National Semiconductor

PRELIMINARY
September 1999

COP87LxxCJ/RJ Family
8-Bit CMOS OTP Microcontrollers with 4k or 32k Memory and Comparator

General Description

The COP87LxxCJ/RJ Family OTP (One Time Programmable) microcontrollers are integrated COP8™ Base core devices with 4k or 32k memory, and an Analog comparator (no brownout). These multi-chip CMOS devices are suited for lower-functionality applications, and as pre-production devices for a ROM design. Low cost, pin and software compatible (plus Brownout) 1k or 2k ROM versions are available (COP820CJ/840CJ Family). Versions are available for use with a range of COP8 software and hardware development tools.

Family features include an 8-bit memory mapped architecture, 10 MHz CKI with 1µs instruction cycle, three clock options (-1 = crystal; -2 = external; -3 = internal RC), one multi-function 16-bit timer/counter, MICROWIRE/PLUS™ serial I/O, one analog comparator, power saving HALT mode with multi-sourced wakeup/interrupt capability, on-chip R/C oscillator capacitor, high current outputs, software selectable I/O options, WATCHDOG™ timer, modulator/timer, Power on Reset, program code security, 2.7V to 5.5V operation and 20/28 pin packages.

In this datasheet, the term COP87L20CJ refers to the COP87L20CJ, and COP87L22CJ. COP840CJ refers to the COP87L40CJ, COP87L42CJ, COP87L40RJ, and COP87L42RJ.

Devices included in this datasheet are:

Device	Memory (bytes)	RAM (bytes)	I/O Pins	Packages	Temperature
COP87L20CJ	4k OTP EPROM	64	24	28 DIP/SOIC	-40 to +85˚C
COP87L22CJ	4k OTP EPROM	64	16	20 DIP/SOIC	-40 to +85˚C
COP87L40CJ	4k OTP EPROM	128	24	28 DIP/SOIC	-40 to +85˚C
COP87L42CJ	4k OTP EPROM	128	16	20 DIP/SOIC	-40 to +85˚C
COP87L40RJ	32k OTP EPROM	128	24	28 DIP/SOIC	-40 to +85˚C
COP87L42RJ	32k OTP EPROM	128	16	20 DIP/SOIC	-40 to +85˚C

Key Features

- Multi-Input Wakeup (on the 8-bit Port L)
- Analog comparator
- Modulator/Timer (high speed PWM timer for IR transmission)
- 16-bit multi-function timer supporting
  - — PWM mode
  - — External event counter mode
  - — Input capture mode
- Integrated capacitor for the R/C oscillator
- 4 or 32 kbyte on-board OTP EPROM with security feature
- 64 or 128 bytes on-chip RAM

I/O Features

- Software selectable I/O options (TRI-STATE®, Push-Pull, Weak Pull-Up Input, High Impedance Input)
- High current outputs (8 pins)
- Schmitt trigger inputs on Port G
- MICROWIRE/PLUS serial I/O
- Packages:
  - — 20 DIP/SO with 16 I/O pins
  - — 28 DIP/SO with 24 I/O pins

CPU/Instruction Set Features

- 1 µs 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 µA)
- Single supply operation: 2.7V to 5.5V
- Temperature range: −40˚C to +85˚C

Development Support

- Emulation device for the COP820CJ/COP840CJ
- Real time emulation and full program debug offered by MetaLink Development Systems

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

Block Diagram

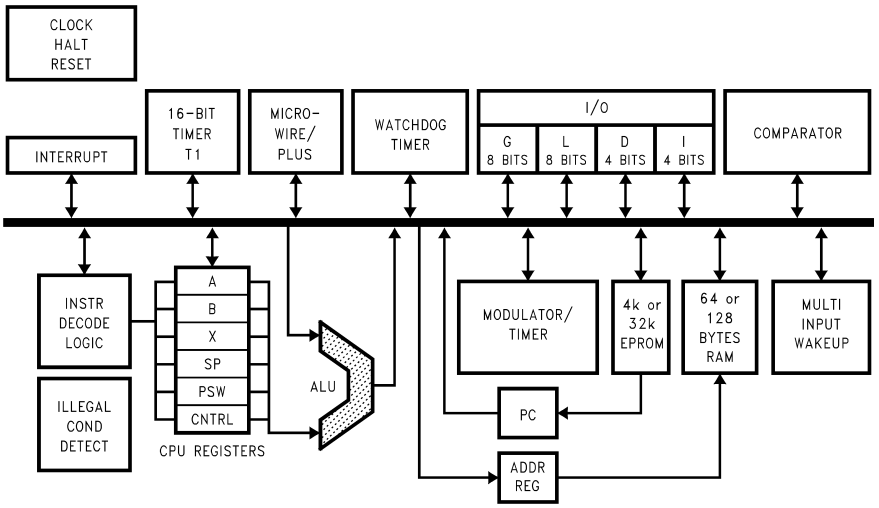
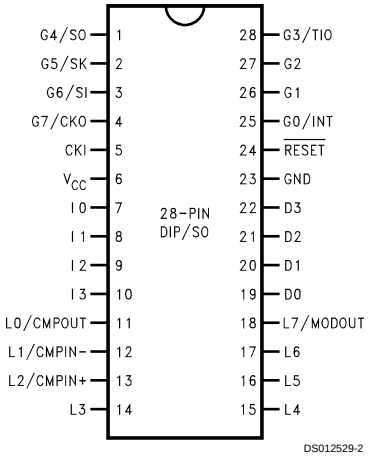
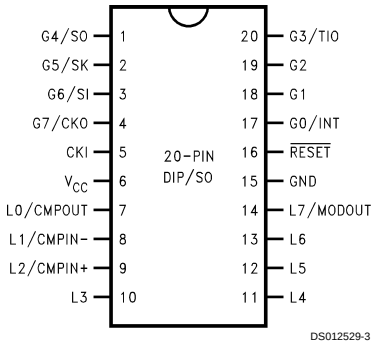


FIGURE 1. Block Diagram

Connection Diagrams



Top View
Order Number
COP87L20CJN (-1N, -2N, -3N), or
COP87L20CJM(-1N, -2N, -3N), or
COP87L40CJN (-1N, -2N, -3N), or
COP87L40CJM (-1N, -2N, -3N), or
COP87L40RJN (-1N, -2N, -3N), or
COP87L40RJM (-1N, -2N, -3N)
See NS Package Number N28B or M28B



Top View
Order Number
COP87L22CJN (-1N, -2N, -3N), or
COP87L22CJM(-1N, -2N, -3N), or
COP87L42CJN (-1N, -2N, -3N), or
COP87L42CJM (-1N, -2N, -3N), or
COP87L42RJN (-1N, -2N, -3N), or
COP87L42RJM (-1N, -2N, -3N)
See NS Package Number N20A or M20B

FIGURE 2. Connection Diagrams

Note: -1 Crystal Oscillator N - Brown out disabled
-2 External Oscillator
-3 R/C Oscillator

Pin Assignment

Port Pin	Typ	ALT Funct.	20 Pin	28 Pin
L0	I/O	MIWU/CMPOUT	7	11
L1	I/O	MIWU/CMPIN−	8	12
L2	I/O	MIWU/CMPIN+	9	13
L3	I/O	MIWU	10	14
L4	I/O	MIWU	11	15
L5	I/O	MIWU	12	16
L6	I/O	MIWU	13	17
L7	I/O	MIWU/MODOUT	14	18
G0	I/O	INTR	17	25
G1	I/O		18	26
G2	I/O		19	27
G3	I/O	TIO	20	28
G4	I/O	SO	1	1
G5	I/O	SK	2	2
G6	I	SI	3	3
G7	I	CKO	4	4
I0	I			7
I1	I			8
I2	I			9
I3	I			10
D0	O			19
D1	O			20
D2	O			21
D3	O			22
V_{CC}			6	6
GND			15	23
CKI			5	5
$\overline{\text{RESET}}$			16	24

Absolute Maximum Ratings (Note 1)

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

Supply Voltage (V_{CC})	7.0V
Voltage at any Pin	−0.3V to V_{CC} + 0.3V
Total Current into V_{CC} pin (Source)	80 mA
Total Current out of GND pin (sink)	80 mA
Storage Temperature Range	−65˚C to +150˚C

Note 1: *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 1 (Note 2)	Peak to Peak			0.1 V_{CC}	V
Supply Current (Note 3)					
CKI = 10 MHz	V_{CC} = 5.5V, tc = 1 µs			12	mA
CKI = 4 MHz	V_{CC} = 4.5V, tc = 2.5 µs			6.5	mA
CKI = 4 MHz (COP87L20CJ)	V_{CC} = 4.0V, tc = 2.5 µs			10	mA
HALT Current (Note 4)	V_{CC} = 5.5V, CKI = 0 MHz			12	µA
INPUT LEVELS (V_{IH}, V_{IL})					
Reset, CKI:					
Logic High		0.8 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
L- and G-Port Hysteresis (Note 7)				0.35 V_{CC}	V
Output Current Levels					
D Outputs:					
Source	V_{CC} = 4.5V, V_{OH} = 3.8V	−0.4			mA
Sink (Note 5)	V_{CC} = 4.5V, V_{OL} = 1.0V	10			mA
L4–L7 Output Sink	V_{CC} = 4.5V, V_{OL} = 2.5V	15			mA
All Others					
Source (Weak Pull-up Mode)	V_{CC} = 4.5V, V_{OH} = 3.2V	−10		−110	µA
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
(COP887L20CJ)	V_{CC} = 5.5V, V_{OL} = 0.4V				
TRI-STATE Leakage		−2.0		+2.0	µA
Allowable Sink/Source Current Per Pin					
D Outputs				15	mA
L4–L7 (Sink)				20	mA
All Others				3	mA
Maximum Input Current without Latchup (Note 6)	Room Temperature			±100	mA
RAM Retention Voltage, V_r	500 ns Rise and Fall Time (Min)	2.0			V
Input Capacitance				7	pF
Load Capacitance on D2				1000	pF

Note 2: Rate of voltage change must be less than 10 V/mS.

