Freescale Semiconductor - MC9S08GT60CFD Datasheet

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	I²C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	39
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-VFQFN Exposed Pad
Supplier Device Package	48-QFN (7x7)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s08gt60cfd

Section Number Title Page

Chapter 8
Central Processor Unit (CPU)

Chapter 9
Keyboard Interrupt (KBI) Module

Chapter 15 Development Support

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

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
$0028	**SPI1C1**	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
$0029	**SPI1C2**	0	0	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
$002A	**SPI1BR**	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
$002B	**SPI1S**	SPRF	0	SPTEF	MODF	0	0	0	0
$002C	Reserved	0	0	0	0	0	0	0	0
$002D	**SPI1D**	Bit 7	6	5	4	3	2	1	Bit 0
$002E	Reserved	0	0	0	0	0	0	0	0
$002F	Reserved	0	0	0	0	0	0	0	0
$0030	**TPM1SC**	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
$0031	**TPM1CNTH**	Bit 15	14	13	12	11	10	9	Bit 8
$0032	**TPM1CNTL**	Bit 7	6	5	4	3	2	1	Bit 0
$0033	**TPM1MODH**	Bit 15	14	13	12	11	10	9	Bit 8
$0034	**TPM1MODL**	Bit 7	6	5	4	3	2	1	Bit 0
$0035	**TPM1C0SC**	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
$0036	**TPM1C0VH**	Bit 15	14	13	12	11	10	9	Bit 8
$0037	**TPM1C0VL**	Bit 7	6	5	4	3	2	1	Bit 0
$0038	**TPM1C1SC**	CH1F	CH1IE	MS1B	MS1A	ELS1B	ELS1A	0	0
$0039	**TPM1C1VH**	Bit 15	14	13	12	11	10	9	Bit 8
$003A	**TPM1C1VL**	Bit 7	6	5	4	3	2	1	Bit 0
$003B	**TPM1C2SC**	CH2F	CH2IE	MS2B	MS2A	ELS2B	ELS2A	0	0
$003C	**TPM1C2VH**	Bit 15	14	13	12	11	10	9	Bit 8
$003D	**TPM1C2VL**	Bit 7	6	5	4	3	2	1	Bit 0
$003E– $003F	Reserved	— —	— —	— —	— —	— —	— —	— —	— —
$0040	**PTFD**	PTFD7	PTFD6	PTFD5	PTFD4	PTFD3	PTFD2	PTFD1	PTFD0
$0041	**PTFPE**	PTFPE7	PTFPE6	PTFPE5	PTFPE4	PTFPE3	PTFPE2	PTFPE1	PTFPE0
$0042	**PTFSE**	PTFSE7	PTFSE6	PTFSE5	PTFSE4	PTFSE3	PTFSE2	PTFSE1	PTFSE0
$0043	**PTFDD**	PTFDD7	PTFDD6	PTFDD5	PTFDD4	PTFDD3	PTFDD2	PTFDD1	PTFDD0
$0044	**PTGD**	PTGD7	PTGD6	PTGD5	PTGD4	PTGD3	PTGD2	PTGD1	PTGD0
$0045	**PTGPE**	PTGPE7	PTGPE6	PTGPE5	PTGPE4	PTGPE3	PTGPE2	PTGPE1	PTGPE0
$0046	**PTGSE**	PTGSE7	PTGSE6	PTGSE5	PTGSE4	PTGSE3	PTGSE2	PTGSE1	PTGSE0
$0047	**PTGDD**	PTGDD7	PTGDD6	PTGDD5	PTGDD4	PTGDD3	PTGDD2	PTGDD1	PTGDD0
$0048	**ICGC1**	0	RANGE	REFS	CLKS		OSCSTEN	0*	0
$0049	**ICGC2**	LOLRE	MFD			LOCRE	RFD		
$004A	**ICGS1**	CLKST		REFST	LOLS	LOCK	LOCS	ERCS	ICGIF
$004B	**ICGS2**	0	0	0	0	0	0	0	DCOS
$004C	**ICGFLTU**	0	0	0	0	FLT			
$004D	**ICGFLTL**	FLT							
$004E	**ICGTRM**	TRIM							

* This bit is reserved for Freescale Semiconductor internal use only. Always write a 0 to this bit.

