What is "<u>Embedded - Microcontrollers</u>"?

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

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	24
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 4x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	32-VFQFN Exposed Pad
Supplier Device Package	32-HVQFN (5x5)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s08gt16amfce

Chapter 9
Internal Clock Generator (S08ICGV4)

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

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
0x0054	ATDPE	ATDPE7	ATDPE6	ATDPE5	ATDPE4	ATDPE3	ATDPE2	ATDPE1	ATDPE0
0x0055–0x0057	Reserved	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
0x0058	IICA	ADDR							0
0x0059	IICF	MULT		ICR					
0x005A	IICC	IICEN	IICIE	MST	TX	TXAK	RSTA	0	0
0x005B	IICS	TCF	IAAS	BUSY	ARBL	0	SRW	IICIF	RXAK
0x005C	IICD	DATA							
0x005D–0x005F	Reserved	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
0x0060	TPM2SC	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
0x0061	TPM2CNTH	Bit 15	14	13	12	11	10	9	Bit 8
0x0062	TPM2CNTL	Bit 7	6	5	4	3	2	1	Bit 0
0x0063	TPM2MODH	Bit 15	14	13	12	11	10	9	Bit 8
0x0064	TPM2MODL	Bit 7	6	5	4	3	2	1	Bit 0
0x0065	TPM2C0SC	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
0x0066	TPM2C0VH	Bit 15	14	13	12	11	10	9	Bit 8
0x0067	TPM2C0VL	Bit 7	6	5	4	3	2	1	Bit 0
0x0068	TPM2C1SC	CH1F	CH1IE	MS1B	MS1A	ELS1B	ELS1A	0	0
0x0069	TPM2C1VH	Bit 15	14	13	12	11	10	9	Bit 8
0x006A	TPM2C1VL	Bit 7	6	5	4	3	2	1	Bit 0
0x006B–0x007F	Reserved	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—

High-page registers, shown in Table 4-3, are accessed much less often than other I/O and control registers so they have been located outside the direct addressable memory space, starting at 0x1800.

Table 4-3. High-Page Register Summary

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
0x1800	SRS	POR	PIN	COP	ILOP	ILAD	ICG	LVD	0
0x1801	SBDFR	0	0	0	0	0	0	0	BDFR
0x1802	SOPT	COPE	COPT	STOPE	—	0	0	BKGDPE	—
0x1803–0x1805	Reserved	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
0x1806	SDIDH					ID11	ID10	ID9	ID8
0x1807	SDIDL	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
0x1808	SRTISC	RTIF	RTIACK	RTICLKS	RTIE	0	RTIS2	RTIS1	RTIS0
0x1809	SPMSC1	LVDF	LVDACK	LVDIE	LVDRE	LVDSE	LVDE	0	0
0x180A	SPMSC2	LVWF	LVWACK	LVDV	LVWV	PPDF	PPDACK	PDC	PPDC
0x180B–0x180F	Reserved	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
0x1810	DBGCAH	Bit 15	14	13	12	11	10	9	Bit 8

Table 4-3. High-Page Register Summary (continued)

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
0x1811	DBGCAL	Bit 7	6	5	4	3	2	1	Bit 0
0x1812	DBGCBH	Bit 15	14	13	12	11	10	9	Bit 8
0x1813	DBGCBL	Bit 7	6	5	4	3	2	1	Bit 0
0x1814	DBGFH	Bit 15	14	13	12	11	10	9	Bit 8
0x1815	DBGFL	Bit 7	6	5	4	3	2	1	Bit 0
0x1816	DBGC	DBGEN	ARM	TAG	BRKEN	RWA	RWAEN	RWB	RWBEN
0x1817	DBGT	TRGSEL	BEGIN	0	0	TRG3	TRG2	TRG1	TRG0
0x1818	DBGS	AF	BF	ARMF	0	CNT3	CNT2	CNT1	CNT0
0x1819–0x181F	Reserved	—	—	—	—	—	—	—	—
0x1820	FCDIV	DIVLD	PRDIV8	DIV5	DIV4	DIV3	DIV2	DIV1	DIV0
0x1821	FOPT	KEYEN	FNORED	0	0	0	0	SEC01	SEC00
0x1822	Reserved	—	—	—	—	—	—	—	—
0x1823	FCNFG	0	0	KEYACC	0	0	0	0	0
0x1824	FPROT	FPS7	FPS6	FPS5	FPS4	FPS3	FPS2	FPS1	FPDIS
0x1825	FSTAT	FCBEF	FCCF	FPVIOL	FACCERR	0	FBLANK	0	0
0x1826	FCMD	FCMD7	FCMD6	FCMD5	FCMD4	FCMD3	FCMD2	FCMD1	FCMD0
0x1827–0x182B	Reserved	—	—	—	—	—	—	—	—

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

Table 4-4. Nonvolatile Register Summary

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
0xFFB0 – 0xFFB7	NVBACKKEY	8-Byte Comparison Key							
0xFFB8 – 0xFFBC	Reserved	—	—	—	—	—	—	—	—
0xFFBD	NVPROT	FPS7	FPS6	FPS5	FPS4	FPS3	FPS2	FPS1	FPDIS
0xFFBE	NVICGTRM[1]	NVTRIM							
0xFFBF	NVOPT	KEYEN	FNORED	0	0	0	0	SEC01	SEC00

