27/
3-30-77
25 to NTIS

ORNL/TM-5773

# The NDT-COMP8 Microcomputer

C. V. Dodd
G. D. Connell

OAK RIDGE NATIONAL LABORATORY
OPERATED BY UNION CARBIDE CORPORATION FOR THE ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION

# DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

# DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
Price: Printed Copy $4.00, Microfiche $3.00

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the Energy Research and Development Administration/United States Nuclear Regulatory Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

ORNL/TM-5773

Contract No. W-7405-eng-26

METALS AND CERAMICS DIVISION

# THE NDT-COMP8 MICROCOMPUTER

C. V. Dodd and G. D. Connell

Date Published - March, 1977

NOTICE
This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

**NOTICE** This document contains information of a preliminary nature. It is subject to revision or correction and therefore does not represent a final report.

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

## CONTENTS

# THE NDT-COMP8 MICROCOMPUTER

C. V. Dodd and G. D. Connell*

## ABSTRACT

An 8080-based microcomputer, the NDT-COMP8, has been developed for control and data reduction of nondestructive tests. The computer is on a single 9.35 in. by 6.2 in. printed circuit board, has up to 4 K of ROM (read only memory), 4 K of RAM (random access memory), 1 serial I/O port (that can be jumpered for TTL, RS232C or 20 ma current loop, with baud rates from 75 Hz to 4800 Hz), and 72 parallel I/O ports. A monitor for program development, a floating-point math package, and a PROM simulator/programmer have been developed to aid in the application of the board.

## I. INTRODUCTION

The NDT-COMP8 Microcomputer is an 8080-based microcomputer designed for control and data reduction of a broad range of nondestructive tests. A digital computer was needed to replace the analog computers in our modular eddy-current instruments and to provide additional control and data-reduction capabilities. The computers and boards made by Intel, Altair, and other manufacturers were too expensive and too large to fit in existing equipment. The computer was designed to provide on a single board the maximum capability at a minimum cost and still be made to fit inside the necessary module. The small size, low cost, low power consumption, large memory, large number of input-output lines and versatility of the computer make it ideal for a large number of diverse applications. Those applications for which this computer has been used thus far will be discussed in greater detail in separate reports. For example, the NDT-COMP8 has been incorporated into two different single frequency eddy-current instruments, one of which controls 23 different coils and measures the coil-to-conductor lift-off spacing, and the other of which makes magnitude-and-phase readings that are used to compute the resistivity and coil-to-conductor spacing for a single sample. The computer has also been installed in dual-frequency and three-frequency instruments and has been used to compute the desired specimen parameters (e.g., conductivity, thickness, lift-off, etc.) from the magnitude-and-phase readings at each frequency. Although difficult to program, the computers have performed excellently in all applications.

The purpose of this report is to provide the documentation and a "user's guide" for the NDT-COMP8 microcomputer for both present and future applications. The software already developed for this microcomputer makes additional applications much simpler. The technical support furnished by Intel Corporation, for both hardware and software, and the universal acceptance of the 8080 microprocessor make the development of advanced systems which are compatible with this one highly probable. Personnel with a limited background in microcomputers can adapt this computer to their particular problems. The only additional hardware requirements are a terminal and a power supply.

*Summer student 1976.

# II. COMPUTER DESCRIPTION

## A. General Description

A general block diagram of the NDT-COMP8 microcomputer is shown in Fig. 1. The system consists of a two-phase crystal-controlled clock generator (8224) which furnishes the timing requirements for the 8080 microprocessor (CPU). The microprocessor sends both control and data signals to the 8228 system controller and receives data from the 8228 over the same lines.

There are three busses in the computer (the address bus, the control bus, and the data bus) which provide communication links between a number of different parts of the computer. The address bus is controlled by the 8080 microprocessor (during normal operation). The control bus signals are generated by the 8228 system controller, and the data bus is controlled by either the 8228, the memory or the input-output chips, depending on the state of the control bus. The memory consists of read only memory (ROM) and random access (read-write) memory (RAM). The input-output (I/O) chips consist of one serial I/O chip, the 8251, and three parallel I/O chips, the 8225s.

## B. Timing Diagrams

The system operation depends on the proper signals being presented by the clock, the microprocessor, the system controller, the memory, and the I/O devices at the proper time. Although more complete descriptions are available from Intel and Advanced Micro Devices, some of the more important timing signals are shown in Fig. 2. The 8224 chip furnishes a two-phase clock signal to the computer, $\emptyset_1$ and $\emptyset_2$. A machine cycle consists of three or more clock cycles. The time for one clock cycle is 488 ns, as determined

ORNL-DWG 76-19090

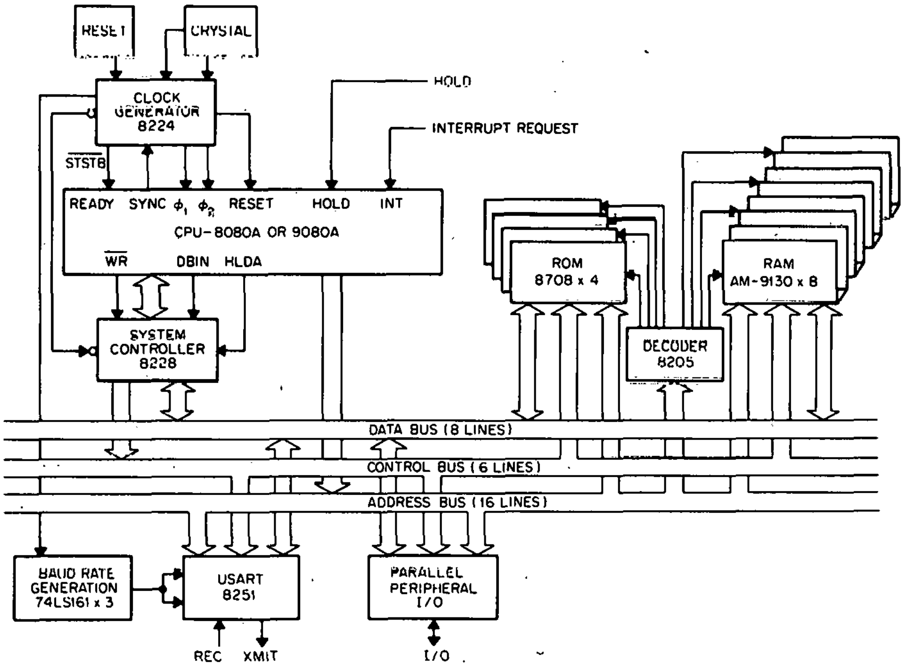

**Fig. 1. Block diagram of the NDT-COMP8 microcomputer.**

ORNL-DWG 76-19089

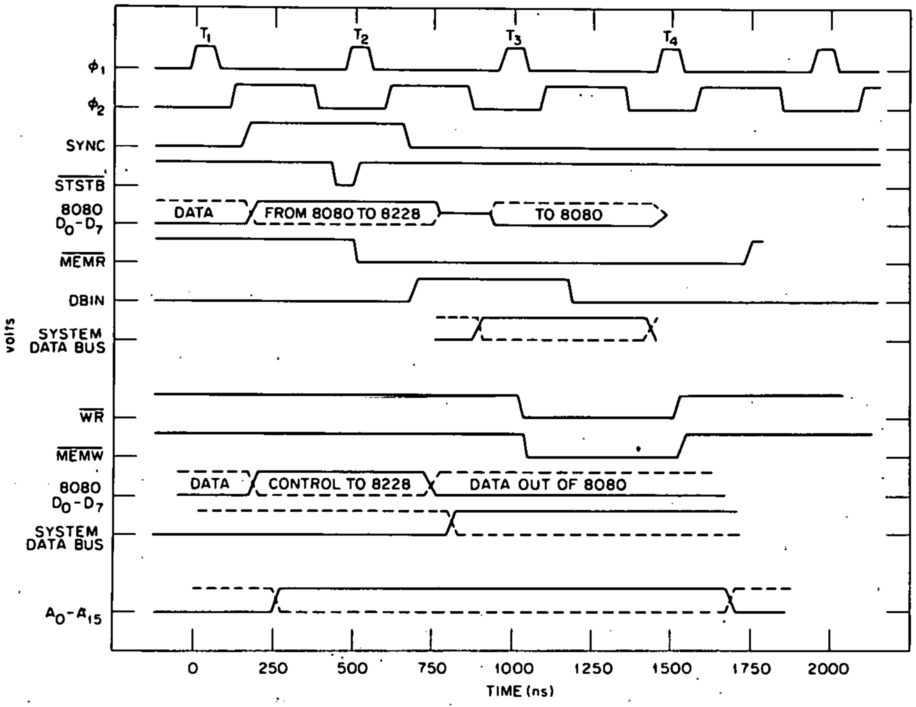

**Fig. 2. Timing diagrams for the NDT-COMP8 microcomputer.**

by the 18.432 MHz crystal (which was chosen in turn to give the proper baud rates). A SYNC signal is supplied by the computer to identify the start of each machine cycle. At the start of each machine cycle, status information that identifies the type of cycle is furnished to the 8228 system controller from the 8080 over the $D_0$-$D_7$ lines. The status strobe ($\overline{\text{STSTB}}$) from the 8224 clock latches the status information into the 8228. The information appears on the control bus as determined by the $\overline{\text{STSTB}}$, the DBIN, the $\overline{\text{WR}}$ and the HLDA signals.

The first type of cycle shown is a memory-read cycle. The $\overline{\text{MEMR}}$ line is latched low, and the 8080 data lines and 8228 data lines are floated (go to a high-impedance state). The $\overline{\text{MEMR}}$ line going low generates the signal that causes the memory chips addressed by $A_0$-$A_{15}$ to display their data on the data bus. The access time for the Intel memory (ROM) is measured from the time the address lines are stable, but the access time for the AMD memory (RAM) is measured from the time the $\overline{\text{MEMR}}$ or $\overline{\text{MEMW}}$ signals are present. The data on the system bus must be stable 55 ns (worst case) before the start of the $T_3$ clock cycle. The data on the bus are transferred to the $D_0$-$D_7$ lines of the 8080 by the 8228 system controller.

