National Semiconductor

March 1997

COP8ACC7 8-Bit One-Time Programmable (OTP) Microcontroller with High Resolution A/D Conversion

General Description

The COP8ACC7 is a member of the COP8™ 8-bit OTP microcontroller family. It is pin and software compatible to the mask ROM COP8ACC5 product family. (Continued)

Key Features

- Analog Function Block for high resolution A/D including
  - — Analog comparator with seven input muxes
  - — Constant Current Source and $V_{CC/2}$ Reference
  - — 16-bit capture timer (upcounter) clocked from CKI with auto-reset on timer start-up
- Quiet design (reduced radiated emissions)
- 4096 bytes on-board OTP EPROM with security feature
- 128 bytes on-board RAM

Additional Peripheral Features

- Idle Timer
- One 16-bit timer with two 16-bit registers supporting:
  - — Processor Independent PWM mode
  - — External Event counter mode
  - — Input Capture mode
- Multi-Input Wake-Up (MIWU) with optional interrupts (4)
- WATCHDOG and clock monitor logic
- MICROWIRE/PLUS serial I/O with programmable shift clock polarity

I/O Features

- Memory mapped I/O
- Software selectable I/O options (Push-Pull Output, Weak Pull-Up Input, High Impedance Input)
- High current outputs
- Schmitt Trigger inputs on ports G and L
- Packages: 28 DIP/SO with 23 I/O pins
  20 SO with 15 I/O pins

CPU/Instruction Set Features

- 1µs instruction cycle time
- Eight multi-source vectored interrupt servicing
  - — External Interrupt
  - — Idle Timer T0
  - — Timer T1 associated Interrupts
  - — MICROWIRE/PLUS
  - — Multi-Input Wake Up
  - — Software Trap
  - — Default VIS
  - — A/D (Capture Timer)
- Versatile and easy to use instruction set
- 8-bit Stack Pointer (SP) - stack in RAM
- Two 8-bit Registers Indirect Data Memory Pointersv (B and X)

Fully Static CMOS

- Low current drain (typically < 5 µA HALT current)
- Two power saving modes: HALT and IDLE
- Single supply operation: 2.7V to 5.5V
- Temperature ranges: 0˚C to +70˚C, –40˚C to +85˚C

Development System

- Emulation device for COP8ACC5
- Real time emulation and full program debug offered by MetaLink development system

Applications

- Battery Chargers
- Appliances
- Data Acquisition systems

Block Diagram

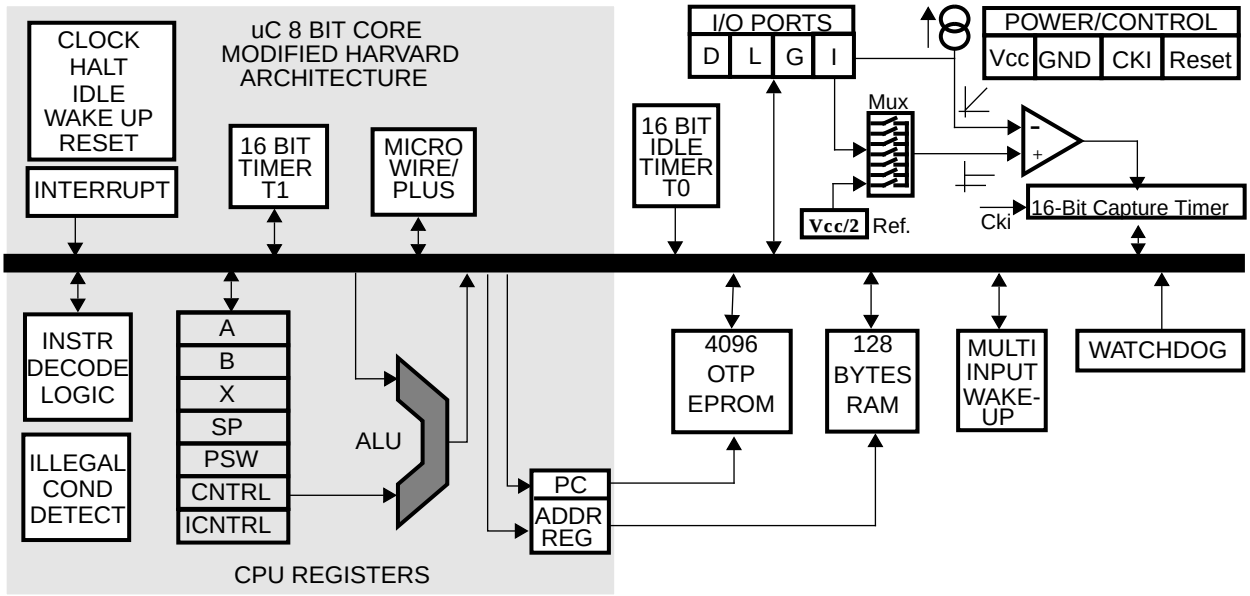


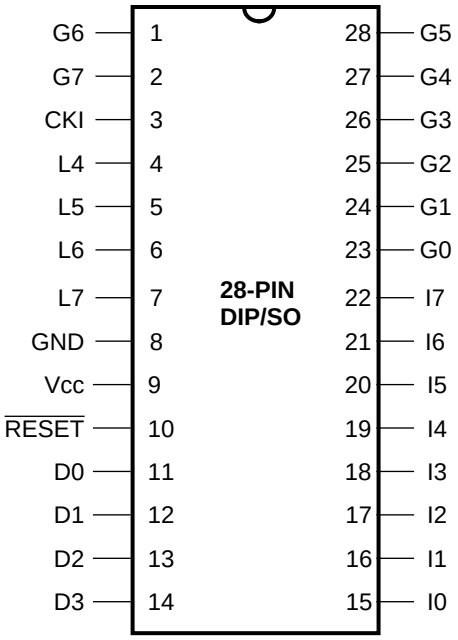
Figure 1. Block Diagram

General Description

The device provides up to 6 channels of A/D performing a measurement with 12-bits of resolution in less than 0.5 msec at a clock-rate of 10Mhz. There is only an external capacitor required to complete the measurement setup and establishing low cost, high-resolution (up to 16 bits) and accurate A/D.

This device is a complete microcomputer containing all system timing, interrupt logic, EPROM, RAM, and I/O necessary to implement dedicated control functions in a variety of applications. Features include an 8-bit memory mapped architecture, MICROWIRE/PLUS serial I/O, one 16-bit PWM-timer with two autoreload registers, multi-sourced interrupts an Analog Function Block and an idle timer WATCHDOG. Each I/O pin has software selectable options to adapt the device to the specific application. The device operates over a voltage range of 2.7 to 5.5 volts. High throughput is achieved with an efficient, regular instruction set operating at a minimum of 1 ms per instruction cycle.

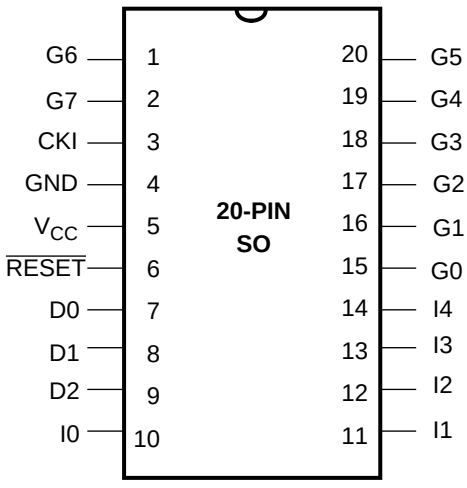
Connection Diagrams



Top View

Order Number COP8ACC728N9, COP8ACC728N8
See NS Molded Package Number N28A

Order Number COP8ACC728M9, COP8ACC728M8
See NS Molded Package Number M28B



Top View

Order Number COP8ACC520M9, COP8ACC720M8
see NS Molded Package Number M20B

Figure 2. Connection Diagram

Connection Diagrams (Continued)

Pinouts for 28-, 20-Pin Packages

Port	Type	Alt. Fun	Alt. Fun	28-Pin DIP/SO	20-Pin SO
L4	I/O	MIWU	Ext. Int.	4	
L5	I/O	MIWU	Ext. Int.	5	
L6	I/O	MIWU	Ext. Int.	6	
L7	I/O	MIWU	Ext. Int.	7	
G0	I/O	INT		23	15
G1	WDOUT			24	16
G2	I/O	T1B		25	17
G3	I/O	T1A		26	18
G4	I/O	SO		27	19
G5	I/O	SK		28	20
G6	I	SI		1	1
G7	I/CKO	HALT Restart		2	2
D0	O			11	7
D1	O			12	8
D2	O			13	9
D3	O			14	
I0	I	Analog CH1		15	10
I1	I	I_{SRC}		16	11
I2	I	Analog CH2		17	12
I3	I	Analog CH3		18	13
I4	I	Analog CH4		19	14
I5	I	Analog CH5		20	
I6	I	Analog CH6		21	
I7	I	C_{OUT}		22	
V_{CC}				9	5
GND				8	4
CKI				3	3
$\overline{\text{RESET}}$				10	6

