

COP87L88CF/COP87L84CF 8-Bit One-Time Programmable (OTP) Microcontroller with A/D Converter

General Description

The COP87L88CF/COP87L84CF OTP microcontrollers are members of the COP8™ feature family using an 8-bit core architecture. It is pin and software compatible to the mask ROM COP888CF/COP884CF product family.

(Continued)

Key Features

- Two 16-bit timers, each with two 16-bit registers supporting:
 - Processor Independent PWM mode
 - External Event counter mode
 - Input Capture mode
- 8-channel A/D converter with prescaler and both differential and single ended modes
- 4 kbytes on-board EPROM with security feature
- 128 bytes on-board RAM

Additional Peripheral Features

- Idle Timer
- Multi-Input Wake Up (MIWU) with optional interrupts (8)
- WATCHDOG™ and Clock Monitor logic
- MICROWIRE/PLUS™ serial I/O

I/O Features

- Memory mapped I/O
- Software selectable I/O options
 - TRI-STATE® Output (push-pull output, weak pull up input, high impedance input)
- Schmitt trigger inputs on ports G and L

Packages:

- 44 PLCC with 38 I/O pins
- 40 DIP with 34 I/O pins
- 28 DIP with 22 I/O pins
- 28 SO with 22 I/O pins (contact local sales office for availability)

CPU/Instruction Set Features

- 1 μ s instruction cycle time
- Ten multi-source vectored interrupts servicing
 - External interrupt
 - Idle timer T0
 - Two timers (each with 2 interrupts)
 - MICROWIRE/PLUS
 - Multi-Input Wake Up
 - Software trap
 - Default VIS (default interrupt)
- Versatile and easy to use instruction set
- 8-bit Stack Pointer SP—stack in RAM
- Two 8-bit Register Indirect Data Memory Pointers (B and X)

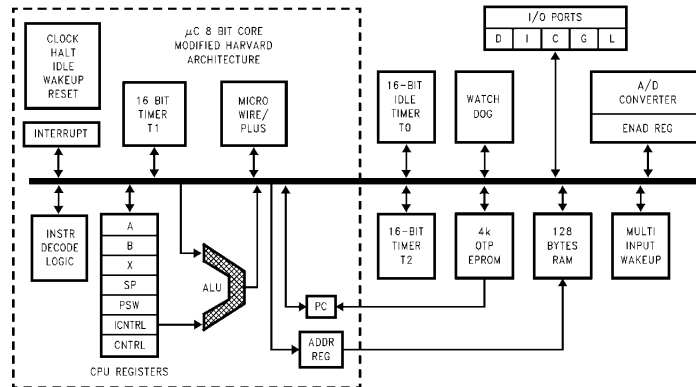
Fully Static CMOS

- Two power saving modes: HALT and IDLE
- Single supply operation: 2.7V–5.5V
- Temperature range: –40°C to +85°C

Development Support

- Emulation device for the COP888CF/COP884CF
- Real time emulation and full program debug offered by MetaLink Development System

Block Diagram



TL/DD/12523-17

TRI-STATE® is a registered trademark of National Semiconductor Corporation.
COP8™, MICROWIRE™, MICROWIRE/PLUS™ and WATCHDOG™ are trademarks of National Semiconductor Corporation.
PC® is a registered trademark of International Business Machines Corporation.
iceMASTER™ is a trademark of MetaLink Corporation.

Connection Diagrams (Continued)

Pinouts for 28-Pin, 40-Pin and 44-Pin Packages

| Port | Type | Alt. Fun | Alt. Fun | 28-Pin Pkg. | 40-Pin Pkg. | 44-Pin Pkg. |
|------------------|--------------------|--------------|----------|-------------|-------------|-------------|
| L0 | I/O | MIWU | | 11 | 17 | |
| L1 | I/O | MIWU | | 12 | 18 | |
| L2 | I/O | MIWU | | 13 | 19 | 19 |
| L3 | I/O | MIWU | | 14 | 20 | 20 |
| L4 | I/O | MIWU | T2A | 15 | 21 | 25 |
| L5 | I/O | MIWU | T2B | 16 | 22 | 26 |
| L6 | I/O | MIWU | | 17 | 23 | 27 |
| L7 | I/O | MIWU | | 18 | 24 | 28 |
| G0 | I/O | INT | | 25 | 35 | 39 |
| G1 | WDOOUT | | | 26 | 36 | 40 |
| G2 | I/O | T1B | | 27 | 37 | 41 |
| G3 | I/O | T1A | | 28 | 38 | 42 |
| G4 | I/O | SO | | 1 | 3 | 3 |
| G5 | I/O | SK | | 2 | 4 | 4 |
| G6 | I | SI | | 3 | 5 | 5 |
| G7 | I/CKO | HALT Restart | | 4 | 6 | 6 |
| D0 | O | | | 19 | 25 | 29 |
| D1 | O | | | 20 | 26 | 30 |
| D2 | O | | | 21 | 27 | 31 |
| D3 | O | | | 22 | 28 | 32 |
| I0 | I | ACH0 | | 7 | 9 | 9 |
| I1 | I | ACH1 | | 8 | 10 | 10 |
| I2 | I | ACH2 | | | 11 | 11 |
| I3 | I | ACH3 | | | 12 | 12 |
| I4 | I | ACH4 | | | 13 | 13 |
| I5 | I | ACH5 | | | 14 | 14 |
| I6 | I | ACH6 | | | | 15 |
| I7 | I | ACH7 | | | | 16 |
| D4 | O | | | | 29 | 33 |
| D5 | O | | | | 30 | 34 |
| D6 | O | | | | 31 | 35 |
| D7 | O | | | | 32 | 36 |
| C0 | I/O | | | | 39 | 43 |
| C1 | I/O | | | | 40 | 44 |
| C2 | I/O | | | | 1 | 1 |
| C3 | I/O | | | | 2 | 2 |
| C4 | I/O | | | | | 21 |
| C5 | I/O | | | | | 22 |
| C6 | I/O | | | | | 23 |
| C7 | I/O | | | | | 24 |
| V _{REF} | + V _{REF} | | | 10 | 16 | 18 |
| AGND | AGND | | | 9 | 15 | 17 |
| V _{CC} | | | | 6 | 8 | 8 |
| GND | | | | 23 | 33 | 37 |
| CKI | | | | 5 | 7 | 7 |
| RESET | | | | 24 | 34 | 38 |

Absolute Maximum Ratings (Note)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

| | |
|------------------------------------------|--------------------------|
| Supply Voltage (V_{CC}) | 7V |
| Voltage at Any Pin | -0.3V to $V_{CC} + 0.3V$ |
| Total Current into V_{CC} Pin (Source) | 100 mA |

| | |
|-------------------------------------|-----------------|
| Total Current out of GND Pin (Sink) | 110 mA |
| Storage Temperature Range | -65°C to +140°C |

Note: Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.

DC Electrical Characteristics -40°C ≤ T_A ≤ +85°C unless otherwise specified

| Parameter | Conditions | Min | Typ | Max | Units |
|------------------------------------------------|----------------------------------|--------------|---------------|---------------|---------|
| Operating Voltage | | 2.7 | | 5.5 | V |
| Power Supply Ripple (Note 1) | Peak-to-Peak | | | 0.1 V_{CC} | V |
| Supply Current (Note 2) | | | | | |
| CKI = 10 MHz | $V_{CC} = 5.5V, t_c = 1 \mu s$ | | | 16.5 | mA |
| CKI = 4 MHz | $V_{CC} = 4.0V, t_c = 2.5 \mu s$ | | | 6.5 | mA |
| HALT Current (Note 3) | $V_{CC} = 5.5V, CKI = 0 MHz$ | | | 12 | μA |
| IDLE Current | | | | | |
| CKI = 10 MHz | $V_{CC} = 5.5V, t_c = 1 \mu s$ | | | 3.5 | mA |
| CKI = 1 MHz | $V_{CC} = 4.0V, t_c = 10 \mu s$ | | | 0.7 | mA |
| Input Levels | | | | | |
| RESET | | | | | |
| Logic High | | 0.8 V_{CC} | | | V |
| Logic Low | | | | 0.2 V_{CC} | V |
| CKI (External and Crystal Osc. Modes) | | | | | |
| Logic High | | 0.7 V_{CC} | | | V |
| Logic Low | | | | 0.2 V_{CC} | V |
| All Other Inputs | | | | | |
| Logic High | | 0.7 V_{CC} | | | V |
| Logic Low | | | | 0.2 V_{CC} | V |
| Hi-Z Input Leakage | $V_{CC} = 5.5V$ | -2 | | +2 | μA |
| Input Pullup Current | $V_{CC} = 5.5V$ | 40 | | 250 | μA |
| G and L Port Input Hysteresis | | | 0.05 V_{CC} | 0.35 V_{CC} | V |
| Output Current Levels | | | | | |
| D Outputs | | | | | |
| Source | $V_{CC} = 4.5V, V_{OH} = 3.3V$ | 0.4 | | | mA |
| Sink (Note 4) | $V_{CC} = 4.5V, V_{OL} = 1V$ | 10 | | | mA |
| All Others | | | | | |
| Source (Weak Pull-Up Mode) | $V_{CC} = 4.5V, V_{OH} = 2.7V$ | 10 | | 100 | μA |
| Source (Push-Pull Mode) | $V_{CC} = 4.5V, V_{OH} = 3.3V$ | 0.4 | | | mA |
| Sink (Push-Pull Mode) | $V_{CC} = 4.5V, V_{OL} = 0.4V$ | 1.6 | | | mA |
| TRI-STATE Leakage | $V_{CC} = 5.5V$ | -2 | | +2 | μA |
| Allowable Sink/Source | | | | | |
| Current per Pin | | | | | |
| D Outputs (Sink) | | | | 15 | mA |
| All Others | | | | 3 | mA |
| Maximum Input Current without Latchup (Note 7) | $T_A = 25^\circ C$ | | | ±100 | mA |
| RAM Retention Voltage, V_r | 500 ns Rise and Fall Time (Min) | 2 | | | V |
| Input Capacitance | | | | 7 | pF |
| Load Capacitance on D2 | | | | 1000 | pF |

Note 1: Rate of voltage change must be less than 0.5 V/ms.

Note 2: Supply current is measured after running 2000 cycles with a square wave CKI input, CKO open, inputs at rails and outputs open.

Note 3: The HALT mode will stop CKI from oscillating in the RC and the Crystal configurations by bringing CKI high. Test conditions: All inputs tied to V_{CC} , L and G ports in the TRI-STATE mode and tied to ground, all outputs low and tied to ground. The A/D is disabled. V_{REF} is tied to AGND (effectively shorting the Reference resistor). The clock monitor is disabled.

Note 4: The user must guarantee that D2 pin does not source more than 10 mA during RESET. If D2 sources more than 10 mA during reset, the device will go into programming mode.

A/D Converter Specifications $V_{CC} = 5V \pm 10\% (V_{SS} - 0.050V) \leq \text{Any Input} \leq (V_{CC} + 0.050V)$

| Parameter | Conditions | Min | Typ | Max | Units |
|----------------------------------|----------------------------------------------------------------|------|-----|-----------|---------------------|
| Resolution | | | | 8 | Bits |
| Reference Voltage Input | AGND = 0V | 3 | | V_{CC} | V |
| Absolute Accuracy | $V_{REF} = V_{CC}$ | | | ± 2 | LSB |
| Non-Linearity | $V_{REF} = V_{CC}$ Deviation from the Best Straight Line | | | $\pm 1/2$ | LSB |
| Differential Non-Linearity | $V_{REF} = V_{CC}$ | | | $\pm 1/2$ | LSB |
| Input Reference Resistance | | 1.6 | | 4.8 | k Ω |
| Common Mode Input Range (Note 8) | | AGND | | V_{REF} | V |
| DC Common Mode Error | | | | $\pm 1/4$ | LSB |
| Off Channel Leakage Current | | | 1 | | μA |
| On Channel Leakage Current | | | 1 | | μA |
| A/D Clock Frequency (Note 6) | | 0.1 | | 1.67 | MHz |
| Conversion Time (Note 5) | | | 12 | | A/D Clock Cycles |

Note 5: Conversion Time includes sample and hold time.

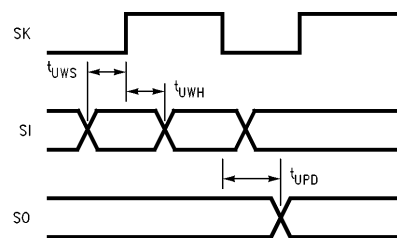
Note 6: See Prescaler description.

Note 7: Pins G6 and RESET are designed with a high voltage input network for factory testing. These pins allow input voltages greater than V_{CC} and the pins will have sink current to V_{CC} when biased at voltages greater than V_{CC} (the pins do not have source current when biased at a voltage below V_{CC}). The effective resistance to V_{CC} is 750 Ω (typical). These two pins will not latch up. The voltage at the pins must be limited to less than 14V.

Note 8: For $V_{IN(-)} \geq V_{IN(+)}$, the digital output code will be 0000 0000. Two on-chip diodes are tied to each analog input. The diodes will forward conduct for analog input voltages below ground or above the V_{CC} supply. Be careful, during testing at low V_{CC} levels (4.5V), as high level analog inputs (5V) can cause this input diode to conduct—especially at elevated temperatures, and cause errors for analog inputs near full-scale. The spec allows 50 mV forward bias of either diode. This means that as long as the analog V_{IN} does not exceed the supply voltage by more than 50 mV, the output code will be correct. To achieve an absolute 0 V_{DC} to 5 V_{DC} input voltage range will therefore require a minimum supply voltage of 4.950 V_{DC} over temperature variations, initial tolerance and loading.

AC Electrical Characteristics $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ unless otherwise specified

| Parameter | Conditions | Min | Typ | Max | Units |
|---------------------------------------------------------|-----------------------------------------------|-----|-----|-----|---------------|
| Instruction Cycle Time (t_c) | | | | | |
| Crystal, Resonator | | 1 | | DC | μs |
| R/C Oscillator | | 3 | | DC | μs |
| Inputs | | | | | |
| t_{SETUP} | | 200 | | | ns |
| t_{HOLD} | | 60 | | | ns |
| Output Propagation Delay | $R_L = 2.2\text{k}, C_L = 100\text{ pF}$ | | | | |
| $t_{\text{PD1}}, t_{\text{PD0}}$ | $4\text{V} \leq V_{\text{CC}} \leq 6\text{V}$ | | | 0.7 | μs |
| SO, SK | $4\text{V} \leq V_{\text{CC}} \leq 6\text{V}$ | | | 1 | μs |
| All Others | | | | | |
| MICROWIRE™ Setup Time (t_{UWS}) | | 20 | | | ns |
| MICROWIRE Hold Time (t_{UWH}) | | 56 | | | ns |
| MICROWIRE Output Propagation Delay (t_{UPD}) | | | | 220 | ns |
| Input Pulse Width | | | | | |
| Interrupt Input High Time | | 1 | | | t_c |
| Interrupt Input Low Time | | 1 | | | t_c |
| Timer Input High Time | | 1 | | | t_c |
| Timer Input Low Time | | 1 | | | t_c |
| Reset Pulse Width | | 1 | | | μs |



TL/DD/12523-4

FIGURE 2. MICROWIRE/PLUS Timing

Pin Descriptions

V_{CC} and GND are the power supply pins.

V_{REF} and AGND are the reference voltage pins for the on-board A/D converter.

CKI is the clock input. This can come from an R/C generated oscillator, or a crystal oscillator (in conjunction with CKO). See Oscillator Description section.

RESET is the master reset input. See Reset Description section.