Note 3: Supply current is measured after running 2000 cycles with a square wave CKI input, CKO open, inputs at rails and outputs open.

DC Electrical Characteristics (Continued)

Note 4: The HALT mode will stop CKI from oscillating in the RC and crystal configurations by bringing CKI high. HALT test conditions: L, and G0..G5 ports configured as outputs and set high. The D port set to zero. All inputs tied to V_{CC}. The comparator is disabled.

Note 5: 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.

Note 6: Pins G6 and $\overline{\text{RESET}}$ are designed with a high voltage input network. 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 750Ω (typical). These two pins will not latch up. The voltage at the pins must be limited to less than 14V.

AC Electrical Characteristics

$-40^{\circ}C \le T_A \le +85^{\circ}C$ unless otherwise specified

Parameter	Conditions	Min	Typ	Max	Units
Instruction Cycle Time (tc)					
Crystal/Resonator	$4.5V \le V_{CC} \le 5.5V$	1		DC	μs
R/C Oscillator	$4.5V \le V_{CC} \le 5.5V$	2		DC	μs
CKI Clock Duty Cycle (Note 7)	fr = Max	40		60	%
Rise Time (Note 7)	fr = 10 MHz ext. Clock			12	ns
Fall Time (Note 7)	fr = 10 MHz ext. Clock			8	ns
Inputs					
t_{Setup}	$4.5V \le V_{CC} \le 5.5V$	200			ns
t_{Hold}	$4.5V \le V_{CC} \le 5.5V$	60			ns
Output Propagation Delay	R_L = 2.2k, CL = 100 pF				
t_{PD1}, t_{PD0}					
SO, SK	$4.5V \le V_{CC} \le 5.5V$			0.7	μs
All Others	$4.5V \le V_{CC} \le 5.5V$			1	μs
Input Pulse Width					
Interrupt Input High Time		1			tc
Interrupt Input Low Time		1			tc
Timer Input High Time		1			tc
Timer Input Low Time		1			tc
MICROWIRE™ Setup Time ($t_{\mu WS}$)		20			ns
MICROWIRE Hold Time ($t_{\mu WH}$)		56			ns
MICROWIRE Output				220	ns
Propagation Delay ($t_{\mu PD}$)					
Reset Pulse Width		1			μs

Note 7: Parameter characterized but not production tested.

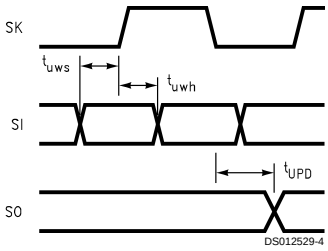


FIGURE 3. MICROWIRE/PLUS Timing

Pin Description

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.

$\overline{\text{RESET}}$ is the master reset input. See Reset description.

PORT I is a 4-bit Hi-Z input port.

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

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

Port L Config.	Port L Data	Port L Setup
0	0	Hi-Z Input (TRI-STATE)
0	1	Input with Weak Pull-up
1	0	Push-pull Zero Output
1	1	Push-pull One Output

Three data memory address locations are allocated for this port, one each for data register [00D0], configuration register [00D1] and the input pins [00D2].

Port L has the following alternate features:

L7 MIWU or MODOUT (high sink current capability)

L6 MIWU (high sink current capability)

L5 MIWU (high sink current capability)

L4 MIWU (high sink current capability)

L3 MIWU

L2 MIWU or CMPIN+

L1 MIWU or CMPIN−

L0 MIWU or CMPOUT

The selection of alternate Port L functions is done through registers WKEN [00C9] to enable MIWU and CNTRL2 [00CC] to enable comparator and modulator.

All eight L-pins have Schmitt Triggers on their inputs.

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:

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

Three data memory address locations are allocated for this port, one for data register [00D4], one for configuration register [00D5] and one for the input pins [00D6]. Since G6 and G7 are Hi-Z 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. Note that the device will be placed in the Halt mode by writing a "1" to the G7 data bit.

Six pins of Port G have alternate features:

G7 CKO crystal oscillator output (selected by mask option) or HALT restart input/general purpose input (if clock option is R/C or external clock)

G6 SI (MICROWIRE serial data input)

G5 SK (MICROWIRE clock I/O)

G4 SO (MICROWIRE serial data output)

G3 TIO (timer/counter input/output)

G0 INTR (an external interrupt)

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

The selection of alternate Port G functions are done through registers PSW [00EF] to enable external interrupt and CNTRL1 [00EE] to select TIO and MICROWIRE operations.

PORT D is a four bit output port that is preset when RESET goes low. One data memory address location is allocated for the data register [00DC]. The user can tie two or more D port outputs (except D2 pin) together in order to get a higher drive.

Note: Care must be exercised with the D2 pin operation. At RESET, the external loads on this pin must ensure that the output voltages stay above 0.8 V_{CC} to prevent the chip from entering special modes. Also keep the external loading on D2 to less than 1000 pF.

Functional Description

The internal architecture is shown in the block diagram. 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 operations in one 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 the 8-bit address register and can be auto incremented or decremented.

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

SP is the 8-bit stack pointer which points to the subroutine stack (in RAM).

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. The SP must be initialized by software before any subroutine call or interrupts occurs.

Memory

The memory is separated into two memory spaces: program and data.

PROGRAM MEMORY

Program memory consists of 4 kbytes of OTP EPROM. These bytes of ROM may be instructions or constant data. The memory is addressed by the 15-bit program counter (PC). ROM can be indirectly read by the LAID instruction for table lookup.

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

Memory (Continued)

SECURITY FEATURE

The 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 arrary 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 unsecure and FF(hex) if secure.

DATA MEMORY

The data memory address space includes on chip RAM, I/O and registers. Data memory is addressed directly by the instruction or indirectly through B, X and SP registers. The device has 128 bytes of RAM. Sixteen bytes of RAM are mapped as "registers", these can be loaded immediately, decremented and tested. Three specific registers: X, B, and SP are mapped into this space, the other registers are available for general usage.

Any bit of data memory can be directly 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, except the write once only bit (WDREN, WATCHDOG Reset Enable), and the unused and read only bits in CNTRL2 and WDREG registers.

Note: RAM contents are undefined upon power-up.

Reset

EXTERNAL RESET

The RESET input pin when pulled low initializes the micro-controller. The user must insure that the RESET pin is held low until V_{CC} is within the specified voltage range and the clock is stabilized. An R/C circuit with a delay 5x greater than the power supply rise time is recommended (*Figure 4*). The device immediately goes into reset state when the RESET input goes low. When the RESET pin goes high the device comes out of reset state synchronously. The device will be running within two instruction cycles of the RESET pin going high. The following actions occur upon reset:

Port L	TRI-STATE
Port G	TRI-STATE
Port D	HIGH
PC	CLEARED
RAM Contents	RANDOM with Power-On-Reset UNAFFECTED with external Reset (power already applied)
B, X, SP	Same as RAM
PSW, CNTRL1, CNTRL2 and WDREG Reg.	CLEARED
Multi-Input Wakeup Reg. WKEDG, WKEN WKPND	 CLEARED UNKNOWN
Data and Configuration Registers for L & G	CLEARED
WATCHDOG Timer	Prescaler/Counter each loaded with FF

The device comes out of the HALT mode when the RESET pin is pulled low. In this case, the user has to ensure that the RESET signal is low long enough to allow the oscillator to restart. An internal 256 t_c delay is normally used in conjunction with the two pin crystal oscillator. When the device comes out of the HALT mode through Multi-Input Wakeup, this delay allows the oscillator to stabilize.

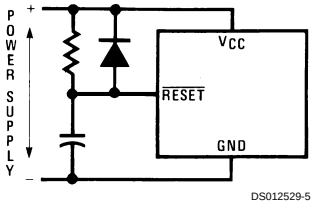
The following additional actions occur after the device comes out of the HALT mode through the RESET pin.

If a two pin crystal/resonator oscillator is being used:

RAM Contents	UNCHANGED
Timer T1 and A Contents	UNKNOWN
WATCHDOG Timer Prescaler/Counter	ALTERED

If the external or RC Clock option is being used:

RAM Contents	UNCHANGED
Timer T1 and A Contents	UNCHANGED
WATCHDOG Timer Prescaler/Counter	ALTERED



RC > 5 x Power Supply Rise Time

FIGURE 4. Recommended Reset Circuit

WATCHDOG RESET

With WATCHDOG enabled, the WATCHDOG logic resets the device if the user program does not service the WATCHDOG timer within the selected service window. The WATCHDOG reset does not disable the WATCHDOG. Upon WATCHDOG reset, the WATCHDOG Prescaler/Counter are each initialized with FF Hex.

