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 "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	S08
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	12
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 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	16-VQFN Exposed Pad
Supplier Device Package	16-QFN-EP (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08qg44cffe

List of Chapters

Chapter 5
Resets, Interrupts, and General System Control

Chapter 6
Parallel Input/Output Control

Chapter 7
Central Processor Unit (S08CPUV2)

Chapter 8
Analog Comparator (S08ACMPV2)

Appendix A
Electrical Characteristics

Appendix B
Ordering Information and Mechanical Drawings

4.2 Reset and Interrupt Vector Assignments

Table 4-1 shows address assignments for reset and interrupt vectors. The vector names shown in this table are the labels used in the Freescale Semiconductor-provided equate file for the MC9S08QG8/4.

Table 4-1. Reset and Interrupt Vectors

Address (High:Low)	Vector	Vector Name
0xFFC0:FFC1 ⇕ 0xFFCE:FFCF	Unused Vector Space (available for user program)	
0xFFD0:FFD1	RTI	Vrti
0xFFD2:FFD3	Reserved	—
0xFFD4:FFD5	Reserved	—
0xFFD6:FFD7	ACMP	Vacmp
0xFFD8:FFD9	ADC Conversion	Vadc
0xFFDA:FFDB	KBI Interrupt	Vkeyboard
0xFFDC:FFDD	IIC	Viic
0xFFDE:FFDF	SCI Transmit	Vscitx
0xFFE0:FFE1	SCI Receive	Vscirx
0xFFE2:FFE3	SCI Error	Vscierr
0xFFE4:FFE5	SPI	Vspi
0xFFE6:FFE7	MTIM Overflow	Vmtim
0xFFE8:FFE9	Reserved	—
0xFFEA:FFEB	Reserved	—
0xFFEC:FFED	Reserved	—
0xFFEE:FFEF	Reserved	—
0xFFF0:FFF1	TPM Overflow	Vtpmovf
0xFFF2:FFF3	TPM Channel 1	Vtpmch1
0xFFF4:FFF5	TPM Channel 0	Vtpmch0
0xFFF6:FFF7	Reserved	—
0xFFF8:FFF9	Low Voltage Detect	Vlvd
0xFFFA:FFFB	IRQ	Virq
0xFFFC:FFFD	SWI	Vswi
0xFFFE:FFFF	Reset	Vreset

5.8.7 System Real-Time Interrupt Status and Control Register (SRTISC)

This high page register contains status and control bits for the RTI.

	7	6	5	4	3	2	1	0
R	RTIF	0	RTICLKS	RTIE	0	RTIS		
W		RTIACK						
Reset:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 5-9. System RTI Status and Control Register (SRTISC)

Table 5-10. SRTISC Register Field Descriptions

Field	Description
7 RTIF	**Real-Time Interrupt Flag** — This read-only status bit indicates the periodic wakeup timer has timed out. 0 Periodic wakeup timer not timed out. 1 Periodic wakeup timer timed out.
6 RTIACK	**Real-Time Interrupt Acknowledge** — This write-only bit is used to acknowledge real-time interrupt request (write 1 to clear RTIF). Writing 0 has no meaning or effect. Reads always return 0.
5 RTICLKS	**Real-Time Interrupt Clock Select** — This read/write bit selects the clock source for the real-time interrupt. 0 Real-time interrupt request clock source is internal 1-kHz oscillator. 1 Real-time interrupt request clock source is external clock.
4 RTIE	**Real-Time Interrupt Enable** — This read-write bit enables real-time interrupts. 0 Real-time interrupts disabled. 1 Real-time interrupts enabled.
2:0 RTIS	**Real-Time Interrupt Delay Selects** — These read/write bits select the period for the RTI. See Table 5-11.