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 $0˚C \le T_A \le +70˚C$ 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 = 4 MHz	V_{CC} = 5.5V, t_c = 2.5 µs			9.5	mA
CKI = 4 MHz	V_{CC} = 4V, t_c = 2.5 µs			6.5	mA
CKI = 1 MHz	V_{CC} = 4V, t_c = 10 µs			5.4	mA
HALT Current (Note 3)	V_{CC} = 5.5V, CKI = 0 MHz		<5	10	µA
	V_{CC} = 4V, CKI = 0 MHz		<3	6	µA
IDLE Current					
CKI = 4 MHz	V_{CC} = 5.5V, t_c = 2.5 µs			1.5	mA
CKI = 1 MHz	V_{CC} = 4V, t_c = 10 µs			0.5	mA
Input Levels (Vih, Vil)					
$\overline{\text{RESET}}$					
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	1		1	µA
Input Pullup Current	V_{CC} = 5.5V, V_{IN} = 0V	–40		–250	µA
G and L Port Input Hysteresis	(Note 5)			0.35 V_{CC}	V
Output Current Levels					
D Outputs					
Source	V_{CC} = 4V, V_{OH} = 3.3V	–0.4			mA
	V_{CC} = 2.7V, V_{OH} = 1.8V	–0.2			mA
Sink	V_{CC} = 4V, V_{OL} = 1V	10			mA
	V_{CC} = 2.7V, V_{OL} = 0.4V	2.0			mA
All Others					
Source (Weak Pull-Up Mode)	V_{CC} = 4V, V_{OH} = 2.7V	–10		–110	µA
	V_{CC} = 2.7V, V_{OH} = 1.8V	–2.5		–33	µA
Source (Push-Pull Mode)	V_{CC}= 4V, V_{OH} = 3.3V	–0.4			mA
	V_{CC} = 2.7V, V_{OH} = 1.8V	–0.2			mA
Sink (Push-Pull Mode)	V_{CC} = 4V, V_{OL} = 0.4V	1.6			mA
	V_{CC} = 2.7V, V_{OL} = 0.4V	0.7			mA
TRI-STATE Leakage	V_{CC} = 5.5V	1		1	µA
Allowable Sink/Source Current per Pin					
D Outputs (Sink)				15	mA
All others				3	mA
Maximum Input Current without Latchup (Note 4)	Room Temp			±200	mA
RAM Retention Voltage, Vr	500 ns Rise and Fall Time (min)	2			V
Input Capacitance	(Note 5)			7	pF
Load Capacitance on D2	(Note 5)			1000	pF

AC Electrical Characteristics $0^{\circ}C \le T_A \le +70^{\circ}C$ unless otherwise specified

Parameter	Conditions	Min	Typ	Max	Units
Instruction Cycle Time (t_c)					
Crystal, Resonator	$2.7V \le V_{CC} < 4V$	2.5		DC	µs
	$4V \le V_{CC} \le 5.5V$	1.0		DC	µs
R/C Oscillator	$2.7V \le V_{CC} < 4V$	7.5		DC	µs
	$4V \le V_{CC} \le 5.5V$	3.0		DC	µs
Inputs					
t_{SETUP}	$4V \le V_{CC} \le 5.5V$	200			ns
	$2.7V \le V_{CC} < 4V$	500			ns
t_{HOLD}	$4V \le V_{CC} \le 5.5V$	60			ns
	$2.7V \le V_{CC} < 4V$	150			ns
Output Propagation Delay (Note 5)	$R_L = 2.2k$, $C_L = 100$ pF				
t_{PD1}, t_{PD0}					
SO, SK	$4V \le V_{CC} \le 5.5V$			0.7	µs
	$2.7V \le V_{CC} < 4V$			1.75	µs
All Others	$4V \le V_{CC} \le 5.5V$			1	µs
	$2.7V \le V_{CC} < 4V$			2.5	µs
MICROWIRE™ Setup Time (t_{UWS}) (Note 5)	$Vcc \ge 4V$	20			ns
MICROWIRE Hold Time (t_{UWH}) (Note 5)	$Vcc \ge 4V$	56			ns
MICROWIRE Output Propagation Delay (t_{UPD})	$Vcc \ge 4V$			220	ns
Input Pulse Width (Note 6)					
Interrupt Input High Time		1			t_c
Interrupt Input Low Time		1			t_c
Timer 1,2, 3 Input High Time		1			t_c
Timer 1,2, 3 Input Low Time		1			t_c
Reset Pulse Width		1			µs

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

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

Note 3: The HALT mode will stop CKI from oscillating in the RC and the Crystal configurations. Measurement of IDD HALT is done with device neither sourcing or 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 Vcc; 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: Pins G6 and $\overline{\text{RESET}}$ are designed with a high voltage input network. These pins allow input voltages greater than V_{CC} and the pins will have sink current to Vcc when biased at voltages greater than Vcc (the pins do not have source current when biased at a voltage below V_{CC}.) The effective resistance to Vcc is 750 ohms (typical). These two pins will not latch up. The voltage at the pins must be limited to less than 14 Volts. **WARNING: Voltages in excess of 14 volts will cause damage to the pins. This warning excludes ESD transients.**

Note 5: The output propagation delay is referenced to the end of the instruction cycle where the output change occurs.

Note 6: Parameter characterized but not tested.

Note 7: tc = Instruction Cycle Time.

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 = 4 MHz	V_{CC} = 5.5V, t_c = 2.5 µs			9.5	mA
CKI = 4 MHz	V_{CC} = 4V, t_c = 2.5 µs			6.5	mA
CKI = 1 MHz	V_{CC} = 4V, t_c = 10 µs			5.4	mA
HALT Current (Note 3)	V_{CC} = 5.5V, CKI = 0 MHz		<5	12	µA
	V_{CC} = 4V, CKI = 0 MHz		<3	8	µA
IDLE Current					
CKI = 4 MHz	V_{CC} = 5.5V, t_c = 2.5 µs			1.5	mA
CKI = 1 MHz	V_{CC} = 4V, t_c = 10 µs			0.5	mA
Input Levels (Vih, Vil)					
$\overline{\text{RESET}}$					
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 5)			0.35 V_{CC}	V
Output Current Levels					
D Outputs					
Source	V_{CC} = 4V, V_{OH} = 3.3V	–0.4			mA
	V_{CC} = 2.7V, V_{OH} = 1.8V	–0.2			mA
Sink	V_{CC} = 4V, V_{OL} = 1V	10			mA
	V_{CC} = 2.7V, V_{OL} = 0.4V	2.0			mA
All Others					
Source (Weak Pull-Up Mode)	V_{CC} = 4V, V_{OH} = 2.7V	–10		–110	µA
	V_{CC} = 2.7V, V_{OH} = 1.8V	–2.5		–33	µA
Source (Push-Pull Mode)	V_{CC}= 4V, V_{OH} = 3.3V	–0.4			mA
	V_{CC} = 2.7V, V_{OH} = 1.8V	–0.2			mA
Sink (Push-Pull Mode)	V_{CC} = 4V, V_{OL} = 0.4V	1.6			mA
	V_{CC} = 2.7V, V_{OL} = 0.4V	0.7			mA
TRI-STATE Leakage	V_{CC} = 5.5V	-2		+2	µA
Allowable Sink/Source Current per Pin					
D Outputs (Sink)				15	mA
All others				3	mA
Maximum Input Current without Latchup (Note 4)	Room Temp			±200	mA
RAM Retention Voltage, Vr	500 ns Rise and Fall Time (min)	2			V
Input Capacitance	(Note 5)			7	pF
Load Capacitance on D2	(Note 5)			1000	pF

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

Parameter	Conditions	Min	Typ	Max	Units
Instruction Cycle Time (t_c)					
Crystal, Resonator	$4.5V \le V_{CC} \le 5.5V$	1.0		DC	µs
R/C Oscillator	$4.5V \le V_{CC} \le 5.5V$	3.0		DC	µs
Inputs					
t_{SETUP}	$4.5 \le V_{CC} \le 5.5V$	200			ns
t_{HOLD}	$4.5V \le V_{CC} \le 5.5V$	60			ns
Output Propagation Delay (Note 5)	$R_L = 2.2k$, $C_L = 100$ pF				
t_{PD1}, t_{PD0}					
SO, SK	$4.5V \le V_{CC} \le 5.5V$			0.7	µs
All Others	$4.5V \le V_{CC} \le 5.5V$			1	µs
MICROWIRE Setup Time (t_{UWS}) (Note 5)	$Vcc \ge 4.5V$	20			ns
MICROWIRE Hold Time (t_{UWH}) (Note 5)	$Vcc \ge 4.5V$	56			ns
MICROWIRE Output Propagation Delay (t_{UPD})	$Vcc \ge 4.5V$			220	ns
Input Pulse Width (Note 6)					
Interrupt Input High Time		1			t_c
Interrupt Input Low Time		1			t_c
Timer 1,2, 3 Input High Time		1			t_c
Timer 1,2, 3 Input Low Time		1			t_c
Reset Pulse Width		1			µs

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

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

Note 3: The HALT mode will stop CKI from oscillating in the RC and the Crystal configurations. Measurement of IDD HALT is done with device neither sourcing or 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 Vcc; 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: Pins G6 and $\overline{\text{RESET}}$ are designed with a high voltage input network. These pins allow input voltages greater than V_{CC} and the pins will have sink current to Vcc when biased at voltages greater than Vcc (the pins do not have source current when biased at a voltage below V_{CC}.) The effective resistance to Vcc is 750 ohms (typical). These two pins will not latch up. The voltage at the pins must be limited to less than 14 Volts. **WARNING: Voltages in excess of 14 volts will cause damage to the pins. This warning excludes ESD transients.**

Note 5: The output propagation delay is referenced to the end of the instruction cycle where the output change occurs.

Note 6: Parameter characterized but not tested.

Note 7: tc = Instruction Cycle Time.

Comparator AC and DC Characteristics V_{CC} = 5V, -40˚C ≤ T_A ≤ +85˚C

Parameter	Conditions	Min	Typ	Max	Units
Input Offset Voltage	0.4V <Vin<Vcc-1.5V		10	25	mV
Input Common Mode Voltage Range (Note 8)		0.4		Vcc–1.5	V
Voltage Gain			300k		V/V
Vcc/2 reference	4.0V<Vcc<5.5V	0.5Vcc–0.04	0.5Vcc	0.5Vcc +0.04	V
DC Supply Current For Comparator (when enabled)	Vcc=5.5V			250	µA
DC Supply Current For Vcc/2 reference (when enabled)	Vcc=5.5V		50	80	µA
DC Supply Current For Constant Current Source (when enabled)	Vcc=5.5V			200	µA
Constant Current Source	4.0V<Vcc<5.5V	7	20	32	µA
Current Source Variation	4.0V<Vcc<5.5V Temp = Constant			2	µA
Current Source Enable Time			1.5	2	µS
Comparator Response Time	10mV overdrive, 100pF 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.

