HEWLETT-PACKARD JOURNAL

AUGUST 1969

Automated Testing

What is automated testing? What does it buy us? How is it evolving?

By Robert A. Grimm

AUTOMATIC TESTING OF ELECTRONIC DEVICES is one of the fastest growing areas of electronic instrumentation. It is the way an increasing amount of our testing is done every day, and it is the way most of our testing will be done in the future. At present, however, most of the testing of electronic components, sub-assemblies, modules, and systems is still done manually. Many companies are just now installing or considering installing their first automatic test systems. While most people involved in electronic testing have read about or seen examples of automatic test systems, many may not be aware of the fundamentals involved in all automatic testing.

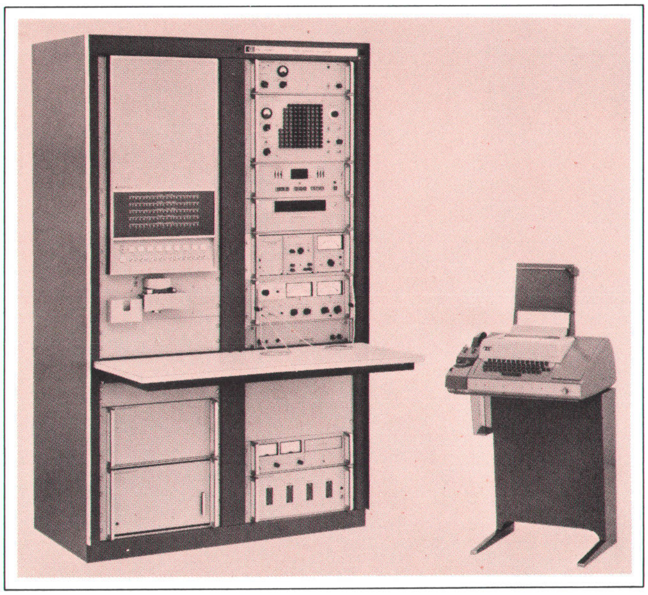


Fig. 1. *HP Model 9500A is a typical computer-controlled, modular, multipurpose test system. It stimulates the unit under test and measures its responses, then processes and records the data. Stimuli available are dc, low-frequency ac (to 100 kHz), and high-frequency ac (to 500 MHz).*

This article and the others in this issue will discuss the fundamentals of automatic testing, its advantages, the equipment used, how programming is done, how to choose or build a system, and what HP automatic test systems are currently being used.

Automatic Versus Manual Testing

Automatic testing of electronic and electrical devices requires the same functional elements as manual testing. These are:

- Stimulation of the unit under test
- Measurement of the unit's responses
- Switching
- Control
- Evaluation of data
- Recording of results

In automatic testing as in manual testing, the stimulus and measurement functions are performed with instruments. Stimuli include oscillators, signal generators, power supplies, and so on. Measurement devices include voltmeters, counters, power meters, and the like.

Switching, control, evaluation, and recording in manual testing are usually performed by the operator. The switching of connections between the test equipment and the unit under test is done by reconnecting cables, by using test probes, or by means of a manually operated, pre-wired switch box.

Control of the test equipment is exercised by the operator, by setting knobs on the panels of the test equipment. Control of the test sequence is also by the operator, usually following a written test sequence.

Evaluation of test results is again done by the operator, who compares the results to pre-established limits or uses his judgment. He may make some calculations or use graphs or charts. Recording of manual testing results

PRINTED IN U.S.A.

