

## COP8780C/COP8781C/COP8782C 8-Bit One-Time Programmable (OTP) Microcontroller

### General Description

The COP8780C, COP8781C and COP8782C are members of the COPS™ 8-bit microcontroller family. They are fully static microcontrollers, fabricated using double-metal, double poly silicon gate microCMOS EPROM technology. These devices are available as UV erasable or One Time Programmable (OTP). These low cost microcontrollers are complete microcomputers containing all system timing, interrupt logic, EPROM, RAM, and I/O necessary to implement dedicated control functions in a variety of applications. Features include an 8-bit memory mapped architecture, MICROWIRE/PLUSTM serial I/O, a 16-bit timer/counter with associated 16-bit autoreload/capture register, and a multi-sourced interrupt. Each I/O pin has software selectable options to adapt the device to the specific application. These devices operate over a voltage range of 4.5V to 6.0V. An efficient, regular instruction set operating at a 1  $\mu$ s instruction cycle rate provides optimal throughput.

The COP8780C, COP8781C and COP8782C can be configured to EMULATE the COP880C, COP840C and COP820C microcontrollers.

### Key Features

- 16-bit multi-function timer supporting
  - PWM mode
  - External event counter mode
  - Input capture mode
- Crystal, RC or External Oscillator, user configurable
- 4 kbytes on-chip OTP EPROM with security feature
- 128 or 64 bytes of on-chip RAM, user configurable

### I/O Features

- Memory-mapped I/O
- Software selectable I/O options (TRI-STATE®, Push-Pull, Weak Pull-Up Input, High Impedance input)

- Schmitt trigger inputs on Port G
- MICROWIRE/PLUS serial I/O
- Packages:
  - 44 PLCC, OTP, Emulates COP880C, 36 I/O pins
  - 40 DIP, OTP, Emulates COP880C, 36 I/O pins
  - 28 DIP, OTP, Emulates COP820C/840C/881C, 24 I/O pins
  - 20 DIP, OTP, Emulates COP822C/842C, 16 I/O pins
  - 28 SO, 20 SO, OTP
  - 44 LDCC, UV Erasable
  - 40 CERDIP, 28 CERDIP, 20 CERDIP, UV Erasable

### CPU/Instruction Set Features

- 1  $\mu$ s instruction cycle time
- Three multi-source interrupts servicing
  - External interrupt with selectable edge
  - Timer interrupt
  - Software 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

- Low current drain (typically < 1  $\mu$ A)
- Extra-low current static HALT mode
- Single supply operation: 4.5V to 6.0V
- Temperature range: -40°C to +85°C

### Development Support

- Emulation device for the COP880C, COP840C, and COP820C
- Real-time emulation and full program debug offered by MetaLink development system

### Block Diagram

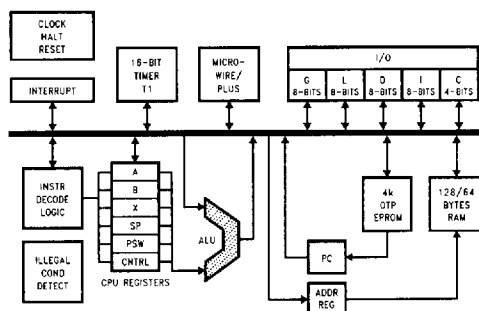
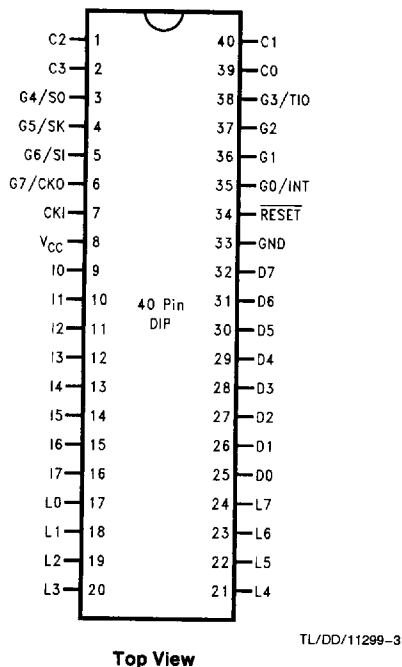


FIGURE 1. Block Diagram

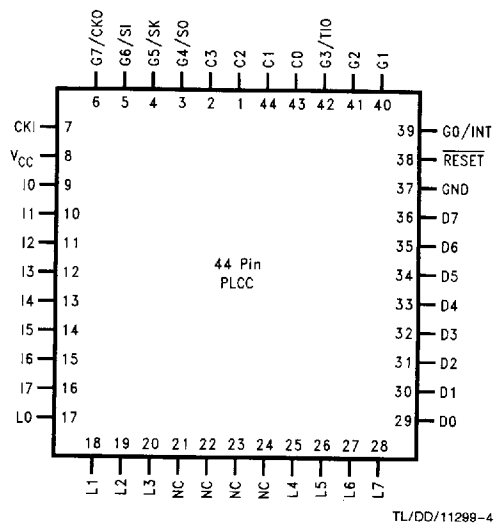
TL/DD/11299-1

# Connection Diagrams



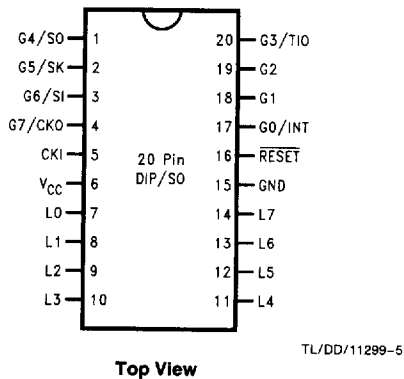
Top View

Order Number COP8780C-XXX/N or COP8780C-XXX/J  
See NS Package Number J40AQ or N40A



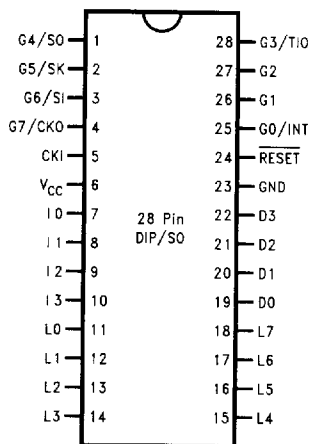
Top View

Order Number COP8780C-XXX/V or COP8780C-XXX/EL  
See NS Package Number EL40C or V44A



Top View

Order Number COP8782C-XXX/J, COP8782C-XXX/N  
or COP8782C-XXX/WM  
See NS Package Number J20AQ, M20B or N20B



Top View

Order Number COP8781C-XXX/J, COP8781C-XXX/N or  
COP8781C-XXX/WM  
See NS Package Number J28AQ, M28B or N28B

FIGURE 3. Connection Diagrams

## COP8780C/COP8781C/COP8782C

## Absolute Maximum Ratings

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

|                                                             |                          |
|-------------------------------------------------------------|--------------------------|
| Supply Voltage ( $V_{CC}$ )                                 | 7V                       |
| Programming Voltage $V_{PP}$ (RESET pin)<br>and ME (pin G6) | 13.4V                    |
| Voltage at any Pin                                          | -0.3V to $V_{CC}$ + 0.3V |

|                                          |                 |
|------------------------------------------|-----------------|
| Total Current into $V_{CC}$ Pin (Source) | 50 mA           |
| Total Current out of GND Pin (Sink)      | 60 mA           |
| Storage Temperature Range                | -65°C to +150°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 COP87XXC; -40°C ≤  $T_A$  ≤ +85°C unless otherwise specified

