# Testing of COP400 Family Devices

National Semiconductor
COP Note 7
April 1991



## Table of Contents



This note will provide some insight into the test mode, the mechanics of testing, and the philosophy of how to implement a test of the COP-400 microcontrollers. Other than the obvious, (verifying that the part meets the specifications), the reason for the test must be considered. Somewhat different criteria may hold, depending on the objective. The manufacturer wafer sort or final test can differ from an incoming inspection at the user's plant, or a field reject test. The first two tests have limited interest as this is not a justification of the testing done on the part during manufacture. Rather, this is a guide for those doing user functional testing.

### 1.0 INTRODUCTION

Since the introduction of the very first semiconductor devices, testing has been a major problem and expense in their production and use. As the complexity has risen, testing has become a more significant factor. With today's single chip microcontrollers like the COPS™ devices this is particularly true as one has a complete computer system in a chip. In order to reduce the testing burden, the facilities to ease the testing have been built into the COPS devices. With the test ability built into the device for production test, the user need only follow set procedures to verify the chip at incoming inspection or field test.

### 2.0 PHILOSOPHY

The basic test philosophy requires that four major areas be exercised. These areas are:

1) Synchronize the device and tester.

2) Test the internal logic and I/O.

3) Test the RAM.

4) Verify the ROM program.

If the devices perform all of these four properly, the device is good. This is a reasonable assumption with a standard device that has a debugged test routine and is ROM programmed. A custom circuit just going into production might not have the accumulated test background. By attacking the problem on a ''sum of the parts'' approach, one need not do any exhaustive functional test on routine production parts. This will be a major gain where lengthy time consuming or time dependent routines are involved. If one attempts to do a functional test of the chip, a sequence that is unique to the application is needed. Thus, a test program must be written and debugged for each ROM pattern. Further, a test box/board must be designed, built, debugged, documented, and maintained for each one. If testing has been considered from the beginning, the chip will have built-in capabilities to exercise the various parts of it. The different functional parts and instructions are tested to verify proper operation at the voltage and frequency limits.

### 3.0 BUILT-IN TEST FEATURES

The first step in testing the COP400 devices is to understand the built-in test control features. This will involve the SI/O and the L lines. The SO pin has been designed to be the control node for testing. The pin will normally be in an active low state and when forced high externally, places the chip in the test mode. It should be noted that this output can sink considerable current and one should not force the pin to the $V_{CC}$ rail. By limiting the voltage to the 2.0/3.0V range one can not damage the device where the application of a higher voltage could. When forced into the test mode the SI pin controls the sub mode of the chip. With SI high the data placed on the L port is used as an instruction. When SI is low (and the L output is enabled) the contents of the ROM will be dumped out through the L port. Certain other internal functions have been implemented to allow these modes but these are not part of the basic operation. Included in this category is the activation of the skip signal to prevent the program counter from jumping out of sequence by executing a program control instruction.

COPS™ and MICROWIRE™ are trademarks of National Semiconductor Corporation.





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

### 3.1 Sync Between Tester and DUT

In order to be able to test a COPS chip, the tester must be in sync with the device under test (DUT). By using an external oscillator the two may be run at the same frequency. This is true regardless of the option or type of oscillator chosen for the chip. Even the RC configuration may be overridden with an external signal that meets the level requirements. In addition to running at the same frequency, the chip and tester must be in sync on a bit basis. See *Figure 1.* The supportive features mentioned above include the condition of the SK signal being a bit (instruction) clock until stopped by software in the program. Hence, one can start the tests based on an edge change of SK. It is important that this be accurate because all data I/O changes will be relative to the SK timing (see the appropriate device data sheet).

It should also be noted that the oscillator frequency is programmed to a rate of 4-32 higher than SK. If one is building a test fixture for more than one device, some method must be available to enter this number. If one is testing a COP420 or COP421 near its upper limit it would be wise to do the SK sync operation at a lower rate and then increase the input frequency. This is desirable because the phase relationship is close to TTL propagation delays at the upper limit. Implementation of the area could be a preset counter that is gated on after a zero to one transition is seen on SK. Continual comparison could be made but once in sync, there should not be any need for the comparison as they should remain in sync.