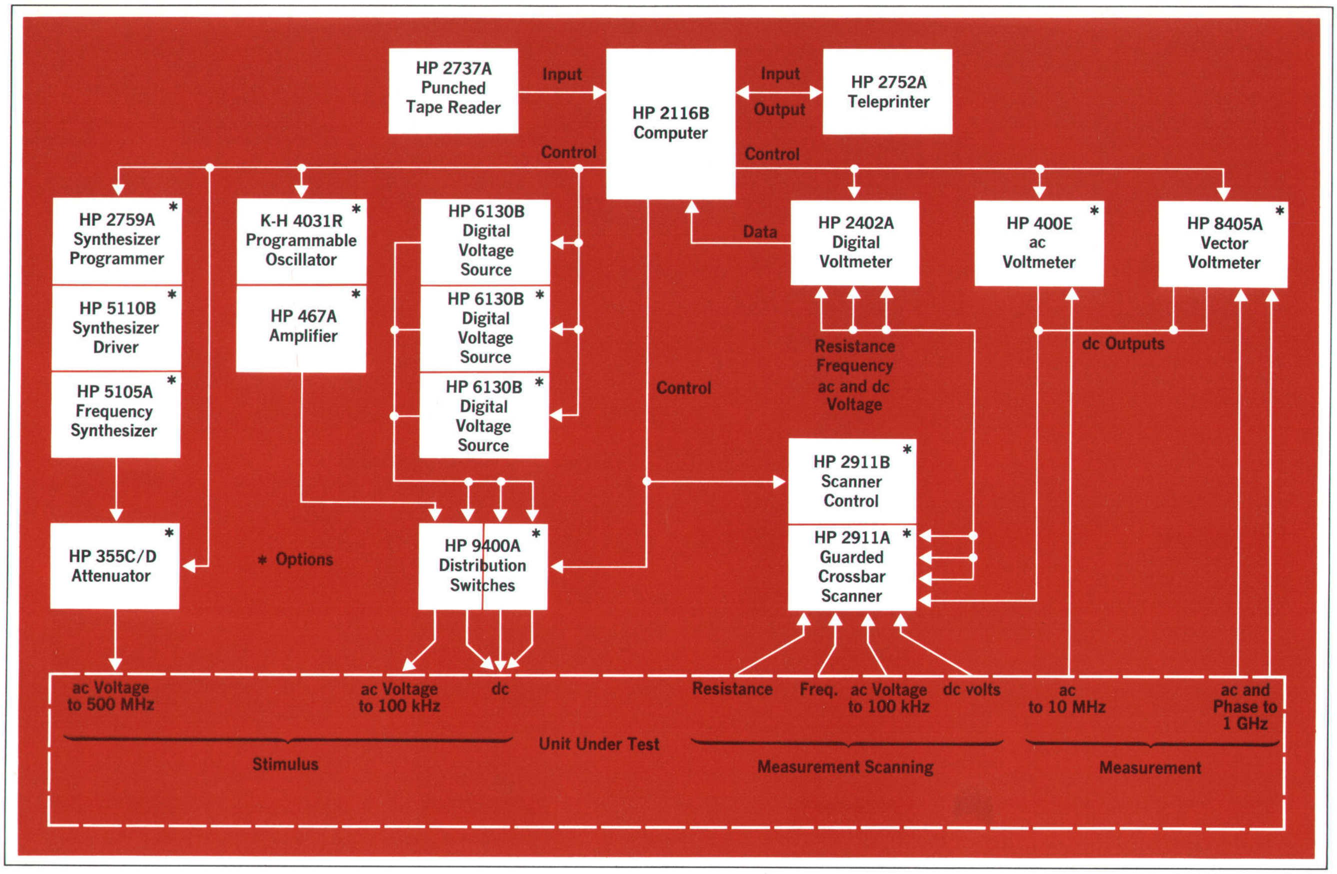

HP 2737A Punched Tape Reader
Input
HP 2116B Computer
Input
Output
HP 2752A Teleprinter
Control
Control
HP 2759A Synthesizer Programmer *
HP 5110B Synthesizer Driver *
HP 5105A Frequency Synthesizer *
K-H 4031R Programmable Oscillator *
HP 467A Amplifier *
HP 6130B Digital Voltage Source
HP 6130B Digital Voltage Source *
HP 6130B Digital Voltage Source *
Data
HP 2402A Digital Voltmeter
HP 400E ac Voltmeter *
HP 8405A Vector Voltmeter *
Resistance Frequency ac and dc Voltage
dc Outputs
Control
HP 2911B Scanner Control *
HP 2911A Guarded Crossbar Scanner *
HP 355C/D Attenuator *
* Options
HP 9400A Distribution Switches *
ac Voltage to 500 MHz
ac Voltage to 100 kHz
dc
Resistance
Freq.
ac Voltage to 100 kHz
dc volts
ac to 10 MHz
ac and Phase to 1 GHz
Stimulus
Unit Under Test
Measurement Scanning
Measurement

Fig. 2. *Model 9500A is a good example of the use of standard instruments and computer interfaces to make a system adaptable and expandable. Many automatic test systems are now being made this way.*

is ordinarily by the operator's filling in a test card.

In automatic test systems, like the typical example shown in Figs. 1 and 2, human intervention is minimized, although it is sometimes not completely eliminated. Most of the test functions are done automatically under control of a computer or other device following a stored program sequence. Switching is done automatically by electrically controllable switches, such as relays, crossbar switches, and switch matrices.

The recording of test results is accomplished in an automatic system using a teleprinter, a digital printer, or punched or magnetic tape. These devices usually record not only the measured data, but often the entire test card, including parameter names, fixed data, and indications of out-of-tolerance results. Some types of output reports are best presented in the form of graphs, showing two or more parameters plotted against one another. X-Y recorders can be used for these reports; they can also print titles and scales on the graphs.

The control function in automatic testing is performed by the program stored in the computer. The program controls test sequences, voltages from power supplies, frequencies from signal generators, pulse widths and repetition rates of pulse generators, and so on. For measurements, the program controls amplitude ranges of voltmeters, function selection on multimeters, gate times of counters, and so on. The computer also controls the switches that route signals to and from the unit under test.

Evaluating results in testing is primarily a matter of

Cover: *HP Model 9213A is a general-purpose computer-controlled, dc-to-12.4-GHz automatic calibration system for laboratory-type instrumentation. For typical instruments it is six or more times faster than manual methods. For example, checking and calibrating the HP Model 3440A Digital Voltmeter takes less than five minutes using the 9213A, but thirty minutes manually.*

