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	Not For New Designs
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	-
Peripherals	LVD, POR, PWM
Number of I/O	13
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 4x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	16-SOIC (0.295", 7.50mm Width)
Supplier Device Package	16-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908qy4cdwer

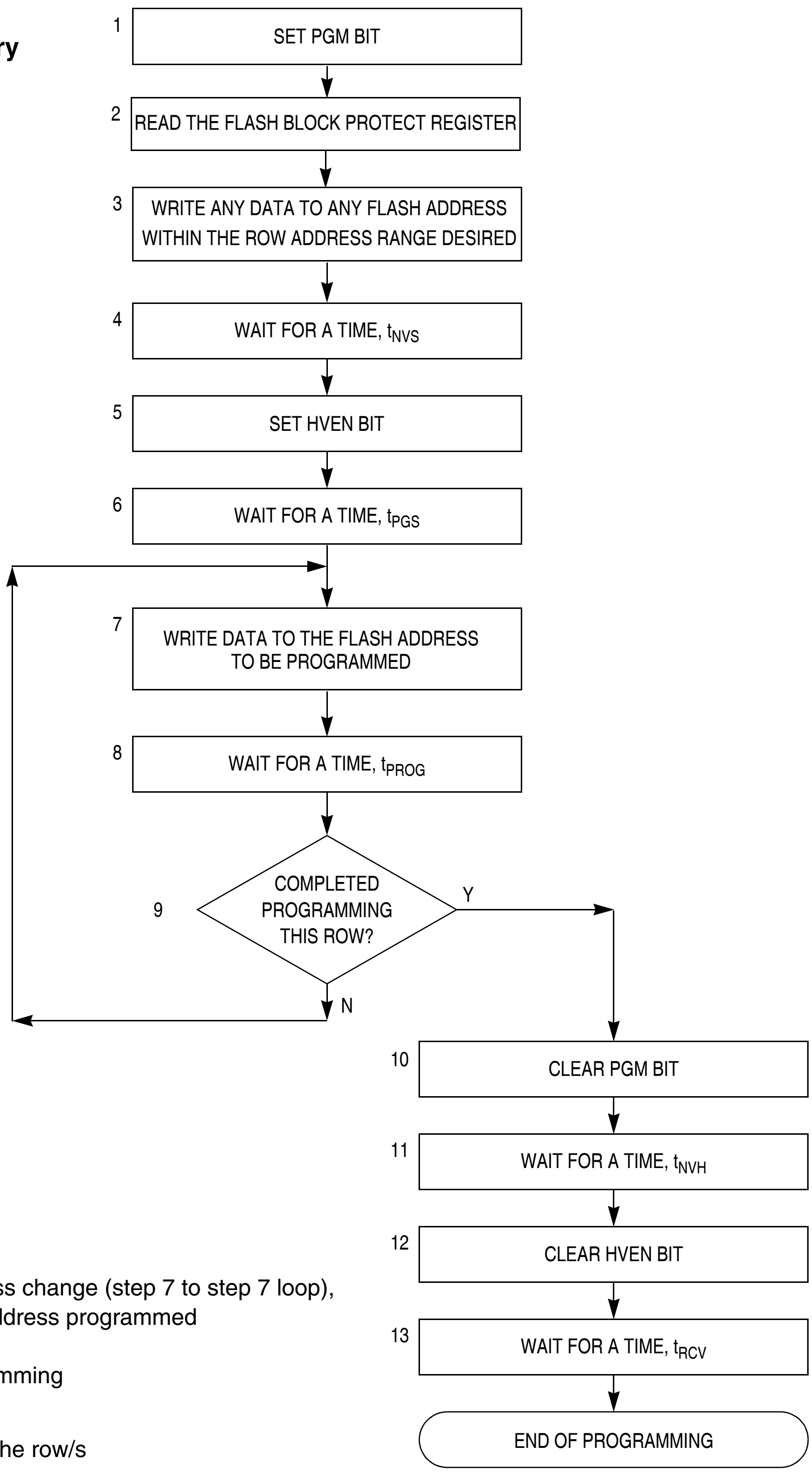
Chapter 11
Oscillator (OSC) Module

11.1 Introduction 93
11.2 Features 93
11.3 Functional Description 93
11.3.1 Internal Signal Definitions 94
11.3.1.1 Oscillator Enable Signal (SIMOSCEN) 94
11.3.1.2 XTAL Oscillator Clock (XTALCLK) 95
11.3.1.3 RC Oscillator Clock (RCCLK) 95
11.3.1.4 Internal Oscillator Clock (INTCLK) 95
11.3.1.5 Bus Clock Times 4 (BUSCLKX4) 95
11.3.1.6 Bus Clock Times 2 (BUSCLKX2) 95
11.3.2 Internal Oscillator 95
11.3.2.1 Internal Oscillator Trimming 95
11.3.2.2 Internal to External Clock Switching 96
11.3.2.3 External to Internal Clock Switching 96
11.3.3 External Oscillator 96
11.3.4 XTAL Oscillator 96
11.3.5 RC Oscillator 98
11.4 Interrupts 98
11.5 Low-Power Modes 98
11.5.1 Wait Mode 98
11.5.2 Stop Mode 98
11.6 OSC During Break Interrupts 99
11.7 I/O Signals 99
11.7.1 Oscillator Input Pin (OSC1) 99
11.7.2 Oscillator Output Pin (OSC2) 99
11.8 Registers 100
11.8.1 Oscillator Status and Control Register 100
11.8.2 Oscillator Trim Register (OSCTRIM) 101

Chapter 12
Input/Output Ports (PORTS)

12.1 Introduction 103
12.2 Unused Pin Termination 103
12.3 Port A 103
12.3.1 Port A Data Register 104
12.3.2 Data Direction Register A 104
12.3.3 Port A Input Pullup Enable Register 105
12.3.4 Port A Summary Table 106
12.4 Port B 106
12.4.1 Port B Data Register 106
12.4.2 Data Direction Register B 107
12.4.3 Port B Input Pullup Enable Register 108
12.4.4 Port B Summary Table 108

Algorithm for Programming a Row (32 Bytes) of FLASH Memory



NOTES:

The time between each FLASH address change (step 7 to step 7 loop), or the time between the last FLASH address programmed to clearing PGM bit (step 7 to step 10) must not exceed the maximum programming time, t_{PROG} max.

This row program algorithm assumes the row/s to be programmed are initially erased.