The device contains three bidirectional 8-bit I/O ports (C, G and L), where each individual bit may be independently configured as an input (Schmitt trigger inputs on ports G and L), output or TRI-STATE under program control. Three data memory address locations are allocated for each of these I/O ports. Each I/O port has two associated 8-bit memory mapped registers, the CONFIGURATION register and the output DATA register. A memory mapped address is also reserved for the input pins of each I/O port. (See the memory map for the various addresses associated with the I/O ports.) Figure 3 shows the I/O port configurations. The DATA and CONFIGURATION registers allow for each port bit to be individually configured under software control as shown below:

| Configuration Register | Data Register | Port Set-Up |
|------------------------|---------------|-------------------------------|
| 0 | 0 | Hi-Z Input (TRI-STATE Output) |
| 0 | 1 | Input with Weak Pull-Up |
| 1 | 0 | Push-Pull Zero Output |
| 1 | 1 | Push-Pull One Output |

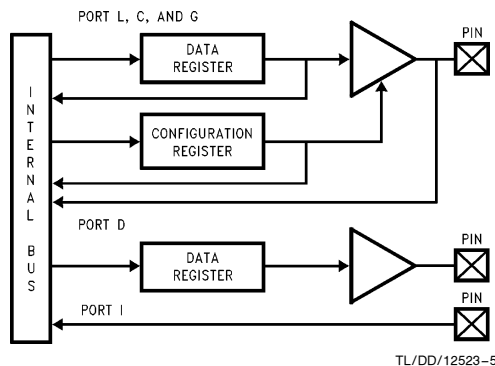


FIGURE 3. I/O Port Configurations

PORT L is an 8-bit I/O port. All L-pins have Schmitt triggers on the inputs.

Port L supports Multi-Input Wakeup (MIWU) on all eight pins. L4 and L5 are used for the timer input functions T2A and T2B. L0 and L1 are not available on the 44-pin version,

since they are replaced by V_{REF} and AGND. L0 and L1 are not terminated on the 44-pin version. Consequently, reading L0 or L1 as inputs will return unreliable data with the 44-pin package, so this data should be masked out with user software when the L port is read for input data. It is recommended that the pins be configured as outputs.

Port L has the following alternate features:

- L0 MIWU
- L1 MIWU
- L2 MIWU
- L3 MIWU
- L4 MIWU or T2A
- L5 MIWU or T2B
- L6 MIWU
- L7 MIWU

Port G is an 8-bit port with 5 I/O pins (G0, G2–G5), an input pin (G6), and two dedicated output pins (G1 and G7). Pins G0 and G2–G6 all have Schmitt Triggers on their inputs. Pin G1 serves as the dedicated WDOOUT WATCHDOG output, while pin G7 is either input or output depending on the oscillator mask option selected. With the crystal oscillator option selected, G7 serves as the dedicated output pin for the CKO clock output. With the single-pin R/C oscillator mask option selected, G7 serves as a general purpose input pin, but is also used to bring the device out of HALT mode with a low to high transition on G7. There are two registers associated with the G Port, a data register and a configuration register. Therefore, each of the 5 I/O bits (G0, G2–G5) can be individually configured under software control.

Since G6 is an input only pin and G7 is the dedicated CKO clock output pin or general purpose input (R/C clock configuration), the associated bits in the data and configuration registers for G6 and G7 are used for special purpose functions as outlined below. Reading the G6 and G7 data bits will return zeros.

Note that the chip will be placed in the HALT mode by writing a "1" to bit 7 of the Port G Data Register. Similarly the chip will be placed in the IDLE mode by writing a "1" to bit 6 of the Port G Data Register.

Writing a "1" to bit 6 of the Port G Configuration Register enables the MICROWIRE/PLUS to operate with the alternate phase of the SK clock. The G7 configuration bit, if set high, enables the clock start up delay after HALT when the R/C clock configuration is used.

| | Config Reg. | Data Reg. |
|----|--------------|-----------|
| G7 | CLKDLY | HALT |
| G6 | Alternate SK | IDLE |

Port G has the following alternate features:

- G0 INTR (External Interrupt Input)
- G2 T1B (Timer T1 Capture Input)
- G3 T1A (Timer T1 I/O)
- G4 SO (MICROWIRE Serial Data Output)
- G5 SK (MICROWIRE Serial Clock)
- G6 SI (MICROWIRE Serial Data Input)

Pin Descriptions (Continued)

Port G has the following dedicated functions:

- G1 WDOOUT WATCHDOG and/or Clock Monitor dedicated output
- G7 CKO Oscillator dedicated output or general purpose input

Port C is an 8-bit I/O port. The 40-pin device does not have a full complement of Port C pins. The unavailable pins are not terminated. A read operation for these unterminated pins will return unpredictable values.

Port I is an 8-bit Hi-Z input port, and also provides the analog inputs to the A/D converter. The 28-pin device does not have a full complement of Port I pins. The unavailable pins are not terminated (i.e. they are floating). A read operation from these unterminated pins will return unpredictable values. The user should ensure that the software takes this into account by either masking out these inputs, or else restricting the accesses to bit operations only. If unterminated, Port I pins will draw power only when addressed. The I port leakage current may be higher in 28-pin devices.

Port D is a recreated 8-bit output port that is preset high when RESET goes low. D port recreation is one clock cycle behind the normal port timing. The user can tie two or more D port outputs (except D2 pin) together in order to get a higher drive.

Functional Description

The architecture of the device is modified Harvard architecture. With the Harvard architecture, the control store program memory (ROM) is separated from the data store memory (RAM). Both ROM and RAM have their own separate addressing space with separate address buses. The architecture, though based on Harvard architecture, permits transfer of data from ROM to RAM.

CPU REGISTERS

The CPU can do an 8-bit addition, subtraction, logical or shift operation in one instruction (t_c) cycle time.

There are five CPU registers:

A is the 8-bit Accumulator Register

PC is the 15-bit Program Counter Register

PU is the upper 7 bits of the program counter (PC)

PL is the lower 8 bits of the program counter (PC)

B is an 8-bit RAM address pointer, which can be optionally post auto incremented or decremented.

X is an 8-bit alternate RAM address pointer, which can be optionally post auto incremented or decremented.

SP is the 8-bit stack pointer, which points to the subroutine/interrupt stack (in RAM). The SP is initialized to RAM address 06F with reset.

All the CPU registers are memory mapped with the exception of the Accumulator (A) and the Program Counter (PC).

PROGRAM MEMORY

Program memory consists of 4 kbytes of OTP EPROM. These bytes may hold program instructions or constant data (data tables for the LAID instruction, jump vectors for the JID instruction, and interrupt vectors for the VIS instruction). The program memory is addressed by the 15-bit program counter (PC). All interrupts vector to program memory location 0FF Hex.

The device can be configured to inhibit external reads of the program memory. This is done by programming the Security Byte.

SECURITY FEATURE

The program memory array has an associate Security Byte that is located outside of the program address range. This byte can be addressed only from programming mode by a programmer tool.

Security is an optional feature and can only be asserted after the memory array has been programmed and verified. A secured part will read all 00(hex) by a programmer. The part will fail Blank Check and will fail Verify operations. A Read operation will fill the programmer's memory with 00(hex). The Security Byte itself is always readable with value of 00(hex) if unsecured and FF(hex) if secured.

DATA MEMORY

The data memory address space includes the on-chip RAM and data registers, the I/O registers (Configuration, Data and Pin), the control registers, the MICROWIRE/PLUS SIO shift register, and the various registers, and counters associated with the timers (with the exception of the IDLE timer). Data memory is addressed directly by the instruction or indirectly by the B, X and SP pointers.

The device has 128 bytes of RAM. Sixteen bytes of RAM are mapped as "registers" at addresses 0F0 to 0FF Hex. These registers can be loaded immediately, and also decremented and tested with the DRSZ (decrement register and skip if zero) instruction. The memory pointer registers X, SP, and B are memory mapped into this space at address locations 0FC to 0FE Hex respectively, with the other registers (other than reserved register 0FF) being available for general usage.

The instruction set permits any bit in memory to be set, reset or tested. All I/O and registers (except A and PC) are memory mapped; therefore, I/O bits and register bits can be directly and individually set, reset and tested. The accumulator (A) bits can also be directly and individually tested.

Reset

The RESET input when pulled low initializes the microcontroller. Initialization will occur whenever the RESET input is pulled low. Upon initialization, the data and configuration registers for Ports L, G, and C are cleared, resulting in these Ports being initialized to the TRI-STATE mode. Pin G1 of the G Port is an exception (as noted below) since pin G1 is dedicated as the WATCHDOG and/or Clock Monitor error output pin. Port D is initialized high with RESET. The PC, PSW, CNTRL, ICNTRL, and T2CNTRL control registers are cleared. The Multi-Input Wakeup registers WKEN, WKEDG, and WKPND are cleared. The A/D control register ENAD is cleared, resulting in the ADC being powered down initially. The Stack Pointer, SP, is initialized to 06F Hex.

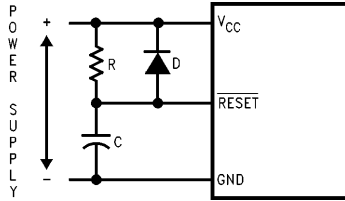
The device comes out of reset with both the WATCHDOG logic and the Clock Monitor detector armed, and with both the WATCHDOG service window bits set and the Clock Monitor bit set. The WATCHDOG and Clock Monitor detector circuits are inhibited during reset. The WATCHDOG service window bits are initialized to the maximum WATCHDOG service window of 64k t_c clock cycles. The Clock Monitor bit

Reset (Continued)

is initialized high, and will cause a Clock Monitor error following reset if the clock has not reached the minimum specified frequency at the termination of reset. A Clock Monitor error will cause an active low error output on pin G1. This error output will continue until $16 t_c - 32 t_c$ clock cycles following the clock frequency reaching the minimum specified value, at which time the G1 output will enter the TRI-STATE mode.

The external RC network shown in Figure 4 should be used to ensure that the $\overline{\text{RESET}}$ pin is held low until the power supply to the chip stabilizes.

Note: In continued state of reset, the device will draw excessive current.



TL/DD/12523-6

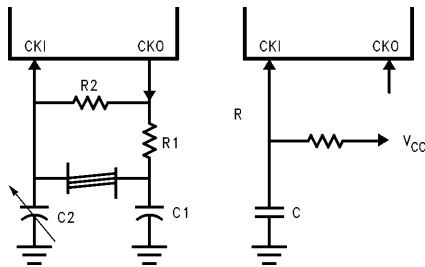
$RC > 5 \times \text{Power Supply Rise Time}$

FIGURE 4. Recommended Reset Circuit

Oscillator Circuits

The chip can be driven by a clock input on the CKI input pin which can be between DC and 10 MHz. The CKO output clock is on pin G7 (crystal configuration). The CKI input frequency is divided down by 10 to produce the instruction cycle clock ($1/t_c$).

Figure 5 shows the Crystal and R/C diagrams.



TL/DD/12523-7

FIGURE 5. Crystal and R/C Oscillator Diagrams

CRYSTAL OSCILLATOR

CKI and CKO can be connected to make a closed loop crystal (or resonator) controlled oscillator.

Table I shows the component values required for various standard crystal values.

TABLE I. Crystal Oscillator Configuration, $T_A = 25^\circ\text{C}$

| R1 (k Ω) | R2 (M Ω) | C1 (pF) | C2 (pF) | CKI Freq (MHz) | Conditions |
|------------------|------------------|---------|---------|----------------|---------------|
| 0 | 1 | 30 | 30-36 | 10 | $V_{CC} = 5V$ |
| 0 | 1 | 30 | 30-36 | 4 | $V_{CC} = 5V$ |
| 0 | 1 | 200 | 100-150 | 0.455 | $V_{CC} = 5V$ |

R/C OSCILLATOR

By selecting CKI as a single pin oscillator input, a single pin R/C oscillator circuit can be connected to it. CKO is available as a general purpose input, and/or HALT restart pin.

Table II shows the variation in the oscillator frequencies as functions of the component (R and C) values.

TABLE II. R/C Oscillator Configuration, $T_A = 25^\circ\text{C}$

| R (k Ω) | C (pF) | CKI Freq (MHz) | Instr. Cycle (μs) | Conditions |
|-----------------|--------|----------------|--------------------------------|---------------|
| 3.3 | 82 | 2.2 to 2.7 | 3.7 to 4.6 | $V_{CC} = 5V$ |
| 5.6 | 100 | 1.1 to 1.3 | 7.4 to 9.0 | $V_{CC} = 5V$ |
| 6.8 | 100 | 0.9 to 1.1 | 8.8 to 10.8 | $V_{CC} = 5V$ |

Note: $3k \leq R \leq 200k$

$50 \text{ pF} \leq C \leq 200 \text{ pF}$

Control Registers

CNTRL REGISTER (ADDRESS X'00EE)

The Timer1 (T1) and MICROWIRE/PLUS control register contains the following bits:

| | |
|-----------|---------------------------------------------------------------------------------------------------|
| SL1 & SL0 | Select the MICROWIRE/PLUS clock divide by (00 = 2, 01 = 4, 1x = 8) |
| IEDG | External interrupt edge polarity select (0 = Rising edge, 1 = Falling edge) |
| MSEL | Selects G5 and G4 as MICROWIRE/PLUS signals SK and SO respectively |
| T1C0 | Timer T1 Start/Stop control in timer Timer T1 Underflow Interrupt Pending Flag in timer mode 3 |
| T1C1 | Timer T1 mode control bit |
| T1C2 | Timer T1 mode control bit |
| T1C3 | Timer T1 mode control bit |

| | | | | | | | |
|-------|------|------|------|------|------|-----|-------|
| T1C3 | T1C2 | T1C1 | T1C0 | MSEL | IEDG | SL1 | SL0 |
| Bit 7 | | | | | | | Bit 0 |

PSW REGISTER (ADDRESS X'00EF)

The PSW register contains the following select bits:

| | |
|--------|---------------------------------------------------------------------------------------------------------------|
| GIE | Global interrupt enable (enables interrupts) |
| EXEN | Enable external interrupt |
| BUSY | MICROWIRE/PLUS busy shifting flag |
| EXPND | External interrupt pending |
| T1ENA | Timer T1 Interrupt Enable for Timer Underflow or T1A Input capture edge |
| T1PNDA | Timer T1 Interrupt Pending Flag (Autoreload RA in mode 1, T1 Underflow in Mode 2, T1A capture edge in mode 3) |
| C | Carry Flag |
| HC | Half Carry Flag |

| | | | | | | | |
|-------|---|--------|-------|-------|------|------|-------|
| HC | C | T1PNDA | T1ENA | EXPND | BUSY | EXEN | GIE |
| Bit 7 | | | | | | | Bit 0 |

The Half-Carry bit is also affected by all the instructions that affect the Carry flag. The SC (Set Carry) and RC (Reset Carry) instructions will respectively set or clear both the carry flags. In addition to the SC and RC instructions, ADC, SUBC, RRC and RLC instructions affect the Carry and Half Carry flags.

ICNTRL REGISTER (ADDRESS X'00E8)

The ICNTRL register contains the following bits:

| | |
|---------|---------------------------------------------------------------------------------------|
| T1ENB | Timer T1 Interrupt Enable for T1B Input capture edge |
| T1PNDB | Timer T1 Interrupt Pending Flag for T1B capture edge |
| WEN | Enable MICROWIRE/PLUS interrupt |
| WPND | MICROWIRE/PLUS interrupt pending |
| T0EN | Timer T0 Interrupt Enable (Bit 12 toggle) |
| T0PND | Timer T0 Interrupt pending |
| LPENL | Port Interrupt Enable (Multi-Input Wakeup/Interrupt) Bit 7 could be used as a flag |
| T2CNTRL | Register (Address X'00C6) |

| | | | | | | | |
|--------|------|-------|------|------|-----|--------|-------|
| Unused | LPEN | T0PND | T0EN | WPND | WEN | T1PNDB | T1ENB |
| Bit 7 | | | | | | | Bit 0 |

The T2CNTRL register contains the following bits:

| | |
|--------|-----------------------------------------------------------------------------------------------------------------|
| T2ENB | Timer T2 Interrupt Enable for T2B Input capture edge |
| T2PNDB | Timer T2 Interrupt Pending Flag for T2B capture edge |
| T2ENA | Timer T2 Interrupt Enable for Timer Underflow or T2A Input capture edge |
| T2PNDA | Timer T2 Interrupt Pending Flag (Autoreload RA in mode 1, T2 Underflow in mode 2, T2A capture edge in mode 3) |
| T2C0 | Timer T2 Start/Stop control in timer modes 1 and 2 Timer T2 Underflow Interrupt Pending Flag in timer mode 3 |
| T2C1 | Timer T2 mode control bit |
| T2C2 | Timer T2 mode control bit |
| T2C3 | Timer T2 mode control bit |

| | | | | | | | |
|-------|------|------|------|--------|-------|--------|-------|
| T2C3 | T2C2 | T2C1 | T2C0 | T2PNDA | T2ENA | T2PNDB | T2ENB |
| Bit 7 | | | | | | | Bit 0 |

Timers

The device contains a very versatile set of timers (T0, T1, T2). All timers and associated autoreload/capture registers power up containing random data.

