MICROWIRE/PLUS™ Serial Interface for COP800 Family

National Semiconductor
Application Note 579
Ramesh Sivakolundu
Sunder Velamuri
May 1989

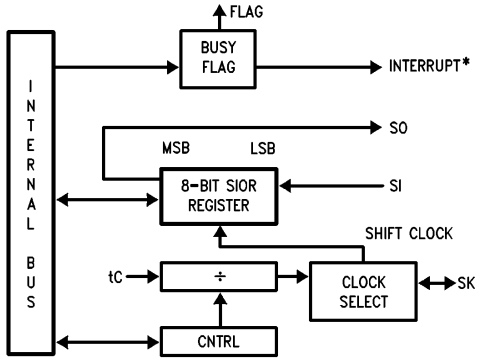
INTRODUCTION

National Semiconductor's COP800 family of full-feature, cost-effective microcontrollers use a new 8-bit single chip core architecture fabricated with M²CMOS process technology. These high performance microcontrollers provide efficient system solutions with a versatile instruction set and high functionality.

The COP800 family of microcontrollers feature the MICROWIRE/PLUS mode of serial communication. MICROWIRE/PLUS is an enhancement of the MICROWIRE™ synchronous serial communications scheme, originally implemented on the COP400 family of microcontrollers. The MICROWIRE/PLUS interface on the COP800 family of microcontrollers enables easy I/O expansion and interfacing to several COPS peripheral devices (A/D converters, EEPROMs, Display drivers etc.), and interfacing with other microcontrollers which support MICROWIRE/PLUS or SPI* modes of serial interface.

MICROWIRE/PLUS DEFINITION

MICROWIRE/PLUS is a versatile three wire, SI (serial input), SO (serial output), and SK (serial clock), bidirectional serial synchronous communication scheme where the COP800 is either the Master providing the Shift Clock (SK) or a slave accepting an external Shift Clock (SK). The COP800 MICROWIRE/PLUS system block diagram is shown in *Figure 1*. The MICROWIRE/PLUS serial interface utilizes an 8-bit memory mapped MICROWIRE/PLUS serial shift register, SIOR, clocked by the SK signal. As the name suggests, the SIOR register serves as the shift register for serial transfers. SI, the serial input line to the COP800 microcontroller, is the shift register input. SO, the shift register output, is the serial output to external devices. SK is the serial synchronous clock. Data is clocked into and out of the peripheral devices with the SK clock. The SO, SK and SI are mapped as alternate functions on pins 4, 5, and 6 respectively of the 8-bit bidirectional G Port.



TL/DD/10252–1

*only in COP888XX series

FIGURE 1. MICROWIRE/PLUS Block Diagram

MICROWIRE/PLUS OPERATION

In MICROWIRE/PLUS serial interface, the input data on the SI pin is shifted high order first into the Least Significant Bit (LSB) of the 8-bit SIOR shift register. The output data is shifted out high order first from the Most Significant Bit (MSB) of the shift register onto the SO pin. The SIOR register is clocked on the falling edge of the SK clock signal. The input data on the SI pin is shifted into the LSB of the SIOR register on the rising edge of the SK clock. The MSB of the SIOR register is shifted out to the SO pin on the falling edge of the SK clock signal. The SK clock signal is generated internally by the COP800 for the master mode of MICROWIRE/PLUS operation. In the slave mode, the SK clock is generated by an external device (which acts as the master) and is input to the COP800.

The MSEL (MICROWIRE Select) flag in the CNTRL register is used to enable MICROWIRE/PLUS operation. Setting the MSEL flag enables the gating of the MICROWIRE/PLUS interface signals through the G port. Pins G4, G5, and G6 of the G port are used for the signals SO, SK and SI, respectively. It should be noted that the G port configuration register must be set up appropriately for MICROWIRE/PLUS operation. Table I illustrates the G-port configurations. In the master mode of MICROWIRE/PLUS operation, G4 and G5 need to be selected as outputs for SO and SK signals. Alternatively, in the slave mode of operation, G5 needs to be configured as an input for the external SK. The SI signal is a dedicated input on G6 and therefore no further setup is required.

TABLE I. G Port Configurations

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

The SL1 and SL0 (S1 and S0 in COP820C and COP840C) bits of the CNTRL register are used to select the clock division factor (2, 4, or 8) for SK clock generation in MICROWIRE/PLUS master mode operation. A clock select table for these bits of the CNTRL register along with the CNTRL register is shown in Table II. The counter associated with

RRD-B30M105/Printed in U. S. A.

the master mode clock division factor is cleared when the MICROWIRE/PLUS BUSY flag is low. The clock division factor is relative to the instruction cycle frequency. For example, if the COP800 is operating with an internal clock of 1 MHz, the SK clock rate would be 500 kHz, 250 kHz, or 125 kHz for SL1 and SL0 values of 00, 01 and 10 (or 11) respectively.

