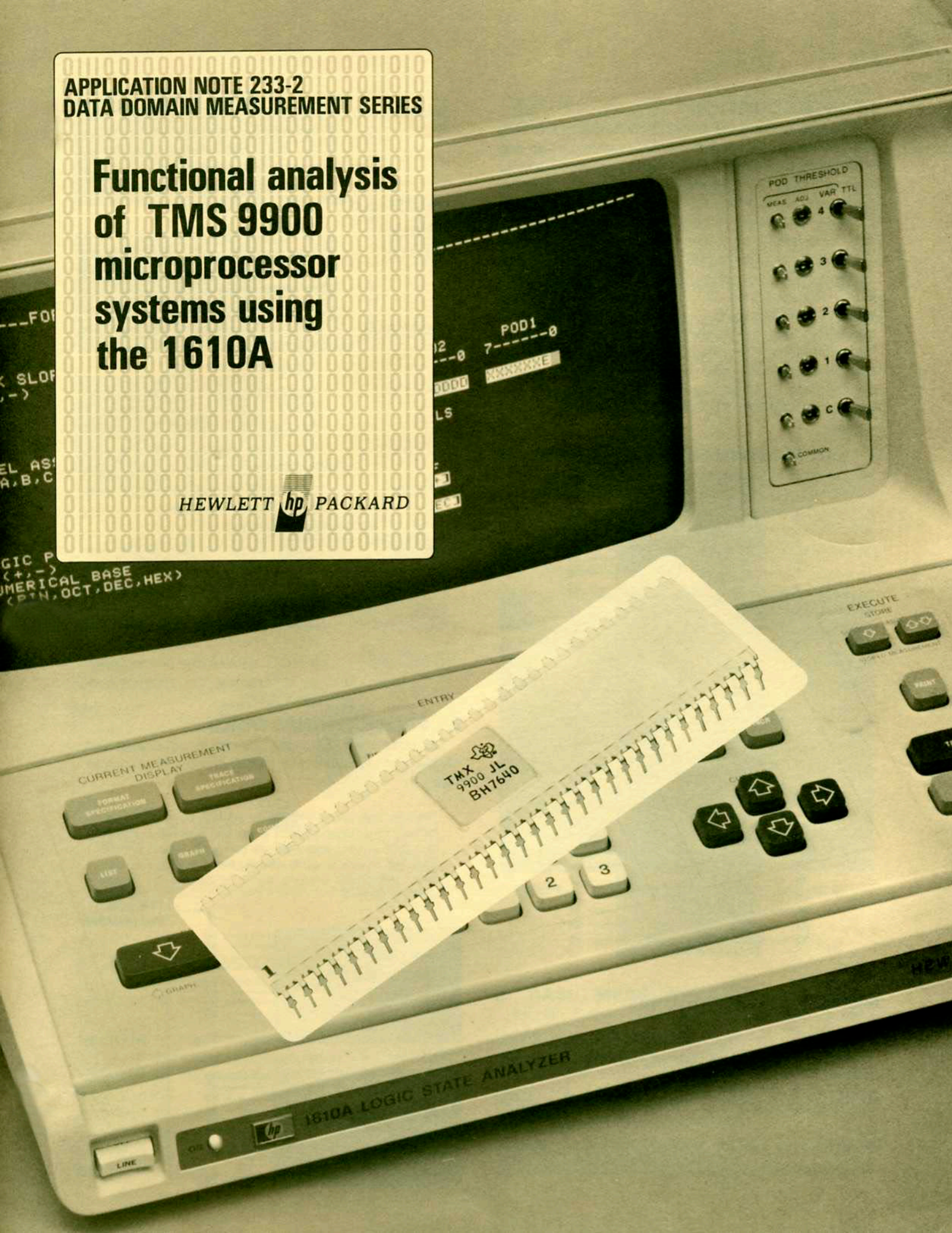
APPLICATION NOTE 233-2
DATA DOMAIN MEASUREMENT SERIES