Figure 2-4. FLASH Programming Flowchart

3.3.4.4 Code Width and Quantization Error

The ADC10 quantizes the ideal straight-line transfer function into 1024 steps (in 10-bit mode). Each step ideally has the same height (1 code) and width. The width is defined as the delta between the transition points from one code to the next. The ideal code width for an N bit converter (in this case N can be 8 or 10), defined as 1LSB, is:

$$1\text{LSB} = (V_{REFH}-V_{REFL}) / 2^N$$

Because of this quantization, there is an inherent quantization error. Because the converter performs a conversion and then rounds to 8 or 10 bits, the code will transition when the voltage is at the midpoint between the points where the straight line transfer function is exactly represented by the actual transfer function. Therefore, the quantization error will be ± 1/2LSB in 8- or 10-bit mode. As a consequence, however, the code width of the first ($000) conversion is only 1/2LSB and the code width of the last ($FF or $3FF) is 1.5LSB.

3.3.4.5 Linearity Errors

The ADC10 may also exhibit non-linearity of several forms. Every effort has been made to reduce these errors but the user should be aware of them because they affect overall accuracy. These errors are:

- Zero-Scale Error (E_{ZS}) (sometimes called offset) — This error is defined as the difference between the actual code width of the first conversion and the ideal code width (1/2LSB). Note, if the first conversion is $001, then the difference between the actual $001 code width and its ideal (1LSB) is used.
- Full-Scale Error (E_{FS}) — This error is defined as the difference between the actual code width of the last conversion and the ideal code width (1.5LSB). Note, if the last conversion is $3FE, then the difference between the actual $3FE code width and its ideal (1LSB) is used.
- Differential Non-Linearity (DNL) — This error is defined as the worst-case difference between the actual code width and the ideal code width for all conversions.
- Integral Non-Linearity (INL) — This error is defined as the highest-value the (absolute value of the) running sum of DNL achieves. More simply, this is the worst-case difference of the actual transition voltage to a given code and its corresponding ideal transition voltage, for all codes.
- Total Unadjusted Error (TUE) — This error is defined as the difference between the actual transfer function and the ideal straight-line transfer function, and therefore includes all forms of error.

3.3.4.6 Code Jitter, Non-Monotonicity and Missing Codes

Analog-to-digital converters are susceptible to three special forms of error. These are code jitter, non-monotonicity, and missing codes.

- Code jitter is when, at certain points, a given input voltage converts to one of two values when sampled repeatedly. Ideally, when the input voltage is infinitesimally smaller than the transition voltage, the converter yields the lower code (and vice-versa). However, even very small amounts of system noise can cause the converter to be indeterminate (between two codes) for a range of input voltages around the transition voltage. This range is normally around ±1/2LSB but will increase with noise.
- Non-monotonicity is defined as when, except for code jitter, the converter converts to a lower code for a higher input voltage. Non-monotonicity is present if the apparent code jitter covers three codes (when the converter's output is indeterminate between three values for a given input voltage) or is greater than 1LSB.
- Missing codes are those which are never converted for any input value. In 8-bit or 10-bit mode, the ADC10 is guaranteed to be monotonic and to have no missing codes.

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 3-4. ADC10 Data Register High (ADRH), 8-Bit Mode

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	AD9	AD8
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 3-5. ADC10 Data Register High (ADRH), 10-Bit Mode

3.8.3 ADC10 Result Low Register (ADRL)

This register holds the LSBs of the result. This register is updated each time a conversion completes. Reading ADRH prevents the ADC10 from transferring subsequent conversion results into the result registers until ADRL is read. If ADRL is not read until the after next conversion is completed, then the intermediate conversion result will be lost. In 8-bit mode, there is no interlocking with ADRH.

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 3-6. ADC10 Data Register Low (ADRL)

3.8.4 ADC10 Clock Register (ADCLK)

This register selects the clock frequency for the ADC10 and the modes of operation.

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	ADLPC	ADIV1	ADIV0	ADICLK	MODE1	MODE0	ADLSMP	ACLKEN
Reset:	0	0	0	0	0	0	0	0

Figure 3-7. ADC10 Clock Register (ADCLK)

ADLPC — ADC10 Low-Power Configuration Bit

ADLPC controls the speed and power configuration of the successive approximation converter. This is used to optimize power consumption when higher sample rates are not required.

1 = Low-power configuration: The power is reduced at the expense of maximum clock speed.
0 = High-speed configuration

LVIPWRD — LVI Power Disable Bit

LVIPWRD disables the LVI module.

1 = LVI module power disabled
0 = LVI module power enabled

LVITRIP — LVI Trip Point Selection Bit

LVITRIP selects the voltage operating mode of the LVI module. The voltage mode selected for the LVI should match the operating V_{DD} for the LVI's voltage trip points for each of the modes.

1 = LVI operates for a 5-V protection
0 = LVI operates for a 3-V protection

NOTE

The LVITRIP bit is cleared by a power-on reset (POR) only. Other resets will leave this bit unaffected.

SSREC — Short Stop Recovery Bit

SSREC enables the CPU to exit stop mode with a delay of 32 BUSCLKX4 cycles instead of a 4096 BUSCLKX4 cycle delay.

1 = Stop mode recovery after 32 BUSCLKX4 cycles
0 = Stop mode recovery after 4096 BUSCLKX4 cycles

NOTE

Exiting stop mode by an LVI reset will result in the long stop recovery.

When using the LVI during normal operation but disabling during stop mode, the LVI will have an enable time of t_{EN}. The system stabilization time for power-on reset and long stop recovery (both 4096 BUSCLKX4 cycles) gives a delay longer than the LVI enable time for these startup scenarios. There is no period where the MCU is not protected from a low-power condition. However, when using the short stop recovery configuration option, the 32 BUSCLKX4 delay must be greater than the LVI's turn on time to avoid a period in startup where the LVI is not protecting the MCU.

STOP — STOP Instruction Enable Bit

STOP enables the STOP instruction.

1 = STOP instruction enabled
0 = STOP instruction treated as illegal opcode

COPD — COP Disable Bit

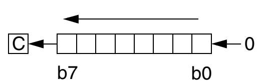
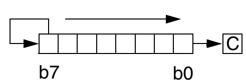
COPD disables the COP module.

1 = COP module disabled
0 = COP module enabled

7.7 Instruction Set Summary

Table 7-1 provides a summary of the M68HC08 instruction set.

