COMPUTE

the Club Of Microprocessor Programmers, Users, and Technical Experts

Georgia Marszalek, Editor • David Graves, Editor • Doug Hall, Technical Consultant

Sponsored by National Semiconductor Corp., Santa Clara, Ca. 95051 **Vol. 3, No. 4, April, 1977**

M+S ANALYZER MAKES SC/MP TRANSPARENT

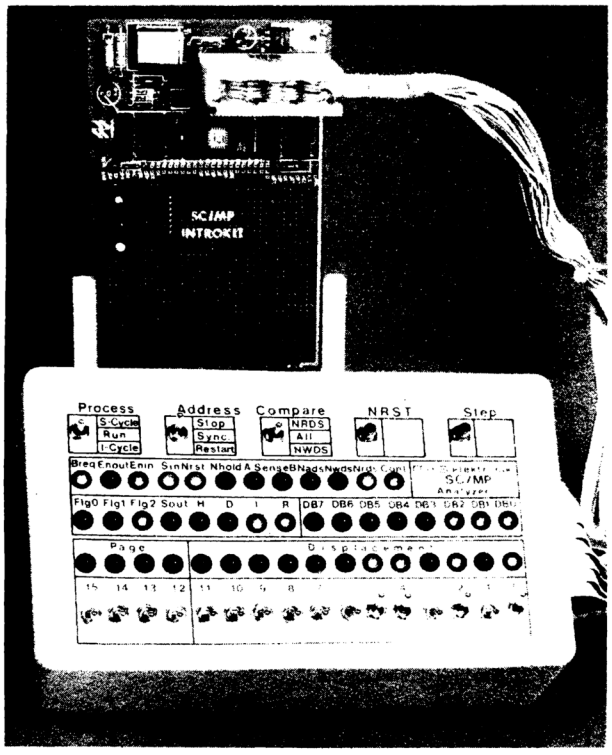
M+S electronik has developed a powerful tool for testing, debugging, and maintaining SC/MP microprocessor software and hardware.

The SC/MP Analyzer clips directly onto the SC/MP chip. The analyzer has a built-in power supply for powering the SC/MP Intro Kit. Additionally, the analyzer has CMOS inputs so that negligible loading to the chip occurs. There are 16 address select switches, 3 mode control switches, 2 push-buttons, and 44 LED indicators to provide communications between the kit and the user. The analyzer contains all the logic necessary to separate and latch all the control, address, and data bits for display.

There are two main modes of operation that are switch selectable to allow monitoring and debugging of system software in either static single-step mode or dynamic run mode.

For further information contact:

M+S elektronik GmbH
8751 Leidersbach, Bergstr. 2
West Germany



M+S SC/MP Analyzer Interfaced to SC/MP Intro Kit

SC/MP-II: A Fast, N-Channel, 8-Bit MPU

SC/MP-II is National's latest offering to the microprocessor marketplace. It's fast — one microsecond per microcycle, with a typical instruction execution time of five microseconds; it operates from a single +5 V supply; and it consumes less power than any other n-channel MPU on the market — less than 200 mW.

The on-chip clock makes use of very inexpensive 3.58- or 4.00-MHZ television-type crystals. Alternatively, SC/MP-II users can drive its clock from a standard TTL timing system. (Besides the clock, all inputs and outputs are TTL-compatible, and interface to MOS and CMOS circuitry.)

In addition to its self-contained timing circuitry, 16-bit (to 65K bytes) addressing capability, serial or parallel data transfer capability and common memory/peripheral instructions, SC/MP-II features:

- A bidirectional, Tri-State® 8-bit data bus
- Separate serial-data I/O ports
- Built-in flags and jump conditions
- Three user-accessible control-flag outputs
- An interrupt structure that has a fast response to asynchronous events
- On-chip bus allocation logic for multiprocessor system applications
- A delay instruction that simplifies timer systems.
- 46 control-oriented instructions
- On-chip handshake bus-access control
- Multiple addressing modes
- Two sense inputs
- A capability to interface with memories or peripherals of any speed

SC/MP-II (part no. ISP-8A/600) is available in a plastic or a ceramic package, and is fully compatible with the original SC/MP in terms of its pin configuration, object code, and software. And, with slight modification of its crystal frequency, SC/MP-II becomes compatible with all SC/MP support equipment.

In fact, we now have a *very* inexpensive SC/MP-II Retrofit Kit. This new kit includes a SC/MP-II CPU; a 2-MHz crystal; all resistors, capacitors, wires, etc.; and all necessary documentation. Designated the ISP-8K/205, the kit enables current users of the SC/MP Kit and SC/MP Keyboard Kit, and potential SC/MP users as well, to evaluate SC/MP-II's object code, software, and pin-out compatibilities.

Build the "Coffee Can Special" EROM Eraser

If you plan to reprogram your erasable PROMs (EROMs) you have probably noticed a general lack of information on cheap ultraviolet sources to do the job.

EROMs like the 1702, 1702A, 2704 and 2708 are erasable with high intensity ultraviolet light at a frequency of 2537Å. Many clothes dryers manufactured in the 1950s used a germicidal ultraviolet lamp to kill bacteria during the drying process. The lamp produces high intensity ultraviolet at 2537Å and can be purchased as a replacement part for less than $5 from your local appliance service departments.

The ultraviolet lamp was wired in series with a 25 to 40 W 115 VAC lamp to limit current in this application, so that the ultraviolet bulb is not burned out.

WARNING

Shortwave ultraviolet light can harm eyes and skin. Avoid looking into the lamp when it is lit.

A safe way to utilize the germicidal ultraviolet lamp as an EROM eraser is to mount it inside a 2 lb. coffee can. The current limiting incandescent lamp can be mounted on the outside, along with an AC toggle switch; in this way the ultraviolet source is not visible while in use and the incandescent lamp can serve as an indicator light — don't lift the can if the light is lit. Figure 1 shows how to arrange things.

The EROM to be erased is placed on a pedestal; its height should allow a one inch clearance between the ultraviolet lamp and the EROM. Erasing time is approximately 10 to 20 minutes. Remember to wire the two lamps in series (figure 1) or you will blow your ultraviolet lamp bulb. A mechanical egg timer with a loud "cling" can be used in conjunction with this unit to time the erasing interval.

© Copyright 1977, Byte Publications Inc.
Reprinted by permission of copyright owner.

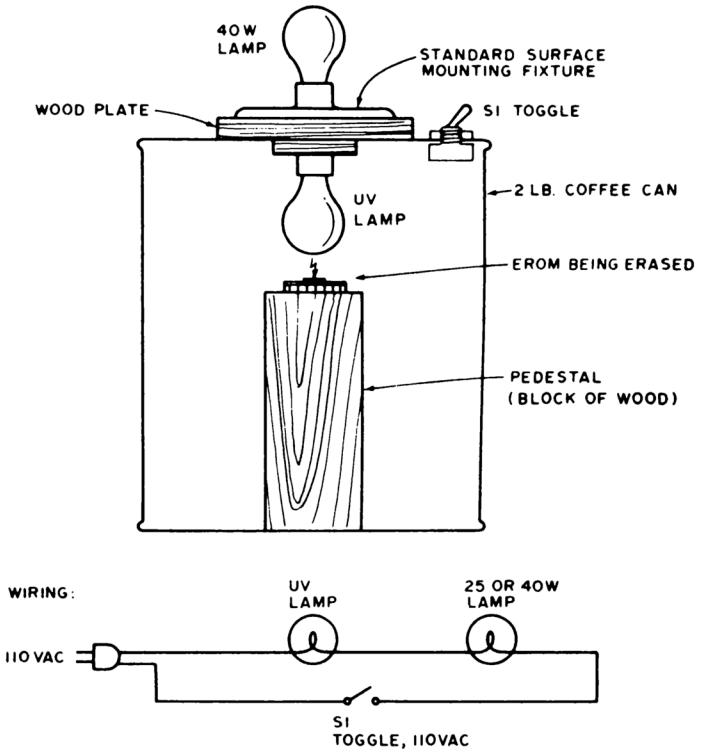


Figure 1: The physical assembly and circuit of the inexpensive EROM erasing apparatus. An ultraviolet bulb available as a replacement part for electric clothes dryers with germicidal cycles provides the source of the ionizing light needed to erase 1702, 2704, 2708 and similar EROM devices.

PRECISION PROGRAMMABLE VOLTAGE REGULATORS

Two new voltage regulator circuits provide specific output voltages to ±0.1% accuracy, *without external components.* Simply by wiring together given combinations of leads, the LH0075 supplies a positive, precise output of 2, 3, 5, 6, 10, 12, 15, or 18 volts; the LH0076 supplies a negative output of 3, 5.2, 6, 9, 12, 15, or 18 volts.

Or, by using a single external resistor, you can program the output voltage magnitude of these regulators to any value between zero and 28 V. And with two more resistors you add programmable output current limiting to 200 mA.

The input voltage range for the LH0075 is +8 V to +32 V; for the LH0076, −8 V to −32 V. Line and load regulations are 0.01%/V and 0.05%, respectively; 120-Hz ripple rejection is 65 dB, typical. The LH0075 and LH0076 are housed in 12-pin TO-8 cans.