| Parameter                                                         | Condition                          | Min          | Typ           | Max          | Units   |
|-------------------------------------------------------------------|------------------------------------|--------------|---------------|--------------|---------|
| Operating Voltage                                                 |                                    | 4.5          |               | 6.0          | V       |
| Power Supply Ripple (Note 1)                                      | Peak to Peak                       |              |               | 0.1 $V_{CC}$ | V       |
| Supply Current                                                    |                                    |              |               | 21           | mA      |
| CKI = 10 MHz (Note 2)                                             | $V_{CC} = 6V, t_c = 1 \mu s$       |              |               | 10           | $\mu A$ |
| HALT Current (Note 3)                                             | $V_{CC} = 6V, CKI = 0 \text{ MHz}$ |              |               |              |         |
| Input Levels                                                      |                                    |              |               |              |         |
| RESET, CKI                                                        |                                    | 0.9 $V_{CC}$ |               | 0.1 $V_{CC}$ | V       |
| Logic High                                                        |                                    |              |               |              | V       |
| Logic Low                                                         |                                    |              |               |              | V       |
| All Other Inputs                                                  |                                    | 0.7 $V_{CC}$ |               | 0.2 $V_{CC}$ | V       |
| Logic High                                                        |                                    |              |               |              | V       |
| Logic Low                                                         |                                    |              |               |              | V       |
| Hi-Z Input Leakage                                                | $V_{CC} = 6.0V$                    | -2           |               | +2           | $\mu A$ |
| Input Pullup Current                                              | $V_{CC} = 6.0V, V_{IN} = 0V$       | -40          |               | -250         | $\mu A$ |
| G Port Input Hysteresis                                           | (Note 6)                           |              | 0.05 $V_{CC}$ |              | V       |
| Output Current Levels                                             |                                    |              |               |              |         |
| D Outputs                                                         |                                    |              |               |              | mA      |
| Source                                                            | $V_{CC} = 4.5V, V_{OH} = 3.8V$     | -0.4         |               |              |         |
| Sink                                                              | $V_{CC} = 4.5V, V_{OL} = 1.0V$     | 10           |               |              | mA      |
| All Others                                                        |                                    |              |               |              | $\mu A$ |
| Source (Weak Pull-Up)                                             | $V_{CC} = 4.5V, V_{OH} = 3.2V$     | -10          |               | -110         | $\mu A$ |
| Source (Push-Pull Mode)                                           | $V_{CC} = 4.5V, V_{OH} = 3.8V$     | -0.4         |               |              | mA      |
| Sink (Push-Pull Mode)                                             | $V_{CC} = 4.5V, V_{OL} = 0.4V$     | 1.6          |               |              | mA      |
| TRI-STATE Leakage                                                 |                                    | -2.0         |               | +2.0         | $\mu A$ |
| Allowable Sink/Source<br>Current per Pin                          |                                    |              |               | 15           | mA      |
| D Outputs (Sink)                                                  |                                    |              |               | 3            | mA      |
| All Others                                                        |                                    |              |               |              |         |
| Maximum Input Current (Notes 4, 6)<br>without Latchup (Room Temp) | Room Temp                          |              |               | ±200         | mA      |
| RAM Retention Voltage, $V_r$<br>(Note 5)                          |                                    | 2.0          |               |              | V       |
| Input Capacitance                                                 | (Note 6)                           |              |               | 7            | pF      |
| Load Capacitance on D2                                            | (Note 6)                           |              |               | 1000         | pF      |

Note 1: Rate of voltage change must be less than 0.5V/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. Halt test conditions: All inputs tied to  $V_{CC}$ , L, C, and G port I/O's configured as outputs and programmed low; D outputs programmed low; the window for UV erasable packages is completely covered with an opaque cover to prevent light from falling onto the die during HALT mode test. Parameter refers to HALT mode entered via setting bit 7 of the G Port data register.

Note 4: 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 $\Omega$  (typ). These two pins will not latch up. The voltage at the pins must be limited to less than 14V.

Note 5: To maintain RAM integrity, the voltage must not be dropped or raised instantaneously.

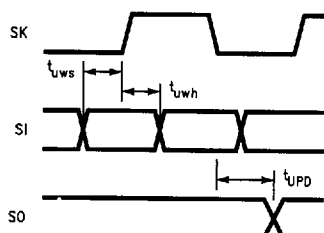
Note 6: Parameter characterized but not tested.

**COP8780C/COP8781C/COP8782C****AC Electrical Characteristics**  $-40^{\circ}\text{C} < T_A < +85^{\circ}\text{C}$  unless otherwise specified

| Parameter                                        | Condition                                       | Min | Typ | Max | Units         |
|--------------------------------------------------|-------------------------------------------------|-----|-----|-----|---------------|
| Instruction Cycle Time ( $t_c$ )                 |                                                 |     |     |     | $\mu\text{s}$ |
| Crystal/Resonator or External Clock              | $V_{CC} \geq 4.5\text{V}$                       | 1   |     | DC  | $\mu\text{s}$ |
| R/C Oscillator Mode                              | $V_{CC} \geq 4.5\text{V}$                       | 3   |     | DC  | $\mu\text{s}$ |
| CKI Clock Duty Cycle (Note 7)                    | $f_r = \text{Max}$                              | 45  |     | 55  | %             |
| Rise Time (Note 7)                               | $f_r = 10\text{ MHz Ext Clock}$                 |     |     | 12  | ns            |
| Fall Time (Note 7)                               | $f_r = 10\text{ MHz Ext Clock}$                 |     |     | 8   | ns            |
| Inputs                                           |                                                 |     |     |     |               |
| $t_{\text{SETUP}}$                               | $V_{CC} \geq 4.5\text{V}$                       | 200 |     |     | ns            |
| $t_{\text{HOLD}}$                                | $V_{CC} \geq 4.5\text{V}$                       | 60  |     |     | ns            |
| Output Propagation Delay                         | $C_L = 100\text{ pF}, R_L = 2.2\text{ k}\Omega$ |     |     |     |               |
| $t_{PD1}, t_{PD0}$                               | $V_{CC} \geq 4.5\text{V}$                       |     |     | 0.7 | $\mu\text{s}$ |
| SO, SK                                           | $V_{CC} \geq 4.5\text{V}$                       |     |     | 1   | $\mu\text{s}$ |
| All Others                                       | $V_{CC} \geq 4.5\text{V}$                       |     |     |     |               |
| 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                                |                                                 | 1.0 |     |     | $\mu\text{s}$ |

**Note 7:** Parameter guaranteed by design, but not tested.

$t_c$  = Instruction Cycle Time.

**Timing Diagram**

**FIGURE 2. MICROWIRE/PLUS Timing**

TL/DD/10802-2

## Pin Descriptions

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

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

RESET is the master reset input. See Reset description.

PORT I is an 8-bit Hi-Z input port. 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 for these unterminated pins will return unpredictable values. The user must ensure that the software takes this into account by either masking or restricting the accesses to bit operations. The unterminated PORT I pins will draw power only when addressed.

PORT L is an 8-bit I/O port.

PORT C is a 4-bit I/O port.

Three memory locations are allocated for the L, G and C ports, one each for data register, configuration register and the input pins. Reading bits 4–7 of the C-Configuration register, data register, and input pins returns undefined data.

There are two registers associated with the L and C ports: a data register and a configuration register. Therefore, each L and C I/O bit can be individually configured under software control as shown below:

| Config. | Data | Ports L and C Setup                  |
|---------|------|--------------------------------------|
| 0       | 0    | Hi-Z Input (TRI-STATE Output)        |
| 0       | 1    | Input with Pull-Up (Weak One Output) |
| 1       | 0    | Push-Pull Zero Output                |
| 1       | 1    | Push-Pull One Output                 |

On the 20- and 28-pin parts, it is recommended that all bits of Port C be configured as outputs to minimize current.

PORT G is an 8-bit port with 6 I/O pins (G0–G5) and 2 input pins (G6, G7). All eight G-pins have Schmitt Triggers on the inputs.

There are two registers associated with the G port: a data register and a configuration register. Therefore, each G port bit can be individually configured under software control as shown below:

| Config. | Data | Port G Setup                         |
|---------|------|--------------------------------------|
| 0       | 0    | Hi-Z Input (TRI-STATE Output)        |
| 0       | 1    | Input with Pull-Up (Weak One Output) |
| 1       | 0    | Push-Pull Zero Output                |
| 1       | 1    | Push-Pull One Output                 |

Since G6 and G7 are input only pins, any attempt by the user to configure them as outputs by writing a one to the configuration register will be disregarded. Reading the G6 and G7 configuration bits will return zeros. The device will be placed in the HALT mode by writing a one to the G7 bit in the G-port data register.

