers SOPT1 and SOPT2 in a similar way as how the MC68HC908AZ60A COP is controlled by bits in the configuration register CONFIG-1 (see Figure 23 and Figure 24).

Key enhancements:

- The MC9S08DZ60 COP is reset by writing 0x55 then 0xAA to the address of the system reset status register (SRS). If the computer fails to do this within the timeout period or writes any value other than 0x55 or 0xAA, the MCU is reset. The MC68HC908AZ60A COP is reset by writing any value to the COP control register (COPCTL), which overlaps the reset vector at 0xFFFF.
- The MC9S08DZ60 can perform windowed COP operation by setting COPW in the SOPT2 register. In this mode, writes to the SRS register to clear the COP must occur in the last 25 percent of the selected timeout period. A premature write immediately resets the MCU. The MC68HC908AZ60A COP cannot perform this function.
- The MC9S08DZ60 COP can use two possible clock sources: the bus clock or an internal 1 kHz clock. The MC68HC908AZ60A COP uses CGMXCLK; the MC68HC908AZ60A MCU external clock reference as its clock source.
- The MC9S08DZ60 COP has three associated timeouts for each clock source (see Figure 23). The MC68HC908AZ60A COP has only two timeouts: 2^{13}–2^{4} CGMXCLK cycles or 2^{18}–2^{4} CGMXCLK cycles.
- The MC9S08DZ60 COP is enabled out of reset to timeout after 2^{10} cycles of the internal 1 kHz clock. It can be disabled by setting COPT[1:0] to 0:0 in SOPT1. The MC68HC908AZ60A COP is

enabled out of reset to time out after $2^{18}–2^4$ CGMXCLK cycles. It can be disabled by setting COPD to 1 in CONFIG-1.

- Register SOPT1 and the COPCLKS and COPW bits in SOPT2 are write-once only. Write to SOPT1 and SOPT2 during the initialization routine to set the desired COP configuration, even if the desired COP configuration is the same as the default reset settings.

Control Bits		Clock Source	COP Window Opens (COP=1)	COP Overflow Count
COPCLKS	COPT[1:0]			
N/A	0:0	N/A	N/A	COP is disabled
0	0:1	1 kHz	N/A	2^5 cycles (32 ms[2])
0	1:0	1 kHz	N/A	2^8 cycles (256 ms[1])
0	1:1	1 kHz	N/A	2^{10} cycles (1.024 s[1])
1	0:1	Bus	6144 cycles	2^{13} cycles
1	1:0	Bus	49,152 cycles	2^{16} cycles
1	1:1	Bus	196,608 cycles	2^{18} cycles

Figure 23. MC9S08DZ60 COP Configuration Options

Address	Register	7	6	5	4	3	2	1	0
0x1800	SRS	POR	PIN	COP	ILOP	ILAD	LOCS	LVD	0
0x1802	SOPT1	COPT		STOPE	SCI2PS	IICPS	0	0	0
0x1803	RTCMOD	COPCLKS	COPW	0	ADHTS	0	MCSEL		

Figure 24. MC9S08DZ60 SOPT1 and SOPT2 Registers

Address: $001F

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	LVISTOP	R	LVIRST	LVIPWR	SSREC	COPL	STOP	COPD
Reset:	0	1	1	1	0	0	0	0
	R	= Reserved						

Figure 25. MC68HC908AZ60A CONFIG-1 Register

8.11 Low-Voltage Detect (LVD) System

Both the MC9S08DZ60 and the MC68HC908AZ60A have systems to protect against low-voltage conditions to preserve memory contents and control MCU operation during supply voltage variations. The MC9S08DZ60 uses a low-voltage detect (LVD) system while the MC68HC908AZ60A uses a low-voltage inhibit (LVI) module. The operation of these is similar, though the MC9S08DZ60 offers some enhancements over the MC68HC908AZ60A.