Table 7-1. Instruction Set Summary (Sheet 1 of 6)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
ADC #*opr* ADC *opr* ADC *opr* ADC *opr*,X ADC *opr*,X ADC ,X ADC *opr*,SP ADC *opr*,SP	Add with Carry	A ← (A) + (M) + (C)	↕	↕	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A9 B9 C9 D9 E9 F9 9EE9 9ED9	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
ADD #*opr* ADD *opr* ADD *opr* ADD *opr*,X ADD *opr*,X ADD ,X ADD *opr*,SP ADD *opr*,SP	Add without Carry	A ← (A) + (M)	↕	↕	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	AB BB CB DB EB FB 9EEB 9EDB	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
AIS #*opr*	Add Immediate Value (Signed) to SP	SP ← (SP) + (16 « M)	–	–	–	–	–	–	IMM	A7	ii	2
AIX #*opr*	Add Immediate Value (Signed) to H:X	H:X ← (H:X) + (16 « M)	–	–	–	–	–	–	IMM	AF	ii	2
AND #*opr* AND *opr* AND *opr* AND *opr*,X AND *opr*,X AND ,X AND *opr*,SP AND *opr*,SP	Logical AND	A ← (A) & (M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A4 B4 C4 D4 E4 F4 9EE4 9ED4	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
ASL *opr* ASLA ASLX ASL *opr*,X ASL ,X ASL *opr*,SP	Arithmetic Shift Left (Same as LSL)		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff	4 1 1 4 3 5
ASR *opr* ASRA ASRX ASR *opr*,X ASR *opr*,X ASR *opr*,SP	Arithmetic Shift Right		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	37 47 57 67 77 9E67	dd ff ff	4 1 1 4 3 5
BCC *rel*	Branch if Carry Bit Clear	PC ← (PC) + 2 + rel ? (C) = 0	–	–	–	–	–	–	REL	24	rr	3
BCLR *n*, *opr*	Clear Bit n in M	Mn ← 0	–	–	–	–	–	–	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	11 13 15 17 19 1B 1D 1F	dd dd dd dd dd dd dd dd	4 4 4 4 4 4 4 4
BCS *rel*	Branch if Carry Bit Set (Same as BLO)	PC ← (PC) + 2 + *rel* ? (C) = 1	–	–	–	–	–	–	REL	25	rr	3
BEQ *rel*	Branch if Equal	PC ← (PC) + 2 + *rel* ? (Z) = 1	–	–	–	–	–	–	REL	27	rr	3
BGE *opr*	Branch if Greater Than or Equal To (Signed Operands)	PC ← (PC) + 2 + *rel* ? (N ⊕ V) = 0	–	–	–	–	–	–	REL	90	rr	3
BGT *opr*	Branch if Greater Than (Signed Operands)	PC ← (PC) + 2 + *rel* ? (Z) | (N ⊕ V) = 0	–	–	–	–	–	–	REL	92	rr	3
BHCC *rel*	Branch if Half Carry Bit Clear	PC ← (PC) + 2 + *rel* ? (H) = 0	–	–	–	–	–	–	REL	28	rr	3
BHCS *rel*	Branch if Half Carry Bit Set	PC ← (PC) + 2 + *rel* ? (H) = 1	–	–	–	–	–	–	REL	29	rr	3
BHI *rel*	Branch if Higher	PC ← (PC) + 2 + *rel* ? (C) | (Z) = 0	–	–	–	–	–	–	REL	22	rr	3

8.4 Interrupts

The following IRQ source can generate interrupt requests:

- Interrupt flag (IRQF) — The IRQF bit is set when the $\overline{\text{IRQ}}$ pin is asserted based on the IRQ mode. The IRQ interrupt mask bit, IMASK, is used to enable or disable IRQ interrupt requests.

8.5 Low-Power Modes

The WAIT and STOP instructions put the MCU in low power-consumption standby modes.

8.5.1 Wait Mode

The IRQ module remains active in wait mode. Clearing IMASK in INTSCR enables IRQ interrupt requests to bring the MCU out of wait mode.

8.5.2 Stop Mode

The IRQ module remains active in stop mode. Clearing IMASK in INTSCR enables IRQ interrupt requests to bring the MCU out of stop mode.

8.6 IRQ Module During Break Interrupts

The system integration module (SIM) controls whether status bits in other modules can be cleared during the break state. The BCFE bit in the break flag control register (BFCR) enables software to clear status bits during the break state. See BFCR in the SIM section of this data sheet.

To allow software to clear status bits during a break interrupt, write a 1 to BCFE. If a status bit is cleared during the break state, it remains cleared when the MCU exits the break state.

To protect status bits during the break state, write a 0 to BCFE. With BCFE cleared (its default state), software can read and write registers during the break state without affecting status bits. Some status bits have a two-step read/write clearing procedure. If software does the first step on such a bit before the break, the bit cannot change during the break state as long as BCFE is cleared. After the break, doing the second step clears the status bit.

8.7 I/O Signals

The IRQ module does not share its pin with any module on this MCU.

8.7.1 IRQ Input Pins ($\overline{\text{IRQ}}$)

The $\overline{\text{IRQ}}$ pin provides a maskable external interrupt source. The $\overline{\text{IRQ}}$ pin contains an internal pullup device.

Chapter 9
Keyboard Interrupt Module (KBI)

9.1 Introduction

The keyboard interrupt module (KBI) provides independently maskable external interrupts.

The KBI shares its pins with general-purpose input/output (I/O) port pins. See Figure 9-1 for port location of these shared pins.

9.2 Features

Features of the keyboard interrupt module include:

- Keyboard interrupt pins with separate keyboard interrupt enable bits and one keyboard interrupt mask
- Programmable edge-only or edge and level interrupt sensitivity
- Edge sensitivity programmable for rising or falling edge
- Level sensitivity programmable for high or low level
- Pullup or pulldown device automatically enabled based on the polarity of edge or level detect
- Exit from low-power modes

9.3 Functional Description

The keyboard interrupt module controls the enabling/disabling of interrupt functions on the KBI pins. These pins can be enabled/disabled independently of each other. See Figure 9-2.

9.3.1 Keyboard Operation

Writing to the KBIEx bits in the keyboard interrupt enable register (KBIER) independently enables or disables each KBI pin. The polarity of the keyboard interrupt is controlled using the KBIPx bits in the keyboard interrupt polarity register (KBIPR). Edge-only or edge and level sensitivity is controlled using the MODEK bit in the keyboard status and control register (KBISCR).