In the memory-write machine cycle (which does not occur in the same machine cycle as a read), the $\overline{\text{MEMW}}$ line is latched low by the $\overline{\text{WR}}$ signal from the 8080 at the start of the $T_3$ clock cycle. (The status signals are latched on the 8228 in the same manner.) The data are transferred from the 8080 to the 8228 data bus and are written into the memory chips selected by the address lines, $A_0$-$A_{15}$, while the $\overline{\text{MEMW}}$ line is low.

The operation of the AM-9130 memory chips is started with a chip-enable (CE) signal, which is obtained by feeding the $\overline{\text{MEMW}}$ and $\overline{\text{MEMR}}$ signals to a NAND gate. The addresses must be present when this signal is received since the CE latches the addresses.

The timing signals for other types of machine operations are similar to these described in this section and are covered in the manufacturers' manuals.

The computer generates a baud rate signal by dividing the 18.432 MHz signal from the clock down to the appropriate frequency. This signal is fed to the 8251 USART serial I/O port, which sends serial information into and out of the computer. Also, up to 72 lines of parallel I/O can be sent through three 8255 programmable peripheral interfaces. The modes of both the serial and parallel ports are controlled by a software command word.

## III. MEMORY MAP OF THE NDT-COMP8 MICROCOMPUTER

The memory of the NDT-COMP8 computer consists of both ROM (read only memory) and RAM (read-write memory), with the address lines wired to give the address locations shown in Fig. 3. The addresses are given in hexadecimal (base 16) numbers.

The minimum amount of memory the system will operate on is shown shaded in Fig. 3, and it can be expanded in 1K (1024) steps until there are 4K (4096) bytes of ROM and the same amount of RAM. The ROM is Intel 8708 erasable, programmable, read-only memory, that is configured as 1024 X 8 bits. The RAM used is Advanced Micro Devices' AM-9130 memory that is configured 1024 X 4 bits. Therefore, two integrated circuits (ICs) are required to make 1K bytes of memory. The system monitor, which will be described in greater detail later, sets the stack pointer at 1FFB (hex). Because the monitor and any other programs written use the stack, it is suggested that user programs not be written into the upper 30 bytes of RAM.

ORNL-DWG 76-19088

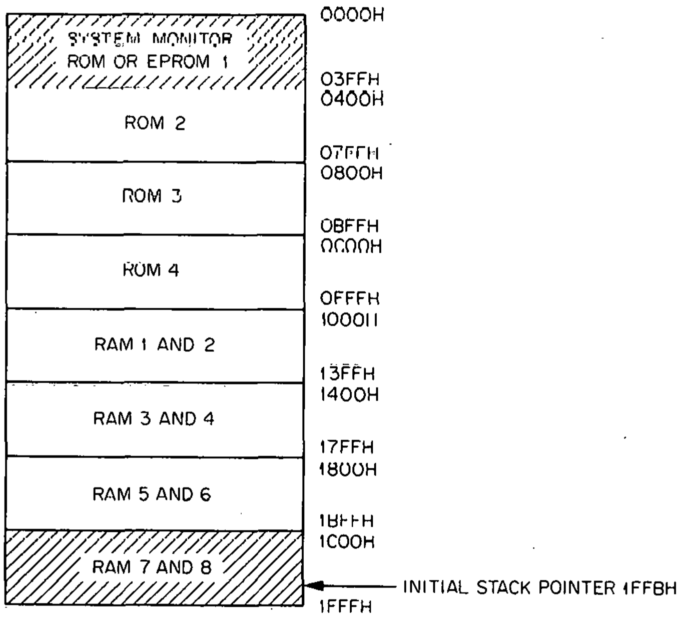

**Fig. 3. Memory map of the NDT-COMP8.**

The format of the memory addresses is shown in Fig. 4. The address lines $A_0$-$A_9$ are on the individual memory chips. The address lines $A_{10}$-$A_{12}$ are run to the 8205, which is a one-of-eight decoder. The eight outputs of the 8205 run to the chip select ($\overline{CS}$) lines of the 4 ROM chips and the 4 pairs of RAM chips as shown in Fig. 4.

## IV. INPUT/OUTPUT

The NDT-COMP8 has both serial and parallel input and output. The input/output chips are selected by the 8080 address lines, $A_0$-$A_7$, when the I/O commands are given to the central processing unit (CPU). Input and output are performed when $\overline{I/OW}$ or $\overline{I/OR}$ signals are present on the control bus. The I/O address format is shown in Fig. 5. The 8251 is programmed by two control words from the monitor that allow it to communicate with RS-232C compatible devices, 20-ma current-loop teletypes, or TTL serial interface devices, depending on the jumper connections made on the board. The 8255 chips can be programmed for a variety of I/O configurations, as described in the Intel 8080 *Microcomputer System User's Manual.*[1] In Table 1, we show the addresses for the various ports and the control word. These addresses follow an input (DB in machine language) or output (D3 in machine language) instruction in the computer program. The first three chips are the 8255s; the last is the 8251 serial port. For cases (such as $A_6$ and $A_7$) where the address lines are not physically connected, we make the bits high so that the address words will not have to be changed if additional chips are added.

## V. PROGRAMS FOR THE NDT-COMP8 MICROCOMPUTER

Several useful programs have been written for the NDT-COMP8 microcomputer. Instructions are given in this section concerning the use of the various programs and the results they produce. The machine

[1] Intel Corporation, *Intel 8080 Microcomputer System User's Manual*, Santa Clara, California (Sept. 1975).

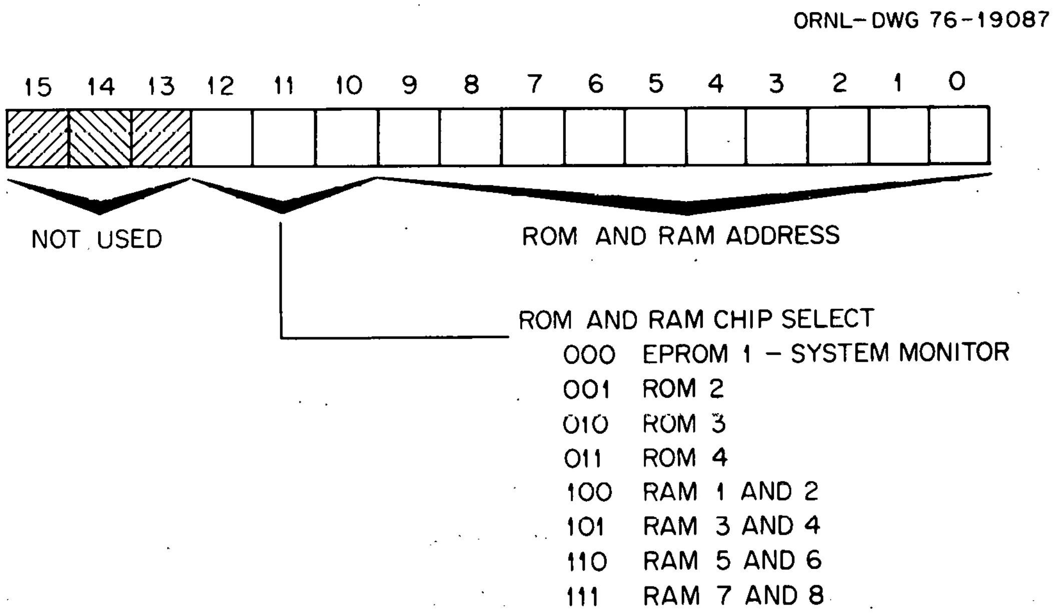

Fig. 4. Memory address format of the NDT-COMP8.

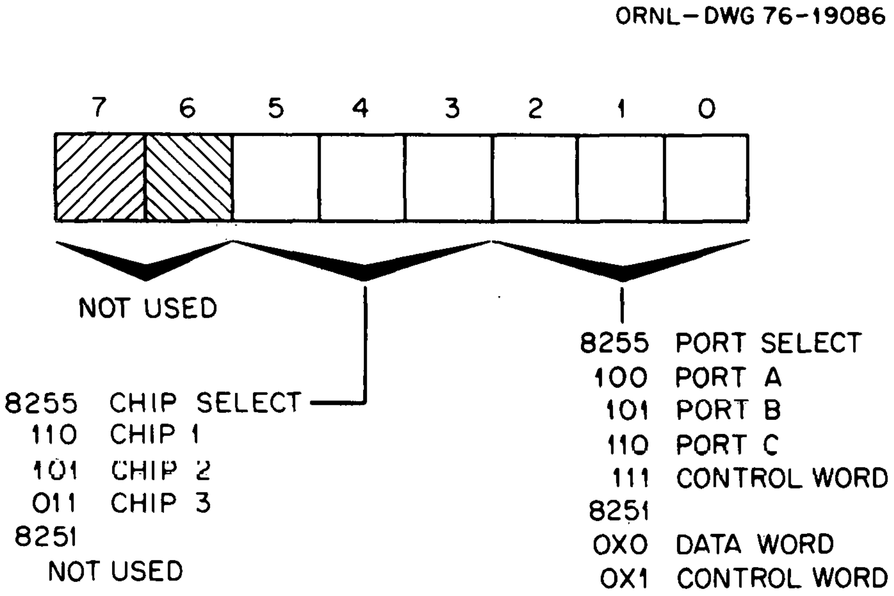

**Fig. 5. Input/output address format for the NDT-COMP8.**

**Table 1. Input/output addresses**

| Chip and plug no. | I | II | III | IV[a] |
|---|---|---|---|---|
| Control word | F7 | EF | DF | FB |
| A | F4 | EC | DC | FA |
| B | F5 | ED | DD | |
| C | F6 | EE | DE | |

[a] Serial connection.

language listings of the programs are also given so that these may be typed directly into (or programmed into) the listed memory locations without having to be assembled. Additional information on programming for the 8080 microcomputers is available from Intel.[2] While assemblers, software development systems, and high-level programming languages are desirable, they are not necessary to develop machine language programs for the NDT-COMP8 microcomputer.

[2] Intel Corporation, *Intel 8080 Assembly Language Programming Manual,* Santa Clara, California (1975).

