National Semiconductor

PRELIMINARY
March 1996

COP87L88EK/COP87L84EK microCMOS One Time Programmable (OTP) Microcontrollers

General Description

The COP87L88EK/COP87L84EK programmable microcontrollers are members of the COP8™ 8-bit OTP microcontroller family. It is pin and software compatible to the mask ROM COP888EK/COP884EK product family. (Continued)

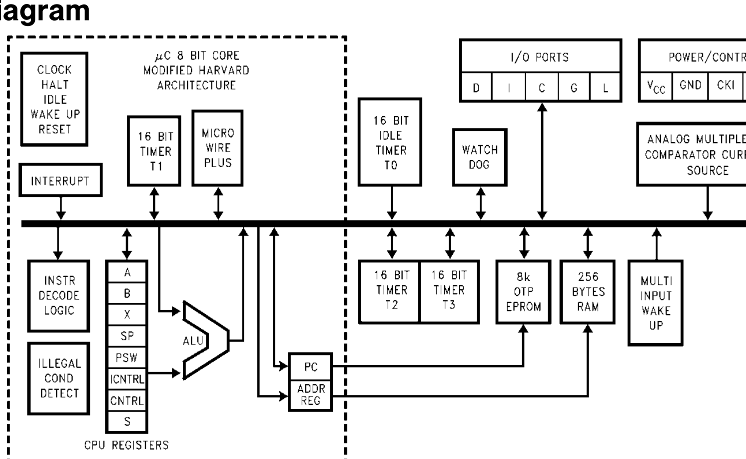
Features

- Low cost 8-bit microcontroller
- Fully static CMOS, with low current drain
- Two power saving modes: HALT and IDLE
- 1 μs instruction cycle time
- 8 kbytes on-board EPROM with security feature

Note: Up to 32 kbytes of OTP EPROM is available on request

- 256 bytes on-board RAM
- Single supply operation: 2.7V–5.5V
- Analog function block with
  - — Analog comparator with seven input multiplexor
  - — Constant current source and $V_{CC}/2$ reference
- MICROWIRE/PLUS™ serial I/O
- WATCHDOG™ and Clock Monitor logic
- Idle Timer
- Multi-Input Wakeup (MIWU) with optional interrupts (8)
- Three 16-bit timers, each with two 16-bit registers supporting:
  - — Processor Independent PWM mode
  - — External Event counter mode
  - — Input Capture mode
- 8-bit Stack Pointer SP (stack in RAM)
- Multi-Input Wakeup (MIWU) with optional interrupts (8)
- Two 8-bit Register Indirect Data Memory Pointers (B and X)
- Twelve multi-source vectored interrupts servicing
  - — External Interrupt
  - — Idle Timer T0
  - — Three Timers (Each with 2 Interrupts)
  - — MICROWIRE/PLUS
  - — Multi-Input Wake Up
  - — Software Trap
  - — Default VIS
- Versatile instruction set
- True bit manipulation
- Memory mapped I/O
- BCD arithmetic instructions
- Package:
  - — 44 PLCC with 39 I/O pins
  - — 40 DIP with 35 I/O pins
  - — 28 SO or 28 DIP, each with 23 I/O pins
- Software selectable I/O options
  - — TRI-STATE® Output
  - — Push-Pull Output
  - — Weak Pull Up Input
  - — High Impedance Input
- Schmitt trigger inputs on ports G and L
- Temperature range: −40°C to +85°C
- Emulation device for COP888EK/COP884EK
- Real time emulation and full program debug offered by MetaLink's Development System

Block Diagram



TL/DD/12520–1

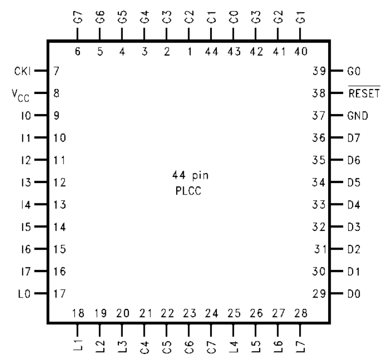
FIGURE 1. Block Diagram

General Description (Continued)

It is a fully static part, fabricated using double-metal silicon gate microCMOS technology. The device is available as One-Time Programmable (OTP). Features include an 8-bit memory mapped architecture, MICROWIRE/PLUS serial I/O, three 16-bit timer/counters supporting three modes (Processor Independent PWM generation, External Event counter, and Input Capture mode capabilities), one analog comparator with seven input multiplexor, and two power saving modes (HALT and IDLE), both with a multi-sourced wakeup/interrupt capability. This multi-sourced interrupt capability may also be used independent of the HALT or IDLE modes. Each I/O pin has software selectable configurations. The devices operate over a voltage range of 2.7V to 5.5V. High throughput is achieved with an efficient, regular instruction set operating at a maximum rate of 1 μs per instruction.

Connection Diagrams

Plastic Chip Carrier

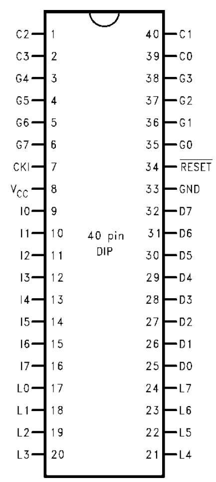


TL/DD/12520–2

Top View

Order Number COP87L88EKV-XE
See NS Plastic Chip Package Number V44A

Dual-In-Line Package

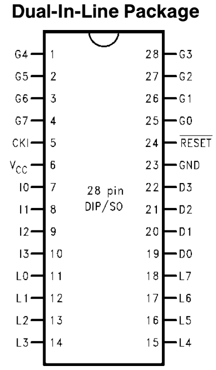


TL/DD/12520–3

Top View

Order Number COP87L84EKN-XE
See NS Molded Package Number N40A

Dual-In-Line Package



TL/DD/12520–4

Top View

Note: -X Crystal Oscillator
-E Halt Enable

Order Number COP87L84EKN-XE
See NS Molded Package Number N28B
Order Number COP87L84EKM-XE
See NS Molded Package Number M28B

FIGURE 2. Connection Diagrams

Connection Diagrams (Continued)

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

Port	Type	Alt. Fun	Alt. Fun	28-Pin Pack.	40-Pin Pack.	44-Pin Pack.
L0	I/O	MIWU		11	17	17
L1	I/O	MIWU		12	18	18
L2	I/O	MIWU		13	19	19
L3	I/O	MIWU		14	20	20
L4	I/O	MIWU	T2A	15	21	25
L5	I/O	MIWU	T2B	16	22	26
L6	I/O	MIWU	T3A	17	23	27
L7	I/O	MIWU	T3B	18	24	28
G0	I/O	INT		25	35	39
G1	WDOUT			26	36	40
G2	I/O	T1B		27	37	41
G3	I/O	T1A		28	38	42
G4	I/O	SO		1	3	3
G5	I/O	SK		2	4	4
G6	I	SI		3	5	5
G7	I/CKO	HALT Restart		4	6	6
D0	O			19	25	29
D1	O			20	26	30
D2	O			21	27	31
D3	O			22	28	32
I0	I	COMPIN1+		7	9	9
I1	I	COMPIN−/Current Source Out		8	10	10
I2	I	COMPIN0+		9	11	11
I3	I	COMPOUT/COMPIN2+		10	12	12
I4	I	COMPIN3+			13	13
I5	I	COMPIN4+			14	14
I6	I	COMPIN5+			15	15
I7	I	COMPOUT			16	16
D4	O				29	33
D5	O				30	34
D6	O				31	35
D7	O				32	36
C0	I/O				39	43
C1	I/O				40	44
C2	I/O				1	1
C3	I/O				2	2
C4	I/O					21
C5	I/O					22
C6	I/O					23
C7	I/O					24
V_{CC}				6	8	8
GND				23	33	37
CKI				5	7	7
$\overline{\text{RESET}}$				24	34	38

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
Voltage at Any Pin	−0.3V to V_{CC} + 0.3V
Total Current into V_{CC} Pin (Source)	100 mA
Total Current out of GND Pin (Sink)	110 mA
Storage Temperature Range	−65°C to +140°C

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

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

Parameter	Conditions	Min	Typ	Max	Units
Operating Voltage		2.7		5.5	V
Power Supply Ripple (Note 1)	Peak-to-Peak			0.1 V_{CC}	V
Supply Current (Note 2)					
CKI = 10 MHz	V_{CC} = 5.5V, t_c = 1 µs			16.5	mA
CKI = 4 MHz	V_{CC} = 4.0V, t_c = 2.5 µs			6.5	mA
HALT Current (Note 3)	V_{CC} = 5.5V, CKI = 0 MHz			12	µA
	V_{CC} = 4.0V, CKI = 0 MHz			8	µA
IDLE Current (Note 2)					
CKI = 10 MHz	V_{CC} = 5.5V, t_c = 1 µs			3.5	mA
CKI = 1 MHz	V_{CC} = 4.0V, t_c = 10 µs			0.7	mA
Input Levels (V_{IH}, V_{IL})					
$\overline{\text{RESET}}$					
Logic High		0.8 V_{CC}			V
Logic Low				0.2 V_{CC}	V
CKI, 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, V_{IN} = 0V	−40		−250	µA
G and L Port Input Hysteresis (Note 7)				0.35 V_{CC}	V
Output Current Levels					
D Outputs					
Source	V_{CC} = 4.5V, V_{OH} = 3.3V	−0.4			mA
Sink (Note 4)	V_{CC} = 4.5V, V_{OL} = 1V	10			mA
All Others					
Source (Weak Pull-Up Mode)	V_{CC} = 4.5V, V_{OH} = 2.7V	−10		−110	µA
Source (Push-Pull Mode)	V_{CC} = 4.5V, V_{OH} = 3.3V	−0.4			mA
Sink (Push-Pull Mode)	V_{CC} = 4.5V, V_{OL} = 0.4V	1.6			mA
TRI-STATE Leakage	V_{CC} = 5.5V	−2		+2	µA
Allowable Sink/Source Current per Pin (Note 7)					
D Outputs (Sink)				15	mA
All others				3	mA
Maximum Input Current without Latchup (Note 5)	Room Temp			±200	mA
RAM Retention Voltage, V_r	500 ns Rise and Fall Time (Min)	2			V
Input Capacitance (Note 6)				7	pF
Load Capacitance on D2 (Note 6)				1000	pF

