Automatic checkout of small computers

by MARVIN S. HOROVITZ

Digital Equipment Corporation
Maynard, Massachusetts

INTRODUCTION

The testing of a computer may take on many forms, one form is as follows. The computer may be tested in a segmented manner.

IC
MODULE
POWER SUPPLY
MEMORY
WIRING
CP TEST (PHASE 1)
ENVIRONMENTAL (PHASE 2)
ACCEPTANCE (PHASE 2)

Each of the above segments makes use of either a dedicated logic exerciser or a dedicated computer or a time allocated computer system. The IC, MODULE, POWER SUPPLY, MEMORY and WIRING are assembled and tested prior to the entrance of Phase 1 and Phase 2 testing. This paper deals only with Phase 1 and 2 testing.

Phase 1 testing

The objective of Phase 1 is to exercise a computer under test and check the resulting status of that machine. In designing a test system of this nature a prime concern is finding any malfunction that may exist within the computer under test. Each mode of every instruction, Memory System, I/O Interface and Manual Control must be completely validated. To accomplish this, initial conditions must be established. The computer under test (PDP-8/I) must then be stimulated and the results of the operation monitored for comparison with anticipated data. If an error is detected, the test system must have the capability of reporting the error and repeating the test the computer failed to perform. The system is designed so that the computer under test can never impede the repeated cycling of any test.

System hardware

To maintain continuous testing of the PDP-8/I at its normal clock speed would require a test system with enormous capabilities. A more practical approach is to design a system that will test the computer at its normal clock speed, but only for short increments of time. The test system hardware can be designed so that when properly preconditioned by the Test System Software, it will cause the PDP-8/I to function at normal speed for the specified interval with no further stimulus. Time between tests is used for checking the results of the previous test and to set up for the next. This time is large compared to the actual time spent testing. This enabled us to use the PDP-8 as the test systems computer. Design is simplified when the test systems computer and the computer under test are the same type machine. Data manipulation by both hardware and software is facilitated when both machines use the same word size. Also, a computer of the same type is usually available in the production area and need not be a permanent part of the test system.

The test system must have an input peripheral device for program loading and an output device for reporting errors. This test system uses a disk for input and a printer for output.

There is a special interface which acts as a control and sensing device to the computer under test and is completely program controlled by the test system. With the appropriate commands, the tester will perform the following test functions in the PDP-8/I under test through the special interface.

1. Manipulate any key or switch logic.
2. Simulate memory read operations by transferring data into the memory buffer at the proper time.
3. Sense memory read and write drive pulses produced in the memory address decoding circuits.

4. Sense memory inhibit drive pulses produced during the memory write cycle.
5. Sense the ON or OFF state of all indicator light source functions.
6. Sense the one or zero state of all timing clocks such that any timing state or pulse can be activated at will.
7. Control the PDP-8/I clock speed.

Figure 1 shows entire Phase 1 Test System Flow Chart and Figure 1/P shows the actual test station.

The PDP-8/I computer under test

Most central processors are equipped with an operator's console which uses switches, keys, and indicator lights. Generally, connection of this console is made to the central processor logic panel with plug-in type cables. While the PDP-8/I is being tested by the system, the operator's console is not used and these cable connections are made to the test system. All manual switch controls are operated remotely by the test system. The Test System program can simulate any of the actions of a console operator from starting and stopping the processor to observing the indicator lights.

For control of timing, appropriate modules are removed and timing control lines are cabled in from the test system. Control is such that the PDP-8/I may be run at any of several speeds from faster than normal to pulse by pulse.

The input-output peripheral buss cables of the PDP-8/I are connected to the test system. This affords an extensive test for the I/O interface in that the initiated peripheral commands can be monitored, and the response to peripheral initiated actions can be sensed by the test system.

During the initial stages of testing, the computer is exercised without a memory stack. Cables from the vacant memory area in the computer logic panel to the test system enable the system to monitor memory read, write, and inhibit pulses and simulate memory core changes on the sense amplifier input lines.

