National Semiconductor

PRELIMINARY

COP8720C/COP8721C/COP8722C Single-Chip microCMOS Microcontrollers

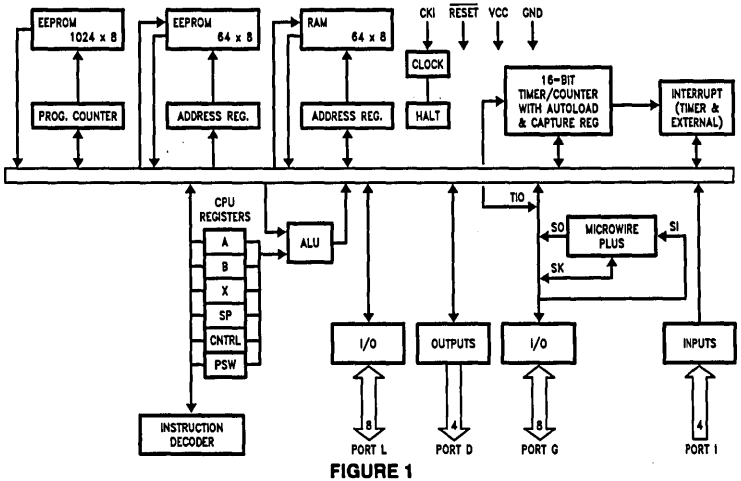
General Description

The COP8720C/COP8721C/COP8722C are members of the COPS™ microcontroller family featuring on-chip EEPROM modules. They are fully static parts, fabricated using double-metal silicon gate microCMOS technology. This low cost microcontroller is a complete microcomputer containing all system timing, interrupt logic, ROM, 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, a 16-bit timer/counter with capture register and a multi-sourced interrupt. Each I/O pin has software selectable options to adapt the COP8720C to the specific application. The part operates over a voltage range of 2.5V to 6.0V. High throughput is achieved with an efficient, regular instruction set operating at a 1 microsecond per instruction rate. The COP8720 is totally compatible with the ROM based COP820C microcontroller. It serves as a form, fit and function emulator device for the COP820 microcontroller family.

Features

- Low Cost 8-bit CORE microcontroller
- Fully static CMOS
- 1 μs instruction time (20 MHz clock)
- Low current drain (2.2 mA at 3 μs instruction rate)
  Low current static HALT mode (Typically < 10 μA)
- Single supply operation: 2.5V to 6.0V
- 1024 bytes EEPROM program memory
- 64 bytes of RAM
- 64 bytes EEPROM data memory
- 16-bit read/write timer operates in a variety of modes
  — Timer with 16-bit auto reload register
  — 16-bit external event counter
  — Timer with 16-bit capture register (selectable edge)
- Multi-source interrupt
  — Reset master clear
  — External interrupt with selectable edge
  — Timer interrupt or capture interrupt
  — Software interrupt
- 8-bit stack pointer (stack in RAM)
- Powerful instruction set, most instructions single byte
- BCD arithmetic instruction
- MICROWIRE/PLUS™ serial I/O
- 28 pin package (optionally 24 or 20 pin package)
- 24 input/output pins
- Software selectable I/O options (TRI-STATE®, push-pull, weak pull-up)
- Schmitt trigger inputs on Port G
- Form, fit and function EEPROM emulation device for COP820C/COP821C/COP822C
- Fully supported by National's MOLE™ development system

Block Diagram



TL/DD/9108-1

FIGURE 1

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
ESD Susceptibility (Note 4)	2000V
Total Current into V_{CC} Pin (Source)	50 mA
Total Current out of GND Pin (Sink)	60 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 $-40°C \le T_A \le +85°C$ unless otherwise specified

Parameter	Condition	Min	Typ	Max	Units
Operating Voltage		2.5		6.0	V
Power Supply Ripple (Note 1)	Peak to Peak			0.1 V_{CC}	V
Operating Voltage during EEPROM Write (Note 7)		4.5		6.0	V
Supply Current (see page 10)					
High Speed Mode, CKI = 20 MHz	V_{CC} = 6V, tc = 1 μs			13	mA
Normal Mode, CKI = 5 MHz	V_{CC} = 6V, tc = 2 μs			7	mA
Normal Mode, CKI = 2 MHz (Note 2)	V_{CC} = 2.5V, tc = 5 μs			2	mA
HALT Current (Note 3)	V_{CC} = 6V, CKI = 0 MHz		<10	30	μA
Input Levels					
RESET, CKI					
Logic High		0.9 V_{CC}			V
Logic Low				0.1 V_{CC}	V
All Other Inputs					
Logic High		0.7 V_{CC}			V
Logic Low				0.2 V_{CC}	V
Hi-Z Input Leakage	V_{CC} = 6.0V	−2		+2	μA
Input Pullup Current	V_{CC} = 6.0V	40		250	μA
G Port Input Hysteresis			0.05 V_{CC}		V
Output Current Levels					
D Outputs					
Source	V_{CC} = 4.5V, V_{OH} = 3.8V	0.4			mA
	V_{CC} = 2.5V, V_{OH} = 1.8V	0.2			mA
Sink	V_{CC} = 4.5V, V_{OL} = 1.0V	10			mA
	V_{CC} = 2.5V, V_{OL} = 0.4V	2.0			mA
All Others					
Source (Weak Pull-Up)	V_{CC} = 4.5V, V_{OH} = 3.2V	10		100	μA
	V_{CC} = 2.5V, V_{OH} = 1.8V	2.5		33	μA
Source (Push-Pull Mode)	V_{CC} = 4.5V, V_{OH} = 3.8V	0.4			mA
	V_{CC} = 2.5V, V_{OH} = 1.8V	0.2			mA
Sink (Push-Pull Mode)	V_{CC} = 4.5V, V_{OL} = 0.4V	1.6			mA
	V_{CC} = 2.5V, V_{OL} = 0.4V	0.7			mA
TRI-STATE Leakage		−2.0		+2.0	μA
Allowable Sink/Source Current Per Pin					
D Outputs (Sink)				15	mA
All Others				3	mA
Maximum Input Current (Room Temp) without Latchup (Note 5)				±100	mA
RAM Retention Voltage, Vr	500 ns Rise and Fall Time (Min)	2.0			V
Input Capacitance				7	pF
Load Capacitance on D2				1000	pF