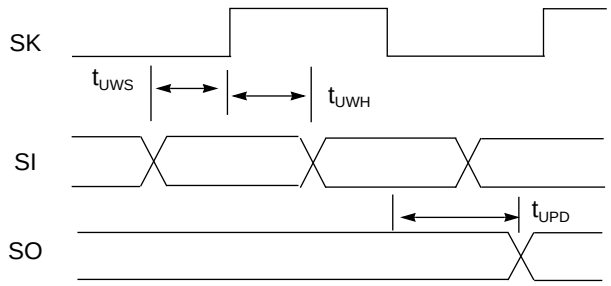


Figure 2. MICROWIRE/PLUS Timing

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.

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

The device contains two bidirectional (one 8-bit, one 4-bit) I/O ports (G and L), where each individual bit may be independently configured as a weak pullup input, TRI-STATE (Hi-Z) input or push pull output under program control. Ports G- and L- feature Schmitt trigger inputs. Three data memory address locations are allocated for each of these I/O ports. Each I/O port has two associated 8-bit memory mapped registers, the CONFIGURATION register and the output DATA register. A memory mapped address is also reserved for the input pins of each I/O port. (See the memory map for the various addresses associated with the I/O ports.) Figure 3 shows the I/O port configurations. The DATA and CONFIGURATION registers allow for each port bit to be individually configured under software control as shown below:

PORT L is an 4-bit I/O port. All L-pins have Schmitt triggers on the inputs.

The Port L supports Multi-Input Wake Up on all four pins. The Port L has the following alternate features:

L4 MIWU or external interrupt
L5 MIWU or external interrupt
L6 MIWU or external interrupt
L7 MIWU or external interrupt

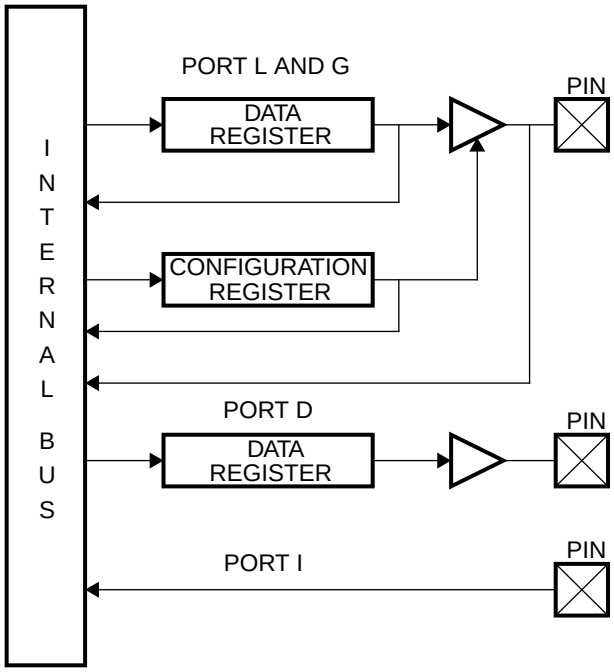


Figure 3. I/O Port Configurations

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

Please note: The lower 4 L-bits read all ones (L0:L3). This is independent from the states of the associated bits in the L-port Data- and Configuration register. The lower 4 bits in the L-port Data- and Configuration register can be used as general purpose status indicators (flags).

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 below. Reading the G6 and G7 data bits will return zeros.

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

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

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

Port G has the following alternate features:

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

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 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 Analog CH1 (Comparator Positive Input 1)
I1 I_{SRC} (Comparator Negative Input/Current Source Out)
I2 Analog CH2 (Comparator Positive Input 2)
I3 Analog CH3 (Comparator Positive Input 3/Comparator Output)
I4 Analog CH4 (Comparator Positive Input 4)
I5 Analog CH5 (Comparator Positive Input 5)
I6 Analog CH6 (Comparator Positive Input 6)
I7 C_{OUT} (Comparator Output)

Port D is a 4-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.

Functional Description

The architecture of the device is a 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 the 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 (tc) 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.

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 4096 bytes of OTP EPROM. These bytes may hold program instructions or constant data (data tables for the LAID instruction, jump vectors for the JID instruction, and interrupt vectors for the VIS instruction). The program memory is addressed by the 15-bit program counter (PC). All interrupts in the device vector to program memory location 0FF Hex.

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

SECURITY FEATURE

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

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

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

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

Note: RAM contents are undefined upon power-up.

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 and G 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 and CNTRL-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

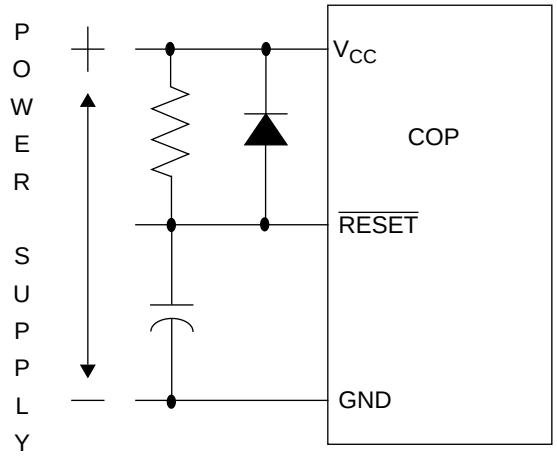


Figure 4. Recommended Reset Circuit

will cause an active low error output on pin G1. This error output will continue until 16 t_C–32 t_C clock cycles following the clock frequency reaching the minimum specified value, at which time the G1 output will enter the TRI-STATE mode.

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

WARNING:

When the device is held in reset for a long time, it will consume high current (typically about 7 mA). This is not true for the equivalent ROM device (COP8ACC5)

Oscillator Circuits

The chip can be driven by a clock input on the CKI input pin which can be between DC and 10MHz. 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 (t_c).

Figure 5 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 1 shows the component values required for various standard crystal values.

Table 1 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

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

Table 2 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 ≤ R ≤ 200k
50 pF ≤ C ≤ 200 pF

Control Registers

CNTRL Register (Address X'00EE)

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

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

T1C3	T1C2	T1C1	T1C0	MSEL	IEDG	SL1	SL0

Bit 7 Bit 0

PSW Register (Address X'00EF)

The PSW register contains the following select bits:

GIE Global interrupt enable (enables interrupts)

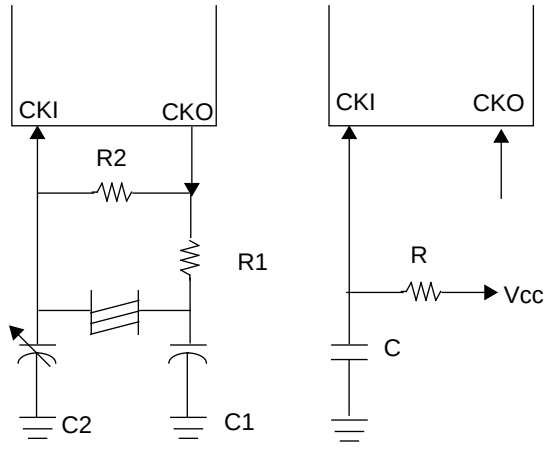


Figure 5. Crystal and R/C Oscillator Diagrams

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 flag 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

Timers

The device contains a very versatile set of timers (T0 and T1). 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, tc. 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

Figure 6 is a functional block diagram showing the structure of the IDLE Timer and its associated interrupt logic.

Bits 11 through 15 of the ITMR register can be selected for triggering the IDLE Timer interrupt. Each time the selected bit underflows (every 4k, 8k, 16k, 32k or 64k instruction cycles), the IDLE Timer interrupt pending bit T0PND is set, thus generating an interrupt (if enabled), and bit 6 of the Port G data register is reset, thus causing an exit from the IDLE mode if the device is in that mode.

In order for an interrupt to be generated, the IDLE Timer interrupt enable bit T0EN must be set, and the GIE (Global Interrupt Enable) bit must also be set. The T0PND flag and T0EN bit are bits 5 and 4 of the ICNTRL register, respectively. The interrupt can be used for any purpose. Typically, it is used to perform a task upon exit from the IDLE mode. For more information on the IDLE mode, refer to the Power Save Modes section.

The Idle Timer period is selected by bits 0-2 of the ITMR register Bits 3-7 of the ITMR Register are reserved and should not be used as software flags.

Table 3 Idle Timer Window Length

ITSEL2	ITSEL1	ITSEL0	Idle Timer Period (Instruction Cycles)
0	0	0	4,096
0	0	1	8,192
0	1	0	16,384
0	1	1	32,768
1	X	X	65,536

The ITMR register is cleared on Reset and the Idle Timer period is reset to 4,096 instruction cycles.

ITMR Register (Address X'0xCF)

Reserved	ITSEL2	ITSEL1	ITSEL0

Bit 7 Bit 3 Bit 0

Any time the IDLE Timer period is changed there is the possibility of generating a spurious IDLE Timer interrupt by setting the T0PND bit. The user is advised to disable IDLE Timer interrupts prior to changing the value of the ITSEL bits of the ITMR Register and then clear the T0PND bit before attempting to synchronize operation to the IDLE Timer.