When some confidence has been established in the basic processor control logic and the memory control circuitry, the pre-tested memory stack and memory sense amplifiers may be inserted and more advanced tests carried on. Various combinations of machine instructions can be loaded into the PDP-8/I memory to form test loops that can be initiated and stepped through much as an operator would do at the operator's console. These program loops can be started and run at normal machine speed. Any program can be loaded and run in this manner which duplicates the condition of the program running in a free standing machine. However, in the event of a program wipe-out, due to a machine malfunction, a programmed high-speed reload and restart enables the system operator to observe repeated failures on an oscilloscope. This continuous cycling feature is not possible on a free standing machine when a wipe-out failure occurs.

System software

The system hardware provides many avenues through which the programmer may channel his efforts in attempting to find all existing problems in the computer under test. Two general approaches will be considered here, one being diagnostic in nature, the other being a specification check.

The specification check consists of outlining in detail every particular operation that the PDP-8/I is designed to perform. A series of tests is then written using various initial conditions and operands for each operation. Each test is provided with several check points at which the test system monitors the state of every available status function. Every status at each check point has been predicted by the programmer and is stored in internal tables of the test system program. If the monitored status is wrong, the comparison of the anticipated status with the monitored status results in an error report at the test systems printer.

The approach discussed above does not anticipate a failure in any particular hardware area at any time, nor does it attempt to exercise one area of the hardware

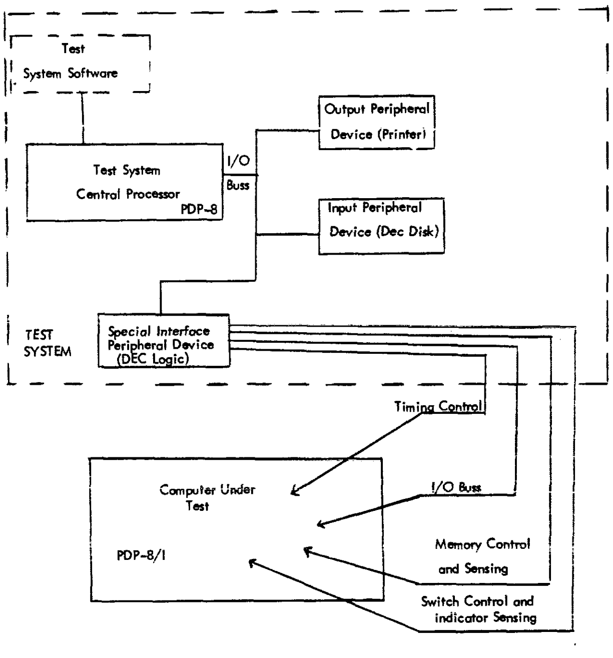

Test
System Software
Output Peripheral Device (Printer)
Test System Central Processor
PDP-8
I/O
Buss
Input Peripheral Device (Dec Disk)
TEST SYSTEM
Special Interface Peripheral Device (DEC Logic)
Timing Control
Computer Under Test
PDP-8/I
I/O Buss
Memory Control and Sensing
Switch Control and indicator Sensing