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

Parameter	Condition	Min	Typ	Max	Units
Instruction Cycle Time (tc)					
High Speed Mode	$V_{CC} \geq 4.5V$	1		DC	μs
(Div-by 20)	$2.5V \leq V_{CC} < 4.5V$	2.5		DC	μs
Normal Mode	$V_{CC} \geq 4.5V$	2		DC	μs
(Div-by 10)	$2.5V \leq V_{CC} < 4.5V$	5		DC	μs
R/C Oscillator Mode	$V_{CC} \geq 4.5V$	3		DC	μs
(Div-by 10)					
	$2.5V \leq V_{CC} < 4.5V$	7.5		DC	μs
CKI Clock Duty Cycle (Note 6)	fr = Max (÷20 Mode)	33		66	%
Rise Time (Note 6)	fr = 20 MHz Ext Clock			12	ns
Fall Time (Note 6)	fr = 20 MHz Ext Clock			8	ns
Inputs					
t_{SETUP}	$V_{CC} \geq 4.5V$	200			ns
	$2.5V \leq V_{CC} < 4.5V$	500			ns
t_{HOLD}	$V_{CC} \geq 4.5V$	60			ns
	$2.5V \leq V_{CC} < 4.5V$	150			ns
Output Propagation Delay t_{PD1}, t_{PD0}	$R_L = 2.2k, C_L = 100$ pF				
SO, SK	$V_{CC} \geq 4.5V$			0.7	μs
	$2.5V \leq V_{CC} < 4.5V$			1.75	μs
All Others	$V_{CC} \geq 4.5V$			1	μs
	$2.5V \leq V_{CC} < 4.5V$			2.5	μs
MICROWIRE™ Setup Time t_{UWS}		20			ns
MICROWIRE Hold Time t_{UWH}		56			ns
MICROWIRE Output Propagation Delay t_{UPD}				220	ns
Input Pulse Width					
Interrupt Input High Time		t_C			
Interrupt Input Low Time		t_C			
Timer Input High Time		t_C			
Timer Input Low Time		t_C			
Reset Pulse Width		1.0			μs

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

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

Note 3: The HALT mode will stop CKI from oscillating in the RC and the Crystal configurations. Test conditions: All inputs tied to V_{CC}, L and G ports are at TRI-STATE and tied to ground, all outputs low and tied to ground.

Note 4: Human body model, 100 pF through 1500Ω.

Note 5: Except pins G6, G7, $\overline{\text{RESET}}$
pins G6, $\overline{\text{RESET}}$: +60 mA
pin G7: −25 mA

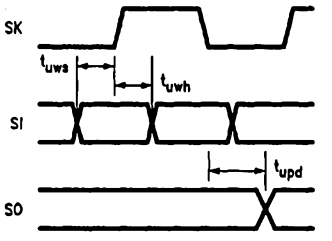
Note 6: Parameter sampled but not 100% tested.

Note 7: The temperature range for write operation is 0°C to 70°C.

EEPROM Characteristics

Parameter	Condition	Min	Typ	Max	Units
EEPROM Write Cycle Time	$4.5V \leq V_{CC} \leq 6.0V$	15	20	25	ms
EEPROM Number of Writes				10000	Cycles
V_{CC} Level for Write Lock Out	V_{LKO}	3.9		4.4	V
Programming Voltage to $\overline{\text{RESET}}$ Pin	V_{prg} $4.5V \leq V_{CC} \leq 6.0V$	11.5	12	12.5	V

Timing Diagrams

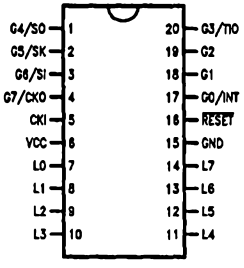


TL/DD/9108-22

FIGURE 2. MICROWIRE/PLUS Timing Diagram

Connection Diagrams

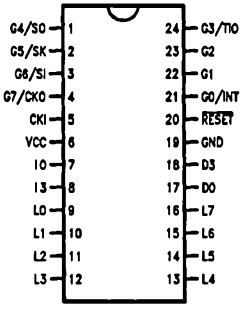
20-Pin Dual-In-Line Package



TL/DD/9108-3

Order Number COP8722CN
See NS Molded Package
Number N20A

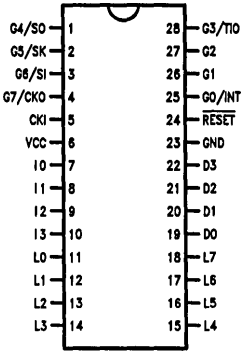
24-Pin Dual-In-Line Package



TL/DD/9108-4

Order Number COP8721CN
See NS Molded Package
Number N24A

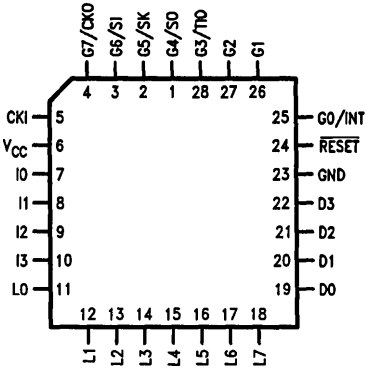
28-Pin Dual-In-Line Package



TL/DD/9108-5

Order Number COP8720CN
See NS Molded Package
Number N28B

28-Pin PLCC



TL/DD/9108-24

Order Number COP8720CV
See NS PLCC Package
Number V28A

FIGURE 3

Connection Diagrams (Continued)

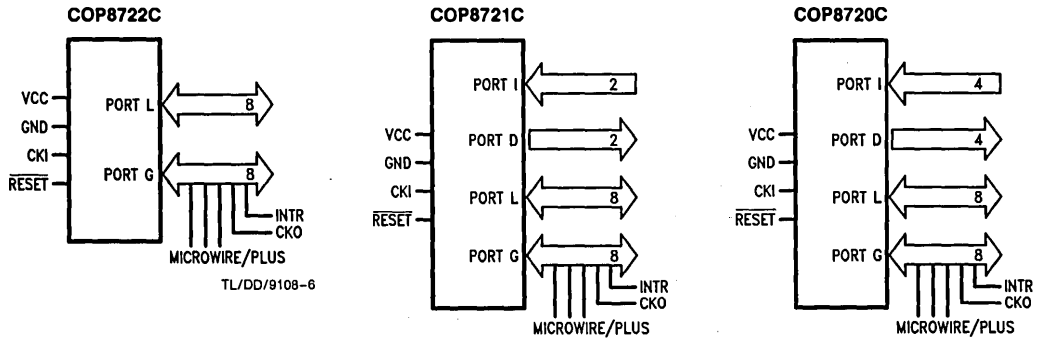


FIGURE 3 (Continued)

Pin Descriptions

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

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

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

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

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

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

Port L Config.	Port L Data	Port L Setup
0	0	Hi-Z Input (TRI-STATE)
0	1	Input With Weak Pull-Up
1	0	Push-Pull "0" Output
1	1	Push-Pull "1" Output

Three data memory address locations are allocated for these ports, one for data register, one for configuration register and one for the input pins.

PORT G is an 8-bit port with 6 I/O pins (G0–G5) and 2 input pins (G6, G7). All eight G-pins have Schmitt Triggers on the inputs. The G7 pin functions as an input pin under normal operation and as the continue pin to exit the HALT mode. There are two registers with each I/O port: a data register and a configuration register. Therefore, each I/O bit can be individually configured under software control as shown below.