Six pins of Port G have alternate features:

G0 INTR (an external interrupt)

G3 TIO (timer/counter input/output)

G4 SO (MICROWIRE/PLUS serial data output)

G5 SK (MICROWIRE/PLUS clock I/O)

G6 SI (MICROWIRE/PLUS serial data input)

G7 CKO crystal oscillator output (selected by programming the ECON register) or HALT Restart/general purpose input

Pins G1 and G2 currently do not have any alternate functions.

PORT D is an 8-bit output port that is preset high when RESET goes low. Care must be exercised with the D2 pin operation. At reset, the external load on this pin must ensure that the output voltage stay above 0.7  $V_{CC}$  to prevent the chip from entering special modes. Also, keep the external loading on D2 to less than 1000 pF.

## Functional Description

Figure 1 shows the block diagram of the internal architecture. Data paths are illustrated in simplified form to depict how the various logic elements communicate with each other in implementing the instruction set of the device.

### ALU AND CPU REGISTERS

The ALU can do an 8-bit addition, subtraction, logical or shift operation in one cycle time.

There are five CPU registers:

A is the 8-bit Accumulator 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 the 8-bit address register, can be auto incremented or decremented.

X is the 8-bit alternate address register, can be incremented or decremented.

SP is the 8-bit stack pointer, which points to the subroutine/interrupt stack in RAM. The SP must be initialized with software (usually to RAM address 06F Hex with 128 bytes of on-chip RAM selected, or to RAM address 02F Hex with 64 bytes of on-chip RAM selected). The SP is used with the subroutine call and return instructions, and with the interrupts.

B, X and SP registers are mapped into the on-chip RAM. The B and X registers are used to address the on-chip RAM. The SP register is used to address the stack in RAM during subroutine calls and returns.

### PROGRAM MEMORY

The device contains 4096 bytes of UV erasable or OTP EPROM memory. This memory is mapped in the program memory address space from 0000 to 0FFF Hex. The program memory may contain either instructions or data constants, and is addressed by the 15-bit program counter (PC). The program memory can be indirectly read by the LAID (Load Accumulator Indirect) instruction for table lookup of constant data.

All locations in the EPROM program memory will contain 0FF Hex (all 1's) after the device is erased. OTP parts are shipped with all locations already erased to 0FF Hex. Unused EPROM locations should always be programmed to 00 Hex so that the software trap can be used to halt runaway program operation.

The device can be configured to inhibit external reads of the program memory. This is done by programming the security bit in the ECON (EPROM configuration) register to zero. See the ECON REGISTER section for more details.

### DATA MEMORY

The data memory address space includes on-chip RAM, I/O, and registers. Data memory is addressed directly by instructions, or indirectly by means of the B, X, or SP point-

## Functional Description (Continued)

ers. The device can be configured to have either 64 or 128 bytes of RAM, depending on the value of the "RAM SIZE" bit in the ECON (EPROM CONFIGURATION) register. The sixteen bytes of RAM located at data memory address 0F0-0FF are designated as "registers". These sixteen registers can be decremented and tested with the DRSZ (Decrement Register and Skip if Zero) instruction.

The three pointers X, B, and SP are memory mapped into this register address space at addresses 0FC, 0FE, and 0FD respectively. The remaining registers are available for general usage.

Any bit of data memory can be directly set, reset or tested. All of the I/O registers and control registers (except A and PC) are memory mapped. Consequently, any of the I/O bits or control register bits can be directly and individually set, reset, or tested.

**Note:** RAM contents are undefined upon power-up.

### ECON (EPROM CONFIGURATION) REGISTER

The ECON register is used to configure the user selectable clock, security, and RAM size options. The register can be programmed and read only in EPROM programming mode. Therefore, the register should be programmed at the same time as the program memory locations 0000 through 0FFF Hex. UV erasable parts are shipped with 0FF Hex in this register while the OTP parts are shipped with 07F Hex in this register. Erasing the EPROM program memory also erases the ECON register.

The device has a security feature which, when enabled, prevents reading of the EPROM program memory. The security bit in the ECON register determines whether security is enabled or disabled. If the security option is enabled, then any attempt to externally read the contents of the EPROM will result in the value E0 Hex being read from all program memory locations. If the security option is disabled, the contents of the internal EPROM may be read. The ECON register is readable regardless of the state of the security bit.

The format of the ECON register is as follows:

TABLE I

| Bit 7 | Bit 6 | Bit 5    | Bit 4 | Bit 3 | Bit 2 | Bit 1    | Bit 0 |
|-------|-------|----------|-------|-------|-------|----------|-------|
| X     | X     | SECURITY | CKI 2 | CKI 1 | X     | RAM SIZE | X     |

Bit 7 = X Don't care.

Bit 6 = X Don't care.

Bit 5 = 1 Security disabled. EPROM read and write are allowed.

= 0 Security enabled. EPROM read and write are not allowed.

Bits 4,3

= 1,1 External CKI option selected.

= 0,1 Not allowed.

= 1,0 RC oscillator option selected.

= 0,0 Crystal oscillator option selected.

Bit 2 = X Don't care.

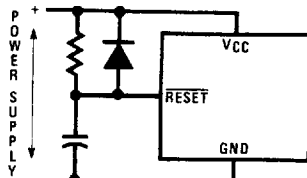
Bit 1 = 1 Selects 128 byte RAM option. This emulates COP840 and COP880.

= 0 Selects 64 byte RAM option. This emulates COP820.

Bit 0 = X Don't care.

### RESET

The RESET input when pulled low initializes the microcontroller. Initialization will occur whenever the RESET input is pulled low. Upon initialization, the Ports L, G and C are placed in the TRI-STATE mode and the Port D is set high. The PC, PSW and CNTRL registers are cleared. The data and configuration registers for Ports L, G and C are cleared. The external RC network shown in Figure 4 should be used to ensure that the RESET pin is held low until the power supply to the chip stabilizes.



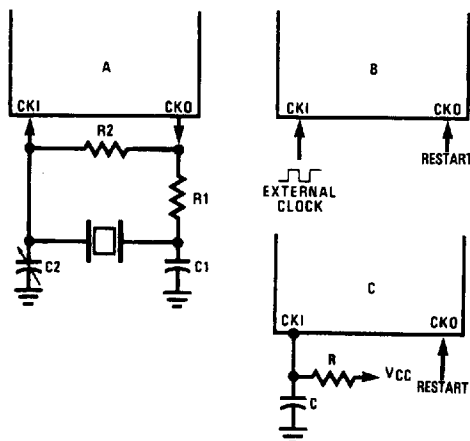
TL/DD/11299-7

RC ≥ 5X Power Supply Rise Time

FIGURE 4. Recommended Reset Circuit

### OSCILLATOR CIRCUITS

Figure 5 shows the three clock oscillator configurations available for the device. The CKI 1 and CKI 2 bits in the ECON register are used to select the clock option. See the ECON REGISTER section for more details.



TL/DD/11299-8

FIGURE 5. Crystal, External and R-C Connection Diagrams

#### A. Crystal Oscillator

The device can be driven by a crystal clock. The crystal network is connected between the pins CKI and CKO.

Table II shows the component values required for various standard crystal frequencies.

#### B. External Oscillator

CKI can be driven by an external clock signal provided it meets the specified duty cycle, rise and fall times, and input levels. In External oscillator mode, G7 is available as a general purpose input and/or HALT restart control.

## Functional Description (Continued)

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

| R1<br>(k $\Omega$ ) | R2<br>(M $\Omega$ ) | C1<br>(pF) | C2<br>(pF) | CKI Freq<br>(MHz) | Conditions    |
|---------------------|---------------------|------------|------------|-------------------|---------------|
| 0                   | 1                   | 30         | 30-36      | 10                | $V_{CC} = 5V$ |
| 0                   | 1                   | 30         | 30-36      | 4                 | $V_{CC} = 5V$ |

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

| R<br>(k $\Omega$ ) | C<br>(pF) | CKI Freq.<br>(MHz) | Instr. Cycle<br>( $\mu\text{s}$ ) | 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: (R/C Oscillator Configuration):  $3k \leq R \leq 200k$ ,  $50 \text{ pF} \leq C \leq 200 \text{ pF}$ .