TIMER T0 (IDLE TIMER)

The device supports applications that require maintaining real time and low power with the IDLE mode. This IDLE mode support is furnished by the IDLE timer T0, which is a 16-bit timer. The Timer T0 runs continuously at the fixed rate of the instruction cycle clock, t_c . The user cannot read or write to the IDLE Timer T0, which is a count down timer.

The Timer T0 supports the following functions:

- Exit out of the Idle Mode (See Idle Mode description)
- WATCHDOG logic (See WATCHDOG description)
- Start up delay out of the HALT mode

The IDLE Timer T0 can generate an interrupt when the thirteenth bit toggles. This toggle is latched into the T0PND pending flag, and will occur every 4 ms at the maximum clock frequency ($t_c = 1$ s). A control flag T0EN allows the interrupt from the thirteenth bit of Timer T0 to be enabled or disabled. Setting T0EN will enable the interrupt, while resetting it will disable the interrupt.

TIMER T1 AND TIMER T2

The device has a set of two powerful timer/counter blocks, T1 and T2. The associated features and functioning of a timer block are described by referring to the timer block Tx. Since the two timer blocks, T1 and T2, are identical, all comments are equally applicable to either timer block.

Each timer block consists of a 16-bit timer, Tx, and two supporting 16-bit autoreload/capture registers, RxA and RxB. Each timer block has two pins associated with it, TxA and TxB. The pin TxA supports I/O required by the timer block, while the pin TxB is an input to the timer block. The powerful and flexible timer block allows the device to easily perform all timer functions with minimal software overhead. The timer block has three operating modes: Processor Independent PWM mode, External Event Counter mode, and Input Capture mode.

The control bits TxC3, TxC2, and TxC1 allow selection of the different modes of operation.

Mode 1. Processor Independent PWM Mode

As the name suggests, this mode allows the device to generate a PWM signal with very minimal user intervention.

The user only has to define the parameters of the PWM signal (ON time and OFF time). Once begun, the timer block will continuously generate the PWM signal completely independent of the microcontroller. The user software services the timer block only when the PWM parameters require updating.

In this mode the timer Tx counts down at a fixed rate of t_c . Upon every underflow the timer is alternately reloaded with the contents of supporting registers, RxA and RxB. The very first underflow of the timer causes the timer to reload from the register RxA. Subsequent underflows cause the timer to be reloaded from the registers alternately beginning with the register RxB.

The Tx Timer control bits, TxC3, TxC2 and TxC1 set up the timer for PWM mode operation.

Figure 6 shows a block diagram of the timer in PWM mode.

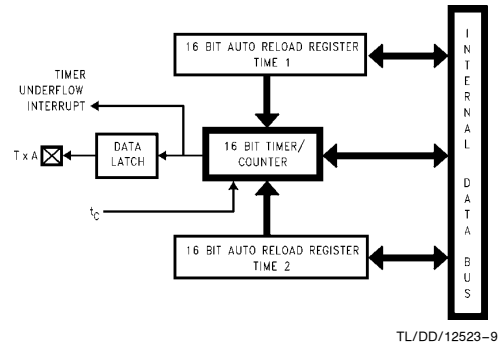


FIGURE 6. Timer in PWM Mode

The underflows can be programmed to toggle the TxA output pin. The underflows can also be programmed to generate interrupts.

Underflows from the timer are alternately latched into two pending flags, TxPNDA and TxPNDB. The user must reset these pending flags under software control. Two control enable flags, TxENA and TxENB, allow the interrupts from the timer underflow to be enabled or disabled. Setting the timer enable flag TxENA will cause an interrupt when a timer underflow causes the RxA register to be reloaded into the timer. Setting the timer enable flag TxENB will cause an interrupt when a timer underflow causes the RxB register to be reloaded into the timer. Resetting the timer enable flags will disable the associated interrupts.

Either or both of the timer underflow interrupts may be enabled. This gives the user the flexibility of interrupting once per PWM period on either the rising or falling edge of the PWM output. Alternatively, the user may choose to interrupt on both edges of the PWM output.

Mode 2. External Event Counter Mode

This mode is quite similar to the processor independent PWM mode described above. The main difference is that the timer, Tx, is clocked by the input signal from the TxA pin. The Tx timer control bits, TxC3, TxC2 and TxC1 allow the timer to be clocked either on a positive or negative edge from the TxA pin. Underflows from the timer are latched into the TxPNDA pending flag. Setting the TxENA control flag will cause an interrupt when the timer underflows.

In this mode the input pin TxB can be used as an independent positive edge sensitive interrupt input if the TxENB control flag is set. The occurrence of a positive edge on the TxB input pin is latched into the TxPNDB flag.

Figure 7 shows a block diagram of the timer in External Event Counter mode.

Note: The PWM output is not available in this mode since the TxA pin is being used as the counter input clock.

Mode 3. Input Capture Mode

The device can precisely measure external frequencies or time external events by placing the timer block, Tx, in the input capture mode.

Timers (Continued)

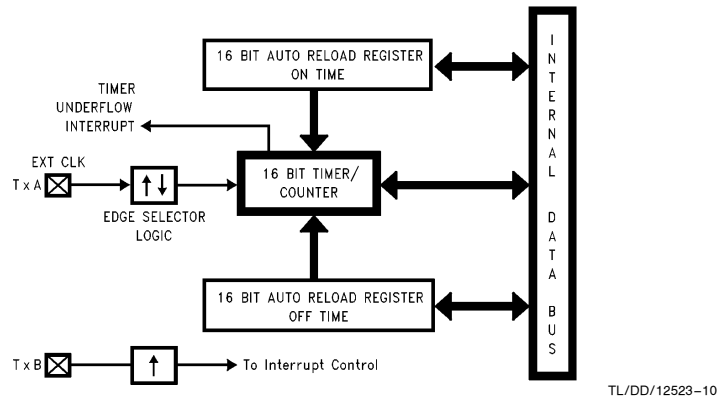


FIGURE 7. Timer in External Event Counter Mode

In this mode, the timer Tx is constantly running at the fixed t_c rate. The two registers, RxA and RxB, act as capture registers. Each register acts in conjunction with a pin. The register RxA acts in conjunction with the TxA pin and the register RxB acts in conjunction with the TxB pin.

The timer value gets copied over into the register when a trigger event occurs on its corresponding pin. Control bits, TxC3, TxC2 and TxC1, allow the trigger events to be specified either as a positive or a negative edge. The trigger condition for each input pin can be specified independently.

The trigger conditions can also be programmed to generate interrupts. The occurrence of the specified trigger condition on the TxA and TxB pins will be respectively latched into the pending flags, TxPNDA and TxPNDB. The control flag TxENA allows the interrupt on TxA to be either enabled or disabled. Setting the TxENA flag enables interrupts to be generated when the selected trigger condition occurs on the TxA pin. Similarly, the flag TxENB controls the interrupts from the TxB pin.

Underflows from the timer can also be programmed to generate interrupts. Underflows are latched into the timer TxCO pending flag (the TxCO control bit serves as the timer underflow interrupt pending flag in the Input Capture mode). Consequently, the TxCO control bit should be reset when entering the Input Capture mode. The timer underflow interrupt is enabled with the TxENA control flag. When a TxA interrupt

occurs in the Input Capture mode, the user must check both whether a TxA input capture or a timer underflow (or both) caused the interrupt.

Figure 8 shows a block diagram of the timer in Input Capture mode.

TIMER CONTROL FLAGS

The timers T1 and T2 have identical control structures. The control bits and their functions are summarized below.

| | |
|--------|-----------------------------------------------------------------------------------------------------------------------------|
| TxC0 | Timer Start/Stop control in Modes 1 and 2 (Processor Independent PWM and External Event Counter), where 1 = Start, 0 = Stop |
| | Timer Underflow Interrupt Pending Flag in Mode 3 (Input Capture) |
| TxPNDA | Timer Interrupt Pending Flag |
| TxPNDB | Timer Interrupt Pending Flag |
| TxENA | Timer Interrupt Enable Flag |
| TxENB | Timer Interrupt Enable Flag |
| | 1 = Timer Interrupt Enabled |
| | 0 = Timer Interrupt Disabled |
| TxC3 | Timer mode control |
| TxC2 | Timer mode control |
| TxC1 | Timer mode control |

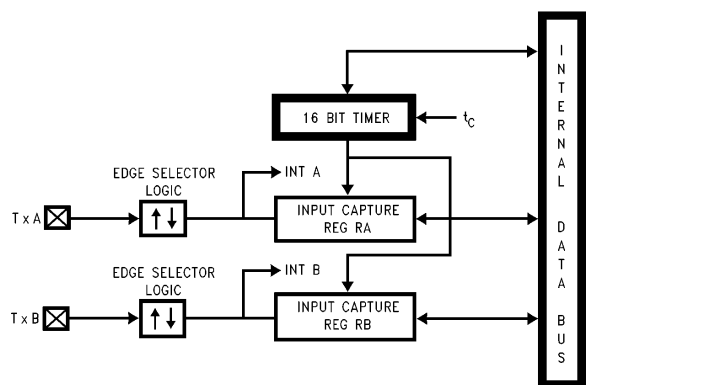


FIGURE 8. Timer in Input Capture Mode

Timers (Continued)

The timer mode control bits (TxC3, TxC2 and TxC1) are detailed below:

| TxC3 | TxC2 | TxC1 | Timer Mode | Interrupt A Source | Interrupt B Source | Timer Counts On |
|------|------|------|--------------------------------------------------------------|-------------------------------------|--------------------|-----------------|
| 0 | 0 | 0 | MODE 2 (External Event Counter) | Timer Underflow | Pos. TxB Edge | TxA Pos. Edge |
| 0 | 0 | 1 | MODE 2 (External Event Counter) | Timer Underflow | Pos. TxB Edge | TxA Neg. Edge |
| 1 | 0 | 1 | MODE 1 (PWM) TxA Toggle | Autoreload RA | Autoreload RB | t_c |
| 1 | 0 | 0 | MODE 1 (PWM) No TxA Toggle | Autoreload RA | Autoreload RB | t_c |
| 0 | 1 | 0 | MODE 3 (Capture) Captures: TxA Pos. Edge TxB Pos. Edge | Pos. TxA Edge or Timer Underflow | Pos. TxB Edge | t_c |
| 1 | 1 | 0 | MODE 3 (Capture) Captures: TxA Pos. Edge TxB Neg. Edge | Pos. TxA Edge or Timer Underflow | Neg. TxB Edge | t_c |
| 0 | 1 | 1 | MODE 3 (Capture) Captures: TxA Neg. Edge TxB Pos. Edge | Neg. TxB Edge or Timer Underflow | Pos. TxB Edge | t_c |
| 1 | 1 | 1 | MODE 3 (Capture) Captures: TxA Neg. Edge TxB Neg. Edge | Neg. TxA Edge or Timer Underflow | Neg. TxB Edge | t_c |

Power Save Modes

The device offers the user two power save modes of operation: HALT and IDLE. In the HALT mode, all microcontroller activities are stopped. In the IDLE mode, the on-board oscillator circuitry and timer T0 are active but all other microcontroller activities are stopped. In either mode, all on-board RAM, registers, I/O states, and timers (with the exception of T0) are unaltered.

HALT MODE

The device is placed in the HALT mode by writing a "1" to the HALT flag (G7 data bit). All microcontroller activities, including the clock, timers, and A/D converter, are stopped. The WATCHDOG logic is disabled during the HALT mode. However, the clock monitor circuitry if enabled remains active and will cause the WATCHDOG output pin (WDOUT) to go low. If the HALT mode is used and the user does not want to activate the WDOUT pin, the Clock Monitor should be disabled after the device comes out of reset (resetting the Clock Monitor control bit with the first write to the WDSVR register). In the HALT mode, the power requirements of the device are minimal and the applied voltage (V_{CC}) may be decreased to V_r ($V_r = 2.0V$) without altering the state of the machine.

The device supports three different ways of exiting the HALT mode. The first method of exiting the HALT mode is with the Multi-Input Wakeup feature on the L port. The second method is with a low to high transition on the CKO (G7) pin. This method precludes the use of the crystal clock configuration (since CKO becomes a dedicated output), and so may be used with an RC clock configuration. The third method of exiting the HALT mode is by pulling the RESET pin low.

Since a crystal or ceramic resonator may be selected as the oscillator, the Wakeup signal is not allowed to start the chip running immediately since crystal oscillators and ceramic resonators have a delayed start up time to reach full amplitude and frequency stability. The IDLE timer is used to generate a fixed delay to ensure that the oscillator has indeed stabilized before allowing instruction execution. In this case, upon detecting a valid Wakeup signal, only the oscillator circuitry is enabled. The IDLE timer is loaded with a value of 256 and is clocked with the t_c instruction cycle clock. The t_c clock is derived by dividing the oscillator clock down by a factor of 10. The Schmitt trigger following the CK1 inverter on the chip ensures that the IDLE timer is clocked only when the oscillator has a sufficiently large amplitude to meet the Schmitt trigger specifications. This Schmitt trigger is not part of the oscillator closed loop. The startup timeout from the IDLE timer enables the clock signals to be routed to the rest of the chip.

If an RC clock option is being used, the fixed delay is introduced optionally. A control bit, CLKDLY, mapped as configuration bit G7, controls whether the delay is to be introduced or not. The delay is included if CLKDLY is set, and excluded if CLKDLY is reset. The CLKDLY bit is cleared on reset.

The WATCHDOG detector circuit is inhibited during the HALT mode. However, the clock monitor circuit if enabled remains active during HALT mode in order to ensure a clock monitor error if the device inadvertently enters the HALT mode as a result of a runaway program or power glitch.

IDLE MODE

The device is placed in the IDLE mode by writing a "1" to the IDLE flag (G6 data bit). In this mode, all activity, except

Power Save Modes (Continued)

the associated on-board oscillator circuitry, the WATCHDOG logic, the clock monitor and the IDLE Timer T0, is stopped.

As with the HALT mode, the device can be returned to normal operation with a reset, or with a Multi-Input Wake Up from the L Port. Alternately, the microcontroller resumes normal operation from the IDLE mode when the thirteenth bit (representing 4.096 ms at internal clock frequency of 1 MHz, $t_c = 1 \mu s$) of the IDLE Timer toggles.

This toggle condition of the thirteenth bit of the IDLE Timer T0 is latched into the T0PND pending flag.

The user has the option of being interrupted with a transition on the thirteenth bit of the IDLE Timer T0. The interrupt can be enabled or disabled via the T0EN control bit. Setting the T0EN flag enables the interrupt and vice versa.

The user can enter the IDLE mode with the Timer T0 interrupt enabled. In this case, when the T0PND bit gets set, the device will first execute the Timer T0 interrupt service routine and then return to the instruction following the "Enter Idle Mode" instruction.

Alternatively, the user can enter IDLE mode with the IDLE Timer T0 interrupt disabled. In this case, the device will resume normal operation with the instruction immediately following the "Enter IDLE Mode" instruction.

Note: It is necessary to program two NOP instructions following both the set HALT mode and set IDLE mode instructions. These NOP instructions are necessary to allow clock resynchronization following the HALT or IDLE modes.

Due to the onboard 8k EPROM with port recreation logic, the HALT/IDLE current is much higher compared to the equivalent masked device.

Multi-Input Wake Up

The Multi-Input Wake Up feature is used to return (Wake Up) the device from either the HALT or IDLE modes. Alternately Multi-Input Wake Up/Interrupt feature may also be used to generate up to 8 edge selectable external interrupts.

Figure 9 shows the Multi-Input Wake Up logic.

The Multi-Input Wakeup feature utilizes the L Port. The user selects which particular L port bit (or combination of L Port bits) will cause the device to exit the HALT or IDLE modes. The selection is done through the Reg: WKEN. The Reg: WKEN is an 8-bit read/write register, which contains a control bit for every L port bit. Setting a particular WKEN bit enables a Wake Up from the associated L port pin.