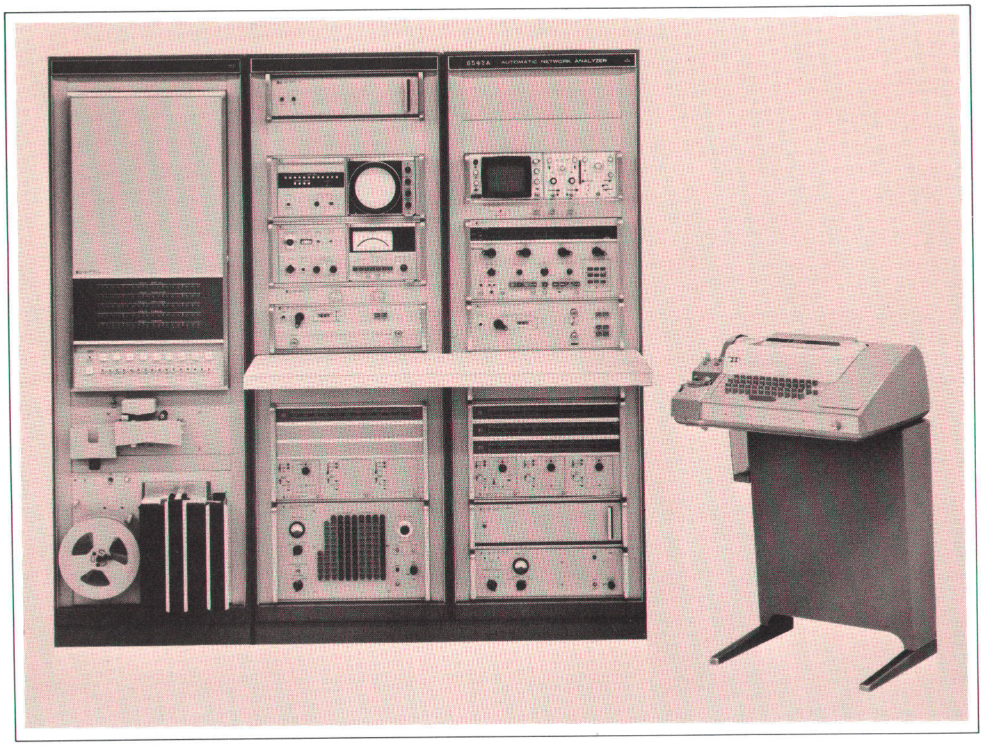


Fig. 3. *HP Model 8542A Automatic Network Analyzer is an example of a standard system designed for a special purpose, in this case microwave network analysis. It is also a good example of the use of a computer to remove systematic errors from measurements, thereby achieving standards-laboratory accuracy in fast, automatic, swept-frequency measurements. The system makes measurements over a frequency range of 0.11 to 18 GHz on such devices as transistors, filters, amplifiers, isolators, and any other networks that can be characterized in terms of their scattering parameters.*

comparing test results against established limits. However, some calculation or manipulation of data may be required before this comparison can be made. For example, if the linearity specification of a dc amplifier is stated as 'within ±0.01% of the best straight line through all data points,' this line must first be calculated and the errors determined, and then a comparison made against the limits. Although this is laborious if done manually, it is straightforward for a computer.

Frequently in testing, the results of a test are used to determine what test should be performed next. This is called 'branching'. At branch points in a program, a system might print out special test results or messages, or jump to a subroutine for fault locating. This is a powerful capability of computer-controlled testing.

To summarize, automatic testing differs from manual testing in how the switching, control, evaluation, and recording are done. The actual testing—the application of stimuli and the measurement of results—is essentially the same in automatic testing as in manual testing. This means that to be able to test a device automatically one must first know how to test it manually; this is especially true for the engineer or technician who is writing the test program.

Why Automated Testing?

Testing started to become automatic when the devices to be tested became too complex, and the required tests too numerous, to be done economically by older testing methods. The testing of these devices called for fast, automatic equipment which could stimulate these devices and measure their outputs according to their complex specifications. Manual methods of testing, while satisfactorily meeting the requirements for sophisticated testing in most cases, tended to be too time-consuming when the quantities of devices to be tested or the number of tests to be performed made it difficult to meet schedules.

The advantages of automatic testing include the following:

- Speed
- Repeatability
- More comprehensive testing
- Self-test and error correction
- Operators can be less skilled.

Computers operate at very high speeds, with individual program steps taking typically less than 2 microseconds. Even though it may take a large number of these steps to perform a control or conversion operation, such operations can often be completed in a few microseconds, or at most a few milliseconds. Electrically controlled, many kinds of test equipment can provide signals and take measurements much faster than an operator can set the controls or record the readings. A computer can take advantage of this speed. Also, the manual reconnecting of cables between tests is a slow process compared to the speeds at which relays operate. All of this means that testing is much faster under computer control. Digital testing, that is, testing of binary logic patterns, is one of the fastest kinds; thousands of tests per second can be done easily.

Case histories of organizations who use automated testing show that non-digital testing can be anywhere from 5 to 60 times faster than manual testing, and digital testing can be faster yet. It's reasonable to expect that most automatic testing will run at least ten times faster than manual testing.

There are some steps in testing that may be difficult or impractical to automate or to speed up. These include steps that the operator must do, such as connection of a device to the system, a change of control settings on the device being tested, or screwdriver adjustments on the device. An engineer designing a new device might well anticipate how it is to be tested, and consider providing for automated adjustment of its controls.

Repeatability is much better when tests are automated. Tests run automatically are done the same way each time, faithfully following the same test sequence. Steps are not skipped. Stimulation of the device under test and the transfer of data from the measuring devices into the computer are done the same way each time, without parallax errors in meter readings, transposition of digits in digital readings, or other human errors.

With automated testing, more comprehensive testing is practical. In manual testing, spot checking of parameters is often necessary to speed up the tests. For example, the gain linearity of an amplifier may be measured and calculated at as few as three or five amplitudes. Or the standing wave ratio of a connector may be checked only at a few selected frequencies. Or the truth table of a digital integrated circuit may be checked with only one voltage input level. Automatic testing can be much more thorough because it is so much faster. It is even easy to do *additional* testing to detect marginal devices which may fail only under certain conditions. The programming to do this usually requires very few, if any, additional statements; for example, it is easy to repeat a test with a changed condition. By knowing completely what his product will do under very many conditions, a manufacturer can give more complete specifications, and can specify the product's performance closer to the tested values, with less allowance for unknown factors.

Self-test and error correction together make up one of the most significant advantages of automatic testing over manual testing. In any test setup, certain errors are introduced into the measurements by cables, connectors, or equipment. Some of these are random errors which vary from test to test; these errors may be caused by noise or other non-repeating parameters, and they are difficult to eliminate, although averaging techniques are sometimes helpful. However, many causes of error are repeatable. These include such things as voltage drops in cables, reflections in connectors, zero offset or full-scale error on a voltmeter, frequency calibration error of an oscillator, and so on. In an automatic test system it is both possible and practical to have the system make measurements based on internal standards to determine these systematic errors, then store the correction data and correct all subsequent measurements before using or displaying them. This capability is used extensively in the HP 8542A Network Analysis System (Fig. 3) to eliminate reflection coefficient measurement errors caused by connector and cable mismatches, coupler directivity, sampling head efficiency, and so on. The result is accuracy approaching that of a standards laboratory, but obtained at a much higher speed.