## A. System Monitor

A monitor has been developed for the computer, and it occupies the 0000 to 03FFH locations in the memory. It aids the user in writing, modifying, running and debugging his programs from a terminal. The monitor is written as a series of useful subroutines. When the system power is turned on or reset, the monitor sets the 8251 port, saves the old values of the registers, resets the stack pointer, and then prints the following message on the terminal (if one is attached to the computer):

```
NDT-COMP8
```

The monitor gives a period when it is awaiting instructions from the terminal. The following commands may be executed:

1. Insert Command, I

   I <ADDRESS>

   This command is used to enter instructions and data into memory starting at "address." Binary instructions and data must be entered symbolized by their hex equivalent. Spaces and carriage returns are ignored. When an ESC is typed the routine will edit. An error input will result in an error "*" mark and termination of the routine.

   Example:

   ```
   .I1CØØ
   21111C4E233E22A9CA2BØØCDE3Ø1C3Ø3
   1C544849532Ø49532Ø414E2Ø4558414D
   5Ø4C4522$
   .
   ```

2. Display Command, D

   D <LOWER ADDRESS>, <UPPER ADDRESS>

   This command will display the contents of memory from "lower address" to "upper address" with a hex representation of the binary word. A header on each line will indicate the starting address of each line with an XXXX. ESC will terminate this command.

   Example:

   ```
   .D1CØØ,1C23
   1CØØ 21 11 1C 4E 23 3E 22 A9 CA 2B ØØ CD E3 Ø1 C3 Ø3
   1C1Ø 1C 54 48 49 53 2Ø 49 53 2Ø 41 4E 2Ø 45 58 41 4D
   1C2Ø 5Ø 4C 45 22
   ```

3. Run Command, G

   G [<ENTRY POINT>]

   This command will run a program beginning at "entry point." If no entry point is specified, program execution will begin at the memory location specified by the contents of the top of stack (top 2 words).

   Example:

   ```
   .G1CØØ
   THIS IS AN EXAMPLE.
   ```

4. Move Memory Command, M

M <LOW ADDRESS>, <UPPER ADDRESS>, <DESTINATION>

This command will copy the block of memory between the "low address" and the "upper address" onto the memory starting at the "destination" address.

Example:

```
.M1C16,1C17,1C19
.D1C16,1C1A
1C16 49 53 2Ø 49 53
.G1CØØ
THIS IS IS EXAMPLE.
```

(IS was moved to overwrite the RAM address of AN)

5. Substitute Command, S

S <ADDRESS> (SP)

This command is implemented differently from those already discussed. The command should be entered with the address of the memory location to be modified and then followed by a space. The present contents of that location will be displayed along with a '-' prompt. You must now enter your modified code. To change the next memory location you need only to "space" again. To terminate this routine, hit a carriage return. An ESC or illegal character will also terminate this routine without replacing the last set of data displayed. If no code is typed in after a space, the present content at that location will remain unchanged.

Example:

```
.S1C19 49-41 53-4E
.D1C19,1C1A
1C19 41 4E
.G1CØØ
THIS IS AN EXAMPLE.
```

6. Examine/Modify Register Command, X

X [<REGISTER IDENTIFIER>]

This command will display the contents of all the CPU registers if you just enter "X." By including a register identifier and by following the technique used with the substitute command (space rather than carriage return), you may alter the value of any register. A carriage return will terminate this routine. An ESC or any illegal character will also terminate this routine. The flags are denoted by *F*. *M* is a contraction of the HL register contents, *P* is the program counter, and *S* is the stack pointer.

Example: The registers are examined without changing contents.

```
.X
A=27 B=2E C=2E D=ØØ E=ØF F=46 H=1F L=ED M=1FED P=1CØØ S=1FEB
.
```

Example: Register A is examined and changed, and all registers are displayed again.

```
.XA 27-ØØ 2E-11
.X
A=ØØ B=11 C=2E D=ØØ E=ØF F=46 H=1F L=ED M=1FED P=1CØØ S=1FEB
.
```

An error made in any of the command routines will result in an error message – "*." Termination of this routine will also occur.

There are many useful routines in the monitor that the user can call. They are listed in Table 2.

**Table 2. NDT-COMP8 routines**

| Address | Function | Destroys regs. |
|---|---|---|
| 002B | Identifies and recognizes monitor commands from terminal (I, D, G, M, S, and X). Transfers control to proper location to execute commands. Reentrant point after your program has finished. | ABCHLF |
| 005E | Executes D (display) command. | ABCDEHLF |
| 0095 | Executes G (go to) command. | ABCDEHLF |
| 00B3 | Executes I (insert) command. | ABCDEHLF |
| 00FD | Executes M (move) command. | ABCDEHLF |
| 011D | Executes S (substitute) command. | ABCDEHLF |
| 0141 | Executes X (examine regs.) command. | ABCDEHLF |
| 01BD | Looks for an escape from terminal carry bit set and returns if there is an escape. Carries bit not set and returns otherwise. | AF |
| 01D0 | Inputs a character into A from the terminal. | AF |
| 01DA | Converts ASC11 character in register C into its binary value and returns it in A. ASCII character 0-9 or A-F. Hex ch. 0-F. | AF |
| 01E3 | Prints a single ASCII character in reg. C on the terminal. | AF |
| 01EE | Outputs a carriage return and line feed to the terminal. | ABCF |
| 01F4 | Takes an ASCII character in register C and prints it on the terminal. An ESC is echoed as a carriage return, line feed. | ABF |
| 020D | Prints an asterisk (*), carriage return and line feed on terminal. Control is then returned to 002B. | ABCF |
| 021B | Sets carry equal to zero and returns. | F |
| 021B | Inputs a character from the terminal, clears parity bit, if any, and returns with ASCII character in C. | ACF |
| 0222 | Accepts a string of characters from 021B above in C register, echoes the character on the terminal, converts to binary and stores last 4 characters in BC registers. Returns when a delimiter (comma, space or carriage return) is encountered with last 4 characters in BC. | ABCDEF |
| 0257 | Will input the number of numbers in the C register between 1 and 3 from the terminal and store them on the stack in reverse order.<br>The first number must be less than or equal to the second, or the second will be set equal to the first. The first number popped from the stack will be the last number put in from the terminal after the program has returned. | ABCDEHLF |

Table 2. NDT-COMP8 routines (continued)

| Address | Function | Destroys regs. |
|---|---|---|
| 029C | Compares DE registers to HL. Carry = 1 if DE ≤ HL. Carry = 0 if DE > HL. | F |
| 02C3 | Takes a hex pair of numbers in A, converts to 2 ASCII characters and prints them on the terminal. | ABCF |
| 02DE | Converts a 0 to F hex number in C to an ASCII character and returns it in the C register. | BCHLF |
| 02E6 | Displays the previous register contents from memory in formatted form, as in the X command. | ABCDEHLF |
| 0317 | Takes a single character in C denoting a register and returns the address of the register contents in memory in BC. Invalid input in C will call the error program. | ABCDEHLF |
| 032E | Restores all registers and flags to original monitor values then executes the G command. | ABCDEHLF |
| 0343 | Sets carry flag equal to 1. | I |
| 0345 | Checks the byte stored at 1FF9 in RAM. If it is set to lower, stores 0 in the 4 L.S. bits of the byte addressed by DE. | ABCHLF |
| 0350 | Puts 4 bit value in C into byte addressed by DE. The lower or upper position is specified by 1FF9 in RAM. | ABCHLF |
| 036F | Takes an ASCII character in register C and checks to see if it represents a valid hex digit. Carry = 1 if valid, 0 if not. | AF |
| 038A | Checks an ASCII character in C to see if it is a valid delimiter (comma, carriage return, or space). Returns with carry = 1 if valid, 0 if not. | AF |