AC Electrical Characteristics $-40°C \leq T_A \leq +85°C$ unless otherwise specified

Parameter	Conditions	Min	Typ	Max	Units
Instruction Cycle Time (t_c)					
Crystal, Resonator,	$4.5V \leq V_{CC} \leq 5.5V$	1.0		DC	μs
R/C Oscillator	$4.5V \leq V_{CC} \leq 5.5V$	3.0		DC	μs
Inputs					
t_{SETUP}	$4.5V \leq V_{CC} \leq 5.5V$	200			ns
t_{HOLD}	$4.5V \leq V_{CC} \leq 5.5V$	60			ns
Output Propagation Delay (Note 6)	$R_L = 2.2k, C_L = 100$ pF				
t_{PD1}, t_{PD0}					
SO, SK	$4.5V \leq V_{CC} \leq 5.5V$			0.7	μs
All Others	$4.5V \leq V_{CC} \leq 5.5V$			1	μs
MICROWIRE™ Setup Time (t_{UWS}) (Note 7)	$V_{CC} \geq 4.5V$	20			ns
MICROWIRE Hold Time (t_{UWH}) (Note 7)	$V_{CC} \geq 4.5V$	56			ns
MICROWIRE Output Propagation Delay (t_{UPD})	$V_{CC} \geq 4.5V$			220	ns
Input Pulse Width (Note 7)					
Interrupt Input High Time		1.0			t_c
Interrupt Input Low Time		1.0			t_c
Timer 1, 2, 3 Input High Time		1.0			t_c
Timer 1, 2, 3 Input Low Time		1.0			t_c
Reset Pulse Width		1.0			μs

t_c = Instruction Cycle Time

Note 1: Maximum rate of voltage change must be < 0.5 V/ms.

Note 2: Supply and IDLE currents are measured with CKI driven with a square wave Oscillator, CKO driven 180° out of phase with CKI, inputs connected to V_{CC} and outputs driven low but not connected to a load.

Note 3: The HALT mode will stop CKI from oscillating in the RC and the Crystal configurations by bringing CKI high. Measurement of I_{DD} HALT is done with device neither sourcing nor sinking current; with L, C, and G0–G5 programmed as low outputs and not driving a load; all outputs programmed low and not driving a load; all inputs tied to V_{CC}; clock monitor and comparator disabled. Parameter refers to HALT mode entered via setting bit 7 of the G Port data register. Part will pull up CKI during HALT in crystal clock mode.

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

Note 5: Pins G6 and $\overline{RESET}$ are designed with a high voltage input network. These pins allow input voltages > V_{CC} and the pins will have sink current to V_{CC} when biased at voltages > 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 < 14V. **WARNING: Voltages in excess of 14V will cause damage to the pins. This warning excludes ESD transients.**

Note 6: The output propagation delay is referenced to the end of the instruction cycle where the output change occurs.

Note 7: Parameter characterized but not tested.

Analog Function Block $V_{CC} = 5.0V, -40°C \leq T_A \leq +85°C$

Parameter	Conditions	Min	Typ	Max	Units
Input Offset Voltage	$0.4V < V_{IN} < V_{CC} - 1.5V$		±10	±25	mV
Input Common Mode Voltage Range (Note 8)		0.4		$V_{CC} - 1.5$	V
$V_{CC}/2$ Reference	$4.5V < V_{CC} < 5.5V$	$0.5\ V_{CC} - 0.04$	$0.5\ V_{CC}$	$0.5\ V_{CC} + 0.04$	V
DC Supply Current for Comparator (when enabled)	$V_{CC} = 5.5V$			250	μA
DC Supply Current for $V_{CC}/2$ Reference (when enabled)	$V_{CC} = 5.5V$		50	80	μA
DC Supply Current for Constant Current Source (when enabled)	$V_{CC} = 5.5V$			200	μA
Constant Current Source	$4.5V < V_{CC} < 5.5V$	10	20	40	μA
Current Source Variation over Common Mode Range	$4.5V < V_{CC} < 5.5V$ Temp = Constant			±2	μA
Current Source Enable Time			1.5	2	μs
Comparator Response Time	100 mV Overdrive, 100 pF Load			1	μs

Note 8: The device is capable of operating over a common mode voltage range of 0 to $V_{CC} - 1.5V$, however increased offset voltage will be observed between 0V and 0.4V.

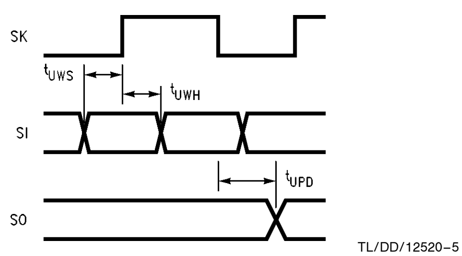


FIGURE 3. MICROWIRE/PLUS Timing

Pin Descriptions

V_{CC} and GND are the power supply pins. All V_{CC} and GND pins must be connected.

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

$\overline{RESET}$ is the master reset input. See Reset Description section.

The device contains three bidirectional 8-bit I/O ports (C, G and L), where each individual bit may be independently configured as an input (Schmitt Trigger inputs on ports L and G), output or TRI-STATE under program control. Three data memory address locations are allocated for each of these I/O ports. Each I/O port has two associated 8-bit memory mapped registers, the CONFIGURATION register and the output DATA register. A memory mapped address is also reserved for the input pins of each I/O port. (See the memory map for the various addresses associated with the I/O ports.) *Figure 4* shows the I/O port configurations. The DATA and CONFIGURATION registers allow for each port bit to be individually configured under software control as shown below:

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

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

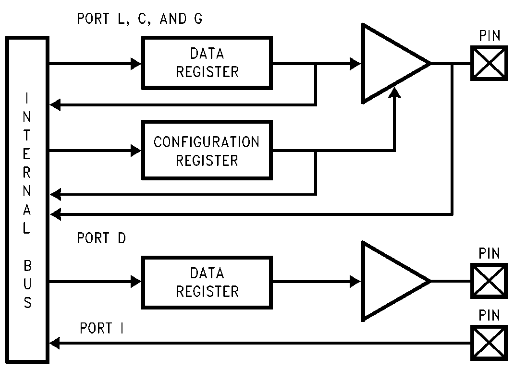
The Port L supports Multi-Input Wake Up on all eight pins. L4 and L5 are used for the timer input functions T2A and T2B. L6 and L7 are used for the timer input functions T3A and T3B.

The Port L has the following alternate features:

L0	MIWU
L1	MIWU
L2	MIWU
L3	MIWU
L4	MIWU or T2A
L5	MIWU or T2B
L6	MIWU or T3A
L7	MIWU or T3B

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

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



TL/DD/12520–6

FIGURE 4. I/O Port Configurations

Pin Descriptions (Continued)

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

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

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

Port G has the following alternate features:

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

Port G has the following dedicated functions:

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

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

Port I is an eight-bit Hi-Z input port.

Port I0–I7 are used for the analog function block.

The Port I has the following alternate features:

- I0 COMPIN1+ (Comparator Positive Input 1)
- I1 COMPIN− (Comparator Negative Input/Current Source Out)
- I2 COMPIN0+ (Comparator Positive Input 0)
- I3 COMPOUT/COMPIN2+ (Comparator Output/Comparator Positive Input 2))
- I4 COMPIN3+ (Comparator Positive Input 3)
- I5 COMPIN4+ (Comparator Positive Input 4)
- I6 COMPIN5+ (Comparator Positive Input 5)
- I7 COMPOUT (Comparator Output)

Port D is an 8-bit output port that is preset high when $\overline{\text{RESET}}$ goes low. The user can tie two or more D port outputs (except D2) 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 <1000 pF.

Functional Description

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

CPU REGISTERS

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

There are six CPU registers:

A is the 8-bit Accumulator Register

PC is the 15-bit Program Counter Register

PU is the upper 7 bits of the program counter (PC)
PL is the lower 8 bits of the program counter (PC)

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

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

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

S is the 8-bit Data Segment Address Register used to extend the lower half of the address range (00 to 7F) into 256 data segments of 128 bytes each.

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

PROGRAM MEMORY

The program memory consists of 8 kbytes of OTP EP ROM. These bytes may hold program instructions or constant data (data tables for the LAID instruction, jump vectors for the JID instruction, and interrupt vectors for the VIS instruction). The program memory is addressed by the 15-bit program counter (PC). All interrupts in the devices vector to program memory location 0FF Hex.

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

Note: Up to 32 kbytes of OTP EPROM is available upon request.

SECURITY FEATURE

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

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

DATA MEMORY

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

The data memory consists of 256 bytes of RAM. Sixteen bytes of RAM are mapped as "registers" at addresses 0F0 to 0FF Hex. These registers can be loaded immediately, and also decremented and tested with the DRSZ (decrement register and skip if zero) instruction. The memory pointer registers X, SP, B and S are memory mapped into this space at address locations 0FC to 0FF Hex respectively, with the other registers being available for general usage.

Functional Description (Continued)

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

Note: RAM contents are undefined upon power-up.