Table 5-11. Real-Time Interrupt Period

RTIS2:RTIS1:RTIS0	Using Internal 1-kHz Clock Source[1] [2]	Using External Clock Source Period = t_{ext}[3]
0:0:0	Disable RTI	Disable RTI
0:0:1	8 ms	t_{ext} x 256
0:1:0	32 ms	t_{ext} x 1024
0:1:1	64 ms	t_{ext} x 2048
1:0:0	128 ms	t_{ext} x 4096
1:0:1	256 ms	t_{ext} x 8192
1:1:0	512 ms	t_{ext} x 16384
1:1:1	1.024 s	t_{ext} x 32768

1 Values are shown in this column based on t_{RTI} = 1 ms. See t_{RTI} in the appendix Section A.8.1, “Control Timing,” for the tolerance of this value.

2 The initial RTI timeout period will be up to one 1-kHz clock period less than the time specified.

3 t_{ext} is the period of the external crystal frequency.

6.3 Pin Behavior in Stop Modes

Pin behavior following execution of a STOP instruction depends on the stop mode that is entered. An explanation of pin behavior for the various stop modes follows:

- In stop1 mode, all internal registers including parallel I/O control and data registers are powered off. Each of the pins assumes its default reset state (output buffer and internal pullup disabled). Upon exit from stop1, all pins must be re-configured the same as if the MCU had been reset by POR.
- Stop2 mode is a partial power-down mode, whereby latches maintain the pin state as before the STOP instruction was executed. CPU register status and the state of I/O registers must be saved in RAM before the STOP instruction is executed to place the MCU in stop2 mode. Upon recovery from stop2 mode, before accessing any I/O, the user must examine the state of the PPDF bit in the SPMSC2 register. If the PPDF bit is 0, I/O must be initialized as if a power on reset had occurred. If the PPDF bit is 1, I/O data previously stored in RAM, before the STOP instruction was executed, and peripherals previously enabled will require being initialized and restored to their pre-stop condition. The user must then write a 1 to the PPDACK bit in the SPMSC2 register. Access of pins is now permitted again in the user application program.
- In stop3 mode, all pin states are maintained because internal logic stays powered up. Upon recovery, all pin functions are the same as before entering stop3.

6.4 Parallel I/O Registers

6.4.1 Port A Registers

This section provides information about the registers associated with the parallel I/O ports.

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

6.4.1.1 Port A Data (PTAD)

	7	6	5	4	3	2	1	0
R	0	0	PTAD5[1]	PTAD4[2]	PTAD3	PTAD2	PTAD1	PTAD0
W								
Reset:	0	0	0	0	0	0	0	0

Figure 6-2. Port A Data Register (PTAD)

1 Reads of bit PTAD5 always return the pin value of PTA5, regardless of the value stored in bit PTADD5.

2 Reads of bit PTAD4 always return the contents of PTAD4, regardless of the value stored in bit PTADD4.

6.4.4.3 Port B Drive Strength Select (PTBDS)

An output pin can be selected to have high output drive strength by setting the corresponding bit in the drive strength select register (PTBDSn). When high drive is selected, a pin is capable of sourcing and sinking greater current. Even though every I/O pin can be selected as high drive, the user must ensure that the total current source and sink limits for the chip are not exceeded. Drive strength selection is intended to affect the DC behavior of I/O pins. However, the AC behavior is also affected. High drive allows a pin to drive a greater load with the same switching speed as a low drive enabled pin into a smaller load. Because of this the EMC emissions may be affected by enabling pins as high drive.

	7	6	5	4	3	2	1	0
R W	PTBDS7	PTBDS6	PTBDS5	PTBDS4	PTBDS3	PTBDS2	PTBDS1	PTBDS0
Reset:	0	0	0	0	0	0	0	0

Figure 6-16. Drive Strength Selection for Port B Register (PTBDS)

Table 6-10. PTBDS Register Field Descriptions

Field	Description
7:0 PTBDS[7:0]	**Output Drive Strength Selection for Port B Bits** — Each of these control bits selects between low and high output drive for the associated PTB pin. For port B pins that are configured as inputs, these bits have no effect. 0 Low output drive strength selected for port B bit n. 1 High output drive strength selected for port B bit n.