Programming Tidbit

If you have the B level teletype firmware for PACE and are using DEBUG, there is a problem when trying to interrupt the listing by hitting the break key or space key on the teletype. The teletype will print pound signs continuously. If you have this problem the following changes in DEBUG are required.

memory location	new data
Ø342	15ED
Ø343	1901
Ø344	7EC7
Ø349	4B02
Ø34B	8000
Ø34C	7BFF
Ø34D	19FE
Ø34E	95F5
Ø34F	1565
Ø350	6400
Ø351	9B1C

the IMP-16 in communication applications

Application Note 134

INTRODUCTION

One of the most promising applications of LSI microprocessors is in the field of data communication. The improved reliability, flexibility and performance offered by these components makes them a natural for such tasks as message switching, smart terminal controllers, pre-processors and data concentrators. A microprocessor can handle many different codes, messages and line protocols; adapt to new operating requirements; and provide the needed real-time response to meet data transfer demands.

National's IMP-16 is a high performance 16-bit processor that is a very cost effective solution for data communication applications. Its powerful 16-bit instruction set, interrupt and input/output structure are the very qualities most important for these applications. The IMP-16 is equally adept at handling 8-bit character data or 16-bit arithmetic computations.

COMMUNICATION PROCESSOR FUNCTIONS

A communication processor may be programmed to provide a wide variety of functions at each level in the network shown in *Figure 1.* Some of these functions include:

1. Line protocol administration involving automatic dial-up, automatic answering and periodic polling of terminals to determine their status.

2. Adaptive line speed control for various speed terminals and communication lines.

3. Code conversion between terminals and the host processor or between dissimilar terminals.

4. Message assembly, syntax checking and reformatting to improve line efficiency and real-time response by off loading the host processor.

5. Data compression to improve bandwidth utilization.

6. Error control including detection of error codes (parity, CRC, block checks), detection of open line condition, inter character time-out, incorrect or no-poll responses, retransmission requests, and reporting unclearable errors.

7. Line monitoring, traffic analysis, message accounting and billing.

8. Task or address based message routing and switching.

9. Terminal control including formatting, cursor control, recognizing keyboard strokes, filling up buffers, data conversion, hard copy printouts, etc.

10. Terminal testing to locate trouble spots. Diagnostic programs can be "down line loaded" from the host or may be stored locally on a floppy disc or cassette.

11. Special functions such as communication instrument control, digital filtering or peripheral control.

ESSENTIAL COMPONENTS OF COMMUNICATIONS MICROPROCESSORS

Listed below are some of the desirable features of communications oriented microprocessors:

1. Multiple addressing modes with wide ranges. Indexed, indirect, immediate and relative addressing provide flexibility and simplify programming.

2. Good character handling, arithmetic and logic instructions for moving, manipulating and testing bits and bytes of data.

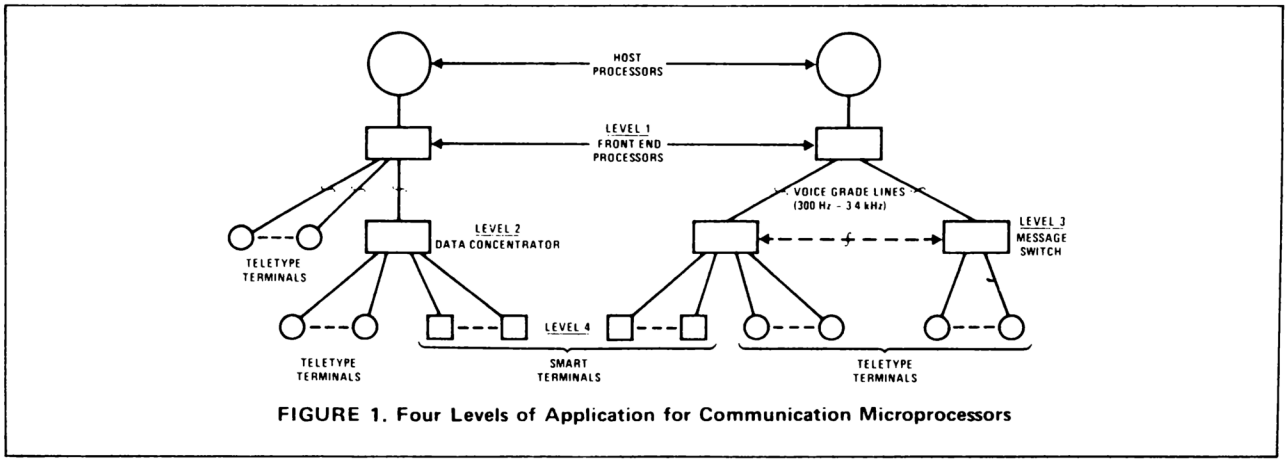


FIGURE 1. Four Levels of Application for Communication Microprocessors

3. Architectural features such as multiple accumulators, multiple index registers, status register, and internal stack. The internal stack can be used to minimize software overhead and excessive memory references during interrupt servicing and subroutine linking. When buffer memory is full temporary data may be saved on the stack.

4. Flexible input/output (I/O) to monitor and control different types of terminals or channels. User flags and jump condition inputs are helpful for serial I/O while parallel data transfer commands and the ability to execute memory reference instructions on peripherals speeds the data flow.

5. Ability to handle synchronous or asynchronous communication lines.

6. Fast responding interrupt capabilities to meet all data transfer demands. General interrupts for simple line or terminal control, multi-level interrupts for intermixing slow and high speed devices, vectored interrupts for very fast real-time response.

7. Direct Memory Access (DMA) capability for throughput enhancement and high speed block transfers of data. The DMA scheme should be limited only by the buffer memory cycle time and not by the microprocessor.

8. Error checking features including the ability to implement a variety of error control procedures.

9. Microprogrammability for special purpose instructions to improve speed and simplify assembly language programming. Code conversion, byte swapping, and Cyclic Redundancy Check (CRC) are examples of custom instructions that could be microcoded.

ENTER THE IMP-16

The IMP-16 meets every one of the above requirements! Furthermore, its 16-bit word length makes it a more efficient data handler than currently available 8-bit processors. Let's take a closer look at 16-bits vs eight.

1. As a pre-processor or front-end, a 16-bit processor increases system thorughput when communicating with the host processor. Since data is transferred as 16 parallel bits, only one instruction or DMA transfer is needed to communicate with a 16-bit host computer; two for a host with word length of up to 32 bits. An 8-bit processor would require twice as many transfers, slowing down the system.

2. A 16-bit instruction word can specify many more operations than an 8-bit instruction. This makes the 16-bit processor easier to program. If an 8-bit processor executes 16-bit instructions, two memory references are required to fetch the instruction. Only one such memory reference is required by the IMP-16, again boosting system throughput.

3. Any one of 64k locations can be addressed in a single instruction of a 16-bit machine. An 8-bit processor must resort to address computations involving double precision arithmetic and multiple registers to formulate 16-bit addresses.

4. Processing efficiency and accuracy are improved by operating on 16 bits in parallel. This is very important in error checking, making transmission line measurements, routing analysis of communication traffic, compilation of error statistics and customer billing.

5. Extra bits in a word may be used for various control or security options. These might include: control or data character, source or destination message, background or foreground, terminal or channel address, odd or even parity.

COMMUNICATIONS SYSTEM CONFIGURATIONS

In *Figure 2* the IMP-16 is a *front end processor* for a large host computer. It is supported by its own program and buffer memories and appears as another peripheral to the host. Communication is over the host's I/O bus on a program controlled or cycle steal basis. Conventional UART's are used as the telephone line interface to the microprocessor. AN-131 provides a detailed look at the IMP-16 in this application.

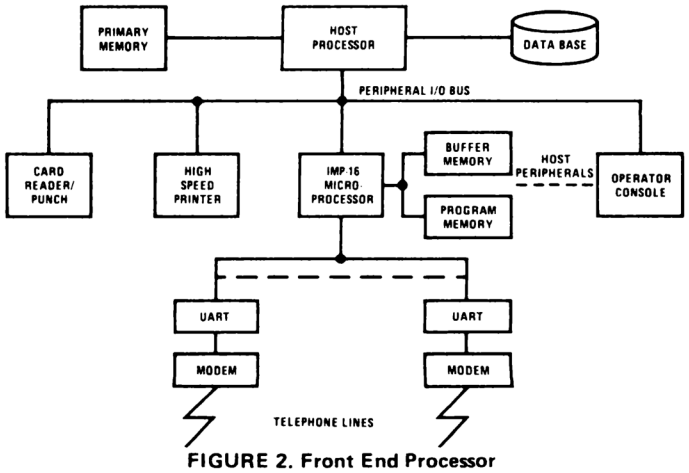


FIGURE 2. Front End Processor

A *data concentrator* or *message switch* will collect, buffer and pre-process information from several teletype or CRT terminals, and transmit that information over phone lines to a large scale computer or another terminal. The functional block diagram of *Figure 3* depicts this arrangement. The terminal controller provides a parallel interface to the IMP-16. The controller makes an interrupt request when it has an input character for the IMP-16 or when it is ready to receive an output character. The interrupt controller is essentially a priority encoder which informs the processor which terminal is requesting service. The powerful IMP-16 interrupt structure (see AN-107) is extremely important for this application. General, multi-level and vectored interrupts may be implemented depending on terminal load and response time requirements.