Functional analysis of TMS 9900 microprocessor systems using the 1610A

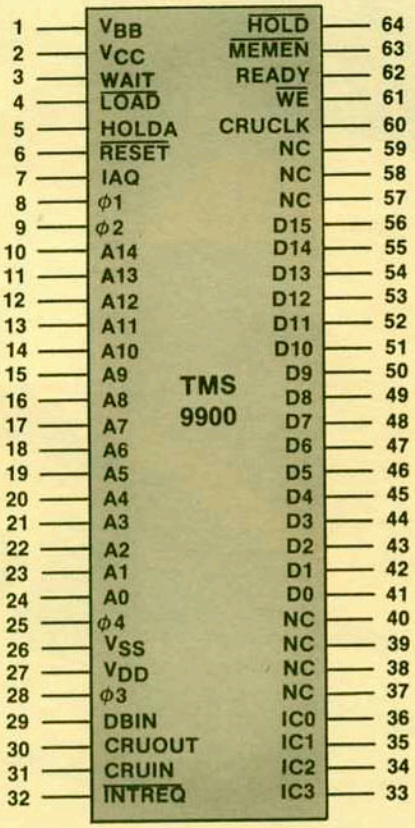
HEWLETT hp PACKARD



1. INTRODUCTION

This application note is intended to assist the TMS 9900 Microprocessor user in initial system turn-on and analysis by explaining the techniques of real time state analysis. These techniques are illustrated by using the general purpose HP Model 1610A Logic State Analyzer to perform the measurements. System instrumentation, data acquisition, data formatting, and measurement techniques are explained. The TMS 9900 Microprocessor is a single-chip 16-bit central processing unit (CPU) that operates to 3 MHz. The instruction set includes the capabilities generally offered by minicomputers with a memory-to-memory architecture having register files resident in memory for fast response to interrupts and flexible programming. There are separate 16-bit address and 16-bit data buses that are compatible with TTL and MOS memory and logic circuits.

2. PIN ASSIGNMENTS



SUMMARY OF CONTROL LINES

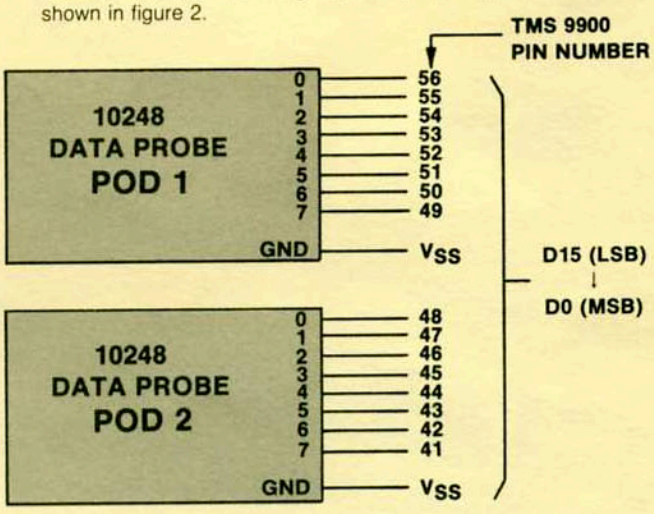
DBIN (out)	High state allows memory to place read data on the data bus during $\overline{\text{MEMEN}}$.
$\overline{\text{MEMEN}}$ (out)	Low state indicates that address bus contains a memory address.
$\overline{\text{WE}}$ (out)	Low state indicates that memory write data is available from the TMS 9900.
CRUCLK (out)	CRUCLK indicates CRUOUT contains CRU (communications register unit) output data or that A0-A2 contain the code for an external instruction.
CRUOUT (out)	CRU output data line.
CRUIN (in)	CRU input data line.
$\overline{\text{HOLD}}$ (in)	Low state places the TMS 9900 in a hold mode, allowing external logic to control memory data transfer as in the case of DMA.
HOLDA (out)	Hold acknowledge (high) indicates that all memory signals are in the high-impedance state.
READY (in)	High state indicates that memory may be read from or written into on the next clock cycle.
WAIT (out)	High state indicates that the TMS 9900 has entered a wait state.
IAQ (out)	Instruction Acquisition (high) indicates that the TMS 9900 is fetching an instruction.
$\overline{\text{LOAD}}$ (in)	$\overline{\text{LOAD}}$ causes the TMS 9900 to execute a nonmaskable interrupt with the trap vector contained in memory address FFFC.
$\overline{\text{RESET}}$ (in)	$\overline{\text{RESET}}$ causes the TMS 9900 to be reset.
$\overline{\text{INTREQ}}$ (in)	$\overline{\text{INTREQ}}$ causes the microprocessor to examine the interrupt code bits IC0-IC3.

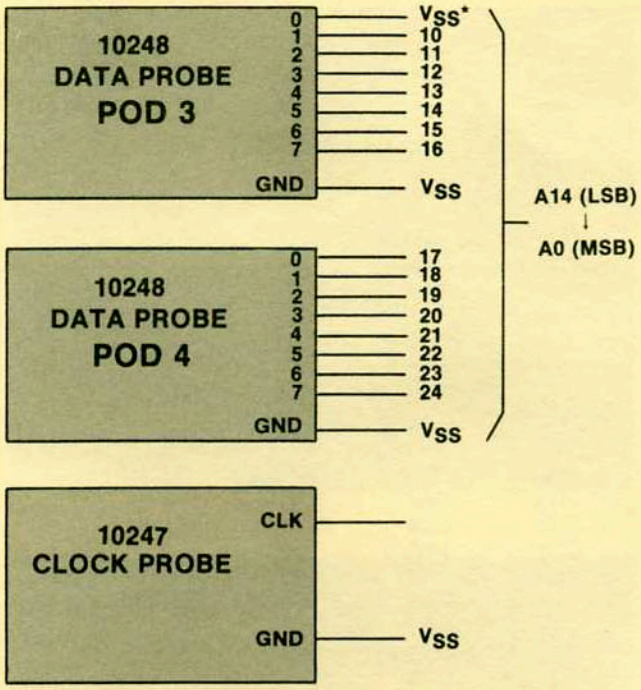
3. PROBE CONNECTIONS

Connection to the system is made directly at the microprocessor using the 1610A probes as shown in figure 1.

