

#### 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

E·XF

Product Status	Active
Core Processor	508
Core Size	8-Bit
Speed	40MHz
Connectivity	I <sup>2</sup> C, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	5
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 4x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	8-SOIC (0.154", 3.90mm Width)
Supplier Device Package	8-SOIC
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s08sh8msc

Email: info@E-XFL.COM

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



## **Section Number**

### Title

## Page

	4.5.3	Program and Erase Command Execution	
	4.5.4	Burst Program Execution	
	4.5.5	Access Errors	51
	4.5.6	FLASH Block Protection	51
	4.5.7	Vector Redirection	52
4.6	Security	7	52
4.7	FLASH	Registers and Control Bits	53
	4.7.1	FLASH Clock Divider Register (FCDIV)	54
	4.7.2	FLASH Options Register (FOPT and NVOPT)	55
	4.7.3	FLASH Configuration Register (FCNFG)	56
	4.7.4	FLASH Protection Register (FPROT and NVPROT)	56
	4.7.5	FLASH Status Register (FSTAT)	57
	4.7.6	FLASH Command Register (FCMD)	58
		-	

# Chapter 5 Resets, Interrupts, and General System Control

5.1	Introduc	ction	
5.2	Features	5	59
5.3	MCU R	eset	59
5.4	Comput	er Operating Properly (COP) Watchdog	60
5.5	Interrup	ts	61
	5.5.1	Interrupt Stack Frame	62
	5.5.2	External Interrupt Request Pin (IRQ)	63
	5.5.3	Interrupt Vectors, Sources, and Local Masks	63
5.6	Low-Vo	ltage Detect (LVD) System	65
	5.6.1	Power-On Reset Operation	65
	5.6.2	Low-Voltage Detection (LVD) Reset Operation	65
	5.6.3	Low-Voltage Warning (LVW) Interrupt Operation	65
5.7	Reset, I	nterrupt, and System Control Registers and Control Bits	65
	5.7.1	Interrupt Pin Request Status and Control Register (IRQSC)	66
	5.7.2	System Reset Status Register (SRS)	67
	5.7.3	System Background Debug Force Reset Register (SBDFR)	68
	5.7.4	System Options Register 1 (SOPT1)	69
	5.7.5	System Options Register 2 (SOPT2)	70
	5.7.6	System Device Identification Register (SDIDH, SDIDL)	71
	5.7.7	System Power Management Status and Control 1 Register (SPMSC1)	72
	5.7.8	System Power Management Status and Control 2 Register (SPMSC2)	73

# Chapter 6 Parallel Input/Output Control

6.1	Port Data and Data Direction
6.2	Pull-up, Slew Rate, and Drive Strength76

MC9S08SH8 MCU Series Data Sheet, Rev. 3



Chapter 3 Modes of Operation



NP

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



### Table 4-4. Nonvolatile Register Summary

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



Chapter 7 Central Processor Unit (S08CPUV2)

# 7.3.5 Extended Addressing Mode (EXT)

In extended addressing mode, the full 16-bit address of the operand is located in the next two bytes of program memory after the opcode (high byte first).

## 7.3.6 Indexed Addressing Mode

Indexed addressing mode has seven variations including five that use the 16-bit H:X index register pair and two that use the stack pointer as the base reference.

## 7.3.6.1 Indexed, No Offset (IX)

This variation of indexed addressing uses the 16-bit value in the H:X index register pair as the address of the operand needed to complete the instruction.

## 7.3.6.2 Indexed, No Offset with Post Increment (IX+)

This variation of indexed addressing uses the 16-bit value in the H:X index register pair as the address of the operand needed to complete the instruction. The index register pair is then incremented (H:X = H:X + 0x0001) after the operand has been fetched. This addressing mode is only used for MOV and CBEQ instructions.

## 7.3.6.3 Indexed, 8-Bit Offset (IX1)

This variation of indexed addressing uses the 16-bit value in the H:X index register pair plus an unsigned 8-bit offset included in the instruction as the address of the operand needed to complete the instruction.