The monitor routines can be called by subroutine calls, except for the error routine (020D) and the main monitor-control program (002B). The other routines will return to the program that called them. When a RESET is issued, the computer will retain the old values in the register memory, reset the registers, stack pointer and serial I/O port, print the sign-on message, and enter the main monitor-control program at 002B. An interrupt (INT) signal will transfer program control to 0400, the first ROM address after the monitor. This allows the user to write his programs, store them on this ROM, and execute them using only a push-button switch (no terminal is required for execution). A machine language listing of the monitor follows:

```
0000 3E CF D3 FB 3E 27 D3 FB 22 F3 1F E1 22 F5 1F 21
0010 00 00 39 22 F7 1F 21 F3 1F F9 F5 C5 D5 21 9D 03
0020 06 0F 4E CD E3 01 23 05 C2 22 00 21 ED 1F F9 0E
0030 2F CD F4 01 FB C3 3B 00 C3 00 04 CD 1B 02 CD F4
0040 01 79 01 06 00 21 B9 03 BE CA 54 00 23 0D C2 48
0050 00 C3 0D 02 21 AB 03 09 09 7E 23 66 6F E9 0F 02
0060 CD 57 02 D1 E1 CD EE 01 7C CD C3 02 7D CD C3 02
0070 0E 20 CD F4 01 7E CD C3 02 CD BD 01 DA 85 00 CD
0080 9C 02 D2 BB 00 CD EE 01 C3 2B 00 23 7D E6 0F C2
0090 70 00 C3 65 00 CD 22 02 D2 AA 00 7A FE 0D C2 0D
00A0 02 21 F5 1F 71 23 70 C3 B0 00 7A FE 0D C2 0D 02
00B0 C3 2F 03 0E 01 CD 57 02 3E FF 32 F9 1F D1 CD 1B
00C0 02 4F CD F4 01 79 FE 1B CA F4 00 CD 8A 03 DA BE
```

```
00D0 00 CD 6F 03 D2 EF 00 CD DA 01 4F CD 50 03 3A F9
00E0 1F B7 C2 E6 00 13 EE FF 32 F9 1F C3 BE 00 CD 45
00F0 03 C3 0D 02 CD 45 03 CD FE 01 C3 2B 00 0F 03 CD
0100 57 02 C1 E1 D1 E5 62 6B 7E 60 69 77 03 78 B1 CA
0110 2B 00 13 F1 CD 9C 02 D2 2B 00 C3 05 01 CD 22 02
0120 C5 E1 7A FE 20 CA 2D 01 FE 2C C2 2B 00 7E CD C3
0130 02 0E 2D CD F4 01 CD 22 02 D2 3D 01 71 23 C3 22
0140 01 CD 1B 02 4F CD F4 01 79 FE 0D C2 54 01 CD E6
0150 02 C3 2B 00 4F CD 17 03 C5 E1 0E 20 CD F4 01 79
0160 32 F9 1F 3A F9 1F FE 20 CA 70 01 FE 2C C2 2B 00
0170 7E B7 C2 7B 01 CD EE 01 C3 2B 00 E5 5E 16 1F 23
0180 46 D5 D5 E1 C5 7E CD C3 02 F1 F5 B7 CA 94 01 2B
0190 7E CD C3 02 0E 2D CD F4 01 CD 22 02 D2 B4 01 7A
01A0 32 F9 1F F1 E1 B7 CA AB 01 70 2B 71 11 03 00 E1
01B0 19 C3 63 01 7A 32 F9 1F D1 D1 C3 AC 01 DB FB E6
01C0 02 CA 18 02 DB FA E6 7F FE 1B CA 43 03 C3 18 02
01D0 DB FB E6 02 CA D0 01 DB FA C9 79 D6 30 FE 0A F8
01E0 D6 07 C9 DB FB F6 01 CA E3 01 79 D3 FA C9 0F 0D
01F0 CD F4 01 C9 41 3E 1B B8 C2 FD 01 0E 24 CD E3 01
0200 3E 0D B8 C2 0B 02 CD F2 03 06 0D 48 C9 0F 2A CD
0210 F4 01 CD EE 01 C3 2B 00 37 3F C9 CD D0 01 E6 7F
0220 4F C9 E5 21 00 00 1F 00 CD 1B 02 4F CD F4 01 CD
0230 8A 03 D2 41 02 51 E5 C1 E1 7B B7 C2 43 03 CA 18
0240 02 CD 6F 03 D2 0D 02 CD DA 01 1E FF 29 29 29 29
0250 06 00 4F 09 C3 2B 02 2E 03 79 E6 03 C8 67 CD 22
0260 02 D2 0D 02 C5 2D 25 CA 73 02 7A FE 0D CA 0D 02
0270 C3 5E 02 7A FE 0D C2 0D 02 01 FF FF 7D B7 CA 86
0280 02 C5 2D C2 81 02 C1 D1 F1 CD 9C 02 D2 91 02 54
0290 5D E3 D5 C5 E5 3D F8 E1 E3 C3 95 02 C5 47 E5 7A
02A0 B3 CA BD 02 23 7C B5 CA BD 02 F1 D5 3E FF AA 57
02B0 3E FF AB 5F 13 7D 83 7C 8A D1 7B C1 C9 E1 78 C1
02C0 C3 43 03 E5 F5 0F 0F 0F 0F E6 0F 4F CD DE 02 CD
02D0 F4 01 F1 E6 0F 4F CD DE 02 CD F4 01 E1 C9 21 BF
02F0 03 06 00 09 4F C9 21 CF 03 4F 79 B7 C2 F3 02 CD
02F0 EE 01 C9 CD F4 01 0E 3D CD F4 01 23 5E 16 1F 23
0300 1A CD C3 02 7E B7 CA 0E 03 1B 1A CD C3 02 0F 20
0310 CD F4 01 23 C3 E9 02 21 CF 03 11 03 00 7E B7 CA
0320 0D 02 B9 CA 2A 03 19 C3 1D 03 23 44 4D C9 F3 21
0330 ED 1F F9 D1 C1 F1 2A F7 1F F9 2A F5 1F E5 2A F3
0340 1F FB C9 37 C9 3A F9 1F B7 C0 0E 00 CD 50 03 C9
0350 D5 E1 79 F6 0F 4F 3A F9 1F B7 C2 63 03 7E E6 F0
0360 B1 77 C9 7E E6 0F 47 79 0F 0F 0F 0F B0 77 C9 79
0370 FE 30 FA 18 02 FE 39 FA 43 03 CA 43 03 FE 41 FA
0380 18 02 FE 47 F2 18 02 C3 43 03 79 FE 2C CA 43 03
0390 FE 0D CA 43 03 FE 20 CA 43 03 C3 18 02 0D 0A 4E
03A0 44 54 2D 43 4F 4D 50 38 00 0D 0A 00 00 41 01 1D
03B0 01 FD 00 B3 00 95 00 5E 00 44 47 49 4D 53 58 30
03C0 31 32 33 34 35 36 37 38 39 41 42 43 44 45 46 41
03D0 F2 00 42 F0 00 43 EF 00 44 EE 00 45 ED 00 46 F1
03E0 00 48 F4 00 4C F3 00 4D F4 01 50 F6 01 53 F8 01
03F0 00 00 06 05 0F 0A CD E3 01 05 C8 0E 00 C3 F6 03
```

## B. Floating-Point Math Package

A floating-point math package has been developed for the NDT-COMP8 from the Intel User's Library of 8080 programs. The math package has signed Add, Subtract, Multiply, Divide, Float, Fix, Binary-to-Decimal and Natural Log routines in it. The numbers are usually stored and returned in the registers in forms of

| | | |
|---|---|---|
| C (exponent) | D (upper mantissa) | E (lower mantissa) |
| B (exponent) | H (upper mantissa) | L (lower mantissa) |

| | | | |
|---|---|---|---|
| X | Y ZZZZZZ | · | BBBB BBBB BBBB BBBB |
| sign of no. | sign of expt | | always = 1 (except when number is 0) |
| 0 = + | 1 − + | | |

The number 0 is 1100 0000 . 0000 0000 0000 0000

or in the hex notation C 0 . 0000

The exponent is represented in binary offset, with $2^{-64}$ as 000 0000, $2^{-1}$ as 011 1111, $2^{0}$ as 100 0000 and $2^{63}$ as 111 1111. The routines are listed in Table 3. The execution times for the various operations are typical, and RAM locations from 1C10 to 1C19 are used. The accuracy of all operations is about 15 ppm.

Table 3. Floating point math package

| Address | Function | Destroys regs |
|---|---|---|
| 0810 | Subtracts floating number in CDE from floating number in BHL and returns floating answer in CDE. 0.6 msec time. | ABCDEHLF |
| 0813 | Adds floating number in CDE to floating number in BHL and returns floating answer in CDE. 0.5 msec time. | ABCDEHLF |
| 08B3 | A positive floating number in CDE is converted into an unsigned fixed number between 1 and 65,535 in DE. If CDE is less than one, zero is returned. If a fix is attempted on a negative number, 40 is returned in A. If a fix is attempted on a number >65,536, 01 is returned in A. A is zeroed for a successful fix, 0.6 msec time. | ACDEHLF |
| 08E0 | Floats a fixed number in DE into CDE as a positive number, 0.06 msec time. | ACDEF |
| 08F8 | Changes the sign of (negates) a signed floating-point number in CDE to CDE (unless the number is zero), 0.04 msec time. | ACF |
| 0902 | Divides a floating number in CDE into a floating number in BHL and returns the result in CDE. Returns a 1 in A register if division by 0 is attempted, 0 otherwise, 2.2 msec time. | ABCDEHLF |
| 097B | Multiplies a floating number in CDE by a floating number in BHL and returns the result in CDE. Exponent overflow not checked, 1.9 msec time. | ABCDEHLF |
| 0A33 | Fixed binary number in DE is converted to BCD number in CDE. | ABCDEHLF |
| 0A81 | Floating binary number in CDE is converted to floating BCD number between 6,555 and 65,535 with sign bit and exponent (base 10) in 20 complement in 1C19 in RAM, 7.2 msec time. | ABCDEHLF |
| 0ABD | Takes natural log of positive floating number in CDE and returns answer in CDE. Negative input numbers give wrong answers, 5.4 msec time. | ABCDEHLF |