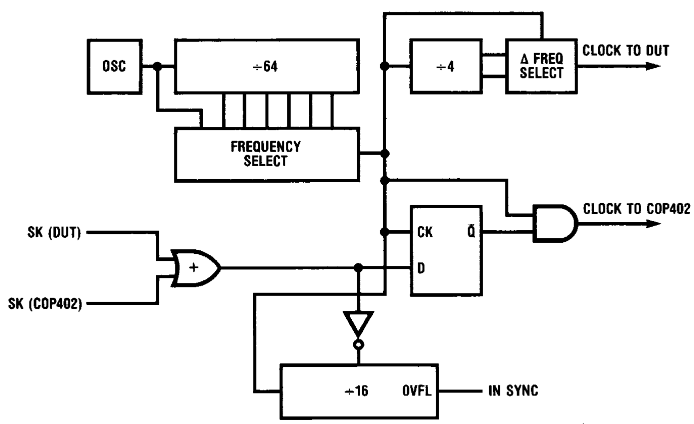
The basic use of this ''sync counter'' is to derive the proper timing for loading data and instructions into the chip and verify the outputs. The COP402 data sheet should be used as a guide for these times, modified properly for the L and C parts. For those designing testers, it is suggested that one not attempt to test worse case timing changes as these could be very difficult to implement. Like other parametric tests these should in general be left to the professional test equipment.

### 3.2 Internal Logic Test

With the device and the tester in sync, actual testing may begin. See the sequence control circuit of *Figure 2.* To place the chip into the test mode the SO output is pulled to a one level (between 2.0 and 3.0 volts). It should be pulled with a circuit that will limit the upper voltage to 3V as this output can have a significant current sink capability. On power up (or after reset) the SO line is set to a zero by the internal logic. An internal sense line will detect the forced condition and provide test control. A delay of 10 ms should be taken after power-up to allow the power on reset circuit to time out before instructions can be executed. If the reset pin is activated in mid-program for some reason, several instructions cycle times should be ignored to insure complete operation.

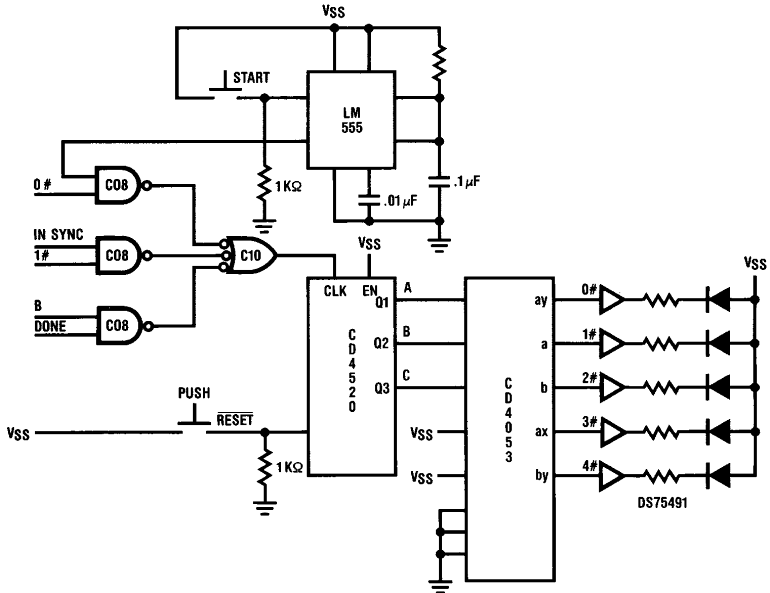
The tester should at this point force instructions into the L port. These instructions will be executed as if they were from the ROM. The sequence of the instructions is not particularly critical. Table I gives an example sequence. The main steps are to be able to detect an output change (OGI) early to verify connection/operation. It is much better to find a problem before going through the steps of loading RAM and then finding that the chip doesn't work. All instructions should be exercised although certain ones should be postponed. Enabling the Q register to the L port is an example. This would interfere with the insertion of instructions on the L port. Another problem is the SO test which could be set up with an XAS and then released from the test mode to check proper data output.