The following actions occur upon WATCHDOG reset that are different from external reset.

WDREN WATCHDOG Reset Enable bit UNCHANGED

WDUDF WATCHDOG Underflow bit UNCHANGED

Additional initialization actions that occur as a result of WATCHDOG reset are as follows:

Port L	TRI-STATE
Port G	TRI-STATE
Port D	HIGH
PC	CLEARED
Ram Contents	UNCHANGED
B, X, SP	UNCHANGED
PSW, CNTRL1 and CNTRL2 (except WDUDF Bit) Registers	CLEARED

Reset (Continued)

Multi-Input Wakeup Registers WKEDG, WKEN WKPND	 CLEARED UNKNOWN
Data and Configuration Registers for L & G	 CLEARED
WATCHDOG Timer	Prescalar/Counter each loaded with FF

Oscillator Circuits

EXTERNAL OSCILLATOR

By selecting the external oscillator option, the CKI pin can be driven by an external clock signal provided it meets the specified duty cycle, rise and fall times, and input levels. The G7/CKO is available as a general purpose input G7 and/or HALT control.

CRYSTAL OSCILLATOR

By selecting the crystal oscillator option, the G7/CKO pin is connected as a clock output, CKI and G7/CKO can be connected to make a crystal controlled oscillator. *Table 1* shows the clock frequency for different component values. See *Figure 5* for the connections.

R/C OSCILLATOR

By selecting R/C oscillator option, connecting a resistor from the CKI pin to V_{CC} makes a R/C oscillator. The capacitor is on-chip. The G7/CKO pin is available as a general purpose input G7 and/or HALT control. Adding an external capacitor will jeopardize the clock frequency tolerance and increase EMI emissions.

Table 2 shows the clock frequency for the different resistor values. The capacitor is on-chip. See *Figure 5* for the connections.

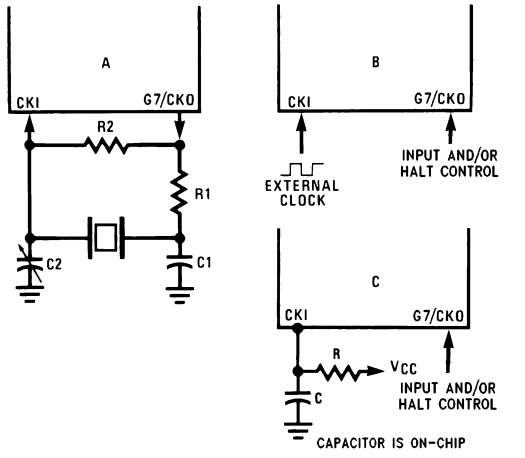


FIGURE 5. Clock Oscillator Configurations

TABLE 1. Crystal Oscillator Configuration

R1 (kΩ)	R2 (MΩ)	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
5.6	1	100	100–156	0.455	V_{CC} = 5V

TABLE 2. RC Oscillator Configuration (Part-To-Part Variation) T_A = 25˚C

R (kΩ)	CK1 Freq. (MHz)	Instr. Cycle (µs)	Conditions
8.2	3.3 ±10%	3.0 ±10%	V_{CC} = 5V
2.2	1.3 ±10%	7.7 ±10%	V_{CC} = 5V
3.9	0.75 ±10%	13.3 ±10%	V_{CC} = 5V

Halt Mode

The device is a fully static device. The device enters the HALT mode by writing a one to the G7 bit of the G data register. Once in the HALT mode, the internal circuitry does not receive any clock signal and is therefore frozen in the exact state it was in when halted. In this mode the chip will only draw leakage current.

The device supports three different methods of exiting the HALT mode. The first 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 is a dedicated output). It may be used either with an RC clock configuration or an external clock configuration. The second method of exiting the HALT mode is with the multi-Input Wakeup feature on the L port. The third method of exiting the HALT mode is by pulling the RESET input low.

If the two pin crystal/resonator oscillator is being used and Multi-Input Wakeup causes the device to exit the HALT mode, the WAKEUP signal does not allow the chip to start running immediately since crystal oscillators have a delayed start up time to reach full amplitude and freuqency stability. The WATCHDOG timer (consisting of an 8-bit prescaler followed by an 8-bit counter) is used to generate a fixed delay of 256tc 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 WATCHDOG Counter and Prescaler are each loaded with a value of FF Hex. The WATCHDOG prescaler is clocked with the tc instruction cycle. (The tc clock is derived by dividing the oscillator clock down by a factor of 10).

The Schmitt trigger following the CKI inverter on the chip ensures that the WATCHDOG timer is clocked only when the oscillator has a sufficiently large amplitude to meet the Schmitt trigger specs. This Schmitt trigger is not part of the oscillator closed loop. The start-up timeout from the WATCHDOG timer enables the clock signals to be routed to the rest of the chip. The delay is not activated when the device comes out of HALT mode through RESET pin. Also, if the clock option is either RC or External clock, the delay is not used, but the WATCHDOG Prescaler/-Counter contents are changed. The Development System will not emulate the 256tc delay.

The $\overline{\text{RESET}}$ pin will cause the device to reset and start executing from address X'0000. A low to high transition on the G7 pin (if single pin oscillator is used) or Multi-Input Wakeup will cause the device to start executing from the address following the HALT instruction.

When $\overline{\text{RESET}}$ pin is used to exit the device from the HALT mode and the two pin crystal/resonator (CKI/CKO) clock option is selected, the contents of the Accumulator and the Timer T1 are undetermined following the reset. All other information except the WATCHDOG Prescaler/Counter contents is retained until continuing. All information except the WATCHDOG Prescaler/Counter contents is retained if the device exits the HALT mode through G7 pin or Multi-Input Wakeup.

G7 is the HALT-restart pin, but it can still be used as an input. If the device is not halted, G7 can be used as a general purpose input.

Note: To allow clock resynchronization, it is necessary to program two NOP's immediately after the device comes out of the HALT mode. The user must program two NOP's following the "enter HALT mode" (set G7 data bit) instruction.

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 6* shows the block diagram of the MICROWIRE/PLUS interface.

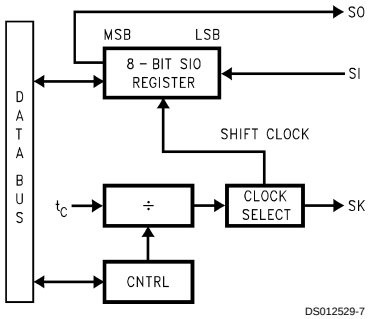


FIGURE 6. MICROWIRE/PLUS Block Diagram

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 3* details the different clock rates that may be selected.

TABLE 3.

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 7* 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 ex-

MICROWIRE/PLUS (Continued)

changes *(Figure 7)*. 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 4* 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 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 4* 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 (see *Figure 7*).

TABLE 4.

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

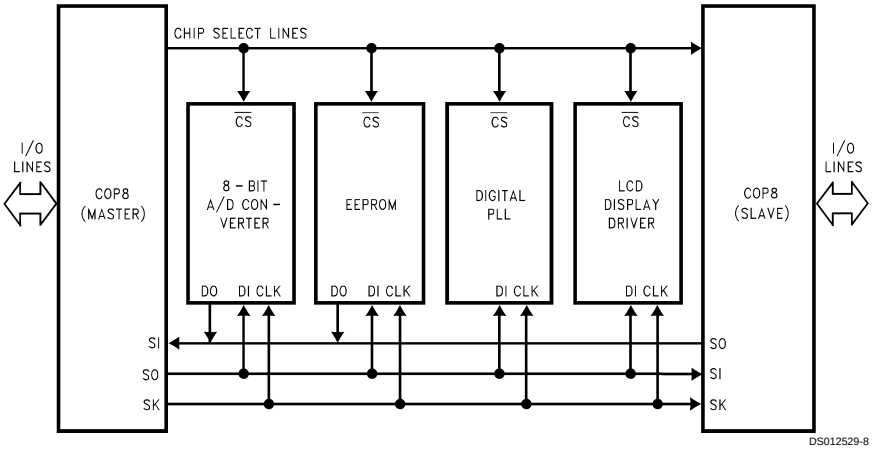


FIGURE 7. MICROWIRE/PLUS Application

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 5* details various timer operating modes and their requisite control settings.

TABLE 5. Timer Operating Modes

CNTRL Bits 7 6 5	Operation Mode	T Interrupt	Timer Counter 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

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 8*).

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 8*).

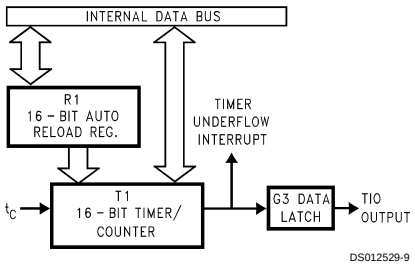


FIGURE 8. Timer/Counter Auto Reload Mode Block Diagram

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 9*).

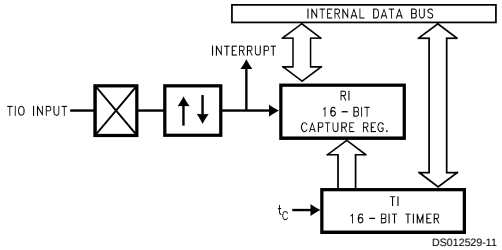


FIGURE 9. Timer Capture Mode Block Diagram

TIMER PWM APPLICATION

Figure 10 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.