Automated testing demands less skill on the part of the test station operator, one reason being that the test procedure is read into or stored in the computer's memory, so the operator does not have to know or follow a complex test procedure. Simple instructions can be typed out

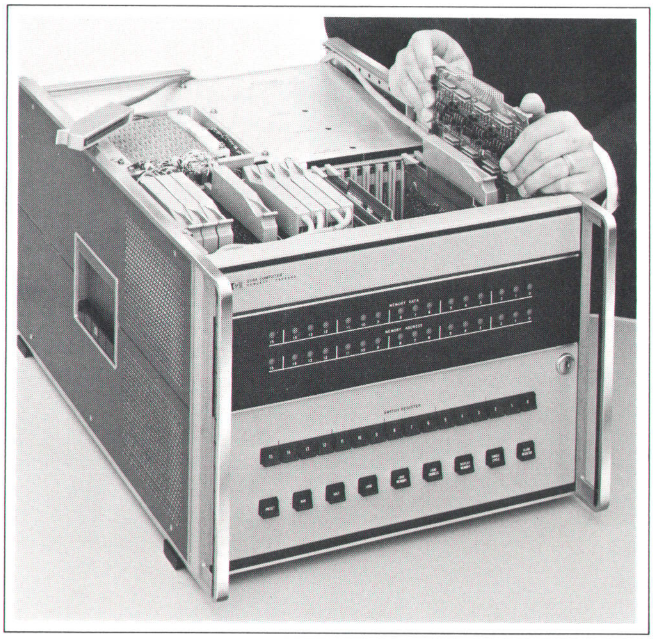


Fig. 4. *Small general-purpose digital computers are being used more and more as controllers in automatic test systems because of the flexibility and versatility they offer. HP Models 2114A (shown here), 2115A, and 2116B are designed to accept interface cards which allow them to communicate with signal sources, measuring instruments, and peripherals.*

automatically to tell the operator what action he should take and what the results of the test are. In testing simple devices, such as electronic components or simple modules, only a go or no-go indication may be required, and this can be in the form of lights. With an automatic handler, good and bad devices can be ejected into appropriate bins. In testing integrated circuits while they are still on the substrate slices, bad units can be marked automatically, perhaps with a dot of ink. In all of these cases, the operator needs to be competent to conduct the test, but doesn't necessarily have to understand it.

There are many other advantages of automated testing besides those discussed here. For example, a computer can do statistical analysis of test results as the data are gathered, and can call attention to recurring problems very quickly. With a computer in a test station, there is no waiting for results to be resolved and tabulated offline at some later time.

Automatic Testing Comes of Age

Much of the early work in automatic testing was done for and by various branches of the armed services, who recognized the long-term potential of the technique. These latter-day pioneers contributed much to helping automatic testing achieve its present practical status.

Automatic test systems have not always been so practical as they are today. For example, in the early days there was a shortage of equipment suitable for systems use. Most of the test equipment available was designed for manual control and visual readout of results. Therefore, automatic systems had to be assembled using custom-built, and consequently expensive, equipment.

Now, however, many standard instruments are designed so they can be remotely programmed, and systems are being assembled using these instruments. This approach not only lowers initial costs, but also can be expected to enhance reliability and maintainability, since standard instruments are usually well debugged and easily serviced. This approach also makes modular systems possible, so a user can have a system assembled to fit only his immediate needs, and yet be able to expand or modify it as his needs change.

Another problem in the early days was that even the smallest computers were relatively large and expensive by today's standards, so small automatic test systems were impractical. The first automatic systems used paper-tape readers to control the test equipment, and hardware limit comparison to check results. Arithmetic capability was very limited and branching to other tests was awkward. Interfacing between the computers and the test equipment was also difficult because everything had been designed separately. Interface boxes often had to be designed to accommodate the differences in logic levels and timing.

Today there are many small general-purpose computers, costing between $10,000 and $25,000. It has become economically practical to use computers almost exclusively as automatic test system controllers. Other controlling devices are expensive by comparison when the many capabilities of computers are considered. Progress is also beginning to be made in standardizing logic levels, codes, and other programming parameters, and prepared interfaces are often available. HP computers accept plug-in interface cards (see Fig. 4); there are standard cards for standard instruments, and custom cards can be manufactured.

Programming of automatic test systems used to be difficult because programs had to be written in tedious machine language. Engineers and technicians who knew what tests had to be performed had to describe these tests to programmers, who would then write the test program. Often neither group understood or appreciated the technical problems, the possible alternatives, or the language of the other. This also complicated debugging and changes, and made software preparation very expensive.

Things are different now. BASIC, AuTest, and other simple, conversational, English-like languages are available. Programs are easily modified and can be written by the test engineer or technician.

Robert A. Grimm

Bob Grimm received BS and MS degrees in electrical engineering from Purdue University in 1950 and 1951. He has had extensive experience in instrumentation, especially in automatic test systems, and he has written a number of papers and articles on automatic test systems. At HP since 1951, Bob has developed oscilloscopes, TV frequency monitors, and signal generators, has been general manager of the HP Dymec Division which manufactured data acquisition systems and computers, and has been manager of corporate systems engineering. He is now marketing manager of the HP Systems Division which produces computer-controlled automatic test systems.

Bob is a senior member of IEEE, a member of Eta Kappa Nu, Tau Beta Pi and Sigma Xi, and a registered professional engineer in California. He devotes much of his spare time to United Fund activities.

Choosing An Automatic Test System

When is an automatic test system justified?
How do you decide on the configuration?

By M. D. Ewy and Stephen C. Shank

WHETHER A TEST STATION SHOULD BE AUTOMATIC, and how automatic it should be, depends upon how it will be used. Selection of a test station should begin with an estimate of what tests are needed and what quantities of units must be tested. These must then be compared with what is possible and what is economically and humanly feasible. There are many choices and tradeoffs; some of the most important ones will be discussed in this article.

Once a manufacturer has an idea of what kind of test system he needs, the best place to get it is usually a systems manufacturer. He should discuss his requirements with a systems manufacturer before deciding on a system. Standard systems, whether special-purpose or multipurpose, are often the best buys, but if these won't do the job, a custom system can be ordered.

If a manufacturer has manpower available with the capability of designing whatever special software and hardware are needed, he may find it most economical to build his own system, using commercially available computers, programmable instruments, and prepared interfaces, wherever possible. Some of the considerations that go into the design of such a system are discussed in the article on page 11.