Key enhancements:

- The MC9S08DZ60 has a choice of low-voltage detection thresholds that trigger a reset. The MC68HC908AZ60A has a single low-voltage threshold.
- The MC9S08DZ60 has a warning flag indicating the supply is approaching the low-voltage condition. The low-voltage warning flag (LVWF) can generate an interrupt if desired. The MC9S08DZ60 has a choice of low-voltage warning thresholds that will set the LVWF. The MC68HC908AZ60A does not have this feature.
- The MC9S08DZ60 LVD is configured using the system power management status and control registers (SPMSC1 and SPMSC2). The MC68HC908AZ60A LVI is configured using CONFIG-1.

Address	Register	7	6	5	4	3	2	1	0
0x1809	SPMSC1	LVWF	LVWACK	LVWIE	LVDRE	LVDSE	LVDE	0	BGBE
0x180A	SPMSC2	0	0	LVDV	LVWV	PPDF	PPDACK	—	PPDC

Figure 26. MC9S08DZ60 SPMSC1 and 2 Registers

NOTE

Bits LVDRE and LVDE can be written only once after reset. Additional writes are ignored. Bit LVDV can be written only once after power-on-reset. Additional writes are ignored. Write these bits during the initialization routine to set the desired LVD configuration, even if the desired LVD configuration is the same as the default reset settings.

Address: $001F

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	LVISTOP	R	LVIRST	LVIPWR	SSREC	COPL	STOP	COPD
Reset:	0	1	1	1	0	0	0	0
	R	= Reserved						

Figure 27. MC68HC908AZ60A CONFIG-1 Register

The choice of the low-voltage detection threshold is determined by the value of LVDV in SPMSC2. Similarly, the low-voltage warning threshold is governed by LVWV. Figure 28 illustrates how these two bits combine to set the thresholds for low-voltage warning and detection. By comparison, the MC68HC908AZ60A has a single low-voltage detection threshold which is 3.8 V minimum.

LVDV:LVWV	LVW Trip Point	LVD Trip Point
0:0	V_{LVW0} = 2.74 V	V_{LVD0} = 2.56 V
0:1	V_{LVW1} = 2.92 V	
1:0	V_{LVW2} = 4.3 V	V_{LVD1} = 4.0 V
1:1	V_{LVW3} = 4.6 V	

Figure 28. MC9S08DZ60 LVD and LVW trip point typical values.

Refer to the electrical characteristics appendix of the MC9S08DZ60 reference manual for maximum and minimum values.

The MC9S08DZ60 LVD is disabled on entering any of the stop modes unless LVDSE is set in SPMSC1. If LVDSE and LVDE are set, the MCU cannot enter stop2 (it will enter stop3 instead), and the current consumption in stop3 with the LVD enabled will be higher.

8.12 Timer

The MC9S08DZ60 features a timer/PWM (TPM) module, which was designed for the HCS08 family. The new timer performs all the functions of the MC68HC908AZ60A's timer interface module (TIM). The MC9S08DZ60 TPM also reduces the complexity of the MC68HC908AZ60A TIM functions and improves the MCU resource usage.

Key enhancements:

- The TPM has an up/down count mode; the TIM only counts up.
- The TPM clock source can be the bus clock, an external clock, or the fixed system clock (XCLK, typically the external oscillator input to the ICG). The TIM clock source is limited to the bus clock or an external clock.
- The TPM has eight selectable prescalers; the TIM has seven.
- Any single channel can be configured for buffered PWM on the TPM. The TIM requires two channels to generate a buffered PWM.
- Center-aligned PWM signals can be created with the TPM. This is not possible with the TIM.

The register interface has been modified to make programming the TPM easier.