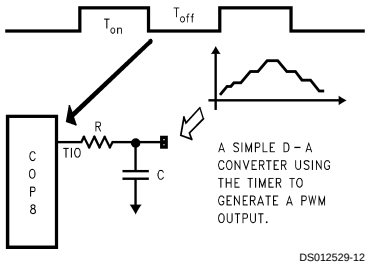


FIGURE 10. Timer Application

WATCHDOG

The device has an on-board 8-bit WATCHDOG timer. The timer contains an 8-bit READ/WRITE down counter clocked by an 8-bit prescaler. Under software control the timer can be dedicated for the WATCHDOG or used as a general purpose counter. *Figure 11* shows the WATCHDOG timer block diagram.

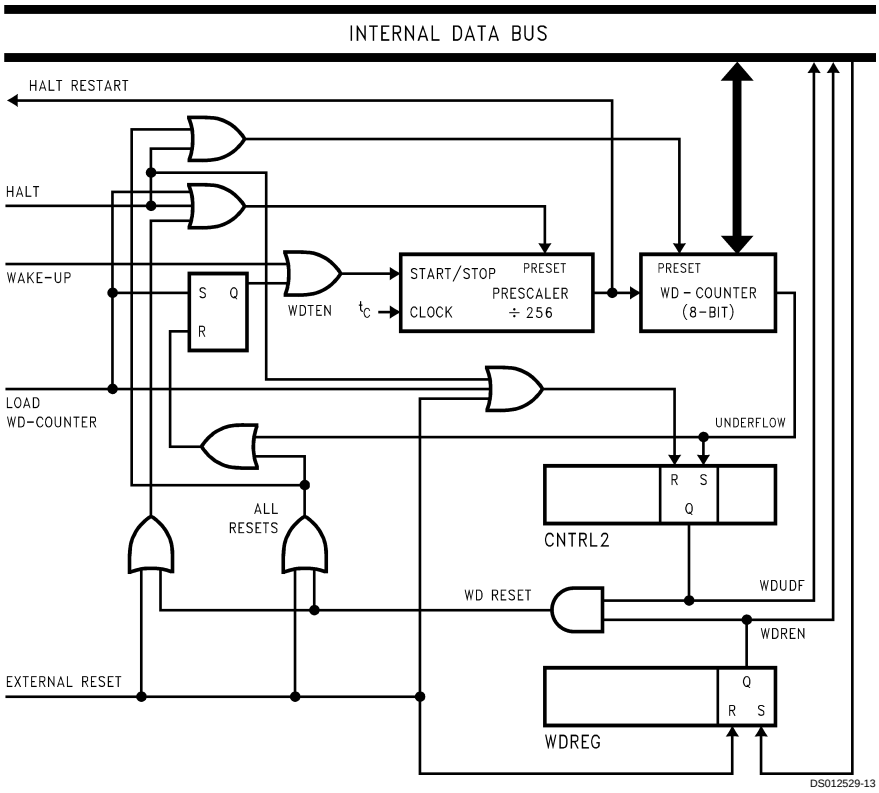


FIGURE 11. WATCHDOG Timer Block Diagram

MODE 1: WATCHDOG TIMER

The WATCHDOG is designed to detect user programs getting stuck in infinite loops resulting in loss of program control or "runaway" programs. The WATCHDOG can be enabled or disabled (only once) after the device is reset as a result of external reset. On power-up the WATCHDOG is disabled. The WATCHDOG is enabled by writing a "1" to WDREN bit (resides in WDREG register). Once enabled, the user program should write periodically into the 8-bit counter before the counter underflows. The 8-bit counter (WDCNT) is memory mapped at address 0CE Hex. The counter is loaded with n-1 to get n counts. The counter underflow resets the device, but does not disable the WATCHDOG. Loading the 8-bit counter initializes the prescaler with FF Hex and starts the prescaler/counter. Prescaler and counter are stopped upon counter underflow. Prescaler and counter are each loaded with FF Hex when the device goes into the HALT mode. The prescaler is used for crystal/resonator start-up when the device exits the HALT mode through Multi-Input Wakeup. In this case, the prescaler/counter contents are changed.

MODE 2: TIMER

In this mode, the prescaler/counter is used as a timer by keeping the WDREN (WATCHDOG reset enable) bit at 0. The counter underflow sets the WDUDF (underflow) bit and the underflow does not reset the device. Loading the 8-bit counter (load n-1 for n counts) sets the WDTEN bit (WATCHDOG Timer Enable) to "1", loads the prescaler with FF, and starts the timer. The counter underflow stops the timer. The WDTEN bit serves as a start bit for the WATCHDOG timer. This bit is set when the 8-bit counter is loaded by the user program. The load could be as a result of WATCHDOG service (WATCHDOG timer dedicated for WATCHDOG function) or write to the counter (WATCHDOG timer used as a general purpose counter). The bit is cleared upon Brown Out reset, WATCHDOG reset or external reset. The bit is not memory mapped and is transparent to the user program.

CONTROL/STATUS BITS

WDUDF: WATCHDOG Timer Underflow Bit

This bit resides in the CNTRL2 Register. The bit is set when the WATCHDOG timer underflows. The underflow resets the device if the WATCHDOG reset enable bit is set (WDREN = 1). Otherwise, WDUDF can be used as the timer underflow

WATCHDOG (Continued)

flag. The bit is cleared upon external reset, load to the 8-bit counter, or going into the HALT mode. It is a read only bit.

WDREN: WD Reset Enable

WDREN bit resides in a separate register (bit 0 of WDREG). This bit enables the WATCHDOG timer to generate a reset. The bit is cleared upon external reset. The bit under software control can be written to only once (once written to, the hardware does not allow the bit to be changed during program execution).

WDREN = 1 WATCHDOG reset is enabled.

WDREN = 0 WATCHDOG reset is disabled.

Table 6 shows the impact of WATCHDOG Reset and External Reset on the Control/Status bits.

TABLE 6. WATCHDOG Control/Status

Parameter	HALT Mode	WD Reset	EXT Reset	Counter Load
8-Bit Prescaler	FF	FF	FF	FF
8-Bit WD Counter	FF	FF	FF	User Value
WDREN Bit	Unchanged	Unchanged	0	No Effect
WDUDF Bit	0	Unchanged	0	0
WDTEN Signal	Unchanged	0	0	1

Modulator/Timer

The Modulator/Timer contains an 8-bit counter and an 8-bit autoreload register (MODRL address 0CF Hex). The Modulator/Timer has two modes of operation, selected by the control bit MC3. The Modulator/Timer Control bits MC1, MC2 and MC3 reside in CNTRL2 Register.

MODE 1: MODULATOR

The Modulator is used to generate high frequency pulses on the modulator output pin (L7). The L7 pin should be configured as an output. The number of pulses is determined by the 8-bit down counter. Under software control the modulator input clock can be either CKI or tC. The tC clock is derived by dividing down the oscillator clock by a factor of 10. Three control bits (MC1, MC2, and MC3) are used for the Modulator/Timer output control. When MC2 = 1 and MC3 = 1, CKI is used as the modulator input clock. When MC2 = 0, and MC3 = 1, tC is used as the modulator input clock. The user loads the counter with the desired number of counts (256 max) and sets MC1 to start the counter. The modulator autoreload register is loaded with n-1 to get n pulses. CKI or tc pulses are routed to the modulator output (L7) until the counter underflows (*Figure 12*). Upon underflow the hardware resets MC1 and stops the counter. The L7 pin goes low and stays low until the counter is restarted by the user program. The user program has the responsibility to timeout the low time. Unless the number of counts is changed, the user program does not have to load the counter each time the counter is started. The counter can simply be started by setting the MC1 bit. Setting MC1 by software will load the counter with the value of the autoreload register. The software can reset MC1 to stop the counter.

MODE 2: PWM TIMER

The counter can also be used as a PWM Timer. In this mode, an 8-bit register is used to serve as an autoreload register (MODRL).

a. 50% Duty Cycle:

When MC1 is 1 and MC2, MC3 are 0, a 50% duty cycle free running signal is generated on the L7 output pin (*Figure 13*). The L7 pin must be configured as an output pin. In this mode the 8-bit counter is clocked by tC. Setting the MC1 control bit by software loads the counter with the value of the autoreload register and starts the counter. The counter underflow toggles the (L7) output pin. The 50% duty cycle signal will be continuously generated until MC1 is reset by the user program.

b. Variable Duty Cycle:

When MC3 = 0 and MC2 = 1, a variable duty cycle PWM signal is generated on the L7 output pin. The counter is clocked by tC. In this mode the 16-bit timer T1 along with the 8-bit down counter are used to generate a variable duty cycle PWM signal. The timer T1 underflow sets MC1 which starts the down counter and it also sets L7 high (L7 should be configured as an output).When the counter underflows the MC1 control bit is reset and the L7 output will go low until the next timer T1 underflow. Therefore, the width of the output pulse is controlled by the 8-bit counter and the pulse duration is controlled by the 16-bit timer T1 (*Figure 14*). Timer T1 must be configured in "PWM Mode/Toggle TIO Out" (CNTRL1 Bits 7,6,5 = 101).

Table 7 shows the different operation modes for the Modulator/Timer.