Man-Machine Tradeoffs

Any test station must be able to perform the following five functions.

- Interpretation of the test procedure
- Equipment setup
- Data acquisition
- Data processing
- Decision making

Both men and machines can do all these things, so both meet the requirements for test stations. There are obviously tradeoffs between the two. A machine is fast, and it can be programmed to repeat tests reliably. A man can learn and adapt, and can interpret complex indicators.

Automatic testing is most appropriate for situations which call for many tests per unit (such as digital logic), or where volume is very high even though few tests per unit are needed. On the other hand, automatic testing is difficult to use when a unit to be tested requires elaborate setup or complex adjustments.

Troubleshooting is still fairly difficult to automate because of the need of a large memory to store all possible troubleshooting trees and because of the difficulty of writing an executive troubleshooting program. However, pretesting *subunits* automatically, before assembly, can often significantly reduce the amount of troubleshooting needed on the assembled instrument or system. This can result in a reduction of total test time, since the time required to troubleshoot an assembled system is usually far greater than the time required to pretest and debug parts before they are assembled.

In general, with the industry's trend to greater complexity, testing can be expected to keep pace, and one way will be greater automation. Where it can be used, automatic testing provides reliability in the test process, a means to reduce troubleshooting by pretesting, and better warranty reports through additional and more thorough testing.

Types of Test Stations

Test stations fall into two basic categories:

- Product-oriented or special-purpose
- Multipurpose (sometimes called universal or general purpose).

A product-oriented station is designed to test a specific

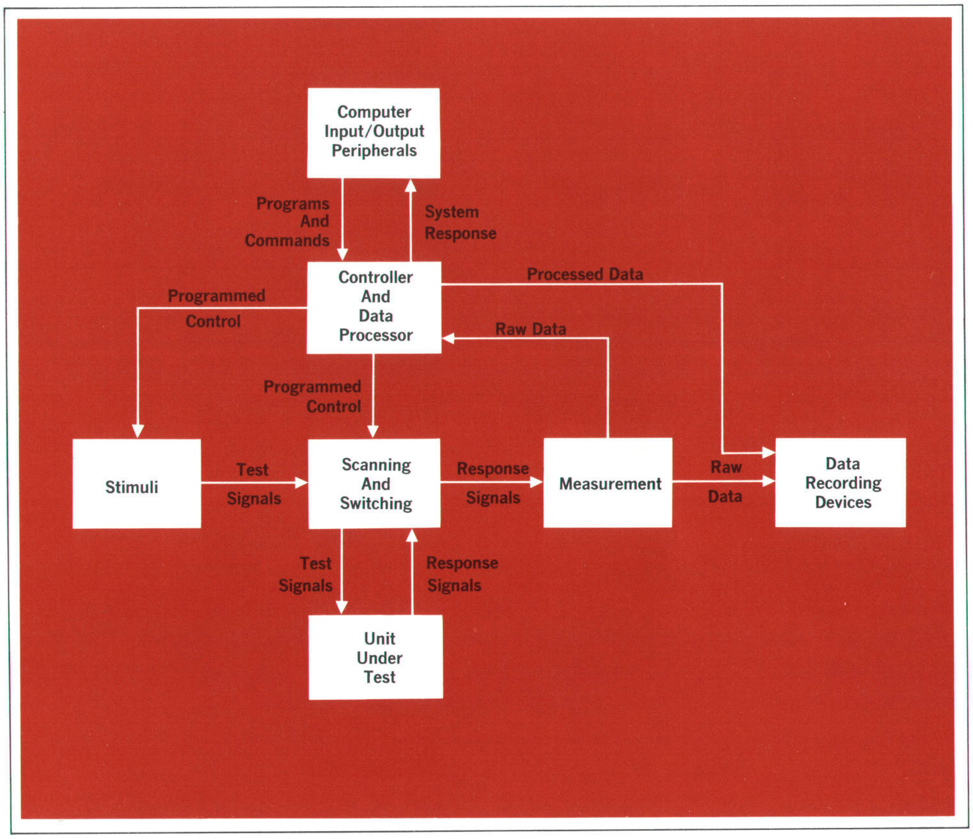


Fig. 1. *This block diagram would fit many modular multipurpose test systems. Such a system can be further divided into subsystems, such as a dc subsystem and a low-frequency ac subsystem, and each subsystem would have separate capabilities for stimulus, switching, and measurement.*

product, while a multipurpose test station has the capability of testing many products without significant modification of the test station.

A major factor influencing the design of a test station is product volume. In cases where the device to be tested is high-volume, the test station can reasonably be device-oriented. The design emphasis is on the hardware and the result is typically a very efficient test system.

A manufacturing environment characterized by many low-volume products calls for a multipurpose test station. A multipurpose station is practical where product-oriented test stations would be idle too much of the time. However, a multipurpose test station may require a larger initial investment, and justification of this investment requires detailed analysis.

As a test station becomes more general, it relies more on software. In some cases, this means a sacrifice in speed and efficiency. However, it makes it possible to develop a common in-plant test language and terminology, which in turn makes it easier to generate test procedures for new products. Multipurpose test stations make it easy for the designer to incorporate test objectives into the design of new products.

Multipurpose systems can be designed so that less-skilled personnel are needed to operate them, since the same test approach can be used for many products. Another advantage is that, with a single station able to service different devices, less total equipment and space is tied up.

Fig. 1 shows a concept of a multipurpose test system. Such a system can be further divided into subsystems, such as a dc test subsystem and a low-frequency ac test subsystem. There must be separate capabilities for stimulus, switching, and measurement in each subsystem. The capability needed in each of the subsystems must be defined by analyzing the requirements of the highest-total-test-time products. The test system should provide at least this capability. Greater capability may be insurance against early obsolescence.

A better way to insure against early obsolescence is to have a test system that can be modified easily to keep pace with the test requirements of new products. Whether the system is product-oriented or multipurpose, it should be adaptable to maximize its useful life. One way to make a system adaptable is to make it modular. Modularity can also help reduce the cost of system design and fabrication.