The user can select whether the trigger condition on the selected L Port pin is going to be either a positive edge (low to high transition) or a negative edge (high to low transition). This selection is made via the Reg: WKEDG, which is an 8-bit control register with a bit assigned to each L Port pin. Setting the control bit will select the trigger condition to be a negative edge on that particular L Port pin. Resetting the bit selects the trigger condition to be a positive edge. Changing an edge select entails several steps in order to avoid a pseudo Wake Up condition as a result of the edge change. First, the associated WKEN bit should be reset, followed by the edge select change in WKEDG. Next, the associated WKPND bit should be cleared, followed by the associated WKEN bit being re-enabled.

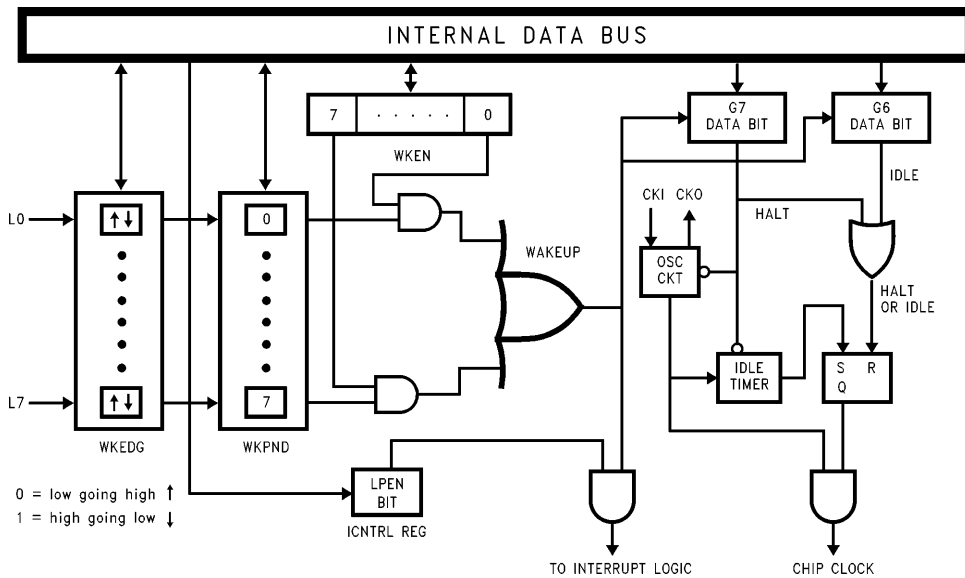


FIGURE 9. Multi-Input Wake Up Logic

TL/DD/12523-12

Multi-Input Wake Up (Continued)

An example may serve to clarify this procedure. Suppose we wish to change the edge select from positive (low going high) to negative (high going low) for L Port bit 5, where bit 5 has previously been enabled for an input interrupt. The program would be as follows:

```
RBIT 5, WKEN
SBIT 5, WKEDG
RBIT 5, WKPND
SBIT 5, WKEN
```

If the L port bits have been used as outputs and then changed to inputs with Multi-Input Wake Up/Interrupt, a safety procedure should also be followed to avoid inherited pseudo Wake Up conditions. After the selected L port bits have been changed from output to input but before the associated WKEN bits are enabled, the associated edge select bits in WKEDG should be set or reset for the desired edge selects, followed by the associated WKPND bits being cleared.

This same procedure should be used following reset, since the L port inputs are left floating as a result of reset.

The occurrence of the selected trigger condition for Multi-Input Wake Up is latched into a pending register called WKPND. The respective bits of the WKPND register will be set on the occurrence of the selected trigger edge on the corresponding Port L pin. The user has the responsibility of clearing these pending flags. Since WKPND is a pending register for the occurrence of selected wakeup conditions, the device will not enter the HALT mode if any Wake Up bit is both enabled and pending. Consequently, the user has the responsibility of clearing the pending flags before attempting to enter the HALT mode.

The WKEN, WKPND and WKEDG are all read/write registers, and are cleared at reset.

PORT L INTERRUPTS

Port L provides the user with an additional eight fully selectable, edge sensitive interrupts which are all vectored into the same service subroutine.

The interrupt from Port L shares logic with the Wake Up circuitry. The register WKEN allows interrupts from Port L to be individually enabled or disabled. The register WKEDG specifies the trigger condition to be either a positive or a negative edge. Finally, the register WKPND latches in the pending trigger conditions.

The GIE (Global Interrupt Enable) bit enables the interrupt function. A control flag, LPEN, functions as a global interrupt enable for Port L interrupts. Setting the LPEN flag will enable interrupts and vice versa. A separate global pending flag is not needed since the register WKPND is adequate.

Since Port L is also used for waking the device out of the HALT or IDLE modes, the user can elect to exit the HALT or IDLE modes either with or without the interrupt enabled. If he elects to disable the interrupt, then the device will restart execution from the instruction immediately following the instruction that placed the microcontroller in the HALT or IDLE modes. In the other case, the device will first execute the interrupt service routine and then revert to normal operation.

The Wake Up signal will not start the chip running immediately since crystal oscillators or ceramic resonators have a finite start up time. The IDLE Timer (T0) generates a fixed delay to ensure that the oscillator has indeed stabilized before allowing the device 40 execute instructions. In this case, upon detecting a valid Wake Up signal, only the oscillator circuitry and the IDLE Timer T0 are enabled. The IDLE Timer is loaded with a value of 256 and is clocked from the t_c instruction cycle clock. The t_c clock is derived by dividing down the oscillator clock by a factor of 10. A Schmitt trigger following the CKI on-chip inverter ensures that the IDLE timer is clocked only when the oscillator has a sufficiently large amplitude to meet the Schmitt trigger specifications. This Schmitt trigger is not part of the oscillator closed loop. The startup timeout from the IDLE timer enables the clock signals to be routed to the rest of the chip.

If the RC clock option is used, the fixed delay is under software control. A control flag, CLKDLY, in the G7 configuration bit allows the clock start up delay to be optionally inserted. Setting CLKDLY flag high will cause clock start up delay to be inserted and resetting it will exclude the clock start up delay. The CLKDLY flag is cleared during reset, so the clock start up delay is not present following reset with the RC clock options.

A/D Converter

The device contains an 8-channel, multiplexed input, successive approximation, A/D converter. Two dedicated pins, VREF and AGND are provided for voltage reference.

OPERATING MODES

The A/D converter supports ratiometric measurements. It supports both Single Ended and Differential modes of operation.

Four specific analog channel selection modes are supported. These are as follows:

Allow any specific channel to be selected at one time. The A/D converter performs the specific conversion requested and stops.

Allow any specific channel to be scanned continuously. In other words, the user will specify the channel and the A/D converter will keep on scanning it continuously. The user can come in at any arbitrary time and immediately read the result of the last conversion. The user does not have to wait for the current conversion to be completed.

Allow any differential channel pair to be selected at one time. The A/D converter performs the specific differential conversion requested and stops.

Allow any differential channel pair to be scanned continuously. In other words, the user will specify the differential channel pair and the A/D converter will keep on scanning it continuously. The user can come in at any arbitrary time and immediately read the result of the last differential conversion. The user does not have to wait for the current conversion to be completed.

The A/D converter is supported by two memory mapped registers, the result register and the mode control register. When the device is reset, the control register is cleared and the A/D is powered down. The A/D result register has unknown data following reset.

A/D Converter (Continued)

A/D Control Register

A control register, Reg: ENAD, contains 3 bits for channel selection, 3 bits for prescaler selection, and 2 bits for mode selection. An A/D conversion is initiated by writing to the ENAD control register. The result of the conversion is available to the user from the A/D result register, Reg: ADRSLT.

Reg: ENAD

| Channel Select | Mode Select | Prescaler Select |
|----------------|-------------|------------------|
| Bits 7, 6, 5 | Bits 4, 3 | Bits 2, 1, 0 |

CHANNEL SELECT

This 3-bit field selects one of eight channels to be the V_{IN+} . The mode selection determines the V_{IN-} input.

Single Ended mode:

| Bit 7 | Bit 6 | Bit 5 | Channel No. |
|-------|-------|-------|-------------|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 2 |
| 0 | 1 | 1 | 3 |
| 1 | 0 | 0 | 4 |
| 1 | 0 | 1 | 5 |
| 1 | 1 | 0 | 6 |
| 1 | 1 | 1 | 7 |

Differential mode:

| Bit 7 | Bit 6 | Bit 5 | Channel Pairs (+, -) |
|-------|-------|-------|----------------------|
| 0 | 0 | 0 | 0, 1 |
| 0 | 0 | 1 | 1, 0 |
| 0 | 1 | 0 | 2, 3 |
| 0 | 1 | 1 | 3, 2 |
| 1 | 0 | 0 | 4, 5 |
| 1 | 0 | 1 | 5, 4 |
| 1 | 1 | 0 | 6, 7 |
| 1 | 1 | 1 | 7, 6 |

MODE SELECT

This 2-bit field is used to select the mode of operation (single conversion, continuous conversions, differential, single ended) as shown in the following table.

| Bit 4 | Bit 3 | Mode |
|-------|-------|---------------------------------------------------------------------------------|
| 0 | 0 | Single Ended mode, single conversion |
| 0 | 1 | Single Ended mode, continuous scan of a single channel into the result register |
| 1 | 0 | Differential mode, single conversion |
| 1 | 1 | Differential mode, continuous scan of a channel pair into the result register |

PRESCALER SELECT

This 3-bit field is used to select one of the seven prescaler clocks for the A/D converter. The prescaler also allows the A/D clock inhibit power saving mode to be selected. The following table shows the various prescaler options.

| Bit 2 | Bit 1 | Bit 0 | Clock Select |
|-------|-------|-------|-------------------|
| 0 | 0 | 0 | Inhibit A/D clock |
| 0 | 0 | 1 | Divide by 1 |
| 0 | 1 | 0 | Divide by 2 |
| 0 | 1 | 1 | Divide by 4 |
| 1 | 0 | 0 | Divide by 6 |
| 1 | 0 | 1 | Divide by 12 |
| 1 | 1 | 0 | Divide by 8 |
| 1 | 1 | 1 | Divide by 16 |

ADC Operation

The A/D converter interface works as follows. Writing to the A/D control register ENAD initiates an A/D conversion unless the prescaler value is set to 0, in which case the ADC clock is stopped and the ADC is powered down. The conversion sequence starts at the beginning of the write to ENAD operation powering up the ADC. At the first falling edge of the converter clock following the write operation (not counting the falling edge if it occurs at the same time as the write operation ends), the sample signal turns on for two clock cycles. The ADC is selected in the middle of the sample period. If the ADC is in single conversion mode, the conversion complete signal from the ADC will generate a power down for the A/D converter. If the ADC is in continuous mode, the conversion complete signal will restart the conversion sequence by deselecting the ADC for one converter clock cycle before starting the next sample. The ADC 8-bit result is loaded into the A/D result register (ADRSLT) except during LOAD clock high, which prevents transient data (resulting from the ADC writing a new result over an old one) being read from ADRSLT.

Inadvertant changes to the ENAD register during conversion are prevented by the control logic of the A/D. Any attempt to write any bit of the ENAD Register except ADBSY, while ADBSY is a one, is ignored. ADBSY must be cleared either by completion of an A/D conversion or by the user before the prescaler, conversion mode or channel select values can be changed. After stopping the current conversion, the user can load different values for the prescaler, conversion mode or channel select and start a new conversion in one instruction.

It is important for the user to realize that, when used in differential mode, only the positive input to the A/D converter is sampled and held. The negative input is constantly connected and should be held stable for the duration of the conversion. Failure to maintain a stable negative input will result in incorrect conversion.

PRESCALER

The A/D Converter (ADC) contains a prescaler option which allows seven different clock selections. The A/D clock frequency is equal to CKI divided by the prescaler value. Note that the prescaler value must be chosen such that the A/D clock falls within the specified range. The maximum A/D frequency is 1.67 MHz. This equates to a 600 ns ADC clock cycle.

The A/D converter takes 12 ADC clock cycles to complete a conversion. Thus the minimum ADC conversion time is 7.2 μ s when a prescaler of 6 has been selected. These 12 ADC clock cycles necessary for a conversion consist of 1 cycle at the beginning for reset, 2 cycles for sampling, 8 cycles for converting, and 1 cycle for loading the result into the A/D result register (ADRSLT). This A/D result register is a read-only register. The user cannot write into ADRSLT.

A/D Converter (Continued)

The prescaler also allows an A/D clock inhibit option, which saves power by powering down the A/D when it is not in use.

Note: The A/D converter is also powered down when the device is in either the HALT or IDLE modes. If the ADC is running when the device enters the HALT or IDLE modes, the ADC will power down during the HALT or IDLE, and then will reinitialize the conversion when the device comes out of the HALT or IDLE modes.

Analog Input and Source Resistance Considerations

Figure 10 shows the A/D pin model in single-ended mode. The differential mode has a similar A/D pin model. The leads to the analog inputs should be kept as short as possible. Both noise and digital clock coupling to an A/D input can cause conversion errors. The clock lead should be kept away from the analog input line to reduce coupling. The A/D channel input pins do not have any internal output driver circuitry connected to them because this circuitry would load the analog input signals due to output buffer leakage current.

Source impedances greater than 1 k Ω on the analog input lines will adversely affect internal RC charging time during input sampling. As shown in Figure 10, the analog switch to the DAC array is closed only during the 2 A/D cycle sample time. Large source impedances on the analog inputs may result in the DAC array not being charged to the correct voltage levels, causing scale errors.

If large source resistance is necessary, the recommended solution is to slow down the A/D clock speed in proportion to the source resistance. The A/D converter may be operated at the maximum speed for R_S less than 1 k Ω . For R_S greater than 1 k Ω , A/D clock speed needs to be reduced.

For example, with $R_S = 2$ k Ω , the A/D converter may be operated at half the maximum speed. A/D converter clock speed may be slowed down by either increasing the A/D prescaler divide-by or decreasing the CKI clock frequency. The A/D clock speed may be reduced to its minimum frequency of 100 kHz.

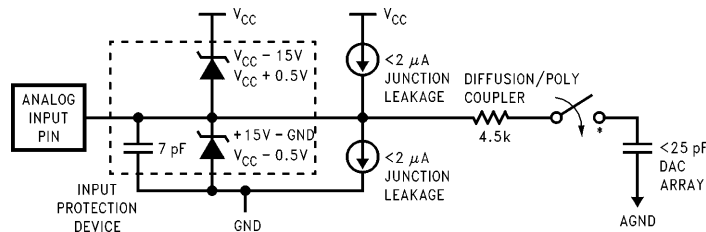
Interrupts

The device supports a vectored interrupt scheme. It supports a total of ten interrupt sources. The following table lists all the possible interrupt sources, their arbitration ranking and the memory locations reserved for the interrupt vector for each source.

Two bytes of program memory space are reserved for each interrupt source. All interrupt sources except the software interrupt are maskable. Each of the maskable interrupts have an Enable bit and a Pending bit. A maskable interrupt is active if its associated enable and pending bits are set. If $GIE = 1$ and an interrupt is active, then the processor will be interrupted as soon as it is ready to start executing an instruction except if the above conditions happen during the Software Trap service routine. This exception is described in the Software Trap sub-section.

The interruption process is accomplished with the INTR instruction (opcode 00), which is jammed inside the Instruction Register and replaces the opcode about to be executed. The following steps are performed for every interrupt:

1. The GIE (Global Interrupt Enable) bit is reset.
2. The address of the instruction about to be executed is pushed into the stack.
3. The PC (Program Counter) branches to address 00FF. This procedure takes 7 t_c cycles to execute.



TL/DD/12523-13

*The analog switch is closed only during the sample time.

FIGURE 10. A/D Pin Model (Single Ended Mode)

Interrupts (Continued)

At this time, since GIE = 0, other maskable interrupts are disabled. The user is now free to do whatever context switching is required by saving the context of the machine in the stack with PUSH instructions. The user would then program a VIS (Vector Interrupt Select) instruction in order to branch to the interrupt service routine of the highest priority interrupt enabled and pending at the time of the VIS. Note that this is not necessarily the interrupt that caused the branch to address location 00FF Hex prior to the context switching.

Thus, if an interrupt with a higher rank than the one which caused the interruption becomes active before the decision of which interrupt to service is made by the VIS, then the interrupt with the higher rank will override any lower ones and will be acknowledged. The lower priority interrupt(s) are still pending, however, and will cause another interrupt immediately following the completion of the interrupt service

routine associated with the higher priority interrupt just serviced. This lower priority interrupt will occur immediately following the RETI (Return from Interrupt) instruction at the end of the interrupt service routine just completed.