Enabling a keyboard interrupt pin also enables its internal pullup or pulldown device based on the polarity enabled. On falling edge or low level detection, a pullup device is configured. On rising edge or high level detection, a pulldown device is configured.

The keyboard interrupt latch is set when one or more enabled keyboard interrupt inputs are asserted.

- If the keyboard interrupt sensitivity is edge-only, for KBIPx = 0, a falling (for KBIPx = 1, a rising) edge on a keyboard interrupt input does not latch an interrupt request if another enabled keyboard pin is already asserted. To prevent losing an interrupt request on one input because another input remains asserted, software can disable the latter input while it is asserted.
- If the keyboard interrupt is edge and level sensitive, an interrupt request is present as long as any enabled keyboard interrupt input is asserted.

9.8.2 Keyboard Interrupt Enable Register (KBIER)

KBIER enables or disables each keyboard interrupt pin.

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	AWUIE	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 9-4. Keyboard Interrupt Enable Register (KBIER)

KBIE5–KBIE0 — Keyboard Interrupt Enable Bits

Each of these read/write bits enables the corresponding keyboard interrupt pin to latch KBI interrupt requests.

1 = KBIx pin enabled as keyboard interrupt pin
0 = KBIx pin not enabled as keyboard interrupt pin

NOTE

AWUIE bit is not used in conjunction with the keyboard interrupt feature. To see a description of this bit, see Chapter 4 Auto Wakeup Module (AWU).

9.8.3 Keyboard Interrupt Polarity Register (KBIPR)

KBIPR determines the polarity of the enabled keyboard interrupt pin and enables the appropriate pullup or pulldown device.

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	KBIP5	KBIP4	KBIP3	KBIP2	KBIP1	KBIP0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 9-5. Keyboard Interrupt Polarity Register (KBIPR)

KBIP5–KBIP0 — Keyboard Interrupt Polarity Bits

Each of these read/write bits enables the polarity of the keyboard interrupt detection.

1 = Keyboard polarity is high level and/or rising edge
0 = Keyboard polarity is low level and/or falling edge

Chapter 11
Oscillator (OSC) Module

11.1 Introduction

The oscillator (OSC) module is used to provide a stable clock source for the MCU system and bus.

The OSC shares its pins with general-purpose input/output (I/O) port pins. See Figure 11-1 for port location of these shared pins. The OSC2EN bit is located in the port A pull enable register (PTAPUEN) on this MCU. See Chapter 12 Input/Output Ports (PORTS) for information on PTAPUEN register.

11.2 Features

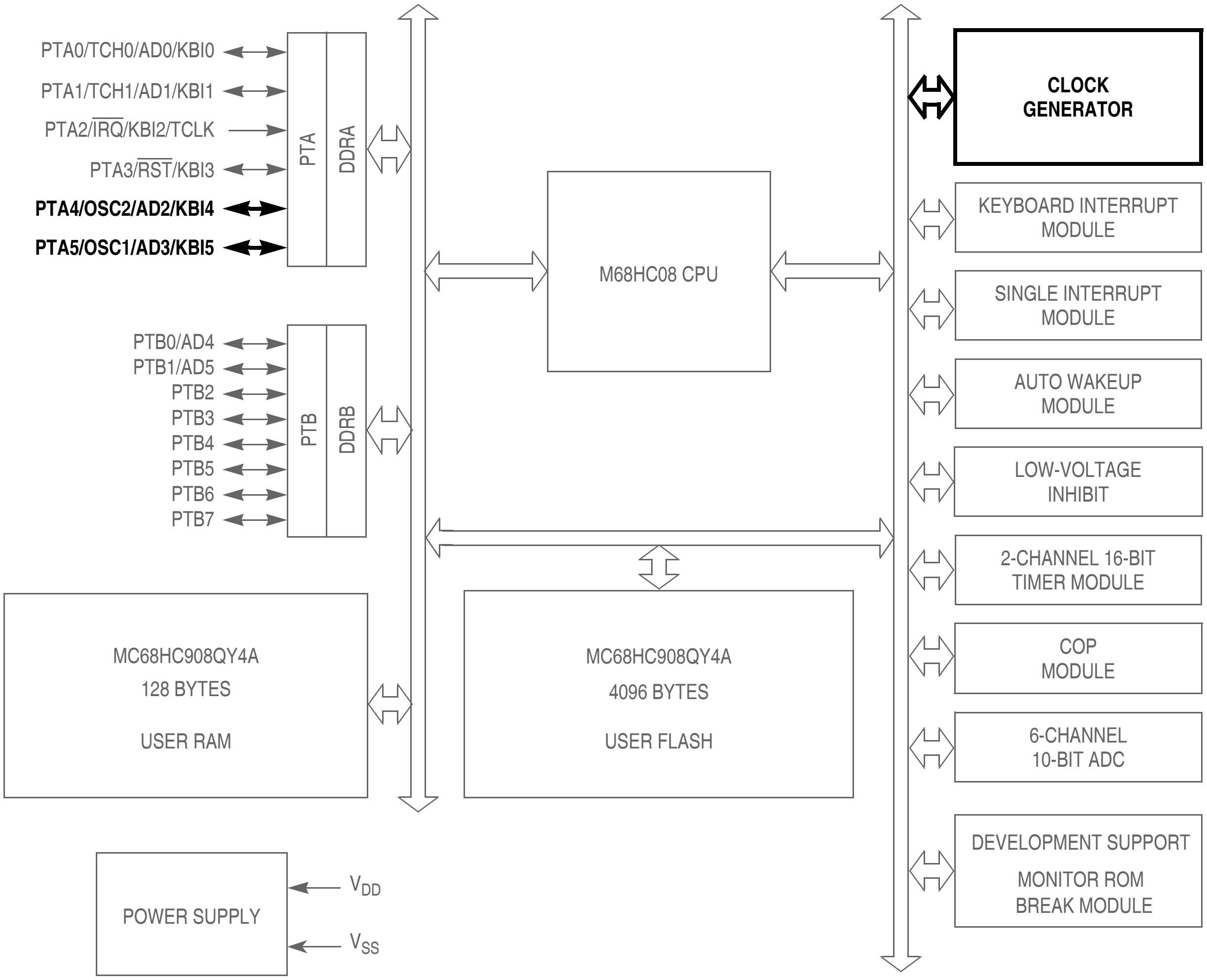
The bus clock frequency is one fourth of any of these clock source options:

1. Internal oscillator: An internally generated, fixed frequency clock, trimmable to ± 0.4%. There are three choices for the internal oscillator,12.8 MHz, 8 MHz, or 4 MHz. The 12.8-MHz internal oscillator is the default option out of reset.
2. External oscillator: An external clock that can be driven directly into OSC1.
3. External RC: A built-in oscillator module (RC oscillator) that requires an external R connection only. The capacitor is internal to the chip.
4. External crystal: A built-in XTAL oscillator that requires an external crystal or ceramic-resonator. There are three crystal frequency ranges supported, 8–32 MHz, 1–8 MHz, and 32–100 kHz.