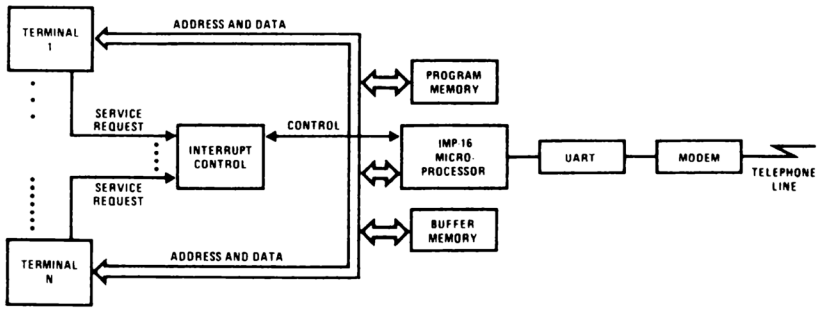


FIGURE 3. Data Concentrator or Message Switch

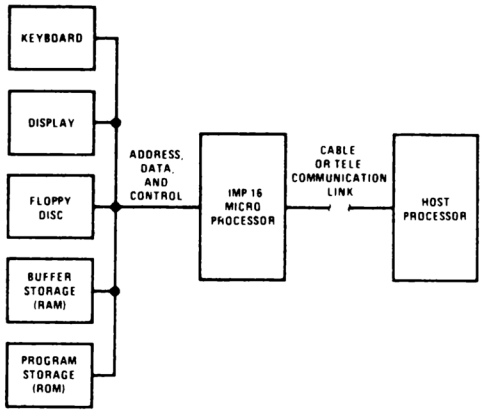


FIGURE 4. Smart Terminal

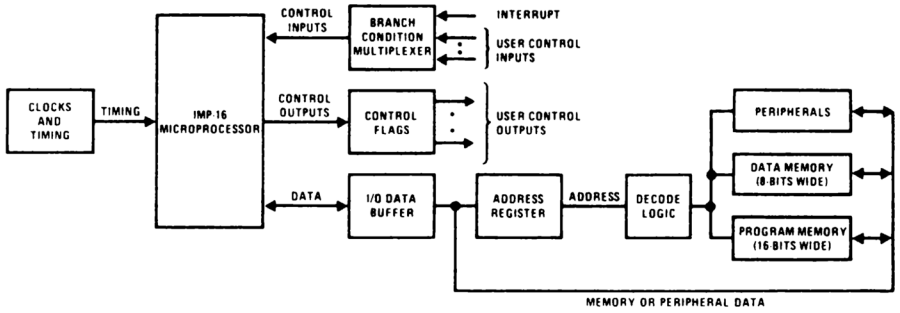


FIGURE 5. IMP-16 Microcomputer System

A *smart terminal* or remote job entry controller is shown in *Figure 4.* Remote processing by the IMP-16 is more cost effective than a central site computer handling a group of dissimilar terminals. The microprocessor control program is specifically tailored to the configuration at the remote site and can handle much of the processing locally. The resultant reduction in communication traffic with the central site provides savings in line costs while improving real-time response and message security.

An 8-bit data memory (RAM) is used with a 16-bit instruction memory (ROM) in this application. This technique eliminates the need to pack and unpack 8-bit characters while retaining the performance benefits of a 16-bit processor. *Figure 5* illustrates one typical implementation.

Characters may be input from the terminal one at a time while IMP-16/host transfers may be in two character (one 16-bit word) units. This maximizes both data handling speed and throughput.

PROGRAMMING EXAMPLES

The power and versatility of the IMP-16 instruction set enables efficient handling of a wide variety of tasks. A few examples will illustrate this point.

Status Monitoring

Consider the task of inputting an I/O status word from a terminal, testing the busy bit in that word and requesting data from the terminal when the bit is set. If the bit is zero we are to continue to monitor the status word. This coding is shown in Table I.

The terminal is addressed as if it were a memory location in this example. It is assumed the terminal controller address is in accumulator 2 so that any arbitrary address can be used when accessing the status or data word from

Table I

```
; BIT 15 OF STATUS WORD IS BUSY BIT
; = 0 IDLE
; = 1 BUSY
; ADDRESS OF TERMINAL CONTROLLER IN ACCUMULATOR 2
; STATUS = TERMINAL STATUS WORD ORDER
; DATA = TERMINAL DATA WORD ORDER
   LD          0, STATUS (2)        ; INPUT STATUS WORD
   BOC         2, -1                ; LOOP TILL READY
   LD          0, DATA (2)          ; INPUT DATA FROM TERMINAL
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   Cycles:     14 + N(10)           where N is the number of times the
                                    status bit is zero when tested
               (196μs for 14 cycles)
   Words:      3
```

the terminal. This technique provides fast handling (5 cycles of 7μs to obtain the data) and programming flexibility (the data can be input into any one of three accumulators) through use of the Load (LD) instruction. Note that AC2 contains the terminal controller address. The Branch-On-Condition (BOC) instruction is used to test bit 15.

In the second example (Table II) we will input the status word, test bits 0 and 1 and take appropriate action. If bit 0 is set we will input data from the terminal and store in memory; if bit 1 is set we'll store the previous data from the terminal; if neither bit is set an alternate action is taken.

This routine might be used in a terminal controller application where two terminals A and B are assigned status bits one and zero, respectively. The data that is to be stored in SAVE will come from the terminal that is ready (status bit set) with terminal B tested first. If neither terminal is ready an alternate action (possibly a recording of the condition) is taken.

To test any number of arbitrary status bits we may shift or rotate the status bit into bit positions 0, 1 or 15 and use the BOC 3, 4 or 2 instruction as required. Alternatively we can use a mask word to test bits in parallel.

Bit Masking

In the example of Table III the status word is inputted and bits are tested for zero via the Skip if AND is Zero (SKAZ) instruction. If any bit is set, data will be input from a peripheral and added to the partial sum in accumulator 2. Note that peripheral input and addition are done in one instruction which takes only five cycles (7μs).

Data Transmission

After assembling a message block, characters must be transferred to the host or communications line. Using load (LD) and store (ST) instructions an effective rate of 130k 8-bit chars/sec (1.04 Megabits/sec) is achieved. This requires a great deal of program storage and is therefore not efficient for large data transfers.

The program controlled loop in Table IV will provide an effective transfer rate of almost 80k characters/second while using only five memory words for instructions. Two 8-bit characters are read from memory and sent out. Up to 127 16-bit words (254 characters) could be transmitted in this fashion.

There are 18 cycles in the loop. To transfer 100 8-bit characters (50 words) takes 901 cycles or 1261.4μs.

Table II

```
TERMB = TERMINAL B DATA ORDER
 TERMINAL CONTROLLER ADDRESS IN ACCUMULATOR 2
; PREVIOUS TERMINAL A DATA IN ACCUMULATOR 1
          LD      0, STATUS (2)
          BOC     3, GDATA            TEST BIT0   BRANCH IF TERMINAL B
          BOC     4, GDATA + 1        , TEST BIT1   BRANCH IF TERMINAL A
 TAKE ALTERNATE ACTION HERE
          •
          •
          •
GDATA:    LD      1, TERMB (2)        , GET TERMINAL B DATA
          ST      1, SAVE             , STORE THE DATA
          •
          •
          •
SAVE:          + 1                    ; MEMORY LOCATION RESERVED FOR
                                       , TERMINAL DATA
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
          Cycles:  21 if bit 0 set     (29.4μs)
                   25 if bit 1 set     (35μs)
                   13 if neither bit set (18.2μs)
          Words:   6
```

Note: Execution time is based on 1.4μs/cycle.

Table III

```
   TERMINAL CONTROLLER ADDRESS IN ACCUMULATOR 2
 , PARTIAL SUM IN ACCUMULATOR 3
          LD        0, STATUS (2)          INPUT STATUS WORD
          SKAZ      0, MASK                , MASK & TEST FOR 0
          ADD       3, DATA (2)            ADD PERIPHERAL DATA TO AC3
          •
          •
          •
 MASK:    WORD      X'F51A                 , MASK 9 BITS
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
          Cycles:   10 if masked bits are zero  (14μs)
                    17 if any masked bit is one (23.8μs)
          Words:    4
```

Table IV

```
NWORDS = NUMBER OF WORDS TO BE TRANSFERRED
OUT    ADDRESS OF HOST OR COMMUNICATIONS LINE
            LD      3, OUTADR       ; HOST ADDRESS IN AC3
            LI      2, NWORDS       ; NO. WORDS TO BE TRANSMITTED
LOOP:       LD      0, IN  1(2)       16 BIT WORD FROM BUFFER MEMORY
            ST      0, OUT (3)      , OUTPUT THE WORD TO HOST
            AISZ    2, -1           ; DECREMENT & TEST = 0
            JMP     LOOP            , LOOP BACK
            •
            •                       , TRANSMISSION COMPLETE
            •
            .BSECT
IN:             + NWORDS            , NUMBER OF WORDS IN BUFFER MEMORY
OUTADR:     WORD X'4000             , ADDRESS OF HOST COMPUTER
```

Table V

```
MESGID = MESSAGE ID WORD ORDER
, (OUTADR) IN ACCUMULATOR 3
            LD          2 ADTABL      , ADDRESS OF POINTER TABLE IN AC2
            LD          1, MESGID (3) ; INPUT MESSAGE ID WORD
CHECK       ISCAN                     , SCAN FOR BIT = 1
            JMP         NOMESG        , NO MESSAGES LEFT
            JSR         @ (2)         , SERVICE THE MESSAGE
            JMP         CHECK           LOOP BACK
NOMESG.     •
            •
            •
            •
            •
ADTABL:       WORD      TABLE 1       , ADDRESS OF POINTER TABLE
TABLE:        WORD      LINE 1        ; ADDRESS OF LINE 1 SUBROUTINE
            •           •             •
            •           •             •
            •           •             •
              WORD      LINE 16       , ADDRESS OF LINE 16 SUBROUTINE
```

Thus one character is transferred in 12.614μs providing a transfer rate of approximately 79.4k chars/sec. This provides double the throughput of an 8-bit processor with equal cycle speed.

