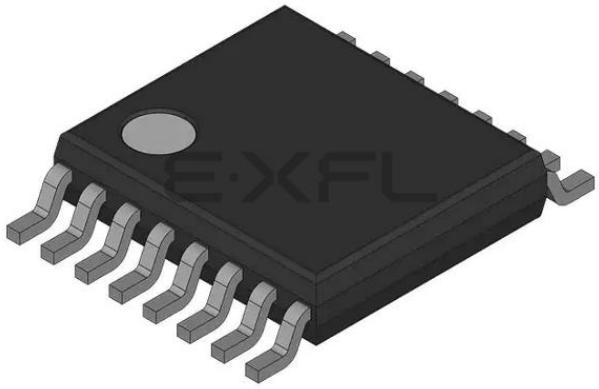


Welcome to E-XFL.COM

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	Active
Core Processor	HC08
Core Size	8-Bit
Speed	2MHz
Connectivity	-
Peripherals	LVD, POR, PWM
Number of I/O	13
Program Memory Size	1.5KB (1.5K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.4V ~ 3.6V
Data Converters	A/D 4x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	16-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	16-TSSOP
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mcl908qy2cdte

MC68HLC908QY4
MC68HLC908QT4
MC68HLC908QY2
MC68HLC908QT2
MC68HLC908QY1
MC68HLC908QT1

Data Sheet

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify you have the latest information available, refer to:

http://freescale.com/

3.3.3 Conversion Time

Sixteen ADC internal clocks are required to perform one conversion. The ADC starts a conversion on the first rising edge of the ADC internal clock immediately following a write to the ADSCR. If the ADC internal clock is selected to run at 1 MHz, then one conversion will take 16 µs to complete. With a 1-MHz ADC internal clock the maximum sample rate is 62.5 kHz.

$$\text{Conversion Time} = \frac{\text{16 ADC Clock Cycles}}{\text{ADC Clock Frequency}}$$

$$\text{Number of Bus Cycles} = \text{Conversion Time} \times \text{Bus Frequency}$$

3.3.4 Continuous Conversion

In the continuous conversion mode (ADCO = 1), the ADC continuously converts the selected channel filling the ADC data register (ADR) with new data after each conversion. Data from the previous conversion will be overwritten whether that data has been read or not. Conversions will continue until the ADCO bit is cleared. The COCO bit (ADSCR, $003C) is set after each conversion and will stay set until the next read of the ADC data register.

When a conversion is in process and the ADSCR is written, the current conversion data should be discarded to prevent an incorrect reading.

3.3.5 Accuracy and Precision

The conversion process is monotonic and has no missing codes.

3.4 Interrupts

When the AIEN bit is set, the ADC module is capable of generating a central processor unit (CPU) interrupt after each ADC conversion. A CPU interrupt is generated if the COCO bit is at 0. The COCO bit is not used as a conversion complete flag when interrupts are enabled.

3.5 Low-Power Modes

The following subsections describe the ADC in low-power modes.

3.5.1 Wait Mode

The ADC continues normal operation during wait mode. Any enabled CPU interrupt request from the ADC can bring the microcontroller unit (MCU) out of wait mode. If the ADC is not required to bring the MCU out of wait mode, power down the ADC by setting the CH[4:0] bits in ADSCR to 1s before executing the WAIT instruction.

3.5.2 Stop Mode

The ADC module is inactive after the execution of a STOP instruction. Any pending conversion is aborted. ADC conversions resume when the MCU exits stop mode. Allow one conversion cycle to stabilize the analog circuitry before using ADC data after exiting stop mode.

3.6 Input/Output Signals

The ADC module has four channels that are shared with I/O port A.

ADC voltage in (ADCVIN) is the input voltage signal from one of the four ADC channels to the ADC module.

3.7 Input/Output Registers

These I/O registers control and monitor ADC operation:

- ADC status and control register (ADSCR)
- ADC data register (ADR)
- ADC clock register (ADICLK)

3.7.1 ADC Status and Control Register

The following paragraphs describe the function of the ADC status and control register (ADSCR). When a conversion is in process and the ADSCR is written, the current conversion data should be discarded to prevent an incorrect reading.

Address: $003C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	COCO	AIEN	ADCO	CH4	CH3	CH2	CH1	CH0
Write:	R							
Reset:	0	0	0	1	1	1	1	1

R = Reserved

Figure 3-3. ADC Status and Control Register (ADSCR)

COCO — Conversions Complete Bit

In non-interrupt mode (AIEN = 0), COCO is a read-only bit that is set at the end of each conversion. COCO will stay set until cleared by a read of the ADC data register. Reset clears this bit.

In interrupt mode (AIEN = 1), COCO is a read-only bit that is not set at the end of a conversion. It always reads as a 0.

1 = Conversion completed (AIEN = 0)
0 = Conversion not completed (AIEN = 0) or CPU interrupt enabled (AIEN = 1)

NOTE

The write function of the COCO bit is reserved. When writing to the ADSCR register, always have a 0 in the COCO bit position.

AIEN — ADC Interrupt Enable Bit

When this bit is set, an interrupt is generated at the end of an ADC conversion. The interrupt signal is cleared when ADR is read or ADSCR is written. Reset clears the AIEN bit.

1 = ADC interrupt enabled
0 = ADC interrupt disabled

ADCO — ADC Continuous Conversion Bit

When set, the ADC will convert samples continuously and update ADR at the end of each conversion. Only one conversion is allowed when this bit is cleared. Reset clears the ADCO bit.

1 = Continuous ADC conversion
0 = One ADC conversion

Chapter 6
Computer Operating Properly (COP)

6.1 Introduction

The computer operating properly (COP) module contains a free-running counter that generates a reset if allowed to overflow. The COP module helps software recover from runaway code. Prevent a COP reset by clearing the COP counter periodically. The COP module can be disabled through the COPD bit in the configuration 1 (CONFIG1) register.