TABLE II

CNTRL Register (Address X'00EE)

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

SL1 & SL0 Select the MICROWIRE 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 Signals SK and SO Respectively

T1C0 Timer T1 Start/Stop Control in Timer Modes 1 and 2

Timer T1 Underflow Interrupt Pending Flag in Timer Mode 3

T1C1 Timer T1 Mode Control Bit

T1C2 Timer T1 Mode Control Bit

T1C3 Timer T1 Mode Control Bit

T1C3	T1C2	T1C1	T1C0	MSEL	IEDG	SL1	SL0

Bit 7 Bit 0

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

Where t_c is the instruction cycle clock

MICROWIRE/PLUS MASTER MODE OPERATION

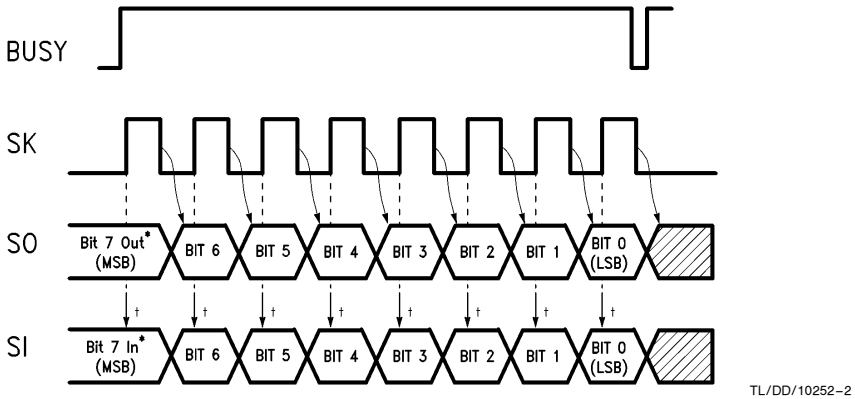
In the MICROWIRE/PLUS master mode, the BUSY flag of PSW (Processor Status Word) is used to control the shifting of the MICROWIRE/PLUS 8-bit shift register. Setting the BUSY flag causes the SIOR register to shift out 8 bits of data from SO at the high order end of the shift register. During the same time, 8 new bits of data from SI are shifted into the low order end of the SIOR register. The BUSY flag is automatically reset after the 8 bits of data have been shifted *(Figure 2)*. The COP888XX series of microcontrollers provide a vectored maskable interrupt when the BUSY goes low indicating the end of an 8-bit shift. Input data is clocked into the SIOR register from the SI pin with the rising edge of the SK clock, while the MSB of the SIOR is shifted onto the SO pin with the falling edge of the SK clock. The user may reset the BUSY bit by software to allow less than 8 bits to shift. However, the user should ensure that the software BUSY resets only occurs when the SK clock is low, in order to avoid a narrow SK terminal clock.

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 configured as an input and the SO pin configured as an output by resetting and setting the appropriate bits in the Port G configuration register. The user must set the BUSY flag immediately upon entering the Slave mode. After eight clock pulses the Busy flag will be cleared and the sequence may be repeated. However, in the Slave mode the COP888 series does not shift data if the BUSY flag is reset, whereas the COP820C and COP840C continues to shift regardless of the BUSY flag, if the SK clock is active.

MICROWIRE/PLUS ALTERNATE SK MODE

The COP888XX series of microcontrollers also allow an additional Alternate SK Phase Operation. In the normal mode data is shifted in on the rising edge of the SK clock and data is shifted out on the falling edge of the SK clock *(Figure 2)*. The SIOR 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 data is shifted out on the rising edge of the SK clock *(Figure 3)*.

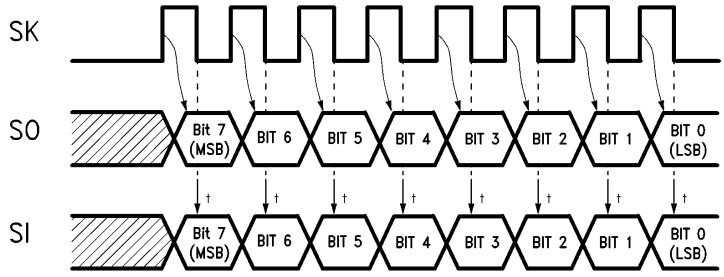


TL/DD/10252–2

*This bit becomes valid immediately after loading the SIOR register of the transmitting device.

†Arrows indicate points at which SI is sampled.

FIGURE 2. MICROWIRE/PLUS Timing



↑ Arrows indicates points at which SI is sampled.

FIGURE 3. Alternate Phase SK Clock Timing

A control flag, SKSEL, allows either the normal SK clock or alternate SK clock to be selected. Resetting SKSEL selects the normal SK clock and setting SKSEL selects the alternate SK clock for the MICROWIRE/PLUS logic. The SKSEL flag is mapped into the G6 configuration bit. The SKSEL flag is reset after power up, selecting the normal SK clock signal. The alternate mode facilitates the usage of the MICROWIRE/PLUS protocol for serial data transfer between peripheral devices which are not compatible with the normal SK clock operation, i.e., shifting data out on the falling edge of the SK clock and shifting in data on the rising edge of the SK clock.