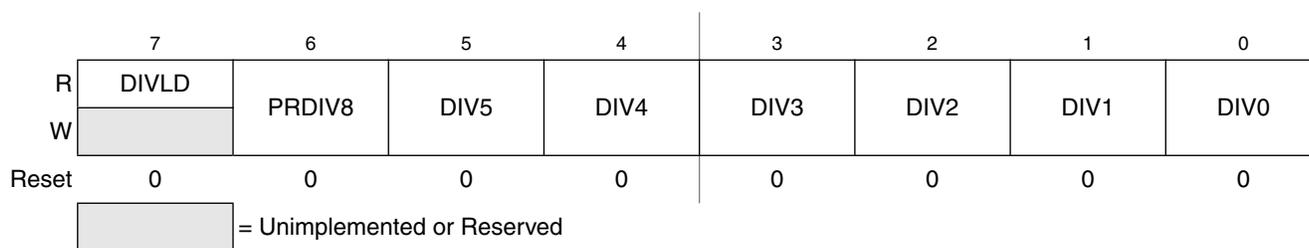[1] NVICGTRM is the factory trim value. This value must be copied to ICGTRM in user code.

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

4.6 Register Definition

The FLASH module has registers in the high-page register space, three locations in the nonvolatile register space in FLASH memory that are copied into three corresponding high-page control registers at reset. There is also an 8-byte comparison key in FLASH memory. Refer to Table 4-3 and Table 4-4 for the absolute address assignments for all FLASH registers. This section refers to registers and control bits only by their names. A Freescale-provided equate or header file normally is used to translate these names into the appropriate absolute addresses.

4.6.1 FLASH Clock Divider Register (FCDIV)

Bit 7 of this register is a read-only status flag. Bits 6 through 0 may be read at any time but can be written only one time. Before any erase or programming operations are possible, write to this register to set the frequency of the clock for the nonvolatile memory system within acceptable limits.

	7	6	5	4	3	2	1	0
R	DIVLD	PRDIV8	DIV5	DIV4	DIV3	DIV2	DIV1	DIV0
W		PRDIV8	DIV5	DIV4	DIV3	DIV2	DIV1	DIV0
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 4-5. FLASH Clock Divider Register (FCDIV)

Table 4-6. FCDIV Field Descriptions

Field	Description
7 DIVLD	**Divisor Loaded Status Flag** — When set, this read-only status flag indicates that the FCDIV register has been written since reset. Reset clears this bit and the first write to this register causes this bit to become set regardless of the data written. 0 FCDIV has not been written since reset; erase and program operations disabled for FLASH. 1 FCDIV has been written since reset; erase and program operations enabled for FLASH.
6 PRDIV8	**Prescale (Divide) FLASH Clock by 8** 0 Clock input to the FLASH clock divider is the bus rate clock. 1 Clock input to the FLASH clock divider is the bus rate clock divided by 8.
5 DIV[5:0]	**Divisor for FLASH Clock Divider** — The FLASH clock divider divides the bus rate clock (or the bus rate clock divided by 8 if PRDIV8 = 1) by the value in the 6-bit DIV5:DIV0 field plus one. The resulting frequency of the internal FLASH clock must fall within the range of 200 kHz to 150 kHz for proper FLASH operations. Program/erase timing pulses are one cycle of this internal FLASH clock, which corresponds to a range of 5 μs to 6.7 μs. The automated programming logic uses an integer number of these pulses to complete an erase or program operation. See Equation 4-1 and Equation 4-2. Table 4-7 shows the appropriate values for PRDIV8 and DIV5:DIV0 for selected bus frequencies.

$$\text{if PRDIV8 = 0 — } f_{FCLK} = f_{Bus} \div ([\text{DIV5:DIV0}] + 1) \qquad \textit{Eqn. 4-1}$$

$$\text{if PRDIV8 = 1 — } f_{FCLK} = f_{Bus} \div (8 \times ([\text{DIV5:DIV0}] + 1)) \qquad \textit{Eqn. 4-2}$$

Field	Description
3 ILAD	**Illegal Address** — Reset was caused by an attempt to access a designated illegal address. 0 Reset not caused by an illegal address access. 1 Reset caused by an illegal address access. Illegal address areas in the MC9S08GT16A are: • 0x0880 - 0x17FF — Gap from end of RAM to start of high page registers • 0x182C - 0xBFFF — Gap from end of high page registers to start of Flash memory Unused and reserved locations in register areas are not considered designated illegaladdresses and do not triggerillegal address resets.
2 ICG	**Internal Clock Generation Module Reset** — Reset was caused by an ICG module reset. 0 Reset not caused by ICG module. 1 Reset caused by ICG module.
1 LVD	**Low Voltage Detect** — If the LVD reset is enabled (LVDE = LVDRE = 1) and the supply drops below the LVD trip voltage, an LVD reset occurs. The LVD function is disabled when the MCU enters stop. To maintain LVD operation in stop, the LVDSE bit must be set. 0 Reset not caused by LVD trip or POR. 1 Reset caused by LVD trip or POR.

5.7.3   System Background Debug Force Reset Register (SBDFR)

This register contains a single write-only control bit. A serial background command such as WRITE_BYTE must be used to write to SBDFR. Attempts to write this register from a user program are ignored. Reads always return 0x00.