### C. R/C Oscillator

CKI can be configured as a single pin RC controlled oscillator. In RC oscillator mode, G7 is available as a general purpose input and/or HALT restart control.

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

### HALT MODE

The device supports a power saving mode of operation: HALT. The controller is placed in the HALT mode by setting the G7 data bit, alternatively the user can stop the clock input. (Stopping the clock input will draw more current than setting the G7 data bit.) In the HALT mode all internal processor activities including the clock oscillator are stopped. The fully static architecture freezes the state of the controller and retains all information until continuing. In the HALT mode, power requirements are minimal as it draws only leakage currents and output current. The applied voltage ( $V_{CC}$ ) may be decreased down to  $V_r$  (minimum RAM retention voltage) without altering the state of the machine.

There are two ways to exit the HALT mode: via the **RESET** or by the G7 pin. A low on the **RESET** line reinitializes the microcontroller and starts execution from address 0000H. In external and RC oscillator modes, a low to high transition on the G7 pin also causes the microcontroller to come out of the HALT mode. Execution resumes at the address following the HALT instruction. Except for the G7 data bit, which gets reset, all RAM locations retain the values they had prior to execution of the "HALT" instruction. It is required that the first instruction following the "HALT" instruction be a "NOP" in order to synchronize the clock.

### INTERRUPTS

The device has a sophisticated interrupt structure to allow easy interface to the real world. There are three possible interrupt sources, as shown below.

A maskable interrupt on external G0 input (positive or negative edge sensitive under software control)

A maskable interrupt on timer underflow or timer capture

A non-maskable software/error interrupt on opcode zero

### INTERRUPT CONTROL

The GIE (global interrupt enable) bit enables the interrupt function. This is used in conjunction with ENI and ENTI to select one or both of the interrupt sources. This bit is reset when interrupt is acknowledged.

ENI and ENTI bits select external and timer interrupts respectively. Thus the user can select either or both sources to interrupt the microcontroller when GIE is enabled.

IEDG selects the external interrupt edge (0 = rising edge, 1 = falling edge). The user can get an interrupt on both rising and falling edges by toggling the state of IEDG bit after each interrupt.

IPND and TPND bits signal which interrupt is pending. After an interrupt is acknowledged, the user can check these two bits to determine which interrupt is pending. This permits the interrupts to be prioritized under software. The pending flags have to be cleared by the user. Setting the GIE bit high inside the interrupt subroutine allows nested interrupts.

The software interrupt does not reset the GIE bit. This means that the controller can be interrupted by other interrupt sources while servicing the software interrupt.

### INTERRUPT PROCESSING

The interrupt, once acknowledged, pushes the program counter (PC) onto the stack and the stack pointer (SP) is decremented twice. The Global Interrupt Enable (GIE) bit is reset to disable further interrupts. The microcontroller then vectors to the address 00FFH and resumes execution from that address. This process takes 7 cycles to complete. At the end of the interrupt subroutine, any of the following three instructions return the processor back to the main program: RET, RETSK or RETI. Either one of the three instructions will pop the stack into the program counter (PC). The stack pointer is then incremented twice. The RETI instruction additionally sets the GIE bit to re-enable further interrupts.

Any of the three instructions can be used to return from a hardware interrupt subroutine. The RETSK instruction should be used when returning from a software interrupt subroutine to avoid entering an infinite loop.

**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.

## Functional Description (Continued)

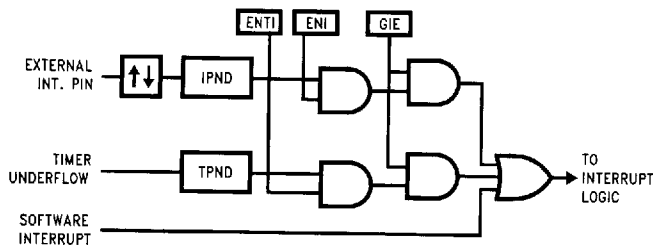


FIGURE 6. Interrupt Block Diagram

TL/DD/11299-9

### DETECTION OF ILLEGAL CONDITIONS

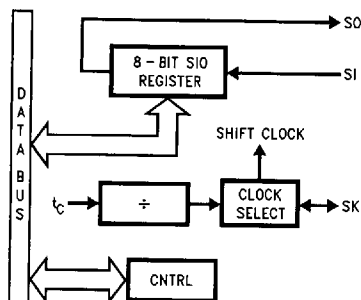
The device incorporates a hardware mechanism that allows it to detect illegal conditions which may occur from coding errors, noise and "brown out" voltage drop situations. Specifically, it detects cases of executing out of undefined EPROM area and unbalanced stack situations.

Reading an undefined EPROM location returns 00 (hexadecimal) as its contents. The opcode for a software interrupt is also "00". Thus a program accessing undefined EPROM will cause a software interrupt.

Reading an undefined RAM location returns an FF (hexadecimal). The subroutine stack on the device grows down for each subroutine call. By initializing the stack pointer to the top of RAM, the first unbalanced return instruction will cause the stack pointer to address undefined RAM. As a result the program will attempt to execute from FFFF (hexadecimal), which is an undefined EPROM location and will trigger a software interrupt.

### MICROWIRE/PLUS

MICROWIRE/PLUS is a serial synchronous bidirectional 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, EEPROMs, etc.) and with other microcontrollers which support the MICROWIRE/PLUS 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 7 shows the block diagram of the MICROWIRE/PLUS interface.



TL/DD/11299-10

FIGURE 7. MICROWIRE/PLUS Block Diagram

The shift clock can be selected from either an internal source or an external source. Operating the MICROWIRE/PLUS interface with the internal clock source is called the Master mode of operation. Operating the MICROWIRE/PLUS interface 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. The SK clock rate is selected by the two bits, SL0 and SL1, in the CNTRL register. Table IV details the different clock rates that may be selected.

TABLE IV

| SL1 | SL0 | SK Cycle Time |
|-----|-----|---------------|
| 0   | 0   | $2t_c$        |
| 0   | 1   | $4t_c$        |
| 1   | x   | $8t_c$        |

where,

$t_c$  is the instruction cycle time.

### MICROWIRE/PLUS OPERATION

Setting the BUSY bit in the PSW register causes the MICROWIRE/PLUS arrangement 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. The device may enter the MICROWIRE/PLUS mode either as a Master or as a Slave. Figure 8 shows how two device microcontrollers and several peripherals may be interconnected using the MICROWIRE/PLUS arrangement.

#### Master MICROWIRE/PLUS Operation

In the MICROWIRE/PLUS Master mode of operation the shift clock (SK) is generated internally by the device. The MICROWIRE/PLUS Master always initiates all data exchanges (Figure 8). The MSEL bit in the CNTRL register must be set to enable the SO and SK functions on 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 V summarizes the bit settings required for Master mode of operation.

#### SLAVE MICROWIRE/PLUS OPERATION

In the MICROWIRE/PLUS Slave mode of operation the SK clock is generated by an external source. Setting the MSEL

## Functional Description (Continued)

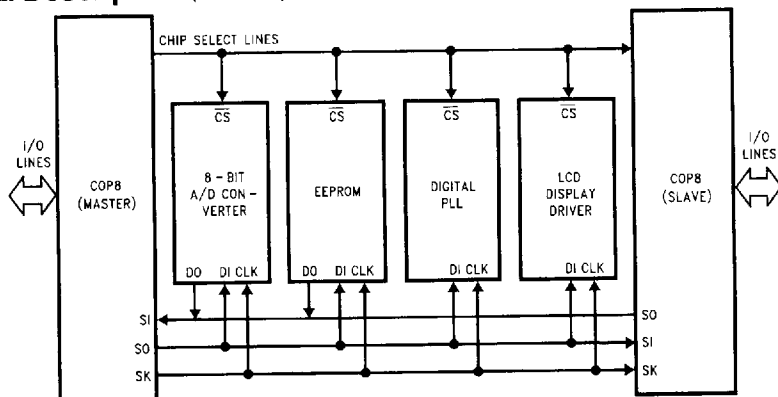


FIGURE 8. MICROWIRE/PLUS Application

TL/DD/11299-11

bit in the CNTRL register enables the SO and SK functions on the G Port. The SK pin must be selected as an input and the SO pin selected as an output pin by appropriately setting up the Port G configuration register. Table V summarizes the settings required to enter the Slave mode of operation.