MICROWIRE/PLUS SAMPLE PROTOCOL

This section gives a sample MICROWIRE/PLUS protocol using a COP888CL and COP840C. The slave mode operating procedure for this sample protocol is explained, and a timing illustration of the protocol is provided.

1. The MSEL bit in the CNTRL register is set to enable MICROWIRE; G0 ($\overline{CS}$) and G5 (SK) are configured as inputs and G4 (SO) as an output. G6 (SI) is always an input.
2. Chip Select line ($\overline{CS}$) from master device is connected to G0 of the slave device. An active-low level on $\overline{CS}$ line causes the slave to interrupt.
3. From the high-to-low transistion on the $\overline{CS}$ line, there is no data transfer on the MICROWIRE until time ''T'' (See *Figure 4*).
4. The master initiates data transfer on the MICROWIRE by turning on the SK clock.
5. A series of data transfers take place between the master and slave devices.
6. The master pulls the $\overline{CS}$ line high to end the MICROWIRE operation. The slave device returns to normal mode of operation.

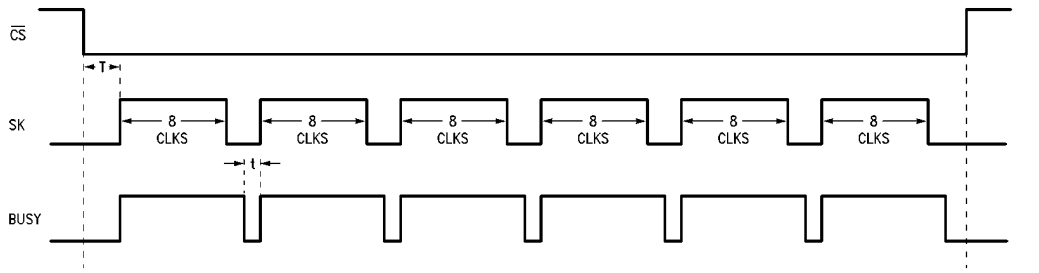
SLAVE MODE OPERATING PROCEDURE

1. The MSEL bit in the CNTRL register is set to enable MICROWIRE; G0 ($\overline{CS}$) and G5 (SK) are configured as inputs and G4 (SO) as an output. G6 (SI) is always an input.
2. Normal mode of operation until interrupted by $\overline{CS}$ going low.
3. Set the BUSY flag and load SIOR register with the data to be sent out on SO. (The shift register shifts 8 bits of data from SO at the high order end of the shift register. During the same time, 8 new bits of data from SI are loaded into the low order end of the shift register.)
4. Wait for the BUSY flag to reset. (The BUSY flag is automatically reset after 8 bits of data have been shifted).
5. If data is being read in, the user should save contents of the SIOR register.
6. The prearranged set of data transfers are performed.
7. Repeat steps 3 through 6. The user must ensure steps 3 through 6 are performed in time ''t'' (See *Figure 4*) as agreed upon in the protocol.

DIFFERENCES BETWEEN COP888 AND COP820/COP840

The COP888 series MICROWIRE/PLUS feature differs from that of the COP820/COP840 in some respects. The COP888 series can be configured to interrupt the processor after the completion of a MICROWIRE/PLUS operation indicated by the BUSY flag going low. The COP888 series supports a vectored interrupt scheme. Two bytes of program memory space are reserved for each interrupt source. The user would do any required context switching and 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 instruction. The addresses of the different interrupt service routines are chosen by the user and stored in ROM in a table starting at 0yE0 where ''y'' depends on the 256 byte block (0y00 to 0yFF) in which the VIS instruction is located. The vector address for the MICROWIRE/PLUS interrupt is 0yF2-0yF3.

Secondly, the COP888 series supports the alternate SK phase mode of MICROWIRE/PLUS operation. This feature facilitates the usage of the MICROWIRE/PLUS protocol for serial data transfer between peripheral devices which are not compatible with the normal SK clock operation, i.e., shifting data out on the falling edge of SK clock and shifting in data on the rising edge of the SK clock.



TL/DD/10252–4

FIGURE 4. MICROWIRE/PLUS Sample Protocol Timing Diagram

INTERFACE CONSIDERATIONS

To preserve the integrity of data exchange using MICROWIRE/PLUS, two aspects have to be considered:

1. Serial data exchange timing.

2. Fan-out/fan-in requirements.

Theoretically, infinite devices can access the same interface and be uniquely enabled sequentially in time. In practice, however, the actual number of devices that can access the same serial interface depends on the following: System data transfer rate, system supply requirement, capacitive loading on SK and SO outputs, the fan-in requirements of the logic families or discrete devices to be interfaced.