Port G Config.	Port G Data	Port G Setup
0	0	Hi-Z Input (TRI-STATE)
0	1	Input With Weak Pull-Up
1	0	Push-Pull "0" Output
1	1	Push-Pull "1" Output

Three data memory address locations are allocated for these ports, one for data register, one for configuration register and one for the input pins. Since G6 and G7 are input only pins, any attempt by the user to set them up as outputs by writing a one to the configuration register will be disregarded. Reading the G6 and G7 configuration bits will return zeros. Note that the chip will be placed in the HALT mode by setting the G7 data bit.

Six bits of Port G have alternate features:

G0 INTR (an external interrupt)

G3 TIO (timer/counter input/output)

G4 SO (MICROWIRE serial data output)

G5 SK (MICROWIRE clock I/O)

G6 SI (MICROWIRE serial data input)

G7 CKO crystal oscillator output (selected by mask option) or HALT restart input (general purpose input)

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

PORT D is a four bit output port that is set high when $\overline{\text{RESET}}$ goes low.

The D2 pin is sampled at reset. If it is held low at reset the COP8720C enters the ROMless mode of operation.

Functional Description

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

ALU AND CPU REGISTERS

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

There are five CPU registers:

A is the 15-bit Program Counter register

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

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

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

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

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

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

Functional Description (Continued)

MEMORY

The COP8720C contains 1 Kbyte of Program EEPROM, 64 bytes of on-chip RAM and Registers, I/O, 64 bytes of Data EEPROM and 256 bytes of firmware ROM.

PROGRAM MEMORY

Program memory for the COP8720C consists of two modules—the 1 Kbyte program EEPROM and the 256 byte ROM which contains the firmware routines for reading and programming the EEPROM.

Memory locations in the 1 Kbyte program EEPROM module are accessed by the address register, EEAR, and the data register, EROMDR. The EEAR is mapped into the address locations E2 and E3. The EROMDR register is located at the address E1.

Under normal conditions, the program EEPROM and the ROM are addressed by the PC and their contents go to the instruction bus. During the EEPROM program and verify cycle, the EEPROM is treated as data memory while the COP8720C is executing out of the firmware ROM. The EEPROM is addressed through the EEAR register. The EROMDR register holds the data read back from the EEPROM location during a verify cycle and holds the data to be written into the EEPROM location during a program cycle. The verify cycle takes 1 instruction cycle and the write cycle takes 20 ms.

Accesses to the program EEPROM is controlled by two flags, AEN and PEN, in the control register, EECR.

AEN	PEN	Access Type
0	0	Normal
0	1	Normal
1	0	EEPROM Read Cycle
1	1	EEPROM Write Cycle

To prevent accidental erasures and over-write situations the application program should not set the AEN and PEN flags in the EECR register. The COP8720C supports application accesses to the EEPROM module via two subroutines in the firmware ROM—an EEPROM read and an EEPROM write subroutine. To program an EEPROM memory location, the user loads the EECR and EROMDR registers and invokes the write subroutine at the address 40C0 Hex. To read an EEPROM location the user loads the EEAR register with the address of the EEPROM memory location and invokes the read subroutine at the address 40D4 Hex. The read subroutine returns the contents of the addressed EEPROM location in the EROMDR register.

DATA MEMORY

The data memory for the COP8720C consists of on-chip RAM, EEPROM, I/O and registers. Data memory is accessed directly by the instruction or indirectly by the B, X and SP registers.

RAM

The COP8720C has 64 bytes of RAM. Sixteen bytes of RAM are mapped as "registers" that can be loaded efficiently, decremented and tested. Three specific registers: B, X and SP are mapped into this space, the other bytes are available for general use.

The instruction set of the COP8720C permits any bit in the data memory to be set, reset or tested. All I/O and the registers (except the A and PC) are memory mapped; therefore, I/O bits and register bits in addition to the normal data RAM can be directly and individually set, reset and tested.

DATA EEPROM

The COP8720C provides 64 bytes of EEPROM for nonvolatile data memory. The data EEPROM can be read and programmed in exactly the same way as the RAM. All instructions that perform read and write operations on the RAM work similarly upon the data EEPROM.

A data EEPROM programming cycle is initiated by an instruction such as X, LD, SBIT or RBIT. The EE memory support circuitry sets the BsyERAM flag in the EECR register immediately upon beginning a data EEPROM write cycle. It will be automatically reset by the hardware at the end of the data EEPROM write cycle. The application program should test the BsyERAM flag before attempting a write operation to the data EEPROM. A second EEPROM write operation while a write operation is in progress will be ignored. The Werr flag in the EECR register is set to indicate the error status.

SIGNATURE AND OPTION REGISTERS

The COP8720C provides a set of six additional registers implemented with EEPROM cells—the Signature and Option registers.

The Signature register is a four-byte register provided for storing ROM code rev. numbers or other application specific information. The Signature register is shadowed behind the data EEPROM cells at addresses 8C to 8F Hex. Two test modes are provided to allow the Signature register to be read or programmed.

The Option register consists of two bytes shadowed behind the addresses 89 and 8B Hex. The Option register allows the COP8720C to be programmed to accurately emulate the different mask options available on the COP820C.

—	—	—	—	ROMemu	x	0		89 Hex
—	—	—	—	HS	RC	XTAL	x	8B Hex

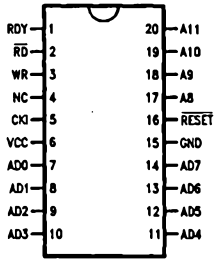
ROMemu: When set, the Data EEPROM and all the EE related registers become inaccessible. Thus, the EE registers look like nonexistent memory locations when addressed by the application program and the Program EEPROM behaves just like ordinary ROM. Thus, setting the ROMemu bit allows the COP8720C to emulate the ROM based COP820C with 100% accuracy.

HS, RC, XTAL: These three bits allow the COP8720C to emulate the clock options of the COP820C. Note that only five out of the possible eight combinations are legal—the combinations 0E, 0C and 06 are illegal combinations.

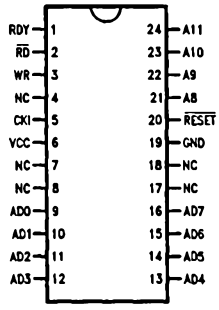
EECR and EE SUPPORT CIRCUITS

The EEPROM program and data modules share a common set of EE support circuits to generate all necessary high

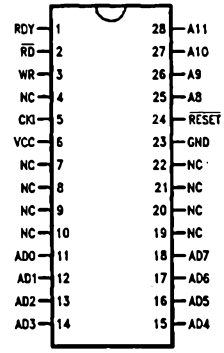
Functional Description (Continued)



TL/DD/9108-18



TL/DD/9108-19



TL/DD/9108-20

FIGURE 4. Pinouts for the COP8720C in Programming Mode

voltage programming pulses. Each programming cycle consists of a 10 ms erase cycle followed by a 10 ms write cycle for each byte. An EEPROM cell in the erase state is read out as a 0 and the written state is read out as a 1. Since the two EE modules share the support circuitry, programming the two modules at the same time is not allowed.

The EECR register provides control, status and test mode functions for the EE modules.

The EECR register bit assignments are shown below.