## 7.3.6.4 Indexed, 8-Bit Offset with Post Increment (IX1+)

This variation of indexed addressing uses the 16-bit value in the H:X index register pair plus an unsigned 8-bit offset included in the instruction as the address of the operand needed to complete the instruction. The index register pair is then incremented (H:X = H:X + 0x0001) after the operand has been fetched. This addressing mode is used only for the CBEQ instruction.

## 7.3.6.5 Indexed, 16-Bit Offset (IX2)

This variation of indexed addressing uses the 16-bit value in the H:X index register pair plus a 16-bit offset included in the instruction as the address of the operand needed to complete the instruction.

## 7.3.6.6 SP-Relative, 8-Bit Offset (SP1)

This variation of indexed addressing uses the 16-bit value in the stack pointer (SP) plus an unsigned 8-bit offset included in the instruction as the address of the operand needed to complete the instruction.



Bit-Mani	pulation	Branch		Rea	ad-Modify-W	/rite		Cor	ntrol			Register	/Memory		
00 5	10 5	20 3	30 5	40 1	50 1	60 5	70 4	80 9	90 3	A0 2	B0 3	C0 4	D0 4	E0 3	F0 3
BRSET0	BSET0	BRA	NEG	NEGA	NEGX	NEG	NEG	RTI	BGE	SUB	SUB	SUB	SUB	SUB	SUB
3 DIR	2 DIR	2 REL	2 DIR	1 INH	1 INH	2 IX1	1 IX	1 INH	2 REL	2 IMM	2 DIR	3 EXT	3 IX2	2 IX1	1 IX
01 5	11 5	21 3	31 5	41 4	51 4	61 5	71 5	81 6	91 3	A1 2	B1 3	C1 4	D1 4	E1 3	F1 3
BRCLR0	BCLR0	BRN	CBEQ	CBEQA	CBEQX	CBEQ	CBEQ	RTS	BLT	CMP	CMP	CMP	CMP	CMP	CMP
3 DIR	2 DIR	2 REL	3 DIR	3 IMM	3 IMM	3 IX1+	2 IX+	1 INH	2 REL	2 IMM	2 DIR	3 EXT	3 IX2	2 IX1	1 IX
02 5	12 5	22 3	32 5	42 5	52 6	62 1	72 1	82 5+	92 3	A2 2	B2 3	C2 4	D2 4	E2 3	F2 3
BRSET1	BSET1	BHI	LDHX	MUL	DIV	NSA	DAA	BGND	BGT	SBC	SBC	SBC	SBC	SBC	SBC
3 DIR	2 DIR	2 REL	3 EXT	1 INH	1 INH	1 INH	1 INH	1 INH	2 REL	2 IMM	2 DIR	3 EXT	3 IX2	2 IX1	1 IX
03 5	13 5	23 3	33 5	43 1	53 1	63 5	73 4	83 11	93 3	A3 2	B3 3	C3 4	D3 4	E3 3	F3 3
BRCLR1	BCLR1	BLS	COM	COMA	COMX	COM	COM	SWI	BLE	CPX	CPX	CPX	CPX	CPX	CPX
3 DIR	2 DIR	2 REL	2 DIR	1 INH	1 INH	2 IX1	1 IX	1 INH	2 REL	2 IMM	2 DIR	3 EXT	3 IX2	2 IX1	1 IX
04 5 BRSET2 3 DIR	14 5 BSET2 2 DIR	24 3 BCC 2 REL	LSR 2 DIR	44 1 LSRA 1 INH	LSRX 1 INH	64 5 LSR 2 IX1	<sup>74</sup> 4 LSR 1 IX	84 1 TAP 1 INH	94 2 TXS 1 INH	A4 2 AND 2 IMM	AND 2 DIR	AND 3 EXT	D4 4 AND 3 IX2	E4 3 AND 2 IX1	F4 3 AND 1 IX
05 5 BRCLR2 3 DIR	15 5 BCLR2 2 DIR	25 3 BCS 2 REL	35 4 STHX 2 DIR	45 3 LDHX 3 IMM	LDHX 2 DIR	65 3 CPHX 3 IMM	75 5 CPHX 2 DIR	85 1 TPA 1 INH	95 2 TSX 1 INH	A5 2 BIT 2 IMM	B5 3 BIT 2 DIR	BIT 3 EXT	BIT 3 IX2	BIT 2 IX1	BIT 1 IX