Register	7	6	5	4	3	2	1	0
TPMxSC	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
TPMxCNTH	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit9	Bit 8
TPMxCNTL	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TPMxMODH	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit9	Bit 8
TPMxMODL	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TPMxCnSC	CHnF	CHnIE	MSnB	MSnA	ELSnB	ELSnA	0	0
TPMxCnVH	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit9	Bit 8
TPMxCnVL	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Figure 29. MC9S08DZ60 TPM Registers

Register	7	6	5	4	3	2	1	0
TSC	TOFO	TOIE	TSTOP	TRST	0	PS2	PS1	PS0
TCNTH	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit9	Bit 8
TCNTL	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TMODH	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit9	Bit 8
TMODL	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TSCN	CHnF	CHnIE	MSnB	MSnA	ELSnB	ELSnA	TOVn	CHnMAX
TCHnH	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit9	Bit 8
TCHnL	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Figure 30. MC68HC908AZ60A TIM Registers

The 16-bit registers (for example, TPMxCNTH:TPMxCNTL) on the TPM can be accessed either high-byte or low-byte first. Accessing one byte latches the counter until the other byte is accessed. On the M68HC08 TIM, the high byte must always be accessed first to latch the current values.

8.12.1 Configuring Buffered PWM M68HC08

Configuring a buffered PWM is easier for an HCS08. To configure a buffered PWM for the M68HC08 TIM:

1. Stop the TIM counter by setting TSTOP, then reset the counter by setting TRST, both in the TSC register. The clock prescaler can also be selected by writing the PS2:PS1:PS0 bits in the same write.
2. Write the appropriate value for the PWM period in the TMODH:TMODL registers. TMODH must be written first or the timer overflow function will not operate.

3. Write the appropriate value for the duty cycle to an even-numbered TIM channel's TCHnH:TCHnL registers. An even numbered channel must be used when configuring a buffered PWM. The next channel, n+1, will not be available for other functions because the TCHn+1H:TCHn+1L registers will change the duty cycle.
4. Write 1:0 to the MSnB:MSnA bits of the even numbered TSCn register to select buffered PWM. Set the TOVn bit to toggle output on overflow and write either 1:0 (clear output on compare) or 1:1 (set output on compare) to the ELSnB:ELSnA bits. Do not set up to toggle output on compare (0:1) because this prevents reliable 0 percent duty cycle generation.
5. Clear the TSTOP bit in the TSC register when starting the PWM.
6. To update the duty cycle, write the new value to TCHn+1H:TCHn+1L registers. The update will take place on the next timer overflow. The next update must be made to the TCHnH:TCHnL registers, because the duty cycle is determined by the last channel's registers to be written. Therefore, software must keep track of which channel was written last.

8.12.2 Configuring Buffered PWM HCS08

For the HCS08 family, the steps to configure buffered PWM are:

1. Stop the TPM by writing 0:0 to the clock source select bits, CLKSB:CLKSA, in the TPMxSC register. Also during this write, set the desired clock prescaler by writing to the PS2:PS1:PS0 bits and set the CPWMS bit if center-aligned PWM is desired (center-aligned PWM is not an option on the M68HC08 TIM).
2. Write the appropriate value for the PWM period in the TPMxMODH:TPMxMODL registers. Either the high or low byte can be written first.
3. Write the appropriate value for the duty cycle to any TPM channel's TPMxCnVH:TPMxCnVL registers. Either the high or low byte can be written first.
4. If edge-aligned PWM has been selected (CPWMS = 0), set the MSnB bit of the TPMxCnSC register. If CPWMS = 1, the MSnB bit has no effect. For either edge- or center-aligned PWM, write 1:0 to the ELSBnB:ELSnA bits to clear output on compare or X:1 to set output on compare.
5. When it is time to start the PWM, write the appropriate value to the clock select bits (CLKSB:CLKSA) in the TPMxSC register to select the desired clock source for the TPM.
6. To update the duty cycle, write the new value to the TPMxCnVH:TPMxCnVL registers. Either the high or low byte can be written first and the new value will take effect the next time the counter matches the modulus value.