EECR Register Bit Assignment

Wr	Test Mode Codes					AEN	PEN	
Rd	Test	Mode	Codes	BsyEROM	BsyERAM	AEN	V_{LKO}	Werr
Bit	7	6	5	4	3	2	1	0

Werr Write Error. Writing to data EEPROM while a previous write cycle is still busy, that is BsyERAM is not 0, causes Werr to be set to 1 indicate error status. Werr is cleared by writing a 0 into it.

PEN A program EEPROM programming cycle is started by setting PEN and AEN to 1 at the same time. PEN is "written thru". It is not latched.

V_{LKO} EECR bit 1 is read as the lock out indicator. A low V_{CC} detector is enabled at the start of the EE programming cycle. If it finds V_{CC} less than V_{LKO}, the V_{LKO} status bit is set and the write cycle is aborted. The V_{LKO} status bit stays latched until the start of another EE programming cycle.

AEN AEN controls the program EEPROM address/data interface. when AEN is 0, the EEPROM is the program memory. It is adressed by PC, and its output data goes onto the instruction bus. When AEN is set to 1, the EEPROM becomes data memory. It is addressed by the EEAR, and it is accessed from the EROMDR.

BsyERAM Set to 1 when data EEPROM is being written, is automatically reset by the hardware upon completion of the write operation.

BsyEROM Set to 1 when program EEPROM is being written, is automatically reset by the hardware upon completion of the write operation.

Bits 3 to 7 of the EECR are used for encoding various EEPROM module test modes, most of which are for factory manufacturing tests. Two of the test modes used for accessing the signature and option registers are described in a previous section. The EE test modes are activated by applying high voltage to the $\overline{\text{RESET}}$ pin. Some of the test modes, if activated improperly, can make the part inoperable. These test modes are reserved for use by the manufacturer only.

The EECR register is cleared by $\overline{\text{RESET}}$. EECR is mapped into address location E0.

When either BsyERAM or BsyEROM is set to 1, that is an EEPROM programming cycle is in progress, the AEN bit is locked up and cannot be changed by the processor.

EXTERNALLY PROGRAMMING THE PROGRAM EEPROM

As shown in the previous section, the COP8720C permits the program EEPROM memory module to be altered under program control via the EECR register. To facilitate ease of development the COP8720C also provides an external mode of loading executable code into the program EEPROM module.

This section describes the programming method for the COP8720C EEPROM.

Programming the COP8720C EEPROM or the special registers is initiated by applying V_{PRG} to the $\overline{\text{RESET}}$ pin. Control gets transferred to the firmware ROM when V_{PRG} is applied to the $\overline{\text{RESET}}$ pin. The program contained in the firmware ROM sets up the I/O of the COP8720C to simulate the I/O requirements of a 2-kbyte memory device. This is done by setting up the COP8720C I/O as eight bits of address/data lines, three address lines, read/write control and a ready signal.

Functional Description (Continued)

Figure 4 shows the three packages and the associated I/O. The pin descriptions are as follows:

V_{CC}	Positive 5V Power Supply
GND	Ground
$\overline{RESET}$	Active Low Reset Input
CKI	Clock Input
AD0–AD7	Multiplexed Address/Data Lines
A8–A11	Address Lines
$\overline{RD}$	Active Low Read Strobe
WR	Active High Write Strobe
RDY	Active High Ready Output

The firmware ROM program allows the user to reference the special registers as EEPROM memory locations in the address range 2048–2070 decimal. The following mapping is used:

Signature Register #1 at EEPROM address 800 Hex
Signature Register #2 at EEPROM address 801 Hex
Signature Register #3 at EEPROM address 802 Hex
Signature Register #4 at EEPROM address 803 Hex

Option Register #1 at EEPROM address 804 Hex
Option Register #2 at EEPROM address 805 Hex

Note that in order to reference these registers the user must come in with addresses in the range 800 Hex to 805 Hex.

PROGRAMMING STEPS

The programmig host has to go through the following steps for the write and verify cycles. (See *Figure 2*)

WRITE:

1. Power is applied with the $\overline{RESET}$ and WR pins low and the $\overline{RD}$ high.
2. $\overline{RESET}$ is then brought up to V_{prg} within 1 μs.
3. The lower byte of the address to be written into is applied to the pins AD0–AD7 and the upper 3 bits of the address applied to the pins A8–A11.
4. Observing the setup times, WR is brought high.
5. The data to be programmed is applied to the pins AD0–AD7.
6. The RDY signal from the COP8720C goes low. This indicates that the WR and data on AD0–AD7 have been accepted and these inputs can be removed.
7. The programming host must now either wait for the RDY signal to go high or wait at least 20 ms before initiating a new programming cycle.

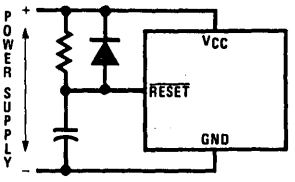
VERIFY:

1. Power is applied with $\overline{RESET}$ and WR pins held low and the $\overline{RD}$ high.
2. The $\overline{RESET}$ pin is brought up to V_{prg} within 1 μs.
3. The lower byte of the address to be read is applied to the pins AD0–AD7 and the upper three bits to the pins AD8–AD11.
4. Observing setup times the $\overline{RD}$ pin is brought low.
5. After a time T7, the RDY signal from the COP8720C goes low and data is ready for the host on the pins AD0–AD7. The data stays until the $\overline{RD}$ signal goes back high after which the RDY signal will go back high.
6. The host must wait for the RDY signal to go back high before the next read cycle is initiated.

RESET

The $\overline{RESET}$ input when pulled low initializes the microcontroller. Initialization will occur whenever the $\overline{RESET}$ input is pulled low. Upon initialization, the ports L and G are placed in the TRI-STATE mode and the Port D is set high. The PC, PSW and CNTRL registers are cleared. The data and configuration registers for Ports L & G are cleared.

The external RC network shown in *Figure 5* should be used to ensure that the $\overline{RESET}$ pin is held low until the power supply to the chip stabilizes.



TL/DD/9108–9

RC ≥ 5X Power Supply Rise Time
FIGURE 5. Recommended Reset Circuit

OSCILLATOR CIRCUITS

Figure 6 shows the three clock oscillator configurations available for the COP8720C.

A. CRYSTAL OSCILLATOR

The COP8720C can be driven by a crystal clock. The crystal network is connected between the pins CKI and CKO.

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

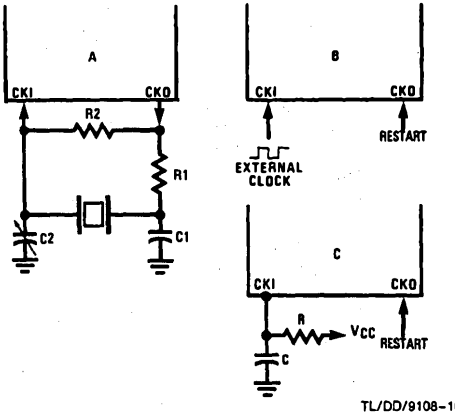
B. EXTERNAL OSCILLATOR

CKI can be driven by an external clock signal. CKO is available as a general purpose input and/or HALT restart control.