NOTE

The logic state analyzer requires a clocking signal during the memory cycle when $\overline{\text{MEMEN}}$ is low and address and read data (RD) or CPU write data are both valid. If your system does not provide a signal meeting these criteria, you can obtain the proper clocking signal by building the circuit shown in figure 2.





*The TMS 9900 accesses external memory in a word (two 8-bit bytes) mode. Thus, all memory locations are on even address boundaries. Ground the 0 input of Pod 3 so that the 1610A display agrees with TMS 9900 addressing (all even addresses).

Figure 1. Data and Clock Probe/Microprocessor Connections.

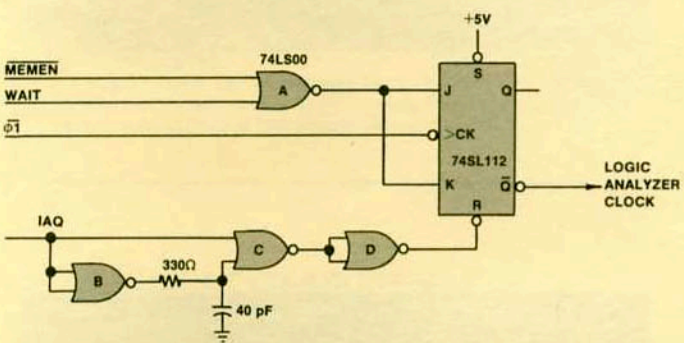


Figure 2. Circuit for Deriving a Clock for the 1610A from the TMS 9900 Control Lines.

4. DATA ACQUISITION

The electrical characteristics of the TMS 9900 Microprocessor are defined to be MOS and TTL compatible. This means that all the 1610A probe pod thresholds can be set to the TTL position.

The timing relationship diagrams of the TMS 9900 indicates that the clock for the 1610A should occur when there is a valid address and when the microprocessor reads data. Therefore, the 1610A should be clocked by a signal that will assure the proper synchronization or by a combination of signals through an external circuit (figure 3).

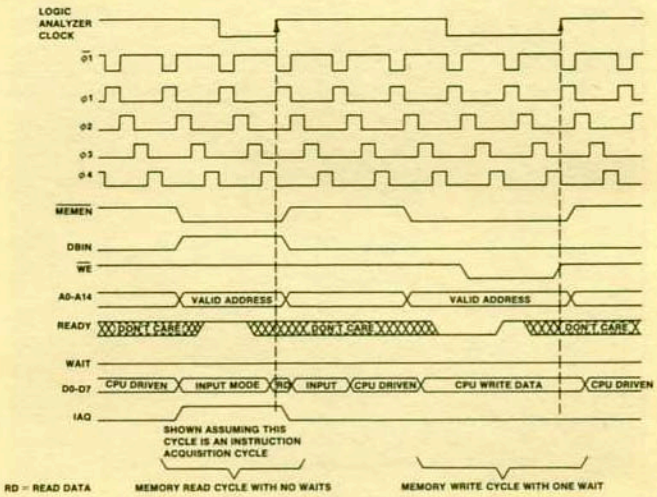


Figure 3. General TMS 9900 Timing Diagram.

5. FORMAT SPECIFICATION

By calling the FORMAT SPECIFICATION menu, the data fields may now be assigned by Label and Bases as shown in figure 4. At this time, the positive [+] clock edge is selected. By assigning the hexadecimal base to Labels A and D the address and data will be displayed in a convenient manner.