Inside the interrupt service routine, the associated pending bit has to be cleared by software. The RETI (Return from Interrupt) instruction at the end of the interrupt service routine will set the GIE (Global Interrupt Enable) bit, allowing the processor to be interrupted again if another interrupt is active and pending.

The VIS instruction looks at all the active interrupts at the time it is executed and performs an indirect jump to the beginning of the service routine of the one with the highest rank. The addresses of the different interrupt service routines, called vectors, are chosen by the user and stored in ROM in a table starting at 01E0 (assuming that VIS is located between 00FF and 01DF). The vectors are 15-bit wide and therefore occupy 2 ROM locations.

| Arbitration Ranking | Source | Description | Vector Address Hi-Low Byte |
|---------------------|----------------|---------------------------------------------|----------------------------|
| (1) Highest | Software | INTR Instruction | 0yFE–0yFF |
| (2) | Reserved | for Future Use | 0yFC–0yFD |
| (3) | External | Pin G0 Edge | 0yFA–0yFB |
| (4) | Timer T0 | Underflow | 0yF8–0yF9 |
| (5) | Timer T1 | T1A/Underflow | 0yF6–0yF7 |
| (6) | Timer T1 | T1B | 0yF4–0yF5 |
| (7) | MICROWIRE/PLUS | BUSY Goes Low | 0yF2–0yF3 |
| (8) | Reserved | for Future Use | 0yF0–0yF1 |
| (9) | Reserved | for UART | 0yEE–0yEF |
| (10) | Reserved | for UART | 0yEC–0yED |
| (11) | Timer T2 | T2A/Underflow | 0yEA–0yEB |
| (12) | Timer T2 | T2B | 0yE8–0yE9 |
| (13) | Reserved | for Future Use | 0yE6–0yE7 |
| (14) | Reserved | for Future Use | 0yE4–0yE5 |
| (15) | Port L/Wakeup | Port L Edge | 0yE2–0yE3 |
| (16) Lowest | Default | VIS Instr. Execution without Any Interrupts | 0yE0–0yE1 |

y is VIS page, y ≠ 0

Interrupts (Continued)

VIS and the vector table must be located in the same 256-byte block (0y00 to 0yFF) except if VIS is located at the last address of a block. In this case, the table must be in the next block. The vector table cannot be inserted in the first 256-byte block.

The vector of the maskable interrupt with the lowest rank is located at 0yE0 (Hi-Order byte) and 0yE1 (Lo-Order byte) and so forth in increasing rank number. The vector of the maskable interrupt with the highest rank is located at 0yFA (Hi-Order byte) and 0yFB (Lo-Order byte).

The Software Trap has the highest rank and its vector is located at 0yFE and 0yFF.

If, by accident, a VIS gets executed and no interrupt is active, then the PC (Program Counter) will branch to a vector located at 0yE0–0yE1.

WARNING

A Default VIS interrupt handler routine must be present. As a minimum, this handler should confirm that the GIE bit is cleared (this indicates that the interrupt sequence has been taken), take care of any required housekeeping, restore context and return. Some sort of Warm Restart procedure should be implemented. These events can occur without any error on the part of the system designer or programmer.

Note: There is always the possibility of an interrupt occurring during an instruction which is attempting to reset the GIE bit or any other interrupt enable bit. If this occurs when a single cycle instruction is being used to reset the interrupt enable bit, the interrupt enable bit will be reset but an interrupt may still occur. This is because interrupt processing is started at the same time as the interrupt bit is being reset. To avoid this scenario, the user should always use a two, three, or four cycle instruction to reset interrupt enable bits.

Figure 11 shows the device Interrupt block diagram.

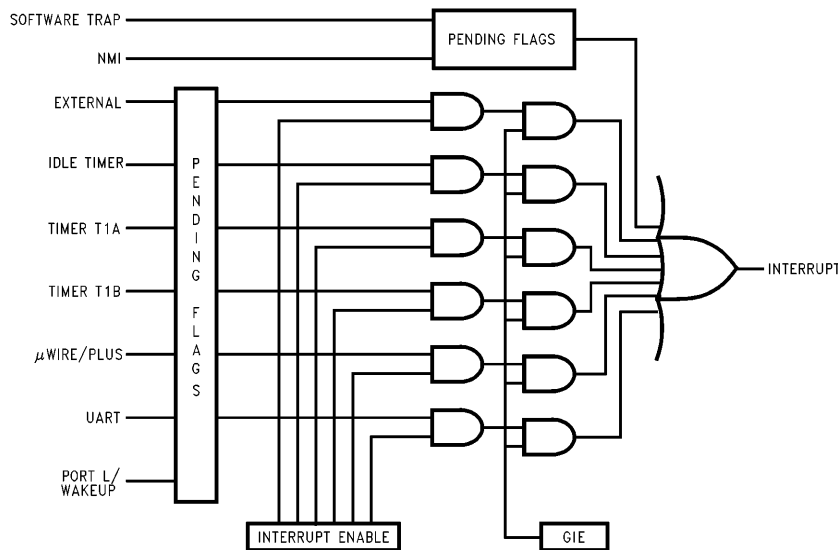


FIGURE 11. Interrupt Block Diagram

TL/DD/12523-14

SOFTWARE TRAP

The Software Trap (ST) is a special kind of non-maskable interrupt which occurs when the INTR instruction (used to acknowledge interrupts) is fetched from ROM and placed inside the instruction register. This may happen when the PC is pointing beyond the available ROM address space or when the stack is over-popped.

When an ST occurs, the user can re-initialize the stack pointer and do a recovery procedure (similar to RESET, but not necessarily containing all of the same initialization procedures) before restarting.

The occurrence of an ST is latched into the ST pending bit. The GIE bit is not affected and the ST pending bit (**not accessible by the user**) is used to inhibit other interrupts and to direct the program to the ST service routine with the VIS instruction. The RPND instruction is used to clear the software interrupt pending bit. This bit is also cleared on reset.

The ST has the highest rank among all interrupts.

Nothing (except another ST) can interrupt an ST being serviced.

WATCHDOG

The device contains a WATCHDOG and clock monitor. The WATCHDOG is designed to detect the user program getting stuck in infinite loops resulting in loss of program control or “runaway” programs. The Clock Monitor is used to detect the absence of a clock or a very slow clock below a specified rate on the CKI pin.

The WATCHDOG consists of two independent logic blocks: WD UPPER and WD LOWER. WD UPPER establishes the upper limit on the service window and WD LOWER defines the lower limit of the service window.

Servicing the WATCHDOG consists of writing a specific value to a WATCHDOG Service Register named WDSVR which is memory mapped in the RAM. This value is composed of three fields, consisting of a 2-bit Window Select, a 5-bit Key Data field, and the 1-bit Clock Monitor Select field. Table III shows the WDSVR register.

TABLE III. WATCHDOG Service Register (WDSVR)

| Window Select | | Key Data | | | | | Clock Monitor |
|---------------|---|----------|---|---|---|---|---------------|
| X | X | 0 | 1 | 1 | 0 | 0 | Y |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

The lower limit of the service window is fixed at 2048 instruction cycles. Bits 7 and 6 of the WDSVR register allow the user to pick an upper limit of the service window.

Table IV shows the four possible combinations of lower and upper limits for the WATCHDOG service window. This flexibility in choosing the WATCHDOG service window prevents any undue burden on the user software.

TABLE IV. WATCHDOG Service Window Select

| WDSVR Bit 7 | WDSVR Bit 6 | Service Window (Lower-Upper Limits) |
|-------------|-------------|-------------------------------------|
| 0 | 0 | 2k–8k t_c Cycles |
| 0 | 1 | 2k–16k t_c Cycles |
| 1 | 0 | 2k–32k t_c Cycles |
| 1 | 1 | 2k–64k t_c Cycles |

TABLE V. WATCHDOG Service Actions

| Key Data | Window Data | Clock Monitor | Action |
|------------|-------------|---------------|---------------------------------------|
| Match | Match | Match | Valid Service: Restart Service Window |
| Don't Care | Mismatch | Don't Care | Error: Generate WATCHDOG Output |
| Mismatch | Don't Care | Don't Care | Error: Generate WATCHDOG Output |
| Don't Care | Don't Care | Mismatch | Error: Generate WATCHDOG Output |

Bits 5, 4, 3, 2 and 1 of the WDSVR register represent the 5-bit Key Data field. The key data is fixed at 01100. Bit 0 of the WDSVR Register is the Clock Monitor Select bit.

Clock Monitor

The Clock Monitor aboard the device can be selected or deselected under program control. The Clock Monitor is guaranteed not to reject the clock if the instruction cycle clock ($1/t_c$) is greater or equal to 10 kHz. This equates to a clock input rate on CKI of greater or equal to 100 kHz.

WATCHDOG Operation

The WATCHDOG and Clock Monitor are disabled during reset. The device comes out of reset with the WATCHDOG armed, the WATCHDOG Window Select (bits 6, 7 of the WDSVR Register) set, and the Clock Monitor bit (bit 0 of the WDSVR Register) enabled. Thus, a Clock Monitor error will occur after coming out of reset, if the instruction cycle clock frequency has not reached a minimum specified value, including the case where the oscillator fails to start.

The WDSVR register can be written to only once after reset and the key data (bits 5 through 1 of the WDSVR Register) must match to be a valid write. This write to the WDSVR register involves two irrevocable choices: (i) the selection of the WATCHDOG service window (ii) enabling or disabling of the Clock Monitor. Hence, the first write to WDSVR Register involves selecting or deselecting the Clock Monitor, select the WATCHDOG service window and match the WATCHDOG key data. Subsequent writes to the WDSVR register will compare the value being written by the user to the WATCHDOG service window value and the key data (bits 7 through 1) in the WDSVR Register. Table V shows the sequence of events that can occur.

The user must service the WATCHDOG at least once before the upper limit of the service window expires. The WATCHDOG may not be serviced more than once in every lower limit of the service window. The user may service the WATCHDOG as many times as wished in the time period between the lower and upper limits of the service window. The first write to the WDSVR Register is also counted as a WATCHDOG service.

WATCHDOG Operation (Continued)

The WATCHDOG has an output pin associated with it. This is the WDOUT pin, on pin 1 of the port G. WDOUT is active low. The WDOUT pin is in the high impedance state in the inactive state. Upon triggering the WATCHDOG, the logic will pull the WDOUT (G1) pin low for an additional $16 t_c$ – $32 t_c$ cycles after the signal level on WDOUT pin goes below the lower Schmitt trigger threshold. After this delay, the device will stop forcing the WDOUT output low.

The WATCHDOG service window will restart when the WDOUT pin goes high. It is recommended that the user tie the WDOUT pin back to V_{CC} through a resistor in order to pull WDOUT high.

A WATCHDOG service while the WDOUT signal is active will be ignored. The state of the WDOUT pin is not guaranteed on reset, but if it powers up low then the WATCHDOG will time out and WDOUT will enter high impedance state.

The Clock Monitor forces the G1 pin low upon detecting a clock frequency error. The Clock Monitor error will continue until the clock frequency has reached the minimum specified value, after which the G1 output will enter the high impedance TRI-STATE mode following $16 t_c$ – $32 t_c$ clock cycles. The Clock Monitor generates a continual Clock Monitor error if the oscillator fails to start, or fails to reach the minimum specified frequency. The specification for the Clock Monitor is as follows:

$1/t_c > 10$ kHz—No clock rejection.

$1/t_c < 10$ Hz—Guaranteed clock rejection.

WATCHDOG AND CLOCK MONITOR SUMMARY

The following salient points regarding the WATCHDOG and CLOCK MONITOR should be noted:

- Both the WATCHDOG and Clock Monitor detector circuits are inhibited during RESET.
 - Following RESET, the WATCHDOG and CLOCK MONITOR are both enabled, with the WATCHDOG having the maximum service window selected.
 - The WATCHDOG service window and Clock Monitor enable/disable option can only be changed once, during the initial WATCHDOG service following RESET.
 - The initial WATCHDOG service must match the key data value in the WATCHDOG Service register WDSVR in order to avoid a WATCHDOG error.
 - Subsequent WATCHDOG services must match all three data fields in WDSVR in order to avoid WATCHDOG errors.
 - The correct key data value cannot be read from the WATCHDOG Service register WDSVR. Any attempt to read this key data value of 01100 from WDSVR will read as key data value of all 0's.
 - The WATCHDOG detector circuit is inhibited during both the HALT and IDLE modes.
 - The Clock Monitor detector circuit is active during both the HALT and IDLE modes. Consequently, the device inadvertently entering the HALT mode will be detected as a Clock Monitor error (provided that the Clock Monitor enable option has been selected by the program).
- With the single-pin R/C oscillator mask option selected and the CLKDLY bit reset, the WATCHDOG service window will resume following HALT mode from where it left off before entering the HALT mode.
 - With the crystal oscillator mask option selected, or with the single-pin R/C oscillator mask option selected and the CLKDLY bit set, the WATCHDOG service window will be set to its selected value from WDSVR following HALT. Consequently, the WATCHDOG should not be serviced for at least 2048 instruction cycles following HALT, but must be serviced within the selected window to avoid a WATCHDOG error.
 - The IDLE timer T0 is not initialized with RESET.
 - The user can sync in to the IDLE counter cycle with an IDLE counter (T0) interrupt or by monitoring the T0PND flag. The T0PND flag is set whenever the thirteenth bit of the IDLE counter toggles (every 4096 instruction cycles). The user is responsible for resetting the T0PND flag.
 - A hardware WATCHDOG service occurs just as the device exits the IDLE mode. Consequently, the WATCHDOG should not be serviced for at least 2048 instruction cycles following IDLE, but must be serviced within the selected window to avoid a WATCHDOG error.
 - Following RESET, the initial WATCHDOG service (where the service window and the CLOCK MONITOR enable/disable must be selected) may be programmed anywhere within the maximum service window (65,536 instruction cycles) initialized by RESET. Note that this initial WATCHDOG service may be programmed within the initial 2048 instruction cycles without causing a WATCHDOG error.

Detection of Illegal Conditions

The device can detect various illegal conditions resulting from coding errors, transient noise, power supply voltage drops, runaway programs, etc.

Reading of undefined ROM gets zeros. The opcode for software interrupt is zero. If the program fetches instructions from undefined ROM, this will force a software interrupt, thus signaling that an illegal condition has occurred.

The subroutine stack grows down for each call (jump to subroutine), interrupt, or PUSH, and grows up for each return or POP, the stack pointer is initialized to RAM location 06F Hex during reset. Consequently, if there are more returns than calls, the stack pointer will point to addresses 070 and 071 Hex (which are undefined RAM). Undefined RAM from addresses 070 to 07F Hex is read as all 1's, which in turn will cause the program to return to address 7FFF Hex. This is an undefined ROM location and the instruction fetched (all 0's) from this location will generate a software interrupt signaling an illegal condition.

Detection of Illegal Conditions

(Continued)

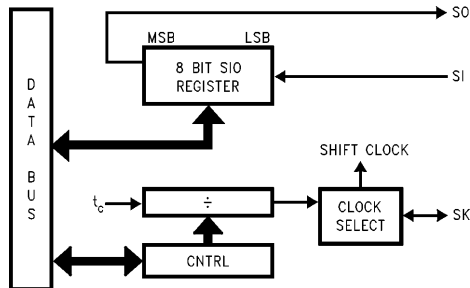
Thus, the chip can detect the following illegal conditions:

1. Executing from undefined ROM.
2. Over "POP"ing the stack by having more returns than calls.

When the software interrupt occurs, the user can re-initialize the stack pointer and do a recovery procedure before re-starting (this recovery program is probably similar to that following reset, but might not contain the same program initialization procedures).

MICROWIRE/PLUS

MICROWIRE/PLUS is a serial synchronous communications interface. The MICROWIRE/PLUS capability enables the device to interface with any of National Semiconductor's MICROWIRE peripherals (i.e. A/D converters, display drivers, E2PROMs etc.) and with other microcontrollers which support the MICROWIRE interface. It consists of an 8-bit serial shift register (SIO) with serial data input (SI), serial data output (SO) and serial shift clock (SK). *Figure 12* shows a block diagram of the MICROWIRE/PLUS logic.