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 (Figure 8).

TABLE V

| G4 Config. Bit | G5 Config. Bit | G4 Fun.   | G5 Fun. | G6 Fun. | Operation        |
|----------------|----------------|-----------|---------|---------|------------------|
| 1              | 1              | SO        | Int. SK | SI      | MICROWIRE Master |
| 0              | 1              | TRI-STATE | Int. SK | SI      | MICROWIRE Master |
| 1              | 0              | SO        | Ext. SK | SI      | MICROWIRE Slave  |
| 0              | 0              | TRI-STATE | Ext. SK | SI      | MICROWIRE Slave  |

## TIMER/COUNTER

The device has a powerful 16-bit timer with an associated 16-bit register enabling it to perform extensive timer functions. The timer T1 and its register R1 are each organized as two 8-bit read/write registers. Control bits in the register CNTRL allow the timer to be started and stopped under software control. The timer-register pair can be operated in one of three possible modes. Table VI details various timer operating modes and their requisite control settings.

## MODE 1. TIMER WITH AUTO-LOAD REGISTER

In this mode of operation, the timer T1 counts down at the instruction cycle rate. Upon underflow the value in the register R1 gets automatically reloaded into the timer which continues to count down. The timer underflow can be programmed to interrupt the microcontroller. A bit in the control register CNTRL enables the TIO (G3) pin to toggle upon timer underflows. This allows the generation of square-wave outputs or pulse width modulated outputs under software control (Figure 9).

## MODE 2. EXTERNAL COUNTER

In this mode, the timer T1 becomes a 16-bit external event counter. The counter counts down upon an edge on the TIO pin. Control bits in the register CNTRL program the counter to decrement either on a positive edge or on a negative edge. Upon underflow the contents of the register R1 are automatically copied into the counter. The underflow can also be programmed to generate an interrupt (Figure 9).

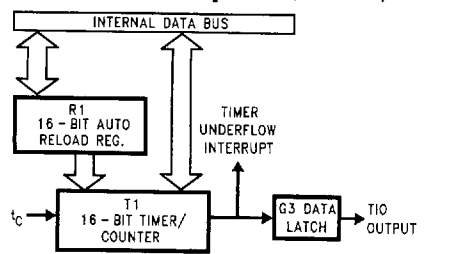
## MODE 3. TIMER WITH CAPTURE REGISTER

Timer T1 can be used to precisely measure external frequencies or events in this mode of operation. The timer T1 counts down at the instruction cycle rate. Upon the occurrence of a specified edge on the TIO pin the contents of the timer T1 are copied into the register R1. Bits in the control register CNTRL allow the trigger edge to be specified either as a positive edge or as a negative edge. In this mode the user can elect to be interrupted on the specified trigger edge (Figure 10).

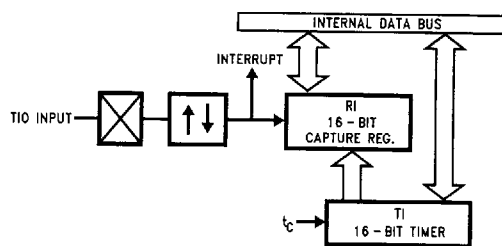
TABLE VI. Timer Operating Modes

| CNTRL Bits 7 6 5 | Operation Mode                        | T Interrupt     | Timer Counts On |
|------------------|---------------------------------------|-----------------|-----------------|
| 0 0 0            | External Counter w/Auto-Load Reg.     | Timer Underflow | TIO Pos. Edge   |
| 0 0 1            | External Counter w/Auto-Load Reg.     | Timer Underflow | TIO Neg. Edge   |
| 0 1 0            | Not Allowed                           | Not Allowed     | Not Allowed     |
| 0 1 1            | Not Allowed                           | Not Allowed     | Not Allowed     |
| 1 0 0            | Timer w/Auto-Load Reg.                | Timer Underflow | $t_c$           |
| 1 0 1            | Timer w/Auto-Load Reg./Toggle TIO Out | Timer Underflow | $t_c$           |
| 1 1 0            | Timer w/Capture Register              | TIO Pos. Edge   | $t_c$           |
| 1 1 1            | Timer w/Capture Register              | TIO Neg. Edge   | $t_c$           |

## Functional Description (Continued)



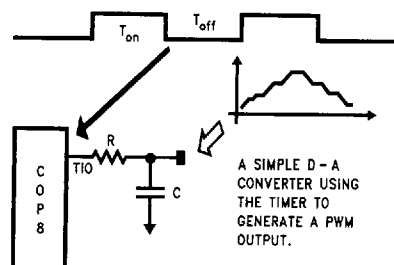
**FIGURE 9. Timer/Counter Auto Reload Mode Block Diagram**



**FIGURE 10. Timer Capture Mode Block Diagram**

### TIMER PWM APPLICATION

Figure 11 shows how a minimal component D/A converter can be built out of the Timer-Register pair in the Auto-Reload mode. The timer is placed in the "Timer with auto reload" mode and the TIO pin is selected as the timer output. At the outset the TIO pin is set high, the timer T1 holds the on time and the register R1 holds the signal off time. Setting TRUN bit starts the timer which counts down at the instruction cycle rate. The underflow toggles the TIO output and copies the off time into the timer, which continues to run. By alternately loading in the on time and the off time at each successive interrupt a PWM frequency can be easily generated.



**FIGURE 11. Timer Application**

## Control Registers

### CNTRL REGISTER (ADDRESS X'00EE)

The Timer and MICROWIRE/PLUS control register contains the following bits:

- SL1 & SL0 Select the MICROWIRE/PLUS clock divide-by
- IEDG External interrupt edge polarity select  
(0 = rising edge, 1 = falling edge)
- MSEL Enable MICROWIRE/PLUS functions SO and SK
- TRUN Start/Stop the Timer/Counter (1 = run, 0 = stop)
- TC3 Timer input edge polarity select (0 = rising edge, 1 = falling edge)
- TC2 Selects the capture mode
- TC1 Selects the timer mode

| TC1   | TC2 | TC3 | TRUN | MSEL  | IEDG | S1 | S0 |
|-------|-----|-----|------|-------|------|----|----|
| Bit 7 |     |     |      | Bit 0 |      |    |    |

### PSW REGISTER (ADDRESS X'00EF)

The PSW register contains the following select bits:

- GIE Global interrupt enable
- ENI External interrupt enable
- BUSY MICROWIRE/PLUS busy shifting
- IPND External interrupt pending
- ENTI Timer interrupt enable
- TPND Timer interrupt pending
- C Carry Flag
- HC Half carry Flag

| HC    | C | TPND | ENTI | IPND  | BUSY | ENI | GIE |
|-------|---|------|------|-------|------|-----|-----|
| Bit 7 |   |      |      | Bit 0 |      |     |     |

## Addressing Modes

### REGISTER INDIRECT

This is the "normal" mode of addressing for the device. The operand is the memory location addressed by the B register or X register.

### DIRECT

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

### IMMEDIATE

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

### REGISTER INDIRECT (AUTO INCREMENT AND DECREMENT)

This is a register indirect mode that automatically increments or decrements the B or X register after executing the instruction.