11.3 Functional Description

The oscillator contains these major subsystems:

- Internal oscillator circuit
- Internal or external clock switch control
- External clock circuit
- External crystal circuit
- External RC clock circuit



RST, IRQ: Pins have internal pull up device
All port pins have programmable pull up device
PTA[0:5]: Higher current sink and source capability
PTB[0:7]: Not available on 8-pin devices

Figure 11-1. Block Diagram Highlighting OSC Block and Pins

11.3.1 Internal Signal Definitions

The following signals and clocks are used in the functional description and figures of the OSC module.

11.3.1.1 Oscillator Enable Signal (SIMOSCEN)

The SIMOSCEN signal comes from the system integration module (SIM) and disables the XTAL oscillator circuit, the RC oscillator, or the internal oscillator in stop mode. OSCENINSTOP in the configuration register can be used to override this signal.

Chapter 12
Input/Output Ports (PORTS)

12.1 Introduction

The MC68HC08QY1A, MC68HC08QY2A and MC68HC08QY4A have thirteen bidirectional input-output (I/O) pins and one input only pin. The MC68HC08QT1A, MC68HC08QT2A and MC68HC08QT4A has five bidirectional I/O pins and one input only pin. All I/O pins are programmable as inputs or outputs.

12.2 Unused Pin Termination

Input pins and I/O port 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:

1. Configuring unused pins as outputs and driving high or low;
2. Configuring unused pins as inputs and enabling internal pull-ups;
3. Configuring unused pins as inputs and using external pull-up or pull-down resistors.

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

Since some general-purpose I/O pins are not available on all packages, these pins must be terminated as well. Either method 1 or 2 above are appropriate.

12.3 Port A

Port A is an 6-bit special function port that shares its pins with the keyboard interrupt (KBI) module (see Chapter 9 Keyboard Interrupt Module (KBI), the 2-channel timer interface module (TIM) (see Chapter 14 Timer Interface Module (TIM)), the 10-bit ADC (see Chapter 3 Analog-to-Digital Converter (ADC10) Module), the external interrupt (IRQ) pin (see Chapter 8 External Interrupt (IRQ)), the reset (RST) pin enabled using a configuration register (see Chapter 5 Configuration Register (CONFIG)) and the oscillator pins (see Chapter 11 Oscillator (OSC) Module).

Each port A pin also has a software configurable pullup device if the corresponding port pin is configured as an input port.

NOTE

PTA2 is input only.

When the $\overline{IRQ}$ function is enabled in the configuration register 2 (CONFIG2), bit 2 of the port A data register (PTA) will always read a logic 0. In this case, the BIH and BIL instructions can be used to read the logic level on the PTA2 pin. When the $\overline{IRQ}$ function is disabled, these instructions will behave as if the PTA2 pin is a logic 1. However, reading bit 2 of PTA will read the actual logic level on the pin.

12.4.2 Data Direction Register B

Data direction register B (DDRB) determines whether each port B pin is an input or an output. Writing a 1 to a DDRB bit enables the output buffer for the corresponding port B pin; a 0 disables the output buffer.

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
Reset:	0	0	0	0	0	0	0	0

Figure 12-6. Data Direction Register B (DDRB)

DDRB[7:0] — Data Direction Register B Bits

These read/write bits control port B data direction. Reset clears DDRB[7:0], configuring all port B pins as inputs.

1 = Corresponding port B pin configured as output

0 = Corresponding port B pin configured as input

NOTE

Avoid glitches on port B pins by writing to the port B data register before changing data direction register B bits from 0 to 1. Figure 12-7 shows the port B I/O logic.

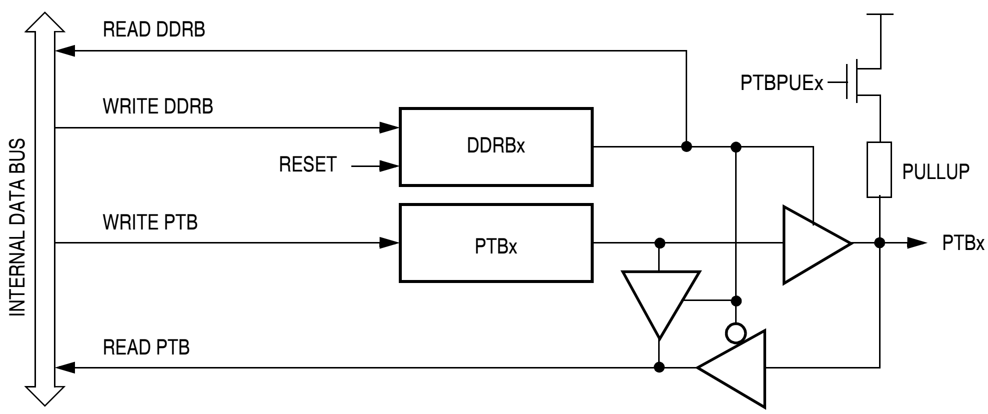


Figure 12-7. Port B I/O Circuit

When DDRBx is a 1, reading PTB reads the PTBx data latch. When DDRBx is a 0, reading PTB reads the logic level on the PTBx pin. The data latch can always be written, regardless of the state of its data direction bit.

13.6.1.2 SWI Instruction

The SWI instruction is a non-maskable instruction that causes an interrupt regardless of the state of the interrupt mask (I bit) in the condition code register.

NOTE

A software interrupt pushes PC onto the stack. A software interrupt does ***not*** *push PC – 1, as a hardware interrupt does.*

13.6.2 Interrupt Status Registers

The flags in the interrupt status registers identify maskable interrupt sources. Table 13-3 summarizes the interrupt sources and the interrupt status register flags that they set. The interrupt status registers can be useful for debugging.