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W								BDFR Note [1]
Reset	0	0	0	0	0	0	0	0

    = Unimplemented or Reserved

[1]   BDFR is writable only through serial background debug commands, not from user programs.

Figure 5-4. System Background Debug Force Reset Register (SBDFR)

Table 5-4. SBDFR Field Descriptions

Field	Description
0 BDFR	**Background Debug Force Reset** — A serial background mode command such as WRITE_BYTE allows an external debug host to force a target system reset. Writing 1 to this bit forces an MCU reset. This bit cannot be written from a user program.

Chapter 6
Parallel Input/Output

6.1   Introduction

This section explains software controls related to parallel input/output (I/O). The MC9S08GT16A/GT8A has six I/O ports which include a total of up to 39 general-purpose I/O pins (one pin, PTG0, is output only). See Chapter 2, "Pins and Connections," for more information about the logic and hardware aspects of these pins.

Many of these pins are shared with on-chip peripherals such as timer systems, external interrupts, or keyboard interrupts. When these other modules are not controlling the port pins, they revert to general-purpose I/O control. For each I/O pin, a port data bit provides access to input (read) and output (write) data, a data direction bit controls the direction of the pin, and a pullup enable bit enables an internal pullup device (provided the pin is configured as an input), and a slew rate control bit controls the rise and fall times of the pins.

Pins that are not used in the application must be terminated. This prevents excess current caused by floating inputs and enhances immunity during noise or transient events. Termination methods include:

- Configuring unused pins as outputs driving high or low
- Configuring unused pins as inputs and using internal or external pullups

Never connect unused pins to V_{DD} or V_{SS}.

NOTE

> Not all general-purpose I/O pins are available on all packages. To avoid extra current drain from floating input pins, the user's reset initialization routine in the application program should either enable on-chip pullup devices or change the direction of unconnected pins to outputs so the pins do not float.

6.1.1   Features

Parallel I/O features, depending on package choice, include:

- A total of 39 general-purpose I/O pins in six ports (PTG0 is output only)
- High-current drivers on port C pins
- Hysteresis input buffers
- Software-controlled pullups on each input pin
- Software-controlled slew rate output buffers
- Eight port A pins shared with KBI

6.4   Stop Modes

Depending on the stop mode, I/O functions differently as the result of executing a STOP instruction. An explanation of I/O behavior for the various stop modes follows:

- When the MCU enters stop1 mode, all internal registers including general-purpose I/O control and data registers are powered down. All of the general-purpose I/O pins assume their reset state: output buffers and pullups turned off. Upon exit from stop1, all I/O must be initialized as if the MCU had been reset.

- When the MCU enters stop2 mode, the internal registers are powered down as in stop1 but the I/O pin states are latched and held. For example, a port pin that is an output driving low continues to function as an output driving low even though its associated data direction and output data registers are powered down internally. Upon exit from stop2, the pins continue to hold their states until a 1 is written to the PPDACK bit. To avoid discontinuity in the pin state following exit from stop2, the user must restore the port control and data registers to the values they held before entering stop2. These values can be stored in RAM before entering stop2 because the RAM is maintained during stop2.

- In stop3 mode, all I/O is maintained because internal logic circuity stays powered up. Upon recovery, normal I/O function is available to the user.

6.5   Register Definition

This section provides information about all registers and control bits associated with the parallel I/O ports.

Refer to tables in Chapter 4, "Memory," for the absolute address assignments for all parallel I/O registers. This section refers to registers and control bits only by their names. A Freescale-provided equate or header file normally is used to translate these names into the appropriate absolute addresses.

6.5.1   Port A Registers (PTAD, PTAPE, PTASE, and PTADD)

Port A includes eight pins shared between general-purpose I/O and the KBI module. Port A pins used as general-purpose I/O pins are controlled by the port A data (PTAD), data direction (PTADD), pullup enable (PTAPE), and slew rate control (PTASE) registers.

If the KBI takes control of a port A pin, the corresponding PTASE bit is ignored since the pin functions as an input. As long as PTADD is 0, the PTAPE controls the pullup enable for the KBI function. Reads of PTAD will return the logic value of the corresponding pin, provided PTADD is 0.
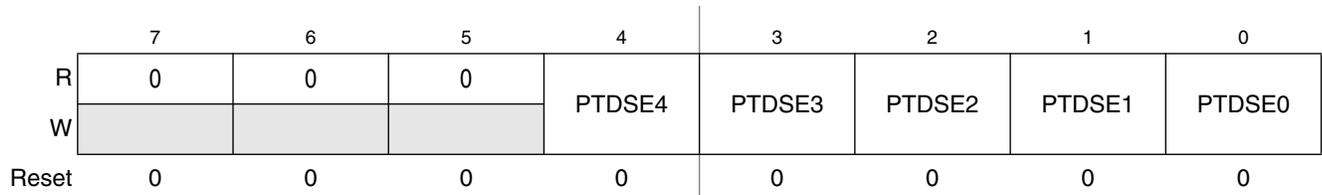
	7	6	5	4	3	2	1	0
R	0	0	0	PTDSE4	PTDSE3	PTDSE2	PTDSE1	PTDSE0
W								
Reset	0	0	0	0	0	0	0	0