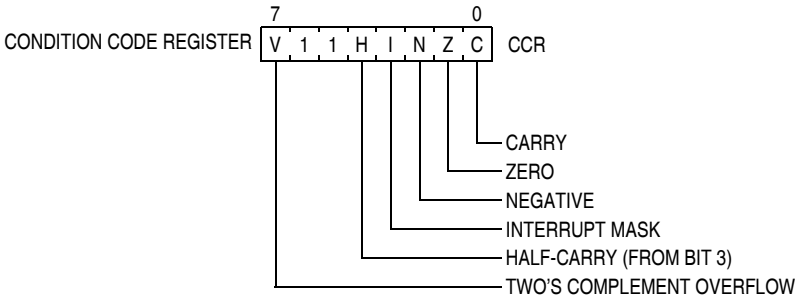


Figure 7-2. Condition Code Register

Table 7-1. 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

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.

Table 7-2. . Instruction Set Summary (Sheet 6 of 9)

Source Form	Operation	Address Mode	Object Code	Cycles	Cyc-by-Cyc Details	Affect on CCR	
						VH	I N Z C

MOV *oprx8*,*oprx8*
MOV *oprx8*,X+
MOV #*oprx8*,*oprx8*
MOV ,X+,*oprx8*
	Move
	$(M)_{destination} \leftarrow (M)_{source}$
	In IX+/DIR and DIR/IX+ Modes,
	H:X ← (H:X) + $0001
		DIR/DIR
		DIR/IX+
		IMM/DIR
		IX+/DIR
			4E dd dd
			5E dd
			6E ii dd
			7E dd
				5
				5
				4
				5
					rpwpp
					rfwpp
					pwpp
					rfwpp
						0 –	– ↕ ↕ –

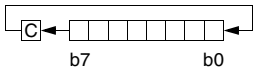
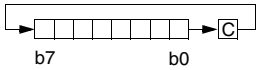
MUL	Unsigned multiply X:A ← (X) × (A)	INH	42	5	ffffp	– 0	– – – 0
NEG *oprx8* NEGA NEGX NEG *oprx8*,X NEG ,X NEG *oprx8*,SP	Negate M ← – (M) = $00 – (M) (Two's Complement) A ← – (A) = $00 – (A) X ← – (X) = $00 – (X) M ← – (M) = $00 – (M) M ← – (M) = $00 – (M) M ← – (M) = $00 – (M)	DIR INH INH IX1 IX SP1	30 dd 40 50 60 ff 70 9E 60 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	↕ –	– ↕ ↕ ↕
NOP	No Operation — Uses 1 Bus Cycle	INH	9D	1	p	– –	– – – –
NSA	Nibble Swap Accumulator A ← (A[3:0]:A[7:4])	INH	62	1	p	– –	– – – –
ORA #*oprx8* ORA *oprx8* ORA *oprx16* ORA *oprx16*,X ORA *oprx8*,X ORA ,X ORA *oprx16*,SP ORA *oprx8*,SP	Inclusive OR Accumulator and Memory A ← (A) | (M)	IMM DIR EXT IX2 IX1 IX SP2 SP1	AA ii BA dd CA hh ll DA ee ff EA ff FA 9E DA ee ff 9E EA ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	0 –	– ↕ ↕ –
PSHA	Push Accumulator onto Stack Push (A); SP ← (SP) – $0001	INH	87	2	sp	– –	– – – –
PSHH	Push H (Index Register High) onto Stack Push (H); SP ← (SP) – $0001	INH	8B	2	sp	– –	– – – –
PSHX	Push X (Index Register Low) onto Stack Push (X); SP ← (SP) – $0001	INH	89	2	sp	– –	– – – –
PULA	Pull Accumulator from Stack SP ← (SP + $0001); Pull (A)	INH	86	3	ufp	– –	– – – –
PULH	Pull H (Index Register High) from Stack SP ← (SP + $0001); Pull (H)	INH	8A	3	ufp	– –	– – – –
PULX	Pull X (Index Register Low) from Stack SP ← (SP + $0001); Pull (X)	INH	88	3	ufp	– –	– – – –
ROL *oprx8* ROLA ROLX ROL *oprx8*,X ROL ,X ROL *oprx8*,SP	Rotate Left through Carry 	DIR INH INH IX1 IX SP1	39 dd 49 59 69 ff 79 9E 69 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	↕ –	– ↕ ↕ ↕
ROR *oprx8* RORA RORX ROR *oprx8*,X ROR ,X ROR *oprx8*,SP	Rotate Right through Carry 	DIR INH INH IX1 IX SP1	36 dd 46 56 66 ff 76 9E 66 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	↕ –	– ↕ ↕ ↕