Data Memory Segment RAM Extension

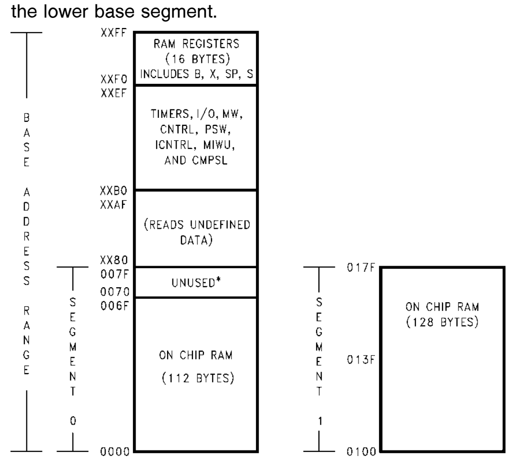
Data memory address 0FF is used as a memory mapped location for the Data Segment Address Register (S).

The data store memory is either addressed directly by a single byte address within the instruction, or indirectly relative to the reference of the B, X, or SP pointers (each contains a single-byte address). This single-byte address allows an addressing range of 256 locations from 00 to FF hex. The upper bit of this single-byte address divides the data store memory into two separate sections as outlined previously. With the exception of the RAM register memory from address locations 00F0 to 00FF, all RAM memory is memory mapped with the upper bit of the single-byte address being equal to zero. This allows the upper bit of the single-byte address to determine whether or not the base address range (from 0000 to 00FF) is extended. If this upper bit equals one (representing address range 0080 to 00FF), then address extension does not take place. Alternatively, if this upper bit equals zero, then the data segment extension register S is used to extend the base address range (from 0000 to 007F) from XX00 to XX7F, where XX represents the 8 bits from the S register. Thus the 128-byte data segment extensions are located from addresses 0100 to 017F for data segment 1, 0200 to 027F for data segment 2, etc., up to FF00 to FF7F for data segment 255. The base address range from 0000 to 007F represents data segment 0.

Figure 5 illustrates how the S register data memory extension is used in extending the lower half of the base address range (00 to 7F hex) into 256 data segments of 128 bytes each, with a total addressing range of 32 kbytes from XX00 to XX7F. This organization allows a total of 256 data segments of 128 bytes each with an additional upper base segment of 128 bytes. Furthermore, all addressing modes are available for all data segments. The S register must be changed under program control to move from one data segment (128 bytes) to another. However, the upper base segment (containing the 16 memory registers, I/O registers, control registers, etc.) is always available regardless of the contents of the S register, since the upper base segment (address range 0080 to 00FF) is independent of data segment extension.

The instructions that utilize the stack pointer (SP) always reference the stack as part of the base segment (Segment 0), regardless of the contents of the S register. The S register is not changed by these instructions. Consequently, the stack (used with subroutine linkage and interrupts) is always located in the base segment. The stack pointer will be intitialized to point at data memory location 006F as a result of reset.

The 128 bytes of RAM contained in the base segment are split between the lower and upper base segments. The first 112 bytes of RAM are resident from address 0000 to 006F in the lower base segment, while the remaining 16 bytes of RAM represent the 16 data memory registers located at addresses 00F0 to 00FF of the upper base segment. No RAM is located at the upper sixteen addresses (0070 to 007F) of the lower base segment.



TL/DD/12520–7

*Reads as all ones.

FIGURE 5. RAM Organization

Additional RAM beyond these initial 128 bytes, however, will always be memory mapped in groups of 128 bytes (or less) at the data segment address extensions (XX00 to XX7F) of the lower base segment. The additional 128 bytes of RAM are memory mapped at address locations 0100 to 017F hex.

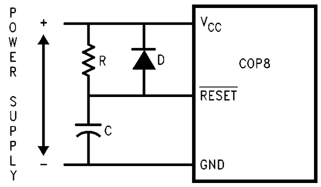
Reset

The $\overline{\text{RESET}}$ input when pulled low initializes the microcontroller. Initialization will occur whenever the $\overline{\text{RESET}}$ input is pulled low. Upon initialization, the data and configuration registers for ports L, G and C are cleared, resulting in these Ports being initialized to the TRI-STATE mode. Pin G1 of the G Port is an exception (as noted below) since pin G1 is dedicated as the WATCHDOG and/or Clock Monitor error output pin. Port D is set high. The PC, PSW, ICNTRL, CNTRL, T2CNTRL and T3CNTRL control registers are cleared. The Comparator Select Register is cleared. The S register is initialized to zero. The Multi-Input Wakeup registers WKEN and WKEDG are cleared. Wakeup register WKPND is unknown. The stack pointer, SP, is initialized to 6F hex.

The device comes out of reset with both the WATCHDOG logic and the Clock Monitor detector armed, with the WATCHDOG service window bits set and the Clock Monitor bit set. The WATCHDOG and Clock Monitor circuits are inhibited during reset. The WATCHDOG service window bits being initialized high default to the maximum WATCHDOG service window of 64k t_C clock cycles. The Clock Monitor bit being initialized high will cause a Clock Monitor error following reset if the clock has not reached the minimum specified frequency at the termination of reset. A Clock Monitor error will cause an active low error output on pin G1. This error output will continue until 16 t_C–32 t_C clock cycles following the clock frequency reaching the minimum specified value, at which time the G1 output will enter the TRI-STATE mode.

The external RC network shown in *Figure 6* should be used to ensure that the $\overline{\text{RESET}}$ pin is held low until the power supply to the chip stabilizes.

Reset (Continued)



TL/DD/12520–8

RC > 5 × Power Supply Rise Time

FIGURE 6. Recommended Reset Circuit

Oscillator Circuits

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

Figure 7 shows the Crystal and R/C oscillator diagrams.

CRYSTAL OSCILLATOR

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

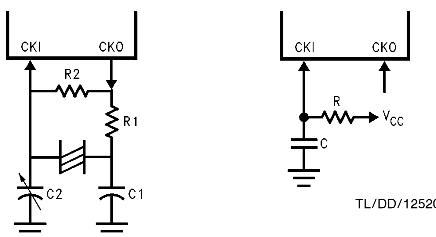
Table A shows the component values required for various standard crystal values.

R/C OSCILLATOR

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

Note: Use of the R/C oscillator option will result in higher electromagnetic emissions.

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



TL/DD/12520–9 TL/DD/12520–10

FIGURE 7. Crystal and R/C Oscillator Diagrams

TABLE A. Crystal Oscillator Configuration, $T_A = 25°C$

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	$V_{CC} = 5V$
0	1	200	100–150	0.455	$V_{CC} = 5V$

TABLE B. RC Oscillator Configuration, $T_A = 25°C$

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

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

$50\ pF \le C \le 200\ pF$

Current Drain

The total current drain of the chip depends on:

1. Oscillator operation mode—I1
2. Internal switching current—I2
3. Internal leakage current—I3
4. Output source current—I4
5. DC current caused by external input not at V_{CC} or GND—I5
6. Comparator DC supply current when enabled—I6
7. Clock Monitor current when enabled—I7

Thus the total current drain, It, is given as:

$$It = I1 + I2 + I3 + I4 + I5 + I6 + I7$$

To reduce the total current drain, each of the above components must be minimum.

The chip will draw more current as the CKI input frequency increases up to the maximum 10 MHz value. Operating with a crystal network will draw more current than an external square-wave. Switching current, governed by the equation below, can be reduced by lowering voltage and frequency. Leakage current can be reduced by lowering voltage and temperature. The other two items can be reduced by carefully designing the end-user's system.

$$I2 = C \times V \times f$$

where C = equivalent capacitance of the chip
V = operating voltage
f = CKI frequency

Control Registers

CNTRL Register (Address X′00EE)

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

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

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

Control Registers (Continued)

PSW Register (Address X′00EF)

The PSW register contains the following select bits:

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

HC	C	T1PNDA	T1ENA	EXPND	BUSY	EXEN	GIE

Bit 7 Bit 0

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

ICNTRL Register (Address X′00E8)

The ICNTRL register contains the following bits:

T1ENB Timer T1 Interrupt Enable for T1B Input capture edge
T1PNDB Timer T1 Interrupt Pending Flag for T1B capture edge
μWEN Enable MICROWIRE/PLUS interrupt
μWPND MICROWIRE/PLUS interrupt pending
T0EN Timer T0 Interrupt Enable (Bit 12 toggle)
T0PND Timer T0 Interrupt pending
LPEN L Port Interrupt Enable (Multi-Input Wakeup/Interrupt)
Bit 7 could be used as a flag

Unused	LPEN	T0PND	T0EN	μWPND	μWEN	T1PNDB	T1ENB

Bit 7 Bit 0

T2CNTRL Register (Address X′00C6)

The T2CNTRL register contains the following bits:

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

T2C3	T2C2	T2C1	T2C0	T2PNDA	T2ENA	T2PNDB	T2ENB

Bit 7 Bit 0

T3CNTRL Register (Address X′00B6)

The T3CNTRL register contains the following bits:

T3ENB Timer T3 Interrupt Enable for T3B
T3PNDB Timer T3 Interrupt Pending Flag for T3B pin (T3B capture edge)
T3ENA Timer T3 Interrupt Enable for Timer Underflow or T3A pin
T3PNDA Timer T3 Interrupt Pending Flag (Autoload RA in mode 1, T3 Underflow in mode 2, T3a capture edge in mode 3)
T3C0 Timer T3 Start/Stop control in timer modes 1 and 2
Timer T3 Underflow Interrupt Pending Flag in timer mode 3
T3C1 Timer T3 mode control bit
T3C2 Timer T3 mode control bit
T3C3 Timer T3 mode control bit

T3C3	T3C2	T3C1	T3C0	T3PNDA	T3ENA	T3PNDB	T3ENB

Bit 7 Bit 0

Timers

The device contains a very versatile set of timers (T0, T1, T2, T3). All timers and associated autoreload/capture registers power up containing random data.