Figure 6-22. Slew Rate Control Enable for Port D (PTDSE)

Table 6-15. PTDSE Field Descriptions

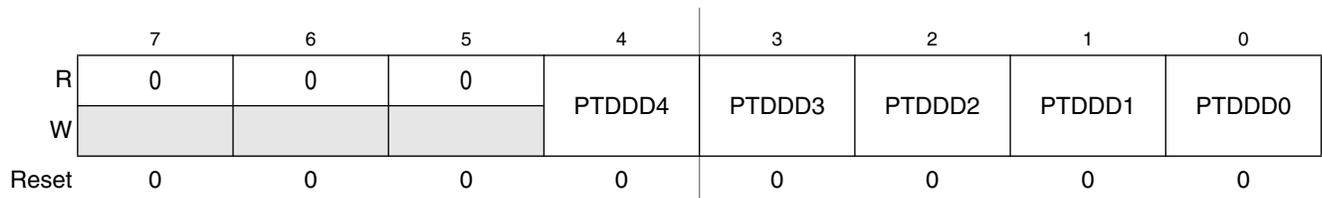| Field	Description
4:0 PTDSE[4:0]	**Slew Rate Control Enable for Port D Bits** — For port D pins that are outputs, these read/write control bits determine whether the slew rate controlled outputs are enabled. For port D pins that are configured as inputs, these bits are ignored. 0 Slew rate control disabled. 1 Slew rate control enabled.

	7	6	5	4	3	2	1	0
R	0	0	0	PTDDD4	PTDDD3	PTDDD2	PTDDD1	PTDDD0
W								
Reset	0	0	0	0	0	0	0	0

Figure 6-23. Data Direction for Port D (PTDDD)

Table 6-16. PTDDD Field Descriptions

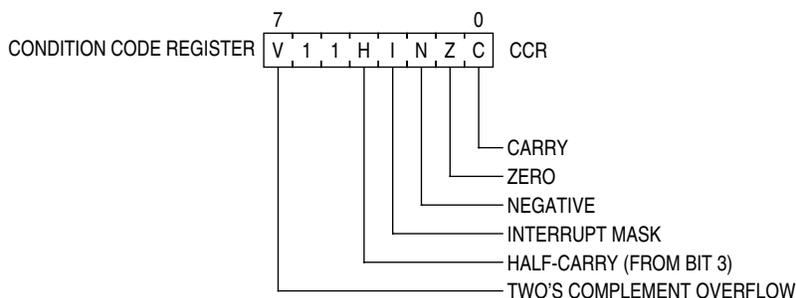| Field	Description
4:0 PTDDD[4:0]	**Data Direction for Port D Bits** — These read/write bits control the direction of port D pins and what is read for PTDD reads. 0 Input (output driver disabled) and reads return the pin value. 1 Output driver enabled for port D bit n and PTDD reads return the contents of PTDDn.

Figure 8-2. Condition Code Register

Table 8-1. CCR Register Field Descriptions

Field	Description
7 V	**Two's Complement Overflow Flag** — The CPU sets the overflow flag when a two's complement overflow occurs. The signed branch instructions BGT, BGE, BLE, and BLT use the overflow flag. 0 No overflow 1 Overflow
4 H	**Half-Carry Flag** — The CPU sets the half-carry flag when a carry occurs between accumulator bits 3 and 4 during an add-without-carry (ADD) or add-with-carry (ADC) operation. The half-carry flag is required for binary-coded decimal (BCD) arithmetic operations. The DAA instruction uses the states of the H and C condition code bits to automatically add a correction value to the result from a previous ADD or ADC on BCD operands to correct the result to a valid BCD value. 0 No carry between bits 3 and 4 1 Carry between bits 3 and 4
3 I	**Interrupt Mask Bit** — When the interrupt mask is set, all maskable CPU interrupts are disabled. CPU interrupts are enabled when the interrupt mask is cleared. When a CPU interrupt occurs, the interrupt mask is set automatically after the CPU registers are saved on the stack, but before the first instruction of the interrupt service routine is executed. Interrupts are not recognized at the instruction boundary after any instruction that clears I (CLI or TAP). This ensures that the next instruction after a CLI or TAP will always be executed without the possibility of an intervening interrupt, provided I was set. 0 Interrupts enabled 1 Interrupts disabled
2 N	**Negative Flag** — The CPU sets the negative flag when an arithmetic operation, logic operation, or data manipulation produces a negative result, setting bit 7 of the result. Simply loading or storing an 8-bit or 16-bit value causes N to be set if the most significant bit of the loaded or stored value was 1. 0 Non-negative result 1 Negative result
1 Z	**Zero Flag** — The CPU sets the zero flag when an arithmetic operation, logic operation, or data manipulation produces a result of 0x00 or 0x0000. Simply loading or storing an 8-bit or 16-bit value causes Z to be set if the loaded or stored value was all 0s. 0 Non-zero result 1 Zero result
0 C	**Carry/Borrow Flag** — The CPU sets the carry/borrow flag when an addition operation produces a carry out of bit 7 of the accumulator or when a subtraction operation requires a borrow. Some instructions — such as bit test and branch, shift, and rotate — also clear or set the carry/borrow flag. 0 No carry out of bit 7 1 Carry out of bit 7

8.3   Addressing Modes

Addressing modes define the way the CPU accesses operands and data. In the HCS08, all memory, status and control registers, and input/output (I/O) ports share a single 64-Kbyte linear address space so a 16-bit binary address can uniquely identify any memory location. This arrangement means that the same instructions that access variables in RAM can also be used to access I/O and control registers or nonvolatile program space.