### RELATIVE

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

## Memory Map

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

| RAM Select                                     | Address                                                           | Contents                                                                                                                                                                                                                                                                |
|------------------------------------------------|-------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 64 On-Chip RAM Bytes<br>Selected by ECON reg.  | 00–2F<br>30–7F                                                    | 48 On-Chip RAM Bytes<br>Unused RAM Address Space (Reads as all 1's)                                                                                                                                                                                                     |
| 128 On-Chip RAM Bytes<br>Selected by ECON reg. | 00–6F<br>70–7F                                                    | 112 On-chip RAM Bytes<br>Unused RAM Address Space (Reads as all 1's)                                                                                                                                                                                                    |
|                                                | 80 to BF                                                          | Expansion Space for On-Chip EERAM                                                                                                                                                                                                                                       |
|                                                | C0 to CF                                                          | Expansion Space for I/O and Registers                                                                                                                                                                                                                                   |
|                                                | D0 to DF<br>D0<br>D1<br>D2<br>D3<br>D4<br>D5<br>D6<br>D7          | On-Chip I/O and Registers<br>Port L Data Register<br>Port L Configuration Register<br>Port L Input Pins (Read Only)<br>Reserved for Port L<br>Port G Data Register<br>Port G Configuration Register<br>Port G Input Pins (Read Only)<br>Port I Input Pins (Read Only)   |
|                                                | D8<br>D9<br>DA<br>DB<br>DC<br>DD–DF                               | Port C Data Register<br>Port C Configuration Register<br>Port C Input Pins (Read Only)<br>Reserved for Port C<br>Port D Data Register<br>Reserved for Port D                                                                                                            |
|                                                | E0 to EF<br>E0–E7<br>E8<br>E9<br>EA<br>EB<br>EC<br>ED<br>EE<br>EF | On-Chip Functions and Registers<br>Reserved for Future Parts<br>Reserved<br>MICROWIRE/PLUS Shift Register<br>Timer Lower Byte<br>Timer Upper Byte<br>Timer Autoload Register Lower Byte<br>Timer Autoload Register Upper Byte<br>CNTRL Control Register<br>PSW Register |
|                                                | F0 to FF<br>FC<br>FD<br>FE                                        | On-Chip RAM Mapped as Registers<br>X Register<br>SP Register<br>B Register                                                                                                                                                                                              |

Reading unused memory locations below 7FH will return all ones. Reading other unused memory locations will return undefined data.

# Instruction Set

## REGISTER AND SYMBOL DEFINITIONS

### 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  | Half Carry                                        |
| GIE | 1-bit of PSW register for global interrupt enable |

### Symbols

|      |                                                            |
|------|------------------------------------------------------------|
| [B]  | Memory indirectly addressed by B register                  |
| [X]  | Memory indirectly addressed by X register                  |
| Mem  | Direct address memory or [B]                               |
| Meml | Direct address 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

|                                                                                                                  |                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ADD<br>ADC<br><br>SUBC<br><br>AND<br>OR<br>XOR<br>IFEQ<br>IFGT<br>IFBNE<br>DRSZ<br>SBIT<br><br>RBIT<br><br>IFBIT | add<br>add with carry<br><br>subtract with carry<br><br>Logical AND<br>Logical OR<br>Logical Exclusive-OR<br>IF equal<br>IF greater than<br>IF B not equal<br>Decrement Reg., skip if zero<br>Set bit<br><br>Reset bit<br><br>If bit  | A ← A + Meml<br>A ← A + Meml + C, C ← Carry<br>HC ← Half Carry<br>A ← A + Meml + C, C ← Carry<br>HC ← Half Carry<br>A ← A and Meml<br>A ← A or Meml<br>A ← A xor Meml<br>Compare A and Meml, Do next if A = Meml<br>Compare A and Meml, Do next if A > Meml<br>Do next if lower 4 bits of B ≠ Imm<br>Reg ← Reg - 1, skip if Reg goes to 0<br>1 to bit,<br>Mem (bit = 0 to 7 immediate)<br>0 to bit,<br>Mem<br>If bit,<br>Mem is true, do next instr. |
| X<br>LD A<br>LD mem<br>LD Reg                                                                                    | Exchange A with memory<br>Load A with memory<br>Load Direct memory Immed.<br>Load Register memory Immed.                                                                                                                              | A ↔ Mem<br>A ← Meml<br>Mem ← Imm<br>Reg ← Imm                                                                                                                                                                                                                                                                                                                                                                                                        |
| X<br>X<br>LD A<br>LD A<br>LD M                                                                                   | Exchange A with memory [B]<br>Exchange A with memory [X]<br>Load A with memory [B]<br>Load A with memory [X]<br>Load Memory Immediate                                                                                                 | A ↔ [B] (B ← B ± 1)<br>A ↔ [X] (X ← X ± 1)<br>A ← [B] (B ← B ± 1)<br>A ← [X] (X ← X ± 1)<br>[B] ← Imm (B ← B ± 1)                                                                                                                                                                                                                                                                                                                                    |
| CLRA<br>INCA<br>DECA<br>LAID<br>DCORA<br>RRCA<br>SWAPA<br>SC<br>RC<br>IFC<br>IFNC                                | Clear A<br>Increment A<br>Decrement A<br>Load A indirect from ROM<br>DECIMAL CORRECT A<br>ROTATE A RIGHT THRU C<br>Swap nibbles of A<br>Set C<br>Reset C<br>If C<br>If not C                                                          | A ← 0<br>A ← A + 1<br>A ← A - 1<br>A ← ROM(PU,A)<br>A ← BCD correction (follows ADC, SUBC)<br>C → A7 → ... → A0 → C<br>A7 ... A4 ↔ A3 ... A0<br>C ← 1, HC ← 1<br>C ← 0, HC ← 0<br>If C is true, do next instruction<br>If C is not true, do next instruction                                                                                                                                                                                         |
| JMPL<br>JMP<br>JP<br>JSRL<br>JSR<br>JID<br>RET<br>RETSK<br>RETI<br>INTR<br>NOP                                   | Jump absolute long<br>Jump absolute<br>Jump relative short<br>Jump subroutine long<br>Jump subroutine<br>Jump indirect<br>Return from subroutine<br>Return and Skip<br>Return from Interrupt<br>Generate an interrupt<br>No operation | PC ← ii (ii = 15 bits, 0 to 32k)<br>PC11..0 ← i (i = 12 bits)<br>PC ← PC + r (r is -31 to +32, not 1)<br>[SP] ← PL, [SP-1] ← PU, SP-2, PC ← ii<br>[SP] ← PL, [SP-1] ← PU, SP-2, PC11..0 ← i<br>PL ← ROM(PU,A)<br>SP+2, PL ← [SP], PU ← [SP-1]<br>SP+2, PL ← [SP], PU ← [SP-1], Skip next instruction<br>SP+2, PL ← [SP], PU ← [SP-1], GIE ← 1<br>[SP] ← PL, [SP-1] ← PU, SP-2, PC ← 0FF<br>PC ← PC + 1                                               |