TABLE 7. Modulator/Timer Modes

Control Bits in CNTRL2(00CC)			Operation Mode L7 Function
MC3	MC2	MC1	
0	0	0	Normal I/O
0	0	1	50% Duty Cycle Mode (Clocked by tc)
0	1	X	Variable Duty Cycle Mode (Clocked by tc) Using Timer 1 Underflow
1	0	X	Modulator Mode (Clocked by tc)
1	1	X	Modulator Mode (Clocked by CKI)

Note: MC1, MC2 and MC3 control bits are cleared upon reset.

Modulator/Timer (Continued)

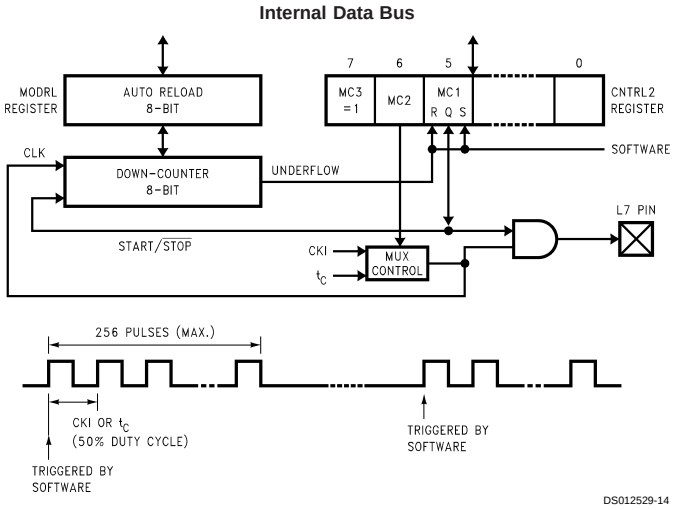


FIGURE 12. Mode 1: Modulator Block Diagram/Output Waveform

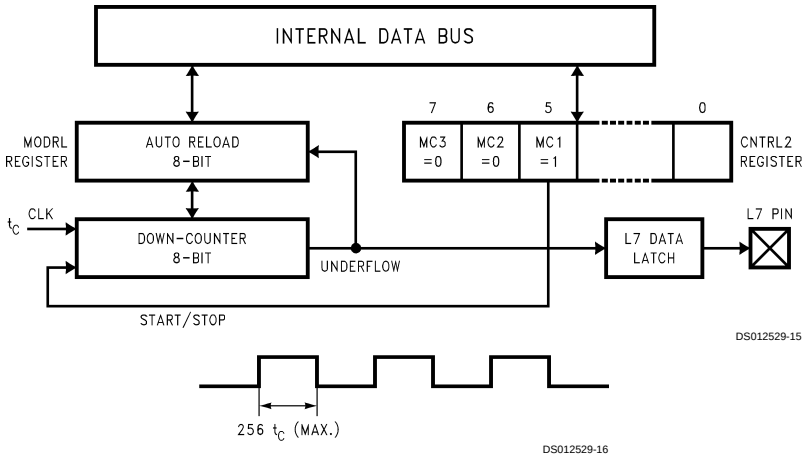


FIGURE 13. Mode 2a: 50% Duty Cycle Output

Modulator/Timer (Continued)

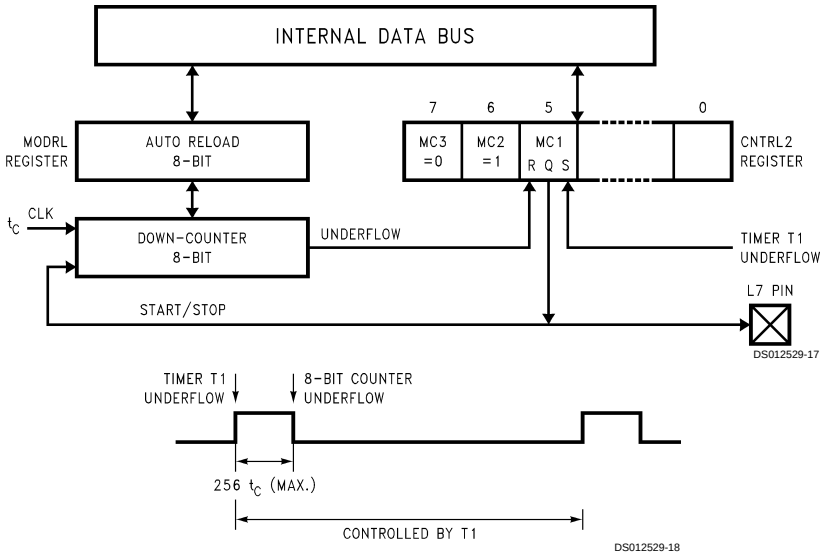


FIGURE 14. Mode 2b: Variable Duty Cycle Output

Comparator

The device has one differential comparator. Ports L0–L2 are used for the comparator. The output of the comparator is brought out to a pin. Port L has the following assignments:

L0 Comparator output

L1 Comparator negative input

L2 Comparator positive input

THE COMPARATOR STATUS/CONTROL BITS

These bits reside in the CNTRL2 Register (Address 0CC)

CMPEN Enables comparator ("1" = enable)

CMPRD Reads comparator output internally
(CMPEN = 1, CMPOE=X)

CMPOE Enables comparator output to pin L0
("1"=enable), CMPEN bit must be set to enable this function. If CMPEN=0, L0 will be 0.

The Comparator Select/Control bits are cleared on RESET (the comparator is disabled). To save power the program should also disable the comparator before the device enters the HALT mode.

The user program must set up L0, L1 and L2 ports correctly for comparator Inputs/Output: L1 and L2 need to be configured as inputs and L0 as output. *Table 8* shows the DC and AC characteristics for the comparator.

TABLE 8. DC and AC Characteristics (Note 8) **4.5V ≤ V_{CC} ≤ 5.5V, −40˚C ≤ T_A ≤ +85˚C**

Parameters	Conditions	Min	Typ	Max	Units
Input Offset Voltage	0.4V < V_{IN} < V_{CC} −1.5V		±10	±25	mV
Input Common Mode Voltage Range		0.4		V_{CC} −1.5	V
Voltage Gain			300k		V/V
DC Supply Current (when enabled)	V_{CC} = 5.5V			250	µA
Response Time	TBD mV Step, TBD mV Overdrive, 100 pF Load			1	µs

Note 8: For comparator output current characteristics see L-Port specs.

Multi-Input Wake Up

The Multi-Input Wakeup feature is used to return (wakeup) the device from the HALT mode. *Figure 15* shows the Multi-Input Wakeup logic.

This 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 mode. Three 8-bit memory mapped registers, Reg:WKEN, Reg:WKEDG, and Reg:WKPND are used in conjunction with the L port to implement the Multi-Input Wakeup feature.

All three registers Reg:WKEN, Reg:WKPND, and Reg-:WKEDG are read/write registers, and are cleared at reset, except WKPND. WKPND is unknown on reset.

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).

Multi-Input Wake Up (Continued)

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 Wakeup 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.

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. The program would be as follows:

```
RBIT   5, WKEN      ; Disable MIWU
SBIT   5, WKEDG     ; Change edge polarity
RBIT   5, WKPND     ; Reset pending flag
SBIT   5, WKEN      ; Enable MIWU
```

If the L port bits have been used as outputs and then changed to inputs with Multi-Input Wakeup, a safety procedure should also be followed to avoid inherited pseudo wakeup 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 Wakeup is latched into a pending register called Reg: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 the Reg:WKPND is a pending register for the occurrence of selected wakeup conditions, the device will not enter the HALT mode if any Wakeup bit is both enabled and pending. Setting the G7 data bit under this condition will not allow the device to enter the HALT mode. Consequently, the user has the responsibility of clearing the pending flags before attempting to enter the HALT mode.

If a crystal oscillator is being used, the Wakeup signal will not start the chip running immediately since crystal oscillators have a finite start up time. The WATCHDOG timer prescaler generates a fixed delay to ensure that the oscillator has indeed stabilized before allowing the device to execute instructions. In this case, upon detecting a valid Wakeup signal only the oscillator circuitry and the WATCHDOG timer are enabled. The WATCHDOG timer prescaler is loaded with a value of FF Hex (256 counts) and is clocked from the tc instruction cycle clock. The tc 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 WATCHDOG timer is clocked only when the oscillator has a sufficiently large amplitude to meet the Schmitt trigger specs. This Schmitt trigger is not part of the oscillator closed loop. The startup timeout from the WATCHDOG timer enables the clock signals to be routed to the rest of the chip.

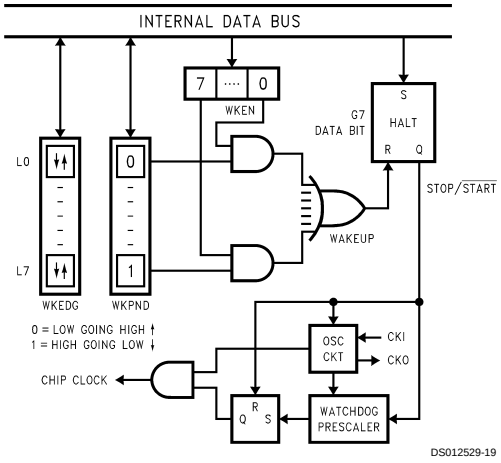


FIGURE 15. Multi-Input Wakeup Logic

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 carry 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

Multi-Input Wake Up (Continued)

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.