TL/DD/12523-15

FIGURE 12. MICROWIRE/PLUS Block Diagram

The shift clock can be selected from either an internal source or an external source. Operating the MICROWIRE/PLUS arrangement with the internal clock source is called the Master mode of operation. Similarly, operating the MICROWIRE/PLUS arrangement with an external shift clock is called the Slave mode of operation.

The CNTRL register is used to configure and control the MICROWIRE/PLUS mode. To use the MICROWIRE/PLUS, the MSEL bit in the CNTRL register is set to one. In the master mode the SK clock rate is selected by the two bits, SL0 and SL1, in the CNTRL register. Table VI details the different clock rates that may be selected.

TABLE VI. MICROWIRE/PLUS Master Mode Clock Selection

| SL1 | SL0 | SK |
|-----|-----|----------------|
| 0 | 0 | $2 \times t_c$ |
| 0 | 1 | $4 \times t_c$ |
| 1 | x | $8 \times t_c$ |

Where t_c is the instruction cycle clock

Thus, the chip can detect the following illegal conditions:

1. Executing from undefined ROM.
2. Over "POP"ing the stack by having more returns than calls.

MICROWIRE/PLUS OPERATION

Setting the BUSY bit in the PSW register causes the MICROWIRE/PLUS to start shifting the data. It gets reset when eight data bits have been shifted. The user may reset the BUSY bit by software to allow less than 8 bits to shift. If enabled, an interrupt is generated when eight data bits have been shifted. The device may enter the MICROWIRE/PLUS mode either as a Master or as a Slave. *Figure 13* shows how two COP888 microcontrollers and several peripherals may be interconnected using the MICROWIRE/PLUS arrangements.

Warning

The SIO register should only be loaded when the SK clock is low. Loading the SIO register while the SK clock is high will result in undefined data in the SIO register. SK clock is normally low when not shifting.

Setting the BUSY flag when the input SK clock is high in the MICROWIRE/PLUS slave mode may cause the current SK clock for the SIO shift register to be narrow. For safety, the BUSY flag should only be set when the input SK clock is low.

MICROWIRE/PLUS Master Mode Operation

In the MICROWIRE/PLUS Master mode of operation the shift clock (SK) is generated internally. The MICROWIRE Master always initiates all data exchanges. The MSEL bit in the CNTRL register must be set to enable the SO and SK functions onto the G Port. The SO and SK pins must also be selected as outputs by setting appropriate bits in the Port G configuration register. Table VI summarizes the bit settings required for Master mode of operation.

MICROWIRE/PLUS Slave Mode Operation

In the MICROWIRE/PLUS Slave mode of operation the SK clock is generated by an external source. Setting the MSEL bit in the CNTRL register enables the SO and SK functions onto the G Port. The SK pin must be selected as an input and the SO pin is selected as an output pin by setting and resetting the appropriate bit in the Port G configuration register. Table VII summarizes the settings required to enter the Slave mode of operation.

MICROWIRE/PLUS (Continued)

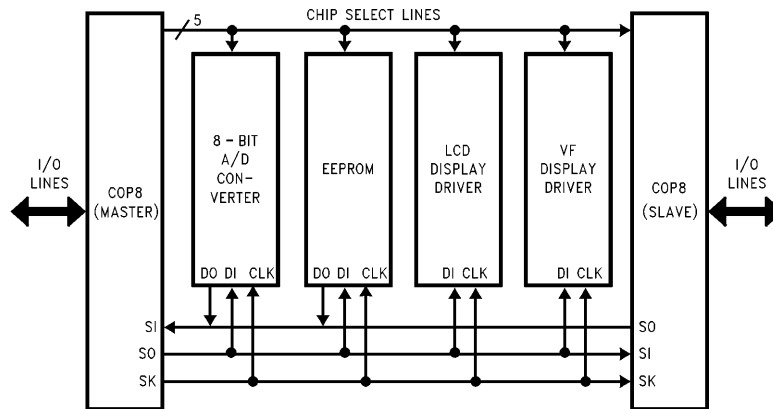


FIGURE 13. MICROWIRE/PLUS Application

TL/DD/12523-16

TABLE VII. MICROWIRE/PLUS Mode Selection

| G4 (SO) Config. Bit | G5 (SK) Config. Bit | G4 Fun. | G5 Fun. | Operation |
|------------------------|------------------------|------------|------------|-----------------------|
| 1 | 1 | SO | Int. SK | MICROWIRE/PLUS Master |
| 0 | 1 | TRI-STATE | Int. SK | MICROWIRE/PLUS Master |
| 1 | 0 | SO | Ext. SK | MICROWIRE/PLUS Slave |
| 0 | 0 | TRI-STATE | Ext. Sk | MICROWIRE/PLUS Slave |

This table assumes that the control flag MSEL is set.

The user must set the BUSY flag immediately upon entering the Slave mode. This will ensure that all data bits sent by the Master will be shifted properly. After eight clock pulses the BUSY flag will be cleared and the sequence may be repeated.

Alternate SK Phase Operation

The device allows either the normal SK clock or an alternate phase SK clock to shift data in and out of the SIO register. In both the modes the SK is normally low. In the normal mode data is shifted in on the rising edge of the SK clock and the data is shifted out on the falling edge of the SK

clock. The SIO register is shifted on each falling edge of the SK clock in the normal mode. In the alternate SK phase mode the SIO register is shifted on the rising edge of the SK clock.

A control flag, SKSEL, allows either the normal SK clock or the alternate SK clock to be selected. Resetting SKSEL causes the MICROWIRE/PLUS logic to be clocked from the normal SK signal. Setting the SKSEL flag selects the alternate SK clock. The SKSEL is mapped into the G6 configuration bit. The SKSEL flag will power up in the reset condition, selecting the normal SK signal.

Memory Map

All RAM, ports and registers (except A and PC) are mapped into data memory address space.

| Address | Contents |
|----------|--------------------------------------------|
| 00 to 6F | On-Chip RAM bytes |
| 70 to BF | Unused RAM Address Space |
| C0 | Timer T2 Lower Byte |
| C1 | Timer T2 Upper Byte |
| C2 | Timer T2 Autoload Register T2RA Lower Byte |
| C3 | Timer T2 Autoload Register T2RA Upper Byte |
| C4 | Timer T2 Autoload Register T2RB Lower Byte |
| C5 | Timer T2 Autoload Register T2RB Upper Byte |
| C6 | Timer T2 Control Register |
| C7 | WATCHDOG Service Register (Reg:WDSVR) |
| C8 | MIWU Edge Select Register (Reg:WKEDG) |
| C9 | MIWU Enable Register (Reg:WKEN) |
| CA | MIWU Pending Register (Reg:WKPND) |
| CB | A/D Converter Control Register (Reg:ENAD) |
| CC | A/D Converter Result Register (Reg:ADRSLT) |
| CD to CF | Reserved |
| D0 | Port L Data Register |
| D1 | Port L Configuration Register |
| D2 | Port L Input Pins (Read Only) |
| D3 | Reserved for Port L |
| D4 | Port G Data Register |
| D5 | Port G Configuration Register |
| D6 | Port G Input Pins (Read Only) |
| D7 | Port I Input Pins (Read Only) |
| D8 | Port C Data Register |
| D9 | Port C Configuration Register |
| DA | Port C Input Pins (Read Only) |
| DB | Reserved for Port C |
| DC | Port D Data Register |
| DD to DF | Reserved for Port D |
| E0 to E5 | Reserved |
| E6 | Timer T1 Autoload Register T1RB Lower Byte |
| E7 | Timer T1 Autoload Register T1RB Upper Byte |
| E8 | ICNTRL Register |
| E9 | MICROWIRE Shift Register |
| EA | Timer T1 Lower Byte |
| EB | Timer T1 Upper Byte |
| EC | Timer T1 Autoload Register T1RA Lower Byte |
| ED | Timer T1 Autoload Register T1RA Upper Byte |
| EE | CNTRL Control Register |
| EF | PSW Register |

Note: Reading memory locations 70–7F Hex will return all ones. Reading other unused memory locations will return undefined data.

| Address | Contents |
|----------|---------------------------------|
| F0 to FB | On-Chip RAM Mapped as Registers |
| FC | X Register |
| FD | SP Register |
| FE | B Register |
| FF | Reserved |

Note: Reading memory locations 70–7F Hex will return all ones. Reading other unused memory locations will return undefined data.

Addressing Modes

There are ten addressing modes, six for operand addressing and four for transfer of control.

OPERAND ADDRESSING MODES

Register Indirect

This is the “normal” addressing mode. The operand is the data memory addressed by the B pointer or X pointer.

Register Indirect (with auto post increment or decrement of pointer)

This addressing mode is used with the LD and X instructions. The operand is the data memory addressed by the B pointer or X pointer. This is a register indirect mode that automatically post increments or decrements the B or X register after executing the instruction.

Direct

The instruction contains an 8-bit address field that directly points to the data memory for the operand.

Immediate

The instruction contains an 8-bit immediate field as the operand.

Short Immediate

This addressing mode is used with the Load B Immediate instruction. The instruction contains a 4-bit immediate field as the operand.

Indirect

This addressing mode is used with the LAID instruction. The contents of the accumulator are used as a partial address (lower 8 bits of PC) for accessing a data operand from the program memory.

TRANSFER OF CONTROL ADDRESSING MODES

Relative

This mode is used for the JP instruction, with the instruction field being added to the program counter to get the new program location. JP has a range from –31 to +32 to allow a 1-byte relative jump (JP + 1 is implemented by a NOP instruction). There are no “pages” when using JP, since all 15 bits of PC are used.

Absolute

This mode is used with the JMP and JSR instructions, with the instruction field of 12 bits replacing the lower 12 bits of the program counter (PC). This allows jumping to any location in the current 4k program memory segment.

Addressing Modes (Continued)

Absolute Long

This mode is used with the JMPL and JSRL instructions, with the instruction field of 15 bits replacing the entire 15 bits of the program counter (PC). This allows jumping to any location up to 32k in the program memory space.

Indirect

This mode is used with the JID instruction. The contents of the accumulator are used as a partial address (lower 8 bits of PC) for accessing a location in the program memory. The contents of this program memory location serve as a partial address (lower 8 bits of PC) for the jump to the next instruction.

Note: The VIS is a special case of the Indirect Transfer of Control addressing mode, where the double byte vector associated with the interrupt is transferred from adjacent addresses in the program memory into the program counter (PC) in order to jump to the associated interrupt service routine.

Instruction Set

REGISTER AND SYMBOL DEFINITION

Registers

| | |
|-----|---------------------------------------------------|
| A | 8-Bit Accumulator Register |
| B | 8-Bit Address Register |
| X | 8-Bit Address Register |
| SP | 8-Bit Stack Pointer Register |
| PC | 15-Bit Program Counter Register |
| PU | Upper 7 Bits of PC |
| PL | Lower 8 Bits of PC |
| C | 1 Bit of PSW Register for Carry |
| HC | 1 Bit of PSW Register for Half Carry |
| GIE | 1 Bit of PSW Register for Global Interrupt Enable |
| VU | Interrupt Vector Upper Byte |
| VL | Interrupt Vector Lower Byte |

Symbols

| | |
|------|------------------------------------------------------------|
| [B] | Memory Indirectly Addressed by B Register |
| [X] | Memory Indirectly Addressed by X Register |
| MD | Direct Addressed Memory |
| Mem | Direct Addressed Memory or [B] |
| Meml | Direct Addressed Memory or [B] or Immediate Data |
| Imm | 8-Bit Immediate Data |
| Reg | Register Memory: Addresses F0 to FF (Includes B, X and SP) |
| Bit | Bit Number (0 to 7) |
| ← | Loaded with |
| ↔ | Exchanged with |

Instruction Set (Continued)

INSTRUCTION SET

| | | | |
|-------|---------|-------------------------------------|----------------------------------------------------------------------------|
| ADD | A,MemI | ADD | $A \leftarrow A + MemI$ |
| ADC | A,MemI | ADD with Carry | $A \leftarrow A + MemI + C, C \leftarrow Carry, HC \leftarrow Half\ Carry$ |
| SUBC | A,MemI | Subtract with Carry | $A \leftarrow A - MemI + C, C \leftarrow Carry, HC \leftarrow Half\ Carry$ |
| AND | A,MemI | Logical AND | $A \leftarrow A \text{ and } MemI$ |
| ANDSZ | A,Imm | Logical AND Immed., Skip if Zero | Skip next if $(A \text{ and } Imm) = 0$ |
| OR | A,MemI | Logical OR | $A \leftarrow A \text{ or } MemI$ |
| XOR | A,MemI | Logical EXclusive OR | $A \leftarrow A \text{ xor } MemI$ |
| IFEQ | MD,Imm | IF Equal | Compare MD and Imm, Do next if $MD = Imm$ |
| IFEQ | A,MemI | IF Equal | Compare A and MemI, Do next if $A = MemI$ |
| IFNE | A,MemI | IF Not Equal | Compare A and MemI, Do next if $A \neq MemI$ |
| IFGT | A,MemI | IF Greater Than | Compare A and MemI, Do next if $A > MemI$ |
| IFBNE | # | IF B Not Equal | Do next if lower 4 bits of $B \neq Imm$ |
| DRSZ | Reg | Decrement Reg., Skip if Zero | $Reg \leftarrow Reg - 1, \text{ Skip if } Reg = 0$ |
| SBIT | #,Mem | Set BIT | 1 to bit, Mem (bit = 0 to 7 immediate) |
| RBIT | #,Mem | Reset BIT | 0 to bit, Mem |
| IFBIT | #,Mem | IF BIT | IF bit in A or Mem is two do next instruction |
| RPND | | Reset PeNDing Flag | Reset Software Interrupt Pending Flag |
| X | A,Mem | EXchange A with Memory | $A \leftrightarrow Mem$ |
| X | A,[X] | EXchange A with Memory [X] | $A \leftrightarrow [X]$ |
| LD | A,MemI | LoaD A with Memory | $A \leftarrow MemI$ |
| LD | A,[X] | LoaD A with Memory [X] | $A \leftarrow [X]$ |
| LD | B,Imm | LoaD B with Immed. | $B \leftarrow Imm$ |
| LD | Mem,Imm | LoaD Memory Immed. | $Mem \leftarrow Imm$ |
| LD | Reg,Imm | LoaD Register Memory Immed. | $Reg \leftarrow Imm$ |
| X | A,[B] | EXchange A with Memory [B] | $A \leftrightarrow [B], (B \leftarrow B - 1)$ |
| X | A,[X] | EXchange A with Memory [X] | $A \leftrightarrow [X], (X \leftarrow X - 1)$ |
| LD | A,[B] | LoaD A with Memory [B] | $A \leftarrow [B], (B \leftarrow B - 1)$ |
| LD | A,[X] | LoaD A with Memory [X] | $A \leftarrow [X], (X \leftarrow X - 1)$ |
| LD | [B],Imm | LoaD Memory [B] Immed | $[B] \leftarrow Imm, (B \leftarrow B - 1)$ |
| CLR | A | CLeaR A | $A \leftarrow 0$ |
| INC | A | INCRement A | $A \leftarrow A + 1$ |
| DEC | A | DECRement A | $A \leftarrow A - 1$ |
| LAID | | Load A InDirect from ROM | $A \leftarrow ROM(PU,A)$ |
| DCOR | A | Decimal CORrect A | $A \leftarrow BCD\ correction\ of\ A\ (follows\ ADC,\ SUBC)$ |
| RRC | A | Rotate A Right thru C | $C \rightarrow A7 \rightarrow \dots \rightarrow A0 \rightarrow C$ |
| RLC | A | Rotate A Left thru C | $C \leftarrow A7 \leftarrow \dots \leftarrow A0 \leftarrow C$ |
| SWAP | A | SWAP nibbles of A | $A7 \dots A4 \leftrightarrow A3 \dots A0$ |
| SC | | Set C | $C \leftarrow 1, HC \leftarrow 1$ |
| RC | | Reset C | $C \leftarrow 0, HC \leftarrow 0$ |
| IF C | | IF C | IF C is true, do next instruction |
| IFNC | | IF Not C | IF C is not true, do next instruction |
| POP | A | POP the stack into A | $SP \leftarrow SP + 1, A \leftarrow [SP]$ |
| PUSH | A | PUSH A onto the stack | $[SP] \leftarrow A, SP \leftarrow SP - 1$ |
| VIS | | Vector to Interrupt Service Routine | $PU \leftarrow [VU], PL \leftarrow [VL]$ |
| JMPL | Addr. | Jump absolute Long | $PC \leftarrow ii\ (ii = 15\ bits, 0\ to\ 32k)$ |
| JMP | Addr. | Jump absolute | $PC9 \dots 0 \leftarrow i\ (i = 12\ bits)$ |
| JP | Disp. | Jump relative short | $PC \leftarrow PC + r\ (r\ is\ -31\ to\ +32, \text{ except } 1)$ |
| JSRL | Addr. | Jump SubRoutine Long | $[SP] \leftarrow PL, [SP-1] \leftarrow PU, SP-2, PC \leftarrow ii$ |
| JSR | Addr | Jump SubRoutine | $[SP] \leftarrow PL, [SP-1] \leftarrow PU, SP-2, PC9 \dots 0 \leftarrow i$ |
| JID | | Jump InDirect | $PL \leftarrow ROM(PU,A)$ |
| RET | | RETurn from subroutine | $SP + 2, PL \leftarrow [SP], PU \leftarrow [SP-1]$ |
| RETSK | | RETurn and SKip | $SP + 2, PL \leftarrow [SP], PU \leftarrow [SP-1]$ |
| RETI | | RETurn from Interrupt | $SP + 2, PL \leftarrow [SP], PU \leftarrow [SP-1], GIE \leftarrow 1$ |
| INTR | | Generate an Interrupt | $[SP] \leftarrow PL, [SP-1] \leftarrow PU, SP-2, PC \leftarrow 0FF$ |
| NOP | | No OPeration | $PC \leftarrow PC + 1$ |

Instruction Execution Time

Most instructions are single byte (with immediate addressing mode instructions taking two bytes).