HARDWARE INTERFACE

For proper data transfer to occur the output should be able to switch between a HIGH level and a LOW level in a predetermined amount of time. The transfer is strictly synchronous and the timing is related to the MICROWIRE/PLUS system clock (SK). For example, if a COPS controller outputs a value at the falling edge of the clock and is latched in by the peripheral device at the rising edge, then the following relationship has to be satisifed:

$$t_{DELAY} + t_{SETUP} \leq t_{CK}$$

where t_{CK} is the time from data output starts to switch to data being latched into the peripheral chip, t_{SETUP} is the setup time for the peripheral device where the data has to be at a valid level, and t_{DELAY} is the time for the output to read the valid level. t_{CK} is related to the system clock provided by the SK pin of the COPS controller and can be increased by increasing the COPS instruction cycle time.

Besides the timing requirements, system supply and fan-out/fan-in requirements also have to be considered when interfacing with MICROWIRE/PLUS. To drive multi-devices on the same MICROWIRE/PLUS, the output drivers of the controller need to source and sink the total maximum leakage current of all the inputs connected to it and keep the signal level within the valid logic ''1'' and ''0'' input voltage levels. Thus, if devices of different types are connected to the same serial interface, output driver of the controller must satisfy all the input requirements of each device. Similarly, devices with TRI-STATE® outputs, when connected to the SI input, must satisfy the minimum valid input level of the controller and the maximum TRI-STATE® leakage current of all outputs.

So, for devices that have incompatible input levels or source/sink requirements, external pull-up resistors or buffers are necessary to provide level-shifting or driving.

TABLE III

Features		Part Number							
		DS890XX	**MM545X**	**COP470**	**COP472**	**ADC83X (COP430)**	**COP498/499**	**COP452L**	**NMC9306 (COP494)**
GENERAL									
Chip Function		AM/PM PLL	LED Display Driver	VF Display Driver	LCD Display Driver	A/D	RAM & Timer	Frequency Generator	E²PROM
Process		ECL	NMOS	PMOS	CMOS	CMOS	CMOS	NMOS	NMOS
V_{CC} Range		4.75V–5.25V	4.5V–11V	−9.5V to −4.5V	3.0V–5.5V	4.5V–0.3V	2.4V–5.5V	4.5V–6.3V	4.5V–5.5V
Pinout		20	40	20	20	8/14/20	14/8	14	14
HARDWARE INTERFACE									
Min V_{IH}/Max V_{IL}		2.1V/0.7V	2.2V/0.8V	−1.5V/−4.0V	0.7 V_{CC}/0.8V	2.0V/0.8V	0.8 V_{CC}/0.4 V_{CC}	2.0V/0.8V	2.0V/0.8V
SK Clock Range		0–625 kHz	0–500 kHz	0–250 kHz	4–250 kHz	10–200 kHz	4–250 kHz	25–250 kHz	0–250 kHz
Write Data DI	Setup Min	0.3 μs	0.3 μs	1.0 μs	1.0 μs	0.2 μs	0.4 μs	800 ns	0.4 μs
	Hold Min	0.8 μs	(Note 3)	50 ns	100 ns (Note 1)	0.2 μs	0.4 μs	1.0 μs	0.4 μs
Read Data Prop Delay		(Note 4)	(Note 3)	(Note 3)	(Note 3)	(Note 3)	2 μs (Note 2)	1 μs (Note 2)	2.0 μs
Chip Enable	Setup	0.275 μs	0.4 μs	1.0 μs Min	1 μs (Note 1)	0.2 μs	0.2 μs (Note 1)	(Note 3)	0.2 μs
	HOLD	0.300 μs	(Note 3)	1.0 μs Min	1 μs (Note 2)	0.2 μs	0 (Note 2)	(Note 3)	0
Max Frequency Range	AM	8 MHz	(Note 3)	(Note 3)	(Note 3)	(Note 3)	(Note 3)	(Note 3)	(Note 3)
	FM	120 MHz	(Note 3)	(Note 3)	(Note 3)	(Note 3)	(Note 3)	(Note 3)	(Note 3)
Max Osc. Freq.		(Note 3)	(Note 3)	250 kHz	(Note 3)	(Note 3)	2.1 MHz (−21) 32 kHz (−15)	256–2100 kHz (−4) 64–525 kHz (−2)	(Note 3)
SOFT									
Serial I/O Protocol		11D1–D20	1D1–D35	8 Bits At a Time	b1–b40	1xxx	1yyxxD6–D0 Start Bit	1yxxxx	1AA–DD
Instruction/ Address Word		None	None	None	None	(Note 4)	(Note 4)	(Note 4)	(Note 4)

Note 1: Reference to SK rising edge.

Note 2: Reference to SK falling edge.

Note 3: Not defined.

Note 4: See data sheet for different modes of operation.