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 ROM area and unbalanced tack situations.

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

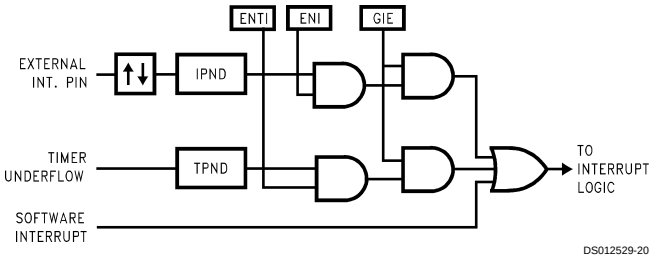


FIGURE 16. Interrupt Block Diagram

Control Registers

CNTRL1 REGISTER (ADDRESS 00EE)

TC3	TC2	TC1	TRUN	MSEL	IEDG	SL1	SL0
Bit 7							Bit 0

The Timer and MICROWIRE control register contains the following bits:

TC3 Timer T1 Mode Control Bit

TC2 Timer T1 Mode Control Bit

TC1 Timer T1 Mode Control Bit

TRUN Used to start and stop the timer/counter (1 = run, 0 = stop)

MSEL Selects G5 and G4 as MICROWIRE signals SK and SO respectively

IEDG External interrupt edge polarity select

SL1 and SL0 Select the MICROWIRE clock divide-by (00 = 2, 01 = 4, 1x = 8)

PSW REGISTER (ADDRESS 00EF)

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

The PSW register contains the following select bits:

HC Half-Carry Flip/Flop

C Carry Flip/Flop

TPND Timer T1 interrupt pending (timer Underflow or capture edge)

ENTI Timer T1 interrupt enable

IPND External interrupt pending

BUSY MICROWIRE busy shifting flag

ENI External interrupt enable

GIE Global interrupt enable (enables interrupts)

The Half-Carry bit is also effected by all the instructions that effect the Carry flag. The flag values depend upon the instruction. For example, after executing the ADC instruction the values of the Carry and the Half-Carry flag depend upon the operands involved. However, instructions like SET C and RESET C will set and clear both the carry flags. *Table *NO TARGET FOR table NS2079** lists the instructions that effect the HC and the C flags.

TABLE 9. Instructions Effecting HC and C Flags

Instr.	HC Flag	C Flag
ADC	Depends on Operands	Depends on Operands
SUBC	Depends on Operands	Depends on Operands
SET C	Set	Set
RESET C	Set	Set
RRC	Depends on Operands	Depends on Operands

Control Registers (Continued)

CNTRL2 REGISTER (ADDRESS 00CC)

MC3	MC2	MC1	CMPEN	CMPRD	CMPOE	WDUDF	Resvd
R/W	R/W	R/W	R/W	R/O	R/W	R/O	
Bit 7							Bit 0

MC3 Modulator/Timer Control Bit
MC2 Modulator/Timer Control Bit
MC1 Modulator/Timer Control Bit
CMPEN Comparator Enable Bit
CMPRD Comparator Read Bit
CMPOE Comparator Output Enable Bit
WDUDF WATCHDOG Timer Underflow Bit (Read Only)
Resvd This bit is reserved and must be zero

WDREG REGISTER (ADDRESS 00CD)

UNUSED	WDREN
Bit 7	Bit 0

WDREN WATCHDOG Reset Enable Bit (Write Once Only)

Memory Map

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

Address	Contents
00 to 2F (820CJ)	On-chip RAM bytes (48 bytes)
00 to 6F (840CJ)	On-chip RAM bytes (112 bytes)
30 to 7F (820CJ)	Unused RAM Address Space (Reads as All Ones)
70 to 7F (840CJ)	Unused RAM Address Space (Reads as All Ones)
80 to BF	Expansion Space for On-Chip EERAM (Reads Undefined Data)
C0 to C7	Reserved
C8	MIWU Edge Select Register (Reg:WKEDG)
C9	MIWU Enable Register (Reg:WKEN)
CA	MIWU Pending Register (Reg:WKPND)
CB	Reserved
CC	Control2 Register (CNTRL2)
CD	WATCHDOG Register (WDREG)
CE	WATCHDOG Counter (WDCNT)
CF	Modulator Reload (MODRL)
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 to DB	Reserved for Port C
DC	Port D Data Register
DD to DF	Reserved for Port D
E0 to EF	On-Chip Functions and Registers
E0 to E7	Reserved for Future Parts
E8	Reserved
E9	MICROWIRE Shift Register
EA	Timer Lower Byte
EB	Timer Upper Byte
EC	Timer1 Autoreload Register Lower Byte
ED	Timer1 Autoreload Register Upper Byte
EE	CNTRL1 Control Register
EF	PSW Register
F0 to FF	On-Chip RAM Mapped as Registers
FC	X Register
FD	SP Register
FE	B Register

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 for the chip. The operand is the data memory addressed by the **B** or **X** pointer.

REGISTER INDIRECT WITH AUTO POST INCREMENT OR DECREMENT

This addressing mode is used with the LD and X instructions. The operand is the data memory addressed by the **B** or **X** pointer. This is a register indirect mode that automatically post increments or post decrements the **B** or **X** pointer 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 issued with the LD B,# instruction, where the immediate # is less than 16. 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 produce the next instruction address. JP has a range from −31 to +32 to allow a one byte relative jump (JP + 1 is implemented by a NOP instruction). There are no “blocks” or “pages” when using JP since all 15 bits of the PC are used.

ABSOLUTE

Addressing Modes (Continued)

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.

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 serves as a partial address (lower 8 bits of PC) for the jump to the next instruction.

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]
MemI	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 (Continued)

INSTRUCTION SET

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

Opcode List

Bits 7–4																	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	SUBC A,#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,#i	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	

where,

i is the immediate data

Md is a directly addressed memory location

* is an unused opcode (see following table)

www.national.com

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		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			(If B < 16)
LD B,Imm				2/3			(If B > 15)
LD Mem,Imm			3/3		2/2		
LD Reg,Imm				2/3			

* = > 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

Development Tools Support

OVERVIEW

National is engaged with an international community of independent 3rd party vendors who provide hardware and software development tool support. Through National's interaction and guidance, these tools cooperate to form a choice of solutions that fits each developer's needs.

This section provides a summary of the tool and development kits currently available. Up-to-date information, selection guides, free tools, demos, updates, and purchase information can be obtained at our web site at: www.national.com/cop8.

SUMMARY OF TOOLS

COP8 Evaluation Tools

- **COP8–NSEVAL:** Free Software Evaluation package for Windows. A fully integrated evaluation environment for COP8, including versions of WCOP8 IDE (Integrated Development Environment), COP8-NSASM, COP8-MLSIM, COP8C, DriveWay™ COP8, Manuals, and other COP8 information.
- **COP8–MLSIM:** Free Instruction Level Simulator tool for Windows. For testing and debugging software instructions only (No I/O or interrupt support).
- **COP8–EPU:** Very Low cost COP8 Evaluation & Programming Unit. Windows based evaluation and hardware-simulation tool, with COP8 device programmer and erasable samples. Includes COP8-NSDEV, Driveway COP8 Demo, MetaLink Debugger, I/O cables and power supply.
- **COP8–EVAL-ICUxx:** Very Low cost evaluation and design test board for COP8ACC and COP8SGx Families, from ICU. Real-time environment with add-on A/D, D/A, and EEPROM. Includes software routines and reference designs.
- **Manuals, Applications Notes, Literature:** Available free from our web site at: www.national.com/cop8.

COP8 Integrated Software/Hardware Design Development Kits

- **COP8-EPU:** Very Low cost Evaluation & Programming Unit. Windows based development and hardware-simulation tool for COPSx/xG families, with COP8 device programmer and samples. Includes COP8-NSDEV, Driveway COP8 Demo, MetaLink Debugger, cables and power supply.
- **COP8-DM:** Moderate cost Debug Module from MetaLink. A Windows based, real-time in-circuit emulation tool with COP8 device programmer. Includes COP8-NSDEV, DriveWay COP8 Demo, MetaLink Debugger, power supply, emulation cables and adapters.