Hardware-Software Tradeoffs

When designing an automatic test system, a common tendency is to use software to perform as many functions as possible. This is not always the best approach. A good example is the HP 2060A (Fig. 2), a system for testing logic circuits, which uses another logic circuit, identical to the one being tested but known to be good, as a test reference. This eliminates the need for generating thousands of limit parameters and storing them in the computer's memory. The system is much easier to program and requires much less computer memory than a system in which the expected responses of the circuit under test are stored in the computer.

If all one needs is to collect and record measurements, the simplicity of a manually controlled or tape-controlled data acquisition system is attractive. However, computers are becoming so inexpensive that many data acquisition systems are now computer-controlled, chiefly because software programming is so much more versatile than manual or tape programming. A computer can not only control the test procedure like a punched-tape controller, but can do it faster and more efficiently, while performing other functions. A computer can:

- Facilitate program changes.
- Do fast branching. A computer doesn't need to search through punched tapes. It goes right to memory for instructions.
- Convert units. For example, it can take a current measurement from a thermocouple, then convert it to temperature.
- Calculate parameters. For example, it can calculate bandwidth from power and frequency measurements.
- Make reports. It can keep failure and analysis data.
- Do self-correction. For example, the computer can measure system errors using a short and a known standard resistance for calibration, then correct all future resistance measurements.
- Diagnose faults. The computer can sometimes be made to compare a problem pattern with prestored patterns and print out the probable cause of the problem.
- Average measurements and give quick trend analysis.

Cost will often be decisive in making hardware-software tradeoffs. A factor that is sometimes overlooked is that the development cost for software is significant, just as it is for hardware.

Operator Interfaces

Another important factor to consider in choosing a test system is how the operator will interact with the system.

Test-Station-to-Operator Software Interface. An appropriate programming language will make it easy for the test engineer to generate the test program. The operator must be able to tell the system to perform the following basic operations.

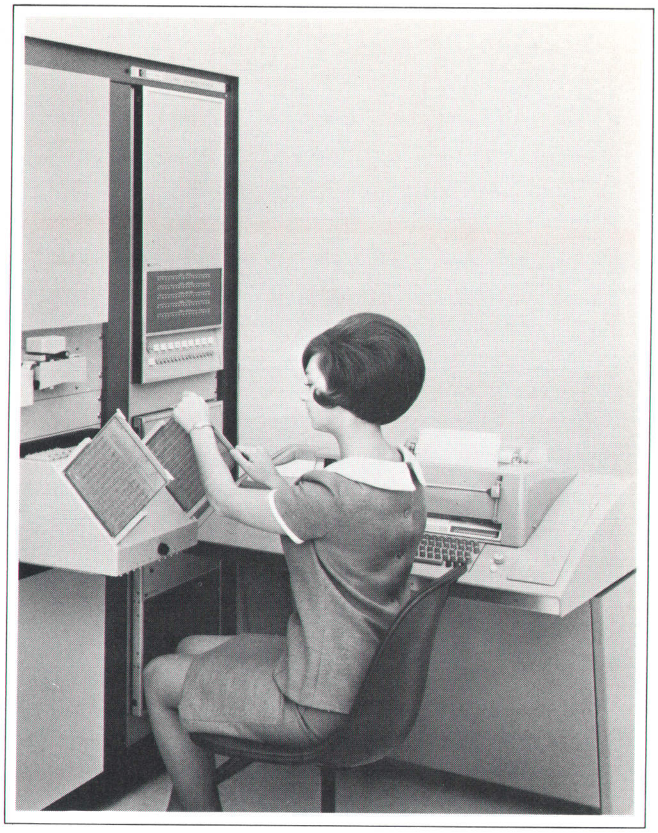


Fig. 2. *Should a system be designed to accomplish a particular task by means of software — i.e., programming — or hardware? HP Model 2060A Digital Logic Module Test System exemplifies this tradeoff. Instead of requiring the programmer to generate thousands of expected responses of the module under test, the system uses an identical known-good module as a reference and compares the responses of the two modules. It thereby saves programming time and memory space.*

- Program instruments. For example, set a digital voltmeter to the desired function, such as dc volts, or cause a switch matrix to connect power, stimulus or measurement devices to the proper pins on the unit under test.
- Collect status information and measurement data— for example, power supply overload status, frequency measurements from a counter, or voltage measurements from a digital voltmeter.
- Provide system timing. For example, while a power supply settles, wait 1 millisecond before taking a measurement; after programming a relay, wait 10 milliseconds for it to close.

- Perform logic and arithmetic operations, such as comparing status to the norm, linearizing data, or comparing against limits.
- Perform program branching. For example, if a measurement is out of limits, take appropriate action.
- Output formatted data, that is, type out a test report.

The test language should allow convenient communication for all these operations. It should be at least a pseudo-English language, so that it can provide easily understood documentation. The language should include convenient means for making changes to the test program.

Test-Station-to-Operator Hardware Interface. This interface allows on-line control of the test station by an operator. There are two approaches to the design of this interface.

- Character-serial input/output, e.g., that provided by a typewriter.
- Functional pushbuttons and lights, e.g., that provided by a control panel.

The basic communication peripheral for small computers is a teleprinter. As an operator interface it is convenient from a design standpoint, and inexpensive. A teleprinter provides a very versatile but somewhat cumbersome test station control capability. Using the teleprinter's alphanumeric input capability, one can modify test programs on-line. On the other hand, several characters or words must be typed to give a command to the test system.

A hardware control panel, although less versatile, can be easier to understand and more foolproof. The design can be human-engineered with the task of electronic testing in mind. An example of such a panel is shown in Fig. 3. It contains system control pushbuttons, lights to show system status and test results, buttons for entering numbers and other data, and displays to show numerical data. A simpler, but less versatile data input capability could be provided by thumbwheels.