TYPICAL APPLICATIONS

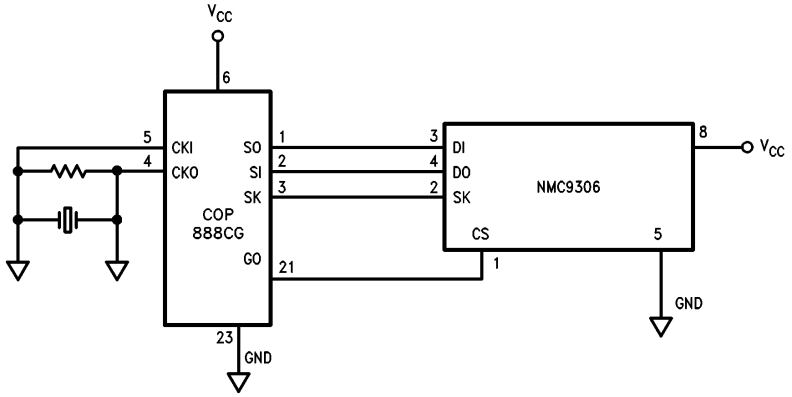
A whole family of off-the shelf devices exist that are directly compatible with MICROWIRE/PLUS protocol. This allows direct interface with the COP800 family of microcontrollers. Table III provides a summary of the existing devices, their function and specification.

NMC9306–COP888CG INTERFACE

The pin connection involved in interfacing an NMC9306 (COP494), a 256 bit E^{2}PROM, with the COP888CG microcontroller is shown in *Figure 5*. Some notes on the NMC9306 interface requirements are:

1. The SK clock frequency should be in the 0 kHz–250 kHz range.
2. $\overline{CS}$ low period following an Erase/Write instruction must not exceed 30 ms maximum. It should be set at typical or minimum specification of 10 ms.
3. The start bit on DI must be set by a ''0'' to ''1'' transition following a $\overline{CS}$ enable (''0'' to ''1'') when executing any instruction. One $\overline{CS}$ enable transition can only execute one instruction.
4. In the read mode, following an instruction and data train, the DI can be a ''don't care'', while the data is being outputted, i.e., for the next 17 bits or clocks. The same is true for other instructions after the instrution and data has been fed in.
5. The data out train starts with a dummy bit 0 and is terminated by chip deselect. Any extra SK cycle after 16 bits is not essential.

   If $\overline{CS}$ is held on after all 16 of the data bits have been outputed, the DO will output the state of DI until another $\overline{CS}$ LO to HI transition starts a new instruction cycle.
6. After a read cycle, the $\overline{CS}$ must be brought low for one SK clock cycle before another instruction cycle starts.



TL/DD/10252–5

FIGURE 5. NMC9306–COP888CG Interface

Instruction Set

Commands	Start Bit	Opcode	Address	Comments
READ	1	0000	A3A2A1A0	Read Register 0–15
WRITE	1	1000	A3A2A1A0	Write Register 0–15
ERASE	1	0100	A3A2A1A0	Erase Register 0–15
EWEN	1	1100	00 01	Write/Erase Enable
ENDS	1	1100	00 10	Write/Erase Disable
***WRAL	1	1100	01 00	Write All Registers
ERAL	1	1100	01 01	Read All Registers

Where A3A2A1A0 corresponds to one of the sixteen 16-bit registers.

All commands, data in, and data out are shifted in/out on the rising edge of the SK clock.

Write/Erase is then done by pulsing $\overline{CS}$ low for 10 ms.

All instructions are initiated by a LO–HI transition on $\overline{CS}$ followed by a LO–HI transition on DI.

READ— After read command is shifted in DI becomes don't care and data can be read out on data out, starting with dummy bit zero.

WRITE— Write command shifted in followed by data in (16 bits) the$\overline{CS}$ pulsed low for 10 ms minimum.

ERASE/ERASE ALL— Command shifted in followed by $\overline{CS}$ low.

WRITE ALL— Pulsing $\overline{CS}$ low for 10 ms.

ENABLE/DISABLE— Command shifted in.

A detailed explanation of the E^{2}PROM timing diagrams, instruction set and the various considerations could be found in the NMC9306 data sheet. A source listing of the software to interface the NMC9306 with the COP888CG is provided.