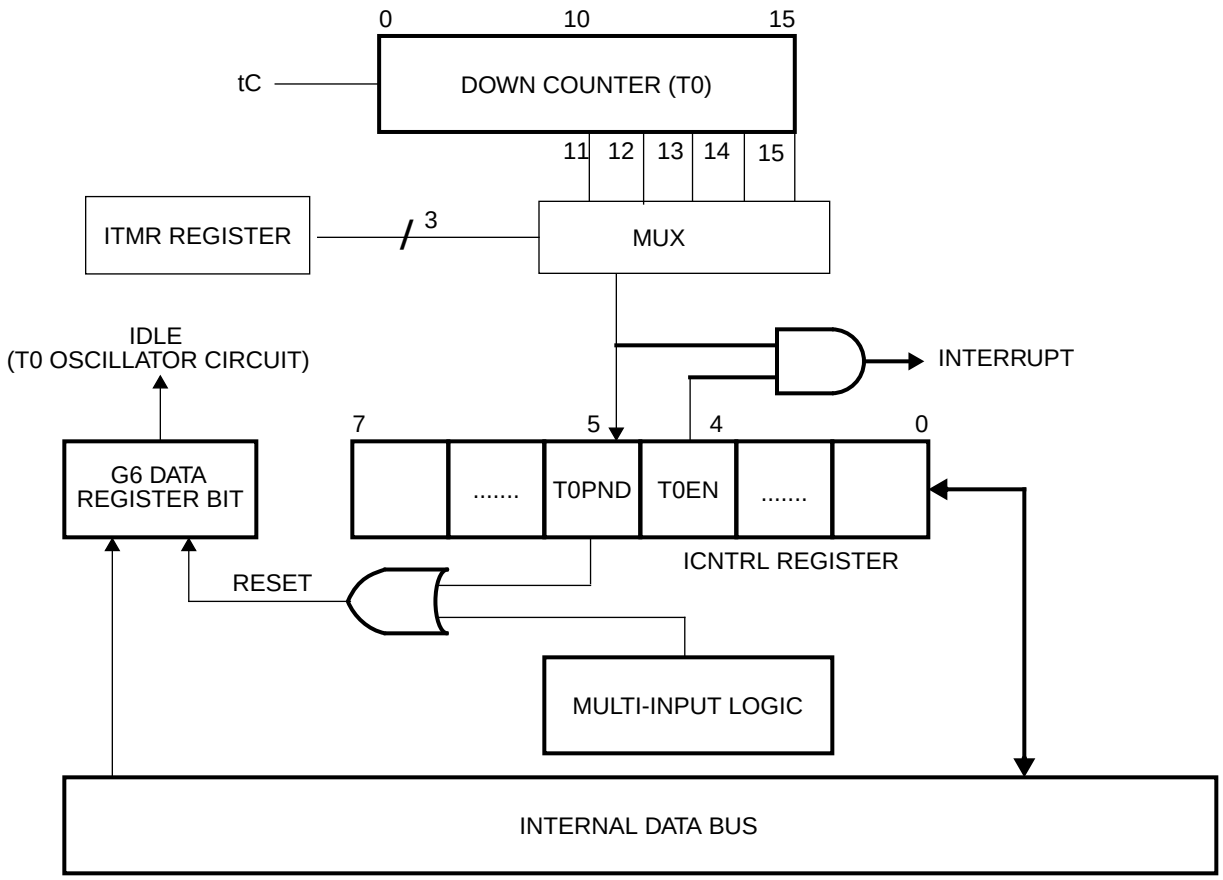


Figure 6. Functional Block Diagram for Idle Timer T0

TIMER T1

The device has a powerful timer/counter block. The timer consists of a 16-bit timer, T1, and two supporting 16-bit autoreload/capture registers, R1A and R1B. The timer block has two pins associated with it, T1A and T1B. The pin T1A supports I/O required by the timer block, while the pin T1B 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 T1C3, T1C2, and T1C1 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 T1 counts down at a fixed rate of tc. Upon every underflow the timer is alternately reloaded with the contents of supporting registers, R1A and R1B. The very first underflow of the timer causes the timer to reload from the register R1A. Subsequent underflows cause the timer to be reloaded from the registers alternately beginning with the register R1B.

The T1 Timer control bits, T1C3, T1C2 and T1C1 set up the timer for PWM mode operation.

Figure 7 shows a block diagram of the timer in PWM mode.

The underflows can be programmed to toggle the T1A output pin. The underflows can also be programmed to generate interrupts.

Underflows from the timer are alternately latched into two pending flags, T1PNDA and T1PNDB. The user must reset these pending flags under software control. Two control enable flags, T1ENA and T1ENB, allow the interrupts from the timer underflow to be enabled or disabled. Setting the timer enable flag T1ENA will cause an interrupt when a timer underflow causes the R1A register to be reloaded into the timer. Setting the timer enable flag T1ENB will cause an interrupt when a timer underflow causes the R1B register to be reloaded into the timer. Resetting the timer enable flags will disable the associated interrupts.

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

Mode 2. External Event Counter Mode

This mode is quite similar to the processor independent PWM mode previously described. The main difference is that the timer, T1, is clocked by the input signal from the T1A pin. The T1 timer control bits, T1C3, T1C2 and T1C1 allow the timer to be clocked either on a positive or negative edge from the T1A pin. Underflows from the timer are latched into the T1PNDA pending flag. Setting the T1ENA control flag will cause an interrupt when the timer underflows.

In this mode the input pin T1B can be used as an independent positive edge sensitive interrupt input if the T1ENB control flag is set. The occurrence of a positive edge on the T1B input pin is latched into the T1PNDB flag.

Figure 8 shows a block diagram of the timer in External Event Counter mode.

Note: The PWM output is not available in this mode since the T1A pin is being used as the counter input clock.

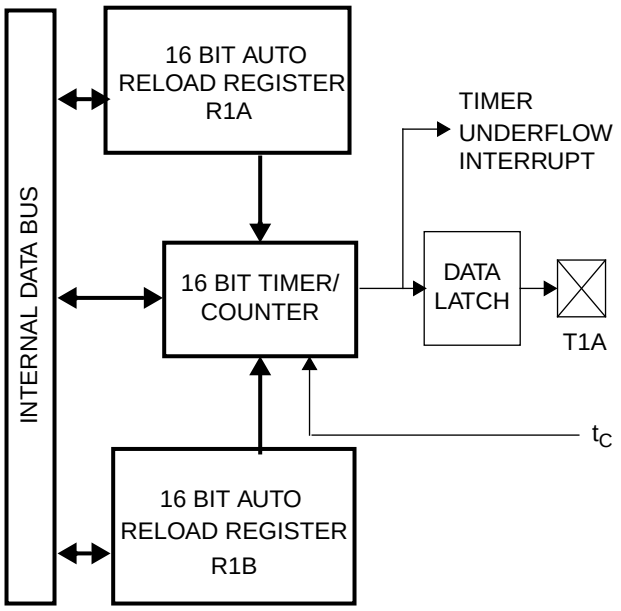


Figure 7. Timer in PWM Mode

Mode 3. Input Capture Mode

The device can precisely measure external frequencies or time external events by placing the timer block, T1, in the input capture mode.

In this mode, the timer T1 is constantly running at the fixed tc rate. The two registers, R1A and R1B, act as capture registers. Each register acts in conjunction with a pin. The register R1A acts in conjunction with the T1A pin and the register R1B acts in conjunction with the T1B pin.

The timer value gets copied over into the register when a trigger event occurs on its corresponding pin. Control bits, T1C3, T1C2 and T1C1, 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 T1A and T1B pins will be respectively latched into the pending flags, T1PNDA and T1PNDB. The control flag T1ENA allows the interrupt on T1A to be either enabled or disabled. Setting the T1ENA flag enables interrupts to be generated when the selected trigger condition occurs on the T1A pin. Similarly, the flag T1ENB controls the interrupts from the T1B pin

Underflows from the timer can also be programmed to generate interrupts. Underflows are latched into the timer T1C0 pending flag (the T1C0 control bit serves as the timer underflow interrupt pending flag in the Input Capture mode). Consequently, the T1C0 control bit should be reset when entering the Input Capture mode. The timer underflow interrupt is enabled with the T1ENA control flag. When a T1A interrupt occurs in the Input Capture mode, the user must check both the T1PNDA and T1C0 pending flags in order to determine whether a T1A input capture or a timer underflow (or both) caused the interrupt.

Figure 9 shows a block diagram of the timer in Input Capture mode.

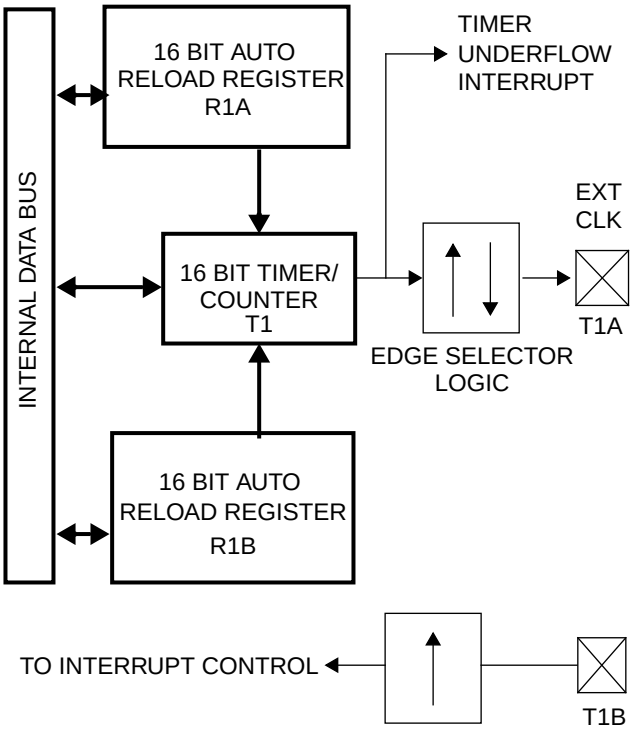


Figure 8. Timer in External Event Counter Mode

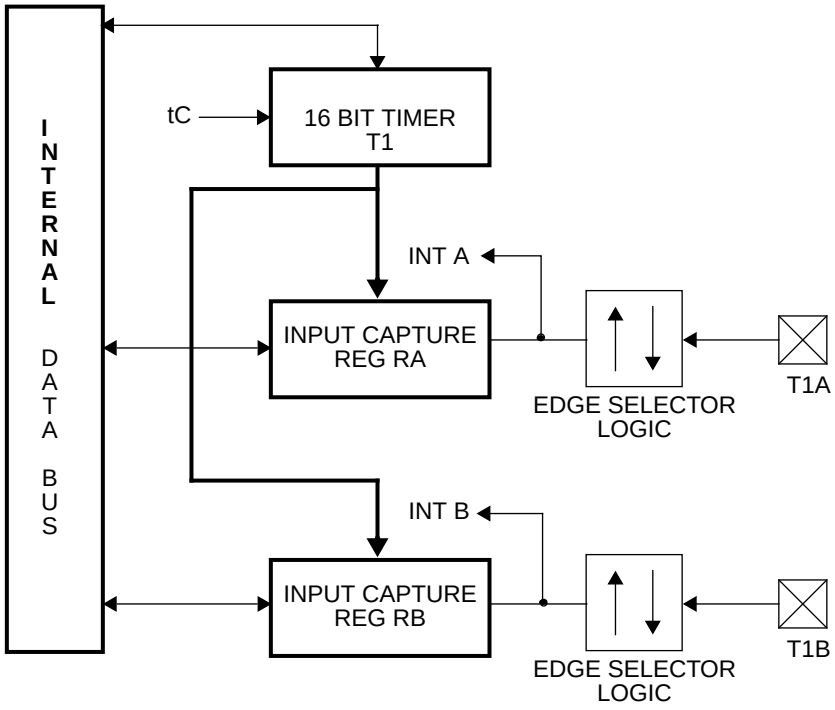


Figure 9. Timer in Input Capture Mode

TIMER CONTROL FLAGS

The Timer T1 control bits and their functions are summarized below.