TIMER T0 (IDLE TIMER)

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

The Timer T0 supports the following functions:

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

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

Timers (Continued)

TIMER T1, TIMER T2 AND TIMER T3

The device has a set of three powerful timer/counter blocks, T1, T2 and T3. The associated features and functioning of a timer block are described by referring to the timer block Tx. Since the three timer blocks, T1, T2 and T3 are identical, all comments are equally applicable to any of the three timer blocks.

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

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

Mode 1. Processor Independent PWM Mode

As the name suggests, this mode allows the device to generate a PWM signal with very minimal user intervention. The user only has to define the parameters of the PWM signal (ON time and OFF time). Once begun, the timer block will continuously generate the PWM signal completely independent of the microcontroller. The user software services the timer block only when the PWM parameters require updating.

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

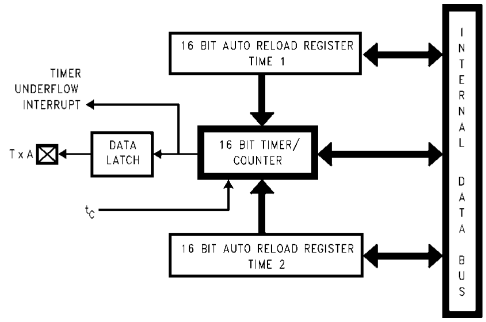
The Tx Timer control bits, TxC3, TxC2 and TxC1 set up the timer for PWM mode operation.

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

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

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

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



TL/DD/12520–11

FIGURE 8. Timer in PWM Mode

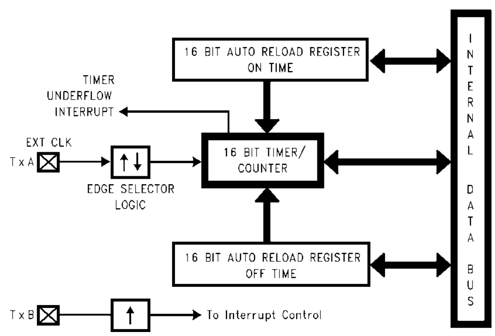
Mode 2. External Event Counter Mode

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

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

Figure 9 shows a block diagram of the timer in External Event Counter mode.

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



TL/DD/12520–12

FIGURE 9. Timer in External Event Counter Mode

Mode 3. Input Capture Mode

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

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

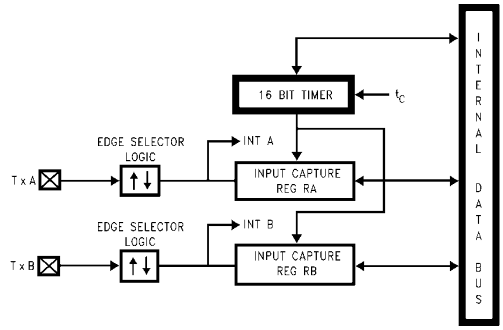
Timers (Continued)

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

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

Underflows from the timer can also be programmed to generate interrupts. Underflows are latched into the timer TxC0 pending flag (the TxC0 control bit serves as the timer underflow interrupt pending flag in the Input Capture mode). Consequently, the TxC0 control bit should be reset when entering the Input Capture mode. The timer underflow interrupt is enabled with the TxENA control flag. When a TxA interrupt occurs in the Input Capture mode, the user must check both the TxPNDA and TxC0 pending flags in order to determine whether a TxA input capture or a timer underflow (or both) caused the interrupt.

Figure 10 shows a block diagram of the timer in Input Capture mode.



TL/DD/12520–13

FIGURE 10. Timer in Input Capture Mode

TIMER CONTROL FLAGS

The timers T1, T2 and T3 have indentical control structures. The control bits and their functions are summarized below.

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

Timers (Continued)

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

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

Power Save Modes

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

HALT MODE

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

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

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

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

Power Save Modes (Continued)

The device has two mask options associated with the HALT mode. The first mask option enables the HALT mode feature, while the second mask option disables the HALT mode. With the HALT mode enable mask option, the device will enter and exit the HALT mode as described above. With the HALT disable mask option, the device cannot be placed in the HALT mode (writing a "1" to the HALT flag will have no effect, the HALT flag will remain "0").

IDLE MODE

The device is placed in the IDLE mode by writing a "1" to the IDLE flag (G6 data bit). In this mode, all activities, except the associated on-board oscillator circuitry, the WATCHDOG logic, the clock monitor and the IDLE Timer T0, are stopped.

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

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

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

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

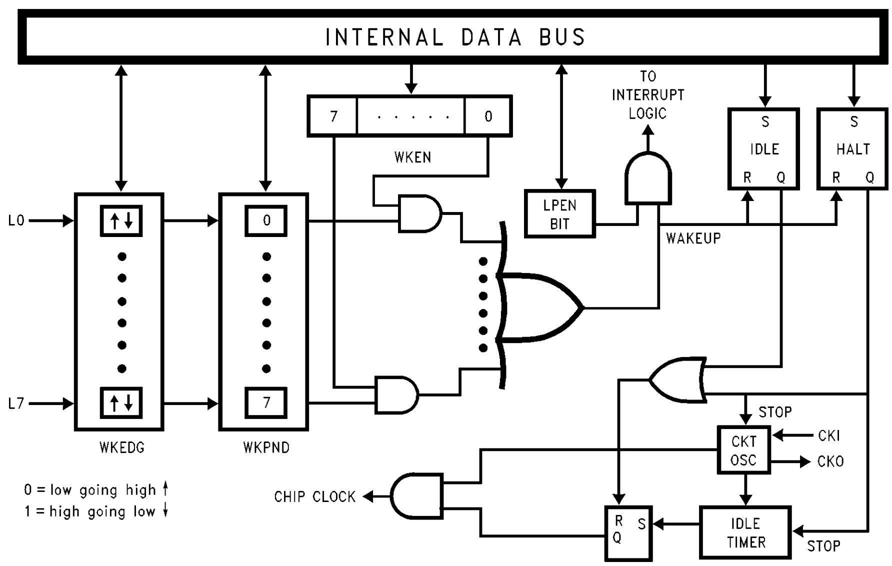
Alternatively, the user can enter the IDLE mode with the IDLE Timer T0 interrupt disabled. In this case, the device will resume normal operation with the instruction immediately following the "Enter IDLE Mode" instruction.

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

Multi-Input Wakeup

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

Figure 11 shows the Multi-Input Wakeup logic.



TL/DD/12520–14

FIGURE 11. Multi-Input Wake Up Logic

Multi-Input Wakeup (Continued)

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

The user can select whether the trigger condition on the selected L Port pin is going to be either a positive edge (low to high transition) or a negative edge (high to low transition). This selection is made via the Reg: WKEDG, which is an 8-bit control register with a bit assigned to each L Port pin. Setting the control bit will select the trigger condition to be a negative edge on that particular L Port pin. Resetting the bit selects the trigger condition to be a positive edge. Changing an edge select entails several steps in order to avoid a pseudo 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 interrupt. The program would be as follows:

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

If the L port bits have been used as outputs and then changed to inputs with Multi-Input Wakeup/Interrupt, 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 WKPND. The respective bits of the WKPND register will be set on the occurrence of the selected trigger edge on the corresponding Port L pin. The user has the responsibility of clearing these pending flags. Since WKPND is a pending register for the occurrence of selected wakeup conditions, the device will not enter the HALT mode if any Wakeup bit is both enabled and pending. Consequently, the user has the responsibility of clearing the pending flags before attempting to enter the HALT mode.

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

PORT L INTERRUPTS

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

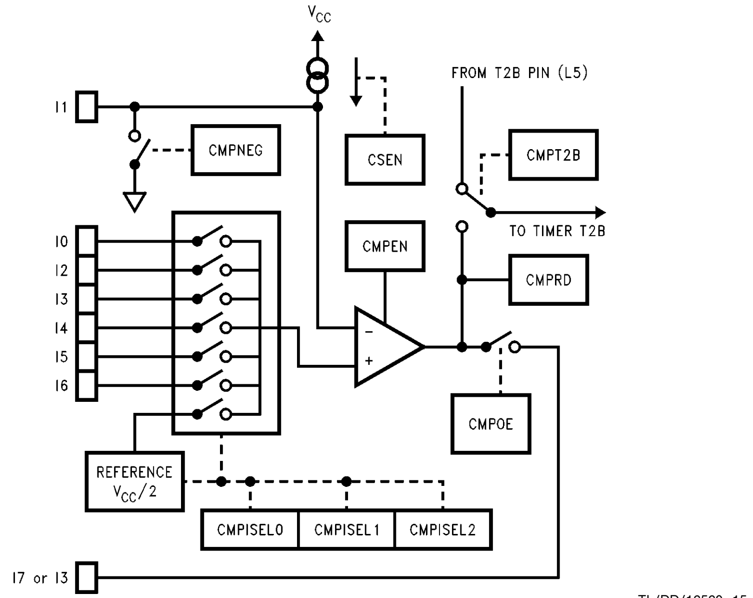
The interrupt from Port L shares logic with the wake up circuitry. The register WKEN allows interrupts from Port L to be individually enabled or disabled. The register WKEDG specifies the trigger condition to be either a positive or a negative edge. Finally, the register WKPND latches in the pending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

The GIE (Global Interrupt Enable) bit enables the interrupt function.

A control flag, LPEN, functions as a global interrupt enable for Port L interrupts. Setting the LPEN flag will enable interrupts and vice versa. A separate global pending flag is not needed since the register WKPND is adequate.

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

Analog Function Block



TL/DD/12520–15

FIGURE 12. COP87L88EK Analog Function Block

This device contains an analog function block with the intent to provide a function which allows for single slope, low cost, A/D conversion of up to 6 channels.