06 5 BRSET3 3 DIR	16 5 BSET3 2 DIR	26 3 BNE 2 REL	36 5 ROR 2 DIR	46 1 RORA 1 INH	RORX 1 INH	66 5 ROR 2 IX1	76 4 ROR 1 IX	PULA 1 INH	96 5 STHX 3 EXT	A6 2 LDA 2 IMM	LDA 2 DIR	LDA 3 EXT	D6 4 LDA 3 IX2	E6 3 LDA 2 IX1	F6 3 LDA 1 IX
07 5	17 5	27 3	37 5	47 1	57 1	67 5	77 4	87 2	97 1	A7 2	B7 3	C7 4	D7 4	E7 3	F7 2
BRCLR3	BCLR3	BEQ	ASR	ASRA	ASRX	ASR	ASR	PSHA	TAX	AIS	STA	STA	STA	STA	STA
3 DIR	2 DIR	2 REL	2 DIR	1 INH	1 INH	2 IX1	1 IX	1 INH	1 INH	2 IMM	2 DIR	3 EXT	3 IX2	2 IX1	1 IX
08 5	18 5	28 3	38 5	48 1	58 1	68 5	78 4	88 3	98 1	A8 2	B8 3	C8 4	D8 4	E8 3	F8 3
BRSET4	BSET4	BHCC	LSL	LSLA	LSLX	LSL	LSL	PULX	CLC	EOR	EOR	EOR	EOR	EOR	EOR
3 DIR	2 DIR	2 REL	2 DIR	1 INH	1 INH	2 IX1	1 IX	1 INH	1 INH	2 IMM	2 DIR	3 EXT	3 IX2	2 IX1	1 IX
09 5	19 5	29 3	39 5	49 1	59 1	69 5	79 4	89 2	99 1	A9 2	B9 3	C9 4	D9 4	E9 3	F9 3
BRCLR4	BCLR4	BHCS	ROL	ROLA	ROLX	ROL	ROL	PSHX	SEC	ADC	ADC	ADC	ADC	ADC	ADC
3 DIR	2 DIR	2 REL	2 DIR	1 INH	1 INH	2 IX1	1 IX	1 INH	1 INH	2 IMM	2 DIR	3 EXT	3 IX2	2 IX1	1 IX
0A 5	1A 5	2A 3	3A 5	4A 1	5A 1	6A 5	7A 4	8A 3	9A 1	AA 2	BA 3	CA 4	DA 4	EA 3	FA 3
BRSET5	BSET5	BPL	DEC	DECA	DECX	DEC	DEC	PULH	CLI	ORA	ORA	ORA	ORA	ORA	ORA
3 DIR	2 DIR	2 REL	2 DIR	1 INH	1 INH	2 IX1	1 IX	1 INH	1 INH	2 IMM	2 DIR	3 EXT	3 IX2	2 IX1	1 IX
0B 5	1B 5	2B 3	3B 7	4B 4	5B 4	6B 7	7B 6	8B 2	9B 1	AB 2	BB 3	CB 4	DB 4	EB 3	FB 3
BRCLR5	BCLR5	BMI	DBNZ	DBNZA	DBNZX	DBNZ	DBNZ	PSHH	SEI	ADD	ADD	ADD	ADD	ADD	ADD
3 DIR	2 DIR	2 REL	3 DIR	2 INH	2 INH	3 IX1	2 IX	1 INH	1 INH	2 IMM	2 DIR	3 EXT	3 IX2	2 IX1	1 IX
0C 5	1C 5	2C 3	3C 5	4C 1	5C 1	6C 5	7C 4	8C 1	9C 1		BC 3	CC 4	DC 4	EC 3	FC 3
BRSET6	BSET6	BMC	INC	INCA	INCX	INC	INC	CLRH	RSP		JMP	JMP	JMP	JMP	JMP
3 DIR	2 DIR	2 REL	2 DIR	1 INH	1 INH	2 IX1	1 IX	1 INH	1 INH		2 DIR	3 EXT	3 IX2	2 IX1	1 IX
0D 5	1D 5	2D 3	3D 4	4D 1	5D 1	6D 4	7D 3		9D 1	AD 5	BD 5	CD 6	DD 6	ED 5	FD 5
BRCLR6	BCLR6	BMS	TST	TSTA	TSTX	TST	TST		NOP	BSR	JSR	JSR	JSR	JSR	JSR
3 DIR	2 DIR	2 REL	2 DIR	1 INH	1 INH	2 IX1	1 IX		1 INH	2 REL	2 DIR	3 EXT	3 IX2	2 IX1	1 IX
0E 5 BRSET7 3 DIR	1E 5 BSET7 2 DIR	2E 3 BIL 2 REL	3E 6 CPHX 3 EXT	4E 5 MOV 3 DD	5E 5 MOV 2 DIX+	6E 4 MOV 3 IMD	7E 5 MOV 2 IX+D	8E 2+ STOP 1 INH	9E Page 2	AE 2 LDX 2 IMM	BE 3 LDX 2 DIR	CE 4 LDX 3 EXT	DE 4 LDX 3 IX2	EE 3 LDX 2 IX1	FE 3 LDX 1 IX
0F 5	1F 5	2F 3	3F 5	4F 1	5F 1	6F 5	7F 4	8F 2+	9F 1	AF 2	BF 3	CF 4	DF 4	EF 3	FF 2
BRCLR7	BCLR7	BIH	CLR	CLRA	CLRX	CLR	CLR	WAIT	TXA	AIX	STX	STX	STX	STX	STX
3 DIR	2 DIR	2 REL	2 DIR	1 INH	1 INH	2 IX1	1 IX	1 INH	1 INH	2 IMM	2 DIR	3 EXT	3 IX2	2 IX1	1 IX