Some instructions use more than one addressing mode. For instance, move instructions use one addressing mode to specify the source operand and a second addressing mode to specify the destination address. Instructions such as BRCLR, BRSET, CBEQ, and DBNZ use one addressing mode to specify the location of an operand for a test and then use relative addressing mode to specify the branch destination address when the tested condition is true. For BRCLR, BRSET, CBEQ, and DBNZ, the addressing mode listed in the instruction set tables is the addressing mode needed to access the operand to be tested, and relative addressing mode is implied for the branch destination.

8.3.1   Inherent Addressing Mode (INH)

In this addressing mode, operands needed to complete the instruction (if any) are located within CPU registers so the CPU does not need to access memory to get any operands.

8.3.2   Relative Addressing Mode (REL)

Relative addressing mode is used to specify the destination location for branch instructions. A signed 8-bit offset value is located in the memory location immediately following the opcode. During execution, if the branch condition is true, the signed offset is sign-extended to a 16-bit value and is added to the current contents of the program counter, which causes program execution to continue at the branch destination address.

8.3.3   Immediate Addressing Mode (IMM)

In immediate addressing mode, the operand needed to complete the instruction is included in the object code immediately following the instruction opcode in memory. In the case of a 16-bit immediate operand, the high-order byte is located in the next memory location after the opcode, and the low-order byte is located in the next memory location after that.

8.3.4   Direct Addressing Mode (DIR)

In direct addressing mode, the instruction includes the low-order eight bits of an address in the direct page (0x0000–0x00FF). During execution a 16-bit address is formed by concatenating an implied 0x00 for the high-order half of the address and the direct address from the instruction to get the 16-bit address where the desired operand is located. This is faster and more memory efficient than specifying a complete 16-bit address for the operand.

IX   =   16-bit indexed no offset

IX+ =   16-bit indexed no offset, post increment (CBEQ and MOV only)

IX1 =   16-bit indexed with 8-bit offset from H:X

IX1+ =   16-bit indexed with 8-bit offset, post increment
(CBEQ only)

IX2 =   16-bit indexed with 16-bit offset from H:X

REL =   8-bit relative offset

SP1 =   Stack pointer with 8-bit offset

SP2 =   Stack pointer with 16-bit offset

### Table 8-2. HCS08 Instruction Set Summary (Sheet 1 of 7)

Source Form	Operation	Description	V	H	I	N	Z	C	Address Mode	Opcode	Operand	Bus Cycles[1]
ADC #opr8i ADC opr8a ADC opr16a ADC oprx16,X ADC oprx8,X ADC ,X ADC oprx16,SP ADC oprx8,SP	Add with Carry	$A \leftarrow (A) + (M) + (C)$	↕	↕	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP2 SP1	A9 B9 C9 D9 E9 F9 9ED9 9EE9	ii dd hh ll ee ff ff ee ff ff	2 3 4 4 3 3 5 4
ADD #opr8i ADD opr8a ADD opr16a ADD oprx16,X ADD oprx8,X ADD ,X ADD oprx16,SP ADD oprx8,SP	Add without Carry	$A \leftarrow (A) + (M)$	↕	↕	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP2 SP1	AB BB CB DB EB FB 9EDB 9EEB	ii dd hh ll ee ff ff ee ff ff	2 3 4 4 3 3 5 4
AIS #opr8i	Add Immediate Value (Signed) to Stack Pointer	$SP \leftarrow (SP) + (M)$ M is sign extended to a 16-bit value	–	–	–	–	–	–	IMM	A7	ii	2
AIX #opr8i	Add Immediate Value (Signed) to Index Register (H:X)	$H:X \leftarrow (H:X) + (M)$ M is sign extended to a 16-bit value	–	–	–	–	–	–	IMM	AF	ii	2
AND #opr8i AND opr8a AND opr16a AND oprx16,X AND oprx8,X AND ,X AND oprx16,SP AND oprx8,SP	Logical AND	$A \leftarrow (A) \, \& \, (M)$	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP2 SP1	A4 B4 C4 D4 E4 F4 9ED4 9EE4	ii dd hh ll ee ff ff ee ff ff	2 3 4 4 3 3 5 4
ASL opr8a ASLA ASLX ASL oprx8,X ASL ,X ASL oprx8,SP	Arithmetic Shift Left (Same as LSL) C ← [b7...b0] ← 0		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff	5 1 1 5 4 6
ASR opr8a ASRA ASRX ASR oprx8,X ASR ,X ASR oprx8,SP	Arithmetic Shift Right [b7...b0] → C		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	37 47 57 67 77 9E67	dd ff ff	5 1 1 5 4 6
BCC rel	Branch if Carry Bit Clear	Branch if (C) = 0	–	–	–	–	–	–	REL	24	rr	3

11.3 Functional Description

The SCI allows full-duplex, asynchronous, NRZ serial communication among the MCU and remote devices, including other MCUs. The SCI comprises a baud rate generator, transmitter, and receiver block. The transmitter and receiver operate independently, although they use the same baud rate generator. During normal operation, the MCU monitors the status of the SCI, writes the data to be transmitted, and processes received data. The following describes each of the blocks of the SCI.