Bits 7-4

| OPCODE LIST |        |            |          |             |             |              |             |              |             |          |          |               |               |         | Bits 3-0  |  |
|-------------|--------|------------|----------|-------------|-------------|--------------|-------------|--------------|-------------|----------|----------|---------------|---------------|---------|-----------|--|
| 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] | *           | LD B, 0F | IFBNE 0  | JSR 0000-00FF | JMP 0000-00FF | JP + 17 | INTR 0    |  |
| JP -14      | JP -30 | LD 0F1, #i | DRSZ 0F1 | *           | SC          | SUBCA, #i    | SUBC A, [B] | IFBIT 1, [B] | *           | LD B, 0E | IFBNE 1  | JSR 0100-01FF | JMP 0100-01FF | 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 0200-02FF | JMP 0200-02FF | 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 0300-03FF | JMP 0300-03FF | JP + 20 | JP + 4 3  |  |
| JP -11      | JP -27 | LD 0F4, #i | DRSZ 0F4 | *           | LAID        | ADD A, #i    | ADD A, [B]  | IFBIT 4, [B] | CLRA        | LD B, 0B | IFBNE 4  | JSR 0400-04FF | JMP 0400-04FF | JP + 21 | JP + 5 4  |  |
| JP -10      | JP -26 | LD 0F5, #i | DRSZ 0F5 | *           | JID         | AND A, #i    | AND A, [B]  | IFBIT 5, [B] | SWAPA       | LD B, 0A | IFBNE 5  | JSR 0500-05FF | JMP 0500-05FF | JP + 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, 9  | IFBNE 6  | JSR 0600-06FF | JMP 0600-06FF | JP + 23 | JP + 7 6  |  |
| JP -8       | JP -24 | LD 0F7, #i | DRSZ 0F7 | *           | *           | OR A, #i     | OR A, [B]   | IFBIT 7, [B] | *           | LD B, 8  | IFBNE 7  | JSR 0700-07FF | JMP 0700-07FF | JP + 24 | JP + 8 7  |  |
| JP -7       | JP -23 | LD 0F8, #i | DRSZ 0F8 | NOP         | *           | LD A, #i     | IFC         | SBIT 0, [B]  | RBIT 0, [B] | LD B, 7  | IFBNE 8  | JSR 0800-08FF | JMP 0800-08FF | JP + 25 | JP + 9 8  |  |
| JP -6       | JP -22 | LD 0F9, #i | DRSZ 0F9 | *           | *           | *            | IFNC        | SBIT 1, [B]  | RBIT 1, [B] | LD B, 6  | IFBNE 9  | JSR 0900-09FF | JMP 0900-09FF | JP + 26 | JP + 10 9 |  |
| JP -5       | JP -21 | LD 0FA, #i | DRSZ 0FA | LD A, [X +] | LD A, [B +] | LD [B +], #i | INCA        | SBIT 2, [B]  | RBIT 2, [B] | LD B, 5  | IFBNE 0A | JSR 0A00-0AFF | JMP 0A00-0AFF | 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, 4  | IFBNE 0B | JSR 0B00-0BFF | JMP 0B00-0BFF | JP + 28 | JP + 12 B |  |
| JP -3       | JP -19 | LD 0FC, #i | DRSZ 0FC | LD Md, #i   | JMPL        | X A, Md      | *           | SBIT 4, [B]  | RBIT 4, [B] | LD B, 3  | IFBNE 0C | JSR 0C00-0CFF | JMP 0C00-0CFF | 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, 2  | IFBNE 0D | JSR 0D00-0DFF | JMP 0D00-0DFF | 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, 1  | IFBNE 0E | JSR 0E00-0EFF | JMP 0E00-0EFF | JP + 31 | JP + 15 E |  |
| JP -0       | JP -16 | LD 0FF, #1 | DRSZ 0FF | *           | *           | *            | RETI        | SBIT 7, [B]  | RBIT 7, [B] | LD B, 0  | IFBNE 0F | JSR 0F00-0FFF | JMP 0F00-0FFF | JP + 32 | JP + 16 F |  |

\* is an unused opcode (see following table)

Md is a directly addressed memory location

i is the immediate data

where,

## Instruction Execution Time

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

Most single 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.

**Arithmetic Instructions (Bytes/Cycles)**

|       | [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    |        |

**Memory Transfer Instructions (Bytes/Cycles)**

|            | Register Indirect<br>[B] [X] |     | Direct | Immed. | Register Indirect<br>Auto Incr & Decr<br>[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 |                              |     | 3/3    |        | 2/2                                                            |     |
| LD Reg,Imm |                              |     |        | 2/3    |                                                                |     |

(If B < 16)

(If B > 15)

\* => Memory location addressed by B or X or directly.

**Instructions Using A & C**

| Instructions | Bytes/Cycles |
|--------------|--------------|
| CLRA         | 1/1          |
| INCA         | 1/1          |
| DECA         | 1/1          |
| LAID         | 1/3          |
| DCORA        | 1/1          |
| RRCA         | 1/1          |
| SWAPA        | 1/1          |
| SC           | 1/1          |
| RC           | 1/1          |
| IFC          | 1/1          |
| IFNC         | 1/1          |

**Transfer of Control Instructions**

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

## Bytes and Cycles per Instruction (Continued)

The following table shows the instructions assigned to unused opcodes. This table is for information only. The operations performed are subject to change without notice. Do not use these opcodes.

| Unused Opcode | Instruction | Unused Opcode | Instruction |
|---------------|-------------|---------------|-------------|
| 60            | NOP         | A9            | NOP         |
| 61            | NOP         | AF            | LD A, [B]   |
| 62            | NOP         | B1            | C → HC      |
| 63            | NOP         | B4            | NOP         |
| 67            | NOP         | B5            | NOP         |
| 8C            | RET         | B7            | X A, [X]    |
| 99            | NOP         | B9            | NOP         |
| 9F            | LD [B], #i  | BF            | LD A, [X]   |
| A7            | X A, [B]    |               |             |
| A8            | NOP         |               |             |

## 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-COP880C—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.

## Development Support (Continued)

### 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 12 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 2.4-5.5VDC 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 kbyte 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-880C20DWPC     | 20 DIP                                 |
| MHW-880C28DWPC     | 28 DIP                                 |
| MHW-880C40DWPC     | 40 DIP                                 |
| MHW-880C44PWPC     | 44 PLCC                                |
| DIP to SO Adapters |                                        |
| MHW-SOIC20         | 20 SO                                  |
| MHW-SOIC28         | 28 SO                                  |

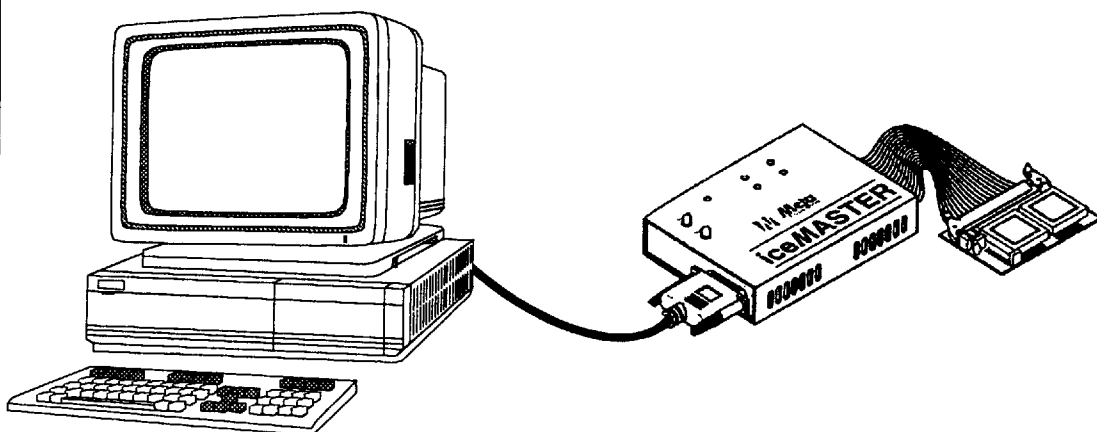


FIGURE 12. COP iceMASTER Environment

TL/DD/11299-15

## 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 13 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 kbyte 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 wall mount power supply.
- On-board V<sub>pp</sub> generator from 5V input or connection to external supply supported. Requires V<sub>pp</sub> 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 Module Unit  |         |
|--------------------|---------|
| COP8-DM/880C       |         |
| Cable Adapters     |         |
| DM-COP8/20D        | 20 DIP  |
| DM-COP8/28D        | 28 DIP  |
| DM-COP8/40D        | 40 DIP  |
| DM-COP8/44P        | 44 PLCC |
| DIP to SO Adapters |         |
| COP8-DM/20D-SO     | 20 SO   |
| COP8-DM/28D-SO     | 28 SO   |

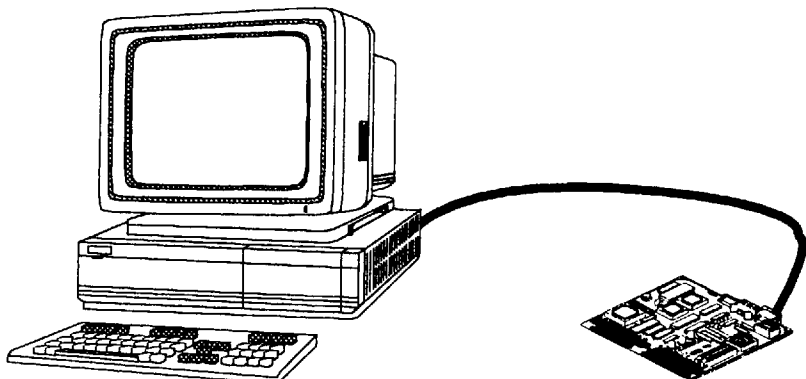


FIGURE 13. COP8-DM Environment

TL/DD/11299-16

## Development Support (Continued)

### IceMASTER EVALUATION PROGRAMMING UNIT (EPU)

The iceMASTER EPU-COP880C is a PC based, in-circuit simulation tool to support the feature family COP8 products. See Figure 14 for configuration.