Certain commands will require more effort than others. To check the program counter during JMP's and sub-routine operation will require that known info at the new address be available. One should execute a JSRP at some known address and release the test mode to see that the operation in the subroutine (e.g., SC) is done and that a return is made to N + 1. At this point test mode can be re-established to continue the test. The main point to remember is to provide a positive indication of the success of that specific test.




TL/DD/6940–1

**FIGURE 1. Tester Clock Generation and Synchronization Circuit**




TL/DD/6940–2

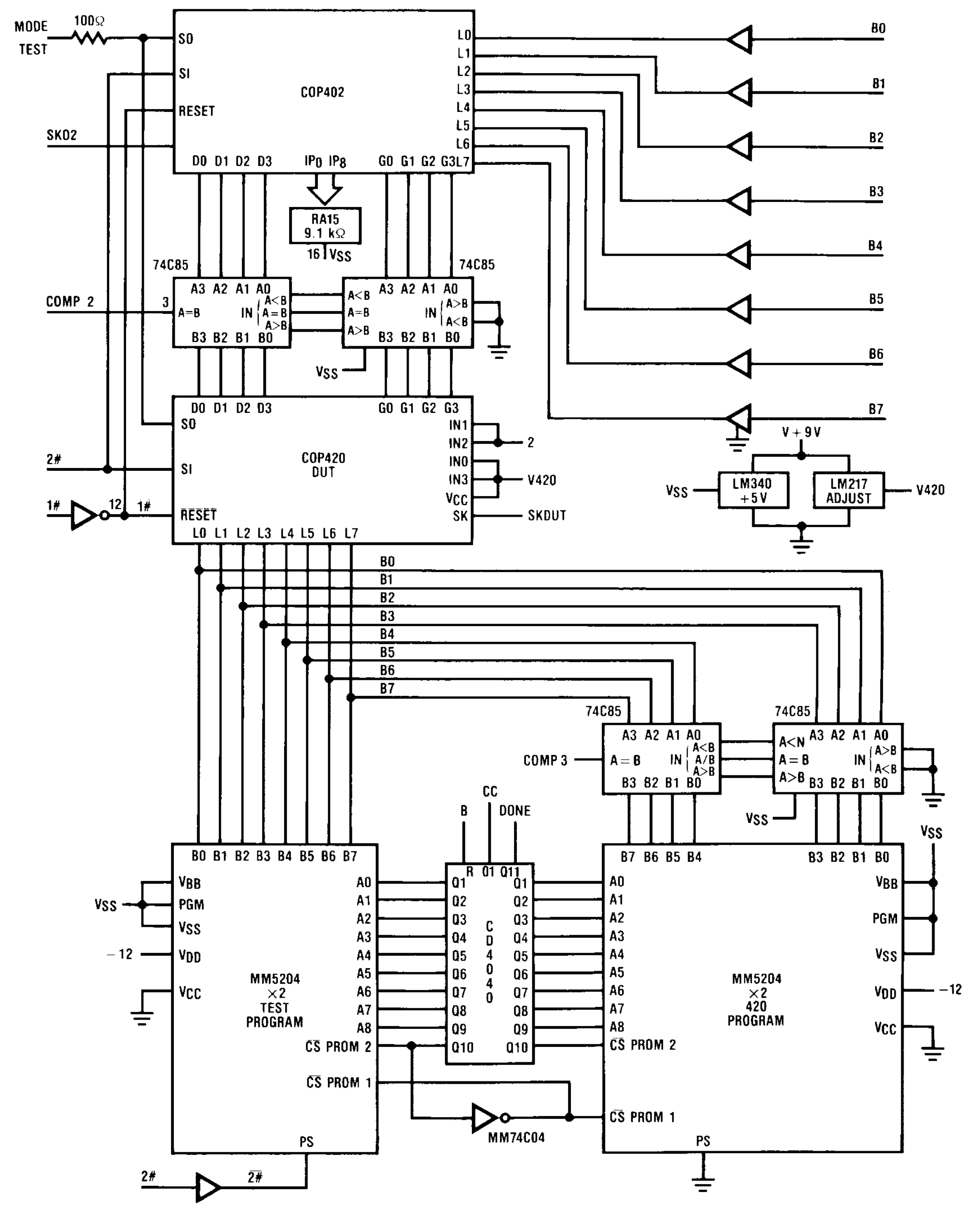
**FIGURE 2. Tester Mode Sequencer**

### 3.3 RAM Test

The verification of RAM is a part of the internal logic test, but is treated separately here. One must check both the RAM and its address register to find all faults. An example of this testing would be to load RAM with a string of STII commands. By then going back and reading this data to the outside (through an OMG instruction in a loop) the tester could verify both RAM and address were functional. One could then load RAM with all 6's and 9's (or 5's and 10's) sequentially to insure that all bits were functional and adjacent bits not shorted. Other similar tests could be run at the discretion of the user to do further testing. All of these tests would utilize the output of data via the G ports to validate the data. See the comparator circuit *Figure 3.*

### 3.4 ROM Dump

Successful operation of the internal logic tests and RAM will lead to the final test phase, ROM comparison. In order to check the ROM contents, the ROM dump mode must be entered. One should force a JMP to an address near the end of the ROM space (3FF for a 420 chip, 1FF for a 410). A desirable point might be 3FA. The program counter will step ahead on each instruction cycle unless a program control is executed. The next step is to load the Q register with a non-conflicting value so that the enabling of the L outputs will not destroy the second byte of the LEI instruction as control is passed