Figure 1—Flow chart of Phase 1 test system

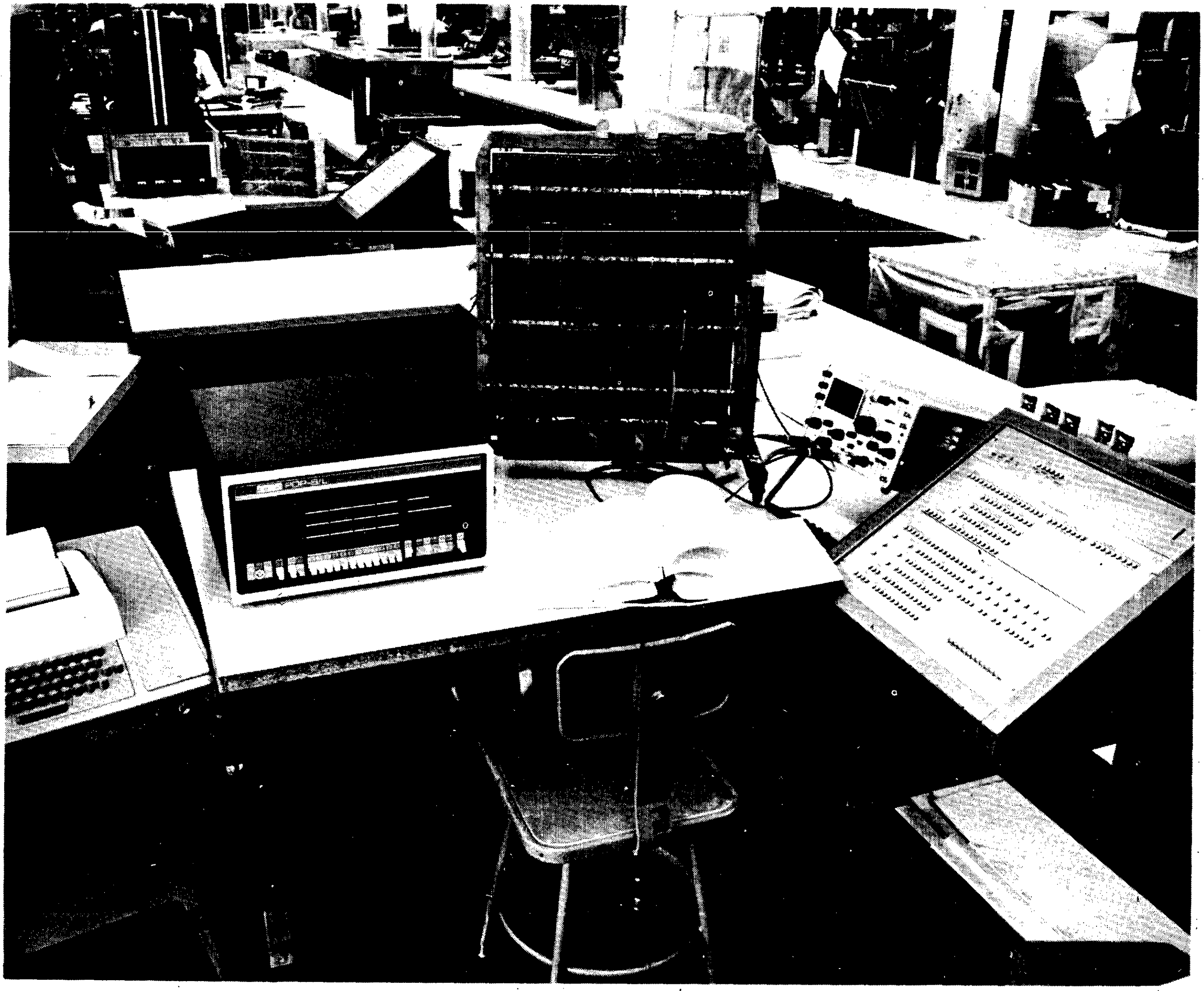


Figure 1/P—Actual Phase 1 test system

more than another, during a selected group of tests. It does, however, cause the PDP-8/I to peiform its specified operations under various conditions and checks for any sign of a malfunction anywhere in the computer during all the tests. In contrast to this specification check, the second software approach is aimed at validating isolated pieces of hardware. The building block method is employed where previously checked logic is used to test other areas.

As individual pieces of hardware are tested in small increments, the table of known functioning logic grows. Validated logic is considered sound when a new test is performed. By analyzing the results of many tests, a failure can be isolated to within a few logic modules. Using the proposed system allows the programmer to expand the set of tests whose results may be used to predict failing modules. This expansion results from the fact that the system can cause the computer to perform all of its specified functions plus many more, such as executing time pulses out of sequence, causing program interrupts at any specific time; stopping the processor clock at any points, etc. In actuality the building block tests are first used and then followed by the specification tests. The Memory System is then integrated with the computers logic. Memory integration tests include exercising memory with simple test patterns through complex test patterns. This is accomplished by using the PDP-8/I data break facility. (Direct memory access.) We now have, for all practical purposes, integrated the memory into the computer under test. The final test is to load small test routines directly into memory and run them at machine speed.