### Table 7-3. Opcode Map (Sheet 1 of 2)

	Inherent Immediate
DIR	Direct
DD IX+D	DIR to DIR

REL IX IX1 IX2 IMD DIX+

Relative Indexed, No Offset Indexed, 8-Bit Offset Indexed, 16-Bit Offset IMM to DIR DIR to IX+

Stack Pointer, 8-Bit Offset Stack Pointer, 16-Bit Offset Indexed, No Offset with Post Increment Indexed, 1-Byte Offset with Post Increment

SP1 SP2 IX+

IX1+

MC9S08SH8 MCU Series Data Sheet, Rev. 3

Opcode in Hexadecimal SUB 1 IX Addressing Mode



Chapter 8 Analog Comparator (S08ACMPV2)



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

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

# 9.3.8 Pin Control 1 Register (APCTL1)

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



Figure 9-11. Pin Control 1 Register (APCTL1)

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

### Table 9-9. APCTL1 Register Field Descriptions



result of the conversion is transferred to ADCRH and ADCRL upon completion of the conversion algorithm.

If the bus frequency is less than the  $f_{ADCK}$  frequency, precise sample time for continuous conversions cannot be guaranteed when short sample is enabled (ADLSMP=0). If the bus frequency is less than 1/11th of the  $f_{ADCK}$  frequency, precise sample time for continuous conversions cannot be guaranteed when long sample is enabled (ADLSMP=1).

The maximum total conversion time for different conditions is summarized in Table 9-12.