COP8 Development Languages and Environments

- **COP8-NSASM:** Free COP8 Assembler v5 for Win32. Macro assembler, linker, and librarian for COP8 software development. Supports all COP8 devices. (DOS/Win16 v4.10.2 available with limited support). (Compatible with WCOP8 IDE, COP8C, and DriveWay COP8).
- **COP8-NSDEV:** Very low cost Software Development Package for Windows. An integrated development environment for COP8, including WCOP8 IDE, COP8-NSASM, COP8-MLSIM.
- **COP8C:** Moderately priced C Cross-Compiler and Code Development System from Byte Craft (no code limit). Includes BCLIDE (Byte Craft Limited Integrated Development Environment) for Win32, editor, optimizing C Cross-Compiler, macro cross assembler, BC-Linker, and MetaLink tools support. (DOS/SUN versions available; Compiler is installable under WCOP8 IDE; Compatible with DriveWay COP8).
- **EWCOP8-KS:** Very Low cost ANSI C-Compiler and Embedded Workbench from IAR (Kickstart version: COP8Sx/Fx only with 2k code limit; No FP). A fully integrated Win32 IDE, ANSI C-Compiler, macro assembler, editor, linker, Liberian, C-Spy simulator/debugger, PLUS MetaLink EPU/DM emulator support.
- **EWCOP8-AS:** Moderately priced COP8 Assembler and Embedded Workbench from IAR (no code limit). A fully integrated Win32 IDE, macro assembler, editor, linker, librarian, and C-Spy high-level simulator/debugger with I/O and interrupts support. (Upgradeable with optional C-Compiler and/or MetaLink Debugger/Emulator support).
- **EWCOP8-BL:** Moderately priced ANSI C-Compiler and Embedded Workbench from IAR (Baseline version: All COP8 devices; 4k code limit; no FP). A fully integrated Win32 IDE, ANSI C-Compiler, macro assembler, editor, linker, librarian, and C-Spy high-level simulator/debugger. (Upgradeable; CWCOP8-M MetaLink tools interface support optional).
- **EWCOP8:** Full featured ANSI C-Compiler and Embedded Workbench for Windows from IAR (no code limit). A fully integrated Win32 IDE, ANSI C-Compiler, macro assembler, editor, linker, librarian, and C-Spy high-level simulator/debugger. (CWCOP8-M MetaLink tools interface support optional).
- **EWCOP8-M:** Full featured ANSI C-Compiler and Embedded Workbench for Windows from IAR (no code limit). A fully integrated Win32 IDE, ANSI C-Compiler, macro assembler, editor, linker, librarian, C-Spy high-level simulator/debugger, PLUS MetaLink debugger/hardware interface (CWCOP8-M).

COP8 Productivity Enhancement Tools

- **WCOP8 IDE:** Very Low cost IDE (Integrated Development Environment) from KKD. Supports COP8C, COP8-NSASM, COP8-MLSIM, DriveWay COP8, and MetaLink debugger under a common Windows Project Management environment. Code development, debug, and emulation tools can be launched from the project window framework.
- **DriveWay-COP8:** Low cost COP8 Peripherals Code Generation tool from Aisys Corporation. Automatically generates tested and documented C or Assembly source code modules containing I/O drivers and interrupt handlers for each on-chip peripheral. Application specific code can be inserted for customization using the integrated editor. (Compatible with COP8-NSASM, COP8C, and WCOP8 IDE.)
- **COP8-UTILS:** Free set of COP8 assembly code examples, device drivers, and utilities to speed up code development.
- **COP8-MLSIM:** Free Instruction Level Simulator tool for Windows. For testing and debugging software instructions only (No I/O or interrupt support).

Development Tools Support

(Continued)

COP8 Real-Time Emulation Tools

- **COP8-DM:** MetaLink Debug Module. A moderately priced real-time in-circuit emulation tool, with COP8 device programmer. Includes COP8-NSDEV, DriveWay COP8 Demo, MetaLink Debugger, power supply, emulation cables and adapters.
- **IM-COP8:** MetaLink iceMASTER®. A full featured, real-time in-circuit emulator for COP8 devices. Includes MetaLink Windows Debugger, and power supply. Package-specific probes and surface mount adaptors are ordered separately.

COP8 Device Programmer Support

- MetaLink's EPU and Debug Module include development device programming capability for COP8 devices.
- Third-party programmers and automatic handling equipment cover needs from engineering prototype and pilot production, to full production environments.
- Factory programming available for high-volume requirements.

TOOLS ORDERING NUMBERS FOR THE COP87L20CJ/COP87L40CJ FAMILY DEVICES

Vendor	Tools	Order Number	Cost	Notes
National	COP8-NSEVAL	COP8-NSEVAL	Free	Web site download
	COP8-NSASM	COP8-NSASM	Free	Included in EPU and DM. Web site download
	COP8-MLSIM	COP8-MLSIM	Free	Included in EPU and DM. Web site download
	COP8-NSDEV	COP8-NSDEV	VL	Included in EPU and DM. Order CD from website
	COP8-EPU	Not available for this device		
	COP8-DM	Contact MetaLink		
	Development Devices	COP87L20/40CJxx COP87L22/42CJxx	VL	4k or 32k OTP devices. No windowed devices
	OTP Programming Adapters	COP8SA-PGMA	L	For programming 16/20/28 SOIC and 44 PLCC on the EPU
		COP8-PGMA-44QFP	L	For programming 44QFP on any programmer
		COP8-PGMA-28CSP	L	For programming 28CSP on any programmer
		COP8-PGMA-28SO	VL	For programming 16/20/28 SOIC on any programmer
	IM-COP8	Contact MetaLink		
MetaLink	COP8-EPU	Not available for this device		
	COP8-DM	DM4-COP8-840CJ (10 MHz), plus PS-10, plus DM-COP8/xxx (ie. 28D)	M	Included p/s (PS-10), target cable of choice (DIP or PLCC; i.e. DM-COP8/28D), 16/20/28/40 DIP/SO and 44 PLCC programming sockets. Add OTP adapter (if needed) and target adapter (if needed)
	DM Target Adapters	MHW-CONVxx (xx = 33, 34 etc.)	L	DM target converters for 16DIP/20/SO/28SO/44QFP/28CSP; (MHW-CNV38 for 20 pin DIP to SO package converter)
	OTP Programming Adapters	MHW-COP8-PGMA-DS	L	For programming 16/20/28 SOIC and 44 PLCC on the EPU
	IM-COP8	IM-COP8-AD-464 (-220) (10 MHz maximum)	H	Base unit 10 MHz; -220 = 220V; add probe card (required) and target adapter (if needed); included software and manuals
		PC-840CJxxDW-AD-10 (xx = 20 or 28)	M	10 MHz 20 or 28 DIP probe card; 2.5V to 6.0V
	IM Probe Target Adapter	MHW-SOICxx (xx = 16, 20, 28)	L	16 or 20 or 28 pin SOIC adapter for probe card
ICU or National	COP8-EVAL-ICUxx	Not available for this device		
KKD	WCOP8-IDE	WCOP8-IDE	VL	Included in EPU and DM
IAR	EWCOP8-xx	See summary above	L - H	Included all software and manuals
Byte Craft	COP8C	COP8C	M	Included all software and manuals
Aisys	DriveWay COP8	DriveWay COP8	L	Included all software and manuals

Development Tools Support (Continued)

OTP Programmers	Contact vendor	L - H	For approved programmer listings and vendor information, go to our OTP support page at: www.national.com/cop8
Cost: Free; VL =< $100; L = $100 - $300; M = $300 - $1k; H = $1k - $3k; VH = $3k - $5k			

WHERE TO GET TOOLS

Tools are ordered directly from the following vendors. Please go to the vendor's web site for current listings of distributors.

Vendor	Home Office	Electronic Sites	Other Main Offices
Aisys	U.S.A.: Santa Clara, CA 1-408-327-8820 fax: 1-408-327-8830	www.aisysinc.com info@aisysinc.com	Distributors
Byte Craft	U.S.A. 1-519-888-6911 fax: 1-519-746-6751	www.bytecraft.com info@bytecraft.com	Distributors
IAR	Sweden: Uppsala +46 18 16 78 00 fax: +46 18 16 78 38	www.iar.se info@iar.se info@iar.com info@iarsys.co.uk info@iar.de	U.S.A.: San Francisco 1-415-765-5500 fax: 1-415-765-5503 U.K.: London +44 171 924 33 34 fax: +44 171 924 53 41 Germany: Munich +49 89 470 6022 fax: +49 89 470 956
ICU	Sweden: Polygonvaegen +46 8 630 11 20 fax: +46 8 630 11 70	www.icu.se support@icu.se support@icu.ch	Switzeland: Hoehe +41 34 497 28 20 fax: +41 34 497 28 21
KKD	Denmark:	www.kkd.dk	
MetaLink	U.S.A.: Chandler, AZ 1-800-638-2423 fax: 1-602-926-1198	www.metaice.com sales@metaice.com support@metaice.com bbs: 1-602-962-0013 www.metalink.de	Germany: Kirchseeon 80-91-5696-0 fax: 80-91-2386 islanger@metalink.de Distributors Worldwide
National	U.S.A.: Santa Clara, CA 1-800-272-9959 fax: 1-800-737-7018	www.national.com/cop8 support@nsc.com europe.support@nsc.com	Europe: +49 (0) 180 530 8585 fax: +49 (0) 180 530 8586 Distributors Worldwide