into the ROM dump mode. After going to this address, one should execute an enable of the L lines to the output port (LEI 4). Having done this the external buffers should be disabled and the SI pin taken low. This will allow data out and remove potential level conflicts. By letting the PC step ahead to address zero one can then begin the byte by byte comparison of data. In this mode the controller is not executing the code because the skip line is enabled throughout the sequence. By halting a counter on a failure, one could determine the questionable address.




TL/DD/6940–3

**FIGURE 3. Functional Logic and RAM Comparison Circuit**

## TABLE I. Typical Test Sequence

| INSTRUCTION | RESULT | COMMENTS |
|---|---|---|
| NOP | NO CHANGE | CHECK NOP & ALLOW TRANSIENT CYCLE FOR MODE |
| OGI 9 | G(0 > 9) | NOT ON 410L/411L |
| OGI 6 | G(9 > 6) | REVERSE ALL G STATES |
| STII 8 | | SET UP 0,0 FOR FUTURE |
| LBI 3,13 | | B TO NEW POSITION (3, 13) |
| OBD | D(0 > 13 | CHECK D |
| CLRA | | MAKE SURE A = 0 |
| XABR | | 3 > A; 0 > Br |
| CAB | | MOVE 3 to Bd |
| OBD | D(13 > 3) | CHECK XABR CAB & D CHANGE |
| CLRA | | ! |
| AISC 2 | | !FORCE A > 2 |
| CAB | | 2 > Bd |
| OBD | D(3 > 2) | VERIFY 2 FROM A > Bd |
| STII 7 | | 7 > 0.2 & Bd > 3 |
| OBD | D(2 > 3) | STII INCREMENTS Bd |
| CAB | | SEE THAT A STILL THE SAME |
| OMG | G(6 > 7) | OMB & RAM CHECK |
| CLRA | | |
| CAB | | B(0,0) |
| OMG | G(7 > 8) | TIE IN RAM, A & G OPERATION |
| SMB 0 | | SMB INST. CHECK |
| OMG | G(8 > 9) | : |
| SMB 1 | | : |
| OMG | G(9 > 11) | : |
| RMB 0 | | : |
| RMB 3 | | : |
| X | | :0 > 0,0;2 > A |
| CAB | | A = 2 > B |
| OMG | G(11 > 7) | OUTPUT M(0,2) |
| LD 1 | | M(0,2) > A; B > 1,2 |
| XAD 0,0 | | A(7) < –> M(0,0) 2 |
| AISC 15 | | AISC CHECK; A = 1 |
| LDD 0,0 | | CHECK SKIP OF 2 BYTE INST. |
| X | | STORE 1 |
| OMB | G(7 > 1) | VERIFY |
| LD 0 | | COPY1,2 BACK TO A |
| ADT | | ADD TEN |
| XDS | | LEAVE 11 IN 1,2;GO 1, 1 WITH 1 |
| XDS | | LEAVE 1 IN 1,1;GO 1,0 W ? |
| OBD | D(2 > 0) | CHECK Bd MOVEMENT |
| STII 5 | | 5 > 1,0;Bd TO 1,1 |
| CBA | | CHECK B > A |
| AISC 3 | | AISC CHECK 4 > A |