A machine language listing of the floating-point math package follows.

```
0810 CD F8 08 CD 22 0A 87 FB 79 48 47 C8 CD 22 0A 87
0820 CA 17 08 F5 21 11 1C 79 77 E6 7F 4F 2B 78 77 E6
0830 7F 47 B9 CA 62 08 DA 46 08 41 4F C5 46 23 4E 70
0840 2B 71 C1 E1 EB E5 E1 78 3C 47 37 3F 7C 1F 67 7D
0850 1F 6F AF 17 32 12 1C 78 B9 C2 48 08 3A 12 1C 85
0860 6F E5 E1 3A 11 1C E6 80 4F 3A 10 1C E6 80 81 F2
0870 9F 08 7A BC CA 94 08 DA 7F 08 EB 79 C3 82 08 79
0880 C6 80 80 4F 37 3F 7B 2F 5F 7A 2F 57 13 19 EB 06
0890 10 C3 E2 08 7B BD CA AD 08 DA 7F 08 C3 7A 08 79
08A0 80 4F 19 EB D0 0C 7A 1F 57 7B 1F 5F C9 0E 40 11
08B0 00 00 C9 AF 67 6F 81 FA D4 08 C6 40 F2 D7 08 F6
08C0 3F FE 11 D2 DC 08 4F CD 12 0A CD 1B 0A 0D C2 C7
08D0 08 AF EB C9 3E 40 C9 11 00 00 78 C9 3E 01 C9 00
08E0 0E 50 CD 26 0A 87 CA F5 08 AF B2 3E 00 F8 CD 12
08F0 0A 0D C3 E9 08 0E 40 C9 CD 22 0A 87 C8 3E 80 81
0900 4F C9 CD 22 0A 87 3E 01 C8 E5 21 13 1C AF 77 23
0910 77 23 36 11 23 36 01 23 78 91 C6 41 E6 7F 77 79
0920 E6 80 00 4F 78 E6 80 81 86 77 AF 47 4F E1 B9 37
0930 C2 34 09 3F 3F E5 21 13 1C 7E 17 77 23 7E 17 77
0940 23 35 CA 6B 09 23 35 E1 CA 4E 09 29 17 47 3A 13
0950 1C E6 01 87 CA 62 09 7D 93 6F 7C 9A 67 78 99 C3
0960 65 09 19 78 89 E6 01 47 C3 2E 09 E1 21 17 1C 4E
0970 2B 2B 2B 56 2B 5E 06 10 C3 E2 08 78 32 15 1C 7C
0980 32 16 1C 7D 32 17 1C 21 10 1C AF 77 23 77 23 77
0990 23 77 23 77 23 7E E6 80 47 7F 81 C6 40 E6 7F 77
09A0 79 E6 80 80 86 77 23 23 46 0E 08 21 00 00 37 3F
09B0 C3 B9 09 CD 12 0A CD 1B 0A 78 1F 47 D2 D3 09 C5
09C0 01 10 1C 0A 83 02 03 0A 8A 02 03 0A 8D 02 03 0A
09D0 8C 02 C1 0D C2 B3 09 01 14 1C 0A 3D CA EB 09 3C
09E0 3C 02 03 03 0A 47 0E 08 C3 83 09 21 15 1C 7E 2B
09F0 2B 46 2B 4E 2B 56 2B 5E 60 69 4F 06 10 AF 84 FA
0A00 10 0A CD 12 0A CD 1B 0A 0D 05 C2 FD 09 C3 AD 08
0A10 EB C9 37 3F 7B 17 5F 7A 17 57 C9 7D 17 6F 7C 17
0A20 67 C9 3E 40 B9 C0 7B B2 DA 30 0A C2 30 0A AF C9
0A30 3E 40 C9 21 10 1C 3E 01 77 3D 23 77 23 77 21 13
0A40 1C 77 23 77 23 77 AF 7A 1F 57 7B 1F 5F D5 DC 6D
0A50 0A CD 67 0A D1 AF BA C2 46 0A BB C2 46 0A 21 13
0A60 1C 5E 23 56 23 4E C9 01 10 1C C3 70 0A 01 13 1C
0A70 21 10 1C 16 03 AF 0A 8E 27 02 15 C8 03 23 C3 76
0A80 0A 21 19 1C 79 E6 80 C6 20 77 79 E6 7F 4F 79 FE
0A90 4E D2 A2 0A E5 21 00 A0 06 44 CD 7B 09 F1 35 C3
0AA0 8E 0A 79 FE 51 DA B6 0A E5 21 CC CC 06 3D CD 7B
0AB0 09 E1 34 C3 A2 0A CD B3 08 CD 33 0A C9 C5 EB 22
0AC0 12 1C 21 00 00 22 14 1C 4C 0C 79 FE 09 CA FA 0A
0AD0 2A 12 1C 41 54 5D AF 7C 1F 67 7D 1F 6F 05 C2 D6
0AE0 0A 19 DA C9 0A 22 12 1C 21 2C 0B 09 09 5E 23 56
0AF0 2A 14 1C 19 22 14 1C C3 C9 0A 2A 12 1C 7D 2F 5F
0B00 7C 2F 57 2A 14 1C 19 23 0E C0 EB CD E2 08 E1 D5
0B10 C5 5D AF 57 6F CD E0 08 06 C7 26 80 CD 13 08 21
0B20 72 B1 06 40 CD 7B 09 E1 45 E1 CD 13 08 C9 CC 67
0B30 20 39 27 1E 85 0F E1 07 F8 03 FE 01 FE 00
```

### C. Prom Programmer Program

A program has been written to program the Intel 8708 PROMs, using the electrical circuit board described in Sect. VI. The plug on the PROM circuit board must be inserted into J2 on the NDT-COMP8 board, the jumper connected to the +30 volt supply, and the PROM to be programmed inserted into the PROM circuit board. The reset line should be held low while these connections are being made. An NDT-COMP8 monitor is also necessary for the program to run. The program is usually stored in PROM and must be moved to location 1000 in RAM, using the Monitor command: M XXXX, XXXX + 6F, 1000. The program is now in RAM. To execute the program, type G1000. The monitor will wait for the lower address and the upper address + 1 of the instructions which are to be programmed to be typed on the terminal. To program 0000 to 03FF we would type:

```
.G1000
.0000  0400
```

The monitor responds with a period when the PROM has been successfully programmed and tested. (A 50% overprogramming is performed.) This will take from one to six minutes, depending on what is to be programmed. The data to be programmed must have the same address as the nine lower address lines of the PROM being programmed. For example, data at 0, 400, 800, C00, 1000, 1400, 1800 and 1C00 will all be copied into the first ROM location (000). A machine language copy of the program follows:

```
1000 0E 02 CD 57 02 D1 C1 3E 05 F5 F1 C6 01 F5 3E 80
1010 D3 EF C5 E1 7D D3 FC 7E D3 ED 3E 03 A4 C6 04 D3
1020 EF C6 10 23 00 D3 EF 3E 00 E3 E3 E3 E3 E3 E3 E3
1030 E3 3C C2 29 10 3E 00 D3 EF 7A 94 C2 14 10 7B 95
1040 C2 14 10 3E 82 D3 EF C5 E1 7D D3 FC 3E 03 A4 D3
1050 EF DB ED AF C2 0A 10 23 7A 94 C2 49 10 7B 95 C2
1060 49 10 F1 D6 01 CA 2B 00 D6 01 C2 0D 10 C3 2B 00
```

## VI. CONSTRUCTION DETAILS FOR THE NDT-COMP8 MICROCOMPUTER

### A. Main Board Construction

The NDT-COMP8 microcomputer has been laid out on a double-sided printed circuit board. The location of the parts is shown in Fig. 6. It is recommended to use sockets for all of the transistors and integrated circuits. The standard precautions must be observed when inserting the MOS circuits. The location of the various jumper connections is also shown. The PROM and RAM memory locations are also given. The 000-3FF PROM and the two 1C00-1CFF RAM memory circuits must be plugged in for normal operation; the rest of the memory is optional. The three 74LS161 circuits and the 8251 must be plugged in to use a terminal, and the three 8255 circuits are used for parallel I/O. The printed circuit positives are shown in Fig. 7 and Fig. 8, and the wiring diagram is shown in Figs. 9 and 10. The parts list is given in Table 4. Ten of these computers were constructed at one time. The construction time averaged 4 hr per computer, and the parts cost ranged from $275 for the minimum amount of memory to $700 for the maximum amount of memory.