T1C0 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)

T1PNDA Timer Interrupt Pending Flag
T1PNDB Timer Interrupt Pending Flag
T1ENA Timer Interrupt Enable Flag
T1ENB Timer Interrupt Enable Flag
1 = Timer Interrupt Enabled
0 = Timer Interrupt Disabled
T1C3 Timer mode control (see Table 4)
T1C2 Timer mode control
T1C1 Timer mode control

The timer mode control bits (T1C3, T1C2 and T1C1) are detailed below:

Table 4 Timer T1 Mode Control

T1C3	T1C2	T1C1	Timer Mode	Interrupt A Source	Interrupt B Source	Timer Counts On
0	0	0	MODE 2 (External Event Counter)	Timer Underflow	Pos. T1B Edge	T1A Pos. Edge
0	0	1	MODE 2 (External Event Counter)	Timer Underflow	Pos. T1B Edge	T1A Neg. Edge
1	0	1	MODE 1 (PWM) T1A Toggle	Autoreload RA	Autoreload RB	t_c
1	0	0	MODE 1 (PWM) No T1A Toggle	Autoreload RA	Autoreload RB	t_c
0	1	0	MODE 3 (Capture) Captures: T1A Pos. Edge T1B Pos. Edge	Pos. T1A Edge or Timer Underflow	Pos. T1B Edge	t_c
1	1	0	MODE 3 (Capture) Captures: T1A Pos. Edge T1B Neg. Edge	Pos. T1A Edge or Timer Underflow	Neg. T1B Edge	t_c
0	1	1	MODE 3 (Capture) Captures: T1A Neg. Edge T1B Pos. Edge	Neg. T1A Edge or Timer Underflow	Pos. T1B Edge	t_c
1	1	1	MODE 3 (Capture) Captures: T1A Neg. Edge T1B Neg. Edge	Neg. T1A Edge or Timer Underflow	Neg. T1B Edge	t_c

HIGH SPEED CAPTURE TIMER

The device provides a 16-bit high-speed capture timer. The timer consists of a 16-bit up-counter that is clocked with the device clock input frequency (CKI) and an 8-bit control register. The 16-bit counter is mapped as two read/write 8-bit registers. This timer is specifically designed to be used in conjunction with the Analog Function Block (comparator, analog multiplexer, constant current source) to implement a low-cost, high-resolution, single-slope A/D.

The timer is automatically stopped in the event of a capture to allow the software to read the timer value. Coming out of reset the counter is disabled (stopped) and reads all `0'.

Setting the Capture Timer Run bit CAPRUN bit in the Capture Control Register (CAPCNTL) will start the counter. The counter will count up until a capture event (negative edge) is received. Upon a capture the counter will be stopped, the Capture Pending bit (CAPPND) is set, and the CAPRUN bit is automatically reset. If capture interrupts are enabled (CAPIEN=1), the capture event will generate an interrupt. Setting the CAPRUN bit again by software will start a new counting cycle. If the Capture Mode bit is reset (CAPMOD=0) the capture timer will be automatically initialized to all `0' with each setting of the CAPRUN bit. If CAPMOD=1 the timer will not be cleared when setting the CAPRUN bit, thus allowing the user's software to pre-load the timer registers with any desired value. This mode can be used in conjunction with the timer's overflow to implement for example a programmable delay counter.

"CAPTURE MODE" is only active when the CAPRUN bit is set, i.e. any capture events received while the timer is stopped (CAPRUN=0) will be ignored and will not cause the CAPPND bit to be set. The capture counter can also be stopped (frozen) by the user's software resetting the CAPRUN bit.

If the user program tries to set the CAPRUN bit at the same time that the hardware gets a capture event and tries to reset the CAPRUN bit, the hardware will have precedence.

Should the counter overflow before a capture condition occurs, the Capture Overflow bit (CAPOVL) bit in the CAPCNTL register will be set. If Capture interrupts are enabled (CAPIEN=1) an overflow will generate an interrupt. It is the user's software which should reset this bit before the next overflow occurs, otherwise subsequent overflow conditions cannot be detected.

Capture Overflow interrupt and Capture Pending interrupt share the same interrupt vector.

CAPCNTL Register (Address X'CE)

The CAPCNTL register contains the following bits:

CAPIEN Capture Interrupts enable,
1= enable interrupt, 0= disable interrupts

CAPPND Capture pending.
Gets automatically set when a capture event occurs. If CAPIEN=1 an interrupt is generated. Has to be reset by the user's software.

CAPOVL Capture Timer Overflow. Gets set to "1" upon timer overflow. Has to be reset by user's software. If CAPIEN=1 an interrupt is generated.

CAPRUN Capture Timer Run. Setting this bit to one will start the capture timer. This bit gets automatically reset to '0' when a capture event occurs. Writing a '0' by software will also reset the bit and stop the timer.

CAPMOD Reset Time.
0: reset timer to '0' when CAPRUN bit gets set
1: DO NOT reset timer to '0' when CAPRUN bit gets set.

UNUSED Can be used as general purpose flags by user's software.

Unused	CAPMOD	CAPRUN	CAPOVL	CAPPND	CAPIEN
Bit 7 Bit 5					Bit 0

Power Save Modes

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

HALT MODE

The device 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 on the device 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 Port L.

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 only 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 tc instruction cycle clock. The tc clock is derived by dividing the oscillator clock down by a factor of 10. The Schmitt trigger following the CKI inverter on the

chip ensures that the 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 start-up time-out from the IDLE timer enables the clock signals to be routed to the rest of the chip.

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

The 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

In the IDLE mode, program execution stops and power consumption is reduced to a very low level as with the HALT mode. However, the on-board oscillator, IDLE Timer (Timer T0), and Clock Monitor continue to operate, allowing real time to be maintained. The device remains idle for a selected amount of time up to 65,536 instruction cycles, or 65.536 milliseconds with a 1 MHz instruction clock frequency, and then automatically exits the IDLE mode and returns to normal program execution.

The device is placed in the IDLE mode under software control by setting the IDLE bit (bit 6 of the Port G data register).

The IDLE timer window is selectable from one of five values, 4k, 8k, 16k, 32k or 64k instruction cycles. Selection of this value is made through the ITMR register.

The IDLE mode uses the on-chip IDLE Timer (Timer T0) to keep track of elapsed time in the IDLE state. The IDLE Timer runs continuously at the instruction clock rate, whether or not the device is in the IDLE mode. Each time the bit of the timer associated with the selected window toggles, the T0PND bit is set, an interrupt is generated (if enabled), and the device exits the IDLE mode if in that mode. If the IDLE timer interrupt is enabled, the interrupt is serviced before execution of the main program resumes. (However, the instruction which was started as the part entered the IDLE mode is completed before the interrupt is serviced. This instruction should be a NOP which should follow the enter IDLE instruction.) The user must reset the IDLE timer pending flag (T0PND) before entering the IDLE mode.

As with the HALT mode, this device can also be returned to normal operation with a reset, or with a Multi-Input Wakeup input. Upon reset the ITMR register is cleared and the ITMR register selects the 4,096 instruction cycle tap of the Idle Timer.

The IDLE timer cannot be started or stopped under software control, and it is not memory mapped, so it cannot be read or written by the software. Its state upon Reset is unknown. Therefore, if the device is put into the IDLE mode at an arbitrary time, it will stay in the IDLE mode for somewhere between 1 and the selected number of instruction cycles.

In order to precisely time the duration of the IDLE state, entry into the IDLE mode must be synchronized to the state of the IDLE Timer. The best way to do this is to use the IDLE Timer interrupt, which occurs on every underflow of the bit of the IDLE Timer which is associated with the selected window. Another method is to poll the state of the IDLE Timer pending bit T0PND, which is set on the same occurrence. The Idle Timer interrupt is enabled by setting bit T0EN in the ICNTRL register.

Any time the IDLE Timer window length is changed there is the possibility of generating a spurious IDLE Timer interrupt by setting the T0PND bit. The user is advised to disable IDLE Timer interrupts prior to changing the value of the ITSEL bits of the ITMR Register and then clear the TOPND bit before attempting to synchronize operation to the IDLE Timer.

Note: As with the HALT mode, it is necessary to program two NOP's to allow clock resynchronization upon return from the IDLE mode. The NOP's are placed either at the beginning of the IDLE timer interrupt routine or immediately following the "enter IDLE mode" instruction.

For more information on the IDLE Timer and its associated interrupt, see the description in the Timers section.

Multi-Input Wakeup

The Multi-Input Wakeup feature is used 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 4 edge selectable external interrupts.

Figure 10 shows the Multi-Input Wakeup logic.