SOURCE LISTING

```
.INCLD COP888.INC
;
;This program provides in the form of subroutines, the ability to erase,enable, disable, read and write to the COP494 EEPROM.
;
;
SNDBUF  = 0                     ;CONTAINS THE COMMAND BYTE TO BE WRITTEN TO COP494
RDATL   = 1                     ;LOWER BYTE OF THE COP494 REGISTER DATA READ
RDATH   = 2                     ;UPPER BYTE OF THE COP494 REGISTER DATA READ
WDATL   = 3                     ;LOWER BYTE OF THE DATA TO BE WRITTEN TO COP494
                                ;REGISTER
WDATH   = 4                     ;UPPER BYTE OF THE DATA TO BE WRITTEN TO COP494
                                ;REGISTER
ADRESS  = 5                     ;THE LOWER 4-BITS OF THIS LOCATION CONTAIN THE
                                ;ADDRESS
                                ;OF THE COP494 REGISTER TO BE READ/WRITTEN
FLAGS   = 6                     ;USED FOR SETTING UP FLAGS
                                ;
                                ; FLAG VALUE     ACTION
                                ; ----------     ------
                                ;   00          ERASE,ENABLE,DISABLE,ERASE ALL
                                ;   01          READ CONTENTS OF COP494 REGISTER
                                ;   03          WRITE TO COP494 REGISTER
                                ;   OTHERS      ILLEGAL COMBINATION
DLYH    = 0F0
DLYL    = 0F1
;
;THE INTERFACE BETWEEN THE COP888CG AND THE COP494 (256-BIT EEPROM) CONSISTS OF FOUR LINES. THE
;G0 (CHIP SELECT LINE), G4 (SERIAL OUT SO), G5 (SERIAL CLOCK SK) ;AND G6 (SERIAL IN SI).
;
;    INITIALIZATION
;
                LD      PORTGC,#031     ;Setup G0,G4,G5 as outputs
                LD      PORTGD,#00      ;Initialize G data reg to zero
                LD      CNTROL,#08      ;Enable MSEL, select MW rate of 2tc
                LD      B,#PSW
                LD      X,#SIOR
;
;THIS ROUTINE ERASES THE MEMORY LOCATION POINTED TO BY THE ADDRESS CONTAINED IN THE LOCATION
;"ADRESS". THE LOWER NIBBLE OF "ADRESS" CONTAINS THE COP494 REGISTER ADDRESS AND THE UPPER NIBBLE
;SHOULD BE SET TO ZERO.
;
ERASE:          LD      A,ADRESS
                OR      A,#0C0
                X       A,SNDBUF
                LD      FLAGS,#0
                JSR     INIT
                RET
;
;THIS ROUTINE ENABLES PROGRAMMING OF THE COP494. PROGRAMMING MUST BE PRECEDED ONCE BY A
;PROGRAMMING ENABLE (EWEN).
;
EWEN:           LD      SNDBUF,#030
```

TL/DD/10252–6

```
                LD          FLAGS,#0
                JSR         INIT
                RET
;
;THIS ROUTINE DISABLES PROGRAMMING OF THE COP494.
;
EWDS:           LD          SNDBUF,#0
                LD          FLAGS,#0
                JSR         INIT
                RET
;
;THIS ROUTINE ERASES ALL REGISTERS OF THE COP494.
;
ERAL:           LD          SNDBUF,#020
                LD          FLAGS,#0
                JSR         INIT
                RET
;
;THIS ROUTINE READS THE CONTENTS OF THE COP494 REGISTER. THE COP494 ADDRESS IS SPECIFIED IN THE
;LOWER NIBBLE OF LOCATION  "ADRESS".  THE UPPER NIBBLE SHOULD BE SET TO ZERO. THE 16-BIT CONTENTS OF
;THE COP494 REGISTER ARE STORED IN RDATL AND RDATH.
;
READ:           LD          A,ADRESS
                OR          A,#080
                X           A,SNDBUF
                LD          FLAGS,#1
                JSR         INIT
                RET
;
;THIS ROUTINE WRITES A 16-BIT VALUE STORED IN WDATL AND WDATH TO THE COP494 REGISTER WHOSE ADDRESS
;IS CONTAINED IN THE LOWER NIBBLE OF THE LOCATION "ADRESS". THE UPPER NIBBLE OF ADDRESS LOCATION
;SHOULD BE SET TO ZERO.
;
WRITE:          LD          A,ADRESS
                OR          A,#040
                X           A,SNDBUF
                LD          FLAGS,#3
                JSR         INIT
                RET
;
;THIS ROUTINE SENDS OUT THE START BIT AND THE COMMAND BYTE. IT ALSO DECIPHERS THE CONTENTS OF THE
;FLAG LOCATION AND TAKES A DECISION REGARDING WRITE, READ OR RETURN TO THE CALLING ROUTINE.
;
INIT:           SBIT        0,PORTGD        ;SET CHIP SELECT HIGH
                LD          SIOR,#001       ;LOAD SIOR WITH START BIT
                SBIT        BUSY,[B]        ;SEND OUT THE START BIT
PUNT1:          IFBIT       BUSY,[B]
                JP          PUNT1
                LD          A,SNDBUF
                X           A,[X]           ;LOAD SIOR WITH COMMAND BYTE
                SBIT        BUSY,[B]        ;SEND OUT COMMAND BYTE
PUNT2:          IFBIT       BUSY,[B]
                JP          PUNT2
                IFBIT       0,FLAGS         ;ANY FURTHER PROCESSING ?
```