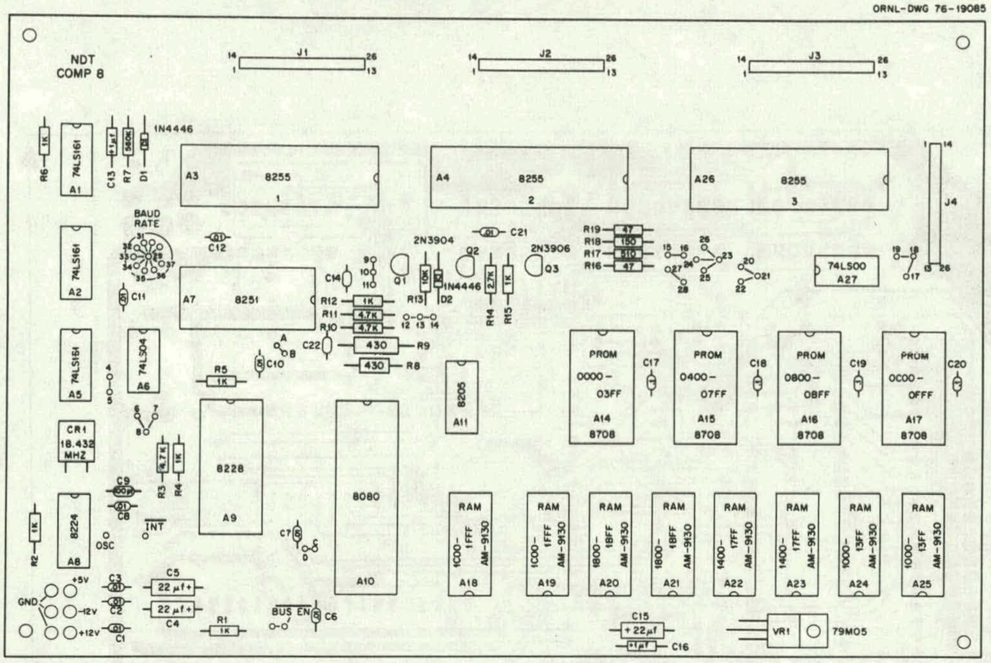

**Fig. 6. Parts layout for NDT-COMP8 microcomputer board.**

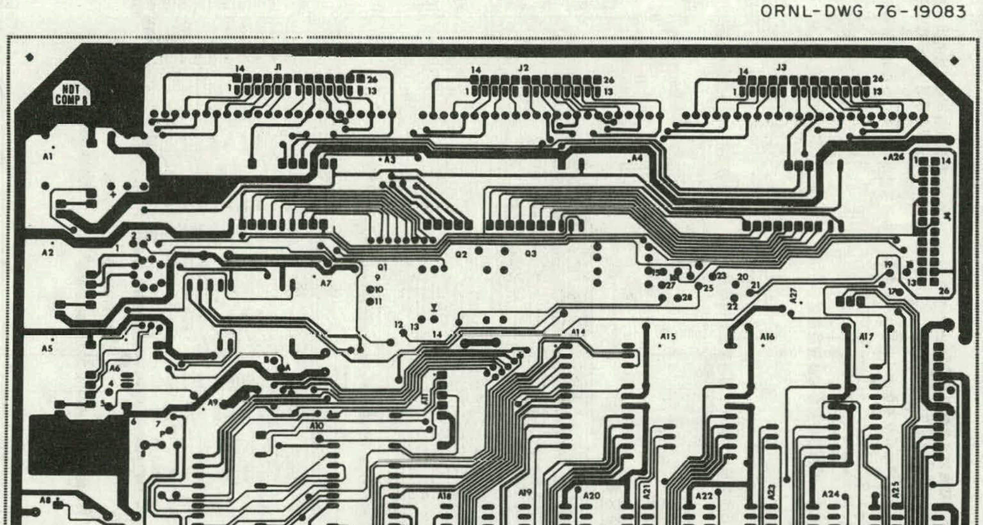

**Fig. 7. NDT-COMP8 printed-circuit board layout (component side).**

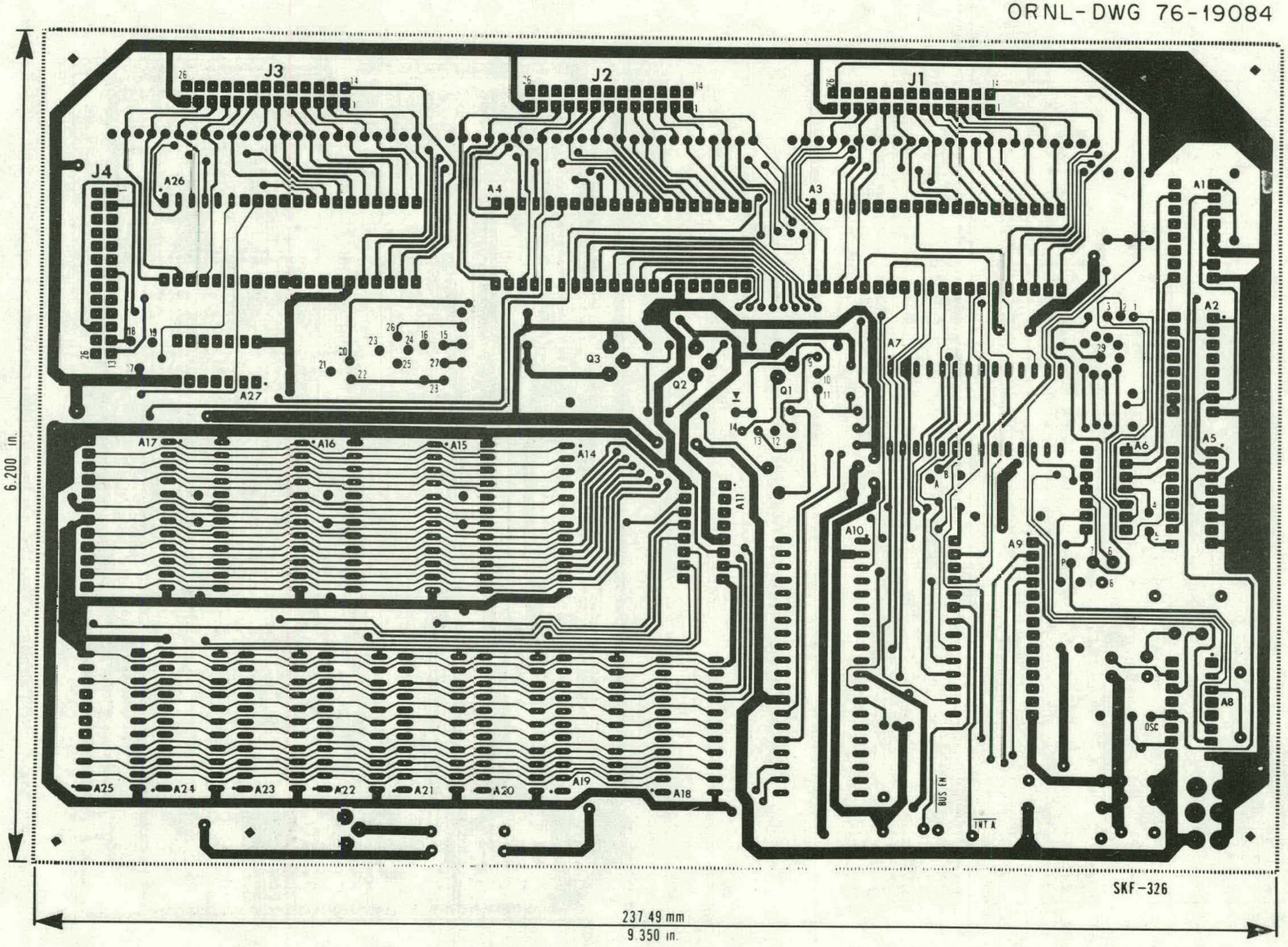

Fig. 8. NDT-COMP8 printed circuit board layout, (noncomponent side).

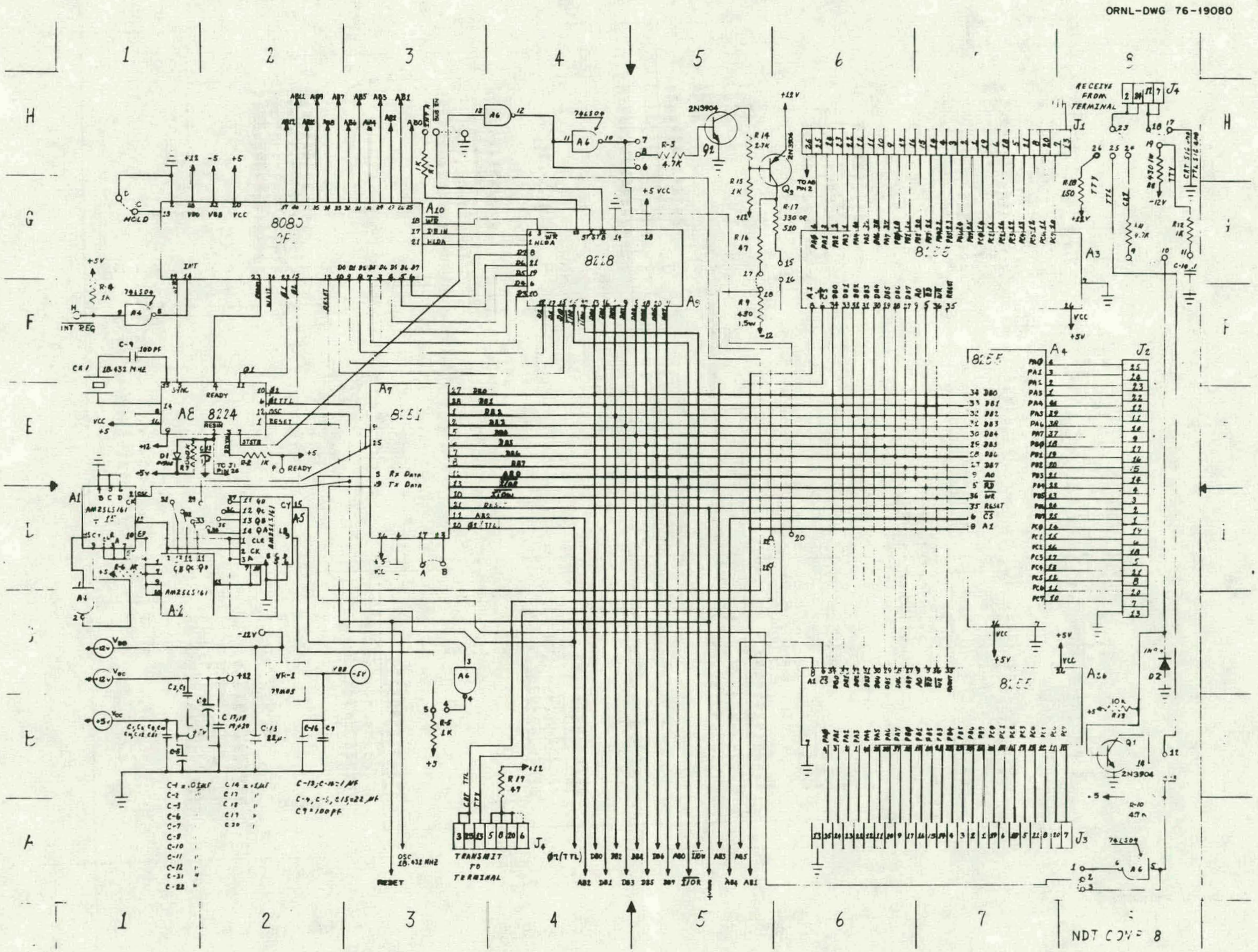

Fig. 9. Circuit diagram for the NDT-COMP8 microcomputer, I-10 and clock section.

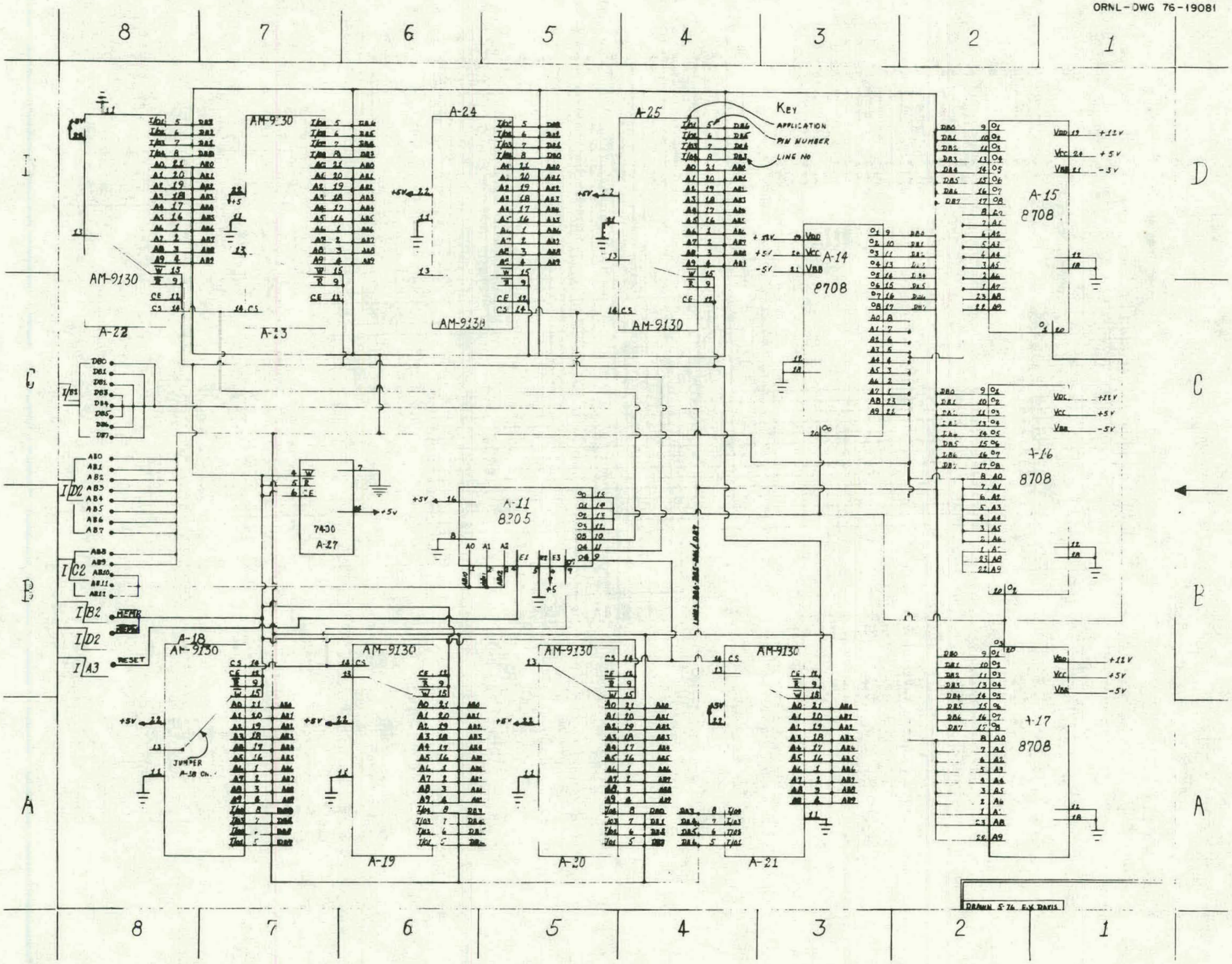

Fig. 10. Circuit diagram for the NDT-COMP8 microcomputer, memory section.