C. R/C OSCILLATOR

CKI is configured as a single pin RC controlled Schmitt trigger oscillator. CKO is available as a general purpose input and/or HALT restart control.

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



TL/DD/9108–10

FIGURE 6. Crystal and R-C Connection Diagrams

OSCILLATOR OPTIONS

The COP8720C can be driven by clock inputs between DC and 20 MHz. For low input clock frequencies (≤ 5 MHz) the instruction cycle frequency can be selected to be the input clock frequency divided by 10. This mode is known as the Normal Mode.

Functional Description (Continued)

TABLE I. Crystal Oscillator Configuration, T_A = 25°C

R1 (kΩ)	R2 (MΩ)	C1 (pF)	C2 (pF)	CKI Freq (MHz)	Conditions
0	1	30	30-36	20	V_{CC} = 5V
0	1	30	30-36	10	V_{CC} = 5V
0	1	30	30-36	4 (÷ 20)	V_{CC} = 2.5V
0	1	200	100-150	0.455	V_{CC} = 2.5V

TABLE II. RC Oscillator Configuration, T_A = 25°C

R (kΩ)	C (pF)	CKI Freq. (MHz)	Instr. Cycle (μs)	Conditions
3.3	82	2.8-2.2	3.6-4.5	V_{CC} = 5V
5.6	100	1.5-1.1	6.7-9	V_{CC} = 5V
6.8	100	1.1-0.8	9-12.5	V_{CC} = 2.5V

For oscillator frequencies that are greater than 5 MHz the chip must run with a divide by 20. This is known as the High Speed mode.

The COP820C microcontroller has five mask options for configuring the clock input. To emulate these mask options 3 bits must be set in the Option register.

HS	RC	XTAL	Mask Option
1	0	1	High Speed Crystal
0	0	1	Normal Mode Crystal
1	0	0	High Speed External
0	0	0	Normal Mode External
0	1	0	R/C Oscillator

The CKI and CKO pins are automatically configured upon selecting a particular option.

— High Speed Crystal (CKI/20) CKO for crystal configuration

— Normal Mode Crystal (CKI/10) CKO for crystal configuration

— High Speed External (CKI/20) CKO available as G7 input

— Normal Mode External (CKI/10) CKO available as G7 input

— R/C (CKI/10) CKO available as G7 input

Where, G7 can be used either as a general purpose input or as a control input to continue from the HALT mode.

CURRENT DRAIN

The total current drain of the chip depends on:

1) Oscillator operating mode—I1

2) Internal switching current—I2

3) Internal leakage current—I3

4) Output source current—I4

5) DC current caused by external input not at V_{CC} or GND—I5

Thus the total current drain, It is given as

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

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

The chip will draw the least current when in the normal mode. The high speed mode will draw additional current. The R/C mode will draw the most. Operating with a crystal network will draw more current than an external squarewave. Switching current, governed by the equation below, can be reduced by lowering voltage and frequency. Leakage current can be reduced by lowering voltage and temperature. The other two items can be reduced by carefully designing the end-user's system.

$I2 = C \times V \times f$

Where

C = equivalent capacitance of the chip. (TBD)

V = operating voltage

f = CKI frequency

The typical capacitance for the COP820C is TBD pF.

Some sample current drain values at V_{CC} = 6V are:

CKI (MHz)	Inst. Cycle (μs)	It (mA)
20	1	13
3.58	3	2.2
2	5	1.2
0.3	33	0.2
0 (HALT)	—	<0.01

HALT MODE

The COP8720C supports a power saving mode of operation: HALT. The controller is placed in the HALT mode by setting the G7 data bit, alternatively the user can stop the clock input. In the HALT mode all internal processor activities including the clock oscillator are stopped. The fully static architecture freezes the state of the controller and retains all information until continuing. In the HALT mode, power requirements are minimal as it draws only leakage currents and output current. The applied voltage (V_{CC}) may be decreased down to Vr (minimum RAM retention voltage) without altering the state of the machine.

There are two ways to exit the HALT mode: via the $\overline{\text{RESET}}$ or by the CKO pin. A low on the $\overline{\text{RESET}}$ line reinitializes the

Functional Description (Continued)

microcontroller and starts executing from the address 0000H. A low to high transition on the CKO pin causes the microcontroller to continue with no reinitialization from the address following the HALT instruction. This also resets the G7 data bit.

INTERRUPTS

The COP8720C has a sophisticated interrupt structure to allow easy interface to the real world. There are three possible interrupt sources, as shown below.

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

A maskable interrupt on timer carry or timer capture.

A non-maskable software/error interrupt on opcode zero.

INTERRUPT CONTROL

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

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

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

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

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

INTERRUPT PROCESSING

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

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

DETECTION OF ILLEGAL CONDITIONS

The COP8720C incorporates a hardware mechanism that allows it to detect illegal conditions which may occur from coding errors, noise and 'brown out' voltage drop situations. Specifically it detects cases of executing out of undefined ROM area and unbalanced stack situations.

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

Reading an undefined RAM location returns an FF (hexadecimal). The subroutine stack on the COP8720C grows down for each subroutine call. By initializing the stack pointer to the top of RAM, the first unbalanced return instruction will cause the stack pointer to address undefined RAM. As a result the program will attempt to execute from FFFF (hexadecimal), which is an undefined ROM location and will trigger a software interrupt.

MICROWIRE/PLUS™

MICROWIRE/PLUS is a serial synchronous bidirectional communications interface. The MICROWIRE/PLUS capability enables the COP8720C to interface with any of National Semiconductor's Microwire peripherals (i.e. A/D converters, display drivers, 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 8* shows the block diagram of the MICROWIRE/PLUS interface.

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.

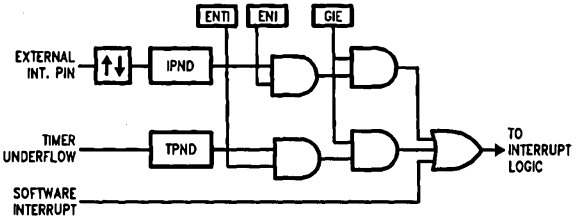
The CNTRL register is used to configure and control the MICROWIRE/PLUS mode. To use the MICROWIRE/PLUS, the MSEL bit in the CNTRL register is set to one. The SK clock rate is selected by the two bits, S0 and S1, in the CNTRL register. Table III details the different clock rates that may be selected.

TABLE III

S1	S0	SK Cycle Time
0	0	$2t_C$
0	1	$4t_C$
1	x	$8t_C$

where,

t_C is the instruction cycle clock.