Data transmission may be speeded up at the expense of program storage. Placing more LD and ST instructions in the loop and decrementing accumulator 2 by the number of pairs of these instructions makes the loop control instructions less of a factor in the data transfer. The actual speed vs memory tradeoff is resolved by analyzing system throughput requirements.

Message Routing and Interrupt Handling

A message switch is responsible for assigning messages to communication lines or local terminals. Assume that each bit set in a message identification word corresponds to a message which is to be serviced by the corresponding line or terminal. Up to 16 messages are to be serviced in a prioritized fashion, with bit 0 assigned the highest priority. Coding is shown in Table V.

This routine may also be used for interrupt servicing with the Interrupt Select Status Word replacing MESGID (3). The ISCAN is an extended IMP-16 instruction which shifts bits out of accumulator one until a one is detected. The number of shifts is added to accumulator two and the next instruction is skipped if accumulator one is not zero. Otherwise, the next instruction is executed.

Error Control

Cyclic Redundancy Check (CRC) is a very efficient method of accounting for transmission errors in the data link. All single bit odd number errors, all double errors, all burst errors less than the length of the CRC word and most of the burst errors longer than the CRC word can be detected.

The algorithm and associated IMP-16 coding in *Figures 6a and 6b* implement a CRC process on parallel 16-bit data. The generation polynomial chosen for this example is $1 + x^5 + x^{12} + x^{16}$, and the initial conditions are defined to cause the CRC word to start with all 1's.

The time required to go through the main loop of the program is about 582μs for 16 bits; this works out to about 37μs per bit per check, comparing quite favorably with corresponding minicomputer rates. The 16-bit format of the IMP-16 microprocessor is largely responsible for the efficiency of the instruction set and the resulting speed of execution of programs.

MEASURING PERFORMANCE

A good yardstick for evaluating data throughput is provided by measuring the time required to transfer a byte (or other convenient unit of data) of information from a peripheral device to a system memory under control of the microprocessor. This operation involves addressing the device, addressing the memory location,

executing the data transfer and updating the memory address. The resulting time can be expressed conveniently as a bit rate (kilobits or megabits per second).

The IMP-16 microprocessor has a throughput rate of 1.04 Megabits measured on this basis. As comparison, other microprocessors range from 75 Kbits (4-bit machines) to 840 Kbits (8-bit machines).

AN-152 illustrates two examples of IMP-16 telephone line communications.

In conclusion, it can be seen that the characteristics of the IMP-16 make it adaptable to various communication

```
SETUP:   COUNT  ←  16          ; INITIALIZE BIT COUNT
         CRC    ←  −1          ; INITIALIZE CRC WORD
         POLY   ←  1021₁₆      ; BINARY EQUIVALENT OF CRC POLYNOMINAL

LOOP:    (A) ← DATA(7) ⊻ CRC(15)
         CRC ← ↑ CRC
         IF A THEN CRC ← CRC ⊻ POLY
         DATA ← ↑ DATA
         COUNT ← COUNT −1
         IF COUNT ≠ 0, GO TO LOOP
         GET NEXT DATA WORD
```

FIGURE 6a. CRC Algorithm

```
CRCSHI
     1                          TITLE  CRCSHIFT
     2
     3        0003     ODD      =      3
     4        0002     SEL      =      2
     5        000A     CYOV     =      10
     6
     7                 ,        JSR    FCRC
     8                 ,        NORMAL RETURN          AC0=CRC IN 2 8 BIT BYTES.
     9                          ON ENTRY
    10                          AC3= ADDRESS OF P(X)
    11                          AC2= G(X)
    12                          AC0= NUMBER OF BYTES +2 IN P(X)
    13                          IF NUMBER OF BYTES ARE ODD
    14                          FIRST BYTE IS IN BITS 8 TO 15
    15                          BITS 0 TO 7 ARE ZERO
    16
    17                 FCRC
    18 0000 AD18 A             ST     3,POINT         ;STORE POINTER TO LIST.
    19 0001 1315 A             BOC    ODD,SEXTB       ;SET EXTRA BYTE
    20 0002 4F10 A             LI     3,16            ;SET BIT COUNT
    21                 LETSG
    22 0003 5CFF A             SHR    0,1              MAKE WORD COUNT
    23 0004 A115 A             ST     0,WORDC         ;STORE WORD COUNT
    24 0005 4CFF A             LI     0,-1            ;GET INITIAL COND OF REG.
    25 0006 2101 A             JMP     +2              FIRST TIME
    26                 CRCBK
    27 0007 4F10 A             LI     3,16            ;SET BIT COUNT
    28 0008 9510 A             LD     1,@POINT        ;GET NEXT P(X) TERM
    29 0009 730F A             ISZ    POINT           ;UPDATE ADDRESS
    30                 CRC.
    31 000A 3000 A             RADD   0,0             ;SHIFT CRC WORD.
    32 000B 1A07 A             BOC    CYOV,CK1        ;BRANCH IF MSB IS A 1
    33 000C 3500 A             RADD   1,1             ;SHIFT AC1 1 LEFT
    34 000D 1A07 A             BOC    CYOV,XOR        ;IF MSB A 1 DO XOR.
    35                 OUT
    36 000E 4BFF A             AISZ   3,-1            ;CHECK IF NEW WORD NEEDED.
    37 000F 21FA A             JMP    CRC             ;NO.
    38 0010 7D09 A             DSZ    WORDC           ;UP DATE WORD COUNT,SKIP IF DONE.
    39 0011 21F5 A             JMP    CRCBK           ;KEEP GOING.
    40 0012 0200 A             RTS                    ;DONE RETURN.
    41                 CK1
    42 0013 3500 A             RADD   1,1             ;SHIFT LEFT 1
    43 0014 1AF9 A             BOC    CYOV,OUT        ;IF MSB =0 DO XOR.
    44                 XOR.
    45 0015 3882 A             RXOR   2,0             ;NOW XOR.
    46 0016 21F7 A             JMP    OUT             ;GET NEXT
    47                 SEXTB
    48 0017 4F08 A             LI     3,8             ;SET BIT COUNT TO 8.
    49 0018 21EA A             JMP    LETSG
    50
    51 0019 0000 A   POINT      WORD   0
    52 001A 0000 A   WORDC      WORD   0
    53
    54        0000              END
```

FIGURE 6b. CRC Coding

environments. The results of 16-bit operations on 8-bit data is increased flexibility, performance and throughput. The potential and power of the IMP-16 can be used to serve a wide segment of the communication market.

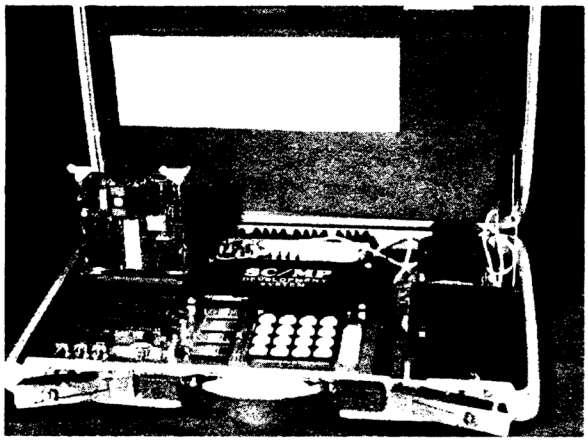
UP and RUNNING at NORTRON

At Nortron the decision was made to select a microprocessor which would meet our immediate requirements for a low cost, easy to use device coupled with an inexpensive evaluation system. The National Semiconductor Corporation's SC/MP and the LCDS (low cost development system) offered us an inexpensive way to "get into microprocessors" at our company.

Almost as soon as our new LCDS was unpacked it became obvious that competition for its use would be a problem. We had the choice of having a stationary machine located in a noisy corner of the lab or making a portable machine which could be easily moved from office to office. We chose the latter concept.

A nice looking but inexpensive three inch attaché case was purchased at a local discount store. The low height available precluded the use of the LCDS base so it was discarded. An aluminum base was formed to provide a vertical mounting for the heat sinks used on the power supplies and stand offs were used on the horizontal section to support the LCDS printed circuit board. The one inch space under the P.C. board was used to house the power supply filter capacitors (we needed to connect many smaller values in parallel to meet the height and capacitance requirements.)

The power transformer for the +5 volt and -12 volt supplies was mounted in a space between the P.C. board and the right hand side of the case. It was mounted as close to the hinges of the case top as possible to give the case stability when it was sitting in the handle-up position. A socket was mounted on the transformer mounting chassis to allow completely disconnecting the power cord so that the cord could be stored in the lid compartment. The power supply regulators were the standard



TO-3 can 3 terminal type. One regulator was used for the -12 volt supply and three were used for the +5 volt supply. It was necessary to place three very low value series Shermistors (TM) between the output terminals of the devices to prevent oscillation. The 5 volt supply was tested to 5 amps without excessive heat being generated in the brief case.