CMPSL REGISTER (ADDRESS X'00B7)

The CMPSL register contains the following bits:

CMPNEG — Will drive I1 to a low level. This bit can be used to discharge an external capacitor. This bit is disabled if the comparator is not enabled (CMPEN = 0).

CMPEN — Enable the comparator ("1" = enable).

CSEN — Enables the internal constant current source. This current source provides a nominal 20 μA constant current at the I1 pin. This current can be used to ensure a linear charging rate on an external capacitor. This bit has no affect and the current source is disabled if the comparator is not enabled (CMPEN = 0).

CMPOE — Enables the comparator output to either pin I3 or pin I7 ("1" = enable) depending on the value of CMPISEL0/1/2.

CMPISEL0/1/2 — Will select one of seven possible sources (I0/I2/I3/I4/I5/I6/internal reference) as a positive input to the comparator (see Table I for more information.)

CMPT2B — Selects the timer T2B input to be driven directly by the comparator output. If the comparator is disabled (CMPEN = 0), this function is disabled, i.e., the T2B input is connected to Port L5.

CMPT2B	CMPISEL2	CMPISEL1	CMPISEL0	CMPOE	CSEN	CMPEN	CMPNEG
Bit 7							Bit 0

The Comparator Select Register is cleared on RESET (the comparator is disabled). To save power the program should also disable the comparator before the μC enters the HALT/IDLE modes. Disabling the comparator will turn off the constant current source and the $V_{CC}/2$ reference, disconnect the comparator output from the T2B input and pin I3 or I7 and remove the low on I1 caused by CMPNEG.

It is often useful for the user's program to read the result of a comparator operation. Since I1 is always selected to be COMPIN− when the comparator is enabled (CMPEN = 1), the comparator output can be read internally by reading bit 1 (CMPRD) of register PORTI (RAM address 0 x D7).

The following table lists the comparator inputs and outputs vs. the value of the CMPISEL0/1/2 bits. The output will only be driven if the CMPOE bit is set to 1.

TABLE I. Comparator Input Selection

Control Bit			Comparator Input Source		Comparator Output
CMPISEL2	CMPISEL1	CMPISEL0	Neg. Input	Pos. Input	
0	0	0	I1	I2	I3
0	0	1	I1	I2	I7
0	1	0	I1	I3	I7
0	1	1	I1	I0	I7
1	0	0	I1	I4	I7
1	0	1	I1	I5	I7
1	1	0	I1	I6	I7
1	1	1	I1	V_{CC}/2 Ref.	I7

Reset

The state of the Comparator Block immediately after RESET is as follows:

1. The CMPSL Register is set to all zeros
2. The Comparator is disabled
3. The Constant Current Source is disabled
4. CMPNEG is turned off
5. The Port I inputs are electrically isolated from the comparator
6. The T2B input is as normally selected by the T2CNTRL Register
7. CMPISEL0–CMPISEL2 are set to zero
8. All Port I inputs are selected to the default digital input mode

The comparator outputs have the same specification as Ports L and G except that the rise and fall times are symmetrical.

Interrupts

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

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

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

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

This procedure takes 7 t_c cycles to execute.

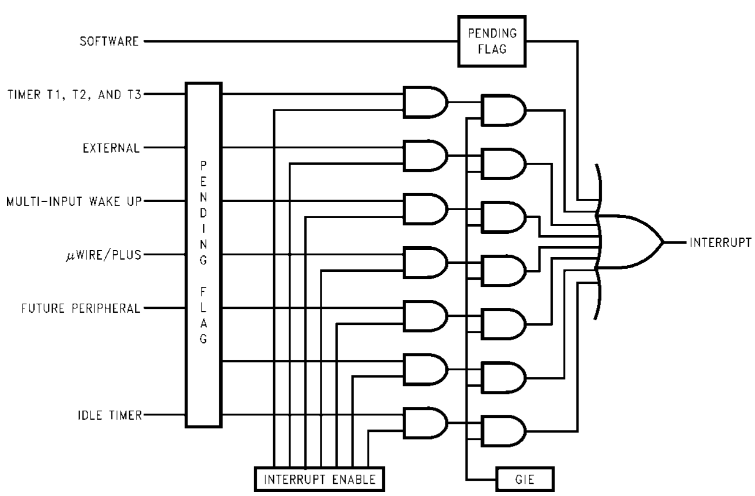


FIGURE 13. Interrupt Block Diagram

Interrupts (Continued)

Arbitration Ranking	Source	Description	Vector* Address Hi-Low Byte
(1) Highest	Software	INTR Instruction	0yFE–0yFF
(2)	Reserved		0yFC–0yFD
(3)	External	G0	0yFA–0yFB
(4)	Timer T0	Underflow	0yF8–0yF9
(5)	Timer T1	T1A/Underflow	0yF6–0yF7
(6)	Timer T1	T1B	0yF4–0yF5
(7)	MICROWIRE/PLUS	BUSY Low	0yF2–0yF3
(8)	Reserved		0yF0–0yF1
(9)	Reserved		0yEE–0yEF
(10)	Reserved		0yEC–0yED
(11)	Timer T2	T2A/Underflow	0yEA–0yEB
(12)	Timer T2	T2B	0yE8–0yE9
(13)	Timer T3	T3A/Underflow	0yE6–0yE7
(14)	Timer T3	T3B	0yE4–0yE5
(15)	Port L/Wakeup	Port L Edge	0yE2–0yE3
(16) Lowest	Default	VIS Instr. Execution without Any Interrupts	0yE0–0yE1

*y is a variable which represents the VIS block. VIS and the vector table must be located in the same 256-byte block except if VIS is located at the last address of a block. In this case, the table must be in the next block.

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

Thus, if an interrupt with a higher rank than the one which caused the interruption becomes active before the decision of which interrupt to service is made by the VIS, then the interrupt with the higher rank will override any lower ones and will be acknowledged. The lower priority interrupt(s) are still pending, however, and will cause another interrupt immediately following the completion of the interrupt service routine associated with the higher priority interrupt just serviced. This lower priority interrupt will occur immediately following the RETI (Return from Interrupt) instruction at the end of the interrupt service routine just completed.

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

The VIS instruction looks at all the active interrupts at the time it is executed and performs an indirect jump to the beginning of the service routine of the one with the highest rank.

The addresses of the different interrupt service routines, called vectors, are chosen by the user and stored in ROM in a table starting at 01E0 (assuming that VIS is located between 00FF and 01DF). The vectors are 15-bit wide and therefore occupy 2 ROM locations.

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

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

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

If, by accident, a VIS gets executed and no interrupt is active, then the PC (Program Counter) will branch to a vector located at 0yE0–0yE1. This vector can point to the Software Trap (ST) interrupt service routine, or to another special service routine as desired.

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

Figure 13 shows the Interrupt block diagram.

Interrupts (Continued)

SOFTWARE TRAP

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

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

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

The ST has the highest rank among all interrupts.

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

WATCHDOG

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

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

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

TABLE II. WATCHDOG Service Register (WDSVR)

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

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

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

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

TABLE III. WATCHDOG Service Window Select

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

Clock Monitor

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

WATCHDOG Operation

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

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

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

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

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

WATCHDOG Operation (Continued)

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

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

$1/t_c > 10$ kHz—No clock rejection.

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

WATCHDOG AND CLOCK MONITOR SUMMARY

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

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

Detection of Illegal Conditions

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

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

The subroutine stack grows down for each call (jump to subroutine), interrupt, or PUSH, and grows up for each return or POP. The stack pointer is initialized to RAM location 06F Hex during reset. Consequently, if there are more returns than calls, the stack pointer will point to addresses 070 and 071 Hex (which are undefined RAM). Undefined RAM from addresses 070 to 07F (Segment 0), 140 to 17F (Segment 1), and all other segments (i.e., Segments 2 . . . etc.) is read as all 1's, which in turn will cause the program to return to address 7FFF Hex. This is an undefined ROM location and the instruction fetched (all 0's) from this location will generate a software interrupt signaling an illegal condition.

Thus, the chip can detect the following illegal conditions:

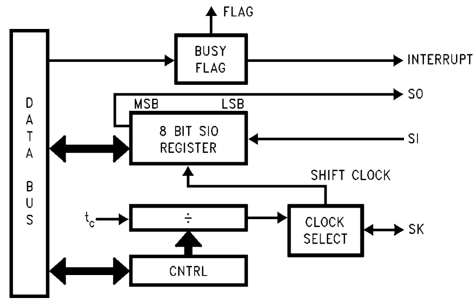
a. Executing from undefined ROM

b. Over ''POP''ing the stack by having more returns than calls.

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

MICROWIRE/PLUS

MICROWIRE/PLUS is a serial synchronous communications interface. The MICROWIRE/PLUS capability enables the device to interface with any of National Semiconductor's MICROWIRE peripherals (i.e. A/D converters, display drivers, E^2PROMs etc.) and with other microcontrollers which support the MICROWIRE interface. It consists of an 8-bit serial shift register (SIO) with serial data input (SI), serial data output (SO) and serial shift clock (SK). *Figure 14* shows a block diagram of the MICROWIRE/PLUS logic.



TL/DD/12520–17

FIGURE 14. MICROWIRE/PLUS Block Diagram

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

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

TABLE IV. WATCHDOG Service Actions

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

TABLE V. MICROWIRE/PLUS Master Mode Clock Select

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

Where t_c is the instruction cycle clock

MICROWIRE/PLUS OPERATION

Setting the BUSY bit in the PSW register causes the MICROWIRE/PLUS to start shifting the data. It gets reset when eight data bits have been shifted. The user may reset the BUSY bit by software to allow less than 8 bits to shift. If enabled, an interrupt is generated when eight data bits have been shifted. The device may enter the MICROWIRE/PLUS mode either as a Master or as a Slave. *Figure 15* shows how two microcontroller devices and several peripherals may be interconnected using the MICROWIRE/PLUS arrangements.