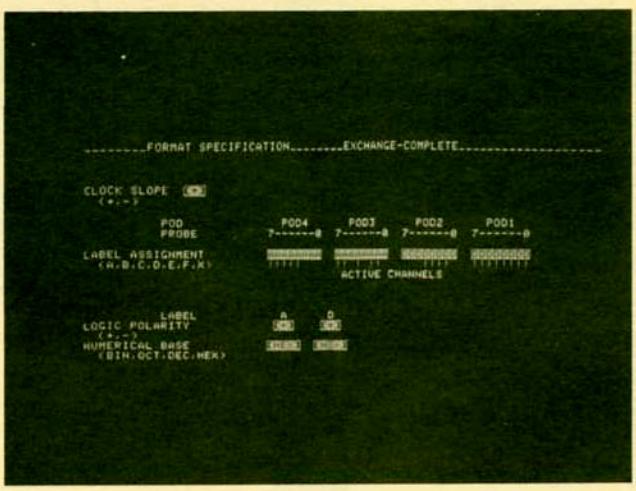
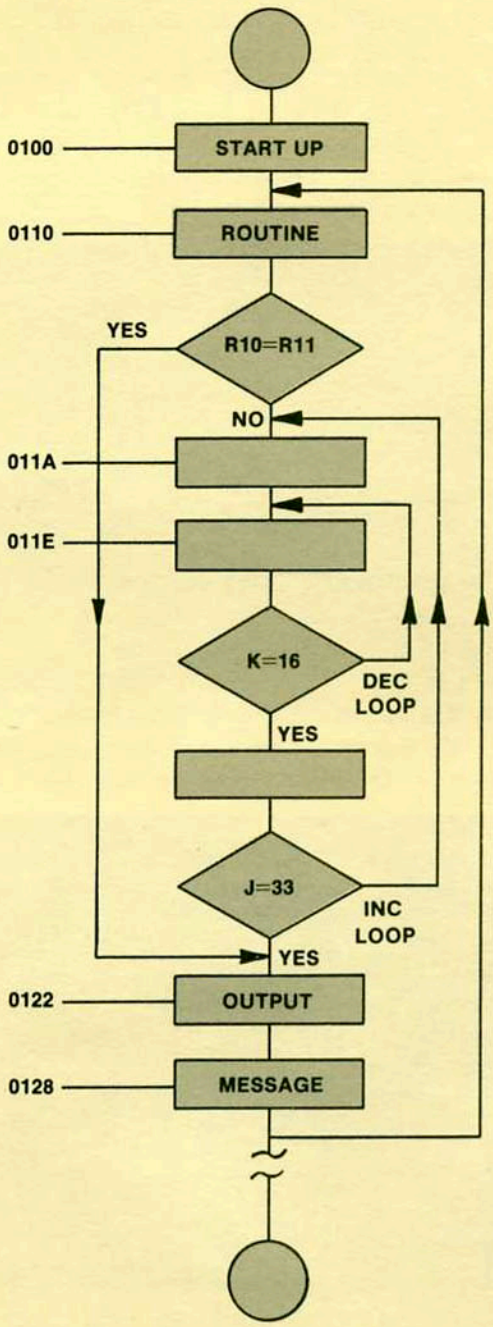


Figure 4. Format Specification with Label and Base Assignments.

6. BASIC MEASUREMENTS

Tracing program execution from its start up point (figure 5) is accomplished by calling the TRACE SPECIFICATION menu and entering the address (in this example 0100_{16}) in figure 6. By pressing the Trace key and starting the program, the 1610A captures the desired information starting at address 0100_{16} (figure 7).

Analysis of the captured data is easily accomplished by comparison to the program listing. The trace point




```
                0100    02E0    LWPI,   >00C0
                0102    00C0
                0104    020A    LI      R10,>1
                0106    0001
                0108    020B    LI      R11,>2
                010A    0002
                010C    0209    LI      R9,>21
                010E    0021
ROUTINE         0110    0B1A    SRC     R10,1
                0112    2288    COC     R10,R11
                0114    1309    JEQ     OUTPUT
                0116    0208    LI      R8,>0
                0118    0000
INC             011A    0207    LI      R7,>10
                0110    0010
DEC             011E    0607    DEC     R7
                0120    16FE    JNE     DEC
                0122    0588    INC     R8
                0124    8209    C       R8,R9
                0126    16F9    JNE     INC
OUTPUT          0128    2FA0    XOP     MESSAGE,14
                012A    0132
                012C    0420    BLWP      FFFC
                012E    FFFC
                0130    10EF    JMP     ROUTINE
MESSAGE         0132    0A0D            LF,CR
                0134    434F            C,O
                0136    4D50            M,P
                0138    4C45            L,E
                013A    5445            T,E
                013C    4420            D,SPACE
                013E    2000            SPACE,NUL
                0140    0380    RTWP
```

Figure 5. Sample TMS 9900 Program Containing a Two-level Nested Loop.

(0100_{16}) is identified by START (figure 7a) which is defined by the Trace Specification. From the program listing (figure 5), it can be confirmed by the trace list (figure 7a) that data $02E0_{16}$ was read from 0100_{16} and that data 0000_{16} follows from address 0102_{16}. By using this comparison procedure, program flow can be examined to determine that the system is operating properly. By pressing the logic analyzer Roll key, program flow can be viewed as it enters a loop (figure 7b).