TL/DD/9108-11

FIGURE 7. Interrupt Block Diagram

Functional Description (Continued)

MICROWIRE/PLUS OPERATION

Setting the BUSY bit in the PSW register causes the MICROWIRE/PLUS arrangement to start shifting the data. It gets reset when eight data bits have been shifted. The user may reset the BUSY bit by software to allow less than 8 bits to shift. The COP8720C may enter the MICROWIRE/PLUS mode either as a Master or as a Slave. *Figure 9* shows how two COP8720C microcontrollers and several peripherals may be interconnected using the MICROWIRE/PLUS arrangement.

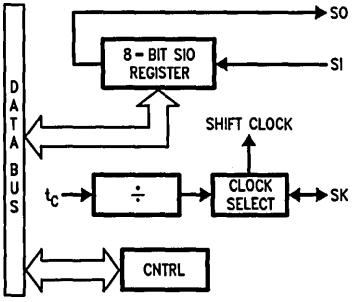
Master MICROWIRE/PLUS Operation

In the MICROWIRE/PLUS Master mode of operation the shift clock (SK) is generated internally by the COP8720C. The MICROWIRE/PLUS Master always initiates all data exchanges. (See *Figure 9.*) 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 IV summaries the bit settings required for Master mode of operation.

SLAVE MICROWIRE/PLUS OPERATION

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

The user must set the BUSY flag immediately upon entering the Slave mode. This will ensure that all data bits sent by the Master will be shifted properly. After eight clock pulses the BUSY flag will be cleared and the sequence may be repeated. (See *Figure 9.*)



TL/DD/9108-12

FIGURE 8. MICROWIRE/PLUS Block Diagram

TABLE IV

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

TIMER/COUNTER

The COP8720C has a powerful 16-bit timer with an associated 16-bit register enabling them to perform extensive timer functions. The timer T1 and its register R1 are each organized as two 8-bit read/write registers. Control bits in the register CNTRL allow the timer to be started and stopped under software control. The timer-register pair can be operated in one of three possible modes. Table V details various timer operating modes and their requisite control settings.

MODE 1. TIMER WITH AUTO-LOAD REGISTER

In this mode of operation the timer T1 counts down at the instruction cycle rate. Upon underflow the value in the register R1 gets automatically reloaded into the timer which continues to count down. The timer underflow can be programmed to interrupt the microcontroller. A bit in the control register CNTRL enables the TIO (G3) pin to toggle upon timer underflows. This allow the generation of square-wave outputs or pulse width modulated outputs under software control. (See *Figure 10.*)

MODE 2. EXTERNAL COUNTER

In this mode, the timer T1 becomes a 16-bit external event counter. The counter counts down upon an edge on the TIO pin. Control bits in the register CNTRL program the counter to decrement either on a positive edge or on a negative edge. Upon underflow the contents of the register R1 are automatically copied into the counter. The underflow can also be programmed to generate an interrupt. (See *Figure 10.*)

MODE 3. TIMER WITH CAPTURE REGISTER

Timer T1 can be used to precisely measure external frequencies or events in this mode of operation. The timer T1 counts down at the instruction cycle rate. Upon the occurrence of a specified edge on the TIO pin the contents of the timer T1 are copied into the register R1. Bits in the control register CNTRL allow the trigger edge to be specified either as a positive edge or as a negative edge. In this mode the user can elect to be interrupted on the specified trigger edge. (See *Figure 11.*)

Functional Description (Continued)

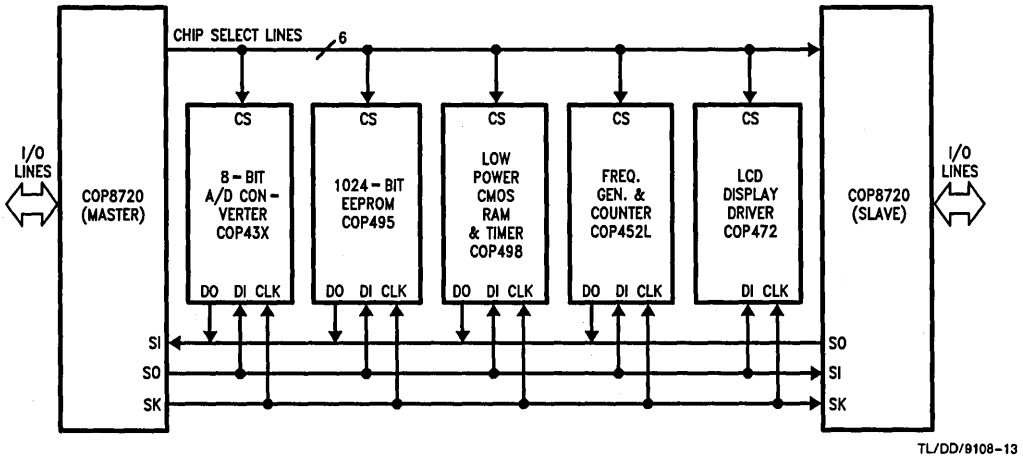


FIGURE 9. MICROWIRE/PLUS Application

TABLE V. Timer Operating Modes

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

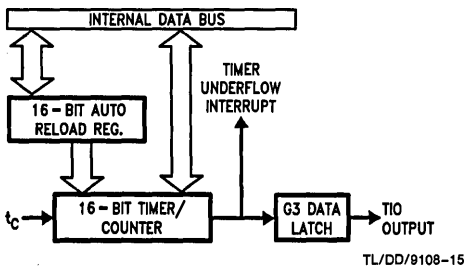


FIGURE 10. Timer/Counter Auto Reload Mode Block Diagram

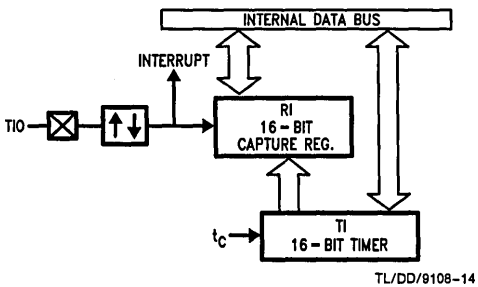
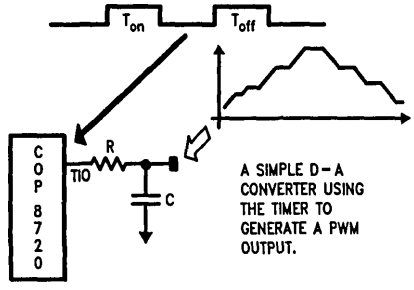


FIGURE 11. Timer Capture Mode Block Diagram

Functional Description (Continued)

TIMER PWM APPLICATION

Figure 12 shows how a minimal component D/A converter can be built out of the Timer-Register pair in the Auto-Reload mode. The timer is placed in the "Timer with auto reload" mode and the TIO pin is selected as the timer output. At the outset the TIO pin is set high, the timer T1 holds the on time and the register R1 holds the signal off time. Setting TRUN bit starts the timer which counts down at the instruction cycle rate. The underflow toggles the TIO output and copies the off time into the timer, which continues to run. By alternately loading in the on time and the off time at each successive interrupt a PWM frequency can be easily generated.