6.2 Functional Description

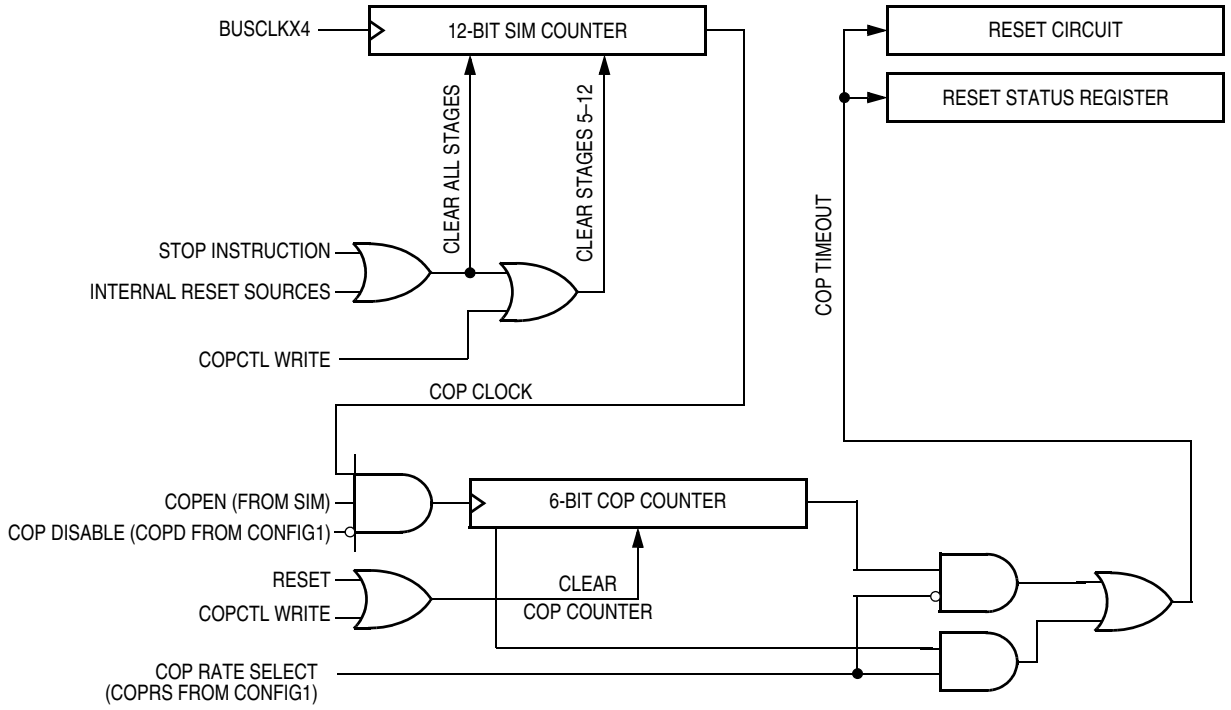


Figure 6-1. COP Block Diagram

The COP counter is a free-running 6-bit counter preceded by the 12-bit system integration module (SIM) counter. If not cleared by software, the COP counter overflows and generates an asynchronous reset after 262,128 or 8176 BUSCLKX4 cycles; depending on the state of the COP rate select bit, COPRS, in configuration register 1. With a 262,128 BUSCLKX4 cycle overflow option, the internal 4.0-MHz oscillator gives a COP timeout period of 65.53 ms. Writing any value to location $FFFF before an overflow occurs prevents a COP reset by clearing the COP counter and stages 12–5 of the SIM counter.

NOTE

Service the COP immediately after reset and before entering or after exiting stop mode to guarantee the maximum time before the first COP counter overflow.

A COP reset pulls the $\overline{\text{RST}}$ pin low (if the RSTEN bit is set in the CONFIG1 register) for 32 × BUSCLKX4 cycles and sets the COP bit in the reset status register (RSR). See 13.8.1 SIM Reset Status Register.

NOTE

Place COP clearing instructions in the main program and not in an interrupt subroutine. Such an interrupt subroutine could keep the COP from generating a reset even while the main program is not working properly.

6.3 I/O Signals

The following paragraphs describe the signals shown in Figure 6-1.

6.3.1 BUSCLKX4

BUSCLKX4 is the oscillator output signal. BUSCLKX4 frequency is equal to the internal oscillator frequency, crystal frequency, or the RC-oscillator frequency.

6.3.2 STOP Instruction

The STOP instruction clears the SIM counter.

6.3.3 COPCTL Write

Writing any value to the COP control register (COPCTL) (see 6.4 COP Control Register) clears the COP counter and clears stages 12–5 of the SIM counter. Reading the COP control register returns the low byte of the reset vector.

6.3.4 Power-On Reset

The power-on reset (POR) circuit in the SIM clears the SIM counter 4096 × BUSCLKX4 cycles after power up.

6.3.5 Internal Reset

An internal reset clears the SIM counter and the COP counter.

6.3.6 COPD (COP Disable)

The COPD signal reflects the state of the COP disable bit (COPD) in the configuration register 1 (CONFIG1). See Chapter 5 Configuration Register (CONFIG).

Chapter 7
Central Processor Unit (CPU)

7.1 Introduction

The M68HC08 CPU (central processor unit) is an enhanced and fully object-code-compatible version of the M68HC05 CPU. The *CPU08 Reference Manual* (document order number CPU08RM/AD) contains a description of the CPU instruction set, addressing modes, and architecture.

7.2 Features

Features of the CPU include:

- Object code fully upward-compatible with M68HC05 Family
- 16-bit stack pointer with stack manipulation instructions
- 16-bit index register with x-register manipulation instructions
- 8-MHz CPU internal bus frequency
- 64-Kbyte program/data memory space
- 16 addressing modes
- Memory-to-memory data moves without using accumulator
- Fast 8-bit by 8-bit multiply and 16-bit by 8-bit divide instructions
- Enhanced binary-coded decimal (BCD) data handling
- Modular architecture with expandable internal bus definition for extension of addressing range beyond 64 Kbytes
- Low-power stop and wait modes

7.3 CPU Registers

Figure 7-1 shows the five CPU registers. CPU registers are not part of the memory map.