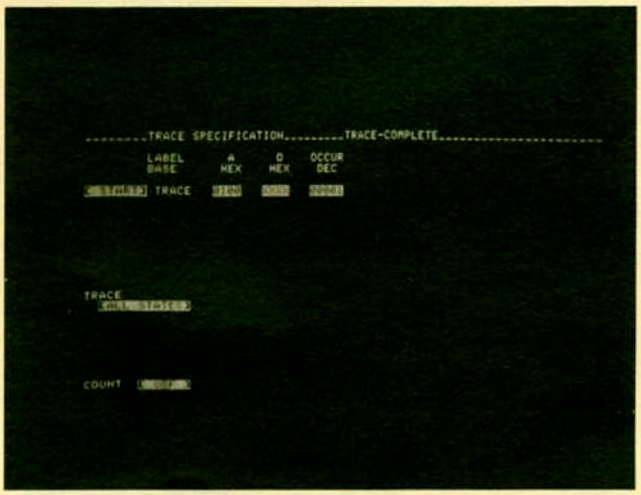
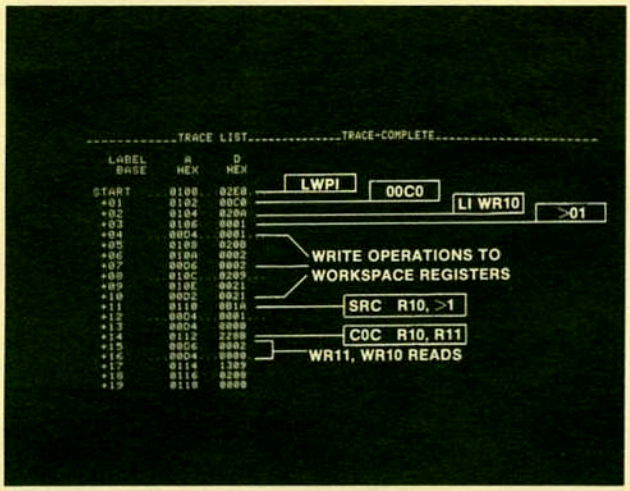
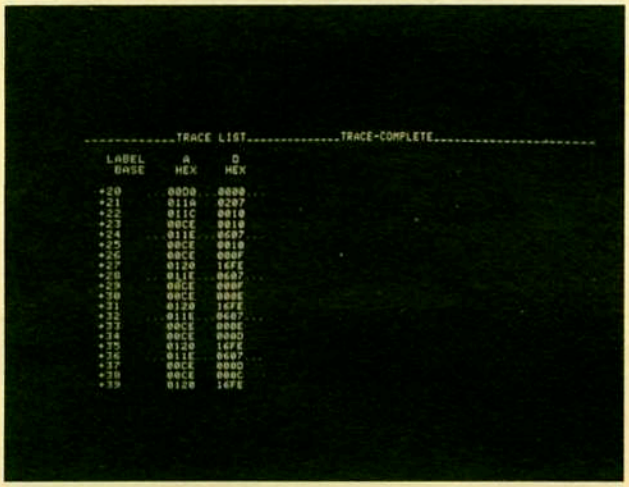


Figure 6. Trace Specification for capturing TMS 9900 Start-up Routine.



a.



b.

Figure 7. Trace List of Program Start-up Routine (a) from the Trace Specification in figure 6. By using the Roll keys program flow can be viewed as the system enters a loop at line 24 (b)

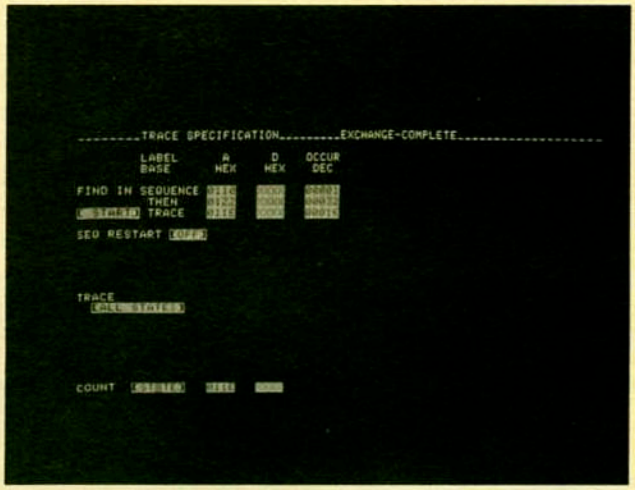


Figure 8. Trace Specification for verifying proper Loop Execution.

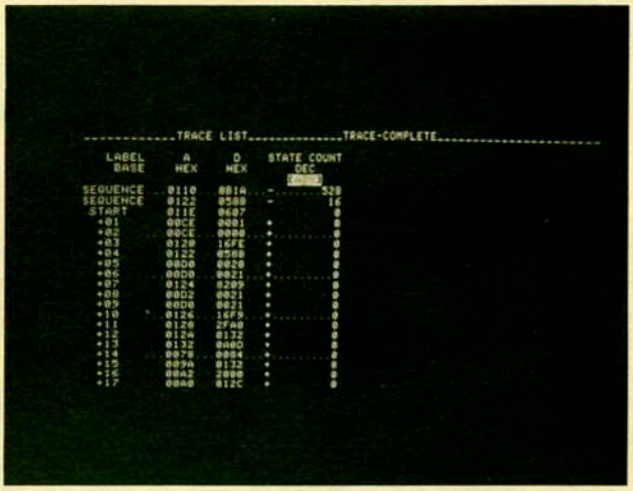


Figure 9. Resultant Trace List from figure 8 Trace Specification with Absolute Count showing proper Loop Operation.

7. COMPLEX MEASUREMENTS

Tracing program flow containing a loop introduces another problem because each iteration of that loop need not be examined. As shown in figure 5 the program contains a major loop (INC) which is executed 33 times and a minor loop (DEC) which is executed 16 times in each INC loop. To verify that the program actually executed the number of specified loops, call the TRACE SPECIFICATION menu and enter the number of occurrences of state $011E_{16}$ to be counted and the sequence specified in figure 8 to capture one pass through the loop. From figure 5, state $011E_{16}$ should occur 528 times before the specified trace starts.

$$\frac{INC}{33} \times \frac{DEC}{16} = 528$$

The resulting trace list with an absolute count of state $011E_{16}$ shows (figure 9) that state $011E_{16}$ occurred 528 times during execution of the loop--exactly as calculated.