Conversion Type	ADICLK	ADLSMP	Max Total Conversion Time
Single or first continuous 8-bit	0x, 10	0	20 ADCK cycles + 5 bus clock cycles
Single or first continuous 10-bit	0x, 10	0	23 ADCK cycles + 5 bus clock cycles
Single or first continuous 8-bit	0x, 10	1	40 ADCK cycles + 5 bus clock cycles
Single or first continuous 10-bit	0x, 10	1	43 ADCK cycles + 5 bus clock cycles
Single or first continuous 8-bit	11	0	5 μs + 20 ADCK + 5 bus clock cycles
Single or first continuous 10-bit	11	0	5 μs + 23 ADCK + 5 bus clock cycles
Single or first continuous 8-bit	11	1	5 μs + 40 ADCK + 5 bus clock cycles
Single or first continuous 10-bit	11	1	5 μs + 43 ADCK + 5 bus clock cycles
Subsequent continuous 8-bit; $f_{BUS} \ge f_{ADCK}$	xx	0	17 ADCK cycles
Subsequent continuous 10-bit; $f_{BUS} \ge f_{ADCK}$	XX	0	20 ADCK cycles
Subsequent continuous 8-bit; $f_{BUS} \ge f_{ADCK}/11$	xx	1	37 ADCK cycles
Subsequent continuous 10-bit; $f_{BUS} \ge f_{ADCK}/11$	xx	1	40 ADCK cycles

Table 9-12. Total Conversion Time vs. Control Conditions

The maximum total conversion time is determined by the clock source chosen and the divide ratio selected. The clock source is selectable by the ADICLK bits, and the divide ratio is specified by the ADIV bits. For example, in 10-bit mode, with the bus clock selected as the input clock source, the input clock divide-by-1 ratio selected, and a bus frequency of 8 MHz, then the conversion time for a single conversion is:

Conversion time =  $\frac{23 \text{ ADCK cyc}}{8 \text{ MHz/1}}$  +  $\frac{5 \text{ bus cyc}}{8 \text{ MHz}}$  = 3.5 µs

Number of bus cycles =  $3.5 \ \mu s \ x \ 8 \ MHz = 28 \ cycles$ 

### NOTE

The ADCK frequency must be between  $f_{ADCK}$  minimum and  $f_{ADCK}$  maximum to meet ADC specifications.

MC9S08SH8 MCU Series Data Sheet, Rev. 3



Field	Description
1	<b>OSC Initialization</b> — If the external reference clock is selected by ERCLKEN or by the ICS being in FEE, FBE, or FBELP mode, and if EREFS is set, then this bit is set after the initialization cycles of the external oscillator clock have completed. This bit is only cleared when either ERCLKEN or EREFS are cleared.
0	<b>ICS Fine Trim</b> — The FTRIM bit controls the smallest adjustment of the internal reference clock frequency. Setting FTRIM will increase the period and clearing FTRIM will decrease the period by the smallest amount possible.

Table 10-5. ICS Status and Control Register Field Descriptions (continued)

# 10.4 Functional Description

## 10.4.1 Operational Modes



Figure 10-7. Clock Switching Modes

The seven states of the ICS are shown as a state diagram and are described below. The arrows indicate the allowed movements between the states.

## 10.4.1.1 FLL Engaged Internal (FEI)

FLL engaged internal (FEI) is the default mode of operation and is entered when all the following conditions occur:



#### Chapter 11 Inter-Integrated Circuit (S08IICV2)

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.

## 11.4.1.7 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 11-10). 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.



### 11.4.1.8 Handshaking

The clock synchronization mechanism can be used as a handshake in data transfer. Slave devices may hold the SCL low after completion of one byte transfer (9 bits). In such a case, it halts the bus clock and forces the master clock into wait states until the slave releases the SCL line.

### 11.4.1.9 Clock Stretching

The clock synchronization mechanism can be used by slaves to slow down the bit rate of a transfer. After the master has driven SCL low the slave can drive SCL low for the required period and then release it. If the slave SCL low period is greater than the master SCL low period then the resulting SCL bus signal low period is stretched.



Chapter 12 Modulo Timer (S08MTIMV1)

# 12.3.3 MTIM Counter Register (MTIMCNT)

MTIMCNT is the read-only value of the current MTIM count of the 8-bit counter.



Figure 12-6. MTIM Counter Register



Field	Description
7:0 COUNT	<b>MTIM Count</b> — These eight read-only bits contain the current value of the 8-bit counter. Writes have no effect to this register. Reset clears the count to \$00.