TL/DD/10252–7

```
                JP      NOTDON          ;YES
                RBIT    0,PORTGD        ;NO, RESET CS AND RETURN
                RET
;
NOTDON:         IFBIT   1,FLAGS         ;READ OR WRITE?
                JP      WR494           ;JUMP TO WRITE ROUTINE
                LD      SIOR,#000       ;NO, READ COP494
                SBIT    BUSY,PSW        ;DUMMY CLOCK TO READ ZERO
                RBIT    BUSY,[B]
                SBIT    BUSY,[B]
PUNT3:          IFBIT   BUSY,[B]
                JP      PUNT3
                X       A,[X]
                SBIT    BUSY,[B]
                X       A,RDATH
PUNT4:          IFBIT   BUSY,[B]
                JP      PUNT4
                LD      A,[X]
                X       A,RDATL
                RBIT    0,PORTGD
                RET
;
WR494:          LD      A,WDATH
                X       A,[X]
                SBIT    BUSY,[B]
PUNT5:          IFBIT   BUSY,[B]
                JP      PUNT5
                LD      A,WDATL
                X       A,[X]
                SBIT    BUSY,[B]
PUNT6:          IFBIT   BUSY,[B]
                JP      PUNT6
                RBIT    0,PORTGD
                JSR     TOUT
                RET
;
;ROUTINE TO GENERATE DELAY FOR WRITE
;
TOUT:           LD      DLYH,#00A
WAIT:           LD      DLYL,#0FF
WAIT1:          DRSZ    DLYL
                JP      WAIT1
                DRSZ    DLYH
                JP      WAIT
                RET
                .END
```

TL/DD/10252–8

COP472-COP820 Interface

The pin connection required for interfacing COP472-3 Liquid Crystal Display (LCD) Controller with COP820C microcontroller is shown in *Figure 6*. The COP472-3 drives a multiplexed liquid crystal display directly. Data is loaded serially and is held in internal latches. One COP472-3 can drive 36 segments and two or more COP472-3's can be cascaded to drive additional segments as long as the output loading capacitance does not exceed specifications.

The COP472-3 requires 40 information bits: 36 data and 4 control. The function of each control bit is described briefly. Data is loaded in serially, in sets of eight bits. Each set of segment data is in the following format:

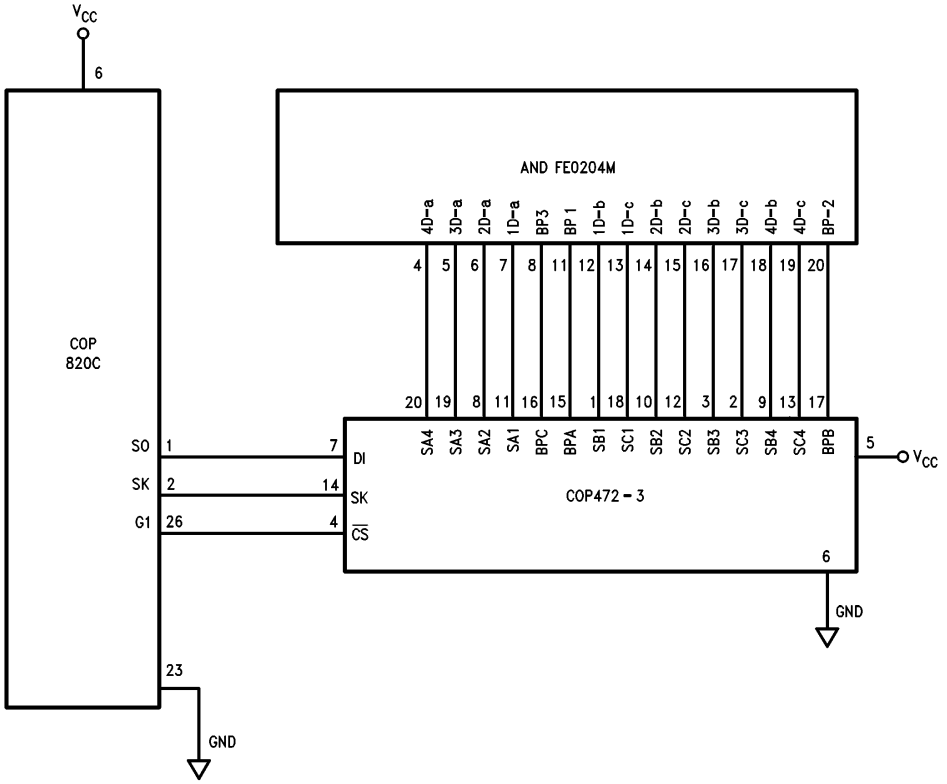
SA	SB	SC	SD	SE	SF	SG	SH

Data is shifted into an eight bit shift register. The first bit of data is for segment H, digit 1, and the eight bit is for segment A, digit 1. A set of eight bits are shifted in and then loaded into the digit one latches. The second, third, and fourth set is then loaded sequentially. The fifth set of data bits contain special segment data and control data in the following format:

SYNC	Q7	Q6	X	SP4	SP3	SP2	SP1

The first four bits shifted in contain the special character segment data. The fifth bit is not used. The sixth and seventh bits program the COP472-3 as a stand alone LCD driver or as a master or slave for cascading COP472-3's. The Table IV summarizes the function of bits six and seven.

The eight bit is used to synchronize two COP472-3's to drive an 8½ digit display. A detailed explanation of the various timing diagrams, loading sequence and segment/backplane multiplex scheme can be found in the data sheets of COP472-3. The source listing of the software used in the interface is provided.