The results of these tests are constantly monitored by the test system. Error control is still handled by the test system. Ultimately through this method, the entire PDP-8/I logic and its memory system are completely tested and a complete computer is born.

Documentation

The more the test technician knows about the tests being made and the expected responses, the easier it will be for him to use the test system effectively in problem solving. For this reason, an extensive set of documents is recommended. This includes a separate timing and print out chart for each test programmed. These charts will list each test system action and the anticipated computer function monitored.

The system error reports will inform the operator of the failing test number, a failing subdivision within that test, the status that is wrong, and what that status should be. With this information the operator can turn to the print out chart for the failing test to get a composite picture of how the test is carried out. Listed, he will see the predicted status of every computer function monitored for evey subdivision of the test. With the entire test outlined before him and the problem area pin-pointed, he can formulate his plan of attack, which

TELETYPE PRINTOUT's

Printout example:	0001	S	GB	G0001	B0201
Printout meaning:	No. of Test	State of 8/I Timing	Group Indicators	Correct Group Indicators	Defective Group Indication
	0001	S	GB	G0001	B0201

Number of tests = test which fault occurs in
State of 8I timing = time or cycle of 8I which test fails in
Group Indicators = registers, states, or significant pulses where test problem occurs
Correct group indication = correct contents of group indicator
Defective group indication = the actual group indication (wrong indication)

GROUP INDICATORS

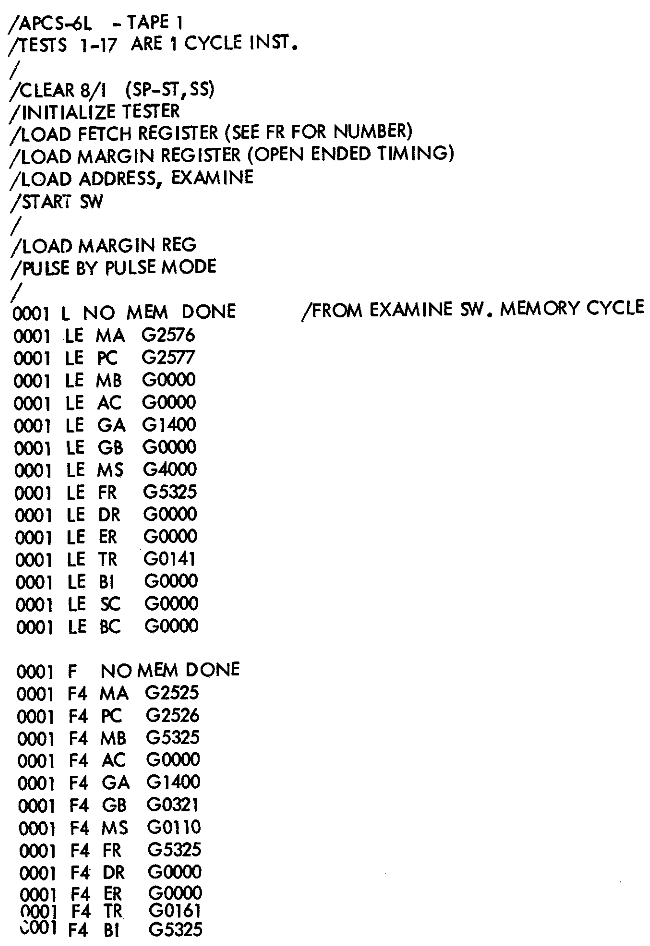
MA	memory address	FR	fetch register
PC	program counter	DR	defer register
MB	memory buffer	ER	execute register
AC	accumulator	TR	transfer register
GA	Group A	BI	buffered MB bits
GB	Group B	SC	step counter
MS	major states	BC	buffered AC