TL/DD/9108-16

FIGURE 12. Timer Application

Control Registers

CNTRL REGISTER (ADDRESS X'OOEE)

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

S1 & S0 Select the MICROWIRE/PLUS clock divide-by

IEDG External interrupt edge polarity select (0 = rising edge, 1 = falling edge)

MSEL Enable MICROWIRE/PLUS functions S0 and SK

TRUN Start/Stop the Timer/Counter (1 = run, 0 = stop)

TC3 Timer input edge polarity select (0 = rising edge, 1 = falling edge)

TC2 Selects the capture mode

TC1 Selects the timer mode

TC1	TC2	TC3	TRUN	MSEL	IEDG	S1	S0

BIT 7 BIT 0

PSW REGISTER (ADDRESS x'00EF)

The PSW register contains the following select bits:

GIE Global interrupt enable

ENI External interrupt enable

BUSY MICROWIRE/PLUS busy shifting

IPND External interrupt pending

ENTI Timer interrupt enable

TPND Timer interrupt pending

C Carry Flag

HC Half carry Flag

HC	C	TPND	ENTI	IPND	BUSY	ENI	GIE

Bit 7 Bit 0

Operating Modes

These controllers have two operating modes: Single Chip mode and the ROMless mode. The operating mode is determined by the state of the D2 pin at power on reset.

SINGLE CHIP MODE

In the Single Chip mode, the controller functions as a self contained microcontroller. It can address internal RAM and ROM. All ports configured as memory mapped I/O ports.

ROMLESS MODE

The COP8720C will enter the ROMless mode of operation if the D2 pin is held at logical "0" at reset. In this case the internal PROGRAM EEPROM is disabled and the controller can now address up to 32 kbytes of external program memory. It continues to use the on board RAM, and DATA EEPROM.

Memory Map

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

Address	Contents
00 to 2F	On Chip RAM Bytes
30 to 7F	Unused RAM Address Space (Reads as all Ones)
80 to BF	64 Bytes DATA EEPROM
C0 to CF	Expansion Space for I/O and Registers
D0 to DF	On Chip I/O and Registers
D0	Port L Data Register
D1	Port L Configuration Register
D2	Port L Input Pins (Read Only)
D3	Reserved for Port L
D4	Port G Data Register
D5	Port G Configuration Register
D6	Port G Input Pins (Read Only)
D7	Port I Input Pins (Read Only)
D8-DB	Reserved for Port C
DC	Port D Data Register
DD-DF	Reserved for Port D
E0 to EF	On Chip Functions and Registers
E0	EECR
E1	EROMDR
E2	EEAR Low Byte
E3	EEAR High Byte
E4-E8	Reserved
E9	MICROWIRE/PLUS Shift Register
EA	Timer Lower Byte
EB	Timer Upper Byte
EC	Timer Autoload Register Lower Byte
ED	Timer Autoload Register Upper Byte
EE	CNTRL Control Register
EF	PSW Register
F0 to FF	On Chip RAM Mapped as Registers
FC	X Register
FD	SP Register
FE	B Register

Memory Map (Continued)

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

Addressing Modes

REGISTER INDIRECT

This is the "normal" mode of addressing for the COP8720C. The operand is the memory addressed by the B register or X register.

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.

REGISTER INDIRECT (AUTO INCREMENT AND DECREMENT)

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

RELATIVE

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

Instruction Set

REGISTER AND SYMBOL DEFINITIONS

Registers

A	8-bit Accumulator register
B	8-bit Address register
X	8-bit Address register
SP	8-bit Stack pointer register
PC	15-bit Program counter register
PU	upper 7 bits of PC
PL	lower 8 bits of PC
C	1-bit of PSW register for carry
HC	Half Carry
GIE	1-bit of PSW register for global interrupt enable

Symbols

[B]	Memory indirectly addressed by B register
[X]	Memory indirectly addressed by X register
Mem	Direct address memory or [B]
Meml	Direct address memory or [B] or Immediate data
Imm	8-bit Immediate data
Reg	Register memory: addresses F0 to FF (Includes B, X and SP)
Bit	Bit number (0 to 7)
←	Loaded with
⟷	Exchanged with

Instruction Set (Continued)

Instruction Set

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

OPCODE LIST

Bits 7–4

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	Bits 3–0
JP -15	JP -31	LD 0F0,#i	DRSZ 0F0	RRCA	RC	ADC A, #i	ADC A, [B]	IFBIT 0,[B]	*	LD B, 0F	IFBNE 0	JSR 0000-00FF	JMP 0000-00FF	JP + 17	INTR	0
JP -14	JP -30	LD 0F1,#i	DRSZ 0F1	*	SC	SUBC A, #i	SUBC A,[B]	IFBIT 1,[B]	*	LD B, 0E	IFBNE 1	JSR 0100-01FF	JMP 0100-01FF	JP + 18	JP + 2	1
JP -13	JP -29	LD 0F2,#i	DRSZ 0F2	X A, [X+]	X A, [B+]	IFEQ A, #i	IFEQ A,[B]	IFBIT 2,[B]	*	LD B, 0D	IFBNE 2	JSR 0200-02FF	JMP 0200-02FF	JP + 19	JP + 3	2
JP -12	JP -28	LD 0F3,#i	DRSZ 0F3	X A, [X−]	X A, [B−]	IFGT A, #i	IFGT A,[B]	IFBIT 3,[B]	*	LD B, 0C	IFBNE 3	JSR 0300-03FF	JMP 0300-03FF	JP + 20	JP + 4	3
JP -11	JP -27	LD 0F4,#i	DRSZ 0F4	*	LAID	ADD A, #i	ADD A,[B]	IFBIT 4,[B]	CLRA	LD B, 0B	IFBNE 4	JSR 0400-04FF	JMP 0400-04FF	JP + 21	JP + 5	4
JP -10	JP -26	LD 0F5,#i	DRSZ 0F5	*	JID	AND A, #i	AND A,[B]	IFBIT 5,[B]	SWAPA	LD B, 0A	IFBNE 5	JSR 0500-05FF	JMP 0500-05FF	JP + 22	JP + 6	5
JP -9	JP -25	LD 0F6,#i	DRSZ 0F6	X A, [X]	X A, [B]	XOR A, #i	XOR A,[B]	IFBIT 6,[B]	DCORA	LD B, 9	IFBNE 6	JSR 0600-06FF	JMP 0600-06FF	JP + 23	JP + 7	6
JP -8	JP -24	LD 0F7,#i	DRSZ 0F7	*	*	OR A, #i	OR A,[B]	IFBIT 7,[B]	*	LD B, 8	IFBNE 7	JSR 0700-07FF	JMP 0700-07FF	JP + 24	JP + 8	7
JP -7	JP -23	LD 0F8,#i	DRSZ 0F8	NOP	*	LD A, #i	IFC	SBIT 0,[B]	RBIT 0,[B]	LD B, 7	IFBNE 8	JSR 0800-08FF	JMP 0800-08FF	JP + 25	JP + 9	8
JP -6	JP -22	LD 0F9,#i	DRSZ 0F9	*	*	*	IFNC	SBIT 1,[B]	RBIT 1,[B]	LD B, 6	IFBNE 9	JSR 0900-09FF	JMP 0900-09FF	JP + 26	JP + 10	9
JP -5	JP -21	LD 0FA,#i	DRSZ 0FA	LD A, [X+]	LD A, [B+]	LD [B+],#i	INCA	SBIT 2,[B]	RBIT 2,[B]	LD B, 5	IFBNE 0A	JSR 0A00-0AFF	JMP 0A00-0AFF	JP + 27	JP + 11	A
JP -4	JP -20	LD 0FB,#i	DRSZ 0FB	LD A, [X−]	LD A, [B−]	LD [B−],#i	DECA	SBIT 3,[B]	RBIT 3,[B]	LD B, 4	IFBNE 0B	JSR 0B00-0BFF	JMP 0B00-0BFF	JP + 28	JP + 12	B
JP -3	JP -19	LD 0FC,#i	DRSZ 0FC	LD Md, #i	JMPL	X A,Md	*	SBIT 4,[B]	RBIT 4,[B]	LD B, 3	IFBNE 0C	JSR 0C00-0CFF	JMP 0C00-0CFF	JP + 29	JP +13	C
JP -2	JP -18	LD 0FD,#i	DRSZ 0FD	DIR	JSRL	LD A, Md	RETSK	SBIT 5,[B]	RBIT 5,[B]	LD B, 2	IFBNE 0D	JSR 0D00-0DFF	JMP 0D00-0DFF	JP + 30	JP +14	D
JP -1	JP -17	LD 0FE,#i	DRSZ 0FE	LD A, [X]	LD A, [B]	LD [B], #i	RET	SBIT 6, [B]	RBIT 6, [B]	LD B, 1	IFBNE 0E	JSR 0E00−0EFF	JMP 0E00−0EFF	JP + 31	JP +15	E
JP -0	JP -16	LD 0FF,#1	DRSZ 0FF	*	*	*	RETI	SBIT 7,[B]	RBIT 7,[B]	LD B, 0	IFBNE 0F	JSR 0F00-0FFF	JMP 0F00-0FFF	JP + 32	JP + 16	F