The simulation capability is a very low cost means of evaluating the general COP8 architecture. In addition, the EPU has programming capability, with added adapters, for programming the whole COP8 product family of OTP and EPROM products. The product includes the following features:

- Non-real-time in-circuit simulation. Program overlay memory is PC resident; instructions are downloaded over RS-232 as executed. Approximate performance is 20 kHz.
- Includes a 40 pin DIP cable adapter. Other target packages are not supported. All processor I/O pins are cabled to the application development environment.
- 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.
- On-chip timer and WATCHDOG™ execution are not well synchronized to the instruction simulation.
- 100 frames of synchronous trace memory. The display can be HLL source (e.g., C source), assembly or mixed. The most recent history prior to a break is available in the trace memory.
- Up to 8 software configured break points; uses INTR instruction which is modestly intrusive.
- Common look-feel debugger software across all Meta-Link products—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.
- Processor specific symbolic display of registers and bit level assignments, configured from master model file.
- Halt/Idle mode notification. Restart requires special handling.
- Programming menu supports full product line of programmable OTP and EPROM COP8 products. Only a 40 ZIF socket is available on the EPU unit. Adapters are available for other part package configurations.
- Integral wall mount power supply provides 5V and develops the required  $V_{pp}$  to program parts.
- Includes a copy of COP8-DEV-IBMA assembler, linker SDK.

#### EPU Order Information

| Evaluation Programming Unit  |                                                                                                         |
|------------------------------|---------------------------------------------------------------------------------------------------------|
| EPU-COP880C                  | Evaluation Programming Unit with debugger and programmer control software and 40 ZIF programming socket |
| General Programming Adapters |                                                                                                         |
| COP8-PGMA-DS                 | 28 and 20 DIP and SOIC adapter                                                                          |
| COP8-PGMA-DS44P              | 28 and 20 DIP and SOIC plus 44 PLCC adapter                                                             |

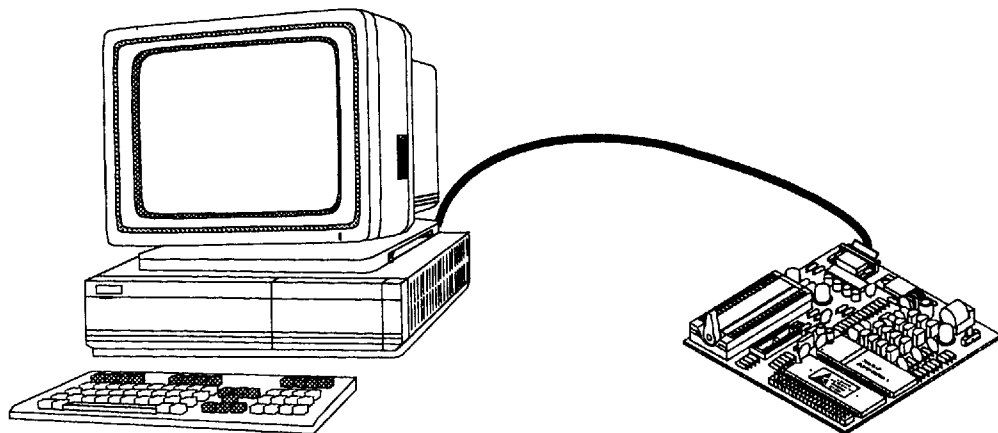


FIGURE 14. EPU-COP8 Tool Environment

TL/DD/11299-17

## 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 source level supported by the MetaLink debugger.

### CROSS REFERENCE TABLE

The following cross reference table lists the COP800 devices which can be emulated with the COP87XXC single-chip, form fit and function emulators.

Single-Chip Emulator Selection Table

| Device Number | Package | Description                 | Emulates                  |
|---------------|---------|-----------------------------|---------------------------|
| COP8780CV     | 44 PLCC | One Time Programmable (OTP) | COP880C                   |
| COP8780CEL    | 44 LDCC | UV Erasable                 | COP880C                   |
| COP8780CN     | 40 DIP  | OTP                         | COP880C                   |
| COP8780CJ     | 40 DIP  | UV Erasable                 | COP880C                   |
| COP8781CN     | 28 DIP  | OTP                         | COP881C, COP840C, COP820C |
| COP8781CJ     | 28 DIP  | UV Erasable                 | COP881C, COP840C, COP820C |
| COP8781CWM    | 28 SO   | OTP                         | COP881C, COP840C, COP820C |
| COP8782CN     | 20 DIP  | OTP                         | COP842C, COP822C          |
| COP8782CJ     | 20 DIP  | UV Erasable                 | COP842C, COP822C          |
| COP8782CWM    | 20 SO   | OTP                         | COP842C, COP822C          |

### 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.

## Development Support (Continued)

Approved List

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

### 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 a WorldWide Web Browser

ftp://nscmicro.nsc.com

### National Semiconductor on the WorldWide Web

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

## Development Support (Continued)

### 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:         | 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 |
| 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     |

## Programming Considerations

In addition to the application program, the ECON register needs to be programmed as well. The following tables provide examples of some ECON register values. For more detailed information refer to the ECON REGISTER section.

### EPROM Security Disabled

| RAM Memory | External CKI | RC Oscillator | Crystal Oscillator |
|------------|--------------|---------------|--------------------|
| 64 Bytes   | 38           | 30            | 20                 |
| 128 Bytes  | 3A           | 32            | 22                 |

### EPROM Security Enabled

| RAM Memory | External CKI | RC Oscillator | Crystal Oscillator |
|------------|--------------|---------------|--------------------|
| 64 Bytes   | 18           | 10            | 00                 |
| 128 Bytes  | 1A           | 12            | 02                 |

EPROM programmer manufacturers may not all calculate a "checksum" the same way. Before implementing an in-house verification by comparing checksums, need to ensure how each programming system utilized calculates a checksum. It is strongly recommended not to include the ECON register in the checksum for not all programmers include this byte in their calculated checksum.

### ERASING THE COP8780C EPROM

The EPROM program memory is erased by exposing the transparent window on the UV erasable packages to an ultraviolet light source. Erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000Å to 4000Å range.

After programming, "truly opaque" labels should be placed over the window of the device to prevent functional failure due to the generation of photo currents, erasure and excessive HALT current. The term "truly opaque" needs additional clarification when used in the context of covering quartz windows on these devices. The typical white colored stickers or labels are normally used for they are easy to write on.

These stickers are not opaque but translucent, they do let a certain percentage of UV-light through. The black write-protect labels supplied with 5.25" floppy disks work extremely well. If problems are encountered during programming (fails blank check) or during normal operation (intermittent functional or logical failures), need to determine first if an appropriate opaque label is being used to cover the quartz window at all times. Note that the device will also draw more current than normal (especially in HALT mode) when the window of the device is not covered with an opaque label.

The recommended erasure procedure for the device is exposure to short wave ultraviolet light which has a wavelength of 2537Å. The integrated dose (UV intensity × exposure time) for erasure should be a minimum of 30W-sec/cm<sup>2</sup>.

The device should be placed within one inch of the lamp tubes during erasure. Some lamps have a filter on their tubes which should be removed before erasure. The following table shows the minimum erasure time for various light intensities.

Minimum Erasure Time

| Light Intensity<br>(Micro-Watts/cm <sup>2</sup> ) | Erasure Time*<br>(Minutes) |
|---------------------------------------------------|----------------------------|
| 15,000                                            | 36                         |
| 10,000                                            | 50                         |
| 8,500                                             | 60                         |

\*Does not include light intensity ramp up time.

An erasure system should be calibrated periodically. The distance from lamp to device should be maintained at one inch. The erasure time increases as the square of the distance. Lamps lose intensity as they age. When a lamp has aged, the system should be checked to make certain that adequate UV dosages are being applied for full erasure.

Common symptoms of insufficient erasure are:

- Inability to be programmed.
- Operational malfunctions associated with V<sub>CC</sub>, temperature, or clock frequency.
- Loss of data in program memory.
- A change in configuration values in the ECON register.