| INSTRUCTION | RESULT | COMMENTS |
|---|---|---|
| XDS | | 1 > A; 4 > 1,1 |
| OMG | G(1 > 5) | FROM 1,0 |
| XDS | | 5 > A; 1 > 1,0; Bd < 15 SKIP |
| LDD 0,0 | | SKIPPED ! |
| OBD | D(0 > 15) | |
| AISC 4 | | 9 > A |
| X | | 9 > 15 |
| OMG | G(5> 9) | |
| CLRA | | |
| COMP | | ONES TO A |
| XOR | | FLIP MEMORY |
| XIS | | 6 > 1,15; 9 > A; Bd > 1,0 |
| LDD 0,0 | | SKIP |
| SKE | | |
| LB 1,2 | | SKIP 2 WORD LBI (NOT IN 410) |
| OBD | D(15 > 0) | VERIFY WORD |
| SKE | | 11 NOT = 9 |
| LBI 1,0 | | BACK TO 1,0 |
| SMB 2 | | : |
| SKE | | : |
| RMB 2 | | : |
| SKE | | :CHECK BIT |
| SMB 3 | | :MANIPULATIONS |
| SKE | | : |
| LDD 0,0 | | : |
| X 3 | | Bd > 2,0 |
| XAD 1,1 | | 9 > 1,1; 4 > A |
| XIS 1 | | 4 > 2,0; Bd > 3,1 |
| ING | | INPUT G PORT |
| X | | STORE |

| INSTRUCTION | RESULT | COMMENTS |
|---|---|---|
| CLRA | | |
| ASC | | CHECK ADD WITH CARRY |
| SC | | CHECK SET CARRY |
| SKC | | CHECK SKIP ON CARRY |
| LDD 0,0 | | |
| X | | STORE A |
| OMG | G = 9 | NO CHANGE |
| CLRA | | |
| ASC | | |
| X | | |
| OMG | G(9 > 10) | CARRY ADDS ONE TO MEMORY |
| CAMQ | | STORE A & M IN Q; 10,9 |
| XDS | | 9 > 3,1; 10 > A; Bd > 3,0 |
| X | | STORE 9 IN 3,0 |
| OMG | G(10 > 9) | |
| LD 2 | | 9 > A; Bd > 1,0 |