Table 7-1. Instruction Set Summary (Sheet 4 of 6)

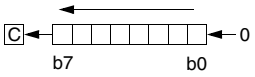
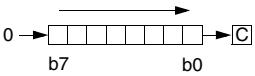
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
JMP *opr* JMP *opr* JMP *opr*,X JMP *opr*,X JMP ,X	Jump	PC ← Jump Address	–	–	–	–	–	–	DIR EXT IX2 IX1 IX	BC CC DC EC FC	dd hh ll ee ff ff	2 3 4 3 2
JSR *opr* JSR *opr* JSR *opr*,X JSR *opr*,X JSR ,X	Jump to Subroutine	PC ← (PC) + *n* (*n* = 1, 2, or 3) Push (PCL); SP ← (SP) – 1 Push (PCH); SP ← (SP) – 1 PC ← Unconditional Address	–	–	–	–	–	–	DIR EXT IX2 IX1 IX	BD CD DD ED FD	dd hh ll ee ff ff	4 5 6 5 4
LDA #*opr* LDA *opr* LDA *opr* LDA *opr*,X LDA *opr*,X LDA ,X LDA *opr*,SP LDA *opr*,SP	Load A from M	A ← (M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A6 B6 C6 D6 E6 F6 9EE6 9ED6	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
LDHX #*opr* LDHX *opr*	Load H:X from M	H:X ← (M:M + 1)	0	–	–	↕	↕	–	IMM DIR	45 55	ii jj dd	3 4
LDX #*opr* LDX *opr* LDX *opr* LDX *opr*,X LDX *opr*,X LDX ,X LDX *opr*,SP LDX *opr*,SP	Load X from M	X ← (M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	AE BE CE DE EE FE 9EEE 9EDE	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
LSL *opr* LSLA LSLX LSL *opr*,X LSL ,X LSL *opr*,SP	Logical Shift Left (Same as ASL)		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff	4 1 1 4 3 5
LSR *opr* LSRA LSR*X* LSR *opr*,X LSR ,X LSR *opr*,SP	Logical Shift Right		↕	–	–	0	↕	↕	DIR INH INH IX1 IX SP1	34 44 54 64 74 9E64	dd ff ff	4 1 1 4 3 5
MOV *opr*,*opr* MOV *opr*,X+ MOV #*opr*,*opr* MOV X+,*opr*	Move	$(M)_{Destination} \leftarrow (M)_{Source}$ H:X ← (H:X) + 1 (IX+D, DIX+)	0	–	–	↕	↕	–	DD DIX+ IMD IX+D	4E 5E 6E 7E	dd dd dd ii dd dd	5 4 4 4
MUL	Unsigned multiply	X:A ← (X) × (A)	–	0	–	–	–	0	INH	42		5
NEG *opr* NEGA NEGX NEG *opr*,X NEG ,X NEG *opr*,SP	Negate (Two's Complement)	M ← –(M) = $00 – (M) A ← –(A) = $00 – (A) X ← –(X) = $00 – (X) M ← –(M) = $00 – (M) M ← –(M) = $00 – (M)	↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	30 40 50 60 70 9E60	dd ff ff	4 1 1 4 3 5
NOP	No Operation	None	–	–	–	–	–	–	INH	9D		1
NSA	Nibble Swap A	A ← (A[3:0]:A[7:4])	–	–	–	–	–	–	INH	62		3
ORA #*opr* ORA *opr* ORA *opr* ORA *opr*,X ORA *opr*,X ORA ,X ORA *opr*,SP ORA *opr*,SP	Inclusive OR A and M	A ← (A) | (M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	AA BA CA DA EA FA 9EEA 9EDA	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
PSHA	Push A onto Stack	Push (A); SP ← (SP) – 1	–	–	–	–	–	–	INH	87		2
PSHH	Push H onto Stack	Push (H); SP ← (SP) – 1	–	–	–	–	–	–	INH	8B		2
PSHX	Push X onto Stack	Push (X); SP ← (SP) – 1	–	–	–	–	–	–	INH	89		2

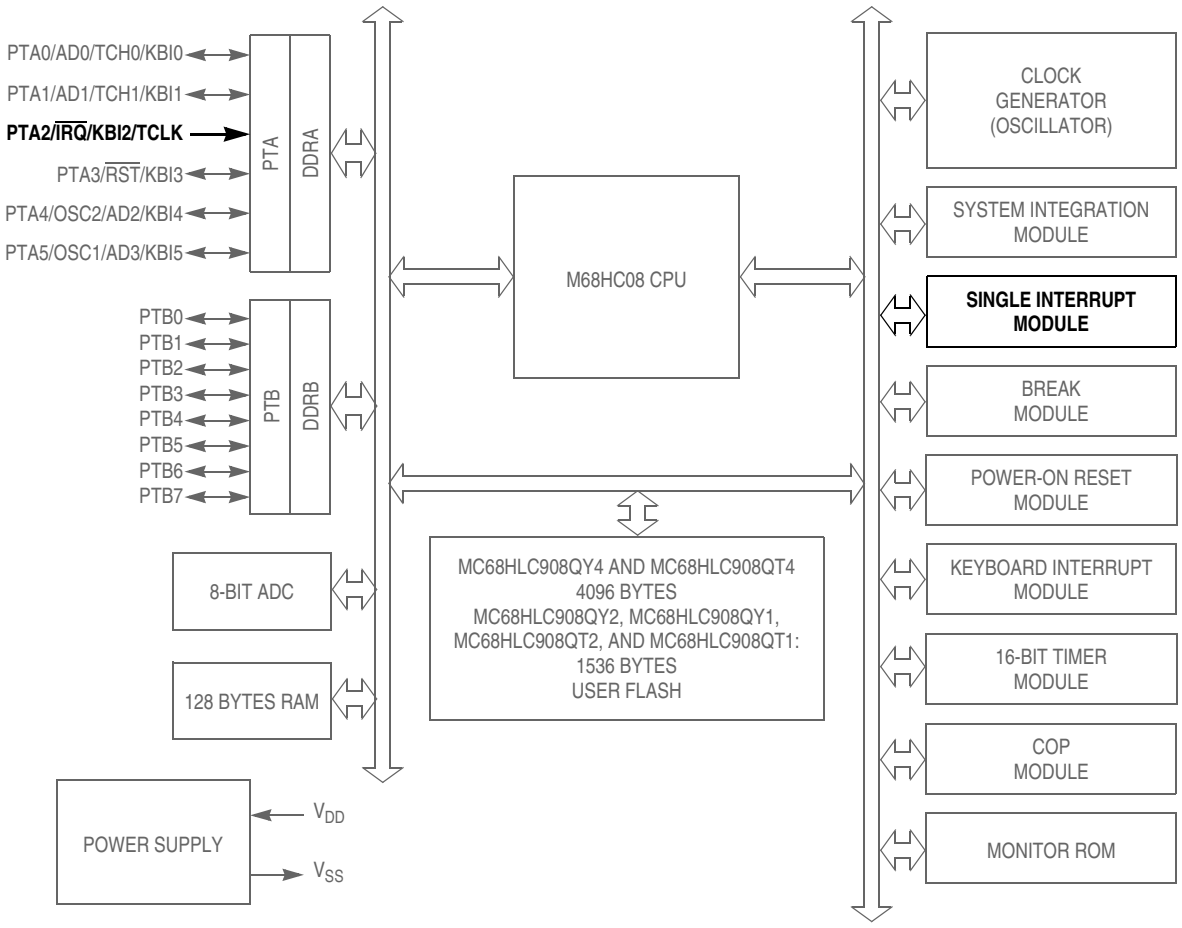
Table 7-1. Instruction Set Summary (Sheet 6 of 6)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
SWI	Software Interrupt	PC ← (PC) + 1; Push (PCL) SP ← (SP) – 1; Push (PCH) SP ← (SP) – 1; Push (X) SP ← (SP) – 1; Push (A) SP ← (SP) – 1; Push (CCR) SP ← (SP) – 1; I ← 1 PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte	–	–	1	–	–	–	INH	83		9
TAP	Transfer A to CCR	CCR ← (A)	↕	↕	↕	↕	↕	↕	INH	84		2
TAX	Transfer A to X	X ← (A)	–	–	–	–	–	–	INH	97		1
TPA	Transfer CCR to A	A ← (CCR)	–	–	–	–	–	–	INH	85		1
TST *opr* TSTA TSTX TST *opr*,X TST ,X TST *opr*,SP	Test for Negative or Zero	(A) – $00 or (X) – $00 or (M) – $00	0	–	–	↕	↕	–	DIR INH INH IX1 IX SP1	3D 4D 5D 6D 7D 9E6D	dd ff ff	3 1 1 3 2 4
TSX	Transfer SP to H:X	H:X ← (SP) + 1	–	–	–	–	–	–	INH	95		2
TXA	Transfer X to A	A ← (X)	–	–	–	–	–	–	INH	9F		1
TXS	Transfer H:X to SP	(SP) ← (H:X) – 1	–	–	–	–	–	–	INH	94		2
WAIT	Enable Interrupts; Wait for Interrupt	I bit ← 0; Inhibit CPU clocking until interrupted	–	–	0	–	–	–	INH	8F		1