11.3.1 Baud Rate Generation

As shown in Figure 11-12, the clock source for the SCI baud rate generator is the bus-rate clock.
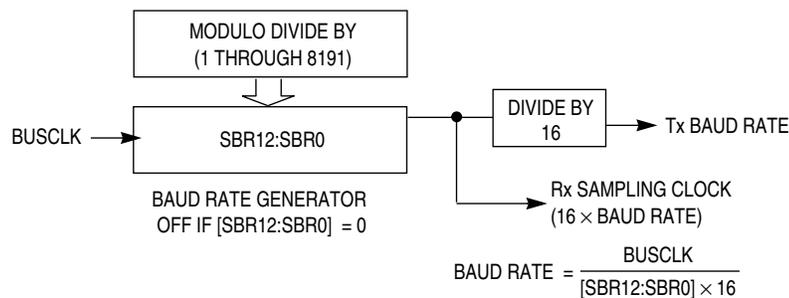
Figure 11-12. SCI Baud Rate Generation

SCI communications require the transmitter and receiver (which typically derive baud rates from independent clock sources) to use the same baud rate. Allowed tolerance on this baud frequency depends on the details of how the receiver synchronizes to the leading edge of the start bit and how bit sampling is performed.

The MCU resynchronizes to bit boundaries on every high-to-low transition, but in the worst case, there are no such transitions in the full 10- or 11-bit time character frame so any mismatch in baud rate is accumulated for the whole character time. For a Freescale Semiconductor SCI system whose bus frequency is driven by a crystal, the allowed baud rate mismatch is about ±4.5 percent for 8-bit data format and about ±4 percent for 9-bit data format. Although baud rate modulo divider settings do not always produce baud rates that exactly match standard rates, it is normally possible to get within a few percent, which is acceptable for reliable communications.

11.3.2 Transmitter Functional Description

This section describes the overall block diagram for the SCI transmitter (Figure 11-2), as well as specialized functions for sending break and idle characters.

The transmitter is enabled by setting the TE bit in SCIxC2. This queues a preamble character that is one full character frame of the idle state. The transmitter then remains idle 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

Figure 12-1. Block Diagram Highlighting the SPI Module
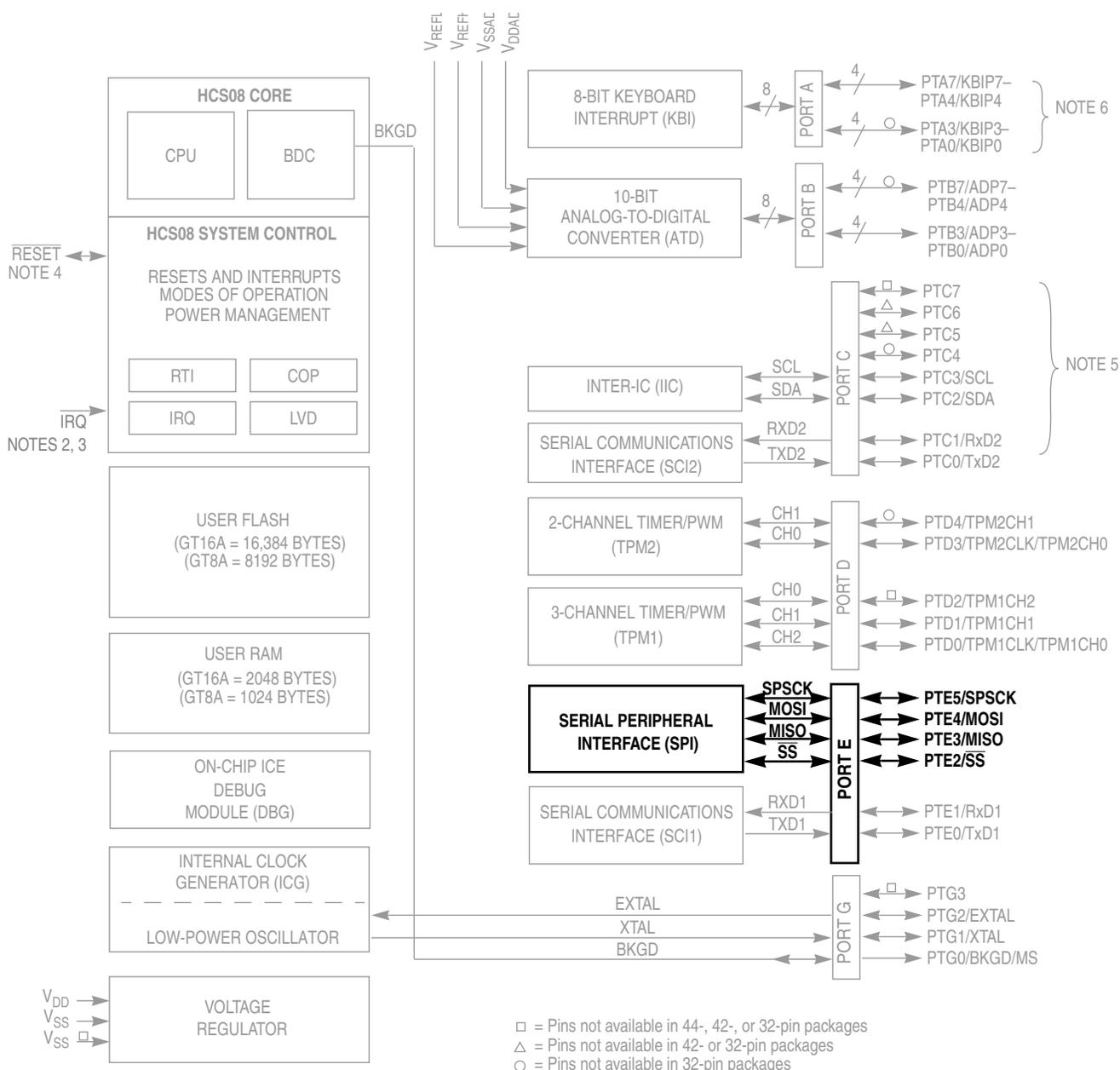
NOTES:
1. Port pins are software configurable with pullup device if input port.
2. Pin contains pullup/pulldown device if IRQ enabled (IRQPE = 1).
3. IRQ does not have a clamp diode to VDD. IRQ should not be driven above VDD.
4. Pin contains integrated pullup device.
5. High current drive
6. Pins PTA[7:4] contain both pullup and pulldown devices. Pulldown available when KBI enabled (KBIPn = 1).