where, i is the immediate data Md is a directly addressed memory location * is an unused opcode (see following table)

Instruction Execution Time

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

Most single instructions take one cycle time (1 μs at 20 MHz) to execute.

See the BYTES and CYCLES per INSTRUCTION table for details.

Bytes and Cycles per Instruction

The following table shows the number of bytes and cycles for each instruction in the format of byte/cycle (a cycle is 1 μs at 20 MHz).

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

Memory Transfer Instructions

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

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

Instructions Using A & C

CLRA	1/1
INCA	1/1
DECA	1/1
LAID	1/3
DCORA	1/1
RRCA	1/1
SWAPA	1/1
SC	1/1
RC	1/1
IFC	1/1
IFNC	1/1

Transfer of Control Instructions

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

Bytes and Cyles per Instruction (Continued)

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

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

Development Support

MOLE DEVELOPMENT SYSTEM

The MOLE (Microcomputer On Line Emulator) is a low cost development system and emulator for all microcontroller products. These include COPs, and the HPC family of products. The MOLE consists of a BRAIN Board, Personality Board and optional host software.

The purpose of the MOLE is to provide the user with a tool to write and assemble code, emulate code for the target microcontroller and assist in both software and hardware debugging of the system.

It is a self contained computer with its own firmware which provides for all system operation, emulation control, communication, PROM programming and diagnostic operations.

To program the COP8720C, a special adapter board is provided. This adapter board contains a socket for the COP8720C and plugs directly into the MOLE prom programmer.

It contains three serial ports to optionally connect to a terminal, a host system, a printer or a modem, or to connect to other MOLEs in a multi-MOLE environment.

MOLE can be used in either a stand alone mode or in conjunction with a selected host system using PC-DOS communicating via a RS-232 port.

How to Order

To order a complete development package, select the section for the microcontroller to be developed and order the parts listed.

Development Tools Selection Table

Microcontroller	Order Part Number	Description	Includes	Manual Number
COP820/ COP840	MOLE-BRAIN	Brain Board	Brain Board Users Manual	420408188-001
	MOLE-COP8-PB1	Personality Board	COP820/840 Personality Board Users Manual	420410806-001
	MOLE-COP8-IBM	Assembler Software for IBM	COP800 Software Users Manual and Software Disk	424410527-001
			PC-DOS Communications Software Users Manual	420040416-001
	420410703-001	Programmer's Manual		420410703-001

Development Support (Continued)

DIAL-A-HELPER

Dial-A-Helper is a service provided by the Microcontroller Applications group. The Dial-A-Helper is an Electronic Bulletin Board Information System and additionally, provides the capability of remotely accessing the MOLE development system at a customer site.

INFORMATION SYSTEM

The Dial-A-Helper system provides access to an automated information storage and retrieval system that may be accessed over standard dial-up telephone lines 24 hours a day. The system capabilities include a MESSAGE SECTION (electronic mail) 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. The minimum requirement for accessing the Dial-A-Helper is a Hayes Compatible modem.

If the user has a PC with a communications package then files from the FILE SECTION can be down loaded to disk for later use.

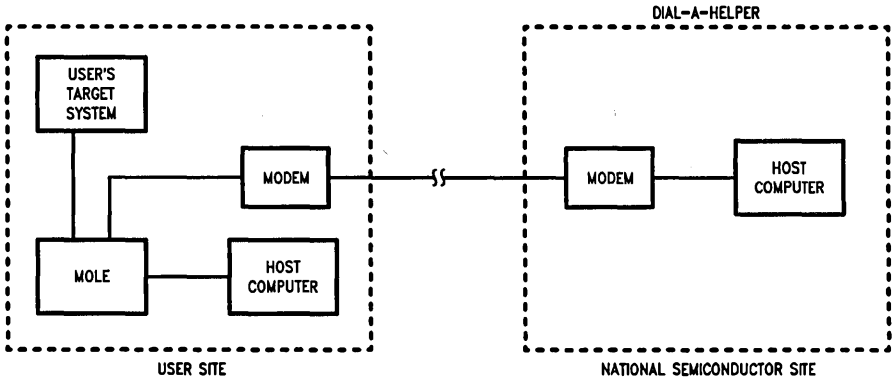
ORDER P/N: MOLE-DIAL-A-HLP

Information System Package Contains:
Dial-A-Helper User's Manual Pin
Public Domain Communications Software

FACTORY APPLICATIONS SUPPORT

Dial-A-Helper also provides immediate factory applications support. If a user is having difficulty in operating a MOLE, he can leave messages on our electronic bulletin board, which we will respond to, or under extraordinary circumstances he can arrange for us to actually take control of his system via modem for debugging purposes.

Voice: (408) 721-5582
Modem: (408) 739-1162
Baud: 300 or 1200 Baud
Setup: Length: 8-Bit
Parity: None
Stop Bit: 1
Operation: 24 Hours, 7 Days



TL/DD/9108-23