| INSTRUCTION | RESULT | COMMENTS |
|---|---|---|
| OMG | G(9 > 1) | |
| LD 3 | | 1 > A; Bd > 2,0 |
| OMG | G(1 > 2) | |
| ADD | | ADD WITHOUT CARRY |
| X | | STORE 3 IN 2,0 |
| SC | | |
| LDD 0,0 | | 7 > A |
| CASC | | CHECK CASC |
| SKC | | |
| X | | STORE 12 |
| OMG | G(2 > 12) | |
| CLRA | | : |
| AISC 3 | | : |
| X | | : |
| SC | | :CHECK |
| SKC | | :SKC/SC |
| X | | : |
| OMG | G(12 > 3) | |
| RC | | : |
| SKC | | :CHECK |
| X | | :RC |
| OMG | G(3 > 12) | : |
| LBI 0,0 | | :CHECK |
| LBI 1,15 | | ;SEQUENTIAL LBI'S |
| LBI 2,7 | | ALSO SKIPPED (LBI 2,7 NOT IN 410) |
| OMG | G(2 > 7) | |
| CQMA | | LOAD CONSTANTS FROM Q |
| OMG | G(7 > 9) | CHECK |
| X | | : |
| OMG | G(9 > 10) | : |
| LEI 1 | | |
| XAS | | STORE A − > S (9) |
| CLRA | | |
| AISC 7 | | : |
| SKGBZ 0 | | : |
| X | | :CHECK |
| OMG | | : |
| SKGBZ 1 | | : |
| X | | ;G BIT |
| OMG | G (10 > 7) | : |
| SKGBZ 2 | | : |
| X | | : |
| OMG | G(7 > 10) | :TESTS |
| SKGBZ 3 | | : |
| X | | : |
| OMG | G(10 > 7) | : |

| INSTRUCTION | RESULT | COMMENTS |
|---|---|---|
| SKGZ | | |
| X | | :CHECK |
| OMG | G(7 > 10) | : |
| OGI 0 | G(10 > 0) | :G TEST |
| SKGZ | | : |
| X | | : |
| OMG | G(0 > 10) | : |
| SKMBZ 0 | | |
| X | | CHECK MEMORY BIT TESTS |
| OMG | | NO CHANGE |
| SKMBZ 1 | | |

**TABLE I. Typical Test Sequence** (Continued)

| INSTRUCTION | RESULT | COMMENTS |
|---|---|---|
| X | | |
| OMG | G(10 > 7) | NO SKIP |
| SKMBZ 2 | | |
| X | | WON'T SKIP |
| OMG | G(7 > 10) | |
| INIL | | SEE THAT L LATCHES RESET |
| ININ | | ASSUME G − > I |
| SKE | | |
| X1 | | Br > 1 |
| OMG | | SHOULD BE EQUAL |
| INIL | | : |
| X | | : |
| SKMBZ 3 | | : |
| OBD | D(15 > 0) | :INIL TEST |
| OGI 1 | | : |
| LBI 3,11 | | : |
| OGI 0 | | : |
| INIL | | : |
| X | | : |
| SKMBZ 0 | | : |
| OBD | D(0 > 11) | : |
| NOP | | |
| XAS | | : |
| X | | :XAS TEST |
| OMG | G(10 > 9) | : |

| INSTRUCTION | RESULT | COMMENTS |
|---|---|---|
| LBI 0,0 | | LOAD RAM WITH |
| STII 7 | | CONSTANTS USING |
| STII 14 | | STII |
| STII 5 | | |
| STII 12 | | |
| STII 3 | | |
| STII 10 | | |
| STII 1 | | |
| STII 8 | | |
| STII 15 | | |
| STII 6 | | |
| STII 13 | | |
| STII 4 | | |
| STII 11 | | |
| STII 2 | | |
| STII 9 | | |
| STII 0 | | |
| LBI 1,0 | | |
| STII 7 | | |
| STII 14 | | |
| STII 5 | | |
| STII 12 | | |
| STII 3 | | |
| STII 10 | | |
| STII 1 | | |
| STII 8 | | |
| STII 15 | | |
| STII 6 | | |
| STII 13 | | |
| STII 4 | | |
| STII 11 | | |
| STII 2 | | |
| STII 9 | | |
| STII 0 | | |
| LBI 2,0 | | |
| STII 7 | | |
| STII 14 | | |
| STII 5 | | |
| STII 12 | | |
| STII 3 | | |
| STII 10 | | |
| STII 1 | | |
| STII 8 | | |
| STII 15 | | |
| STII 6 | | |
| STII 13 | | |