14.1   Introduction

The ATD module is an analog-to-digital converter with a successive approximation register (SAR) architecture with sample and hold.

14.1.1   Features

- 8-/10-bit resolution
- 14.0 μsec, 10-bit single conversion time at a conversion frequency of 2 MHz
- Left-/right-justified result data
- Left-justified signed data mode
- Conversion complete flag or conversion complete interrupt generation
- Analog input multiplexer for up to eight analog input channels
- Single or continuous conversion mode

14.1.2   Modes of Operation

The ATD has two modes for low power

- Stop mode
- Power-down mode

14.1.2.1   Stop Mode

When the MCU goes into stop mode, the MCU stops the clocks and the ATD analog circuitry is turned off, placing the module into a low-power state. Once in stop mode, the ATD module aborts any single or continuous conversion in progress. Upon exiting stop mode, no conversions occur and the registers have their previous values. As long as the ATDPU bit is set prior to entering stop mode, the module is reactivated coming out of stop.

14.1.2.2   Power Down Mode

Clearing the ATDPU bit in register ATDC also places the ATD module in a low-power state. The ATD conversion clock is disabled and the analog circuitry is turned off, placing the module in power-down mode. (This mode does not remove power to the ATD module.) Once in power-down mode, the ATD module aborts any conversion in progress. Upon setting the ATDPU bit, the module is reactivated. During power-down mode, the ATD registers are still accessible.

Note that the reset state of the ATDPU bit is zero. Therefore, the module is reset into the power-down state.

14.1.3   Block Diagram

Figure 14-2 illustrates the functional structure of the ATD module.

A.10.1   Control Timing

Table A-12. Control Timing

Parameter	Symbol	Min	Typical	Max	Unit
Bus frequency ($t_{cyc} = 1/f_{Bus}$) $V_{DD} \geq 2.1$ V $V_{DD} < 2.1$ V	f_{Bus}	0 0	—	20 8	MHz
Real-time interrupt internal oscillator period	t_{RTI}	750	1150	1550	µs
External reset pulse width[1]	t_{extrst}	1.5 x f_{Self_reset}		—	ns
Reset low drive[2]	t_{rstdrv}	34 x f_{Self_reset}		—	ns
Active background debug mode latch setup time	t_{MSSU}	25		—	ns
Active background debug mode latch hold time	t_{MSH}	25		—	ns
IRQ pulse width[3]	t_{ILIH}	1.5 x t_{cyc}		—	ns

Port rise and fall time (load = 50 pF)[4]
    Slew rate control disabled
    Slew rate control enabled
	t_{Rise}, t_{Fall}	—
		—
			3
			30
					ns

[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 125°C.
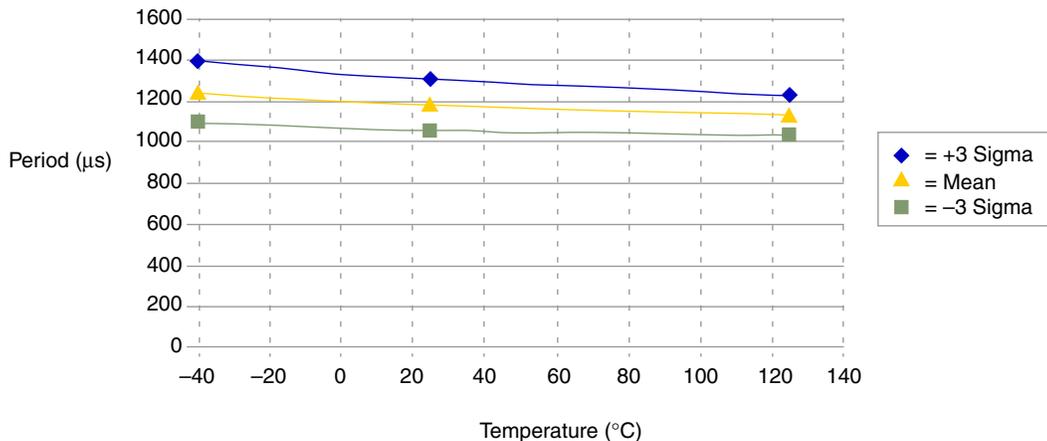
Figure A-11. Typical RTI Clock Period vs. Temperature