Figure 2—Error message explanation group indicators

GROUP INDICATORS

MA	PC	MB	AC	GA	GB	MS	FR	DR	ER	TR	BI	SC	BC	BIT #
MA Ø	PC Ø	MB Ø	AC Ø	LINC	EXT. MA Ø	AND	FR BIT Ø	DR BIT Ø	ER BIT Ø	IOP 1	BMB Ø (1)	BUFF. BRK.	BAC Ø (1)	0
				PARITY FMR	EXT. MA 1	TAD				IOP 2		BUFF. RUN		1
				PARITY DMR	EXT. MA 2	ISZ				IOP 4				2
				PARITY EMR	$\overline{\text{BMB 3}}$	DCA				ADDR. ACC				3
				W. C.	$\overline{\text{BMB 4}}$	JMS				W. C. OVER FLOW				4
				C. A.	$\overline{\text{BMB 5}}$	JMP				BT 2A				5
				D FØ	$\overline{\text{BMB 6}}$	IOT				B T 1				6
				D F 1	$\overline{\text{BMB 7}}$	OPR				POWER CLEAR		SC 1		7
				D F 2	$\overline{\text{BMB 8}}$	FETCH						SC 2		8
				IF Ø	PAUSE	EXEC.						SC 3		9
				IF 1	I O N	DEFER						SC 4		10
MA 11	PC 11	MB 11	AC 11	IF 2	R U N	BRK.	BIT 11	BIT 11	BIT 11	POWER O K	BMB 11	SC 5	BAC 11	11

Figure 3—Group indicator table



```
/APCS-6L  - TAPE 1
/TESTS 1-17  ARE 1 CYCLE INST.
/
/CLEAR 8/I  (SP-ST,SS)
/INITIALIZE TESTER
/LOAD FETCH REGISTER (SEE FR FOR NUMBER)
/LOAD MARGIN REGISTER (OPEN ENDED TIMING)
/LOAD ADDRESS, EXAMINE
/START SW
/
/LOAD MARGIN REG
/PULSE BY PULSE MODE
/
0001 L  NO MEM  DONE        /FROM EXAMINE SW. MEMORY CYCLE
0001 LE  MA  G2576
0001 LE  PC  G2577
0001 LE  MB  G0000
0001 LE  AC  G0000
0001 LE  GA  G1400
0001 LE  GB  G0000
0001 LE  MS  G4000
0001 LE  FR  G5325
0001 LE  DR  G0000
0001 LE  ER  G0000
0001 LE  TR  G0141
0001 LE  BI  G0000
0001 LE  SC  G0000
0001 LE  BC  G0000

0001 F   NO MEM DONE
0001 F4  MA  G2525
0001 F4  PC  G2526
0001 F4  MB  G5325
0001 F4  AC  G0000
0001 F4  GA  G1400
0001 F4  GB  G0321
0001 F4  MS  G0110
0001 F4  FR  G5325
0001 F4  DR  G0000
0001 F4  ER  G0000
0001 F4  TR  G0161
0001 F4  BI  G5325
```

Figure 4—Error message examples

is the most important phase of any attempt at solving a problem.

Error format

- Column 1 designates the Test Number within the program.
- Column 2 is the check point number within a test. It may be a number (1 to 7) or a letter-number combination (i.e., F3, D3, E1, etc.). If a letter and number, the letter is the cycle name (i.e., F = fetch, D = Defer, etc.) and the number is the time state in which the test is performed. If a single number, the cycle and time should be given under the particular tests format.
- No Mem Done—This indicates the memory was cycled by either a switch function or by programming command and did not complete its cycle.
- Column 3 gives the abbreviation of the register in which an error occurred. (See master chart.) Numbers loaded into the 8/I will appear in some of these registers in the printout marked "G" (Good) after the register. Individual test formats will designate when this is true.
- Column 4 is the "G" or Good number. That is the number or instruction loaded into or predicted from the 8/I under test. They are printed in octal.
- Column 5 is the "B" or Bad number. These are not included in these listings. The "B" column gives the incorrect number from the 8/I. They are printed in octal.

Phase 2

The System Controller is composed of a PDP-8, I/O peripheral equipment, an Inter-Processor Buffer and a residing software package.