Warning:

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

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

MICROWIRE/PLUS Master Mode Operation

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

MICROWIRE/PLUS Slave Mode Operation

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

Alternate SK Phase Operation

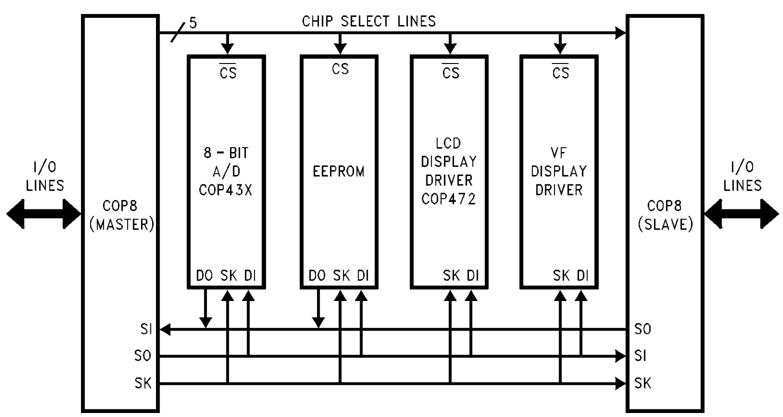
The device allows either the normal SK clock or an alternate phase SK clock to shift data in and out of the SIO register. In both the modes the SK is normally low. In the normal mode data is shifted in on the rising edge of the SK clock and the data is shifted out on the falling edge of the SK clock.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

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

TABLE VI

This table assumes that the control flag MSEL is set.

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



TL/DD/12520–18

FIGURE 15. MICROWIRE/PLUS Application

Memory Map

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

Address S/ADD REG	Contents
0000 to 006F	On-Chip RAM bytes (112 bytes)
0070 to 007F	Unused RAM Address Space (Reads As All Ones)
xx80 to xxAF	Unused RAM Address Space (Reads Undefined Data)
xxB0	Timer T3 Lower Byte
xxB1	Timer T3 Upper Byte
xxB2	Timer T3 Autoload Register T3RA Lower Byte
xxB3	Timer T3 Autoload Register T3RA Upper Byte
xxB4	Timer T3 Autoload Register T3RB Lower Byte
xxB5	Timer T3 Autoload Register T3RB Upper Byte
xxB6	Timer T3 Control Register
xxB7	Comparator Select Register (CMPSL)
xxB8 to xxBF	Reserved
xxC0	Timer T2 Lower Byte
xxC1	Timer T2 Upper Byte
xxC2	Timer T2 Autoload Register T2RA Lower Byte
xxC3	Timer T2 Autoload Register T2RA Upper Byte
xxC4	Timer T2 Autoload Register T2RB Lower Byte
xxC5	Timer T2 Autoload Register T2RB Upper Byte
xxC6	Timer T2 Control Register
xxC7	WATCHDOG Service Register (Reg:WDSVR)
xxC8	MIWU Edge Select Register (Reg:WKEDG)
xxC9	MIWU Enable Register (Reg:WKEN)
xxCA	MIWU Pending Register (Reg:WKPND)
xxCB	Reserved
xxCC	Reserved
xxCD to xxCF	Reserved

Address S/ADD REG	Contents
xxD0	Port L Data Register
xxD1	Port L Configuration Register
xxD2	Port L Input Pins (Read Only)
xxD3	Reserved for Port L
xxD4	Port G Data Register
xxD5	Port G Configuration Register
xxD6	Port G Input Pins (Read Only)
xxD7	Port I Input Pins (Read Only)
xxD8	Port C Data Register
xxD9	Port C Configuration Register
xxDA	Port C Input Pins (Read Only)
xxDB	Reserved for Port C
xxDC	Port D
xxDD to xxDF	Reserved
xxE0 to xxE5	Reserved
xxE6	Timer T1 Autoload Register T1RB Lower Byte
xxE7	Timer T1 Autoload Register T1RB Upper Byte
xxE8	ICNTRL Register
xxE9	MICROWIRE/PLUS Shift Register
xxEA	Timer T1 Lower Byte
xxEB	Timer T1 Upper Byte
xxEC	Timer T1 Autoload Register T1RA Lower Byte
xxED	Timer T1 Autoload Register T1RA Upper Byte
xxEE	CNTRL Control Register
xxEF	PSW Register
xxF0 to FB	On-Chip RAM Mapped as Registers
xxFC	X Register
xxFD	SP Register
xxFE	B Register
xxFF	S Register
0100 to 017F	On-Chip 128 RAM Bytes

Reading memory locations 0070H–007FH (Segment 0) will return all ones. Reading unused memory locations 0080H–00AFH (Segment 0) will return undefined data. Reading memory locations from other unused Segments (i.e., Segment 2, Segment 3, … etc.) will return all ones.

Addressing Modes

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

OPERAND ADDRESSING MODES

Register Indirect

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

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

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

Direct

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

Immediate

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

Short Immediate

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

Indirect

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

TRANSFER OF CONTROL ADDRESSING MODES

Relative

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

Absolute

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

Absolute Long

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

Indirect

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

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

Instruction Set

Register and Symbol Definition

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

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

Instruction Set (Continued)

INSTRUCTION SET

ADD	A,Meml	ADD	A ← A + Meml
ADC	A,Meml	ADD with Carry	A ← A + Meml + C, C ← Carry HC ← Half Carry
SUBC	A,Meml	Subtract with Carry	A ← A − $\overline{\text{Meml}}$ + C, C ← Carry HC ← Half Carry
AND	A,Meml	Logical AND	A ← A and Meml
ANDSZ	A,Imm	Logical AND Immed., Skip if Zero	Skip next if (A and Imm) = 0
OR	A,Meml	Logical OR	A ← A or Meml
XOR	A,Meml	Logical EXclusive OR	A ← A xor Meml
IFEQ	MD,Imm	IF EQual	Compare MD and Imm, Do next if MD = Imm
IFEQ	A,Meml	IF EQual	Compare A and Meml, Do next if A = Meml
IFNE	A,Meml	IF Not Equal	Compare A and Meml, Do next if A ≠ Meml
IFGT	A,Meml	IF Greater Than	Compare A and Meml, Do next if A > Meml
IFBNE	#	If B Not Equal	Do next if lower 4 bits of B ≠ Imm
DRSZ	Reg	Decrement Reg., Skip if Zero	Reg ← Reg − 1, Skip if Reg = 0
SBIT	#,Mem	Set BIT	1 to bit, Mem (bit = 0 to 7 immediate)
RBIT	#,Mem	Reset BIT	0 to bit, Mem
IFBIT	#,Mem	IF BIT	If bit in A or Mem is true do next instruction
RPND		Reset PeNDing Flag	Reset Software Interrupt Pending Flag
X	A,Mem	EXchange A with Memory	A ←→ Mem
X	A,[X]	EXchange A with Memory [X]	A ←→ [X]
LD	A,Meml	LoaD A with Memory	A ← Meml
LD	A,[X]	LoaD A with Memory [X]	A ← [X]
LD	B,Imm	LoaD B with Immed.	B ← Imm
LD	Mem,Imm	LoaD Memory Immed	Mem ← Imm
LD	Reg,Imm	LoaD Register Memory Immed.	Reg ← Imm
X	A, [B ±]	EXchange A with Memory [B]	A ←→ [B], (B ← B ±1)
X	A, [X ±]	EXchange A with Memory [X]	A ←→ [X], (X ← ±1)
LD	A, [B±]	LoaD A with Memory [B]	A ← [B], (B ← B ±1)
LD	A, [X±]	LoaD A with Memory [X]	A ← [X], (X ← X±1)
LD	[B±],Imm	LoaD Memory [B] Immed.	[B] ← Imm, (B ← B±1)
CLR	A	CLeaR A	A ← 0
INC	A	INCrement A	A ← A + 1
DEC	A	DECrement A	A ← A − 1
LAID		Load A InDirect from ROM	A ← ROM (PU,A)
DCOR	A	Decimal CORrect A	A ← BCD correction of A (follows ADC, SUBC)
RRC	A	Rotate A Right thru C	C → A7 → . . . → A0 → C
RLC	A	Rotate A Left thru C	C ← A7 ← . . . ← A0 ← C
SWAP	A	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
POP	A	POP the stack into A	SP ← SP + 1, A ← [SP]
PUSH	A	PUSH A onto the stack	[SP] ← A, SP ← SP − 1
VIS		Vector to Interrupt Service Routine	PU ← [VU], PL ← [VL]
JMPL	Addr.	Jump absolute Long	PC ← ii (ii = 15 bits, 0 to 32k)
JMP	Addr.	Jump absolute	PC9 . . . 0 ← i (i = 12 bits)
JP	Disp.	Jump relative short	PC ← PC + r (r is −31 to +32, except 1)
JSRL	Addr.	Jump SubRoutine Long	[SP] ← PL, [SP−1] ← PU,SP−2, PC ← ii
JSR	Addr	Jump SubRoutine	[SP] ← PL, [SP−1] ← PU,SP−2, PC9 . . . 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]
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

Instruction Execution Time

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

Most single byte instructions take one cycle time to execute.