A	Accumulator	*n*	Any bit
C	Carry/borrow bit	*opr*	Operand (one or two bytes)
CCR	Condition code register	PC	Program counter
dd	Direct address of operand	PCH	Program counter high byte
dd rr	Direct address of operand and relative offset of branch instruction	PCL	Program counter low byte
DD	Direct to direct addressing mode	REL	Relative addressing mode
DIR	Direct addressing mode	*rel*	Relative program counter offset byte
DIX+	Direct to indexed with post increment addressing mode	rr	Relative program counter offset byte
ee ff	High and low bytes of offset in indexed, 16-bit offset addressing	SP1	Stack pointer, 8-bit offset addressing mode
EXT	Extended addressing mode	SP2	Stack pointer 16-bit offset addressing mode
ff	Offset byte in indexed, 8-bit offset addressing	SP	Stack pointer
H	Half-carry bit	U	Undefined
H	Index register high byte	V	Overflow bit
hh ll	High and low bytes of operand address in extended addressing	X	Index register low byte
I	Interrupt mask	Z	Zero bit
ii	Immediate operand byte	&	Logical AND
IMD	Immediate source to direct destination addressing mode	|	Logical OR
IMM	Immediate addressing mode	⊕	Logical EXCLUSIVE OR
INH	Inherent addressing mode	( )	Contents of
IX	Indexed, no offset addressing mode	–( )	Negation (two's complement)
IX+	Indexed, no offset, post increment addressing mode	#	Immediate value
IX+D	Indexed with post increment to direct addressing mode	«	Sign extend
IX1	Indexed, 8-bit offset addressing mode	←	Loaded with
IX1+	Indexed, 8-bit offset, post increment addressing mode	?	If
IX2	Indexed, 16-bit offset addressing mode	:	Concatenated with
M	Memory location	↕	Set or cleared
N	Negative bit	—	Not affected

7.8 Opcode Map

See Table 7-2.



RST, IRQ: Pins have internal (about 30K Ohms) pull up
PTA[0:5]: High current sink and source capability
PTA[0:5]: Pins have programmable keyboard interrupt and pull up
PTB[0:7]: Not available on 8-pin devices – MC68HLC908QT1, MC68HLC908QT2, and MC68HLC908QT4 (see note in 12.1 Introduction)
ADC: Not available on the MC68HLC908QY1 and MC68HC9L08QT1

Figure 8-1. Block Diagram Highlighting IRQ Block and Pins

When set, the IMASK bit in INTSCR masks the IRQ interrupt request. A latched interrupt request is not presented to the interrupt priority logic unless IMASK is clear.

NOTE

The interrupt mask (I) in the condition code register (CCR) masks all interrupt requests, including the IRQ interrupt request.

A falling edge on the IRQ pin can latch an interrupt request into the IRQ latch. An IRQ vector fetch, software clear, or reset clears the IRQ latch.