# 12.3.4 MTIM Modulo Register (MTIMMOD)



Figure 12-7. MTIM Modulo Register

### Table 12-5. MTIM Modulo Register Field Descriptions

Field	Description
7:0 MOD	<b>MTIM Modulo</b> — These eight read/write bits contain the modulo value used to reset the count and set TOF. A value of \$00 puts the MTIM in free-running mode. Writing to MTIMMOD resets the COUNT to \$00 and clears TOF. Reset sets the modulo to \$00.



Chapter 12 Modulo Timer (S08MTIMV1)

## 12.4.1 MTIM Operation Example

This section shows an example of the MTIM operation as the counter reaches a matching value from the modulo register.

selected clock source								
MTIM clock (PS=%0010)			<b></b>		<b></b>	<b></b>		
MTIMCNT	\$A7	\$A8	\$A9	\$AA	\$00	\$01		
TOF								
MTIMMOD:	\$AA							

### Figure 12-8. MTIM counter overfl w example

In the example of Figure 12-8, the selected clock source could be any of the five possible choices. The prescaler is set to PS = %0010 or divide-by-4. The modulo value in the MTIMMOD register is set to \$AA. When the counter, MTIMCNT, reaches the modulo value of \$AA, the counter overflows to \$00 and continues counting. The timer overflow flag, TOF, sets when the counter value changes from \$AA to \$00. An MTIM overflow interrupt is generated when TOF is set, if TOIE = 1.



message characters. At the end of a message, or at the beginning of the next message, all receivers automatically force RWU to 0 so all receivers wake up in time to look at the first character(s) of the next message.

### 14.3.3.2.1 Idle-Line Wakeup

When WAKE = 0, the receiver is configured for idle-line wakeup. In this mode, RWU is cleared automatically when the receiver detects a full character time of the idle-line level. The M control bit selects 8-bit or 9-bit data mode that determines how many bit times of idle are needed to constitute a full character time (10 or 11 bit times because of the start and stop bits).

When RWU is one and RWUID is zero, the idle condition that wakes up the receiver does not set the IDLE flag. The receiver wakes up and waits for the first data character of the next message which will set the RDRF flag and generate an interrupt if enabled. When RWUID is one, any idle condition sets the IDLE flag and generates an interrupt if enabled, regardless of whether RWU is zero or one.

The idle-line type (ILT) control bit selects one of two ways to detect an idle line. When ILT = 0, the idle bit counter starts after the start bit so the stop bit and any logic 1s at the end of a character count toward the full character time of idle. When ILT = 1, the idle bit counter does not start until after a stop bit time, so the idle detection is not affected by the data in the last character of the previous message.

### 14.3.3.2.2 Address-Mark Wakeup

When WAKE = 1, the receiver is configured for address-mark wakeup. In this mode, RWU is cleared automatically when the receiver detects a logic 1 in the most significant bit of a received character (eighth bit in M = 0 mode and ninth bit in M = 1 mode).

Address-mark wakeup allows messages to contain idle characters but requires that the MSB be reserved for use in address frames. The logic 1 MSB of an address frame clears the RWU bit before the stop bit is received and sets the RDRF flag. In this case the character with the MSB set is received even though the receiver was sleeping during most of this character time.

## 14.3.4 Interrupts and Status Flags

The SCI system has three separate interrupt vectors to reduce the amount of software needed to isolate the cause of the interrupt. One interrupt vector is associated with the transmitter for TDRE and TC events. Another interrupt vector is associated with the receiver for RDRF, IDLE, RXEDGIF and LBKDIF events, and a third vector is used for OR, NF, FE, and PF error conditions. Each of these ten interrupt sources can be separately masked by local interrupt enable masks. The flags can still be polled by software when the local masks are cleared to disable generation of hardware interrupt requests.

The SCI transmitter has two status flags that optionally can generate hardware interrupt requests. Transmit data register empty (TDRE) indicates when there is room in the transmit data buffer to write another transmit character to SCIxD. If the transmit interrupt enable (TIE) bit is set, a hardware interrupt will be requested whenever TDRE = 1. Transmit complete (TC) indicates that the transmitter is finished transmitting all data, preamble, and break characters and is idle with TxD at the inactive level. This flag is often used in systems with modems to determine when it is safe to turn off the modem. If the transmit complete interrupt enable (TCIE) bit is set, a hardware TC = 1.