The Multi-Input Wakeup feature utilizes the L Port. The user selects which particular L port bit (or combination of L Port bits) will cause the device to exit the HALT or IDLE modes. The selection is done through the register WKEN. The register 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 register 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 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
```

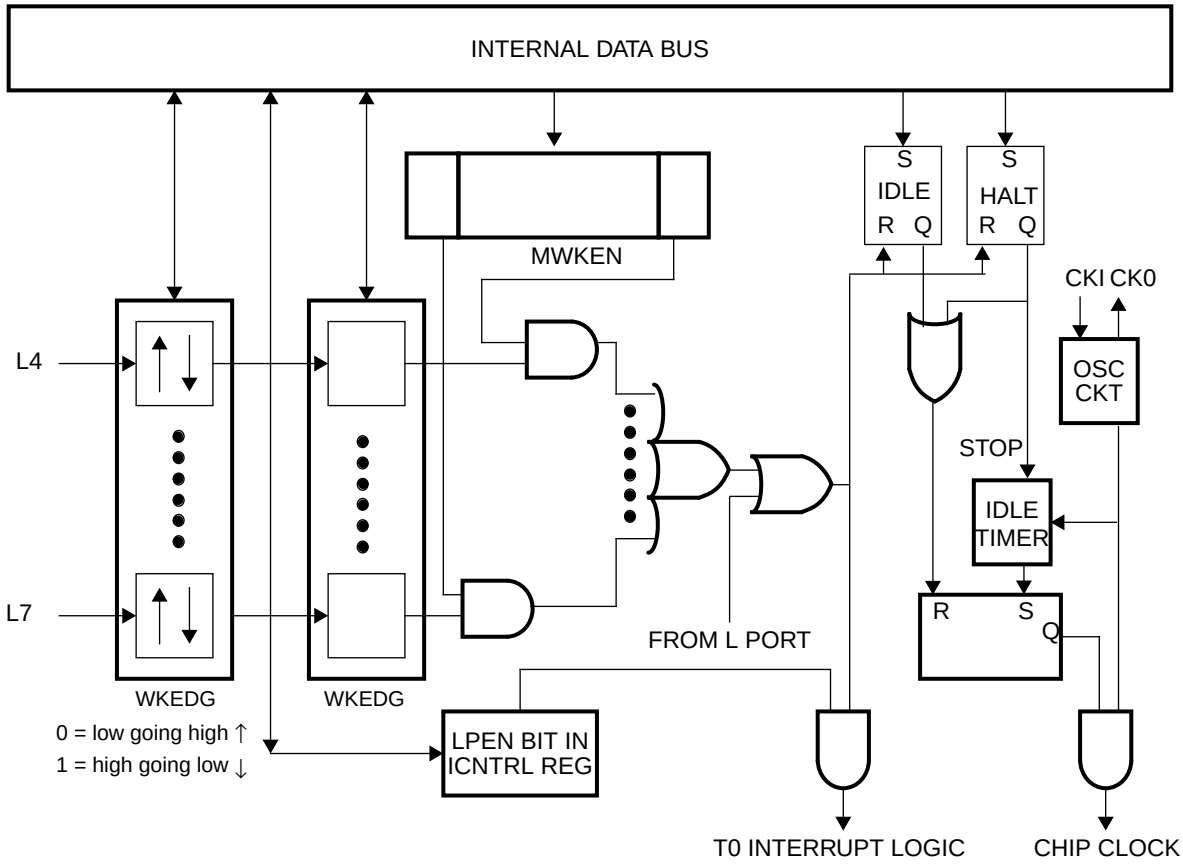


Figure 10. Multi-Input Wake Up Logic

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 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 must clear 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

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 5 for more information.)

CMPT2B Selects the “High Speed 16-Bit Capture Timer” input to be driven directly by the comparator output. If the comparator is disabled (CMPEN=0), this function is disabled, i.e. the capture timer input is connected to GND.

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 Vcc/2 reference, disconnect the comparator output from the capture timer input and pin I3/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 0xD7).

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 5 Comparator Input Selection

Control Bit			Comparator Input Source			Comparator Output
CMPISEL2	CMPISEL1	CMPISEL0	Neg. Input	Pos. Input		
0	0	0	I1	I2	CH2	I3
0	0	1	I1	I2	CH2	I7
0	1	0	I1	I3	CH3	I7
0	1	1	I1	I0	CH1	I7
1	0	0	I1	I4	CH4	I7
1	0	1	I1	I5	CH5	I7
1	1	0	I1	I6	CH6	I7
1	1	1	I1	Vcc/2 Ref.		I7

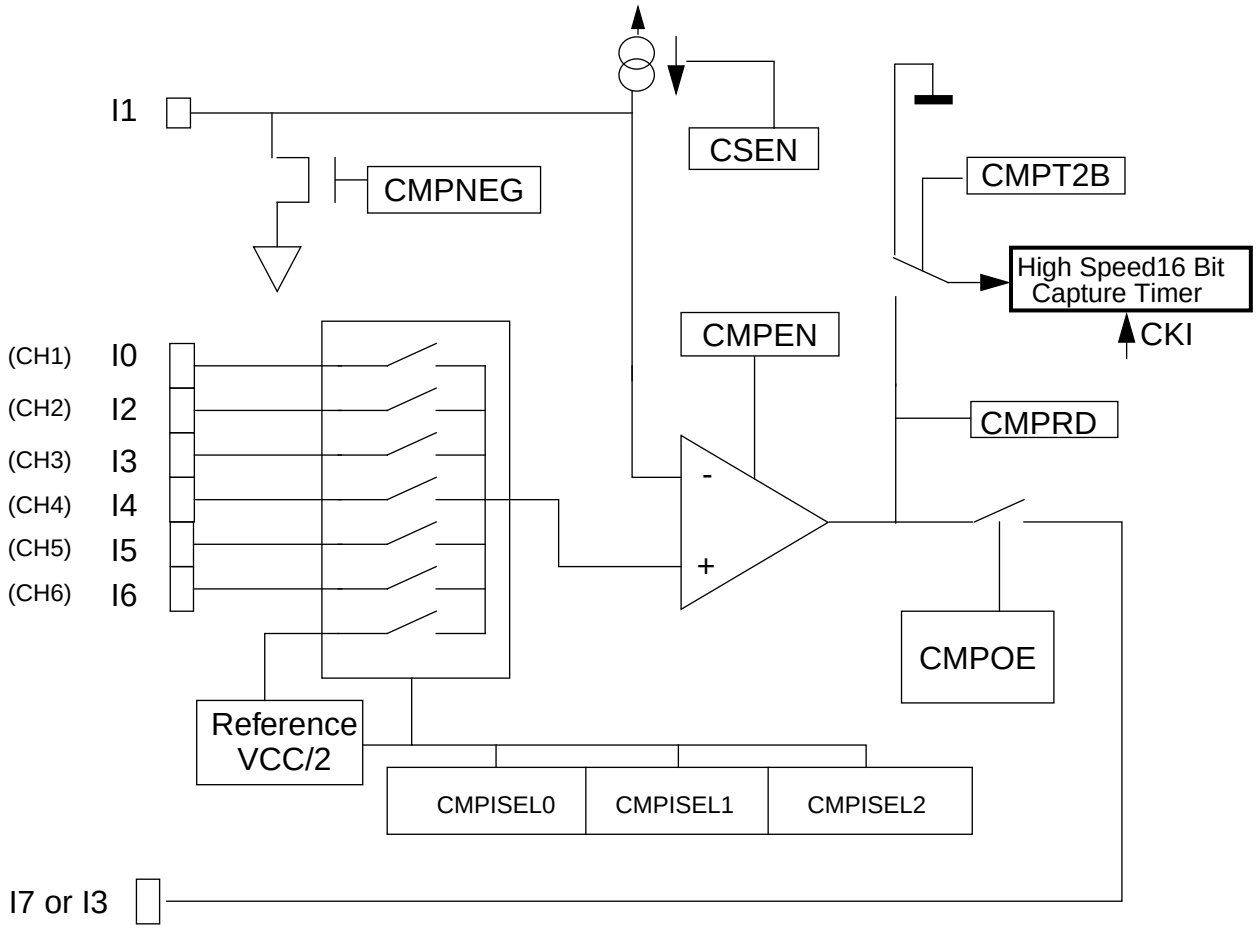


Figure 11. Analog Function Block

Reset

The state of the Analog 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 capture timer input is connected to GND
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 twelve interrupt sources. The following table lists all the possible device interrupt sources, their arbitration rankings 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 one or more Pending bits. 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.

Table 6 Interrupt Vector Table

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	Idle Timer	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)	High Speed Capture Timer	Capture Overflow/ Capture Pending	0yEA - 0yEB
(12)	Reserved		0yE8 - 0yE9
(13)	Reserved		0yE6 - 0yE7
(14)	Reserved		0yE4 - 0yE5
(15)	Port L/Wakeup	Port L Edge	0yE2 - 0yE3
(16) Lowest	Default VIS	Reserved	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.

WARNING

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

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

Figure 12 shows the Interrupt block diagram.

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 devices contain 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.

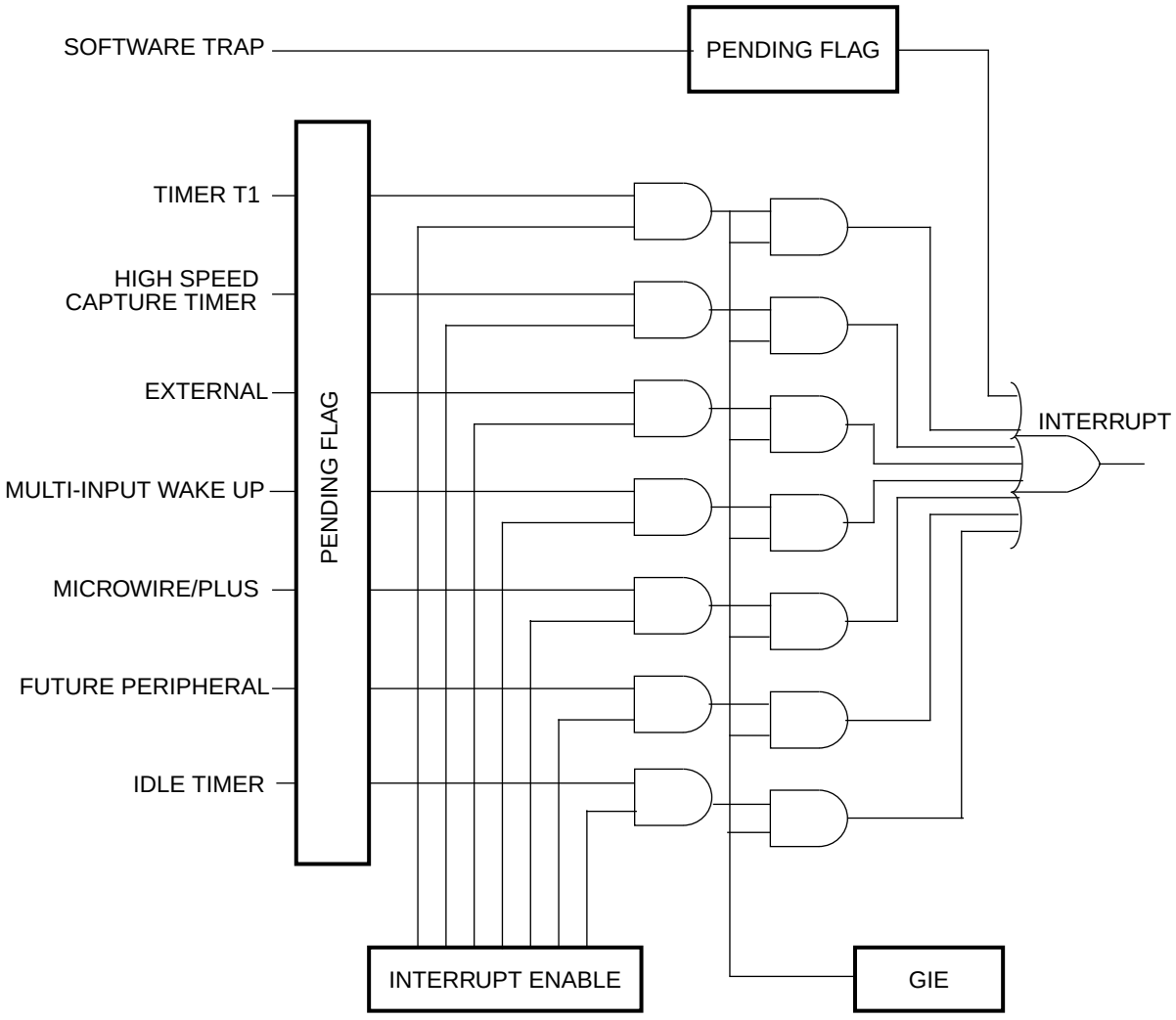


Figure 12. Interrupt Block Diagram

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 7 shows the WDSVR register.

Table 7 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 8 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 8 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 9 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 tc–32 tc 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 VCC through a resistor in order to pull WDOUT high.

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

The Clock Monitor forces the G1 pin low upon detecting a clock frequency error. The Clock Monitor error will continue until the clock frequency has reached the minimum specified value, after which the G1 output will enter the high impedance TRI-STATE mode following 16 tc–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 00. 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), and all other segments (i.e., Segments 4... 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:

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

When the software interrupt occurs, the user can re-initialize the stack pointer and do a recovery procedure before 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.

Table 9 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

MICROWIRE/PLUS

MICROWIRE/PLUS is a serial synchronous communications interface. The MICROWIRE/PLUS capability enables the device to interface with any of National Semiconductor's MICROWIRE peripherals (i.e. A/D converters, display drivers, E2PROMs etc.) and with other microcontrollers which support the MICROWIRE interface. It consists of an 8-bit serial shift register (SIO) with serial data input (SI), serial data output (SO) and serial shift clock (SK). Figure 13 shows a block diagram of the MICROWIRE/PLUS logic.

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 10 details the different clock rates that may be selected.

Table 10 MICROWIRE/PLUS Master Mode Clock Select

SL1	SL0	SK period
0	0	2 X t_c
0	1	4 X t_c
1	x	8 X 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 14 shows how two devices, microcontrollers and several peripherals may be interconnected using the MICROWIRE/PLUS arrangements.

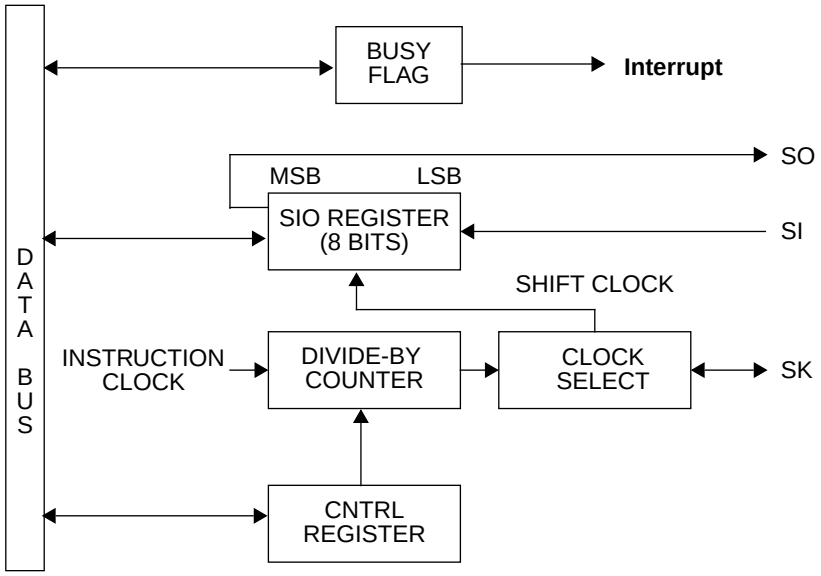


Figure 13. MICROWIRE/PLUS Block Diagram

Warning:

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

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

MICROWIRE/PLUS Master Mode Operation