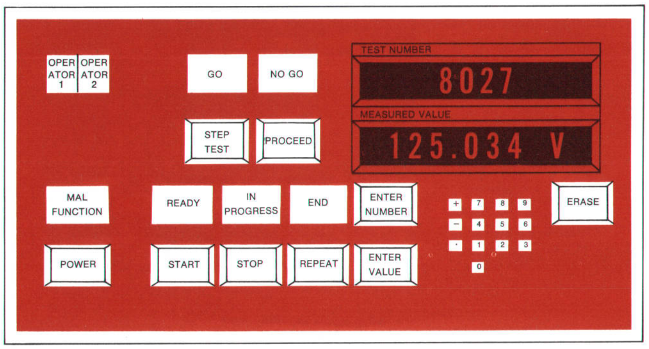


Fig. 3. *One thing to consider when deciding on the form of a test system is how the operator will control the system. A teleprinter is a very versatile method which is used in most multipurpose systems. A control panel like this one is less versatile but easier to understand and more foolproof.*

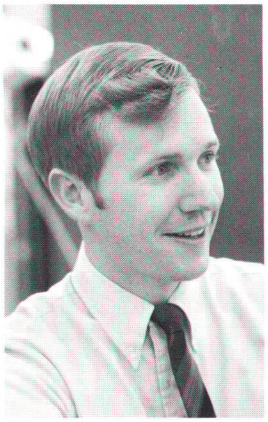
Data output for permanent records and test reports can be provided by various standard data-handling computer peripherals, such as punched or magnetic tape. Recorded test results in permanent computer-compatible form might be retained for future statistical analysis.

Other Considerations

Questions about operator training, warranty and service, installation and checkout, and so on, are also important to the prospective owner of an automatic test system. Systems manufacturers vary in their policies on these points. A user who puts his own system together will do these things himself to fit his requirements.

M. D. Ewy

Dale Ewy has been building automatic systems since he came to HP in 1965. He is now development engineering manager of the HP Systems Division. After receiving his BS degree in physics from Oklahoma State University in 1954, and his MS degree in applied physics from UCLA in 1956, Dale worked on lasers, frequency standards, and electro-optical instrumentation before taking up his present specialty. He is a member of Phi Kappa Phi, and has taught basic physics. When he wants to relax, he often does it by hiking and camping.



Stephen C. Shank

Steve Shank is a Purdue graduate (BSEE, 1965) and has an MSEE from Stanford (1967) which he earned on the HP Honors Cooperative Program. He joined HP in 1965 and took part in the initial design of the 2116A Computer. He also worked on the design of the 2760A Optical Card Reader. Recently Steve completed a study on test processes which led to the design of a semi-automatic test station for the 432A Power Meter. Steve is a member of Eta Kappa Nu. He is presently studying classical guitar. Other hobbies include sailing and tennis.

Building An Automatic Test System

What are the things you have to think about when putting a system together?

By M. D. Ewy

BUILDING AN AUTOMATIC TEST SYSTEM poses problems in both hardware and software design. This is true whether the system is specially built or put together using off-the-shelf subsystems.

The type of system that is required depends on a number of factors, such as the kind of testing to be done, the kind of units to be tested, the manufacturer's total test needs, and so on. These are discussed in the preceding article. This article will deal with the things that must be done after the basic form of the system has been decided upon. Specific points that require attention are:

- Choice of instruments, computer, etc.
- Specifications
- Interface design, including logic levels, codes, noise rejection, ground isolation, data-transfer control and timing
- Switching
- System timing
- Software
- Checkout

Standard Instruments and Computer, or Special?

There are numerous advantages to using standard, off-the-shelf instruments in an automatic test system, even if they seem overdesigned for the particular jobs they will be doing in the system. For one thing, the cost of specially designing 'optimum' instruments is saved. For another, standard instruments have usually been well tested and specified, and should be more reliable and easier to maintain. The importance of reliability and maintainability in individual units emerges clearly when one considers that the time between *system* failures is going to be much shorter than the time between failures of any one instrument. Specially designed equipment can rarely be as well tested as standard equipment, because it isn't economical to do a thorough job on low-volume equipment.

Now that relatively inexpensive general-purpose digital computers are available, they are being used more and more as controllers in automatic test systems. Their attractiveness stems in large part from the versatility of software programming. However, it is well to use one that is designed for easy interfacing with instruments.

Specifications

The maximum measurement capabilities—i.e., accuracies, rates, etc.—of a system may not be as great as those of the individual instruments. For example, a digital voltmeter may not measure the voltage at the other end of a system cable as accurately as it can measure the voltage at its own terminals. However, with a computer in the system it may be possible to have the system measure itself and correct for such systematic errors as voltage offsets in cables. See the article on page 2 for more about self test and error correction.

Interface Design

Given a typical computer and several instruments selected at random, a designer can have difficult interface problems. To allow the computer and the instruments to communicate, he may have to design and debug a great