Table 4. Parts list for NDT-COMP8 microcomputer

| Component no. | Description |
|---|---|
| A1, A2, A5 | 75LS161 Dividers |
| A3, A4, A26 | 8255 Programmable peripheral interface |
| A6 | 74LS04 Hex inverter |
| A7 | 8251 Programmable USART |
| A8 | 8224 Clock generator |
| A9 | 8228 System controller |
| A10 | 8080A or AMD9080A |
| A11 | 8205 or AM25LS138 |
| A14, A15, A16, A17 | 8708 (A14 is preprogrammed with monitor) |
| A18, A19, A20, A21 | |
| A22, A23, A24, A25 | AM-9130 |
| A27 | 74LS00 |
| R1, R2, R4, R5, R6 | |
| R12,R15 | 1 kΩ 1/4 W 5% or better |
| R3, R10, R11 | 4.7 kΩ 5% or better |
| R7 | 560 kΩ 5% or better |
| R8 | 430 Ω 1 W |
| R9 | 430 Ω 1.5 W |
| R13 | 10 kΩ 5% or better |
| R14 | 2.7 kΩ 1/4 W 5% or better |
| R16, R19 | 47 Ω 5% or better |
| R17 | 510 Ω 5% or better |
| R18 | 150 Ω 5% or better |
| C1, C2, C3, C6, C7 | |
| C8, C10, C11, C12, C21, C22 | 0.01 μf 15V |
| C4, C5, C15 | 22 μf 15V |
| C9 | 100 pf >15V |
| C13, C16 | 1 μf 15V |
| C14, C17, C18, C19, C20 | 0.1 μf >15V |
| D1, D2 | IN4446 Signal diode |
| VR1 | 79M05 −5V Voltage regulator |
| Q1, Q2 | 2N3904 NPN transistor |
| Q3 | 2N3906 PNP transistor |
| J1, J2, J3, J4 | 3M3429 3M 26 Pin male connector |
| CR1 | I8801 Intersil 18.432 MHZ crystal |
| PC | NDT-COMP8-Printed circuit board |
| Baud rate switch | (Optional) Spectrol 87-12-19 |
| Sockets | (Optional) Robinson Nugent, Inc., part no. |
| 3 ea | 3 Pin, 18-32 |
| 2 ea | 14 Pin, ICN-143-S2 |
| 5 ea | 16 Pin, ICN-163-S2 |
| 8 ea | 22 Pin, ICN-163-S2 |
| 4 ea | 24 Pin, ICN-246-S4 |
| 2 ea | 28 Pin, ICN-286-S4 |
| 4 ea | 40 Pin, ICN-406-S4 |

## B. Optional Jumper Connections

There are several jumper options for the computer board, depending on the type of terminal and computer operation desired (Table 5).

**Table 5. Jumper connections for different terminals**

| Standard RS-232-C | Jumper connection for | |
|---|---|---|
| | 20 mil A Current loop | TTL |
| 23 to 24 | 23 to 26 | 23 to 25 |
| 17 to 18 | 18 to 19 | 17 to 18 |
| 9 to 10 | 10 to 11 | 12 to 13 |
| 13 to 14 | 13 to 14 | 2 to 3 |
| 2 to 3 | 1 to 2 | 20 to 21 |
| 6 to 8 | 7 to 8 | |
| 27 to 28 | 15 to 16 | |
| 21 to 22 | 21 to 22 | |
| 5 to 6 on J4 | | |

If the board is to be interfaced directly to a terminal device, the user should jumper pad A to B. If a modem will be used, leave pad A and B open.

To select the baud rate for the microcomputer, jumper:

PAD 29 to –31 For 4800 BAUD
–32 For 2400 BAUD
–33 For 1200 BAUD
–34 For 600 BAUD
–35 For 300 BAUD
–36 For 150 BAUD
–37 For 75 BAUD
For 110 BAUD, jumper 29 to 37 and 4 to 5

The optional Spectrol rotary switch will replace all the connections to pin 29.

J4 has the pin assignment shown in Table 6. Note that only one type of interface at a time is available at this plug, as specified by the selection of interface.

Pad C is the HOLD line on the 8080A on which a logic "1" is used to halt the 8080A and put the data and address busses in a high impedence mode, thereby allowing Direct Memory Access (DMA) to occur. For normal operation of the system, jumper pad C to D (GROUND). This will allow the 8080A to maintain control of the busses and operate freely.

The pad $\overline{\text{BUS EN}}$ is used to enable the 8228 data drivers. A logic "1" will put these drivers in their high-impedance state, freeing the data bus for a direct memory access (DMA) operation. (Note that both HOLD-PAD C and $\overline{\text{BUS EN}}$ must be high for implementation of DMA on this system.) Again, for normal system operation, jumper pad $\overline{\text{BUS EN}}$ to GND (adjacent hole). A system reset switch may be installed between pin 26 of J1 and ground. The SPST-NO (normally open) switch may be used to reset the entire NDT-COMP8 system. It will also restart the 8080 execution at address location 0000H.

Table 6. Pin assignments for J4 plug

| J4 | Assignment as a function of mode | | |
|---|---|---|---|
| Pin number | RS-232/C | 20 mil A loop | TTL |
| 1 | NC* | NC* | NC* |
| 2 | Receive data | NC | Receive data |
| 3 | Transmit data | NC | Transmit data |
| 4 | NC | NC | NC |
| 5 | +12VDC | +12VDC | +12VDC |
| 6 | +12VDC | NC | NC |
| 7 | Signal ground | NC | Signal ground |
| 8 | +12VDC | +12VDC | +12VDC |
| 9 to 11 | NC | NC | NC |
| 12 | NC | Receive–return data | NC |
| 13 | NC | Transmit data | NC |
| 14 to 19 | NC | NC | NC |
| 20 | +12VDC | +12VDC | +12VDC |
| 21 to 23 | NC | NC | NC |
| 24 | NC | Receive data | NC |
| 25 | NC | Transmit data return | NC |
| 26 | NC | NC | NC |

*NC signifies NO CONNECTION.

A similar switch can be wired between the $\overline{\text{INT}}$ pad and ground. When this pad is grounded, the 8228 issues a RST7 instruction on the data bus. The 8080 performs this instruction, which transfers execution to address 38 (hex) in the monitor, which contains a jump to 400 (hex).

The parallel I/O ports have the following pin connections, as shown in Table 7.

The first number designates the port number (1, 2, or 3); the letter denotes the port (A, B, or C) and the following number denotes the data bit (0 to 7). The pin assignments given in Tables 6 and 7 actually refer to the holes photo-etched in the board. If the male connector (3M3429) is soldered on the component side of the board, the pin numbers on the connector will agree with the pin numbers on the board. However, if the male connector is put on the noncomponent side of the board (as may be done to shorten the ribbon length), the board pin numbers will not agree with the connector pin numbers. In this case, the board numbers should be used, and the connector numbers ignored.

## VII. PROM SIMULATOR AND PROGRAMMER BOARD

An extra, piggyback board has been designed to plug into any of the four ROM sockets on the NDT-COMP8 computer board. The board has a clip that fits onto the 74LS00 (A27) and contains 1K of RAM. The RAM can be written into at the address of the ROM socket into which it is plugged when the WRITE switch is on, and can be used as read-only memory when the WRITE switch is off. (This prevents inadvertent overwriting of the ROM program you are simulating.) When the $\overline{\text{RESET}}$ is pressed and the clip is on A27, the board can be moved from one ROM location to another without losing its contents.

When the power clip is connected to a +30V, 20 ma supply, and the ribbon cable is connected to port J2, an 8708 PROM can be programmed using the program in Section V. A drawing of the circuit diagram of

Table 7. Pin assignments for J1, J2, and J3 plugs

| J Pin no. | Port I | Port II | Port III |
|---|---|---|---|
| | (J1) | (J2) | J(3) |
| 1 | 1B7 | 2B7 | 3B7 |
| 2 | 1B6 | 2B6 | 3B6 |
| 3 | 1B5 | 2B5 | 3B5 |
| 4 | 1B4 | 2B4 | 3B4 |
| 5 | 1C3 | 2C3 | 3C3 |
| 6 | 1C1 | 2C1 | 3C1 |
| 7 | 1C7 | 2C7 | 3C7 |
| 8 | 1C5 | 2C5 | 3C5 |
| 9 | 1A7 | 2A7 | 3A7 |
| 10 | 1A6 | 2A6 | 3A6 |
| 11 | 1A5 | 2A5 | 3A5 |
| 12 | 1A4 | 2A4 | 3A4 |
| 13 | GND | GND | GND |
| 14 | 1B3 | 2B3 | 3B3 |
| 15 | 1B2 | 2B2 | 3B2 |
| 16 | 1B1 | 2B1 | 3B1 |
| 17 | 1B0 | 2B0 | 3B0 |
| 18 | 1C2 | 2C2 | 3C2 |
| 19 | 1C0 | 2C0 | 3C0 |
| 20 | 1C6 | 2C6 | 3C6 |
| 21 | 1C4 | 2C4 | 3C4 |
| 22 | 1A3 | 2A3 | 3A3 |
| 23 | 1A2 | 2A2 | 3A2 |
| 24 | 1A1 | 2A1 | 3A1 |
| 25 | 1A0 | 2A0 | 3A0 |
| 26 | RESET | NC | NC |

the board is shown in Fig. 11, and a mechanical drawing of the board is shown in Fig. 12. The parts list for the board is given in Table 8.