8. TRACING SPECIFIC PATHS IN BRANCHING NETWORKS

When tracing program flow, it is often desired to trace a specific path in a branching network to determine if a problem occurs only when that path is taken. Examination of the flow chart in figure 5 shows that the output routine (0128_{16}) can be entered from the INC loop or by a direct jump from address 0114_{16}. Assume that you want a view of the output routine only when it is entered from the direct jump. If a simple trigger on 0114_{16} or a find in sequence of 0114_{16} and 0128_{16} is used, both branches would satisfy the trigger specification. However, by using the sequence restart capability of the 1610A, a trace specification unique to the direct jump path can be defined.

As shown in figure 10, a sequence restart of don't cares (all X's) will direct the 1610A to first look to see if a sequence condition is satisfied and, if not, if a restart condition is satisfied. Therefore, if any state other than 0128_{16} occurs after 0114_{16} the 1610A will return and look for 0114_{16}. Only when the sequence of 0114_{16} directly followed by 0128_{16} is found will a trace be captured. Examination of the

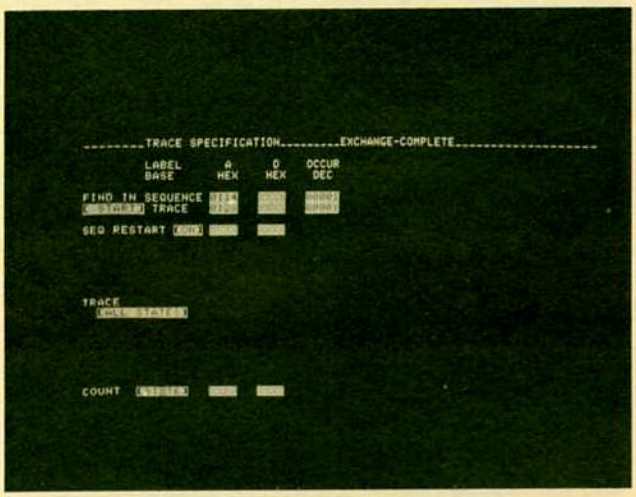


Figure 10. Trace Specification for Capturing a Direct Jump in a Branching Network.

resulting trace (figure 11) specified in figure 10 verifies the direct jump to the output subroutine and the state count also verifies that state 0114_{16} was followed directly by state 0128_{16}.

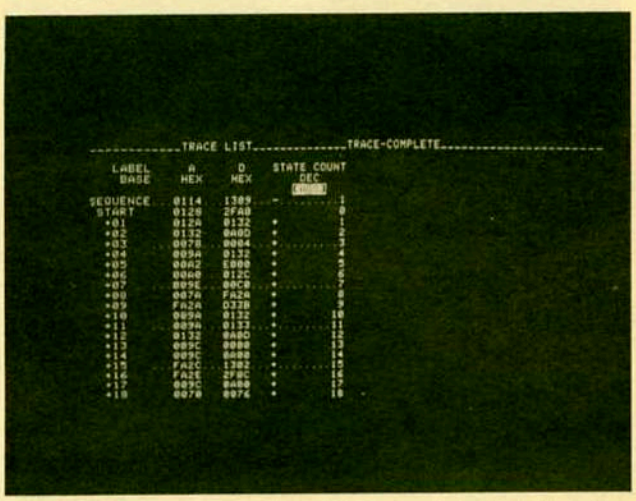


Figure 11. Trace List of the Direct Jump defined by the Trace Specification in figure 10.

9. DISPLAY QUALIFICATION

If the system configuration and/or measurement requirement is such that it is not necessary to monitor address bit A0, i.e. the monitored program resides within address locations 0000_{16} to $7FFF_{16}$ and no external instructions are executed, bit 7 of pod 4 can be used to monitor control lines. In this example, probe 7 of pod 4 is disconnected from address bit A0 and connected to the DBIN line (pin 29 of the CPU). Call the Format Specification and relabel pod 4, bit 7, as E and select binary base for Label E (figure 12).

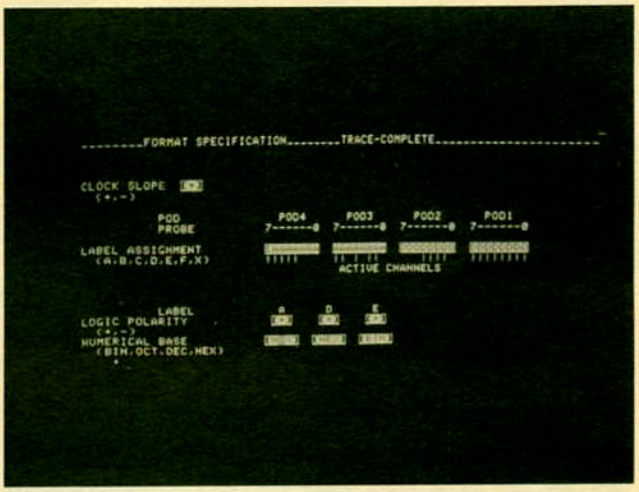


Figure 12. Format Specification when using Pod 4, Bit 7, as a Qualifier to Acquire Only Memory Write Transactions.