Comparing these steps shows the TPM is easier to program for buffered PWM signals because the TPM does not require tracking two different registers to perform buffered updates. The other functions of the timer modules include: input captures, output compares, and unbuffered PWM signals (the TPM always buffers the PWM) of the two timer modules are similarly configured. However, no limitations are placed on the order of accessing the 16-bit registers in the TPM.

9 Development Support Systems

Like all HCS08 microcontrollers, the MC9S08DZ60 uses a unique on-chip development support system that avoids the need for expensive traditional bus analyzers and emulators. This is a significant improvement on the existing application debugging techniques of the MC68HC908AZ60A and other HC08 devices, which are heavily dependent upon such tools.

The MC9S08DZ60 uses background debug mode (BDM) for development support. The hardware that supports background debug mode consists of a background debug controller (BDC) and an on-chip debug system (DBG). These modules are integrated into the hardware of every HCS08 microcontroller. The BDC and DBG modules can be configured to perform the most important functions associated with bus analyzers and emulators without additional cost.

9.1 Background Debug Mode vs. Monitor Mode

The MC9S08DZ60 uses background debug mode (BDM) instead of monitor mode used by the MC68HC908AZ60A. BDM offers several significant advantages over monitor mode. Key enhancements of the MC9S08DZ60 BDM are:

- BDM requires a simple interface to communicate with the MCU, using only the background debug pin (BKGD) for mode entry and serial communications (see Figure 31). It does not require any voltages beyond those normally used to supply the device. By contrast, monitor mode uses a significant amount of external circuitry and requires four I/O pins plus a high voltage on IRQ (see Figure 32)
- BDM operates at the full operating voltage and frequency range of the MC9S08DZ60. Monitor mode on the MC68HC908AZ60A is optimized for a limited number of frequencies and restricted to two baud rates per frequency.
- BDM maintains communication if the application bus clock frequency changes. Monitor mode loses communication if the bus clock frequency is altered.
- BDM is non-intrusive. The BDM interface can be connected to a running application without disturbing the program or forcing a reset. Monitor mode requires a reset of the target MCU.
- BDM can view and change internal registers and memory while running an application. Monitor mode interrupts the application to change registers and memory.
- BDM is active in stop3 and wait modes. Monitor mode is not available in stop or wait.
- There are no MCU pin function limitations with the MC9S08DZ60 BDM as it uses a dedicated pin (BKGD) for mode entry and communication. MC68HC908AZ60A Monitor mode compromises the functionality on five pins: IRQ, PTA0, PTC0, PTC1, and PTC3.
- In-circuit debugging is readily available on the MC9S08DZ60 due to the dedicated BKGD pin with no compromise on pin functionality. The MC68HC908AZ60A has limited in-circuit debugging due to the restrictions of monitor mode. Use of an emulator and expanded bus mode to overcome these restrictions result in further compromise to pin functionality as I/O signals on the MCU pins are replaced by data and address buses. With many I/O pins unavailable their original functions must be re-built outside the MCU. As a result, the re-built functions may not be exactly the same as the actual MCU.