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

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
$004F	Reserved	0	0	0	0	0	0	0	0
$0050	ATD1C	ATDPU	DJM	RES8	SGN	PRS			
$0051	ATD1SC	CCF	ATDIE	ATDCO	ATDCH				
$0052	ATD1RH	Bit 7	6	5	4	3	2	1	Bit 0
$0053	ATD1RL	Bit 7	6	5	4	3	2	1	Bit 0
$0054	ATD1PE	ATDPE7	ATDPE6	ATDPE5	ATDPE4	ATDPE3	ATDPE2	ATDPE1	ATDPE0
$0055– $0057	Reserved	— —	— —	— —	— —	— —	— —	— —	— —
$0058	IIC1A	ADDR							0
$0059	IIC1F	MULT		ICR					
$005A	IIC1C	IICEN	IICIE	MST	TX	TXAK	RSTA	0	0
$005B	IIC1S	TCF	IAAS	BUSY	ARBL	0	SRW	IICIF	RXAK
$005C	IIC1D	DATA							
$005D– $005F	Reserved	— —	— —	— —	— —	— —	— —	— —	— —
$0060	TPM2SC	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
$0061	TPM2CNTH	Bit 15	14	13	12	11	10	9	Bit 8
$0062	TPM2CNTL	Bit 7	6	5	4	3	2	1	Bit 0
$0063	TPM2MODH	Bit 15	14	13	12	11	10	9	Bit 8
$0064	TPM2MODL	Bit 7	6	5	4	3	2	1	Bit 0
$0065	TPM2C0SC	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
$0066	TPM2C0VH	Bit 15	14	13	12	11	10	9	Bit 8
$0067	TPM2C0VL	Bit 7	6	5	4	3	2	1	Bit 0
$0068	TPM2C1SC	CH1F	CH1IE	MS1B	MS1A	ELS1B	ELS1A	0	0
$0069	TPM2C1VH	Bit 15	14	13	12	11	10	9	Bit 8
$006A	TPM2C1VL	Bit 7	6	5	4	3	2	1	Bit 0
$006B	TPM2C2SC	CH2F	CH2IE	MS2B	MS2A	ELS2B	ELS2A	0	0
$006C	TPM2C2VH	Bit 15	14	13	12	11	10	9	Bit 8
$006D	TPM2C2VL	Bit 7	6	5	4	3	2	1	Bit 0
$006E	TPM2C3SC	CH3F	CH3IE	MS3B	MS3A	ELS3B	ELS3A	0	0
$006F	TPM2C3VH	Bit 15	14	13	12	11	10	9	Bit 8
$0070	TPM2C3VL	Bit 7	6	5	4	3	2	1	Bit 0
$0071	TPM2C4SC	CH4F	CH4IE	MS4B	MS4A	ELS4B	ELS4A	0	0
$0072	TPM2C4VH	Bit 15	14	13	12	11	10	9	Bit 8
$0073	TPM2C4VL	Bit 7	6	5	4	3	2	1	Bit 0
$0074– $007F	Reserved	— —	— —	— —	— —	— —	— —	— —	— —

Each of these sources, with the exception of the background debug forced reset, has an associated bit in the system reset status register. Whenever the MCU enters reset, the internal clock generator (ICG) module switches to self-clocked mode with the frequency of f_{Self_reset} selected. The reset pin is driven low for 34 internal bus cycles where the internal bus frequency is half the ICG frequency. After the 34 cycles are completed, the pin is released and will be pulled up by the internal pullup resistor, unless it is held low externally. After the pin is released, it is sampled after another 38 cycles to determine whether the reset pin is the cause of the MCU reset.

5.4 Computer Operating Properly (COP) Watchdog

The COP watchdog is intended to force a system reset when the application software fails to execute as expected. To prevent a system reset from the COP timer (when it is enabled), application software must reset the COP timer periodically. If the application program gets lost and fails to reset the COP before it times out, a system reset is generated to force the system back to a known starting point. The COP watchdog is enabled by the COPE bit in SOPT (see Section 5.8.4, “System Options Register (SOPT)” for additional information). The COP timer is reset by writing any value to the address of SRS. This write does not affect the data in the read-only SRS. Instead, the act of writing to this address is decoded and sends a reset signal to the COP timer.

After any reset, the COP timer is enabled. This provides a reliable way to detect code that is not executing as intended. If the COP watchdog is not used in an application, it can be disabled by clearing the COPE bit in the write-once SOPT register. Also, the COPT bit can be used to choose one of two timeout periods (2^{18} or 2^{13} cycles of the bus rate clock). Even if the application will use the reset default settings in COPE and COPT, the user should still write to write-once SOPT during reset initialization to lock in the settings. That way, they cannot be changed accidentally if the application program gets lost.

The write to SRS that services (clears) the COP timer should not be placed in an interrupt service routine (ISR) because the ISR could continue to be executed periodically even if the main application program fails.

When the MCU is in active background mode, the COP timer is temporarily disabled.

5.5 Interrupts

Interrupts provide a way to save the current CPU status and registers, execute an interrupt service routine (ISR), and then restore the CPU status so processing resumes where it left off before the interrupt. Other than the software interrupt (SWI), which is a program instruction, interrupts are caused by hardware events such as an edge on the IRQ pin or a timer-overflow event. The debug module can also generate an SWI under certain circumstances.

If an event occurs in an enabled interrupt source, an associated read-only status flag will become set. The CPU will not respond until and unless the local interrupt enable is set to 1 to enable the interrupt. The I bit in the CCR is 0 to allow interrupts. The global interrupt mask (I bit) in the CCR is initially set after reset which masks (prevents) all maskable interrupt sources. The user program initializes the stack pointer and performs other system setup before clearing the I bit to allow the CPU to respond to interrupts.