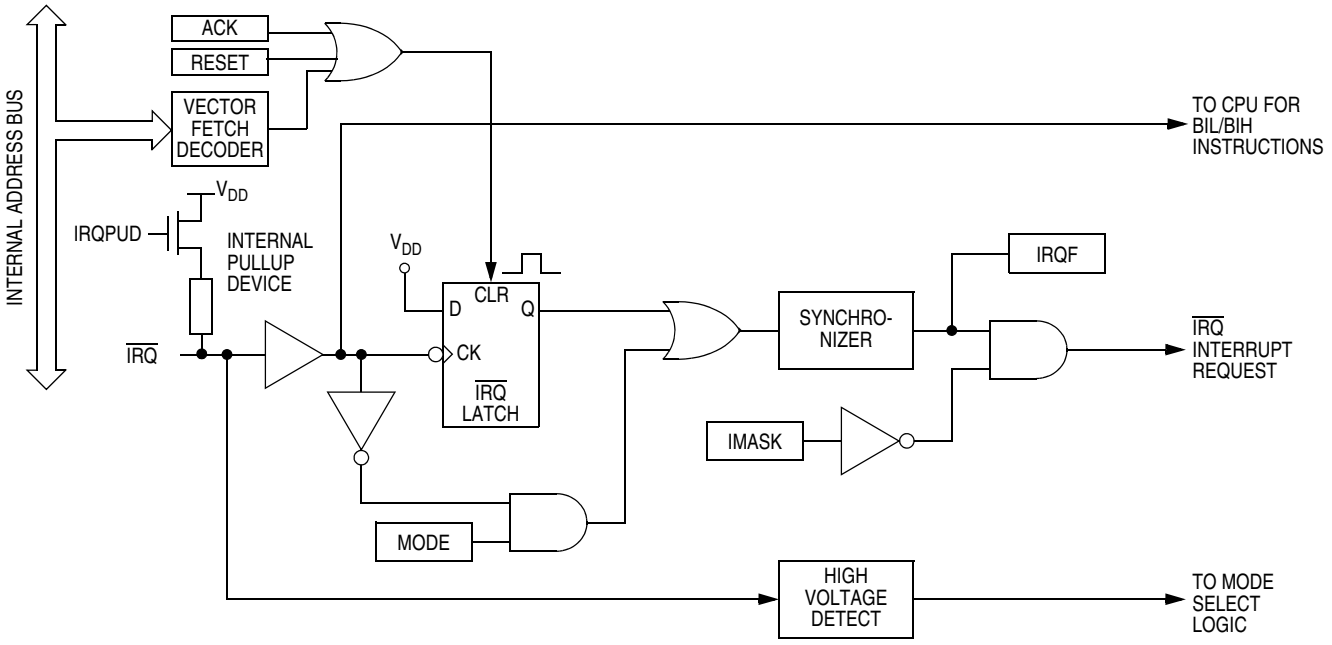


Figure 8-2. IRQ Module Block Diagram

8.3.1 MODE = 1

If the MODE bit is set, the $\overline{IRQ}$ pin is both falling edge sensitive and low level sensitive. With MODE set, both of the following actions must occur to clear the $\overline{IRQ}$ interrupt request:

- Return of the $\overline{IRQ}$ pin to a high level. As long as the $\overline{IRQ}$ pin is low, the IRQ request remains active.
- IRQ vector fetch or software clear. An IRQ vector fetch generates an interrupt acknowledge signal to clear the IRQ latch. Software generates the interrupt acknowledge signal by writing a 1 to ACK in INTSCR. The ACK bit is useful in applications that poll the $\overline{IRQ}$ pin and require software to clear the IRQ latch. Writing to ACK prior to leaving an interrupt service routine can also prevent spurious interrupts due to noise. Setting ACK does not affect subsequent transitions on the $\overline{IRQ}$ pin. A falling edge that occurs after writing to ACK latches another interrupt request. If the IRQ mask bit, IMASK, is clear, the CPU loads the program counter with the IRQ vector address.

The IRQ vector fetch or software clear and the return of the $\overline{IRQ}$ pin to a high level may occur in any order. The interrupt request remains pending as long as the $\overline{IRQ}$ pin is low. A reset will clear the IRQ latch and the MODE control bit, thereby clearing the interrupt even if the pin stays low.

Use the BIH or BIL instruction to read the logic level on the $\overline{IRQ}$ pin.

8.3.2 MODE = 0

If the MODE bit is clear, the $\overline{IRQ}$ pin is falling edge sensitive only. With MODE clear, an IRQ vector fetch or software clear immediately clears the IRQ latch.

The IRQF bit in INTSCR can be read to check for pending interrupts. The IRQF bit is not affected by IMASK, which makes it useful in applications where polling is preferred.

NOTE

When using the level-sensitive interrupt trigger, avoid false IRQ interrupts by masking interrupt requests in the interrupt routine.

Chapter 10
Low-Voltage Inhibit (LVI)

10.1 Introduction

This section describes the low-voltage inhibit (LVI) module, which monitors the voltage on the V_{DD} pin and can force a reset when the V_{DD} voltage falls below the LVI trip falling voltage, V_{TRIPF}.

10.2 Features

Features of the LVI module include:

- Programmable LVI reset
- Programmable power consumption
- Selectable LVI trip voltage
- Programmable stop mode operation

10.3 Functional Description

Figure 10-1 shows the structure of the LVI module. LVISTOP, LVIPWRD, LVDLVR, and LVIRSTD are user selectable options found in the configuration register (CONFIG1). See Chapter 5 Configuration Register (CONFIG).

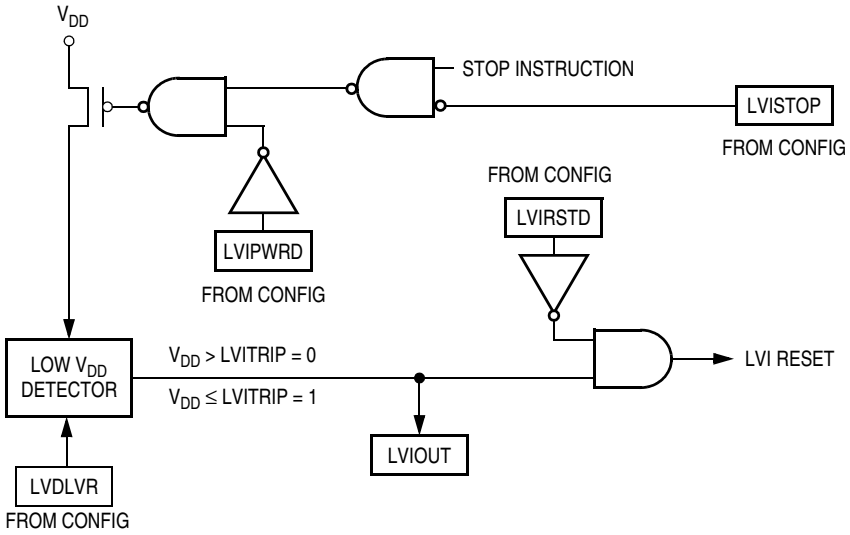


Figure 10-1. LVI Module Block Diagram

Chapter 11
Oscillator Module (OSC)

11.1 Introduction

The oscillator module is used to provide a stable clock source for the microcontroller system and bus. The oscillator module generates two output clocks, BUSCLKX2 and BUSCLKX4. The BUSCLKX4 clock is used by the system integration module (SIM) and the computer operating properly module (COP). The BUSCLKX2 clock is divided by two in the SIM to be used as the bus clock for the microcontroller. Therefore the bus frequency will be one forth of the BUSCLKX4 frequency.

11.2 Features

The oscillator has these four clock source options available:

1. Internal oscillator: An internally generated, fixed frequency clock, trimmable to ±5%. This 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 oscillator module (XTAL oscillator) that requires an external crystal or ceramic-resonator.