TL/DD/10252–12

FIGURE 6. COP472–COP820C Interface

SOURCE LISTING

```
;THIS PROGRAM DISPLAYS FOUR DIGITS OF THE RAM SPECIFIED BY; THE ADDRESS POINTER "HEAD" ON A 4 DIGIT 3
;DECIMAL POINT (MULTIPLEXED) LCD DISPLAY. THE DATA STREAM IS SENT OUT SERIALLY THROUGH THE
;MICROWIRE/PLUS INTERFACE TO THE COP472 LCD DISPLAY DRIVER. NOTE: THE RAM CONTENTS SHOULD BE
;BETWEEN "0" AND "F".
;
                .TITLE      LCD
;                                                                          TL/DD/10252–9
,
                .CHIP       820
;
;
                PORTGD      = 0D4           ;PORT G DATA REGISTER
                PORTGC      = 0D5           ;PORT G CONFIGURATION
;
;
                SIO         = 0E9           ;MICROWIRE SHIFT REGISTER
;
;
                PSW         = 0EF           ;PSW REGISTER
                CNTRL       = 0EE           ;CNTRL REGISTER
;
;
                CONTRL      = 04            ;MEMORY LOCATION FOR THE
;                                           ;COP472 CONTROL WORD
                HEAD        = 00            ;STARTING MEMORY LOC FOR
                                            ;DATA TO BE DISPLAYED
                MEMSTR      = 05            ;STARTING MEMORY LOC FOR
                                            ;STORING SEGMENT DATA
                MEMEND      = 08            ;MEMORY LOC FOR LAST
                                            ;SEGMENT DATA
;
;
START:          LD          CNTRL,#08       ;SET MSEL BIT IN CNTRL
                LD          PORTGC,#032     ;SET G5,G4& G1 AS OUTPUTS
                LD          CONTRL,#0FC     ;SET COP472 IN STAND ALONE MODE
;                                           ;
;                                           ;
;
;THIS ROUTINE  GETS THE SEGMENT DATA FOR RAM DIGITS POINTED BY B REGISTER AND STORES IN RAM MEMORY
;POINTED BY X REGISTER
;
;
AGAIN:          LD          B,#HEAD         ;POINTER TO START ADDRESS
                LD          X,#MEMSTR       ;POINTER TO STORE ADDRESS
NEXDIG:         LD          A,[B+]          ;LOAD A WITH RAM DIGIT AND
                                            ;INCREMENT B POINTER
                ADD         A,#0F0          ;ADD OFFSET TO THE DIGIT
                LAID                        ;LOOKUP SEGMENT DATA TO A
                X           A,[X+]          ;STORE IN MEMORY
                IFBNE       #04             ;CHECK FOR END OF FOUR
                                            ;DIGITS AND REPEAT
                JP          NEXDIG          ;IF NECESSARY
;
;
; THIS ROUTINE DISPLAYS THE CONTENTS OF FOUR MEMORY LOCATION
; ON THE LCD DISPLAY.
;
;
DSP:            LD          B,#MEMEND       ;LOAD THE START ADDRESS
                RBIT        1,PORTGD        ;BIT G1 IS USED TO SELECT
                                            ;COP472 (PIN 4)
```

TL/DD/10252–10

```
REPEAT:     LD      A,[B-]          ;SEGMENT DATA TO A
            X       A,SIO           ;LOAD THE SIO REGISTER
            SBIT    #2,PSW          ;SET BUSY BIT IN PSW
WAIT:       IFBIT   #2,PSW          ;WAIT TILL SHIFTING IS
            JP      WAIT            ;COMPLETE
            IFBNE   #04             ;CHECK FOR END OF FOUR
            JP      REPEAT          ;DIGITS AND REPEAT
            SBIT    1,PORTGD        ;DESELECT COP472
LOOP:       JP      LOOP            ;DONE DISPLAYING
;
;
; STORE THE LOOKUP TABLE FOR SEGMENT DATA IN ROM LOCATION 0F0
;
;
            .=0F0
;
            .BYTE   03F,006,05B,04F ;DATA FOR 0,1,2,3
            .BYTE   066,06D,07D,07  ;DATA FOR 4,5,6,7
            .BYTE   07F,067,077,07C ;DATA FOR 8,9,A,B
            .BYTE   039,05E,079,071 ;DATA FOR C,D,E,F
;
;
            .END
```

TL/DD/10252–11

The code listed in this App Note is available on Dial-A-Helper.

Dial-A-Helper is a service provided by the Microcontroller Applications Group. 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 communicating to and from the Microcontroller Applications Group and a FILE SECTION mode that can be used to search out and retrieve application data about NSC Microcontrollers. The minimum system requirement is a dumb terminal, 300 or 1200 baud modem, and a telephone.

With a communications package and a PC, the code detailed in this App Note can be downloaded from the FILE SECTION to disk for later use. The Dial-A-Helper telephone lines are:

Modem (408) 739-1162
Voice (408) 721-5582

For Additional Information, Please Contact Factory

Lit. # 100579

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

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

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

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408