PTCD

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PTCD7	PTCD6	PTCD5	PTCD4	PTCD3	PTCD2	PTCD1	PTCD0
Reset:	0	0	0	0	0	0	0	0

PTCPE

Read: Write:	PTCPE7	PTCPE6	PTCPE5	PTCPE4	PTCPE3	PTCPE2	PTCPE1	PTCPE0
Reset:	0	0	0	0	0	0	0	0

PTCSE

Read: Write:	PTCSE7	PTCSE6	PTCSE5	PTCSE4	PTCSE3	PTCSE2	PTCSE1	PTCSE0
Reset:	0	0	0	0	0	0	0	0

PTCDD

Read: Write:	PTCDD7	PTCDD6	PTCDD5	PTCDD4	PTCDD3	PTCDD2	PTCDD1	PTCDD0
Reset:	0	0	0	0	0	0	0	0

Figure 6-11. Port C Registers

PTCDn — Port C Data Register Bit n (n = 0–7)

For port C pins that are inputs, reads return the logic level on the pin. For port C pins that are configured as outputs, reads return the last value written to this register.

Writes are latched into all bits of this register. For port C pins that are configured as outputs, the logic level is driven out the corresponding MCU pin.

Reset forces PTCD to all 0s, but these 0s are not driven out the corresponding pins because reset also configures all port pins as high-impedance inputs with pullups disabled.

PTCPEn — Pullup Enable for Port C Bit n (n = 0–7)

For port C pins that are inputs, these read/write control bits determine whether internal pullup devices are enabled. For port C pins that are configured as outputs, these bits are ignored and the internal pullup devices are disabled.

1 = Internal pullup device enabled.
0 = Internal pullup device disabled.

PTCSEn — Slew Rate Control Enable for Port C Bit n (n = 0–7)

For port C pins that are outputs, these read/write control bits determine whether the slew rate controlled outputs are enabled. For port B pins that are configured as inputs, these bits are ignored.

1 = Slew rate control enabled.
0 = Slew rate control disabled.

PTEDDn — Data Direction for Port E Bit n (n = 0–7)

These read/write bits control the direction of port E pins and what is read for PTED reads.
1 = Output driver enabled for port E bit n and PTED reads return the contents of PTEDn.
0 = Input (output driver disabled) and reads return the pin value.

6.6.6 Port F Registers (PTFD, PTFPE, PTFSE, and PTFDD)

Port F includes eight general-purpose I/O pins that are not shared with any peripheral module. Port F pins used as general-purpose I/O pins are controlled by the port F data (PTFD), data direction (PTFDD), pullup enable (PTFPE), and slew rate control (PTFSE) registers.

PTFD		Bit 7	6	5	4	3	2	1	Bit 0
	Read: Write:	PTFD7	PTFD6	PTFD5	PTFD4	PTFD3	PTFD2	PTFD1	PTFD0
	Reset:	0	0	0	0	0	0	0	0
PTFPE									
	Read: Write:	PTFPE7	PTFPE6	PTFPE5	PTFPE4	PTFPE3	PTFPE2	PTFPE1	PTFPE0
	Reset:	0	0	0	0	0	0	0	0
PTFSE									
	Read: Write:	PTFSE7	PTFSE6	PTFSE5	PTFSE4	PTFSE3	PTFSE2	PTFSE1	PTFSE0
	Reset:	0	0	0	0	0	0	0	0
PTFDD									
	Read: Write:	PTFDD7	PTFDD6	PTFDD5	PTFDD4	PTFDD3	PTFDD2	PTFDD1	PTFDD0
	Reset:	0	0	0	0	0	0	0	0

Figure 6-14. Port F Registers

PTFDn — Port PTF Data Register Bit n (n = 0–7)

For port F pins that are inputs, reads return the logic level on the pin. For port F pins that are configured as outputs, reads return the last value written to this register.

Writes are latched into all bits of this register. For port F pins that are configured as outputs, the logic level is driven out the corresponding MCU pin.

Reset forces PTFD to all 0s, but these 0s are not driven out the corresponding pins because reset also configures all port pins as high-impedance inputs with pullups disabled.

ICGC2 = $30 (%00110000)

Bit 7	LOLRE	0	Generates an interrupt request on loss of lock
Bit 6:4	MFD	011	Sets the MFD multiplication factor to 10
Bit 3	LOCRE	0	Generates an interrupt request on loss of clock
Bit 2:0	RFD	000	Sets the RFD division factor to ÷1