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

11.3.1 Internal Oscillator

The internal oscillator circuit is designed for use with no external components to provide a clock source with tolerance less than ±25% untrimmed. An 8-bit trimming register allows adjustment to a tolerance of less than ±5%.

The internal oscillator will generate a clock of 4.0 MHz typical (INTCLK) resulting in a bus speed (internal clock ÷ 4) of 1.0 MHz.

Figure 11-3 shows how BUSCLKX4 is derived from INTCLK and, like the RC oscillator, OSC2 can output BUSCLKX4 by setting OSC2EN in PTAPUE register. See Chapter 12 Input/Output Ports (PORTS).

Chapter 13
System Integration Module (SIM)

13.1 Introduction

This section describes the system integration module (SIM), which supports up to 24 external and/or internal interrupts. Together with the central processor unit (CPU), the SIM controls all microcontroller unit (MCU) activities. A block diagram of the SIM is shown in Figure 13-1. The SIM is a system state controller that coordinates CPU and exception timing.

The SIM is responsible for:

- Bus clock generation and control for CPU and peripherals
  - Stop/wait/reset/break entry and recovery
  - Internal clock control
- Master reset control, including power-on reset (POR) and computer operating properly (COP) timeout
- Interrupt control:
  - Acknowledge timing
  - Arbitration control timing
  - Vector address generation
- CPU enable/disable timing

13.2 $\overline{\text{RST}}$ and $\overline{\text{IRQ}}$ Pins Initialization

$\overline{\text{RST}}$ and $\overline{\text{IRQ}}$ pins come out of reset as PTA3 and PTA2 respectively. $\overline{\text{RST}}$ and $\overline{\text{IRQ}}$ functions can be activated by programing CONFIG2 accordingly. Refer to Chapter 5 Configuration Register (CONFIG).

Table 13-1. Signal Name Conventions

Signal Name	Description
BUSCLKX4	Buffered clock from the internal, RC or XTAL oscillator circuit.
BUSCLKX2	The BUSCLKX4 frequency divided by two. This signal is again divided by two in the SIM to generate the internal bus clocks (bus clock = BUSCLKX4 ÷ 4).
Address bus	Internal address bus
Data bus	Internal data bus
PORRST	Signal from the power-on reset module to the SIM
IRST	Internal reset signal
R/$\overline{\text{W}}$	Read/write signal

IF1 and IF3–IF5 — Interrupt Flags

These flags indicate the presence of interrupt requests from the sources shown in Table 13-3.

1 = Interrupt request present
0 = No interrupt request present

Bit 0, 1, 3, and 7 — Always read 0

13.6.2.2 Interrupt Status Register 2

Address: $FE05

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	IF14	0	0	0	0	0	0	0
Write:	R	R	R	R	R	R	R	R
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Figure 13-12. Interrupt Status Register 2 (INT2)

IF14 — Interrupt Flags

This flag indicates the presence of interrupt requests from the sources shown in Table 13-3.

1 = Interrupt request present
0 = No interrupt request present

Bit 0–6 — Always read 0

13.6.2.3 Interrupt Status Register 3

Address: $FE06

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	0	IF15
Write:	R	R	R	R	R	R	R	R
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Figure 13-13. Interrupt Status Register 3 (INT3)

IF15 — Interrupt Flags

These flags indicate the presence of interrupt requests from the sources shown in Table 13-3.

1 = Interrupt request present
0 = No interrupt request present

Bit 1–7 — Always read 0

13.6.3 Reset

All reset sources always have equal and highest priority and cannot be arbitrated.

13.6.4 Break Interrupts

The break module can stop normal program flow at a software programmable break point by asserting its break interrupt output. (See Chapter 15 Development Support.) The SIM puts the CPU into the break

14.6 Wait Mode

The WAIT instruction puts the MCU in low power-consumption standby mode.

The TIM remains active after the execution of a WAIT instruction. In wait mode the TIM registers are not accessible by the CPU. Any enabled CPU interrupt request from the TIM can bring the MCU out of wait mode.

If TIM functions are not required during wait mode, reduce power consumption by stopping the TIM before executing the WAIT instruction.

14.7 TIM During Break Interrupts

A break interrupt stops the TIM counter.

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 13.8.2 Break Flag Control Register.

To allow software to clear status bits during a break interrupt, write a 1 to the BCFE bit. 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 the BCFE bit. With BCFE at 0 (its default state), software can read and write I/O 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 at 0. After the break, doing the second step clears the status bit.

14.8 Input/Output Signals

Port A shares three of its pins with the TIM. Two TIM channel I/O pins are PTA0/TCH0 and PTA1/TCH1 and an alternate clock source is PTA2/TCLK.

14.8.1 TIM Clock Pin (PTA2/TCLK)

PTA2/TCLK is an external clock input that can be the clock source for the TIM counter instead of the prescaled internal bus clock. Select the PTA2/TCLK input by writing 1s to the three prescaler select bits, PS[2–0]. (See 14.9.1 TIM Status and Control Register.) When the PTA2/TCLK pin is the TIM clock input, it is an input regardless of port pin initialization.

14.8.2 TIM Channel I/O Pins (PTA0/TCH0 and PTA1/TCH1)

Each channel I/O pin is programmable independently as an input capture pin or an output compare pin. PTA0/TCH0 can be configured as a buffered output compare or buffered PWM pin.

14.9 Input/Output Registers

The following I/O registers control and monitor operation of the TIM:

- TIM status and control register (TSC)
- TIM counter registers (TCNTH:TCNTL)
- TIM counter modulo registers (TMODH:TMODL)
- TIM channel status and control registers (TSC0 and TSC1)
- TIM channel registers (TCH0H:TCH0L and TCH1H:TCH1L)

16.3 Functional Operating Range

Characteristic	Symbol	Value	Unit	Temp Code
Operating temperature range (T_L to T_H)	T_A	–40 to 85 0 to 70	°C	C —
Operating voltage range[1] (V_{DDMIN} to V_{DDMAX}) –40 to 85°C 0 to 70°C	V_{DD}	2.4 to 3.6 2.2 to 3.6	V	C —

1. V_{DD} must be above V_{TRIPR} upon power on.

16.4 Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal resistance 8-pin PDIP 8-pin SOIC 8-pin DFN 16-pin PDIP 16-pin SOIC 16-pin TSSOP	θ_{JA}	 105 142 173 76 90 133	°C/W
I/O pin power dissipation	$P_{I/O}$	User determined	W
Power dissipation[1]	P_D	$P_D = (I_{DD} \times V_{DD}) + P_{I/O} = K/(T_J + 273°C)$	W
Constant[2]	K	$P_D \times (T_A + 273°C) + P_D^2 \times \theta_{JA}$	W/°C
Average junction temperature	T_J	$T_A + (P_D \times \theta_{JA})$	°C
Maximum junction temperature	T_{JM}	150	°C

1. Power dissipation is a function of temperature.
2. K constant unique to the device. K can be determined for a known T_A and measured P_D. With this value of K, P_D and T_J can be determined for any value of T_A.