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 and Logic Instructions

	[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
IFNE	1/1	3/4	2/2
IFGT	1/1	3/4	2/2
IFBNE	1/1		
DRSZ		1/3	
SBIT	1/1	3/4	
RBIT	1/1	3/4	
IFBIT	1/1	3/4	

RPND	1/1

Instructions Using A & C

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

Transfer of Control Instructions

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

Memory Transfer Instructions

	Register Indirect		Direct	Immed.	Register Indirect Auto Incr. & Decr.		
	[B]	[X]			[B+, B−]	[X+, X−]	
X A,*	1/1	1/3	2/3		1/2	1/3	
LD A,*	1/1	1/3	2/3	2/2	1/2	1/3	
LD B, Imm				1/1			(IF B < 16)
LD B, Imm				2/2			(IF B > 15)
LD Mem, Imm	2/2		3/3		2/2		
LD Reg, Imm			2/3				
IFEQ MD, Imm			3/3				

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

Opcode Table

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

Where,

i is the immediate data

Md is a directly addressed memory location

* is an unused opcode

Note: The opcode 60 Hex is also the opcode for IFBIT #i,A

Development Support

IN-CIRCUIT EMULATOR

The MetaLink iceMASTER™-COP8 Model 400 In-Circuit Emulator for the COP8 family of microcontrollers features high-performance operation, ease of use, and an extremely flexible user-interface for maximum productivity. Interchangeable probe cards, which connect to the standard common base, support the various configurations and packages of the COP8 family.

The iceMASTER provides real time, full speed emulation up to 10 MHz, 32 kBytes of emulation memory and 4k frames of trace buffer memory. The user may define as many as 32k trace and break triggers which can be enabled, disabled, set or cleared. They can be simple triggers based on code or address ranges or complex triggers based on code address, direct address, opcode value, opcode class or immediate operand. Complex breakpoints can be ANDed and ORed together. Trace information consists of address bus values, opcodes and user selectable probe clips status (external event lines). The trace buffer can be viewed as raw hex or as diassembled instructions. The probe clip bit values can be displayed in binary, hex or digital waveform formats.

During single-step operation the dynamically annotated code feature displays the contents of all accessed (read and write) memory locations and registers, as well as flow-of-control direction change markers next to each instruction executed.

The iceMASTER's performance analyzer offers a resolution of better than 6 μs. The user can easily monitor the time spent executing specific portions of code and find ''hot spots'' or ''dead code''. Up to 15 independent memory areas based on code address or label ranges can be defined. Analysis results can be viewed in bar graph format or as actual frequency count.

Emulator memory operations for program memory include single line assembler, disassembler, view, change and write to file. Data memory operations include fill, move, compare, dump to file, examine and modify. The contents of any memory space can be directly viewed and modified from the corresponding window.

The iceMASTER comes with an easy to use windowed interface. Each window can be sized, highlighted, color-controlled, added, or removed completely. Commands can be accessed via pull-down-menus and/or redefineable hot keys. A context sensitive hypertext/hyperlinked on-line help system explains clearly the options the user has from within any window.

The iceMASTER connects easily to a PC® via the standard COMM port and its 115.2 kBaud serial link keeps typical program download time shorter.

The following tables list the emulator and probe cards ordering information.

Emulator Ordering Information

Part Number	Description
IM-COP8/400/1‡	MetaLink base unit in-circuit emulator for all COP8 devices, symbolic debugger software and RS 232 serial interface cable, with 110V @ 60 Hz Power Supply.
IM-COP8/400/2‡	MetaLink base unit in-circuit emulator for all COP8 devices, symbolic debugger software and RS 232 serial interface cable, with 220V @ 50 Hz Power Supply.

‡These parts include National's COP8 Assembler/Linker/Librarian Package (COP8-DEV-IBMA).

Development Support (Continued)

Probe Card Ordering Information

Part Number	Package	Voltage Range	Emulates
MHW-884EK28D5PC	28 DIP	4.5V–5.5V	COP884EK
MHW-884EK28DWPC	28 DIP	2.5V–5.5V	COP884EK
MHW-888EK40D5PC	40 DIP	4.5V–5.5V	COP888EK
MHW-888EK40DWPC	40 DIP	2.5V–5.5V	COP888EK
MHW-888EK44P5PC	44 PLCC	4.5V–5.5V	COP888EK
MHW-888EK44PWPC	44 PLCC	2.5V–5.5V	COP888EK
MHW-SOIC28	28 SO Adapter*		

*Required in addition to the probe card.

METALINK iceMASTER-COP8 DEBUG MODULE

The Metalink iceMASTER-COP8 Debug Module has most of the salient features of the IceMASTER-400 and in addition has a COP8 PROM and OTP programmer. It is a low-cost tool for designing, debugging and evaluating COP8 Microcontroller Unit (MCU) devices. This provides all of the essential MCU timing, I/O circuitry and therefore simplifies the evaluation of the prototype hardware/software product.

The Debug Module is controlled by an IBM PC (or compatible) running MS-DOS communicating over a serial port at 9600 baud. The Debug Module uses the same menu driven user interface as the Metalink iceMASTER-400.

The Debug Module can be connected to a target system in place of the microcontroller (using an optional target interface cable) or operated independently in the stand alone mode. Stand alone mode allows you to emulate hardware and/or execute code without a target system (provided no interaction with external devices is needed).

Hardware designers can use the Debug Module to develop and debug their designs. All available features of a given device are accessible interactively, as well as through the application programs. Software designers have complete emulation capability as well. The Debug Module will execute the code just like the real part because it uses a real part for emulation.

The Debug Module utilizes the INTR instruction to implement software breakpoints. If your application code contains any INTR instructions, they will never be executed. The debug module behaves exactly as though a breakpoint occurred when it passes through such locations. In the iceMASTER-400 breakpoints are implemented in hardware and if the application were to contain INTR instructions, they will not be treated as breakpoints.

When a breakpoint is set on an instruction which could be potientially skipped, the emulation will break only when the instruction is actually executed. Emulation will never break when the instruction is skipped. When a breakpoint is set on an instruction which could be potentially skipped and that instruction is skipped during emulation, the address and data values captured in the trace will be different from a real COP8 processor. The first cycle will be that of a skipped INTR instruction. For multi-byte instructions, subsequent cycles will be the same as that of executed NOP instructions.

When a breakpoint is set on an instruction, emulation breaks before the instruction executes. Two bytes of the stack will be written when the breakpoint is reached. This is because INTR instructions are used to implement breakpoints.

For the basic family COP8 processors, the timers are shut off at breakpoints shortly after emulation stops. When the emulation resumes, the timers are restarted just before the emulation of the target application program actually begins. To allow for clock resynchronization in the COP microcontroller, it is necessary to program two NOP instructions immediately after the processor comes out of the HALT mode. When the multi-input wakeup interrupt is enabled, the first two instructions of the interrupt routine must be NOP's. If no interrupts are used to enter the HALT mode, then two NOP's must follow "enter HALT mode" (set G7 data bit) instruction.

Debug Module Ordering Information

Part Number	Description
COP8-DM/888EK‡	Metalink IceMaster Debug Module. This is the low cost version of the Metalink IceMaster.
DM-COP8/28D	28 DIP Target Cable
DM-COP8/28D-SO	28 SO Adapter*
DM-COP8/40D	40 DIP Target Cable
DM-COP8/44P	44 PLCC Target Cable

‡This is bundled with the COP8 assembler package.

*Required in addition to target cable.

MACRO CROSS ASSEMBLER

National Semiconductor offers a relocatable COP8 macro cross assembler. It runs on industry standard compatible PCs and supports all of the full-symbolic debugging features of the MetaLink IceMASTER emulators.

Assembler Ordering Information

Part Number	Description	Manual
COP8-DEV-IBMA	COP8 Assembler/Linker/ Librarian for IBM®, PC/XT®, AT® or compatible. Current Version: 4.6.	620896

OTP/EMULATOR SUPPORT

The COP8 family is supported by single chip OTP emulators.

COP8788EK/COP8784EK Ordering Information

Device Number	Clock Option	Package	Emulates
COP87L88EKV-XE	Crystal/HALT En	44 PLCC	COP888EK
COP87L88EKN-XE	Crystal/HALT En	40 DIP	COP888EK
COP87L84EKN-XE	Crystal/HALT En	28 DIP	COP884EK
COP87L84EKM-XE	Crystal/HALT En	28 SO	COP884EK

*Check with the local sales office about the availability.

Development Support (Continued)

PROGRAMMING SUPPORT

COP8 EPU

Programming of these OTP devices is supported by the MetaLink iceMASTER-COP8 Evaluation and Programming Unit (COP8-EPU).

The COP8-EPU is a low cost unit which can be used to program COP8 OTP microcontrollers. The host computer for the COP8-EPU is a standard PC (or compatible) running the DOS operating System. The interface to the COP8-EPU is over the RS-232C serial channel at 115,200 baud.

The on-board voltage generator supplies all the voltages required to program the EPROM/OTP using only the wall mounted power supply provided.

User Interface

The COP8-EPU offers a friendly, pop-up/pull-down window, menu driven user interface featuring programmable function keys, a context sensitive, hyperlinked on-line help system and configurable displays.

Home Computer Specification

- IBM PC (or compatible) with minimum 640 bytes of memory
- One 3.5″, 1.44 MB floppy diskette drive
- One hard drive
- RS-232C serial channel
- PC-DOS/MS-DOS version 3.1 or later

Ordering Information

The following table is to provide a list of the EPU adapters and their ordering information:

COP888—EPU Ordering Information

Not recommended for volume programming.

Part Number	Devices Supported
EPU-COP888GG-1‡	EPU with 110V power supply
EPU-COP888GG-2‡	EPU with 220V power supply
COP8-PGMA-DS	Programming Adaptor for all COP8 Basic and Feature family devices in 20, 28, 40 dip; 20, 28 SO packages.
COP8-PGMA-DS44P	Programming Adaptor for all COP8 Basic and Feature family devices in 20, 28, 40 DIP; 20, 28 SO; and 44 PLCC packages.