| INSTRUCTION | RESULT | COMMENTS |
|---|---|---|
| STII 4 | | |
| STII 11 | | |
| STII 2 | | |
| STII 9 | | |
| STII 0 | | |
| LBI 3,0 | | |
| STII 7 | | |
| STII 14 | | |
| STII 5 | | |
| STII 12 | | |
| STII 3 | | |
| STII 10 | | |
| STII 1 | | |
| STII 8 | | |
| STII 15 | | |
| STII 6 | | |
| STII 13 | | |
| STII 4 | | |
| STII 11 | | |
| STII 2 | | |
| STII 9 | | |
| STII 0 | | |

| INSTRUCTION | RESULT | COMMENTS |
|---|---|---|
| LBI 0,0 | | CHECK FOR RAM DATA |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |

| INSTRUCTION | RESULT | COMMENTS |
|---|---|---|
| LBI 1,0 | | CHECK FOR RAM DATA |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |

**TABLE I. Typical Test Sequence** (Continued)

| INSTRUCTION | RESULT | COMMENTS |
|---|---|---|
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |

| INSTRUCTION | RESULT | COMMENTS |
|---|---|---|
| LBI 1,0 | | CHECK FOR RAM DATA |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |

| INSTRUCTION | RESULT | COMMENTS |
|---|---|---|
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |

| INSTRUCTION | RESULT | COMMENTS |
|---|---|---|
| LBI 3,0 | | CHECK FOR RAM DATA |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |
| OMG | | OUTPUT DATA |
| LD | | : |
| XIS | | :MOVE TO NEXT DIGIT |

| INSTRUCTION | RESULT | COMMENTS |
|---|---|---|
| JMP X | INITIALIZE—SELECT ADDRESS X FOR OGI OR OMG (SELECT LBI FOR KNOWN DATA) | |
| RELEASE TEST MODE | OBD (SELECT B FOR KNOWN CONDITION) CHECKS JMP | |

**TABLE I. Typical Test Sequence** (Continued)

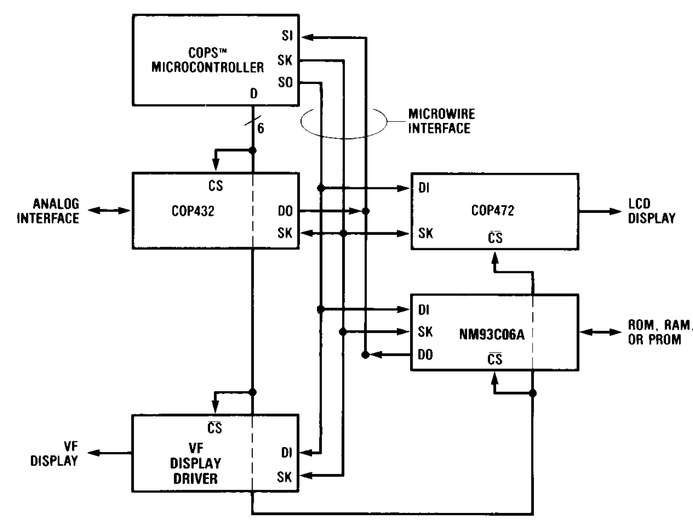
| INSTRUCTION | RESULT | COMMENTS |
|---|---|---|
| SET TEST MODE | | |
| JP X-2 | : | |
| JSR Y | CHECK JP & JSR | |
| RELEASE TEST MODE | "Y" SHOULD CHANGE THE OUTPUT | |
| | CONDITIONS OF "X" | |
| EXECUTE CODE (Y) | IF AT ALL POSSIBLE | |
| SET TEST MODE | | |
| RET | | |
| RELEASE TEST MODE | | |
| EXECUTE "X" AGAIN | VERIFIES RET | |
| SET TEST MODE | | |
| JP X-2 | | |
| JSRP Z | CHECK JSRP & RETSK | |
| RELEASE TEST MODE | | |
| EXECUTE CODE | "Z" SHOULD CHANGE "X" | |
| | OUTPUT CONDITIONS | |
| SET TEST MODE | | |
| RETSK | DON'T CHANGE Z CONDITIONS — | |
| | RETSK | |
| RELEASE TEST MODE | | |
| EXECUTE | | " " |
| SET TEST MODE | | |
| LOAD A & M TO | FIND VALUE OF ADDRESS IN BLOCK | |
| | (4 PAGES) | |
| VALUE OF ADDRESS | AT OR JUST BEFORE AN OUTPUT | |
| TO GO TO | CHANGE SET A & M TO ADDRESS | |
| OUTPUT CHANGE | OF "VALUE" | |
| JID | CHECKS JID | |
| RELEASE TEST MODE | | |
| EXECUTE OUTPUT | | |
| SET TEST MODE | LOAD A & M WITH A UNIQUE ADDRESS | |
| LOAD A & M | SUCH THAT CONTENTS OF THAT | |
| | ADDRESS WILL BE SEEN ON G | |
| LQID | | |
| X064 | ;OR USE THIS CAUSE THE DATA COMES | |
| | ;FROM YOUR TESTER ANYWAY | |
| CQMA | | |
| OMG | LQUID & CQMA CHECKED | |
| X | | |
| OMG | " | |
| INL | | : |
| OMG | G − > 2 | INL TEST (COPY OF 2nd BYTE) |
| X | | |
| OMG | G − > E | : |