Set up a Trace Specification to [START] a Trace at address 0110_{16} and to Trace [ONLY STATES] E (DBIN) of 0_2 (DBIN LOW) (figure 13) which will direct the analyzer to acquire only memory write

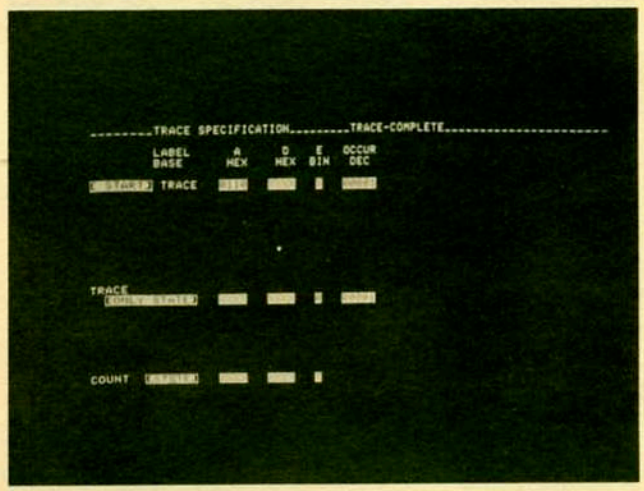


Figure 13. Trace Specification to Qualify the Logic Analyzer to only Capture Memory Write Transactions.

transactions. The resulting trace list (figure 14) shows that the analyzer only traced write operations. Observe that the contents of memory location $000E_{16}$ (WR7) is decremented 16 times for each time the contents of memory location 0000_{16} (WR8) is incremented which again verifies proper operation of the nested loop. Qualification effectively permits analysis of a much larger segment of program execution than 64 sequential lines and displays only states of interest which is verified by the state count of 257 at line 63 (figure 14). To capture only memory read operations, simply change the Trace Specification to Trace [ONLY STATE] E=1.

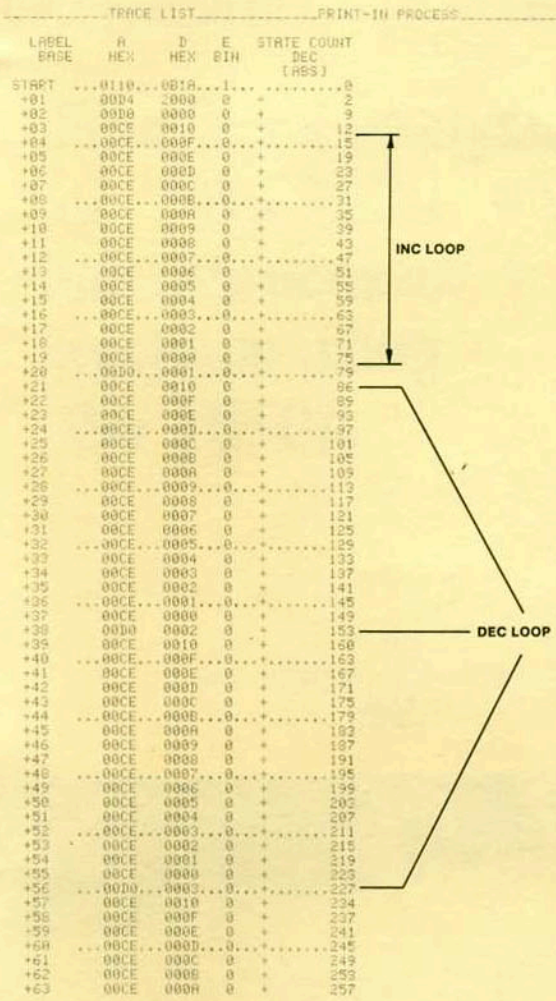


Figure 14. Qualified Trace List shows that only Memory Write Transactions were captured.

If you only need to view address flow, the 16 data lines are available for monitoring control signals and other lines in your system as well as being used as qualifiers.

Another TMS 9900 control line that is useful as a qualifier is the IAQ line (pin 7). Tracing only states that occur when IAQ is high results in a trace list containing only instruction fetches. In this example, the Trace Specification is set to trace only memory transactions corresponding with the high states of the IAQ line (figure 15) and to start a trace at address 0100_{16}. Now, by pressing Trace and restarting the program, a trace list containing only instruction fetches is obtained (figure 16).