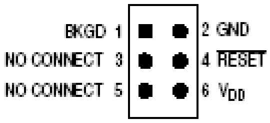
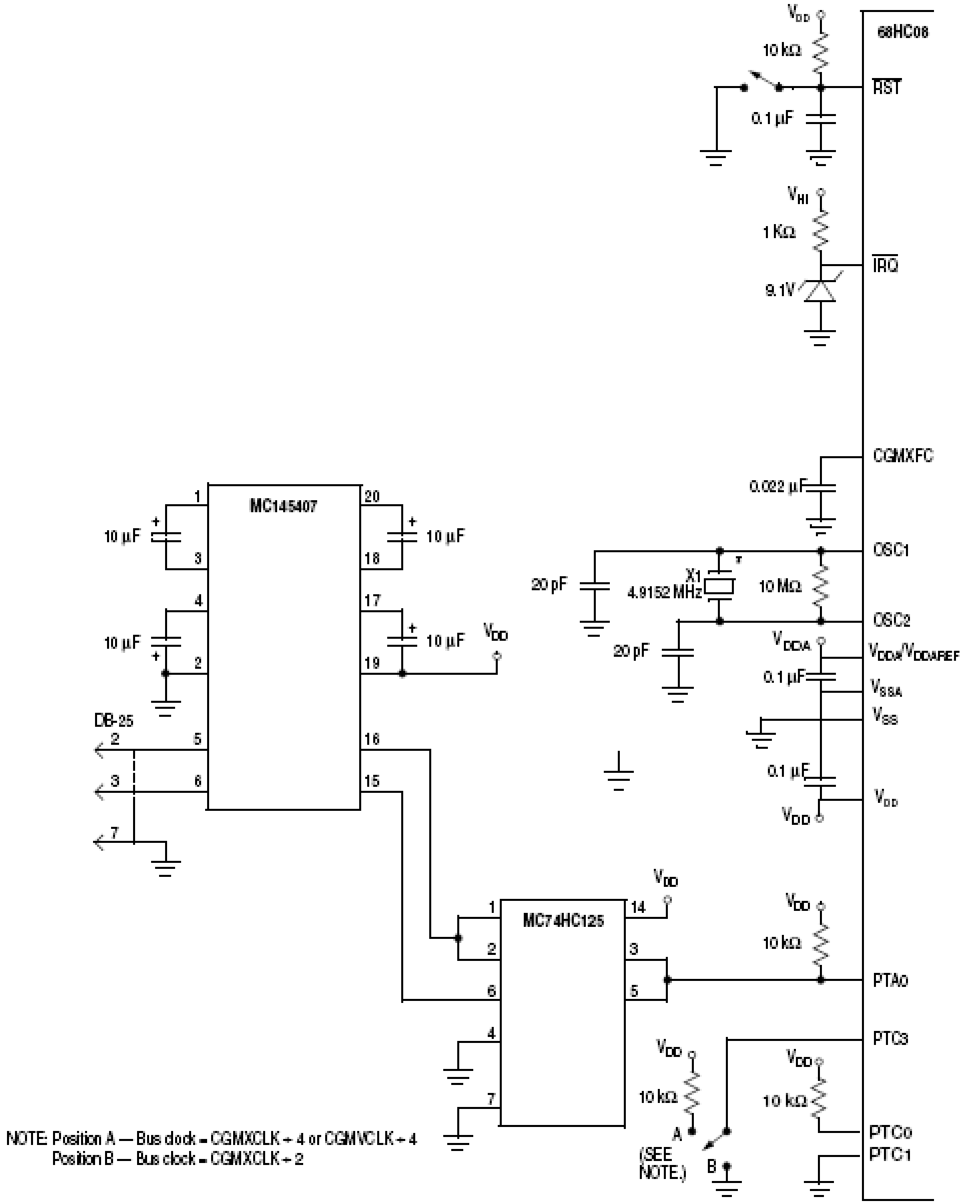


Figure 31. BDM Interface



NOTE: Position A — Bus clock = CGMXCLK ÷ 4 or CGMVCLK ÷ 4
Position B — Bus clock = CGMXCLK ÷ 2

* = Refer to Table 14-9 and Table 14-10 for correct value.

Figure 32. Monitor-Mode Interface

9.2 Monitor Mode

Monitor mode is an on-chip debug mode that has five commands: READ, WRITE, IREAD, IWRITE, and RUN. Monitor mode is implemented by a combination of hardware and firmware that redirects the reset and SWI vectors when enabled. These redirected vectors force the CPU to execute the monitor firmware instead of the normal user program. Therefore, monitor mode is always intrusive because it disrupts the normal program flow. This code takes control of one of the I/O port pins, PTA0, so a pin is lost during debug sessions. Monitor mode is also dependent on the internal bus frequency. Changing the bus frequency during a debug session will cause the host to lose communication with the MCU.