Table 13-3. Interrupt Sources

Priority	Source	Flag	Mask(1)	INT Register Flag	Vector Address
Highest	Reset	—	—	—	\$FFFE–\$FFFF
↑	SWI instruction	—	—	—	\$FFFC–\$FFFD
	$\overline{\text{IRQ}}$ pin	IRQF	IMASK	IF1	\$FFFA–\$FFFB
	Timer channel 0 interrupt	CH0F	CH0IE	IF3	\$FFF6–\$FFF7
	Timer channel 1 interrupt	CH1F	CH1IE	IF4	\$FFF4–\$FFF5
	Timer overflow interrupt	TOF	TOIE	IF5	\$FFF2–\$FFF3
↓	Keyboard interrupt	KEYF	IMASKK	IF14	\$FFE0–\$FFE1
Lowest	ADC conversion complete interrupt	COCO	AIEN	IF15	\$FFDE–\$FFDF

1. The I bit in the condition code register is a global mask for all interrupt sources except the SWI instruction.

the end of the current pulse) could cause two output compares to occur in the same counter overflow period.

14.3.3.2 Buffered Output Compare

Channels 0 and 1 can be linked to form a buffered output compare channel whose output appears on the TCH0 pin. The TIM channel registers of the linked pair alternately control the output.

Setting the MS0B bit in TIM channel 0 status and control register (TSC0) links channel 0 and channel 1. The output compare value in the TIM channel 0 registers initially controls the output on the TCH0 pin. Writing to the TIM channel 1 registers enables the TIM channel 1 registers to synchronously control the output after the TIM overflows. At each subsequent overflow, the TIM channel registers (0 or 1) that control the output are the ones written to last. TSC0 controls and monitors the buffered output compare function, and TIM channel 1 status and control register (TSC1) is unused. While the MS0B bit is set, the channel 1 pin, TCH1, is available as a general-purpose I/O pin.

NOTE

In buffered output compare operation, do not write new output compare values to the currently active channel registers. User software should track the currently active channel to prevent writing a new value to the active channel. Writing to the active channel registers is the same as generating unbuffered output compares.

14.3.4 Pulse Width Modulation (PWM)

By using the toggle-on-overflow feature with an output compare channel, the TIM can generate a PWM signal. The value in the TIM counter modulo registers determines the period of the PWM signal. The channel pin toggles when the counter reaches the value in the TIM counter modulo registers. The time between overflows is the period of the PWM signal.

As Figure 14-3 shows, the output compare value in the TIM channel registers determines the pulse width of the PWM signal. The time between overflow and output compare is the pulse width. Program the TIM to clear the channel pin on output compare if the polarity of the PWM pulse is 1 (ELSxA = 0). Program the TIM to set the pin if the polarity of the PWM pulse is 0 (ELSxA = 1).

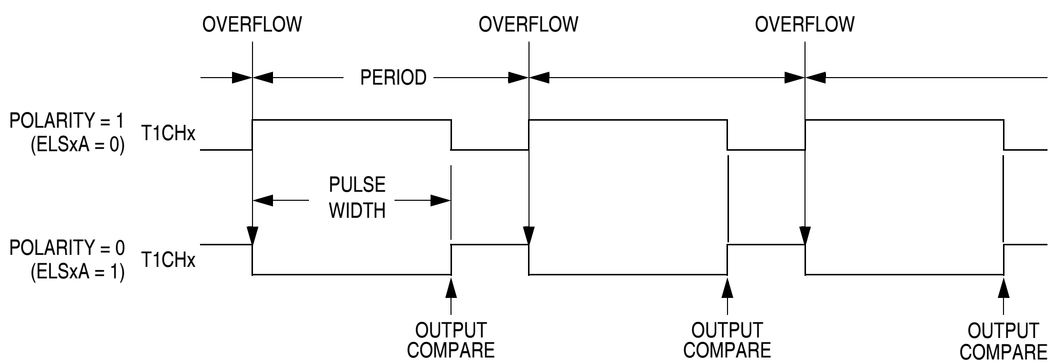


Figure 14-3. PWM Period and Pulse Width

The value in the TIM counter modulo registers and the selected prescaler output determines the frequency of the PWM output The frequency of an 8-bit PWM signal is variable in 256 increments. Writing $00FF (255) to the TIM counter modulo registers produces a PWM period of 256 times the internal bus clock period if the prescaler select value is 000. See 14.8.1 TIM Status and Control Register.

Chapter 16
Electrical Specifications

16.1 Introduction

This section contains electrical and timing specifications.

16.2 Absolute Maximum Ratings

Maximum ratings are the extreme limits to which the microcontroller unit (MCU) can be exposed without permanently damaging it.

NOTE

This device is not guaranteed to operate properly at the maximum ratings. Refer to 16.5 5-V DC Electrical Characteristics and 16.8 3-V DC Electrical Characteristics for guaranteed operating conditions.

Characteristic[1]	Symbol	Value	Unit
Supply voltage	V_{DD}	–0.3 to +6.0	V
Input voltage	V_{IN}	V_{SS} –0.3 to V_{DD} +0.3	V
Mode entry voltage, $\overline{\text{IRQ}}$ pin	V_{TST}	V_{SS} –0.3 to +9.1	V
Maximum current per pin excluding PTA0–PTA5, V_{DD}, and V_{SS}	I	±15	mA
Maximum current for pins PTA0–PTA5	I_{PTA0}—I_{PTA5}	±25	mA
Storage temperature	T_{STG}	–55 to +150	°C
Maximum current out of V_{SS}	I_{MVSS}	100	mA
Maximum current into V_{DD}	I_{MVDD}	100	mA

1. Voltages references to V_{SS}.

NOTE

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{IN} and V_{OUT} be constrained to the range $V_{SS} \leq (V_{IN}$ or $V_{OUT}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (for example, either V_{SS} or V_{DD}.)