This test sequence is not to be taken as a recommended test routine and is only shown as an example of what might be done to test various COPS parts. It is also advisable to approach measurements in the test mode with some caution. As stated earlier, one can force a large current into the SO node to place the chip in the test mode. Not only can this current do damage if unlimited, but it can also cause local current overloading such that some I/O conditions may be adversely affected. Obviously this will be more pronounced at higher $V_{CC}$ voltages. A specific example is that the L output current sink test should only be tested at a $V_{OUT}$ of 0.4V and 0.36 mA as the more stringent tests can exceed power limits when combined with the SO current.

## MICROWIRE™

National's super-sensible MICROWIRE serial data exchange standard allows interfacing to any number of specialized peripherals using an absolute minimum number of valuable I/O pins; this leaves more I/O lines available for system interfacing and/or may permit the COPS controller to be packaged in a smaller (and even lower cost) package. (MICROWIRE peripherals may also be used with non-COPS controllers). For further applications information, refer to COPS Briefs 8 and 9. MICROWIRE makes sense.

The example below illustrates the power and versatility of MICROWIRE via an extreme example—using one of each type of peripheral with a single controller.




TL/DD/6940–4

### COP431 SERIES, 8-BIT A/D CONVERTERS

The COP431 series is an 8-bit successive approximation A/D converter with a serial I/O and configurable input multiplexer with up to 8 channels. The serial I/O is configured to comply with the NSC MICROWIRE serial data exchange standard for easy interface to the COPS family of processors, and can interface with standard shift registers or other μPs.

The 2, 4 or 8 channel multiplexers are software configured for single-ended or differential inputs as well as channel assignment.

The differential analog voltage input allows increasing the common-mode rejection and offsetting the analog zero input voltage value. In addition, the voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 8 bits of resolution.

### COP472-3 LIQUID CRYSTAL DISPLAY CONTROLLER

The COP472-3 Liquid Crystal Display (LCD) Controller drives a multiplexed liquid crystal display directly. Data is loaded serially and is held in internal latches. The COP472-3 contains an on-chip oscillator and generates all the multilevel waveforms for backplanes and segment outputs on a triplex display. One COP472-3 can drive 36 segments multiplexed as $3 \times 12$ (4½ digit display). Two COP472-3 devices can be used together to drive 72 segments ($3 \times 24$) which could be an 8½ digit display.

### NM93C06A 256-BIT SERIAL ELECTRICALLY ERASABLE PROGRAMMABLE MEMORY

The NM93C06A is a 256-bit non-volatile memory. The device contains 256 bits of read/write memory divided into 16 registers of 16 bits each. Each register is serially read or written by the COP400 Family Controller. Written information is stored in a floating gate cell with at least 10 years of retention.

**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