The objective of Phase 2 is to provide the technician with a quick means of reloading a Maintenance Program to worst case Margin Test a PDP-8/I. It is also used to automatically accept unattended computers.

The System Controller can be used as a high-speed program loader, or to automatically run a computer under test with diagnostic programs in a predetermined sequence. The programs are stored on a disk at the System Controller. Many stations can be tied to the controller's bus for access to the diagnostic programs. The number of stations is limited only by the dwell time for servicing a station.

Drawn in Figure 5 is a block diagram of the system.

Each station has a console which has switches that the operator uses to select a diagnostic program. He then presses a switch to indicate to the System Controller that he wants a program. Indicators are used

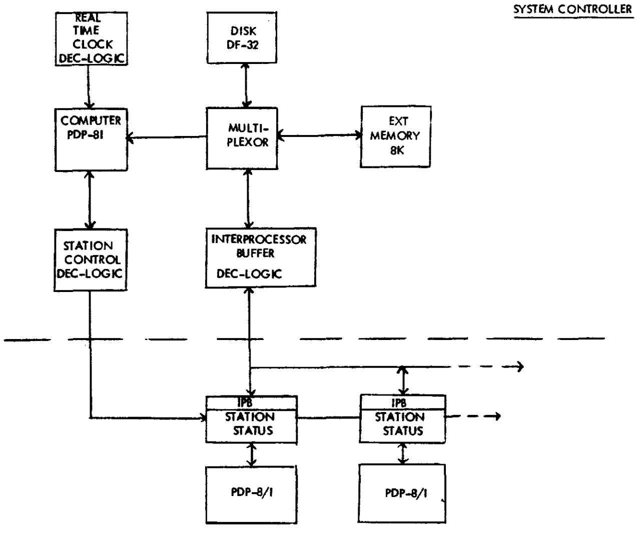


Figure 5—Flow chart of Phase 2 test system

to inform the operator of a successful load or an error condition.

The System Controller connects one station at a time to the bus by enabling that station's gates. The station is then interrogated for program selection. If a program is requested, the desired program is brought from the source disk to the extended memory field of the controller and checksummed. From the extended memory, the program is then loaded into the computer under test with the aid of an Inter-Processor Buffer (computer to computer transfer). After the program has been loaded, the extended memory of the controller is cleared and the program is then sent back to it from the computer under test. The controller then makes a checksum test to determine if the program was successfully loaded If a bit were picked up or lost during the transfer, an error condition would exist. The operator is notified. via the station indicators whether there was an error or successful load (Program loading with overhead 2 seconds).

When being used as a high-speed loader (Figure 5/P), the operator manually calls for the individual Processor Tests from the System Controller. These tests are designed to catch worst case failures caused by temperature, vibration and voltage margins. These are our standard field-oriented maintenance programs.

When being used in the auto accept mode (Figure 5/PA) following a successful load, the computer under test will be told to jump to the beginning of the program and run it for some specified length of time. While the computer under test is running, the controller monitors each on-line station for errors. If an error exists the Sys-