In the MICROWIRE/PLUS Master mode of operation the shift clock (SK) is generated internally. The MICROWIRE Master always initiates all data exchanges. The MSEL bit in the CNTRL register must be set to enable the SO and SK functions onto the G Port. The SO and SK pins must also be selected as outputs by setting appropriate bits in the Port G configuration register. Table 11 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 11 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.

Table 11 MICROWIRE Mode Settings

G4 (SO) Config. Bit	G5 (SK) Config. Bit	G4 Fun.	G5 Fun.	Operation
1	1	SO	Int. SK	MICROWIRE/PLUS Master
0	1	TRI-STATE	Int. SK	MICROWIRE/PLUS Master
1	0	SO	Ext. SK	MICROWIRE/PLUS Slave
0	0	TRI-STATE	Ext. SK	MICROWIRE/PLUS Slave
This table assumes that the control flag MSEL is set.				

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.

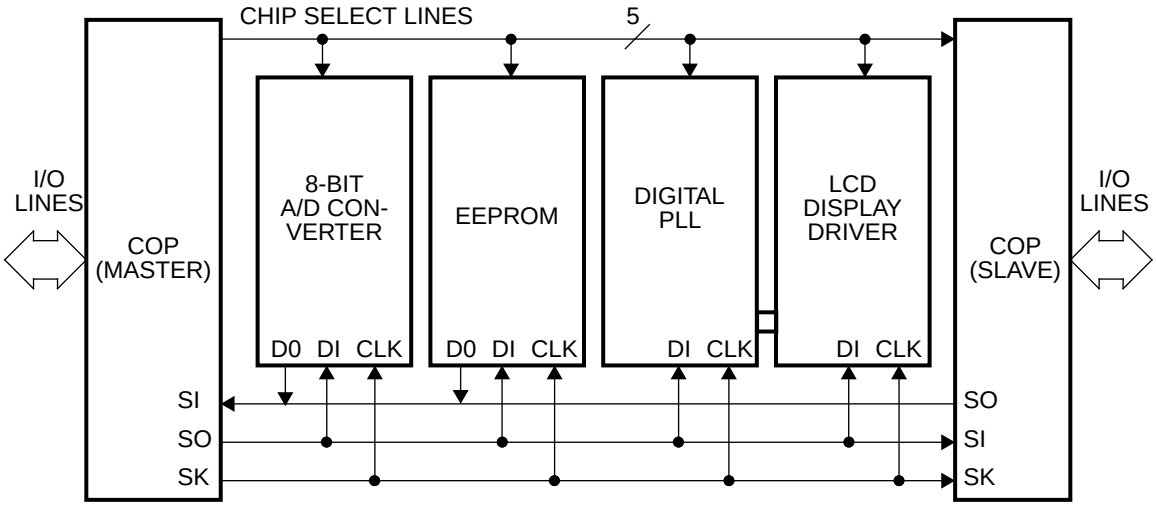


Figure 14. MICROWIRE/PLUS Application

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	Reserved
XXB1	Reserved
xxB2	Reserved
xxB3	Reserved
xxB4	Reserved
xxB5	Reserved
xxB6	Reserved
xxB7	Comparator Select Register (CMPSL)
xxB8 to xxBF	Reserved
xxC0	Reserved
xxC1	Reserved
xxC2	Reserved
xxC3	Reserved
xxC4	Reserved
xxC5	Reserved
xxC6	Reserved
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	CAPTLO (Capture Timer Low-Byte)
xxCD	CAPTHI (Capture Timer High-Byte)
xxCE	CAPCNTL (Capture Timer Control Register)
xxCF	Idle Timer Control Register
xxD0	Port L Data Register
xxD1	Port L Configuration Register
xxD2	Port L Input Pins (Read Only)
xxD3	Reserved
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	Reserved
xxD9	Reserved
xxDA	Reserved
xxDB	Reserved
xxDC	Port D
xxDD to DF	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	Reserved
0100–017F	Reserved

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 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.

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

Instruction Set

Register and Symbol Definition

Registers	
A	8-Bit Accumulator Register
B	8-Bit Address Register
X	8-Bit Address Register
SP	8-Bit Stack Pointer Register
PC	15-Bit Program Counter Register
PU	Upper 7 Bits of PC
PL	Lower 8 Bits of PC
C	1 Bit of PSW Register for Carry
HC	1 Bit of PSW Register for Half Carry
GIE	1 Bit of PSW Register for Global Interrupt Enable
VU	Interrupt Vector Upper Byte
VL	Interrupt Vector Lower Byte
Symbols	
[B]	Memory Indirectly Addressed by B Register
[X]	Memory Indirectly Addressed by X Register
MD	Direct Addressed Memory
Mem	Direct Addressed Memory or [B]
Meml	Direct Addressed Memory or [B] or Immediate Data
Imm	8-Bit Immediate Data
Reg	Register Memory: Addresses F0 to FF (Includes B, X and SP)
Bit	Bit Number (0 to 7)
←	Loaded with
↔	Exchanged with

INSTRUCTION SET

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 - MemI + C, C ← Carry, HC ← Half Carry
AND	A,Meml	Logical AND	A ← A and $\overline{\text{Meml}}$
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 #,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 ←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], skip next instruction
RETI		RETurn from Interrupt	SP + 2, PL ← [SP],PU← [SP-1],GIE←1
INTR		Generate an Interrupt	[SP]← PL, [SP-1]← PU, SP-2, PC← 0FF
NOP		No OPeration	PC← PC + 1

Instruction Execution Time

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

Most single byte instructions take one cycle time to execute.

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

See the BYTES and CYCLES per INSTRUCTION table for details.

Bytes and Cycles per Instruction

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

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
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
DCORA	1/1
RRCA	1/1
RLCA	1/1
SWAPA	1/1
SC	1/1
RC	1/1
IFC	1/1
IFNC	1/1
PUSHA	1/3
POPA	1/3
ANDSZ	2/2

Transfer of Control Instructions

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

Memory Transfer Instructions

	Register Indirect		Direct	Immed.	Register Indirect Auto Incr & Decr		
	[B]	[X]			[B+, B-]	[X+, X-]	
X A,*	1/1	1/3	2/3		1/2	1/3	
LD A,*	1/1	1/3	2/3	2/2	1/2	1/3	
LD B,Imm				1/1			(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																
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	SUBC 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

LOWER NIBBLE

where,

i is the immediate data
Md is a directly addressed memory location
* is an unused opcode

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

Development Support

Summary

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

IceMASTER (IM) IN-CIRCUIT EMULATION

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

See Figure 15 for configuration.

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

- Real-time in-circuit emulation; full 2.4-5.5V operation range, full DC-10MHz clock. Chip options are programmable or jumper selectable.
- Direct connection to application board by package compatible socket or surface mount assembly.
- Full 32K byte of loadable programming space that overlays (replaces) the on-chip ROM or EPROM. On-chip RAM and I/O blocks are used directly or recreated on the probe as necessary.
- Full 4k frame synchronous trace memory. Address, instruction, and eight unspecified, circuit connectable trace lines. Display can be HLL source (e.g., C source), assembly or mixed.
- A full 64k hardware configurable break, trace on, trace off control, and pass count increment events.