ICGS1 = $xx

This is read only except for clearing interrupt flag

ICGS2 = $xx

This is read only. Should read DCOS before performing any time critical tasks

ICGFLTLU/L = $xx

Not used in this example

ICGTRM

Not used in this example

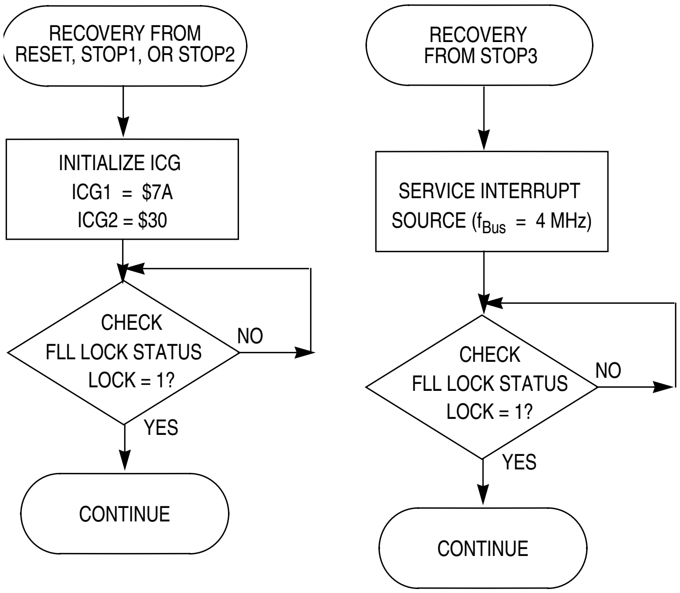

RECOVERY FROM RESET, STOP1, OR STOP2
INITIALIZE ICG
ICG1 = $7A
ICG2 = $30
CHECK FLL LOCK STATUS LOCK = 1?
NO
YES
CONTINUE
RECOVERY FROM STOP3
SERVICE INTERRUPT SOURCE (fBus = 4 MHz)
CHECK FLL LOCK STATUS LOCK = 1?
NO
YES
CONTINUE

Figure 7-9. ICG Initialization and Stop Recovery for Example #2

7.4.4 Example #3: No External Crystal Connection, 5.4 MHz Bus Frequency

In this example, the FLL will be used (in FEI mode) to multiply the internal 243 kHz (approximate) reference clock up to 10.8 MHz to achieve 5.4 MHz bus frequency. This system will also use the trim function to fine tune the frequency based on an external reference signal.

After the MCU is released from reset, the ICG is in self-clocked mode (SCM) and supplies approximately 8 MHz on ICGOUT which corresponds to a 4 MHz bus frequency (f_{Bus}).

Table 8-1. HCS08 Instruction Set Summary (Sheet 5 of 7)

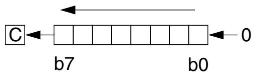
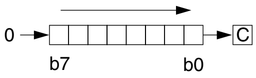
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Bus Cycles[1]
			V	H	I	N	Z	C				
LDA *#opr8i* LDA *opr8a* LDA *opr16a* LDA *oprx16*,X LDA *oprx8*,X LDA ,X LDA *oprx16*,SP LDA *oprx8*,SP	Load Accumulator from Memory	A ← (M)	0	–	–			–	IMM DIR EXT IX2 IX1 IX SP2 SP1	A6 B6 C6 D6 E6 F6 9ED6 9EE6	ii dd hh ll ee ff ff ee ff ff	2 3 4 4 3 3 5 4
LDHX *#opr16i* LDHX *opr8a* LDHX *opr16a* LDHX ,X LDHX *oprx16*,X LDHX *oprx8*,X LDHX *oprx8*,SP	Load Index Register (H:X) from Memory	H:X ← (M:M + $0001)	0	–	–			–	IMM DIR EXT IX IX2 IX1 SP1	45 55 32 9EAE 9EBE 9ECE 9EFE	jj kk dd hh ll ee ff ff ff	3 4 5 5 6 5 5
LDX *#opr8i* LDX *opr8a* LDX *opr16a* LDX *oprx16*,X LDX *oprx8*,X LDX ,X LDX *oprx16*,SP LDX *oprx8*,SP	Load X (Index Register Low) from Memory	X ← (M)	0	–	–			–	IMM DIR EXT IX2 IX1 IX SP2 SP1	AE BE CE DE EE FE 9EDE 9EEE	ii dd hh ll ee ff ff ee ff ff	2 3 4 4 3 3 5 4
LSL *opr8a* LSLA LSLX LSL *oprx8*,X LSL ,X LSL *oprx8*,SP	Logical Shift Left (Same as ASL)			–	–				DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff	5 1 1 5 4 6
LSR *opr8a* LSRA LSR*X* LSR *oprx8*,X LSR ,X LSR *oprx8*,SP	Logical Shift Right			–	–	0			DIR INH INH IX1 IX SP1	34 44 54 64 74 9E64	dd ff ff	5 1 1 5 4 6
MOV *opr8a*,*opr8a* MOV *opr8a*,X+ MOV *#opr8i*,*opr8a* MOV ,X+,*opr8a*	Move	$(M)_{destination} \leftarrow (M)_{source}$ H:X ← (H:X) + $0001 in IX+/DIR and DIR/IX+ Modes	0	–	–			–	DIR/DIR DIR/IX+ IMM/DIR IX+/DIR	4E 5E 6E 7E	dd dd dd ii dd dd	5 5 4 5
MUL	Unsigned multiply	X:A ← (X) × (A)	–	0	–	–	–	0	INH	42		5
NEG *opr8a* NEGA NEGX NEG *oprx8*,X NEG ,X NEG *oprx8*,SP	Negate (Two's Complement)	M ← – (M) = $00 – (M) A ← – (A) = $00 – (A) X ← – (X) = $00 – (X) M ← – (M) = $00 – (M) M ← – (M) = $00 – (M) M ← – (M) = $00 – (M)		–	–				DIR INH INH IX1 IX SP1	30 40 50 60 70 9E60	dd ff ff	5 1 1 5 4 6
NOP	No Operation	Uses 1 Bus Cycle	–	–	–	–	–	–	INH	9D		1
NSA	Nibble Swap Accumulator	A ← (A[3:0]:A[7:4])	–	–	–	–	–	–	INH	62		1
ORA *#opr8i* ORA *opr8a* ORA *opr16a* ORA *oprx16*,X ORA *oprx8*,X ORA ,X ORA *oprx16*,SP ORA *oprx8*,SP	Inclusive OR Accumulator and Memory	A ← (A) | (M)	0	–	–			–	IMM DIR EXT IX2 IX1 IX SP2 SP1	AA BA CA DA EA FA 9EDA 9EEA	ii dd hh ll ee ff ff ee ff ff	2 3 4 4 3 3 5 4
PSHA	Push Accumulator onto Stack	Push (A); SP ← (SP) – $0001	–	–	–	–	–	–	INH	87		2
PSHH	Push H (Index Register High) onto Stack	Push (H); SP ← (SP) – $0001	–	–	–	–	–	–	INH	8B		2
PSHX	Push X (Index Register Low) onto Stack	Push (X); SP ← (SP) – $0001	–	–	–	–	–	–	INH	89		2