To further facilitate the portability of the unit, a digital modem was designed which used a 3.58 MHz T.V. crystal. This unit was also placed under the LCDS P.C. board. A 256 byte program was written which then allowed the LCDS to load and unload information through the modem and onto a small hand held audio cassette recorder. The recorder rests in a form pad in front of the transformer and to the right of the P.C. board.

We store the microprocessor card and 2K RAM card in conductive plastic bags and place them in the lid flap along with the LCDS manuals. Each user has his own cassette tape so the RAM card can be quickly reloaded with the user's program each time he uses the system.

Paul Kendall
Dan Pike
Don Sherman

MICROPROCESSOR SPARES

For information on ordering, pricing, and delivery of spare parts, write or call:

Microprocessor Service Center
National Semiconductor Corp.
2921 Copper Road
Santa Clara, CA 95051
(408) 737-6270 or 6279

Assembly No.	Description
CARDS	
9301952	IMP-16L Programmers Panel I-Face Board
9801953	STD Programmers Panel Board
9802085-1	IMP-IPC Card Reader/TTY I-Face Board
9802085-2	IMP-IPC Card Reader/TTY I-Face Board (W/CRT)
9802086	IMP-16P Programmer I-Face Board
9802573	IPC Development CPU Board
9802708	IMP-16P Card Reader/TTY I-Face Board (W/PROMs)
CARDS	
9803120	IPC Programmer Panel I-Face Board
9803122	IPC Card Reader/TTY I-Face Board (W/PROMs)
9803217	IPC Development CPU Cable and Paddle Board
9804640	Floppy Disc I-Face Board (W/PROMs)
HARDWARE	
4004786	Power Supply +5V @ 18A, -12V @ 3A (STD)
4004183	Power Supply +5V @ 30A, -12V @ 10A (Heavy Duty)
6012337	Programmer Panel I-Face Cable
SOFTWARE	
2054514	Diskette, Blank (IBM #2305830)
FIRMWARE	
4100937	MM5214J ROM Scamp Kit Debug
9355094	IMP to IPC Conversion Firmware (6 PROMs)

the Bit-Bucket

Dear Ms. Marszalek:

I would appreciate a listing of user library SL0027A SC/MP Math Package.

Having just received the November issue (Vol. 2, NO. 11) of the Compute Newsletter, I was very interested in Hal Chamberlins article on the IMP-16 microcomputer system. If back issues of the newsletter are available which contain parts 1 and 2 of this series, I would appreciate receiving them.

I would also be interested in knowing if there are any listing yet available to use as test routines for the SC/MP kit.

Thank you. Sincerely yours,

Alton D. Floyd, Ph.D.
Assistant Professor, Anatomy
Indiana University School of Medicine
Bloomington, Indiana 47401

SL0027A is reprinted in Appendix D of the SC/MP Microprocessor Applications Handbook. On page D-4 of this manual a CCL instruction should be inserted between lines 105 and 106. By completing the form in Vol. 2 #12 all the members of Compute can obtain a copy.

Other copies of this manual can be ordered for $5.00 each from Marketing Services/MS520, National Semiconductor, 2900 Semiconductor Dr., Santa Clara, CA 95051.

Vol. 2 No. 6 and Vol. 2 No. 7 contain Part 1 and Part 2 of Hal's article on the IMP-16 and can be ordered along with other back issues.

The only other programs we currently have in the library for SC/MP are SL0041, SQRT, a program for finding the square root of a 15-bit positive integer and the NIBL program available for $15.00. This price includes a source listing, paper tape, and instructions for using National Semiconductor's version of TINY BASIC. This package requires at least 4K of memory for the program and addition memory for the NIBL programs you may write.

Dear People –

Will you send me a copy of the instruction set for your PACE microprocessor? I am willing to pay a nominal amount for it (e.g., $1 is acceptable, $20 is not).

Mostly this is to satisfy my curiosity: I have an instruction set for another microprocessor and began wondering about others. (I read an article about your PACE in the October issue of "Byte" magazine.)

Perhaps someday I'll get my own computer (I've always wanted one), but not this year as I am much too confused for making an intelligent decision.

Leigh Janes
23B Robbins Lane
Rocky Hill, CT 06067

A complimentary copy of the PACE instruction set is on the way.

Dear Sir:

Do you know if National has any Application Notes or other literature on a DMA implementation for the IMP-16P? I want to add a dual floppy disk that works with DMA to my IMP-16P system. I would rather not reinvent the wheel if this has been done before.

I would appreciate any help you may give me.

Sincerely,

John Linder, Engineer
New Mexico State University
Physical Science Laboratory
Box 3548
Las Cruces, New Mexico 88003

The only information I have are the Applications notes

AN142	*Using a Microprocessor Beyond apparent speed*
AN135	*IMP-16L DMA*

AN142 may be of interest since it discusses cycle stealing to speed up I/O and references the IMP-16P. The IMP-16L is an early version of the IMP-16 which was designed to support DMA type application.

A FAMILY OF RAM SUPPORT CIRCUITS

We'd like you to know about several of our interface products that are ideal for 16K RAM support applications.

The DS3642/DS3672, for example, are fast, Schottky-clamped, dual TTL-to-MOS clock drivers with the high output current and voltage capabilities needed to drive high-capacitance (to 500 pF) MOS memories. Perfect for CAS/RAS driver use, the DS3642/DS3672 feature a bootstrapping pin which, when tied to the output through a small capacitor, eliminates the need for an additional supply to provide a higher voltage to the output stage.

The DS3644/DS3674 – another pair for your consideration – are quad TTL-to-MOS clock drivers, again well suited to CAS/RAS applications. These parts, which are pin and function compatible with Intel's 3235 and Motorola's MC3460, feature two common Enable inputs, a Refresh input, and a clock control input – all of which simplify system design. Again, the DS3644/DS3674 use Schottky-clamped circuitry, and drive highly capacitive loads at high speeds.

Schottky-clamped, the DS3648/DS3678 are quad, two-input address multiplexer/drivers with Tri-State® outputs. A pnp input structure minimizes input loading, which reduces driver loading in large memory systems. The DS3648/DS3678 outputs drive loads to 500 pF.

Finally, the DS3649 series of hex, Tri-State, TTL-to-MOS drivers have the same general features as the rest of the family – low input loading, high-capacitance drive capability, etc., etc. These high-speed Schottky-clamped circuits have an output rise time of 9 ns (maximum) into 50 pF, 35 ns (maximum) into 500 pF, and are ideal buffer/drivers for the control of signal and address lines.

SC/MP HOMEBREW COMPUTER SYSTEM

by Dan Grove, Microprocessor Instructor, Santa Clara

With a SC/MP IC, a few other off-the-shelf ICs, and the following schematic and control programs, you can build a low-cost key-entry computer suitable for tinkering around the home (i.e., burglar alarms, lawn sprinkler controller, etc.).

As shown in Figure 1, the system contains a 256 by 8-bit PROM, a 256 by 8-bit RAM, data entry switches, and a display panel.

The PROM contains a program to allow data to be entered into the RAM from the eight addressable data switches. (You can keep the physical size of the unit to a minimum by using a 16-pin mini switch IC for the data entry switches. The miniswitch IC contains eight switches that are manipulated with a pencil point.)

The addressable display panel is useful for system debug. With it you can insert breakpoints that store any data you want to see followed by a programmed halt. Figure 2 shows how the programmed halt may be implemented.

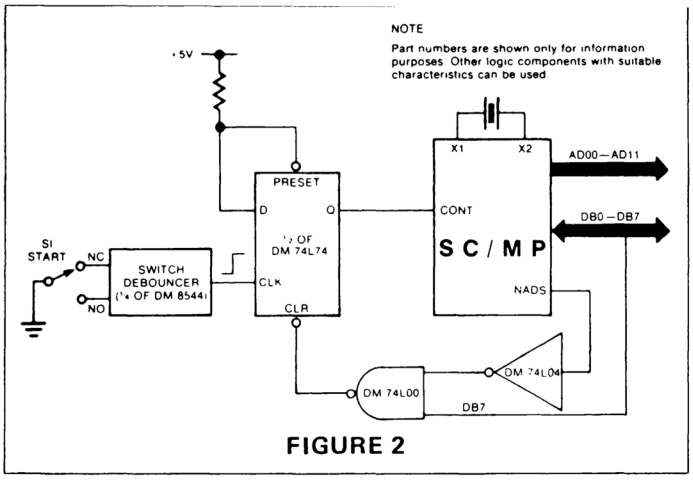


FIGURE 2

SC/MP is TTL compatible, and each output can drive a 1.6 mA TTL load. So, if you drive only MOS, CMOS or low power TTL, it needs no buffering.

SC/MP Timing Element used is a 1000 pf capacitor across pins 37 and 38. Any crystal up to 1 Mhz will also work across the same pins. The only advantage of the more expensive crystal is software timing accuracy.