16.5 5-V DC Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output high voltage I_{Load} = –2.0 mA, all I/O pins I_{Load} = –10.0 mA, all I/O pins I_{Load} = –15.0 mA, PTA0, PTA1, PTA3–PTA5 only	V_{OH}	 V_{DD}–0.4 V_{DD}–1.5 V_{DD}–0.8	 — — —	 — — —	V
Maximum combined I_{OH} (all I/O pins)	I_{OHT}	—	—	50	mA
Output low voltage I_{Load} = 1.6 mA, all I/O pins I_{Load} = 10.0 mA, all I/O pins I_{Load} = 15.0 mA, PTA0, PTA1, PTA3–PTA5 only	V_{OL}	 — — —	 — — —	 0.4 1.5 0.8	V
Maximum combined I_{OL} (all I/O pins)	I_{OHL}	—	—	50	mA
Input high voltage PTA0–PTA5, PTB0–PTB7	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input low voltage PTA0–PTA5, PTB0–PTB7	V_{IL}	V_{SS}	—	0.3 x V_{DD}	V
Input hysteresis[3]	V_{HYS}	0.06 x V_{DD}	—	—	V
DC injection current, all ports[4]	I_{INJ}	–2	—	+2	mA
Total dc current injection (sum of all I/O)[4]	I_{INJTOT}	–25	—	+25	mA
Ports Hi-Z leakage current	I_{IL}	–1	±0.1	+1	µA
Capacitance Ports (as input)[3]	C_{IN}	—	—	8	pF
POR rearm voltage	V_{POR}	750	—	—	mV
POR rise time ramp rate[3][5]	R_{POR}	0.035	—	—	V/ms
Monitor mode entry voltage [3]	V_{TST}	$V_{DD + 2.5}$	—	9.1	V
Pullup resistors[6] PTA0–PTA5, PTB0–PTB7	R_{PU}	16	26	36	kΩ
Pulldown resistors[7] PTA0–PTA5	R_{PD}	16	26	36	kΩ
Low-voltage inhibit reset, trip falling voltage	V_{TRIPF}	3.90	4.20	4.50	V
Low-voltage inhibit reset, trip rising voltage	V_{TRIPR}	4.00	4.30	4.60	V
Low-voltage inhibit reset/recover hysteresis	V_{HYS}	—	100	—	mV

1. V_{DD} = 4.5 to 5.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted.
2. Typical values reflect average measurements at midpoint of voltage range, 25•C only.
3. Values are based on characterization results, not tested in production.
4. Guaranteed by design, not tested in production.
5. If minimum V_{DD} is not reached before the internal POR reset is released, the LVI will hold the part in reset until minimum V_{DD} is reached.
6. R_{PU} is measured at V_{DD} = 5.0 V.
7. R_{PD} is measured at V_{DD} = 5.0 V, Pulldown resistors only available when KBIx is enabled with KBIxPOL =1.

16.7 5-V Control Timing

Characteristic[1]	Symbol	Min	Max	Unit
Internal operating frequency	f_{OP} (f_{BUS})	—	8	MHz
Internal clock period ($1/f_{OP}$)	t_{cyc}	125	—	ns
$\overline{\text{RST}}$ input pulse width low[2]	t_{RL}	100	—	ns
$\overline{\text{IRQ}}$ interrupt pulse width low (edge-triggered)[2]	t_{ILIH}	100	—	ns
$\overline{\text{IRQ}}$ interrupt pulse period[2]	t_{ILIL}	Note[3]	—	t_{cyc}

1. V_{DD} = 4.5 to 5.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H; timing shown with respect to 20% V_{DD} and 70% V_{SS}, unless otherwise noted.
2. Values are based on characterization results, not tested in production.
3. The minimum period is the number of cycles it takes to execute the interrupt service routine plus 1 t_{cyc}.

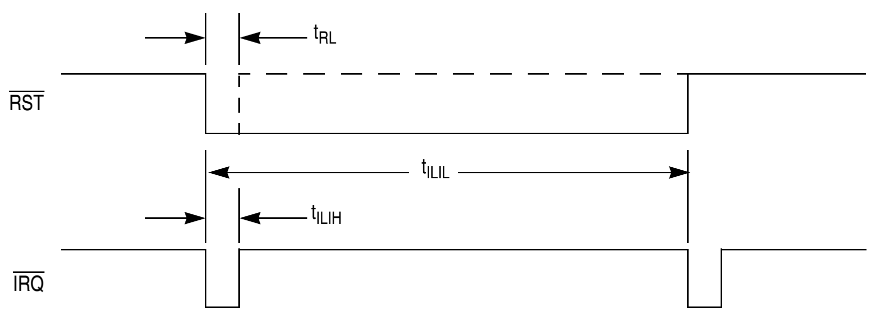


Figure 16-3. $\overline{\text{RST}}$ and $\overline{\text{IRQ}}$ Timing

16.11 Oscillator Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
Internal oscillator frequency[1] ICFS1:ICFS0 = 00 ICFS1:ICFS0 = 01 ICFS1:ICFS0 = 10 (not allowed if V_{DD} <2.7V)	f_{INTCLK}	— — —	4 8 12.8	— — —	MHz
Trim accuracy[2][3]	Δ_{TRIM_ACC}	—	± 0.4	—	%
Deviation from trimmed Internal oscillator[3][4] 4, 8, 12.8MHz, V_{DD} ± 10%, 0 to 70°C 4, 8, 12.8MHz, V_{DD} ± 10%, –40 to 125°C	Δ_{INT_TRIM}	— —	± 2 —	— ± 5	%
External RC oscillator frequency, RCCLK [1][3]	f_{RCCLK}	2	—	10	MHz
External clock reference frequencyy[1][5][6] $V_{DD} \geq 4.5V$ $V_{DD} < 4.5V$	$f_{OSCXCLK}$	dc dc	—	32 16	MHz
RC oscillator external resistor[3] V_{DD} = 5 V V_{DD} = 3 V	R_{EXT}		See Figure 16-7 See Figure 16-8		—
Crystal frequency, XTALCLK[1][7][8] ECFS1:ECFS0 = 00 ($V_{DD} \geq 4.5$ V) ECFS1:ECFS0 = 00 ECFS1:ECFS0 = 01 ECFS1:ECFS0 = 10	$f_{OSCXCLK}$	8 8 1 30	—	32 16 8 100	MHz MHz MHz kHz
ECFS1:ECFS0 = 00 [9] Feedback bias resistor Crystal load capacitance[10] Crystal capacitors[10]	R_B C_L C_1,C_2	— — —	1 20 (2 x C_L) – 5pF	— — —	MΩ pF pF
ECFS1:ECFS0 = 01[9] Crystal series damping resistor $f_{OSCXCLK}$ = 1 MHz $f_{OSCXCLK}$ = 4 MHz $f_{OSCXCLK}$ = 8 MHz Feedback bias resistor Crystal load capacitance[10] Crystal capacitors[10]	R_S R_B C_L C_1,C_2	— — — — — —	20 10 0 5 18 (2 x C_L) –10 pF	— — — — — —	kΩ kΩ kΩ MΩ pF pF
AWU module internal RC oscillator frequency	f_{INTRC}	—	32	—	kHz