Table 8-2. Opcode Map (Sheet 2 of 2)

Bit-Manipulation		Branch	Read-Modify-Write					Control		Register/Memory					
						9E60 6 NEG 3 SP1							9ED0 5 SUB 4 SP2	9EE0 4 SUB 3 SP1	
						9E61 6 CBEQ 4 SP1							9ED1 5 CMP 4 SP2	9EE1 4 CMP 3 SP1	
													9ED2 5 SBC 4 SP2	9EE2 4 SBC 3 SP1	
						9E63 6 COM 3 SP1							9ED3 5 CPX 4 SP2	9EE3 4 CPX 3 SP1	9EF3 6 CPHX 3 SP1
						9E64 6 LSR 3 SP1							9ED4 5 AND 4 SP2	9EE4 4 AND 3 SP1	
													9ED5 5 BIT 4 SP2	9EE5 4 BIT 3 SP1	
						9E66 6 ROR 3 SP1							9ED6 5 LDA 4 SP2	9EE6 4 LDA 3 SP1	
						9E67 6 ASR 3 SP1							9ED7 5 STA 4 SP2	9EE7 4 STA 3 SP1	
						9E68 6 LSL 3 SP1							9ED8 5 EOR 4 SP2	9EE8 4 EOR 3 SP1	
						9E69 6 ROL 3 SP1							9ED9 5 ADC 4 SP2	9EE9 4 ADC 3 SP1	
						9E6A 6 DEC 3 SP1							9EDA 5 ORA 4 SP2	9EEA 4 ORA 3 SP1	
						9E6B 8 DBNZ 4 SP1							9EDB 5 ADD 4 SP2	9EEB 4 ADD 3 SP1	
						9E6C 6 INC 3 SP1									
						9E6D 5 TST 3 SP1									
										9EAE 5 LDHX 2 IX	9EBE 6 LDHX 4 IX2	9ECE 5 LDHX 3 IX1	9EDE 5 LDX 4 SP2	9EEE 4 LDX 3 SP1	9EFE 5 LDHX 3 SP1
						9E6F 6 CLR 3 SP1							9EDF 5 STX 4 SP2	9EEF 4 STX 3 SP1	9EFF 5 STHX 3 SP1

INH	**Inherent**	**REL**	**Relative**	**SP1**	**Stack Pointer, 8-Bit Offset**
IMM	**Immediate**	**IX**	**Indexed, No Offset**	**SP2**	**Stack Pointer, 16-Bit Offset**
DIR	**Direct**	**IX1**	**Indexed, 8-Bit Offset**	**IX+**	**Indexed, No Offset with Post Increment**
EXT	**Extended**	**IX2**	**Indexed, 16-Bit Offset**		
DD	**DIR to DIR**	**IMD**	**IMM to DIR**	**IX1+**	**Indexed, 1-Byte Offset with Post Increment**
IX+D	**IX+ to DIR**	**DIX+**	**DIR to IX+**		

Note: All Sheet 2 Opcodes are Preceded by the Page 2 Prebyte (9E)