Most single byte instructions take one cycle time to execute.

Skipped instructions require x number of cycles to be skipped, where x equals the number of bytes in the skipped instruction opcode.

See the BYTES and CYCLES per INSTRUCTION table for details.

Bytes and Cycles per Instruction

The following table shows the number of bytes and cycles for each instruction in the format of byte/cycle.

Logic and Arithmetic Instructions

| Instr. | [B] | Direct | Immed. |
|--------|-----|--------|--------|
| ADD | 1/1 | 3/4 | 2/2 |
| ADC | 1/1 | 3/4 | 2/2 |
| SUBC | 1/1 | 3/4 | 2/2 |
| AND | 1/1 | 3/4 | 2/2 |
| OR | 1/1 | 3/4 | 2/2 |
| XOR | 1/1 | 3/4 | 2/2 |
| IFEQ | 1/1 | 3/4 | 2/2 |
| IFGT | 1/1 | 3/4 | 2/2 |
| IFBNE | 1/1 | | |
| DRSZ | | 1/3 | |
| SBIT | 1/1 | 3/4 | |
| RBIT | 1/1 | 3/4 | |
| IFBIT | 1/1 | 3/4 | |

| | |
|------|-----|
| RPND | 1/1 |
|------|-----|

Instructions Using A and C

| | |
|-------|-----|
| CLRA | 1/1 |
| INCA | 1/1 |
| DECA | 1/1 |
| LAID | 1/3 |
| DCORA | 1/1 |
| RRCA | 1/1 |
| RLCA | 1/1 |
| SWAPA | 1/1 |
| SC | 1/1 |
| RC | 1/1 |
| IFC | 1/1 |
| IFNC | 1/1 |
| PUSHA | 1/3 |
| POPA | 1/3 |
| ANDSZ | 2/2 |

Transfer of Control Instructions

| | |
|-------|-----|
| JMPL | 3/4 |
| JMP | 2/3 |
| JP | 1/3 |
| JSRL | 3/5 |
| JSR | 2/5 |
| JID | 1/3 |
| VIS | 1/5 |
| RET | 1/5 |
| RETSK | 1/5 |
| RETI | 1/5 |
| INTR | 1/7 |
| NOP | 1/1 |

Memory Transfer Instructions

| | Register Indirect | | Direct | Immed. | Register Indirect Auto Incr & Decr | |
|-------------|-------------------|-----|--------|--------|------------------------------------|----------|
| | [B] | [X] | | | [B+, B-] | [X+, X-] |
| X A,* | 1/1 | 1/3 | 2/3 | | 1/2 | 1/3 |
| LD A,* | 1/1 | 1/3 | 2/3 | 2/2 | 1/2 | 1/3 |
| LD B,Imm | | | | 1/1 | | |
| LD B,Imm | | | | 2/3 | | |
| LD Mem,Imm | | 2/2 | 3/3 | | 2/2 | |
| LD Reg,Imm | | | 2/3 | | | |
| IFEQ MD,Imm | | | 3/3 | | | |

(If B < 16)

(If B > 15)

* ≥ Memory location addressed by B or X or directly

COP8788CF/COP8784CF Opcode Table

| Upper Nibble | | | | | | | | | | | | | | | | |
|--------------|-------|------------|----------|-------------|-------------|-------------|-------------|--------------|-------------|-----------|----------|---------------|---------------|-------|---------|--|
| F | E | D | C | B | A | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| JP-15 | JP-31 | LD 0F0, #i | DRSZ 0F0 | RRCA | RC | ADC A, #i | ADC A, [B] | IFBIT 0, [B] | ANDSZ A, #i | LD B, #0F | IFBNE 0 | JSR x000-x0FF | JMP x000-x0FF | JP+17 | INTR 0 | |
| JP-14 | JP-30 | LD 0F1, #i | DRSZ 0F1 | * | SC | SUBC A, #i | SUBCA, [B] | IFBIT 1, [B] | * | LD B, #0E | IFBNE 1 | JSR x100-x1FF | JMP x100-x1FF | JP+18 | JP+2 1 | |
| JP-13 | JP-29 | LD 0F2, #i | DRSZ 0F2 | X A, [X+] | X A, [B+] | IFEQ A, #i | IFEQ A, [B] | IFBIT 2, [B] | * | LD B, #0D | IFBNE 2 | JSR x200-x2FF | JMP x200-x2FF | JP+19 | JP+3 2 | |
| JP-12 | JP-28 | LD 0F3, #i | DRSZ 0F3 | X A, [X-] | X A, [B-] | IFGT A, #i | IFGT A, [B] | IFBIT 3, [B] | * | LD B, #0C | IFBNE 3 | JSR x300-x3FF | JMP x300-x3FF | JP+20 | JP+4 3 | |
| JP-11 | JP-27 | LD 0F4, #i | DRSZ 0F4 | VIS | LAID | ADD A, #i | ADD A, [B] | IFBIT 4, [B] | CLRA | LD B, #0B | IFBNE 4 | JSR x400-x4FF | JMP x400-x4FF | JP+21 | JP+5 4 | |
| JP-10 | JR-26 | LD 0F5, #i | DRSZ 0F5 | RPND | JID | AND A, #i | AND A, [B] | IFBIT 5, [B] | SWAPA | LD B, #0A | IFBNE 5 | JSR x500-x5FF | JMR x500-x5FF | JR+22 | JP+6 5 | |
| JP-9 | JP-25 | LD 0F6, #i | DRSZ 0F6 | X A, [X] | X A, [B] | XOR A, #i | XOR A, [B] | IFBIT 6, [B] | DCORA | LD B, #09 | IFBNE 6 | JSR x600-x6FF | JMP x600-x6FF | JP+23 | JP+7 6 | |
| JP-8 | JP-24 | LD 0F7, #i | DRSZ 0F7 | * | * | OR A, #i | OR A, [B] | IFBIT 7, [B] | PUSHA | LD B, #08 | IFBNE 7 | JSR x700-x7FF | JMR x700-x7FF | JR+24 | JP+8 7 | |
| JP-7 | JP-23 | LD 0F8, #i | DRSZ 0F8 | NOP | RLCA | LD A, #i | IFC | SBIT 0, [B] | RBIT 0, [B] | LD B, #07 | IFBNE 8 | JSR x800-x8FF | JMP x800-x8FF | JP+25 | JP+9 8 | |
| JP-6 | JP-22 | LD 0F9, #i | DRSZ 0F9 | IFNE A, [B] | IFEQ Md, #i | IFNE A, #i | IFNC | SBIT 1, [B] | RBIT 1, [B] | LD B, #06 | IFBNE 9 | JSR x900-x9FF | JMP x900-x9FF | JP+26 | JP+10 9 | |
| JP-5 | JP-21 | LD 0FA, #i | DRSZ 0FA | LD A, [X+] | LD B, [B+] | LD [B+], #i | INCA | SBIT 2, [B] | RBIT 2, [B] | LD B, #05 | IFBNE 0A | JSR xA00-xAFF | JMP xA00-xAFF | JP+27 | JP+11 A | |
| JP-4 | JP-20 | LD 0FB, #i | DRSZ 0FB | LD A, [X-] | LD A, [B-] | LD [B-], #i | DECA | SBIT 3, [B] | RBIT 3, [B] | LD B, #04 | IFBNE 0B | JSR xB00-xBFF | JMP xB00-xBFF | JP+28 | JP+12 B | |
| JP-3 | JP-19 | LD 0FC, #i | DRSZ 0FC | LD Md, #i | JMPL | X A, Md | POPA | SBIT 4, [B] | RBIT 4, [B] | LD B, #03 | IFBNE 0C | JSR xC00-xCFF | JMP xC00-xCFF | JP+29 | JP+13 C | |
| JP-2 | JP-18 | LD 0FD, #i | DRSZ 0FD | DIR | JSRL | LD A, Md | RETSK | SBIT 5, [B] | RBIT 5, [B] | LD B, #02 | IFBNE 0D | JSR xD00-xDFF | JMP xD00-xDFF | JP+30 | JP+14 D | |
| JP-1 | JP-17 | LD 0FE, #i | DRSZ 0FE | LD A, [X] | LD A, [B] | LD [B], #i | RET | SBIT 6, [B] | RBIT 6, [B] | LD B, #01 | IFBNE 0E | JSR xE00-xEFF | JMP xE00-xEFF | JP+31 | JP+15 E | |
| JP-0 | JP-16 | LD 0FF, #i | DRSZ 0FF | * | * | LD B, #i | RETI | SBIT 7, [B] | RBIT 7, [B] | LD B, #00 | IFBNE 0F | JSR xF00-xFFF | JMP xF00-xFFF | JP+32 | JP+16 F | |

where, i is the immediate data

Md is a directly addressed memory location

* is an unused opcode

The opcode 60 Hex is also the opcode for IFBIT #i.A

Development Support

SUMMARY

- iceMASTER™: IM-COP8/400—Full feature in-circuit emulation for all COP8 products. A full set of COP8 Basic and Feature Family device and package specific probes are available.
- COP8 Debug Module: Moderate cost in-circuit emulation and development programming unit.
- COP8 Evaluation and Programming Unit: EPU-COP888GG—low cost In-circuit simulation and development programming unit.
- Assembler: COP8-DEV-IBMA. A DOS installable cross development Assembler, Linker, Librarian and Utility Software Development Tool Kit.
- C Compiler: COP8C. A DOS installable cross development Software Tool Kit.
- OTP/EPROM Programmer Support: Covering needs from engineering prototype, pilot production to full production environments.

iceMASTER (IM) IN-CIRCUIT EMULATION

The iceMASTER IM-COP8/400 is a full feature, PC based, in-circuit emulation tool developed and marketed by Meta-Link Corporation to support the whole COP8 family of products. National is a resale vendor for these products.

See *Figure 14* for configuration.

The iceMASTER IM-COP8/400 with its device specific COP8 Probe provides a rich feature set for developing, testing and maintaining product:

- Real-time in-circuit emulation; full 2.4V–5.5V operation range, full DC-10 MHz clock. Chip options are programmable or jumper selectable.
- Direct connection to application board by package compatible socket or surface mount assembly.
- Full 32 kbytes of loadable programming space that overlays (replaces) the on-chip ROM or EPROM. On-chip RAM and I/O blocks are used directly or recreated on the probe as necessary.
- Full 4k frame synchronous trace memory. Address, instruction, and 8 unspecified, circuit connectable trace lines. Display can be HLL source (e.g., C source), assembly or mixed.
- A full 64k hardware configurable break, trace on, trace off control, and pass count increment events.

- Tool set integrated interactive symbolic debugger—supports both assembler (COFF) and C Compiler (.COD) linked object formats.
- Real time performance profiling analysis; selectable bucket definition.
- Watch windows, content updated automatically at each execution break.
- Instruction by instruction memory/register changes displayed on source window when in single step operation.
- Single base unit and debugger software reconfigurable to support the entire COP8 family; only the probe personality needs to change. Debugger software is processor customized, and reconfigured from a master model file.
- Processor specific symbolic display of registers and bit level assignments, configured from master model file.
- Halt/Idle mode notification.
- On-line HELP customized to specific processor using master model file.
- Includes a copy of COP8-DEV-IBMA assembler and linker SDK.

IM Order Information

| Base Unit | |
|-------------------------|----------------------------------------|
| IM-COP8/400-1 | iceMASTER Base Unit, 110V Power Supply |
| IM-COP8/400-2 | iceMASTER Base Unit, 220V Power Supply |
| iceMASTER Probe | |
| MHW-884CF28DWPC | 28 DIP |
| MHW-888CF40DWPC | 40 DIP |
| MHW-888CF44PWPC | 44 PLCC |
| 28 DIP to 28 SO Adapter | |
| MHW-SOIC28 | 28 SO |

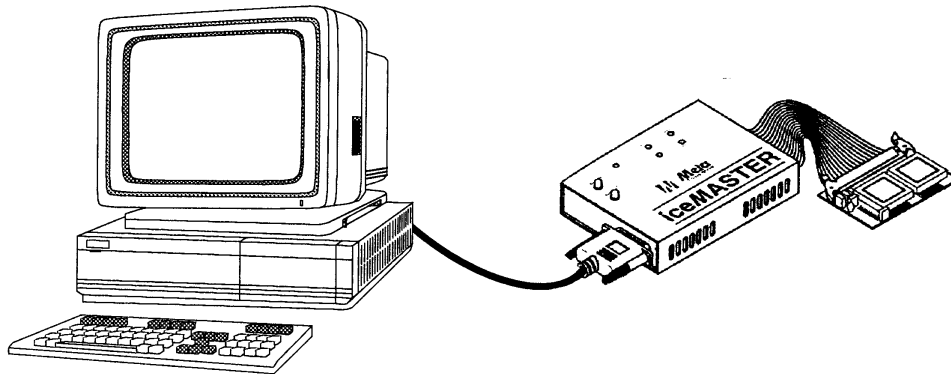


FIGURE 14. COP8 iceMASTER Environment

TL/DD/12523-18

Development Support (Continued)

iceMASTER DEBUG MODULE (DM)

The iceMASTER Debug Module is a PC based, combination in-circuit emulation tool and COP8 based OTP/EPROM programming tool developed and marketed by MetaLink Corporation to support the whole COP8 family of products. National is a resale vendor for these products.

See *Figure 15* for configuration.

The iceMASTER Debug Module is a moderate cost development tool. It has the capability of in-circuit emulation for a specific COP8 microcontroller and in addition serves as a programming tool for COP8 OTP and EPROM product families. Summary of features is as follows:

- Real-time in-circuit emulation; full operating voltage range operation, full DC-10 MHz clock.
- All processor I/O pins can be cabled to an application development board with package compatible cable to socket and surface mount assembly.
- Full 32 kbytes of loadable programming space that overlays (replaces) the on-chip ROM or EPROM. On-chip RAM and I/O blocks are used directly or recreated as necessary.
- 100 frames of synchronous trace memory. The display can be HLL source (C source), assembly or mixed. The most recent history prior to a break is available in the trace memory.
- Configured break points; uses INTR instruction which is modestly intrusive.
- Software—only supported features are selectable.
- Tool set integrated interactive symbolic debugger—supports both assembler (COFF) and C Compiler (.COD) SDK linked object formats.
- Instruction by instruction memory/register changes displayed when in single step operation.
- Debugger software is processor customized, and reconfigured from a master model file.
- Processor specific symbolic display of registers and bit level assignments, configured from master model file.
- Halt/Idle mode notification.
- Programming menu supports full product line of programmable OTP and EPROM COP8 products. Program data is taken directly from the overlay RAM.
- Programming of 44 PLCC and 68 PLCC parts requires external programming adapters.
- Includes wallmount power supply.
- On-board V_{PP} generator from 5V input or connection to external supply supported. Requires V_{PP} level adjustment per the family programming specification (correct level is provided on an on-screen pop-down display).
- On-line HELP customized to specific processor using master model file.
- Includes a copy of COP8-DEV-IBMA assembler and linker SDK.