Chapter 15 Serial Peripheral Interface (S08SPIV3)



Figure 15-3. SPI Module Block Diagram

## 15.1.3 SPI Baud Rate Generation

As shown in Figure 15-4, the clock source for the SPI baud rate generator is the bus clock. The three prescale bits (SPPR2:SPPR1:SPPR0) choose a prescale divisor of 1, 2, 3, 4, 5, 6, 7, or 8. The three rate select bits (SPR2:SPR1:SPR0) divide the output of the prescaler stage by 2, 4, 8, 16, 32, 64, 128, or 256 to get the internal SPI master mode bit-rate clock.



Chapter 15 Serial Peripheral Interface (S08SPIV3)

# 15.5 Functional Description

An SPI transfer is initiated by checking for the SPI transmit buffer empty flag (SPTEF = 1) and then writing a byte of data to the SPI data register (SPID) in the master SPI device. When the SPI shift register is available, this byte of data is moved from the transmit data buffer to the shifter, SPTEF is set to indicate there is room in the buffer to queue another transmit character if desired, and the SPI serial transfer starts.

During the SPI transfer, data is sampled (read) on the MISO pin at one SPSCK edge and shifted, changing the bit value on the MOSI pin, one-half SPSCK cycle later. After eight SPSCK cycles, the data that was in the shift register of the master has been shifted out the MOSI pin to the slave while eight bits of data were shifted in the MISO pin into the master's shift register. At the end of this transfer, the received data byte is moved from the shifter into the receive data buffer and SPRF is set to indicate the data can be read by reading SPID. If another byte of data is waiting in the transmit buffer at the end of a transfer, it is moved into the shifter, SPTEF is set, and a new transfer is started.

Normally, SPI data is transferred most significant bit (MSB) first. If the least significant bit first enable (LSBFE) bit is set, SPI data is shifted LSB first.

When the SPI is configured as a slave, its  $\overline{SS}$  pin must be driven low before a transfer starts and  $\overline{SS}$  must stay low throughout the transfer. If a clock format where CPHA = 0 is selected,  $\overline{SS}$  must be driven to a logic 1 between successive transfers. If CPHA = 1,  $\overline{SS}$  may remain low between successive transfers. See Section 15.5.1, "SPI Clock Formats" for more details.

Because the transmitter and receiver are double buffered, a second byte, in addition to the byte currently being shifted out, can be queued into the transmit data buffer, and a previously received character can be in the receive data buffer while a new character is being shifted in. The SPTEF flag indicates when the transmit buffer has room for a new character. The SPRF flag indicates when a received character is available in the receive data buffer. The received character must be read out of the receive buffer (read SPID) before the next transfer is finished or a receive overrun error results.

In the case of a receive overrun, the new data is lost because the receive buffer still held the previous character and was not ready to accept the new data. There is no indication for such an overrun condition so the application system designer must ensure that previous data has been read from the receive buffer before a new transfer is initiated.

# 15.5.1 SPI Clock Formats

To accommodate a wide variety of synchronous serial peripherals from different manufacturers, the SPI system has a clock polarity (CPOL) bit and a clock phase (CPHA) control bit to select one of four clock formats for data transfers. CPOL selectively inserts an inverter in series with the clock. CPHA chooses between two different clock phase relationships between the clock and data.

Figure 15-10 shows the clock formats when CPHA = 1. At the top of the figure, the eight bit times are shown for reference with bit 1 starting at the first SPSCK edge and bit 8 ending one-half SPSCK cycle after the sixteenth SPSCK edge. The MSB first and LSB first lines show the order of SPI data bits depending on the setting in LSBFE. Both variations of SPSCK polarity are shown, but only one of these waveforms applies for a specific transfer, depending on the value in CPOL. The SAMPLE IN waveform applies to the MOSI input of a slave or the MISO input of a master. The MOSI waveform applies to the MOSI output



Chapter 16 Timer/PWM Module (S08TPMV3)



Figure 16-2. TPM Block Diagram



• 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 17-1. BDM Tool Connector

## 17.2.1 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 17.2.2, "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 17.2.2, "Communication Details," for more detail.