## VIII. POWER REQUIREMENTS FOR THE NDT-COMP8 MICROCOMPUTER

The power requirements for the computer vary considerably depending upon the memory and I/O requirements. These requirements are shown in Table 9 below.

The RAM uses about 74 ma at +5 volts per 1K of memory; the 8255 parallel-output ports use about 50 ma at +5 volts each; the 8251 serial port uses about 45 ma at +5 volts (although the ±12 volt drain increases by about 50 ma when serial I/O is being performed). The 8708, 1K PROM chips each use about 10 ma at +5 volts, 36 ma at +12 volts, and 22 ma at −12 volts. A +30 volt, 20 ma supply is also needed to program the PROMs if that board is used. These figures should give the user an estimate of the power requirements for his system. They are typical measured values, and somewhat less than the manufacturers' listed maximum worst case values.

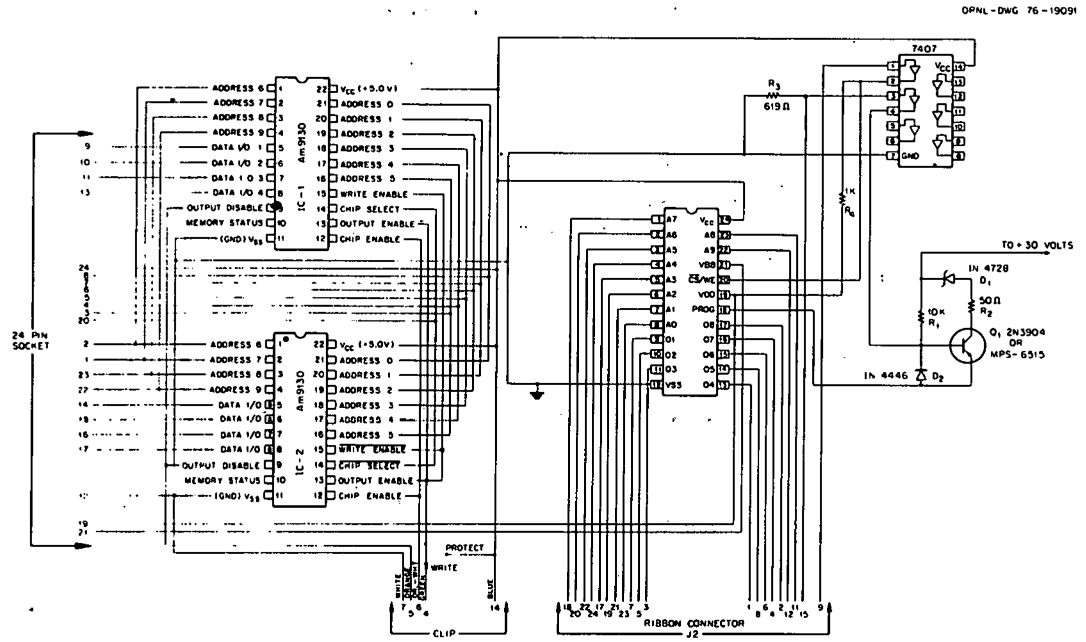

Fig. 11. PROM programmer and simulator.

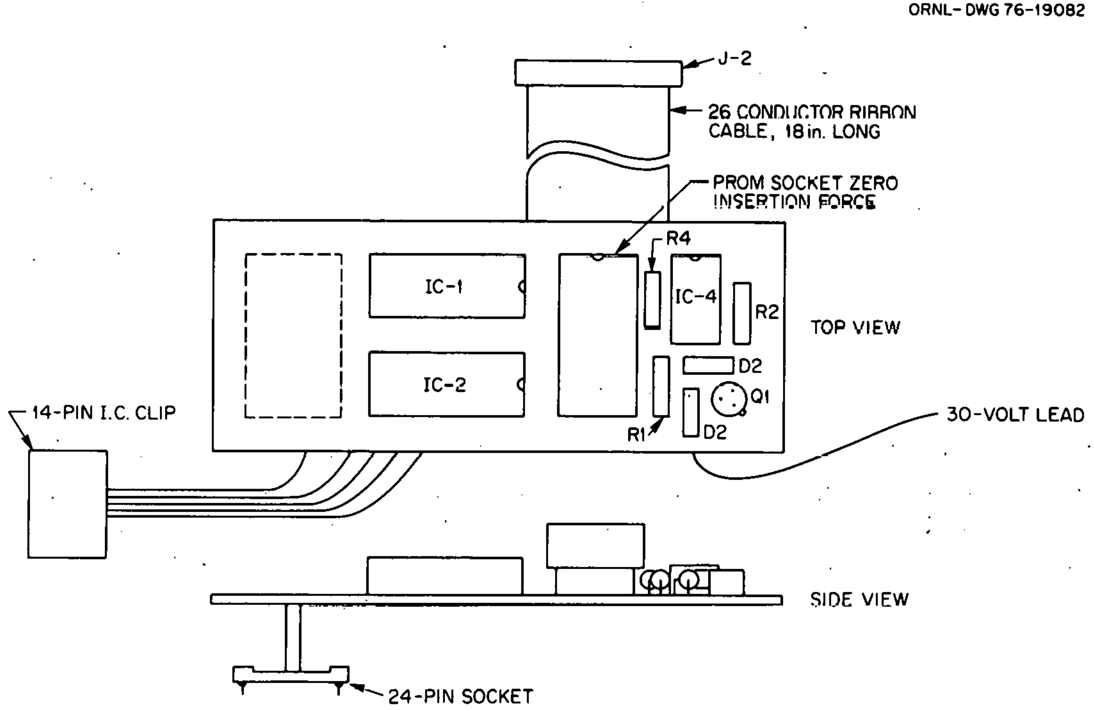

Fig. 12. PROM simulator and programmer board.

Table 8. Parts list for prom simulator and programmer

| Part no. | Description |
|---|---|
| A1 | 2N3904 or MPS-6515 |
| D1 | 1N4728 Zener diode |
| D2 | 1N4446 |
| IC-4 | N7407 Buffer open collector |
| IC1 & 2 | AM9130 1KV4 Random access memories |
| R1 | 10K 1/4 Watt |
| R2 | 50 ohm 1/4 Watt |
| R3 | 619 ohm 1/4 Watt |
| R4 | 1K 1/4 Watt |
| SW-1 | SPDT Slide switch |
| 2 each | 22 Pin I.C. socket ICN-224-S2-T (Robinson-Nugent) |
| 1 each | 14 Pin I.C. socket ICN-143-S2-T (Robinson-Nugent) |
| 1 each | 3 Pin transistor socket |
| 1 each | Zero force insertion socket, 24 pin TEXTOOL 224-3344 |
| 1 each | 24 Pin discrete component socket, MPB-246 (Robinson-Nugent) |
| 1 each | 18" 26 Conductor ribbon cable |
| 1 each | 14 Pin I.C. clip |
| 1 each | 26 Pin ribbon connector |
| 2 each | 6/32 1/2" long stand offs |

Table 9. Power requirements for NDT-COMP8 microcomputer with two different memory and I/O configurations

| +5V ±5% | +12V ±5% | −12V ±5% | RAM | PROM | I/O Ports |
|---|---|---|---|---|---|
| 950 ma | 230 ma | 150 ma | 4K | 4K | All |
| 323 ma | 88 ma | 28 ma | 1K | 1K | None |

## IX. ACKNOWLEDGEMENTS

The authors would like to express their appreciation to C. E. Stevenson for laying out the board, N. W. McCoy and C. F. Ruffner for constructing the computers, E. V. Davis and L. D. Chitwood for making the circuit drawings, Martin L. Bauer and W. A Simpson for their helpful suggestions for both the computers and this report, and W. E. Deeds for his review of this manuscript. Finally, we wish to acknowledge the Technical Publications Department for editing and final preparation of the manuscript.

## *INTERNAL DISTRIBUTION*

1–2. Central Research Library
3. Document Reference Section
4–10. Laboratory Records Department
11. Laboratory Records, ORNL RC
12. ORNL Patent Office
13. R. K. Adams
14. M. L. Bauer
15. C. R. Brinkman
16. D. A. Canonico
17. K. V. Cook
18. K. J. Cross
19. F. L. Culler
20. J. E. Cunningham
21, J. H. DeVan
22–35. C. V. Dodd
36. R. G. Donnelly
37. G. Fee
38. B. E. Foster
39. J. Garber (ORGDP)
40. W. O. Harms
41. R. F. Hibbs
42–44. M. R. Hill
45. D. O. Hobson
46. V. C. Jackson (Y-12)
47. W. R. Martin (Y-12)
48. D. L. Mason (Y-12)
49. R. W. McClung
50. H. E. McCoy
51. J. M. McGuffey
52. C. J. McHargue
53. E. McNeese
54. F. Mundt (ORGDP)
55. P. Patriarca
56. H. Postma
57. M. J. Roberts
58. G. W. Scott
59. R. L. Simpson
60. W. A. Simpson
61. G. M. Slaughter
62. J. H. Smith
63. D. B. Trauger
64. J. R. Weir, Jr.
65. N. E. Promisel (consultant)

## *EXTERNAL DISTRIBUTION*

66–67. ERDA DIVISION OF NUCLEAR FUEL CYCLE AND PRODUCTION, Washington, DC 20545

Director

68–69. ERDA DIVISION OF NUCLEAR RESEARCH AND APPLICATION

70–71. ERDA DIVISION OF REACTOR DEVELOPMENT AND DEMONSTRATION, Washington, DC 20545

Director

72–73. ERDA OAK RIDGE OPERATIONS OFFICE, P.O. Box E, Oak Ridge, TN 37830

Director, Research and Technical Support Division
Director, Reactor Division

74–100. ERDA TECHNICAL INFORMATION CENTER, Office of Information Services, P.O. Box 62, Oak Ridge, TN 37830