‡These parts include National's COP8 Assembler/Linker/Librarian Package (COP8-DEV-IBMA).

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 630006.

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

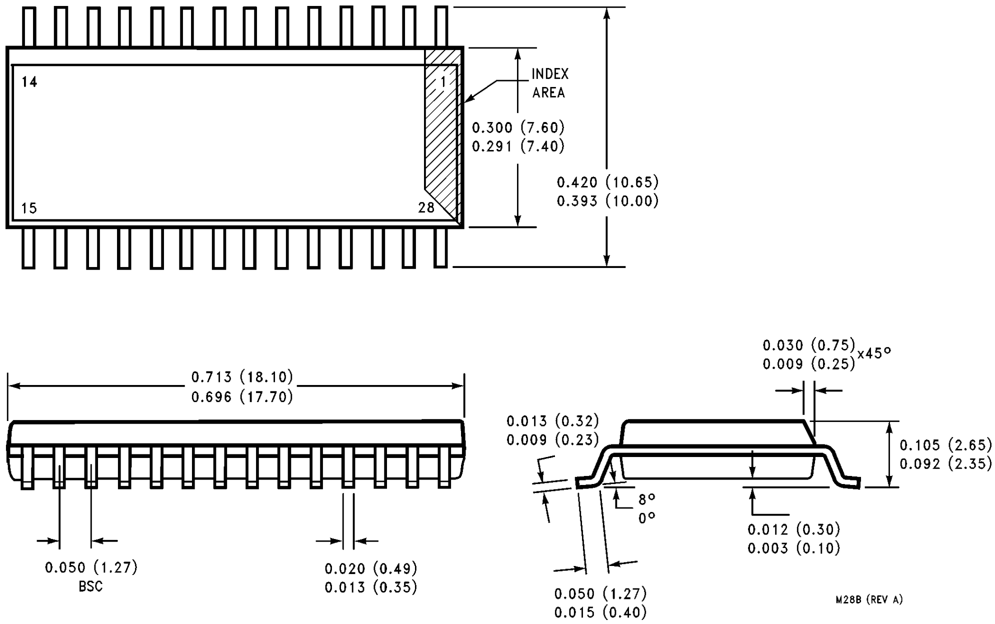
See us on the WorldWide Web at: http://www.natsemi.com

CUSTOMER RESPONSE CENTER

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

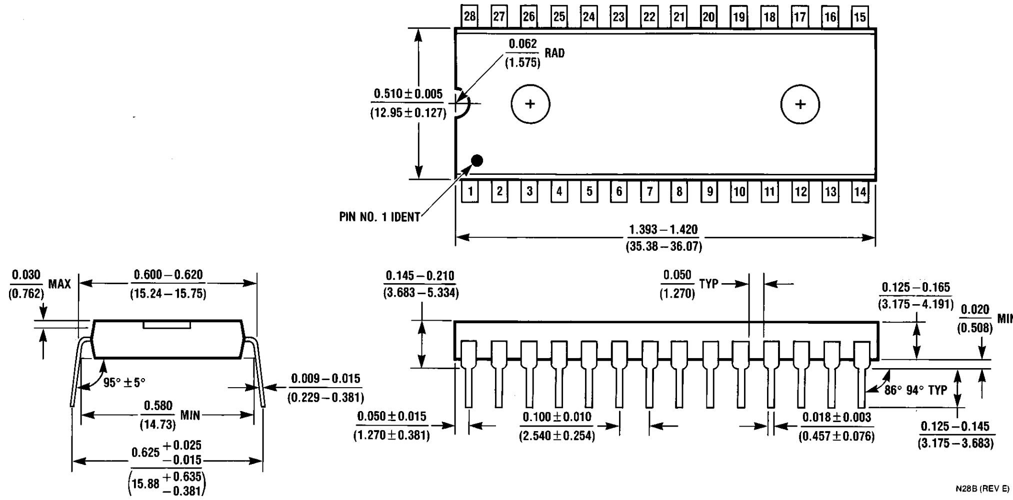
CANADA/US:	Tel:	(800) 272-9959
	email:	support @tevm2.nsc.com
EUROPE:	email:	europe.support@nsc.com
	Deutsch Tel:	+49 (0) 180-530 85 85
	English Tel:	+49 (0) 180-532 78 32
	Français Tel:	+49 (0) 180-532 93 58
	Italiano Tel:	+49 (0) 180-534 16 80
JAPAN:	Tel:	+81-043-299-2309
S.E. ASIA:	Beijing Tel:	(+86) 10-856-8601
	Shanghai Tel:	(+86) 21-6249-6062
	Hong Kong Tel:	(+852) 2737-1600
	Korea Tel:	(+82) 2-3771-6909
	Malaysia Tel:	011-644-9061
	Singapore Tel:	(+65) 255-2226
	Taiwan Tel:	+886-2-521-3288
AUSTRALIA:	Tel:	(+61) 3-9558-9999
INDIA:	Tel:	(+91) 80-559-9467

Physical Dimensions inches (millimeters)

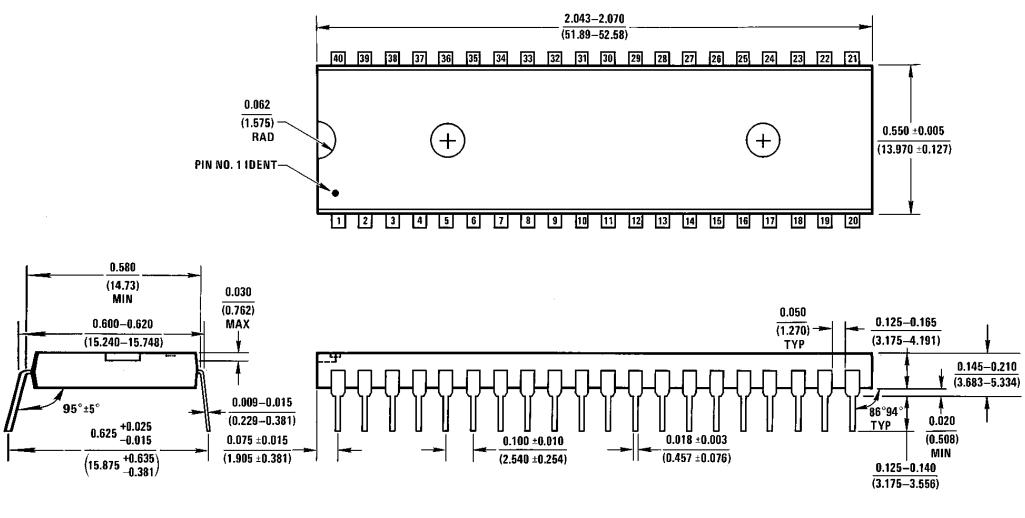


Molded SO Wide Body Package (M)
Order Number COP87L84EKM-XE, COP87L84EKM-XD,
COP87L84EKM-RE or COP87L84EKM-RD
NS Package Number M28B

Physical Dimensions inches (millimeters) (Continued)

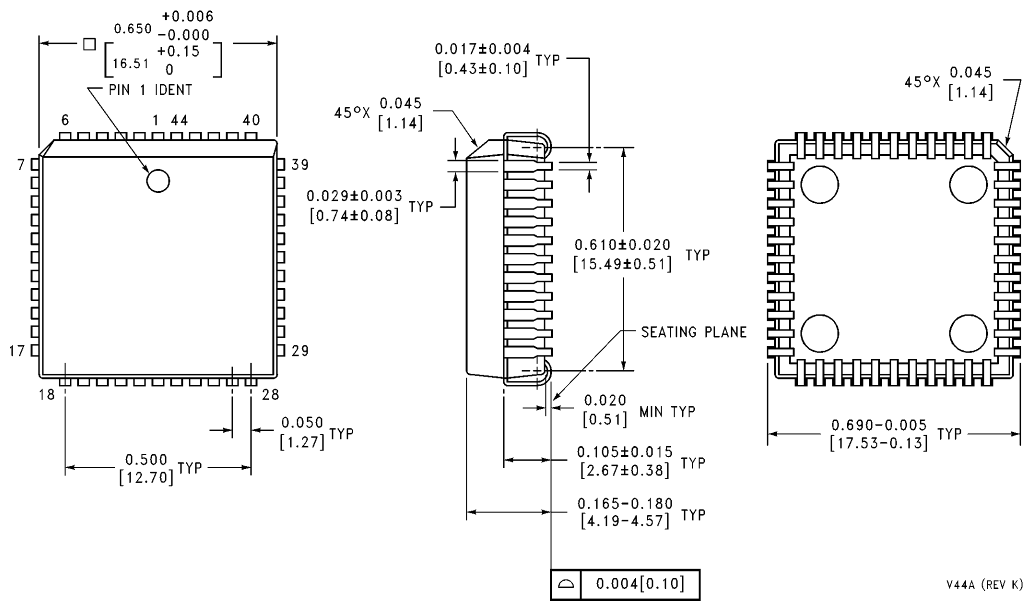


Molded Dual-In-Line Package (N)
Order Number COP87L84EKN-XE, COP87L84EKN-XD,
COP87L84EKN-RE or COP87L84EKN-RD
NS Package Number N28B



Molded Dual-In-Line Package (N)
Order Number COP87L84EKN-XE, COP87L84EKN-XD,
COP87L84EKN-RE or COP87L84EKN-RD
NS Package Number N40A

Physical Dimensions inches (millimeters) (Continued)



Plastic Leaded Chip Carrier (V)
Order Number COP87L88EKV-XE, COP87L88EKV-XD,
COP87L88EKV-RE or COP87L88EKV-RD
NS Package Number V44A

LIFE SUPPORT POLICY

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

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

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

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

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2308
Fax: 81-043-299-2408

http://www.national.com