16.10 Analog-to-Digital (ADC) Converter Characteristics

16.10.1 ADC Electrical Operating Conditions

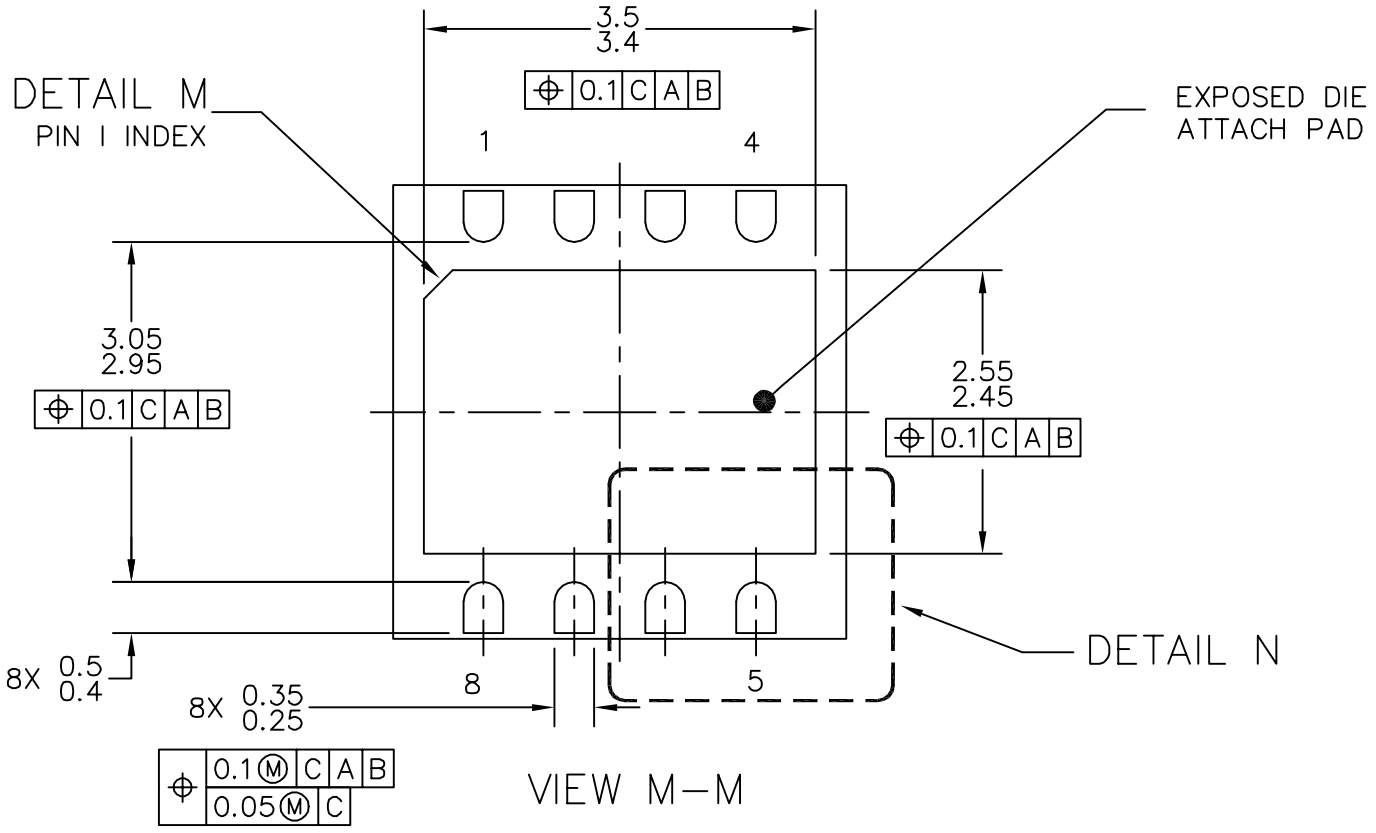
The ADC accuracy characteristics below are guaranteed over two operating conditions as stated here.

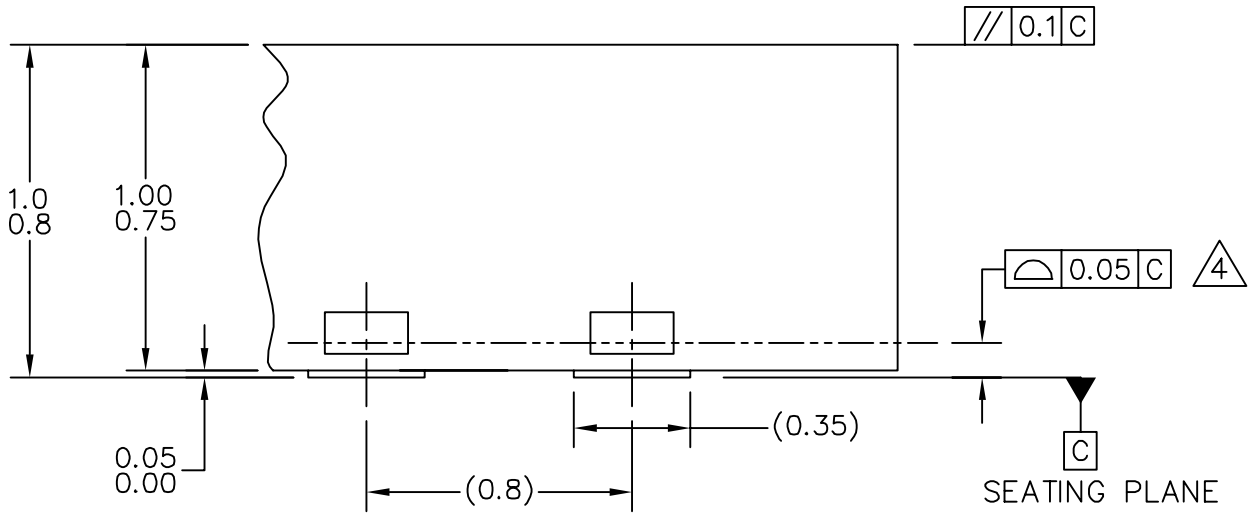
	Characteristic	Symbol	Min	Max	Unit
Condition A	ATD supply	V_{DD}	2.7	3.6	V
	ADC internal clock	f_{ADIC}	0.008	1	MHz
	Ambient temperature	T_A	T_L	T_H	°C
Condition B	ATD supply	V_{DD}	2.3	2.7	V
	ADC internal clock	f_{ADIC}	8	63	kHz
	Ambient temperature	T_A	0	T_H	°C