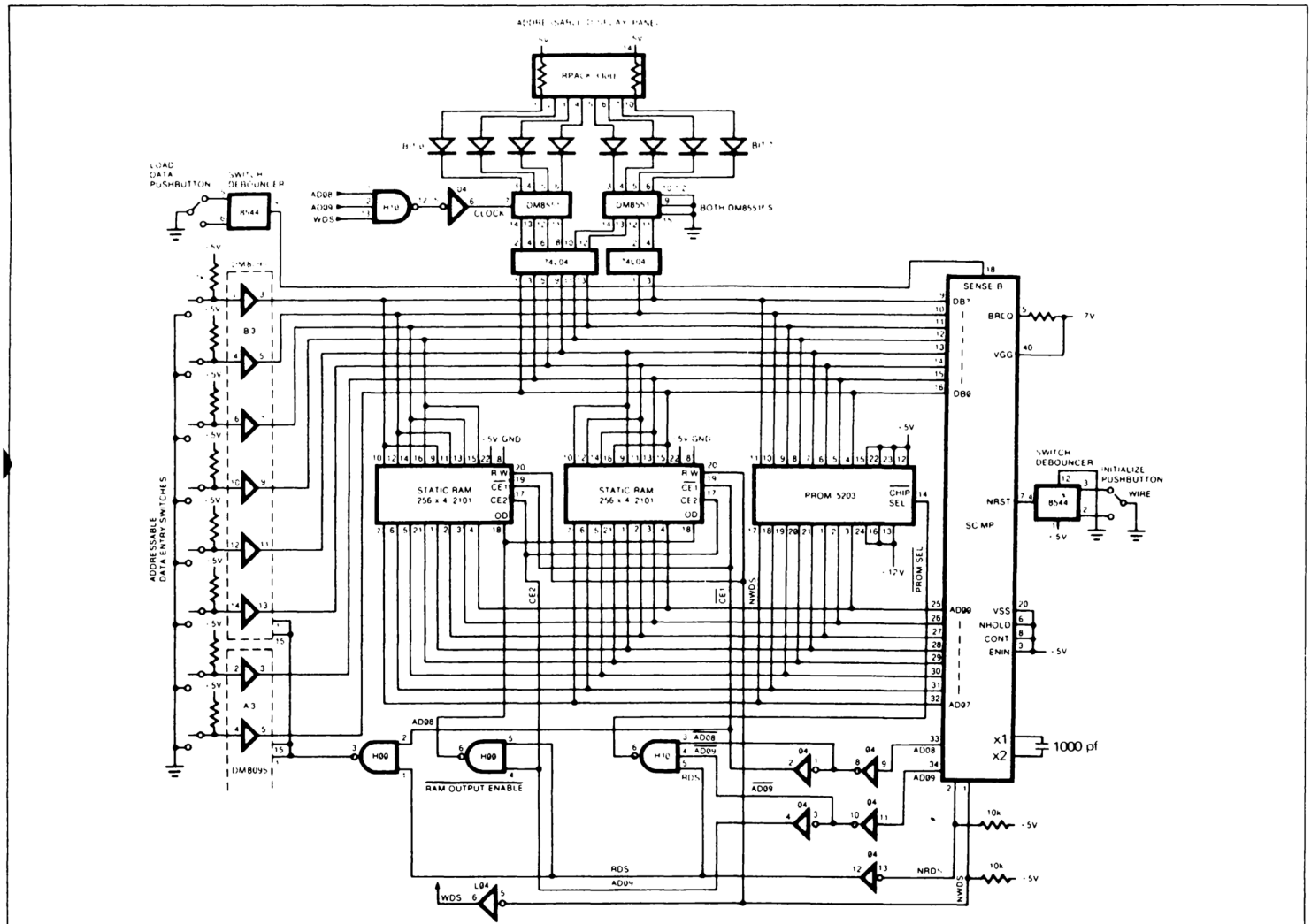


FIGURE 1

When initially debugging the system the display panel clock can be tied to various strobes to monitor sections of the system to determine if they are functioning without error. For example, a useful strobe for determining what is being written into RAM is the Write Data strobe.

Only one of the two sense inputs is used, so the second one is available for system use as sense A or as the interrupt input.

A peripheral device you intend only to write data into can share address 11 with the LEDs.

Address Bits		Device	Operation
9	8		
0	0	PROM	READ
0	1	SWITCHES	READ
1	0	RAM	READ/WRITE
1	1	LEDS	WRITE

The two programs (and flowcharts) shown are used to load the RAM and then transfer control to the RAM. The second program is the more versatile of the two, but it requires more locations in PROM than the first program. Program one is probably easier to understand since it is well documented. ▨

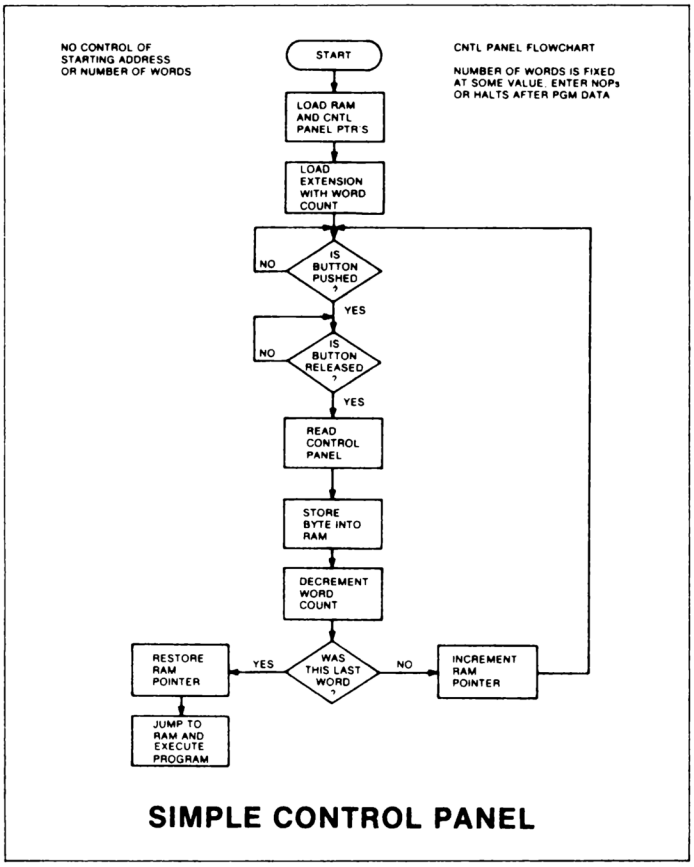
Reference: SC/MP Technical Description

SIMPLE CONTROL PANEL PROGRAM

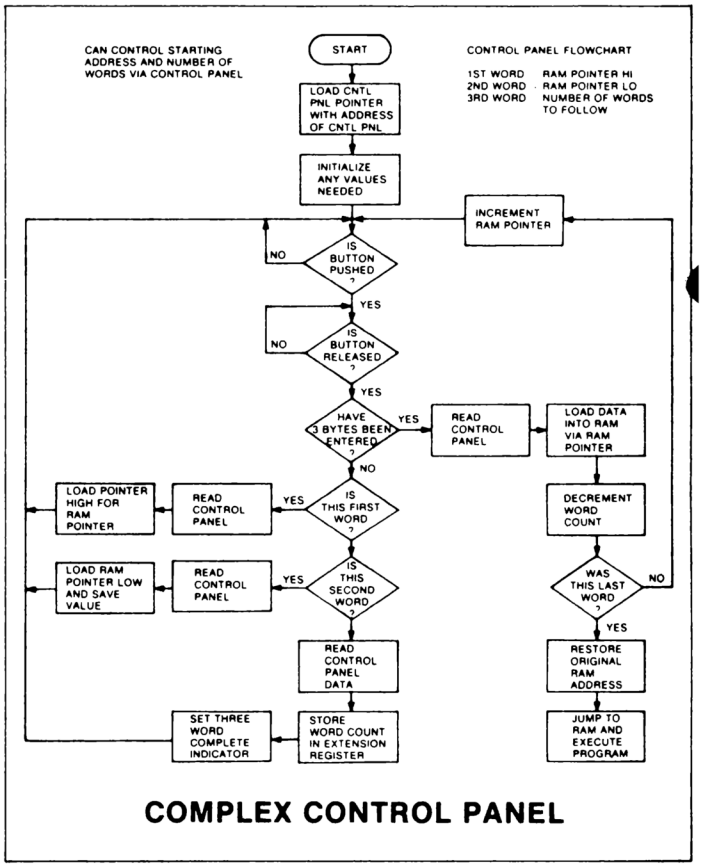
```
1                       .TITLE CPANEL
2  0000 08                      NOP             ;GO TO 0001
3  0001 C402                    LDI   2         ;AC=0002
4  0003 35                      XPAH  1         ;RAM PTR=P1
5  0004 C400                    LDI   0         ;AC=0
6  0006 31                      XPAL  1         ;LO P1=0
7  0007 C401                    LDI   1         ;AC=0001
8  0009 36                      XPAH  2         ;CTRL PTR=P2
9  000A C41E                    LDI   30        ;# OF WORDS
10 000C 01                      XAE             ;E REG=# WORDS
11 000D 06              PUSH:   CSA             ;AC=SR
12 000E D01A                    AND   MASKI     ;SB BIT
13 0010 98FB                    JZ    PUSH      ;LOOP IF
14                              ;NO PUSH
15 0012 06              REL:    CSA             ;AC=SR
16 0013 D015                    AND   MASKI     ;SB BIT
17 0015 9CFB                    JNZ   REL       ;LOOP TIL
18                              ;RELEASE
19 0017 C200                    LD    (2)       ;AC=CTRL PNL
20 0019 CD01                    ST    @1(1)     ;RAM=PNL
21 001B 01                      XAE             ;# OF WORDS
22 001C 02                      CCL
23 001D F4FF                    ADI   -1        ;# OF WORDS-1
24 001F 9803                    JZ    RUN       ;ALL INPUT DONE
25 0021 01                      XAE
26 0022 90E9                    JMP   PUSH
27 0024 C400            RUN:    LDI   0
28 0026 31                      XPAL  1
29 0027 91FE                    JMP   -1(1)
30 0029 20              MASKI:  .BYTE 020
31      0000                    .END

MASKI   0029     PUSH   000D
REL     0012     RUN    0024

NO ERROR LINES
SOURCE CHECKSUM = 9837
```