Prebyte (9E) and Opcode in Hexadecimal	9E60 6	**HCS08 Cycles**
	NEG	**Instruction Mnemonic**
Number of Bytes	3 SP1	**Addressing Mode**

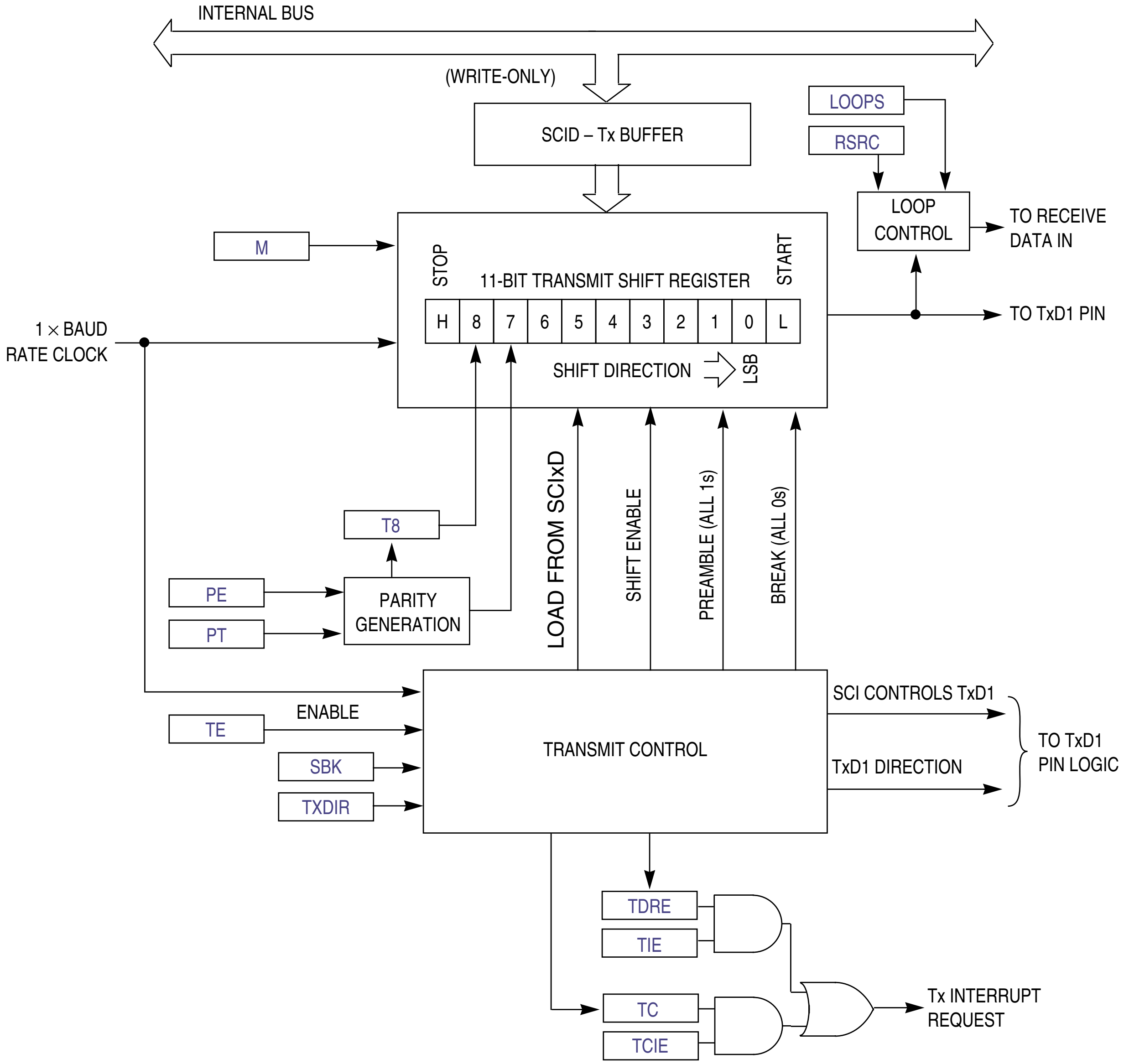


Figure 11-3. SCI Transmitter Block Diagram

The transmitter is enabled by setting the TE bit in SCIxC2. This queues a preamble character that is one full character frame of logic high. The transmitter then remains idle (TxD1 pin remains high) until data is available in the transmit data buffer. Programs store data into the transmit data buffer by writing to the SCI data register (SCIxD).

The central element of the SCI transmitter is the transmit shift register that is either 10 or 11 bits long depending on the setting in the M control bit. For the remainder of this section, we will assume M = 0, selecting the normal 8-bit data mode. In 8-bit data mode, the shift register holds a start bit, eight data bits, and a stop bit. When the transmit shift register is available for a new SCI character, the value waiting in the transmit data register is transferred to the shift register (synchronized with the baud rate clock) and the transmit data register empty (TDRE) status flag is set to indicate another character may be written to the transmit data buffer at SCIxD.