Table 9-8. Input Clock Select

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

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.

	7	6	5	4	3	2	1	0
R W	ADPC7	ADPC6	ADPC5	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0
Reset:	0	0	0	0	0	0	0	0

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

Table 9-9. APCTL1 Register Field Descriptions

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

Table 11-4. IIC Divider and Hold Values

ICR (hex)	SCL Divider	SDA Hold Value
00	20	7
01	22	7
02	24	8
03	26	8
04	28	9
05	30	9
06	34	10
07	40	10
08	28	7
09	32	7
0A	36	9
0B	40	9
0C	44	11
0D	48	11
0E	56	13
0F	68	13
10	48	9
11	56	9
12	64	13
13	72	13
14	80	17
15	88	17
16	104	21
17	128	21
18	80	9
19	96	9
1A	112	17
1B	128	17
1C	144	25
1D	160	25
1E	192	33
1F	240	33

ICR (hex)	SCL Divider	SDA Hold Value
20	160	17
21	192	17
22	224	33
23	256	33
24	288	49
25	320	49
26	384	65
27	480	65
28	320	33
29	384	33
2A	448	65
2B	512	65
2C	576	97
2D	640	97
2E	768	129
2F	960	129
30	640	65
31	768	65
32	896	129
33	1024	129
34	1152	193
35	1280	193
36	1536	257
37	1920	257
38	1280	129
39	1536	129
3A	1792	257
3B	2048	257
3C	2304	385
3D	2560	385
3E	3072	513
3F	3840	513

11.7 Initialization/Application Information

Module Initialization (Slave)

1. Write: IICA
   — to set the slave address
2. Write: IICC
   — to enable IIC and interrupts
3. Initialize RAM variables (IICEN = 1 and IICIE = 1) for transmit data
4. Initialize RAM variables used to achieve the routine shown in Figure 11-11

Module Initialization (Master)

1. Write: IICF
   — to set the IIC baud rate (example provided in this chapter)
2. Write: IICC
   — to enable IIC and interrupts
3. Initialize RAM variables (IICEN = 1 and IICIE = 1) for transmit data
4. Initialize RAM variables used to achieve the routine shown in Figure 11-11
5. Write: IICC
   — to enable TX
6. Write: IICC
   — to enable MST (master mode)
7. Write: IICD
   — with the address of the target slave. (The LSB of this byte will determine whether the communication is master receive or transmit.)

Module Use

The routine shown in Figure 11-11 can handle both master and slave IIC operations. For slave operation, an incoming IIC message that contains the proper address will begin IIC communication. For master operation, communication must be initiated by writing to the IICD register.

Register Model

IICA	ADDR							0

Address to which the module will respond when addressed as a slave (in slave mode)

IICF	MULT		ICR					

Baud rate = BUSCLK / (2 x MULT x (SCL DIVIDER))