16.10.2 ADC Performance Characteristics

Characteristic		Symbol	Min	Max	Unit	Comments
Input voltages		V_{ADIN}	V_{SS}	V_{DD}	V	—
Resolution (1 LSB)	Condition A Condition B	RES	10.5 8.99	14.1 10.5	mV	—
Absolute accuracy (Total unadjusted error)	Condition A Condition B	E_{TUE}	— —	± 1.5 ± 2.0	LSB	Includes quantization
Conversion range		V_{AIN}	V_{SS}	V_{DD}	V	—
Power-up time		t_{ADPU}	16	—	t_{ADIC} cycles	$t_{ADIC} = 1/f_{ADIC}$
Conversion time		t_{ADC}	16	17	t_{ADIC} cycles	$t_{ADIC} = 1/f_{ADIC}$
Sample time[1]		t_{ADS}	5	—	t_{ADIC} cycles	$t_{ADIC} = 1/f_{ADIC}$
Zero input reading[2]		Z_{ADI}	00	01	Hex	$V_{IN} = V_{SS}$
Full-scale reading[3]		F_{ADI}	FE	FF	Hex	$V_{IN} = V_{DD}$
Input capacitance		C_{ADI}	—	8	pF	Not tested
Input leakage[3]		I_{IL}	—	± 1	µA	—
ADC supply current (V_{DD} = 3 V)		I_{ADAD}	Typical = 0.45		mA	Enabled

1. Source impedances greater than 10 kΩ adversely affect internal RC charging time during input sampling.
2. Zero-input/full-scale reading requires sufficient decoupling measures for accurate conversions.
3. The external system error caused by input leakage current is approximately equal to the product of R source and input current.





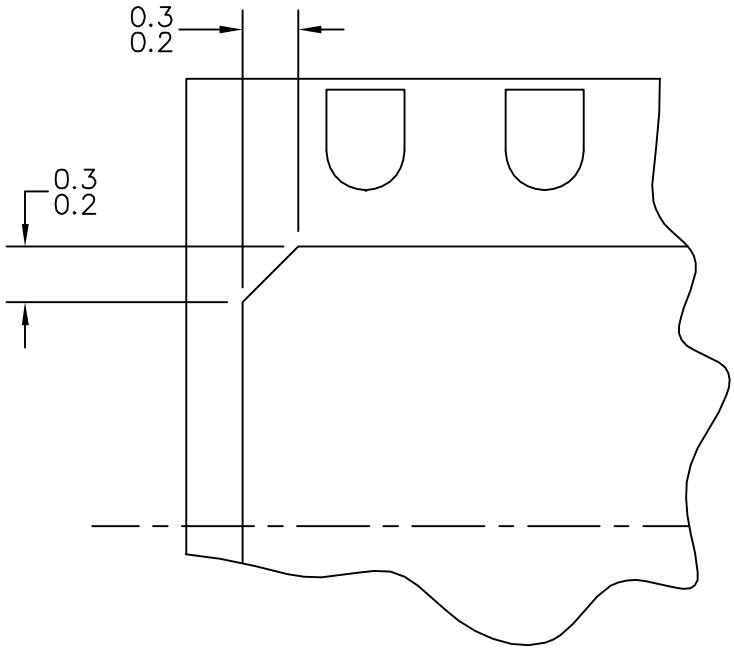
DETAIL G

TITLE: THERMALLY ENHANCED DUAL FLAT NO LEAD PACKAGE (DFN) 8 TERMINAL, 0.8 PITCH (4 X 4 X 1)

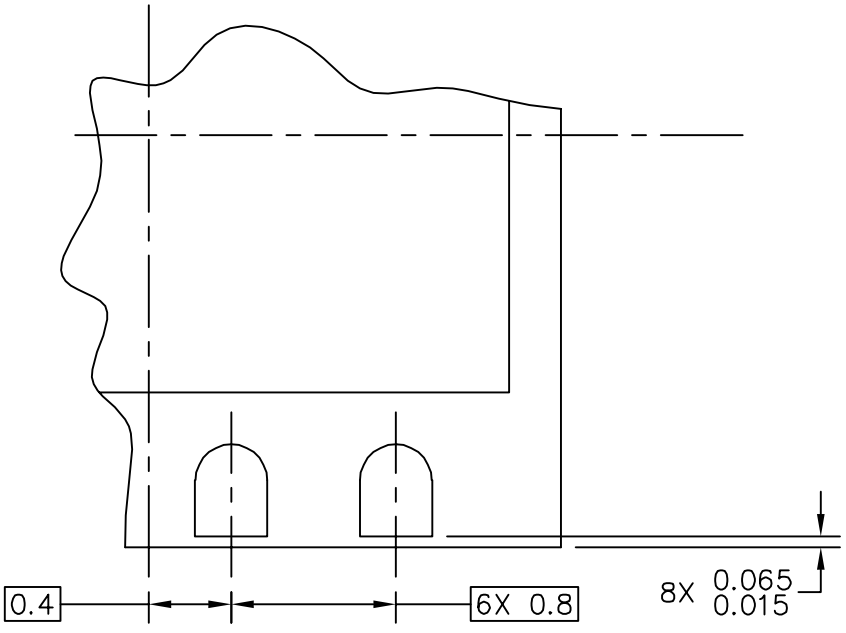
CASE NUMBER: 1452–01

STANDARD: NON–JEDEC

PACKAGE CODE: 6165



DETAIL M
BACKSIDE PIN 1 INDEX



DETAIL N

TITLE: THERMALLY ENHANCED DUAL FLAT NO LEAD PACKAGE (DFN) 8 TERMINAL, 0.8 PITCH (4 X 4 X 1)

CASE NUMBER: 1452-01
STANDARD: NON-JEDEC
PACKAGE CODE: 6165
SHEET: 3 OF 5