9.3 Background Debug Mode

Background debug mode uses the background debug controller (BDC) module to communicate with the microcontroller without interrupting the application. It can also employ the on-chip debug (DBG) module to monitor the address and data buses to debug the application, virtually eliminating the need for external emulators. BDM does not require any on-chip firmware.

9.3.1 Background Debug Controller

The BDC module provides a true single-pin communication interface to an HCS08 MCU. Usually this pin is reserved for BDC, so no additional general-purpose I/O port pin is needed during debugging. BDC can run from either the bus clock or an alternate 8-MHz clock source from the ICG when the bus clock is too slow for communication.

The MC9S08DZ60 BDC consists entirely of hardware logic. Firmware, such as the MC68HC908AZ60A monitor ROM, is not embedded inside the MCU. As a result the BDC does not need to use the CPU or its instructions. This means the BDC can access internal memory and registers without interrupting software.

The BDC has 30 commands (see Table 10), 13 of which are non-intrusive and allow reads and writes of on-chip memory without interrupting CPU operation. Also included is a sync command which allows the host to determine the communication speed so that the bus clock can be changed during a debug session without losing communication. Unlike the MC68HC908AZ60A monitor mode, the BDC never requires high voltage on any pin.

Table 10. BDC Command Summary

Command Mnemonic	Active BDM / Non-intrusive	Description
SYNC	Non-intrusive	Request a timed reference pulse to determine target BDC communication speed
ACK_ENABLE	Non-intrusive	Enable acknowledge protocol (refer to Freescale document order number HCS08RM v.1/D)
ACK_DISABLE	Non-intrusive	Disable acknowledge protocol (refer to Freescale document order number HCS08RM v.1/D)
BACKGROUND	Non-intrusive	Enter active background mode if enabled (ignore if ENBDM bit equals 0)
READ_STATUS	Non-intrusive	Read BDC status from BDCSCR
WRITE_CONTROL	Non-intrusive	Write BDC controls in BDCSCR
READ_BYTE	Non-intrusive	Read a byte from target memory
READ_BYTE_WS	Non-intrusive	Read a byte and report status

Table 10. BDC Command Summary (continued)

Command Mnemonic	Active BDM / Non-intrusive	Description
READ_LAST	Non-intrusive	Re-read byte from address just read and report status
WRITE_BYTE	Non-intrusive	Write a byte to target memory
WRITE_BYTE_WS	Non-intrusive	Write a byte and report status
READ_BKPT	Non-intrusive	Read BDCPKPT breakpoint register
WRITE_BKPT	Non-intrusive	Write BDCBKPT breakpoint register
GO	Active BDM	Go to execute the user application program staring at the address currently in the PC
TRACE1	Active BDM	Trace 1 user instruction at the address in the PC, then return to active background mode
TAGGO	Active BDM	Same as GO but enable external tagging (HCS08 devices have no external tagging pin)
READ_A	Active BDM	Read accumulator A
READ_CCR	Active BDM	Read condition code register
READ_PC	Active BDM	Read program counter
READ_HX	Active BDM	Read H and X register pair (H:X)
READ_SP	Active BDM	Read stack pointer (SP)
READ_NEXT	Active BDM	Increment H:X by one then read memory byte located at H:X
READ_NEXT_WS	Active BDM	Increment H:X by one then read memory byte located at H:X; report stats and data
WRITE_A	Active BDM	Write accumulator (A)
WRITE_CCR	Active BDM	Write condition code register
WRITE_PC	Active BDM	Write program counter (PC)
WRITE _HX	Active BDM	Write H and X register pair (H:X)
WRITE_SP	Active BDM	Write stack pointer (SP)
WRITE_NEXT	Active BDM	Increment H:X by one, then write memory byte located at H:X
WRITE_NEXT_WS	Active BDM	Increment H:X by one, then write memory byte located at H:X; also report status

9.3.2 On-Chip Debug System

The other hardware module that supports debugging in the DZ60 is the on-chip debug system (DBG). Unlike the MC68HC908AZ60A, the MC9S08DZ60 does not have any form of expanded bus mode. Therefore, to perform in-circuit emulation functions, the most important aspects of an emulator have been included on-chip.