The most common uses of the SPI system include connecting simple shift registers for adding input or output ports or connecting small peripheral devices such as serial A/D or D/A converters. Although Figure 12-2 shows a system where data is exchanged between two MCUs, many practical systems involve simpler connections where data is unidirectionally transferred from the master MCU to a slave or from a slave to the master MCU.

12.2.2 SPI Module Block Diagram

Figure 12-3 is a block diagram of the SPI module. The central element of the SPI is the SPI shift register. Data is written to the double-buffered transmitter (write to SPI1D) and gets transferred to the SPI shift register at the start of a data transfer. After shifting in a byte of data, the data is transferred into the double-buffered receiver where it can be read (read from SPI1D). Pin multiplexing logic controls connections between MCU pins and the SPI module.

When the SPI is configured as a master, the clock output is routed to the SPSCK1 pin, the shifter output is routed to MOSI1, and the shifter input is routed from the MISO1 pin.

When the SPI is configured as a slave, the SPSCK1 pin is routed to the clock input of the SPI, the shifter output is routed to MISO1, and the shifter input is routed from the MOSI1 pin.

In the external SPI system, simply connect all SPSCK pins to each other, all MISO pins together, and all MOSI pins together. Peripheral devices often use slightly different names for these pins.

Table 12-2. SPI Baud Rate Prescaler Divisor

SPPR2:SPPR1:SPPR0	Prescaler Divisor
0:0:0	1
0:0:1	2
0:1:0	3
0:1:1	4
1:0:0	5
1:0:1	6
1:1:0	7
1:1:1	8

SPR2:SPR1:SPR0 — SPI Baud Rate Divisor

This 3-bit field selects one of eight divisors for the SPI baud rate divider as shown in Figure 12-3. The input to this divider comes from the SPI baud rate prescaler (see Figure 12-4). The output of this divider is the SPI bit rate clock for master mode.

Table 12-3. SPI Baud Rate Divisor

SPR2:SPR1:SPR0	Rate Divisor
0:0:0	2
0:0:1	4
0:1:0	8
0:1:1	16
1:0:0	32
1:0:1	64
1:1:0	128
1:1:1	256

13.2.1.1 START Signal

When the bus is free; i.e., no master device is engaging the bus (both SCL and SDA lines are at logical high), a master may initiate communication by sending a START signal. As shown in Figure 13-3, a START signal is defined as a high-to-low transition of SDA while SCL is high. This signal denotes the beginning of a new data transfer (each data transfer may contain several bytes of data) and brings all slaves out of their idle states.

13.2.1.2 Slave Address Transmission

The first byte of data transferred immediately after the START signal is the slave address transmitted by the master. This is a seven-bit calling address followed by a R/W bit. The R/W bit tells the slave the desired direction of data transfer.

1 = Read transfer, the slave transmits data to the master.
0 = Write transfer, the master transmits data to the slave.

Only the slave with a calling address that matches the one transmitted by the master will respond by sending back an acknowledge bit. This is done by pulling the SDA low at the 9th clock (see Figure 13-3).

No two slaves in the system may have the same address. If the IIC module is the master, it must not transmit an address that is equal to its own slave address. The IIC cannot be master and slave at the same time. However, if arbitration is lost during an address cycle, the IIC will revert to slave mode and operate correctly even if it is being addressed by another master.

13.2.1.3 Data Transfer

Before successful slave addressing is achieved, the data transfer can proceed byte-by-byte in a direction specified by the R/W bit sent by the calling master.

All transfers that come after an address cycle are referred to as data transfers, even if they carry sub-address information for the slave device

Each data byte is 8 bits long. Data may be changed only while SCL is low and must be held stable while SCL is high as shown in Figure 13-3. There is one clock pulse on SCL for each data bit, the MSB being transferred first. Each data byte is followed by a 9th (acknowledge) bit, which is signalled from the receiving device. An acknowledge is signalled by pulling the SDA low at the ninth clock. In summary, one complete data transfer needs nine clock pulses.

If the slave receiver does not acknowledge the master in the 9th bit time, the SDA line must be left high by the slave. The master interprets the failed acknowledge as an unsuccessful data transfer.

If the master receiver does not acknowledge the slave transmitter after a data byte transmission, the slave interprets this as an end of data transfer and releases the SDA line.

In either case, the data transfer is aborted and the master does one of two things:

- Relinquishes the bus by generating a STOP signal.
- Commences a new calling by generating a repeated START signal.

Note that the TX bit in IIC1C must correctly reflect the desired direction of transfer in master and slave modes for the transmission to begin. For instance, if the IIC is configured for master transmit but a master receive is desired, then reading the IIC1D will not initiate the receive.

Reading the IIC1D will return the last byte received while the IIC is configured in either master receive or slave receive modes. The IIC1D does not reflect every byte that is transmitted on the IIC bus, nor can software verify that a byte has been written to the IIC1D correctly by reading it back.

In master transmit mode, the first byte of data written to IIC1D following assertion of MST is used for the address transfer and should comprise of the calling address (in bit 7–bit 1) concatenated with the required R/W bit (in position bit 0).