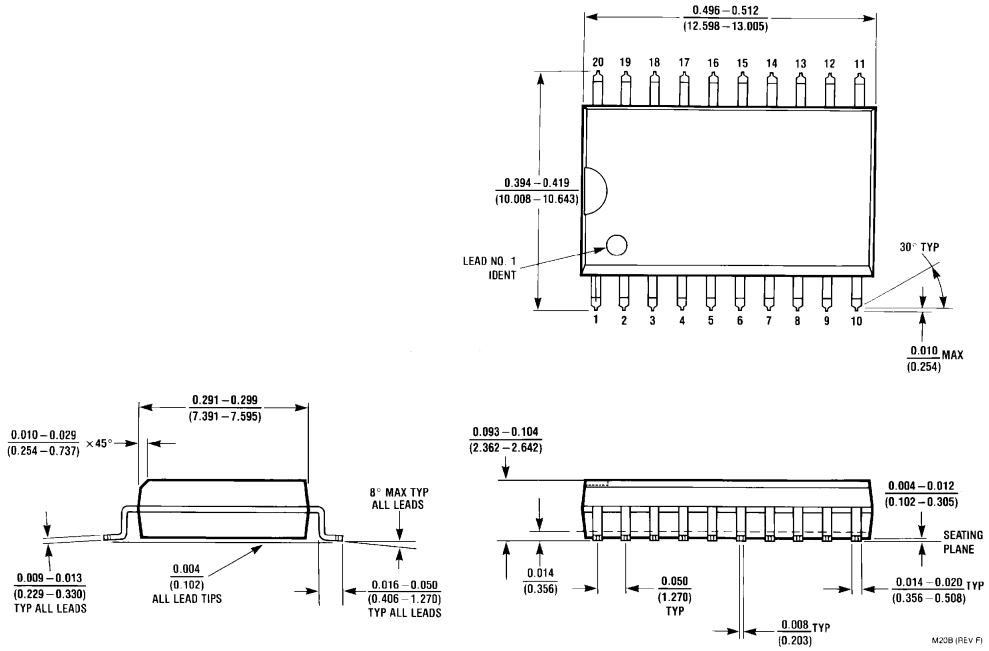
The following companies have approved COP8 programmers in a variety of configurations. Contact your local office or distributor. You can link to their web sites and get the latest listing of approved programmers from National's COP8 OTP Support page at: www.national.com/cop8.

Advantech; Advin; BP Microsystems; Data I/O; Hi-Lo Systems; ICE Technology; Lloyd Research; Logical Devices; MQP; Needhams; Phyton; SMS; Stag Programmers; System General; Tribal Microsystems; Xeltek.

Customer Support

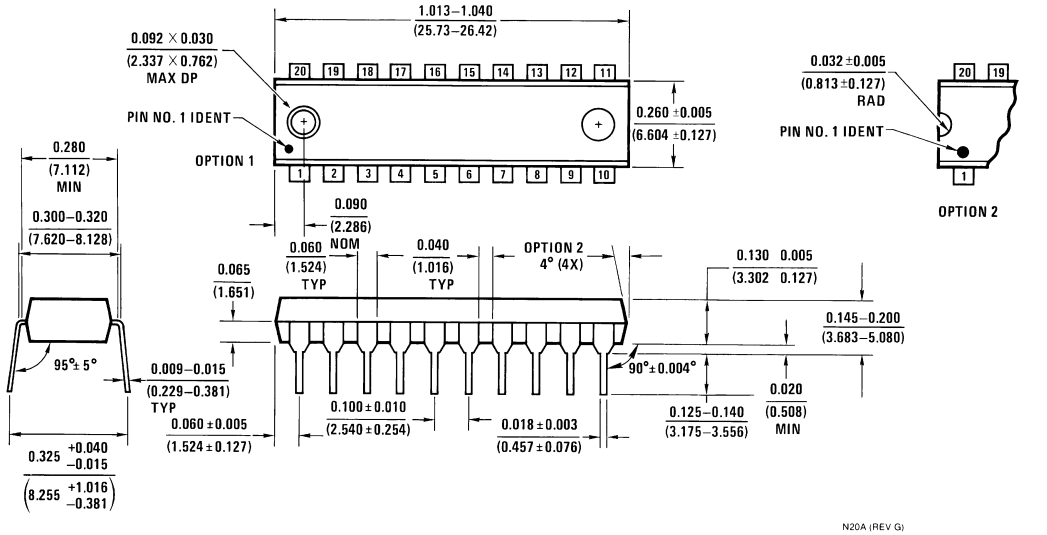
Complete product information and technical support is available from National's customer response centers, and from our on-line COP8 customer support sites.

Physical Dimensions inches (millimeters) unless otherwise noted

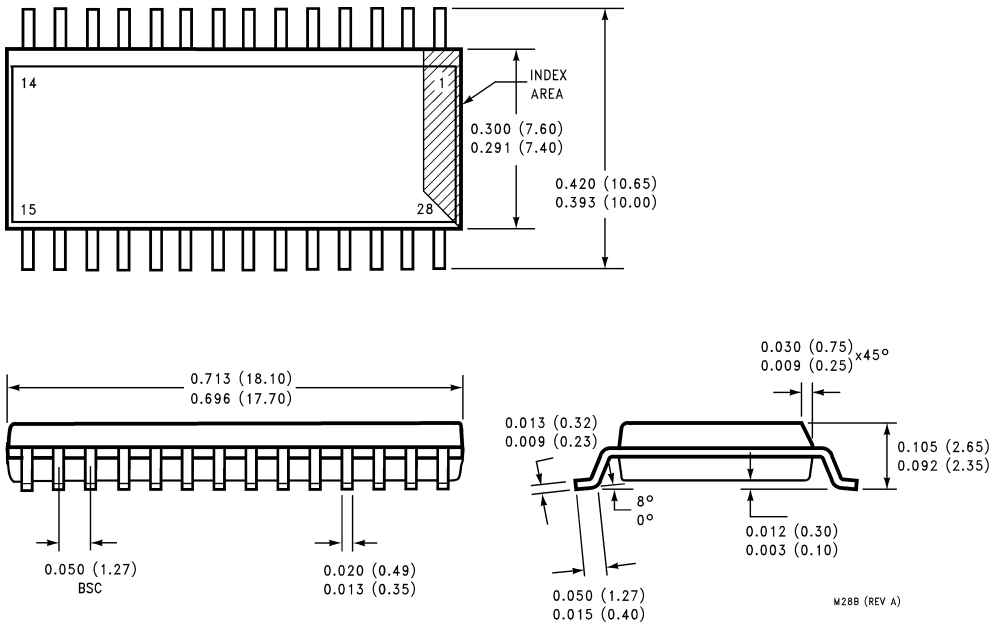


20-Lead Surface Mount Package (M)
Order Number COP87L22CJM (-1B, -1N, -2B, -2N, -3B, -3N), or
Order Number COP87L42CJM (-1B, -1N, -2B, -2N, -3B, -3N), or
Order Number COP87L42RJM (-1B, -1N, -2B, -2N, -3B, -3N)
NS Package Number M20B

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

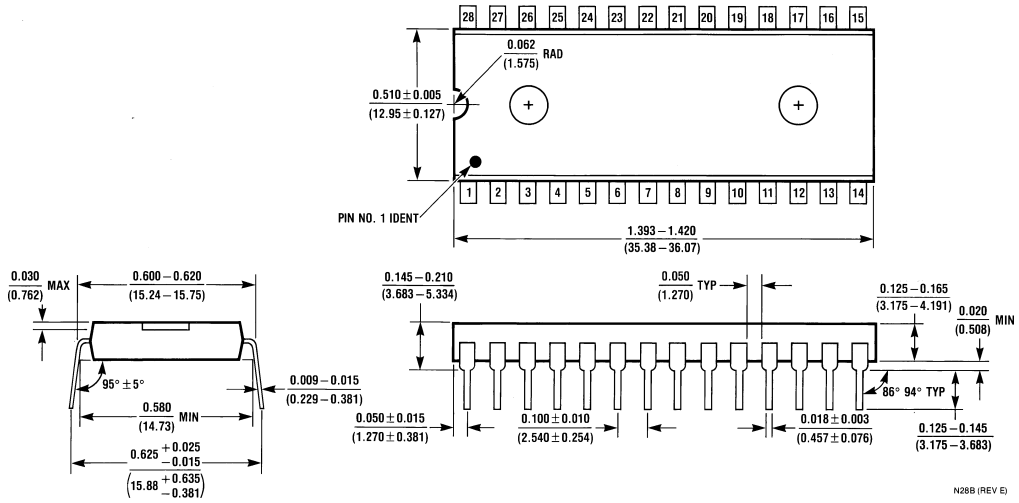


20-Lead Molded Dual-In-Line Package (N)
Order Number COP87L22CJN (-1N, -2N, -3N), or
Order Number COP87L42CJN (-1N, -2N, -3N), or
Order Number COP87L42RJN (-1N, -2N, -3N)
NS Package Number N20A



28-Lead Molded Dual-In-Line Package (M)
Order Number COP87L20CJM (-1N, -2N, -3N), or
Order Number COP87L40CJM (-1N, -2N, -3N), or
Order Number COP87L40RJM (-1N, -2N, -3N)
NS Package Number M28B

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



28-Lead Molded Dual-In-Line Package (N)
Order Number COP87L20CJN (-1N, -2N, -3N), or
Order Number COP87L40CJN (-1N, -2N, -3N), or
Order Number COP87L40RJN (-1N, -2N, -3N)
NS Package Number N28B

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 AND GENERAL COUNSEL 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
Americas
Tel: 1-800-272-9959
Fax: 1-800-737-7018
Email: support@nsc.com
www.national.com

National Semiconductor Europe
Fax: +49 (0) 1 80-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 1 80-530 85 85
English Tel: +49 (0) 1 80-532 78 32
Français Tel: +49 (0) 1 80-532 93 58
Italiano Tel: +49 (0) 1 80-534 16 80

National Semiconductor Asia Pacific Customer Response Group
Tel: 65-2544466
Fax: 65-2504466
Email: sea.support@nsc.com

National Semiconductor Japan Ltd.
Tel: 81-3-5639-7560
Fax: 81-3-5639-7507

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.