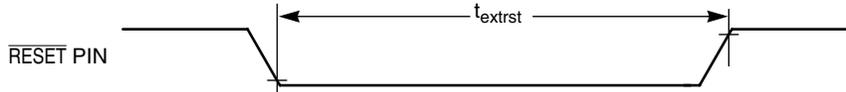**MC9S08GT16A/GT8A Data Sheet, Rev. 1**

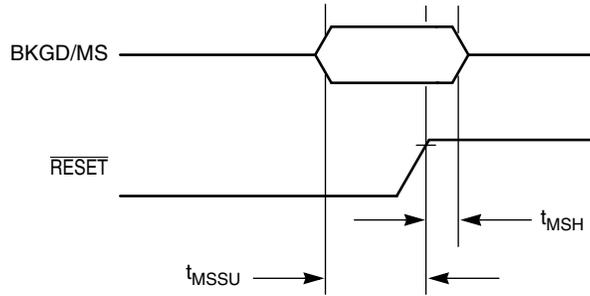**Figure A-12. Reset Timing**

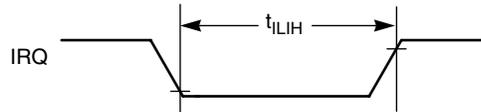**Figure A-13. Active Background Debug Mode Latch Timing**

Figure A-14. IRQ Timing

A.10.2 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.

Table A-13. TPM Input Timing

Function	Symbol	Min	Max	Unit
External clock frequency	f_{TPMext}	dc	$f_{Bus}/4$	MHz
External clock period	t_{TPMext}	4	—	t_{cyc}
External clock high time	t_{clkh}	1.5	—	t_{cyc}
External clock low time	t_{clkl}	1.5	—	t_{cyc}
Input capture pulse width	t_{ICPW}	1.5	—	t_{cyc}

A.11 FLASH Specifications

This section provides details about program/erase times and program-erase endurance for the FLASH memory.

Program and erase operations do not require any special power sources other than the normal V_{DD} supply. For more detailed information about program/erase operations, see Chapter 4, "Memory."

Table A-15. FLASH Characteristics

Characteristic	Symbol	Min	Typical	Max	Unit
Supply voltage for program/erase $T \leq 85°C$ $T > 85°C$	$V_{prog/erase}$	1.8 2.1		3.6 3.6	V V
Supply voltage for read operation $0 < f_{Bus} < 8$ MHz $0 < f_{Bus} < 20$ MHz	V_{Read}	1.8 2.08		3.6 3.6	V
Internal FCLK frequency[1]	f_{FCLK}	150		200	kHz
Internal FCLK period (1/FCLK)	t_{Fcyc}	5		6.67	µs
Byte program time (random location)[2]	t_{prog}		9		t_{Fcyc}
Byte program time (burst mode)[2]	t_{Burst}		4		t_{Fcyc}
Page erase time[2]	t_{Page}		4000		t_{Fcyc}
Mass erase time[2]	t_{Mass}		20,000		t_{Fcyc}
Byte program current[3]	RI_{DDBP}	—	4	—	mA
Page erase current[3]	RI_{DDPE}	—	6	—	mA
Program/erase endurance[4] T_L to T_H = −40°C to +125°C T = 25°C		10,000	 100,000	— —	cycles
Data retention[5]	t_{D_ret}	15	100	—	years

[1] The frequency of this clock is controlled by a software setting.

[2] These values are hardware state machine controlled. User code does not need to count cycles. This information supplied for calculating approximate time to program and erase.

[3] The program and erase currents are additional to the standard run I_{DD}. These values were measured at room temperatures with V_{DD} = 3.0 V, bus frequency = 4.0 MHz.

[4] **Typical endurance for FLASH** was evaluated for this product family on the 9S12Dx64. For additional information on how Freescale Semiconductor defines typical endurance, please refer to Engineering Bulletin EB619/D, *Typical Endurance for Nonvolatile Memory*.

[5] **Typical data retention** values are based on intrinsic capability of the technology measured at high temperature and de-rated to 25°C using the Arrhenius equation. For additional information on how Freescale Semiconductor defines typical data retention, please refer to Engineering Bulletin EB618/D, *Typical Data Retention for Nonvolatile Memory*.

NOTES:

1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M-1994.

2. ALL DIMENSIONS IN MILLIMETERS.

3. DIMENSION TO CENTER OF LEAD WHEN FORMED PARALLEL.

4. DIMENSION DOES NOT INCLUDE MOLD FLASH.   MAXIMUM MOLD FLASH 0.25.

	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE

TITLE:	DOCUMENT NO: 98ASB42767B	REV: A
42 LD PDIP	CASE NUMBER: 858-01	24 OCT 2005
	STANDARD: NON JEDEC	