The DBG module provides the bus state analysis function where it replaces an expensive external bus analyzing tool and is capable of capturing real-time bus information without stopping the application program. The data captured by the DBG can be retrieved via the BKGD pin of background debug controller.

The debug system consists of two trigger comparators and an 8-word FIFO buffer. Of the two comparators, one is always associated with the address bus and the other can be used to match values on the address or the data bus. The FIFO buffer can capture change of flow addresses or event only data. The information gathered in the FIFO is dependent upon the trigger mode of which nine are available.

The available trigger modes and the data they gather are:

- A only
  - Trigger when the address matches A
  - Captures change of flow address information
- A or B
  - Trigger when the address matches A or B
  - Captures change of flow address information
- A then B
  - Trigger when the address matches B but only after address A has matched first
  - Captures change of flow address information
- A and B data
  - Trigger when the address matches A and the data matches B
  - Captures change of flow address information
- A and NOT B data
  - Trigger when the address matches A and the data does not matches B
  - Captures change of flow address information
- Event Only B
  - Trigger each time the address matches B
  - Captures data information.
- A then Event Only B
  - Trigger each time the address matches B but only after address A has matched first
  - Captures data information
- Inside range (A address B)
  - Trigger when the address is inside the range defined by A and B
  - Captures change of flow address information
  - Outside range (address < A or address > B)
  - Trigger when the address is inside the range defined by A and B
  - Captures change of flow address information

When the 8-stage FIFO is used to capture change of flow address information, 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, the path of execution through many instructions can be reconstructed from the change-of-flow information stored in the FIFO.

The reason for these advanced debugging features is to eliminate the need for a costly external emulator system. With the DBG, you can perform these debugging functions using the actual silicon instead of an emulator that may or may not perform exactly like the real silicon.

For more information on the MC9S08DZ60 background debug mode, see Freescale document AN2497/D: *HCS08 Background Debug Mode Versus HC08Monitor Mode*.

10 Conclusion

Freescale’s MC9S08DZ60 microcontroller represents an evolutionary step from the MC68HC908AZ60A with modules offering improved performance and enhanced features. These changes should not present any difficulties for programmers or designers familiar with the MC68HC908AZ60A who wish to further enhance their application by migrating to the MC9S08DZ60.

All new features and enhancements should be taken into account when migrating from the MC68HC908AZ60A to the MC9S08DZ60. Read the relevant chapters of the latest MC9S08DZ60 and MC68HC908AZ60A specifications to ensure all new features and any differences have been fully captured.

11 Useful Links

Online Training

- Online HCS08 Family Training Courses

Online Development Tools

- MC68S08DZ60 Virtual Lab

Software Development Tools

- CodeWarrior for HC(S)08/RS08 Microcontrollers

Application Notes

- AN3335—Introduction to HCS08 Background Debug Mode
- AN3305—On-Chip System Protection Basics for HCS08 Automotive Microcontrollers
- AN2717—M68HC08 to HCS08 Transition
- AN2497—HCS08/RS08 Background Debug versus HC08 Monitor Mode
- AN2616—Getting Started with HCS08 and CodeWarrior Using C
- AN2295—Developer's Serial Bootloader for M68HC08 and HCS08 MCUs
- AN2596—Using the HCS08 Family On-Chip In-Circuit Emulator (ICE)

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals”, must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-compliant and/or non-Pb-free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale’s Environmental Products program, go to http://www.freescale.com/epp.