- Tool set integrated interactive symbolic debugger - supports both assembler (COFF) and C Compiler (.COD) linked object formats.
- Real time performance profiling analysis; selectable bucket definition.
- Watch windows, content updated automatically at each execution break.
- Instruction by instruction memory / register changes displayed on source window when in single step operation.
- Single base unit & debugger software reconfigurable to support the entire COP8 family; only the probe personality needs to change. Debugger software is processor customized, and reconfigured from a master model file.
- Processor specific symbolic display of registers and bit level assignments, configured from master model file.
- Halt / Idle mode notification.
- On-line HELP customized to specific processor using master model file.
- Includes a copy of COP8-DEV-IBMA assembler & linker SDK.

IM Order-Information

Base Unit	
IM-COP8/400-1	iceMASTER base unit, 110V power supply
IM-COP8/400-2	iceMASTER base unit, 220V power supply
iceMASTER Probe	
COP8AC-IM20/28N	20 & 28 DIP
Surface Mount Adapter	
MHW-SOIC28 MHW-SOIC20	28 SO 20 SO

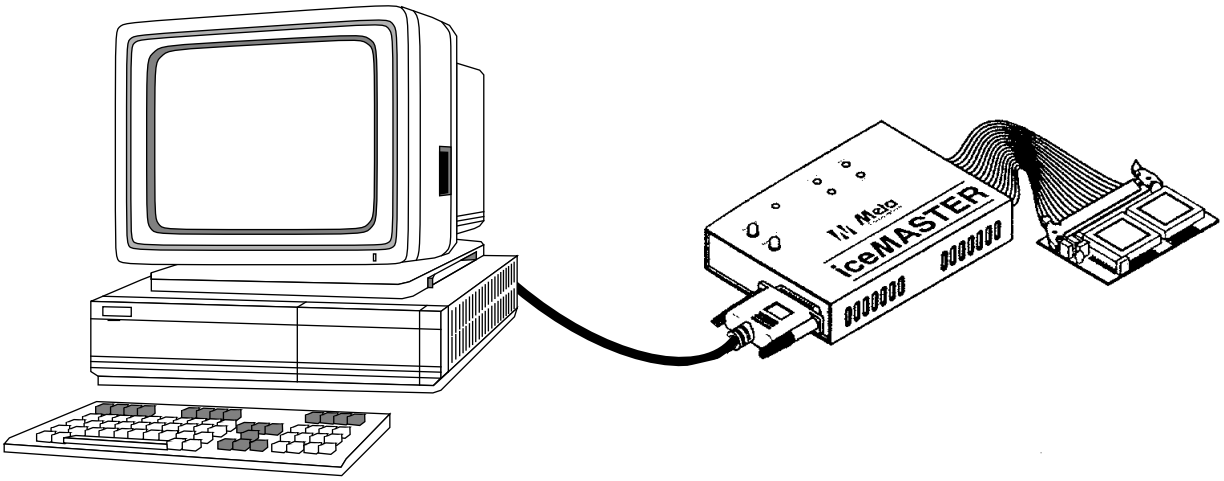


Figure 15. COP8 iceMASTER Environment

IceMASTER DEBUG MODULE (DM)

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

See Figure 16 for configuration.

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

- Real-time in-circuit emulation; full operating voltage range operation, full DC-10MHz clock.
- All processor I/O pins can be cabled to an application development board with package compatible cable to socket and surface mount assembly.
- Full 32 kbyte of loadable programming space that overlays (replaces) the on-chip ROM or EPROM. On-chip RAM and I/O blocks are used directly or recreated as necessary.
- 100 frames of synchronous trace memory. The display can be HLL source (C source), assembly or mixed. The most recent history prior to a break is available in the trace memory.
- configured break points; uses INTR instruction which is modestly intrusive.
- software - only supported features are selectable.
- Tool set integrated interactive symbolic debugger - supports both assembler (COFF) and C Compiler (.COD) SDK linked object formats.
- Instruction by instruction memory / register changes displayed when in single step operation.
- Debugger software is processor customized, and reconfigured from a master model file.
- Processor specific symbolic display of registers and bit level assignments, configured from master model file.
- Halt / Idle mode notification.
- Programming menu supports full product line of programmable OTP and EPROM COP8 products. Program data is taken directly from the overlay RAM.
- Programming of COP8ACC7xxx requires a secondary programming adapter which is supplied with the COP8AC-DM.
- Includes wallmount 5VDC power supply.
- On-board VPP generator from 5V input or connection to external supply supported. Requires VPP level adjustment per the family programming specification (correct level is provided on an on-screen pop-down display).
- On-line HELP customized to specific processor using master model file.
- Includes a copy of COP8-DEV-IBMA assembler & linker SDK.

DM Order-Information

Debug Module Unit	
COP8AC-DM*	
Cable Adapters	
DM-COP8/28D DM-COP8/20D	28 DIP 20 DIP
Surface Mount Adapters	
DM-COP8/28D-SO	28 DIP to SO
DM-COP8/20D-SO	20 DIP to SO
* The Debug Module Kit contains one set of adapters, DIP and SO. It also contains a programming adapter.	

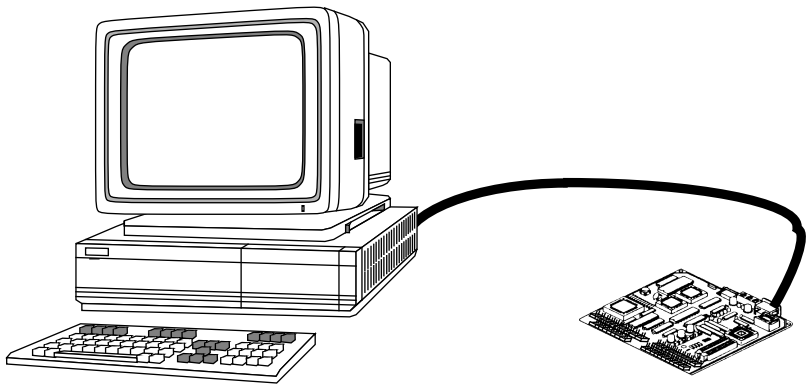


Figure 16. COP8-DM Environment

COP8 ASSEMBLER / LINKER SOFTWARE DEVELOPMENT TOOL KIT

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

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

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

Order-Information

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

COP8 C COMPILER

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

Features are summarized as follows:

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

SINGLE CHIP OTP/EMULATOR SUPPORT

The COP8 family is supported by single chip OTP emulators. For detailed information refer to the emulator specific datasheet and the emulator selection table below:

OTP Emulator Ordering Information

Device Number	Clock Option	Package	Emulates
COP8ACC720Mx	Crystal	20 SO	COP8ACC520Mx
COP8ACC728Nx	Crystal	28 DIP	COP8ACC528Nx
COP8ACC728Mx	Crystal	28 SO	COP8ACC528Mx
x = temp. range (8, 9)		(9 = 0˚C ≤ T_A ≤ +70˚C 8 = –40˚C ≤ T_A ≤ +85˚C)	

INDUSTRY WIDE OTP / EPROM PROGRAMMING SUPPORT

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

Visit National’s web site at www.national.com for up-to-date information, or contact your local sales office.

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.national.com

CUSTOMER RESPONSE CENTER

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

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

Physical Dimensions inches (millimeters)

Order Number COP8ACC728N9,
COP8ACC728N8
See NS Molded Package Number N28B

Order Number COP8ACC728M9,
COP8ACC728M8
See NS Molded Package Number M28B

Order Number COP8ACC720M9,
COP8ACC720M8
see NS Molded Package Number M20B

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, T1 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
Francais 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