IICC	IICEN	IICIE	MST	TX	TXAK	RSTA	0	0

Module configuration

IICS	TCF	IAAS	BUSY	ARBL	0	SRW	IICIF	RXAK

Module status flags

IICD	DATA							

Data register; Write to transmit IIC data read to read IIC data

Figure 11-10. IIC Module Quick Start

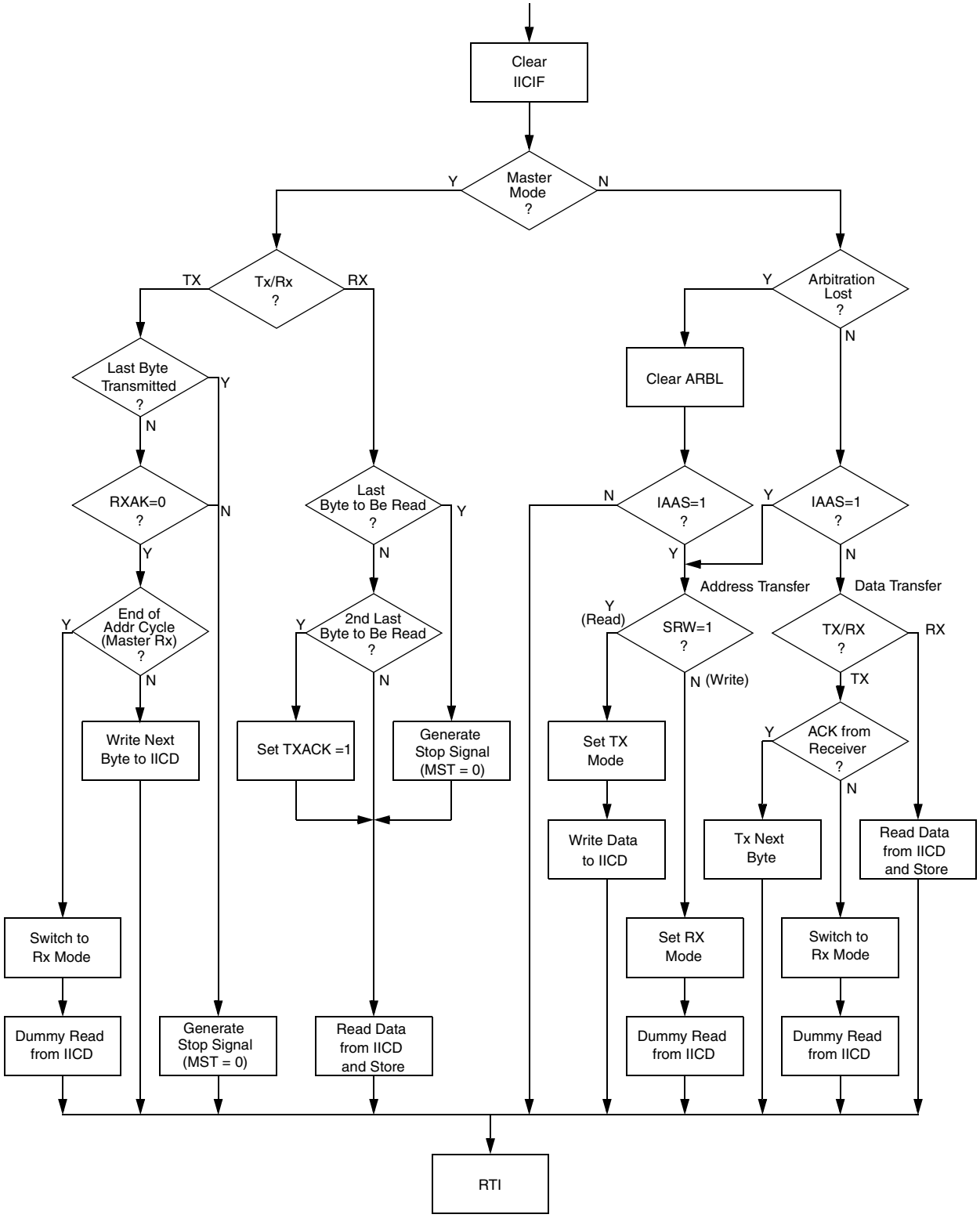
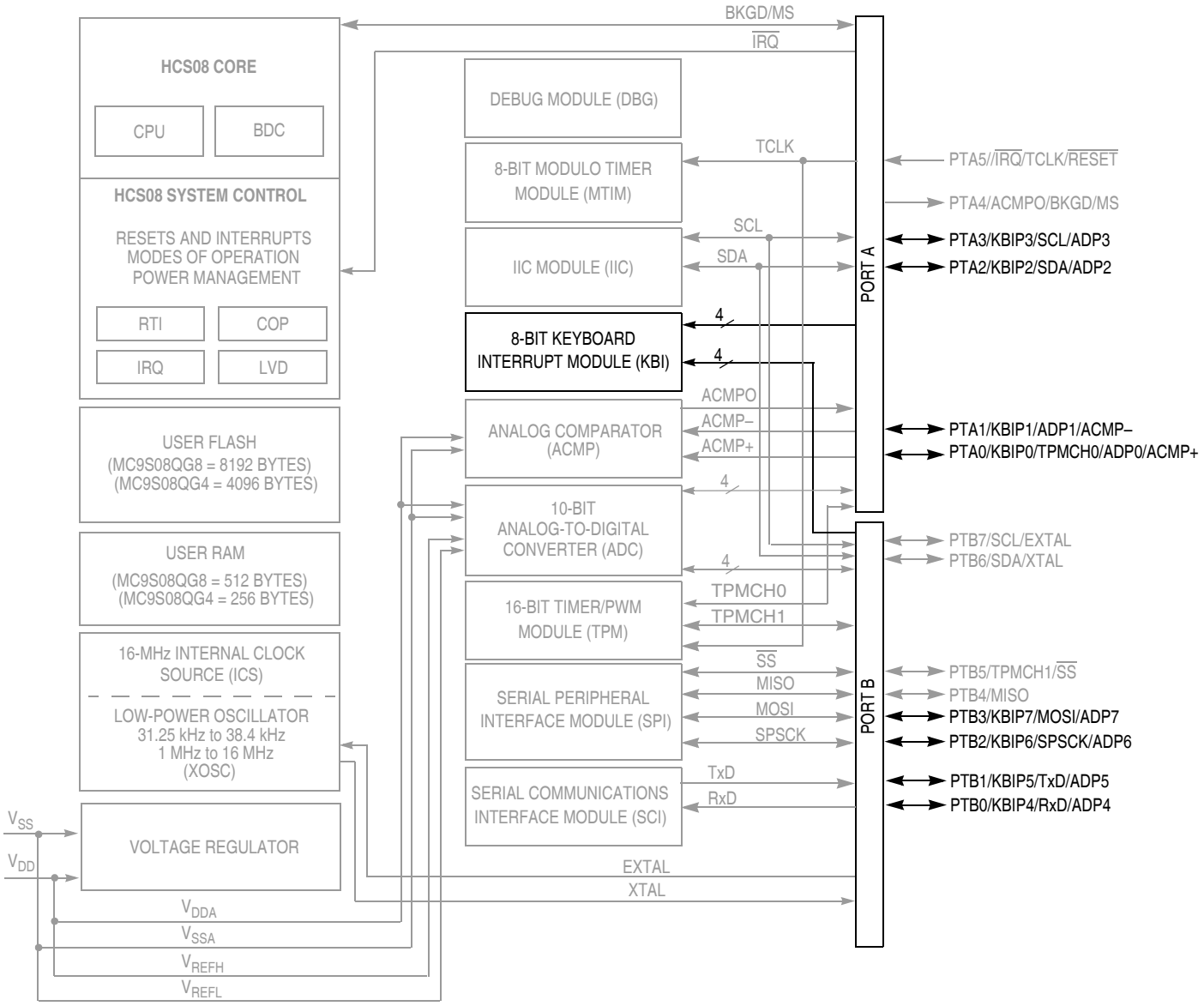