Chapter 17 Development Support

# 17.2.3 BDC Commands

BDC commands are sent serially from a host computer to the BKGD pin of the target HCS08 MCU. All commands and data are sent MSB-first using a custom BDC communications protocol. Active background mode commands require that the target MCU is currently in the active background mode while non-intrusive commands may be issued at any time whether the target MCU is in active background mode or running a user application program.

Table 17-1 shows all HCS08 BDC commands, a shorthand description of their coding structure, and the meaning of each command.

### **Coding Structure Nomenclature**

This nomenclature is used in Table 17-1 to describe the coding structure of the BDC commands.

Commands begin with an 8-bit hexadecimal command code in the host-to-target direction (most significant bit first)

- / = separates parts of the command
- d = delay 16 target BDC clock cycles
- AAAA = a 16-bit address in the host-to-target direction
  - RD = 8 bits of read data in the target-to-host direction
  - WD = 8 bits of write data in the host-to-target direction
- RD16 = 16 bits of read data in the target-to-host direction
- WD16 = 16 bits of write data in the host-to-target direction
  - SS = the contents of BDCSCR in the target-to-host direction (STATUS)
  - CC = 8 bits of write data for BDCSCR in the host-to-target direction (CONTROL)
- RBKP = 16 bits of read data in the target-to-host direction (from BDCBKPT breakpoint register)
- WBKP = 16 bits of write data in the host-to-target direction (for BDCBKPT breakpoint register)



#### **Appendix A Electrical Characteristics**

The average chip-junction temperature  $(T_J)$  in °C can be obtained from:

$$T_{J} = T_{A} + (P_{D} \times \theta_{JA})$$
 Eqn. A-1

where:

$$\begin{split} T_A &= \text{Ambient temperature, }^\circ\text{C}\\ \theta_{JA} &= \text{Package thermal resistance, junction-to-ambient, }^\circ\text{C/W}\\ P_D &= P_{int} + P_{I/O}\\ P_{int} &= I_{DD} \times V_{DD}, \text{Watts} \ \text{membrane chip internal power}\\ P_{I/O} &= \text{Power dissipation on input and output pins} \ \text{membrane user determined} \end{split}$$

For most applications,  $P_{I/O} \ll P_{int}$  and can be neglected. An approximate relationship between  $P_D$  and  $T_J$  (if  $P_{I/O}$  is neglected) is:

$$P_{D} = K \div (T_{J} + 273^{\circ}C)$$
 Eqn. A-2

Solving Equation A-1 and Equation A-2 for K gives:

$$K = P_D \times (T_A + 273^{\circ}C) + \theta_{JA} \times (P_D)^2 \qquad Eqn. A-3$$

where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring  $P_D$  (at equilibrium) for a known  $T_A$ . Using this value of K, the values of  $P_D$  and  $T_J$  can be obtained by solving Equation A-1 and Equation A-2 iteratively for any value of  $T_A$ .



## A.12.2 TPM/MTIM Module Timing

Synchronizer circuits determine the shortest input pulses that can be recognized or the fastest clock that can be used as the optional external source to the timer counter. These synchronizers operate from the current bus rate clock.

Num	с	Rating	Symbol	Min	Max	Unit
1	_	External clock frequency (1/t <sub>TCLK</sub> )	f <sub>TCLK</sub>	dc	f <sub>Bus</sub> /4	MHz
2	—	External clock period	t <sub>TCLK</sub>	4	_	t <sub>cyc</sub>
3	_	External clock high time	t <sub>clkh</sub>	1.5	_	t <sub>cyc</sub>
4	_	External clock low time	t <sub>clkl</sub>	1.5	_	t <sub>cyc</sub>
5	_	Input capture pulse width	t <sub>ICPW</sub>	1.5	—	t <sub>cyc</sub>

#### Table A-14. TPM Input Timing



Figure A-12. Timer External Clock



Figure A-13. Timer Input Capture Pulse