Chapter 14 Analog-to-Digital Converter (ATD) Module

The MC9S08GB/GT provides one 8-channel analog-to-digital (ATD) module. The eight ATD channels share port B. Each channel individually can be configured for general-purpose I/O or for ATD functionality. All features of the ATD module as described in this section are available on the MC9S08GB/GT. Electrical parametric information for the ATD may be found in Appendix A, “Electrical Characteristics.”

15.4 On-Chip Debug System (DBG)

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

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

Most of the debug module's functions are used during development, and user programs rarely access any of the control and status registers for the debug module. The one exception is that the debug system can provide the means to implement a form of ROM patching. This topic is discussed in greater detail in Section 15.4.6, "Hardware Breakpoints."

15.4.1 Comparators A and B

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

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

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

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

15.4.2 Bus Capture Information and FIFO Operation

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

A.3 Thermal Characteristics

This section provides information about operating temperature range, power dissipation, and package thermal resistance. Power dissipation on I/O pins is usually small compared to the power dissipation in on-chip logic and voltage regulator circuits and it is user-determined rather than being controlled by the MCU design. In order to take $P_{I/O}$ into account in power calculations, determine the difference between actual pin voltage and V_{SS} or V_{DD} and multiply by the pin current for each I/O pin. Except in cases of unusually high pin current (heavy loads), the difference between pin voltage and V_{SS} or V_{DD} will be very small.

Table A-2. Thermal Characteristics

Rating	Symbol	Value	Unit	Temp. Code
Operating temperature range (packaged)	T_A	–40 to 85	°C	C
Thermal resistance 64-pin LQFP (GB60) 48-pin QFN (GT60) 42-pin SDIP (GT60) 44-pin QFP (GT60)	θ_{JA}[1, 2]	 65 82 57 118	°C/W	—

[1] Junction temperature is a function of die size, on-chip power dissipation, package thermal resistance, mounting site (board) temperature, ambient temperature, airflow, power dissipation of other components on the board, and board thermal resistance.

[2] Per SEMI G38-87 and JEDEC JESD51-2 with the single layer board horizontal. Single layer board is designed per JEDEC JESD51-3.

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

$$\mathbf{T_J = T_A + (P_D \times \theta_{JA})} \qquad \textit{Eqn. A-1}$$

where:

T_A = Ambient temperature, °C
θ_{JA} = Package thermal resistance, junction-to-ambient, °C/W
$P_D = P_{int} + P_{I/O}$
$P_{int} = I_{DD} \times V_{DD}$, Watts — chip internal power
$P_{I/O}$ = Power dissipation on input and output pins — user determined

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

$$\mathbf{P_D = K \div (T_J + 273°C)} \qquad \textit{Eqn. A-2}$$

Solving equations 1 and 2 for K gives:

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

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 equations 1 and 2 iteratively for any value of T_A.

[1] This is the shortest pulse that is guaranteed to be recognized as a reset pin request. Shorter pulses are not guaranteed to override reset requests from internal sources.

[2] When any reset is initiated, internal circuitry drives the reset pin low for about 34 cycles of f_{Self_reset} and then samples the level on the reset pin about 38 cycles later to distinguish external reset requests from internal requests.

[3] This is the minimum pulse width that is guaranteed to pass through the pin synchronization circuitry. Shorter pulses may or may not be recognized. In stop mode, the synchronizer is bypassed so shorter pulses can be recognized in that case.

[4] Timing is shown with respect to 20% V_{DD} and 80% V_{DD} levels. Temperature range –40°C to 85°C.

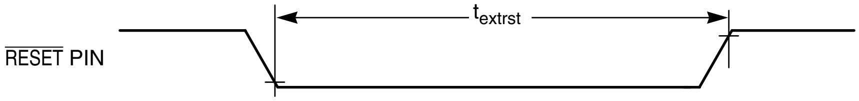


Figure A-11. Reset Timing

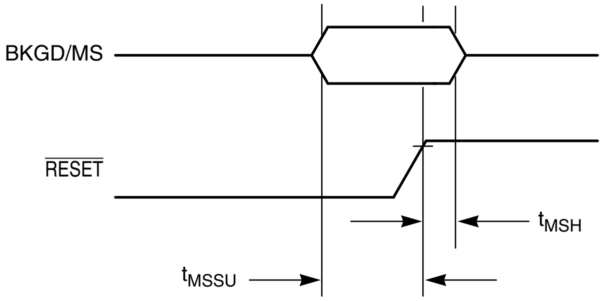


Figure A-12. Active Background Debug Mode Latch Timing

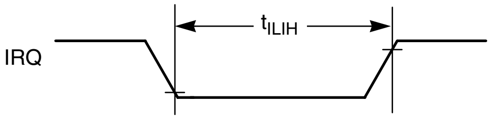


Figure A-13. IRQ Timing

A.9.2 Timer/PWM (TPM) 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.

B.4 48-Pin QFN Package Drawing