Figure 11-11. Typical IIC Interrupt Routine



NOTES:

1. Not all pins or pin functions are available on all devices, see Table 1-1 for available functions on each device.
2. Port pins are software configurable with pullup device if input port.
3. Port pins are software configurable for output drive strength.
4. Port pins are software configurable for output slew rate control.
5. $\overline{IRQ}$ contains a software configurable (IRQPDD) pullup device if PTA5 enabled as $\overline{IRQ}$ pin function (IRQPE = 1).
6. $\overline{RESET}$ contains integrated pullup device if PTA5 enabled as reset pin function (RSTPE = 1).
7. PTA4 contains integrated pullup device if BKGD enabled (BKGDPE = 1).
8. SDA and SCL pin locations can be repositioned under software control (IICPS), defaults on PTA2 and PTA3.
9. When pin functions as KBI (KBIPEn = 1) and associated pin is configured to enable the pullup device, KBEDGn can be used to reconfigure the pullup as a pulldown device.

Figure 12-1. MC9S08QG8/4 Block Diagram Highlighting KBI Block and Pins

13.3.2 MTIM Clock Configuration Register (MTIMCLK)

MTIMCLK contains the clock select bits (CLKS) and the prescaler select bits (PS).

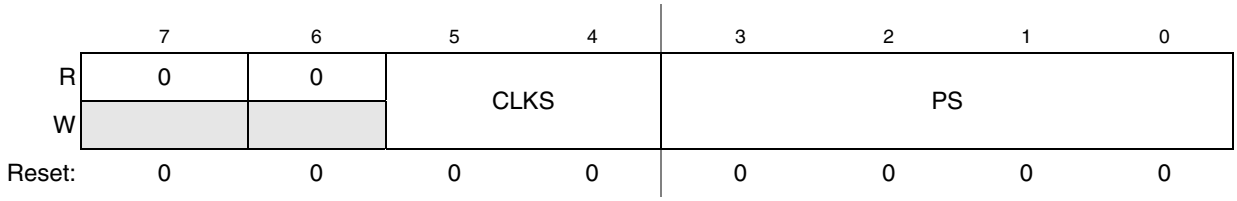


Figure 13-5. MTIM Clock Configuration Register

Table 13-3. MTIM Clock Configuration Register Field Description

Field	Description
7:6	Unused register bits, always read 0.
5:4 CLKS	**Clock Source Select** — These two read/write bits select one of four different clock sources as the input to the MTIM prescaler. Changing the clock source while the counter is active does not clear the counter. The count continues with the new clock source. Reset clears CLKS to 000. 00 Encoding 0. Bus clock (BUSCLK) 01 Encoding 1. Fixed-frequency clock (XCLK) 10 Encoding 3. External source (TCLK pin), falling edge 11 Encoding 4. External source (TCLK pin), rising edge All other encodings default to the bus clock (BUSCLK).
3:0 PS	**Clock Source Prescaler** — These four read/write bits select one of nine outputs from the 8-bit prescaler. Changing the prescaler value while the counter is active does not clear the counter. The count continues with the new prescaler value. Reset clears PS to 0000. 0000 Encoding 0. MTIM clock source ÷ 1 0001 Encoding 1. MTIM clock source ÷ 2 0010 Encoding 2. MTIM clock source ÷ 4 0011 Encoding 3. MTIM clock source ÷ 8 0100 Encoding 4. MTIM clock source ÷ 16 0101 Encoding 5. MTIM clock source ÷ 32 0110 Encoding 6. MTIM clock source ÷ 64 0111 Encoding 7. MTIM clock source ÷ 128 1000 Encoding 8. MTIM clock source ÷ 256 All other encodings default to MTIM clock source ÷ 256.

The SYNC command is unlike other BDC commands because the host does not necessarily know the correct communications speed to use for BDC communications until after it has analyzed the response to the SYNC command.

To issue a SYNC command, the host:

- Drives the BKGD pin low for at least 128 cycles of the slowest possible BDC clock (The slowest clock is normally the reference oscillator/64 or the self-clocked rate/64.)
- Drives BKGD high for a brief speedup pulse to get a fast rise time (This speedup pulse is typically one cycle of the fastest clock in the system.)
- Removes all drive to the BKGD pin so it reverts to high impedance
- Monitors the BKGD pin for the sync response pulse

The target, upon detecting the SYNC request from the host (which is a much longer low time than would ever occur during normal BDC communications):

- Waits for BKGD to return to a logic high
- Delays 16 cycles to allow the host to stop driving the high speedup pulse
- Drives BKGD low for 128 BDC clock cycles
- Drives a 1-cycle high speedup pulse to force a fast rise time on BKGD
- Removes all drive to the BKGD pin so it reverts to high impedance

The host measures the low time of this 128-cycle sync response pulse and determines the correct speed for subsequent BDC communications. Typically, the host can determine the correct communication speed within a few percent of the actual target speed and the communication protocol can easily tolerate speed errors of several percent.