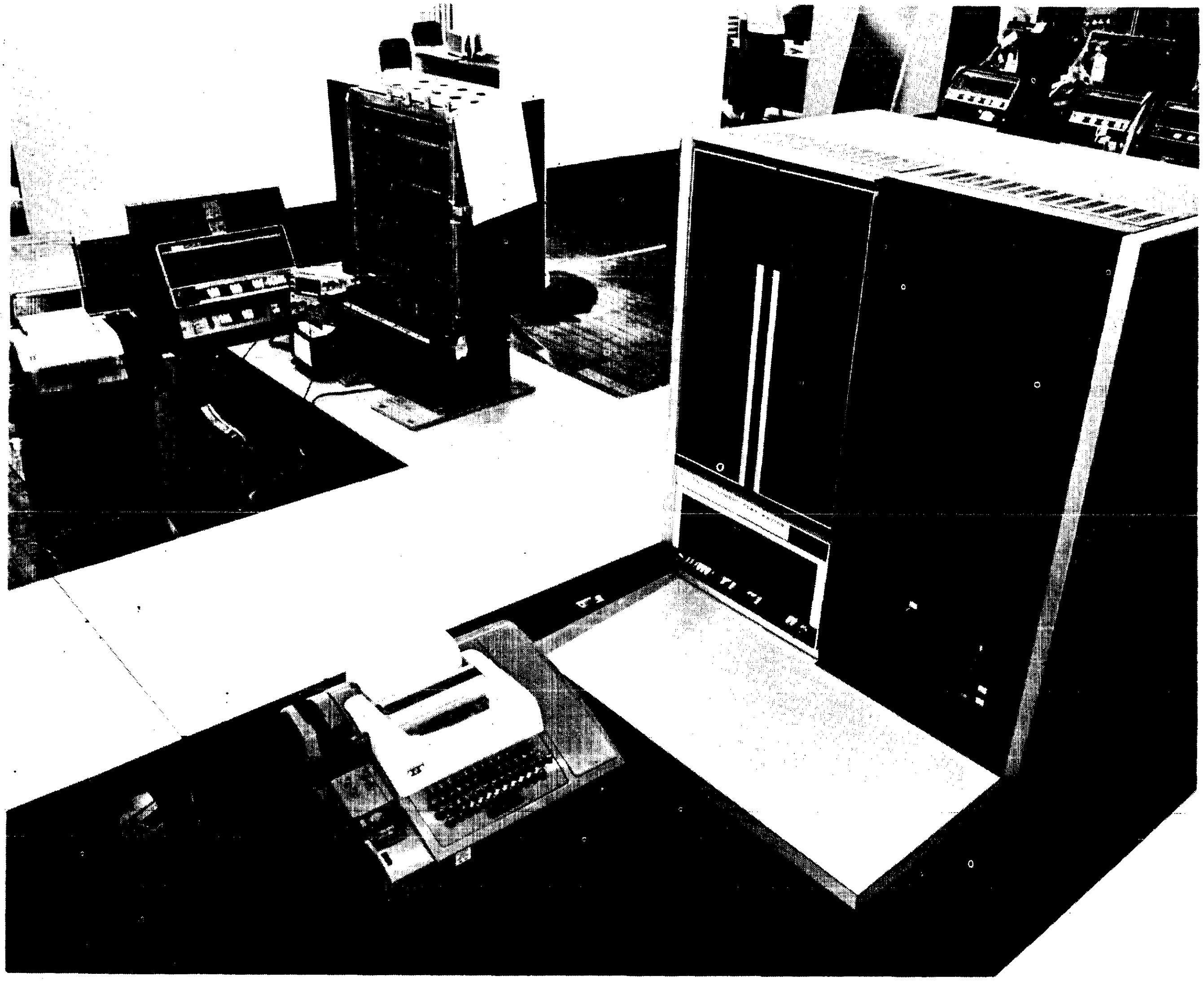


Figure 5/P—Actual Phase 2 environmental test system

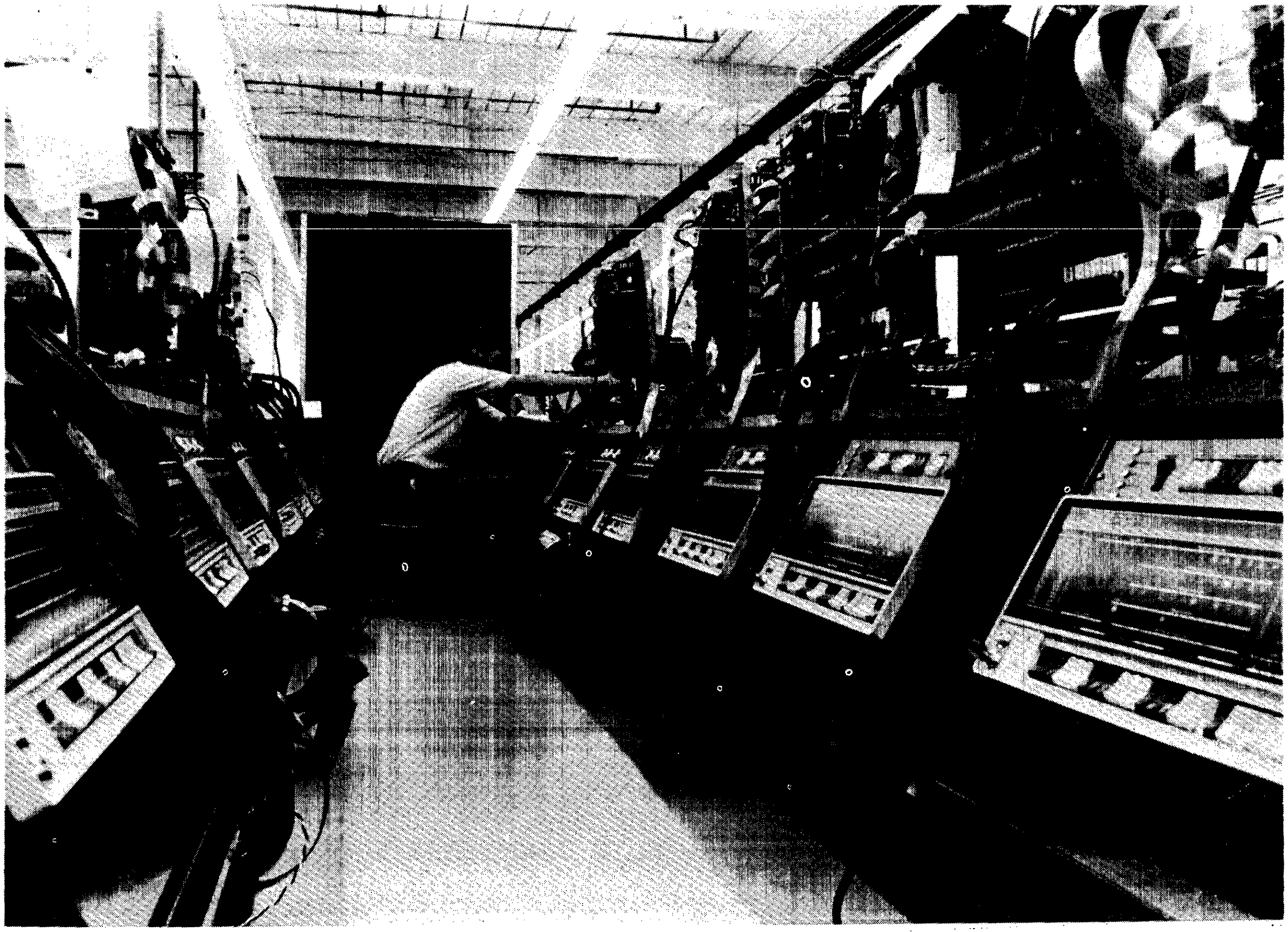


Figure 5/PA—Actual Phase 2 auto-acceptance test system

tem Controller will output the error information on a printer as well as the operators console. The controller has the ability to re-try running that program to gather further information as to whether is is a transient or a repetitive failure. If the error is catastrophic, the station will be disconnected from further service. If no errors have developed after a specified length of time, the next diagnostic program is loaded and run. After all diagnostic programs have been run successfully, the controller will type an accept message for that station and start the routine over again. After a specified number of hours of error free operation, the computer under test is ready to be shipped. The auto accept mode of operation does not require any operator intervention.

CONCLUSIONS

The proposed system provides a very effective tool for use in debugging new computer central processors at a production facility. Any malfunction that may exist within most computer control processors can be detected and found by using this system. In general these problems will be found easier when using the test system than when running a diagnostic program in a free standing machine.

While many of the details of this system are oriented towards computers produced by Digital Equipment Corporation, I believe that the basic idea can be adapted to a wide variety of computers, produced by many manufacturers.

The advantages of using a computer test system for checkout on a computer production line are fast turn on time, less experienced test technicians, resulting in greater product reliability at lower cost.