1. Bus frequency, f_{OP}, is oscillator frequency divided by 4.
2. Factory trimmed to provided 12.8MHz accuracy requirement (± 5%, @25•C) for forced monitor mode communication. User should trim in-circuit to obtain the most accurate internal oscillator frequency for the application.
3. Values are based on characterization results, not tested in production.
4. Deviation values assumes trimming in target application @25•C and midpoint of voltage range, for example 5.0 V for 5 V ± 10% operation.
5. No more than 10% duty cycle deviation from 50%.
6. When external oscillator clock is greater than 1MHz, ECFS1:ECFS0 must be 00 or 01
7. Use fundamental mode only, do **not** use overtone crystals or overtone ceramic resonators
8. Due to variations in electrical properties of external components such as, ESR and Load Capacitance, operation above 16 MHz is not guaranteed for all crystals or ceramic resonators. Operation above 16 MHz requires that a Negative Resistance Margin (NRM) characterization and component optimization be performed by the crystal or ceramic resonator vendor for every different type of crystal or ceramic resonator which will be used. This characterization and optimization must be performed at the extremes of voltage and temperature which will be applied to the microcontroller in the application. The NRM must meet or exceed 10x the maximum ESR of the crystal or ceramic resonator for acceptable performance.
9. Do not use damping resistor when ECFS1:ECFS0 = 00 or 10
10. Consult crystal vendor data sheet.

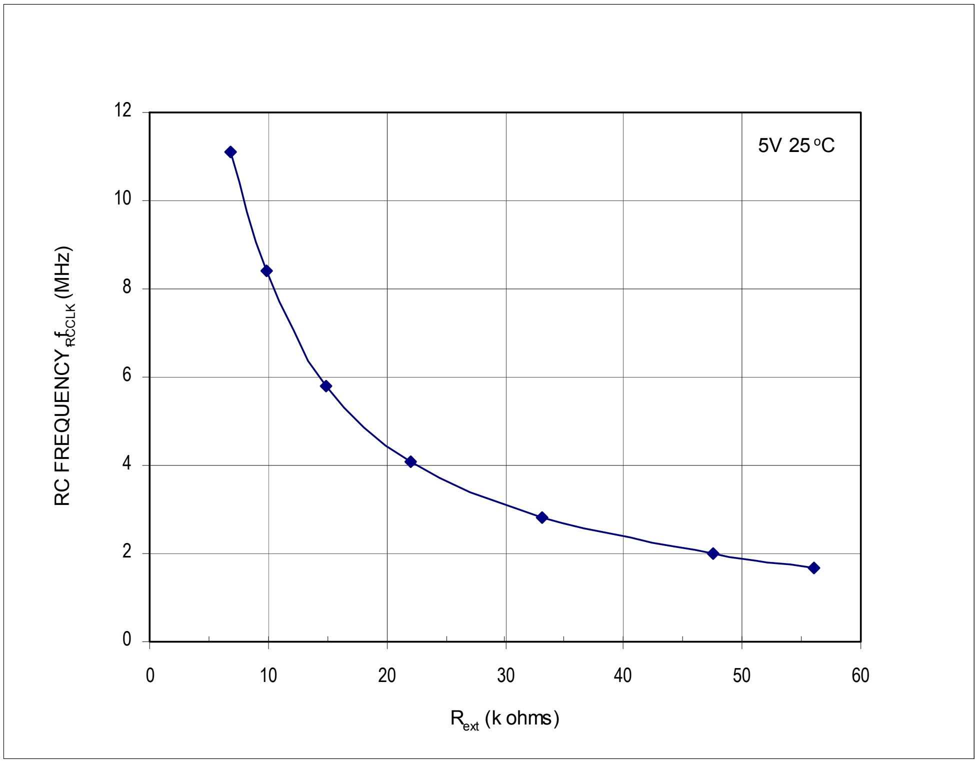


Figure 16-7. RC versus Frequency (5 Volts @ 25°C)

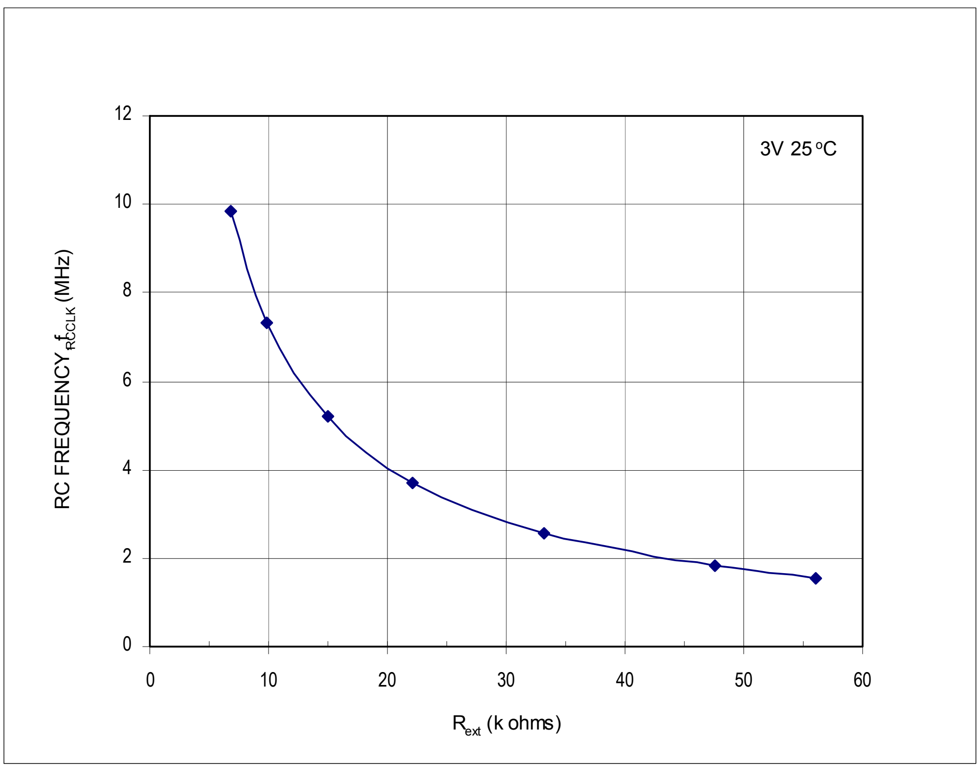


Figure 16-8. RC versus Frequency (3 Volts @ 25°C)