SIMPLE CONTROL PANEL



COMPLEX CONTROL PANEL

COMPLEX CONTROL PANEL PROGRAM

```
 1                .TITLE PNL
 2  0000 08               NOP
 3  0001 C401             LDI   1
 4  0003 37               XPAH  3
 5  0004 C403             LDI   3
 6  0006 31               XPAL  1
 7  0007 C403             LDI   3
 8  0009 01               XAE
 9  000A 06       TEST:   CSA
10  000B D420             ANI   020
11  000D 98FB             JZ    TEST
12  000F 06       WAIT:   CSA
13  0010 D420             ANI   020
14  0012 9CFB             JNZ   WAIT
15  0014 31               XPAL  1
16  0015 982B             JZ    LDDATA
17  0017 31               XPAL  1
18  0018 C403             LDI   3
19  001A 60               XRE
20  001B 980A             JZ    STHI
21  001D C402             LDI   2
22  001F 60               XRE
23  0020 980D             JZ    STLO
24  0022 C401             LDI   1
25  0024 60               XRE
26  0025 9813             JZ    CNT
27  0027 C300     STHI:   LD    (3)
28  0029 36               XPAH  2
29  002A C402             LDI   2
30  002C 01               XAE
31  002D 90DB             JMP   TEST
32  002F C300     STLO:   LD    (3)
33  0031 32               XPAL  2
34  0032 C300             LD    (3)
35  0034 35               XPAH  1
36  0035 C401             LDI   1
37  0037 01               XAE
38  0038 90D0             JMP   TEST
39  003A C400     CNT:    LDI   0
40  003C 31               XPAL  1
41  003D C300             LD    (3)
42  003F 01               XAE
43  0040 90C8             JMP   TEST
44  0042 31       LDDATA: XPAL  1
45  0043 C300             LD    (3)
46  0045 CE01             ST    @1(2)
47  0047 C4FF             LDI   -1
48  0049 02               CCL
49  004A 70               ADE
50  004B 9803             JZ    EXC
51  004D 01               XAE
52  004E 90BA             JMP   TEST
53  0050 35       EXC:    XPAH  1
54  0051 32               XPAL  2
55  0052 92FE             JMP   -1(2)
56       0000             .END


CNT     003A      EXC     0050
LDDATA  0042      STHI    0027
STLO    002F      TEST    000A
WAIT    000F

NO ERROR LINES
SOURCE CHECKSUM = 967B
```

A number-crunching MICROPROCESSOR

Our MM57109 is a digit-oriented microprocessor intended for number-crunching applications in instrumentation, test equipment, process controllers, navigation systems — even MPU/minicomputer extensions.

Usable as a stand-alone processor (with external ROM/PROM memory and a program counter) or as a peripheral device on the bus of an MPU or a mini, the MM57109 features scientific calculator instructions (RPN), flexible I/O, branch control, and easy interfacing.

Because its algorithms are preprogrammed and stored in an on-chip ROM, programming the MM57109 is performed in calculator keyboard-level language, which means that software development is simple and the generated code is reliable.

With the MM57109 data or instructions can be synchronous or asynchronous; digit count, calculation mode, and error control are user programmable; and there is single-bit control of the sense input and the flag outputs.

EDITORIAL

We would like to broaden the scope of the reviews section of *COMPUTE*, and we would like your help.

In addition to books and articles in trade publications, we would like to begin review material from other newsletters, non-trade publications, etc. Articles on computer art and music, films and fiction, or novel uses for computers are all fair game.

This is where you come in. If you have read a good article that should be shared with other *COMPUTE* members, drop us a few lines describing it, being sure to include enough information to enable someone else to obtain a copy. Or if you can't do that, or if copies would be hard to obtain, a Xerox copy would be greatly appreciated. Some of the smaller articles we will try to print in *COMPUTE* (with permission of course). OK people, the ball's in your court. Remember this is your newsletter.

single-chip character generator

National Semiconductor has introduced the industry's first single-chip character generator. Primarily for CRT displays, the DM8678's combination of low cost, low power, and low component count makes it applicable as well to home video games and standard TV sets.

The bipolar LSI circuit is a 64-character unit housed in a standard 16-pin DIP. It performs such system functions as parallel-to-serial shifting, character address latching, character spacing, and character-line spacing without additional packages, and does it more cheaply than is possible in present systems in which a character-generating ROM requires two to four additional chips. The DM8678 is unique also in that the on-chip ROM is mask programmable and can carry either 7-by-9 or 5-by-7 CRT fonts.

NEW LIBRARY PROGRAMS

IMP-16 PROGRAM SL0038A TAPE
(from Walter Probert, Rexnord Inc.,
P.O. Box 2022, Milwaukee, WI 53201)
Source paper tape $5.00 each

TAPE is a program that tests two tapes to see if they are the same.

```
 1                .TITLE TAPE ,'CMTP-40-0 4/21/76'
 2 0000           .ASECT          ;THIS PROGRAM CAN BE USED TO TEST
 3 0000           .=0             ;SAMENESS OF TAPES. OR USING ONE
 4 0000 9801 A    .WORD 09801     ;TAPE IT CAN HELP IDENTIFY FAULTY
 5 0001           .=1             ;TTY OR HIGH SPEED TAPE READER.
 6 0001 000A A    .WORD 0A
 7 0002           .=9             ;LOAD ABSTTY. RESIDES IN 0 THRU 2F
 8 0009 0000 A    .WORD 0         ;AND USES 2F THRU 1FFF (8K) WORDS
 9 000A           .=10
10       0001 A REQ0     =       1
11       0000 A R0       =       0
12       0001 A R1       =       1
13       0002 A R2       =       2
14       0003 A R3       =       3
15 000A 2824 A TAPE:    JSR     BEGIN   ;INIT. PUSH RUN.
16 000B A300 A LP1:     ST      R0,(R3)
17 000C 4B01 A          AISZ    R3,1
18 000D 49FF A          AISZ    R1,-1
19 000E 200B A          JMP     LP1
20 000F 0000 A          HALT            ;LOAD TAPE. PUSH RUN.
21 0010 2824 A          JSR     BEGIN
22 0011 2829 A          JSR     LEADER
23 0012 2014 A          JMP     .+2
24 0013 2C2C A LP2:     JSR     @GETC   ;READ TAPE INTO MEMORY.
25 0014 A300 A          ST      R0,(R3) ;TAPE MAY EXCEED MEM & WILL
26 0015 4B01 A          AISZ    R3,1    ;HALT. MARK TAPE FOR 2ND SET.
27 0016 49FF A          AISZ    R1,-1   ;AT END OF SHORTER TAPES. PUSH
28 0017 2013 A          JMP     LP2     ;HALT. SET PC = 19. THEN
29 0018 0000 A H1:      HALT            ;LOAD TAPE. PUSH RUN.
30 0019 2824 A          JSR     BEGIN
31 001A 2829 A          JSR     LEADER
32 001B 201D A          JMP     .+2
33 001C 2C2C A LP3:     JSR     @GETC
34 001D F300 A          SKNE    R0,(R3) ;COMPARE TAPE/MEM
35 001E 2020 A          JMP     .+2     ;READ PC(-1) TO FIND HALT LOC.
36 001F 0000 A          HALT            ;OK IF TAPE IS FINISHED.IF NOT
37 0020 4B01 A          AISZ    R3,1    ;PUSH RUN TO SEEK MORE ERRORS.
38 0021 49FF A          AISZ    R1,-1
39 0022 201C A          JMP     LP3
40 0023 0000 A          HALT            ;OK IF NOT HALTED BEFORE.
41 0024 4C00 A BEGIN:   LI      R0,0
42 0025 A009 A          ST      R0,9
43 0026 8C2D A          LD      R3,START
44 0027 842F A          LD      R1,RANGE
45 0028 0200 A          RTS
46 0029 2C2C A LEADER:  JSR     @GETC   ;START TAPE.
47 002A 11FE A          BOC     REQ0,.-1
48 002B 0200 A          RTS
49 002C 7E3B A GETC:    .WORD 07E3B
50 002D 002F A START:   .WORD 02F
51 002E 1FB1 A RANGE:   .WORD 01FB1
52       000A A          .END TAPE


BEGIN  0024  A    GETC   002C  A
H1     0018  A*   LEADER 0029  A
LP1    000B  A    LP2    0013  A
LP3    001C  A    R0     0000  A
R1     0001  A    R2     0002  A*
R3     0003  A    RANGE  002E  A
REQ0   0001  A    START  002D  A
TAPE   000A  A


NO ERROR LINES
SOURCE CHECKSUM =3A49
```