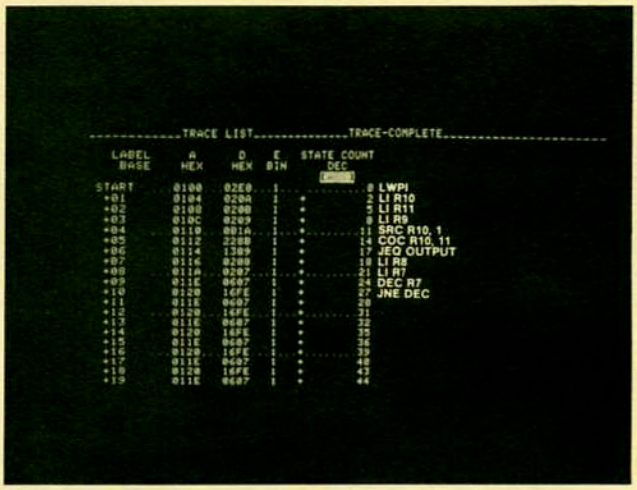


Figure 16. Trace List of Qualified Display from figure 15 showing only Instruction Fetches.

10. STATE FLOW GRAPH

The graph mode provides an overview of all 64 states of one label in memory which are displayed as a graph of state magnitude vs state flow (figure 17). The graph in figure 17 shows the activity of label A resulting from the trace in figure 16. The cyclic nature of the nested loop is very apparent and is displayed as a tight two-state decrement loop with a jump to INC after 16 passes. A dynamic graph of system activity can be obtained by pressing and holding the Trace key until Trace continuous is displayed by the Analyzer.

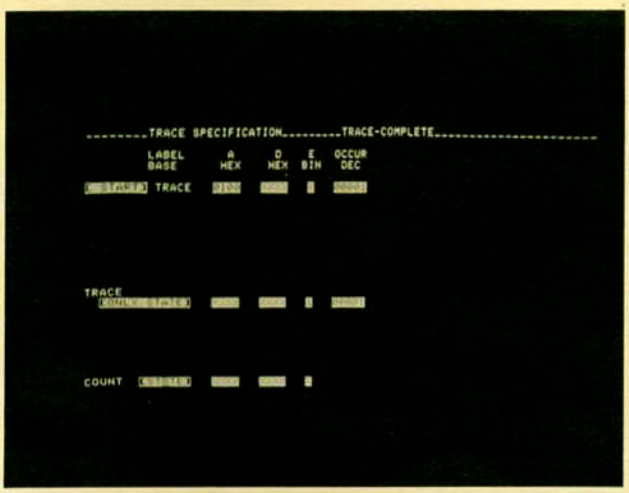


Figure 15. Trace Specification for using the IAQ Line as a Qualifier to capture Instruction Fetches.

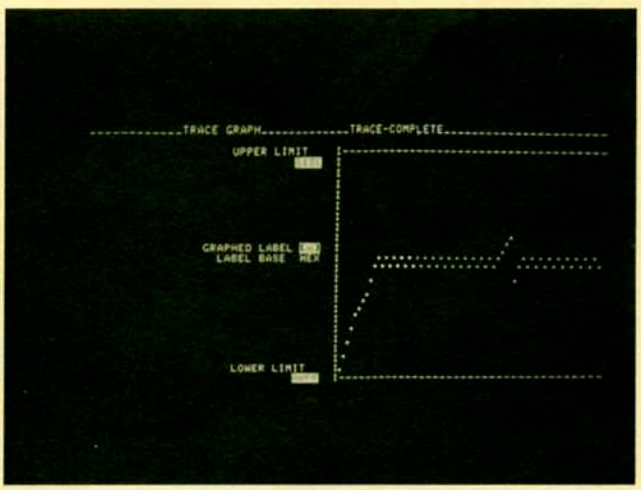


Figure 17. Trace Graph provides an overview of all 64 words in memory for an overview of machine operation.

HEWLETT hp PACKARD

Application Notes in the 233 series for the HP 1610A.

233-1 Functional Analysis of 2650 Microprocessor Systems (1610A).
233-2 Functional Analysis of TM9900 Microprocessor Systems (1610A).
233-3 Functional Analysis of Z80 Microprocessor Systems (1610A).
233-4 Functional Analysis of 8080 Microprocessor Systems (1610A).
233-5 Functional Analysis of 6800 Microprocessor Systems (1610A).

Application Notes in the 167 series for analysis of microprocessor based systems with the HP 1600S.

167-9 Functional Analysis of M6800 Microprocessor Systems (1600S).
167-11 Functional Analysis of 8008 Microprocessor Systems (1600S).
167-12 Functional Analysis of F8 Microprocessor Systems (1600S).
167-14 Functional Analysis of 8080 Microprocessor Systems (1600S).
167-15 Functional Analysis of 4004 Microprocessor Systems (1600S).
167-16 Functional Analysis of 4040 Microprocessor Systems (1600S).
167-17 Functional Analysis of IMP Microprocessor Systems (1600S).

For More Information, Call Your Local HP Sales Office or, in U.S., East (301) 948-6370. Midwest (312) 255-9800. South (404) 434-4000. West (213) 877-1282. Or, Write: Hewlett-Packard, 1501 Page Mill Road, Palo Alto, California 94304. — In Europe, Post Office Box 85, CH-1217 Meyrin 2, Geneva, Switzerland. — In Japan, YHP,1-59-1, Yoyogi, Shibuya-ku, Tokyo, 151.

5952-2058

PRINTED IN U.S.A.