DM Order Information

| Debug Model Unit | |
|-------------------------|---------|
| COP8-DM/888CF | |
| Cable Adapters | |
| DM-COP8/28D | 28 DIP |
| DM-COP8/40D | 40 DIP |
| DM-COP8/44P | 44 PLCC |
| 28 DIP to 28 SO Adapter | |
| DM-COP8/28D-SO | 28 SO |

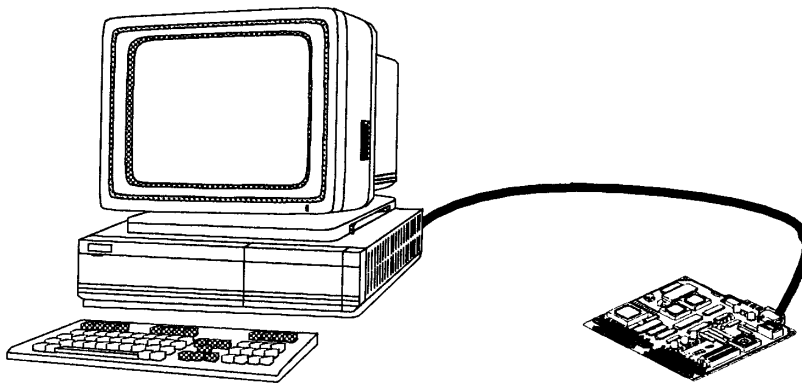


FIGURE 15. COP8-DM Environment

TL/DD/12523-19

Development Support (Continued)

COP8 ASSEMBLER/LINKER SOFTWARE DEVELOPMENT TOOL KIT

National Semiconductor offers a relocateable COP8 macro cross assembler, linker, librarian and utility software development tool kit. Features are summarized as follows:

- Basic and Feature Family instruction set by "device" type.
- Nested macro capability.
- Extensive set of assembler directives.
- Supported on PC/DOS platform.
- Generates National standard COFF output files.
- Integrated Linker and Librarian.
- Integrated utilities to generate ROM code file outputs.
- DUMPCOFF utility.

This product is integrated as a part of MetaLink tools as a development kit, fully supported by the MetaLink debugger. It may be ordered separately or it is bundled with the MetaLink products at no additional cost.

Order Information

| Assembler SDK | |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| COP8-DEV-IBMA | Assembler SDK on installable 3.5" PC/DOS Floppy Disk Drive format. Periodic upgrades and most recent version is available on National's BBS and Internet. |

COP8 C COMPILER

A C Compiler is developed and marketed by Byte Craft Limited. The COP8C compiler is a fully integrated development tool specifically designed to support the compact embedded configuration of the COP8 family of products.

Features are summarized as follows:

- ANSI C with some restrictions and extensions that optimize development for the COP8 embedded application.
- BITS data type extension. Register declaration #pragma with direct bit level definitions.
- C language support for interrupt routines.
- Expert system, rule based code generation and optimization.
- Performs consistency checks against the architectural definitions of the target COP8 device.
- Generates program memory code.
- Supports linking of compiled object or COP8 assembled object formats.
- Global optimization of linked code.
- Symbolic debug load format fully sourced level supported by the MetaLink debugger.

OTP EMULATOR SUPPORT

The COP87L88CF/COP87L84CF devices provide emulation and OTP support for the COP888CF/COP884CF mask programmable devices.

Approved List

| Manufacturer | North America | Europe | Asia |
|-----------------|---------------------------------------------------------|-------------------------------------------|-------------------------------------------|
| BP Microsystems | (800) 225-2102 (713) 688-4600 Fax: (713) 688-0920 | + 49-8152-4183 + 49-8856-932616 | + 852-234-16611 + 852-2710-8121 |
| Data I/O | (800) 426-1045 (206) 881-6444 Fax: (206) 882-1043 | + 44-0734-440011 | Call North America |
| HI-LO | (510) 623-8860 | Call Asia | + 886-2-764-0215 Fax: + 886-2-756-6403 |
| ICE Technology | (800) 624-8949 (919) 430-7915 | + 44-1226-767404 Fax: 0-1226-370-434 | |
| MetaLink | (800) 638-2423 (602) 926-0797 Fax: (602) 693-0681 | + 49-80 9156 96-0 Fax: + 49-80 9123 86 | + 852-737-1800 |
| Systems General | (408) 263-6667 | + 41-1-9450300 | + 886-2-917-3005 Fax: + 886-2-911-1283 |
| Needhams | (916) 924-8037 Fax: (916) 924-8065 | | |

Development Support (Continued)

OTP Emulator Ordering Information

| Device Number | Clock Option | Package | Emulates |
|----------------|-----------------|---------|----------|
| COP87L88CFV-XE | Crystal/Halt En | 44 PLCC | COP888CF |
| COP87L88CFN-XE | Crystal/Halt En | 40 DIP | COP888CF |
| COP87L84CFN-XE | Crystal/Halt En | 28 DIP | COP884CF |
| COP87L84CFM-XE | Crystal/Halt En | 28 SO | COP884CF |

INDUSTRY WIDE OTP/EPROM PROGRAMMING SUPPORT

Programming support, in addition to the MetaLink development tools, is provided by a full range of independent approved vendors to meet the needs from the engineering laboratory to full production.

AVAILABLE LITERATURE

For more information, please see the COP8 Basic Family User's Manual, Literature Number 620895, COP8 Feature Family User's Manual, Literature Number 620897 and National's Family of 8-bit Microcontrollers COP8 Selection Guide, Literature Number 630009.

DIAL-A-HELPER SERVICE

Dial-A-Helper is a service provided by the Microcontroller Applications group. The Dial-A-Helper is an Electronic Information System that may be accessed as a Bulletin Board System (BBS) via data modem, as an FTP site on the Internet via standard FTP client application or as an FTP site on the Internet using a standard Internet browser such as Netscape or Mosaic.

The Dial-A-Helper system provides access to an automated information storage and retrieval system. The system capabilities include a MESSAGE SECTION (electronic mail, when accessed as a BBS) for communications to and from the Microcontroller Applications Group and a FILE SECTION which consists of several file areas where valuable application software and utilities could be found.

DIAL-A-HELPER BBS via a Standard Modem

Modem: CANADA/U.S.: (800) NSC-MICRO
(800) 672-6427
EUROPE: (+49) 0-8141-351332
Baud: 14.4k
Set-Up: Length: 8-Bit
Parity: None
Stop Bit: 1
Operation: 24 Hours, 7 Days

DIAL-A-HELPER via FTP

ftp nscmicro.nsc.com
user: anonymous
password: username@yourhost.site.domain

DIAL-A-HELPER via WorldWide Web Browser

ftp://nscmicro.nsc.com

National Semiconductor on the WorldWide Web

See us on the WorldWide Web at: <http://www.national.com>

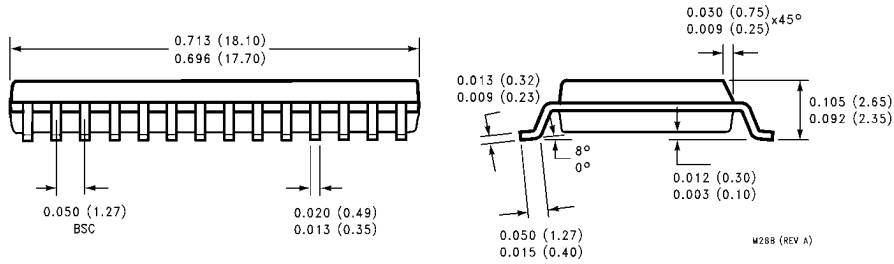
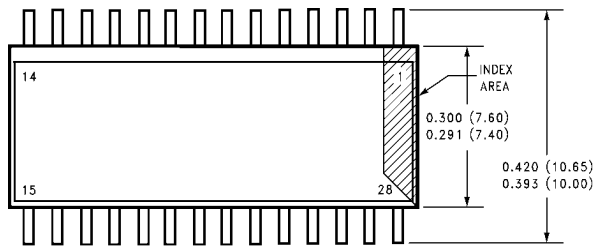
CUSTOMER RESPONSE CENTER

Complete product information and technical support is available from National's customer response centers.

| | | |
|--------------|----------------|------------------------|
| CANADA/U.S.: | Tel: | (800)272-9959 |
| | email: | support@tevm2.nsc.com |
| EUROPE: | email: | europa.support@nsc.com |
| | Deutsch Tel: | + 49 (0) 180-530 85 85 |
| | English Tel: | + 49 (0) 180-532 78 32 |
| | Français Tel: | + 49 (0) 180-532 93 58 |
| | Italiano Tel: | + 49 (0) 180-534 16 80 |
| JAPAN: | Tel: | + 81-043-299-2309 |
| S.E. ASIA: | Beijing Tel: | (+ 86) 10-6856-8601 |
| | Shanghai Tel: | (+ 86) 21-6415-4092 |
| | Hong Kong Tel: | (+ 852) 2737-1600 |
| | Korea Tel: | (+ 82) 2-3771-6909 |
| | Malaysia Tel: | (+ 60-4) 644-9061 |
| | Singapore Tel: | (+ 65) 255-2226 |
| | Taiwan Tel: | + 886-2-521-3288 |
| AUSTRALIA: | Tel: | (+ 61) 3-9558-9999 |
| INDIA: | Tel: | (+ 91) 80-559-9467 |

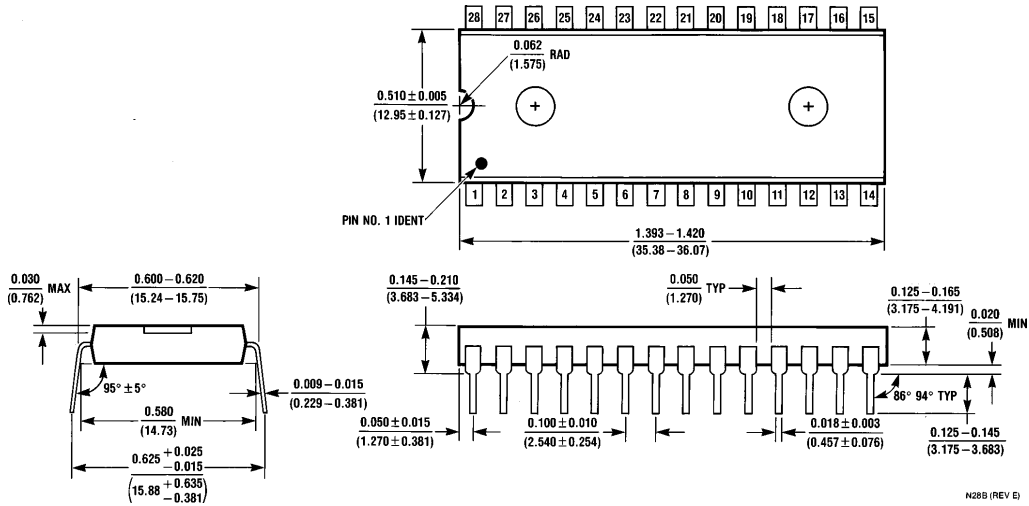


Physical Dimensions inches (millimeters) unless otherwise noted

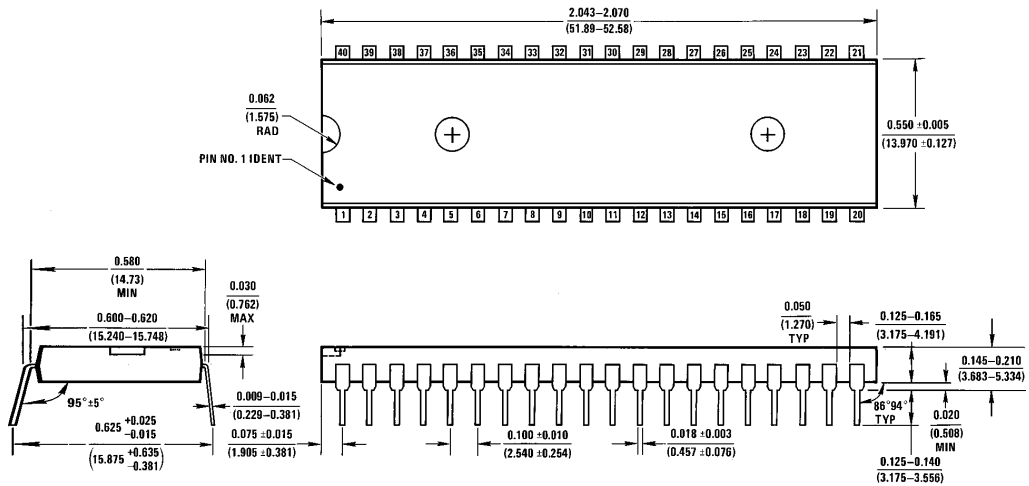


Molded SO Wide Body Package (M)
Order Number COP87L84CFM-XE
NS Package Number M28B

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



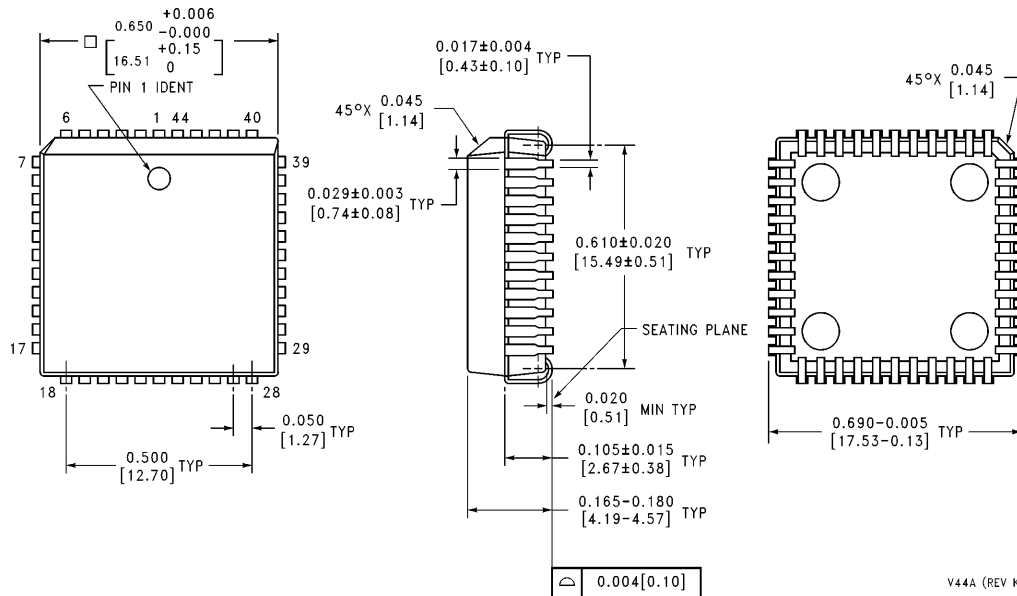
Molded Dual-In-Line Package (N)
Order Number COP87L84CFN-XE
NS Package Number N28B



Molded Dual-In-Line Package (N)
Order Number COP87L84CFN-XE
NS Package Number N40A

COP87L88CF/COP87L84CF 8-Bit One-Time Programmable (OTP) Microcontroller with A/D Converter

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



Plastic Leaded Chip Carrier (V)
Order Number COP87L88CFV-XE
NS Package Number V44A

V44A (REV K)

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.



National Semiconductor Corporation
 1111 West Bardin Road
 Arlington, TX 76017
 Tel: (800) 272-9959
 Fax: (800) 737-7018

<http://www.national.com>

National Semiconductor Europe

Fax: +49 (0) 180-530 85 86
 Email: europe.support@nsc.com
 Deutsch Tel: +49 (0) 180-530 85 85
 English Tel: +49 (0) 180-532 78 32
 Français Tel: +49 (0) 180-532 93 58
 Italiano Tel: +49 (0) 180-534 16 80

National Semiconductor Hong Kong Ltd.

19th Floor, Straight Block,
 Ocean Centre, 5 Canton Rd.
 Tsimshatsui, Kowloon
 Hong Kong
 Tel: (852) 2737-1600
 Fax: (852) 2736-9960

National Semiconductor Japan Ltd.

Tel: 81-043-299-2308
 Fax: 81-043-299-2408

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.