17.2.4 BDC Hardware Breakpoint

The BDC includes one relatively simple hardware breakpoint that compares the CPU address bus to a 16-bit match value in the BDCBKPT register. This breakpoint can generate a forced breakpoint or a tagged breakpoint. A forced breakpoint causes the CPU to enter active background mode at the first instruction boundary following any access to the breakpoint address. The tagged breakpoint causes the instruction opcode at the breakpoint address to be tagged so that the CPU will enter active background mode rather than executing that instruction if and when it reaches the end of the instruction queue. This implies that tagged breakpoints can only be placed at the address of an instruction opcode while forced breakpoints can be set at any address.

The breakpoint enable (BKPTEN) control bit in the BDC status and control register (BDCSCR) is used to enable the breakpoint logic (BKPTEN = 1). When BKPTEN = 0, its default value after reset, the breakpoint logic is disabled and no BDC breakpoints are requested regardless of the values in other BDC breakpoint registers and control bits. The force/tag select (FTS) control bit in BDCSCR is used to select forced (FTS = 1) or tagged (FTS = 0) type breakpoints.

The on-chip debug module (DBG) includes circuitry for two additional hardware breakpoints that are more flexible than the simple breakpoint in the BDC module.

A.7 External Oscillator (XOSC) and Internal Clock Source (ICS) Characteristics

Reference Figure A-7 for crystal or resonator circuit.

Table A-8. XOSC and ICS Specifications (Temperature Range = –40 to 125°C Ambient)

Characteristic	Symbol	Min	Typ	Max	Unit
Internal reference frequency — factory trimmed at V_{DD} = 3.6V and temperature = 25°**C**	f_{int_ft}	—	31.25	—	kHz
Oscillator crystal or resonator (EREFS = 1, ERCLKEN = 1) Low range (RANGE = 0) High range (RANGE = 1) FEE or FBE mode [1] High range (RANGE = 1), high gain (HGO = 1), FBELP mode High range (RANGE = 1), low power (HGO = 0), FBELP mode	 f_{lo} f_{hi} f_{hi} f_{hi}	 32 1 1 1	 — — — —	 38.4 5 16 8	 kHz MHz MHz MHz
Load capacitors	C_1 C_2	See Note [2]			
Feedback resistor Low range (32 kHz to 38.4 kHz) High range (1 MHz to 16 MHz)	R_F		10 1		MΩ MΩ
Series resistor — Low range Low Gain (HGO = 0) High Gain (HGO = 1)	R_S	— —	0 100	— —	kΩ
Series resistor — High range Low Gain (HGO = 0) High Gain (HGO = 1) ≥ 8 MHz 4 MHz 1 MHz	R_S	 — — —	 0 0 0	 0 10 20	kΩ
Crystal start-up time [3, 4] Low range, low power Low range, high power High range, low power High range, high power	 t_{CSTL} t_{CSTH}	 — — — —	 200 400 5 15	 — — — —	ms
Internal reference start-up time	t_{IRST}	—	60	100	µs
Square wave input clock frequency (EREFS = 0, ERCLKEN = 1) FEE or FBE mode [2] FBELP mode	f_{extal}	0.03125 0	— —	5 20	MHz MHz
Internal reference frequency - untrimmed[5]	f_{int_ut}	25	32.7	41.66	kHz
Internal reference frequency - trimmed	f_{int_t}	31.25	—	39.06	kHz
DCO output frequency range - untrimmed[5] f_{dco}= 512 * f_{int_ut}	f_{dco_ut}	12.8	16.8	21.33	MHz
DCO output frequency range - trimmed	f_{dco_t}	16	—	20	MHz