SC/MP Program SL0041A SCSQRT

Source paper tape $5.00 each

SCSQRT takes the square root of a 15-bit positive integer in memory. The result is an 8-bit positive integer. The largest input value is 32,768.

```
 1                .TITLE  SCSQRT , ' SC/MP SQUARE ROOT'
 2                ;SL0041A  SC/MP  COMPUTE LIBRARY  11/76
 3                ;
 4                ;       ED SCHOELL
 5                ;       NS ELECTRONICS AUSTRALIA
 6                ;       13 OCT 1975
 7                ;
 8                ;THIS ROUTINE TAKES THE SQUARE ROOT OF
 9                ;THE 15-BIT POSITIVE NUMBER IN LS,MS AND
10                ;LEAVES THE RESULT IN LS AS AN 8-BIT
11                ;POSITIVE NUMBER.
12                ;
13                ;THIS IS TOP OF STACK, RESULT IS ON
14                ;STACK (CORRECTED) RANGE OF NUMBERS IS
15                ;32768 AS INPUT GIVING 181 AS RESULT
16                ;IN LS.  RESULT IS INTEGER WHOSE SQUARE
17                ;IS LESS THAN ENTRY NUMBER.
18                ;
19                ;TIME IN 172 +N(158) +<N-128(126)FOR N>128>
20                ;IE  1320 MICROSEC FOR N=1
21                ;    40.8 MILLISEC FOR N=128
22                ;    70.8 MILLISEC FOR N=181  (N IS RESULT)
23                ;
24                ;
25                ;
26                ;RESULT IS INTEGER,IE SQRT OF 25 IS 5
27                ;AND 5 IS GIVEN AS RESULT  FOR ANY INPUT
28                ;IN RANGE OF 25 TO 35. 36 GIVES 6 AS RES.
29                ;
30                ;P2 IS USED AS STACK POINTER TO RAM
31                ;P1 AND P3 ARE NOT USED.
32                ;
33                ;ALGORITHM USED IS (N+1)^2-N^2=2N+1
34                ;SUCCESSIVE VALUES OF 2N+1 ARE SUBTRACTED
35                ;FROM THE NUMBER AS N IS ACCUMULATED UNTIL
36                ;THE ORIGINAL NUMBER IS LESS THAN ZERO.
37                ;
38                ;EG....7 SQUARED IS 49
39                ;       6 SQUARED IS 36
40                ;       DIFFERENCE...13...=2*6+1
41                ;
42                ;
43                ;EXAMPLE OF STACK USAGE.
44                ;
45                ;        REL       ENTRY          EXIT
46                ;        ------------------------------
47                ;       -1
48                ;(P2)-> 0           7F             00
49                ;       +1          F9       (P2)->B5
50                ;
51                ;   IE SQRT OF 7FF9 (32,768) IS B5 (181)
52                .PAGE
53       0000   MS       =       0
54       0001   LS       =       1
55 0000 00      SQRT:    HALT
56 0001 02               CCL
57 0002 C400             LDI     0
58 0004 FA01             CAD     LS(2)
59 0006 CA01             ST      LS(2)
60 0008 C400             LDI     0
61 000A FA00             CAD     MS(2)
62 000C CA00             ST      MS(2)
63                ;BOTH PARTS NOW TWO'S COMPLEMENTED.
64 000E C400             LDI     0
65 0010 01               XAE             ;EXT REG STORES N
66 0011 03      LOOP:    SCL             ;ADD 1
67 0012 40               LDE
68 0013 F201             ADD     LS(2)   ;ADD N+1 TO COMP NO
69 0015 CA01             ST      LS(2)   ;AND SAVE IT
70 0017 06               CSA             ;TEST CARRY
71 0018 940B             JP      NOCRY1
72 001A C200             LD      MS(2)   ;CARRY INTO MS
73 001C F400             ADI     0       ;ADD IT IN
74 001E CA00             ST      MS(2)
75 0020 06               CSA             ;TEST FOR END
76 0021 D480             ANI     080     ;MASK CRY BIT
77 0023 9C11             JNZ     EXIT
78 0025 40      NOCRY1:  LDE             ;CRY IS CLEAR
79 0026 F201             ADD     LS(2)   ;ADD N TO COMP NO
80 0028 CA01             ST      LS(2)
81 002A 06               CSA             ;TEST CRY
82 002B 9410             JP      NOCRY2
83 002D C200             LD      MS(2)   ;CRY TO MS
84 002F F400             ADI     0
85 0031 CA00             ST      MS(2)
86 0033 06               CSA             ;TEST FOR END?
87 0034 9407             JP      NOCRY2
88 0036 C601    EXIT:    LD      @1(2)   ;ADJ STACK
89 0038 40               LDE
90 0039 CA00             ST      MS(2)   ;PUT SQRT TO STACK
91 003B 9006             JMP     CONT    ;COULD BE XPPC (IE PTS)
92 003D 40      NOCRY2:  LDE
93 003E F401             ADI     1       ;INC N
94 0040 01               XAE
95 0041 90CE             JMP     LOOP
96 0043 00      CONT:    HALT            ;PROG TERMINATES HERE
97       0000            .END

CONT   0043    EXIT   0036
LOOP   0011    LS     0001
MS     0000    NOCRY1 0025
NOCRY2 003D    SQRT   0000 *

NO ERROR LINES
SOURCE CHECKSUM = FA02
```

MICROPROCESSOR

Courses

	EASTERN TRAINING CENTER	WESTERN TRAINING CENTER
MICROPROCESSOR FUNDAMENTALS	May 2-5 June 6-9	June 6-9
SC/MP APPLICATIONS	May 9-12 June 13-16	May 2-5 June 20-23
PACE APPLICATIONS	May 16-19 June 20-23	June 13-16
ADVANCED PROGRAMMING	May 23-26 June 27-30	May 9-12

Training Center Locations

National Semiconductor Corporation
Eastern Training Center
1320 South Dixie Hwy., Suite 870
Coral Gables, Florida 33146
Telephone: (305) 661-7969 or 661-7971

National Semiconductor Corporation
Western Microprocessor Training Center
1333 Lawrence Expwy., Suite 430
Santa Clara, CA 95051
Telephone: (408) 247-7924

SC/MP WORKSHOPS IN EUROPE!

BELGIUM

May (dates to be confirmed)
Location: Brussels
Language: French
Contact: J. P. LEMAIRE
Rampe Gauloise la
1020 Bruxelles
Telephone: 0032-02-4784847
Telex: 24610

DENMARK

June 20-22
Location: Copenhagen
Language: Danish
Contact: MULTIKOMPONENT A/S
Herstedvangen 7c
DK-2620 Albertslund
Telephone: 02-644477
Telex: 19155

ITALY

May 24-27
June 14-17
September 13-17
Location: Milano
Language: Italian
Contact: ADELSY S.P.A.
Via Domenicino 12
Milano 20749
Telephone: 02-4985051
Telex: 204339423

NETHERLANDS

Dates: May 12-14
Location: Rijswijk
Language: Dutch
Contact: Cor Vromans
RODELCO B. V. ELECTRONICS
P.O. Box 296
29 Verrijn Stuartlaan
Rijswijk ZH 2109
Telephone: 0031-70-995750
Telex: 32506

SPAIN

May 3
Location: Madrid
Language: Spanish
Contact: Hans Gotz
ORTRSD
Calle la Sofora 13
Madrid 20
Telephone: 2790809
Telex: 27556

UNITED KINGDOM

Dates: Walk-in
Location: Berkshire
Language: English
Contact: N-SIGN
P.O. Box 119
Reading
Berkshire RG31NQ, England
Telephone: Reading (0734) 594911
Telex: 849391
Dates: On request
Location: Kent
Language: English
Contact: JERMYN INDUSTRIES
Vestry Estates
Sevenoaks
Kent, England
Telephone: Sevenoaks (0732) 50144
Telex: 95143

Contact your local National distributor, sales office, or Phil Hughes, National Semiconductor, Germany, for details of the SC/MP workshops and seminars.

On The Road With SC/MP

SC/MP Applications

Portland, OR	May 16—20	(503) 292-3534
Detroit, MI	May 17—19	(313) 477-0400
Calgary, Alberta, Canada	May 23—27	(403) 273-4630
Philadelphia, PA	May 24—26	(215) 628-8877

SC/MP Advanced Applications

Pittsburgh, PA	May 3-5	(412) 782-3770
Minneapolis, MN	May 10-12	(612) 888-3060
N. Haven, CO	May 17-19	(203) 226-7527

UNITED STATES
COMPUTE/208
National Semiconductor Corp.
2900 Semiconductor Drive
Santa Clara, CA. 95051
Tel: (408) 247-7924
TWX: 910-338-0537

GERMANY
National Semiconductor GmbH
808 Fuerstenfeldbruck
Industriestrasse 10
Tel: 08141/1371
Telex: 05-27649

AUSTRALIA
NS Electronics Pty Ltd.
Cnr. Stud Road & Mtn. Highway
Bayswater, Victoria 3153
Tel: 03-729-6333
Telex: 32096