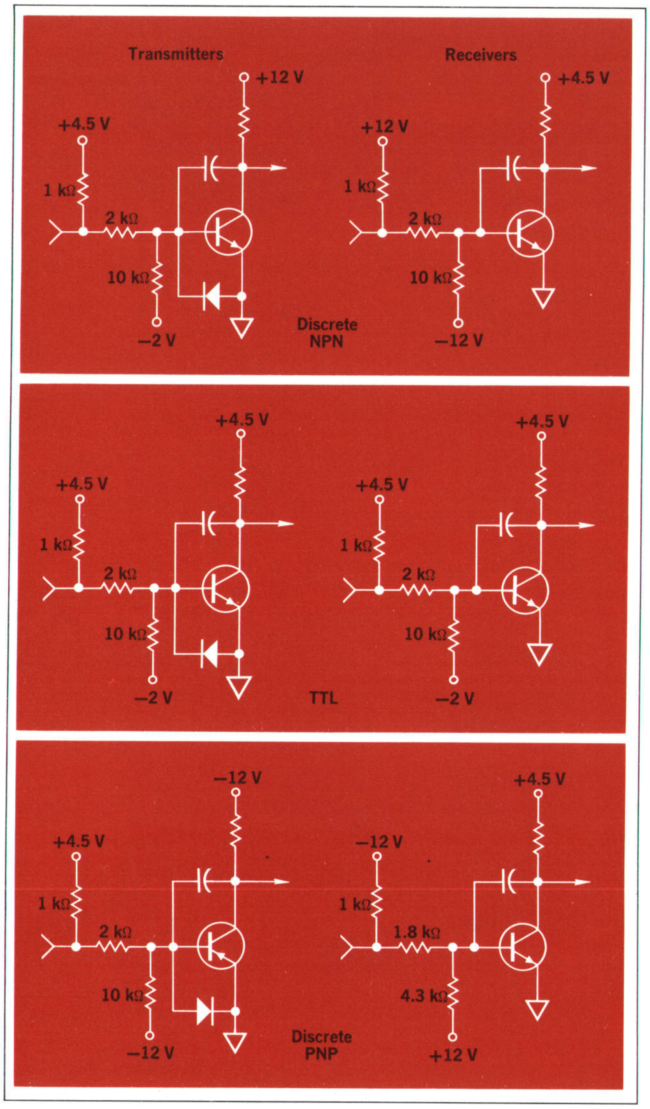


Fig. 1. *Interface design is an important part of the design of an automatic test system. An interface must translate the computer's logic levels and codes into a form which will control an instrument. Different instruments need different interface circuits. Here are some of the more common ones.*

deal of special hardware. Some of the considerations are discussed in the following paragraphs.

Logic Levels. Many different logic levels are used in computers and instruments, especially in instruments designed with discrete circuits. Recently, because of the availability and low cost of TTL and DTL current-sinking integrated-circuit logic, these types of circuits are becoming *de facto* interface standards for systems that don't have to contend with much external noise.

HP has several computer I/O cards that can be loaded with appropriate components to interface to instruments that require different logic levels. Fig. 1 shows typical transmitter and receiver circuits for current-sinking IC logic, discrete NPN circuits, and discrete PNP circuits.

Codes. Codes are the meanings assigned to the logic levels on an interface line or a combination of interface lines. Any time the code of a sending device differs from the code of the receiving device, an interface must be designed to translate.

There is a possibility of great confusion in this area. 8421 code is now standard for new instruments using a binary-coded-decimal format. But there are still differences as to whether HI = 1 or LO = 1. Also, some instruments use binary code, not binary-coded-decimal, and there are two's complement binary, magnitude-and-sign binary, and so on. This is only for numerical data transmission; there are other codes for letters and symbols, and there are different system-command codes—for example, does HI = 'Encode' or LO = 'Encode'? Efforts are being made to standardize these things, but much remains to be done.

Noise and Interface Power Levels. Higher power levels often have to be used in interface circuits to improve the rejection of noise induced from sources such as switched power supply lines, tape punches, relays, and so on.

Standard IC logic gates operate at a power level of a few milliwatts. Loaded logic gates or IC line drivers operate at about 50 to 100 mW power levels. Some HP interface circuits designed for operating over long interface lines operate at a power level of a few watts. One must decide what is needed in a particular application, then design accordingly.

Interface Circuit Ground Isolation. If long interface cable runs are needed, then good ground isolation between the instruments on each end of the cable may be needed to eliminate voltage offsets caused by ground-loop currents. Transformers and pulse logic are sometimes used to provide ground isolation. However, this requires time-dependent logic, which is undesirably complicated for nonsynchronous peripheral devices—which most instrumentation peripherals are. It's usually better to provide ground isolation in a low-pass or dc-coupled transmission system. This can be done using a mod-demod transformer technique or photon-coupled isolators. The latter technique is used in some recent HP equipment.

Cable Design for Noise Rejection. Electrostatic and electromagnetic coupling of noise into signal lines in the interface cables can be minimized by proper cable design. If the required transmission rates do not exceed about

500,000 bits per second for long cables (several hundred feet), then the most reliable cable design is an individually shielded twisted pair for each signal. The shield is connected at the receiver end of the cable only. For short cables entirely contained within a system cabinet, these precautions are not normally necessary.

Fig. 2 is a simplified circuit diagram illustrating the interface and cable design techniques just described.

Data-Transfer Control and Timing. The computer must communicate with all of the other equipment in an automatic test system and must distinguish one instrument from another. Therefore, there must be an input-output data bus and an addressing capability in the interface system. This concept is shown in Fig. 3. Since each peripheral or instrument has a unique address, the address-decoding circuitry allows data transfer through only one data-enable circuit at a time.

When programming an instrument (putting a voltmeter on the proper range, for example), there must be hardware memory in the interface somewhere outside the data bus, so the program parameters can be remembered while the computer is servicing other instruments. This program storage is also shown in Fig. 3.

In HP computers the data bus is in the computer backplane and the address-decoding and data-enable circuits are on plug-in interface cards. Program storage is either in the instruments or on the interface cards.

A common way to control the transmission of data across an interface is to use a strobe line that tells the receiver when to look at the data. The strobe pulse occurs after new data have settled on the interface lines. This is a useful and reliable technique for high-speed data transmission. An even more reliable technique, which is somewhat slower but more than fast enough for most instrument applications, is to use another line, called a flag line. The receiver uses the flag line to signal back to the transmitter that the strobe command (which can now be a single transition instead of a pulse) has been received and acted upon. Using this technique, the system designer need not concern himself with the bandpass characteristics (speed of response) of the transmitter or receiver. The interface will 'handshake' using the strobe line and the flag line, and will thereby communicate reliably at a rate determined by the combined response times of the transmitter and the receiver.

Switching

A computer-controlled automatic test system may contain many different types of switches. For example, one type of relay might be used to distribute and apply power; another type might be used to distribute dc or low-frequency stimuli to a unit under test; high-voltage switching might require another type; measurement instruments like digital voltmeters require 'scanning' or 'monitoring' switch functions; and still another type of relay is required

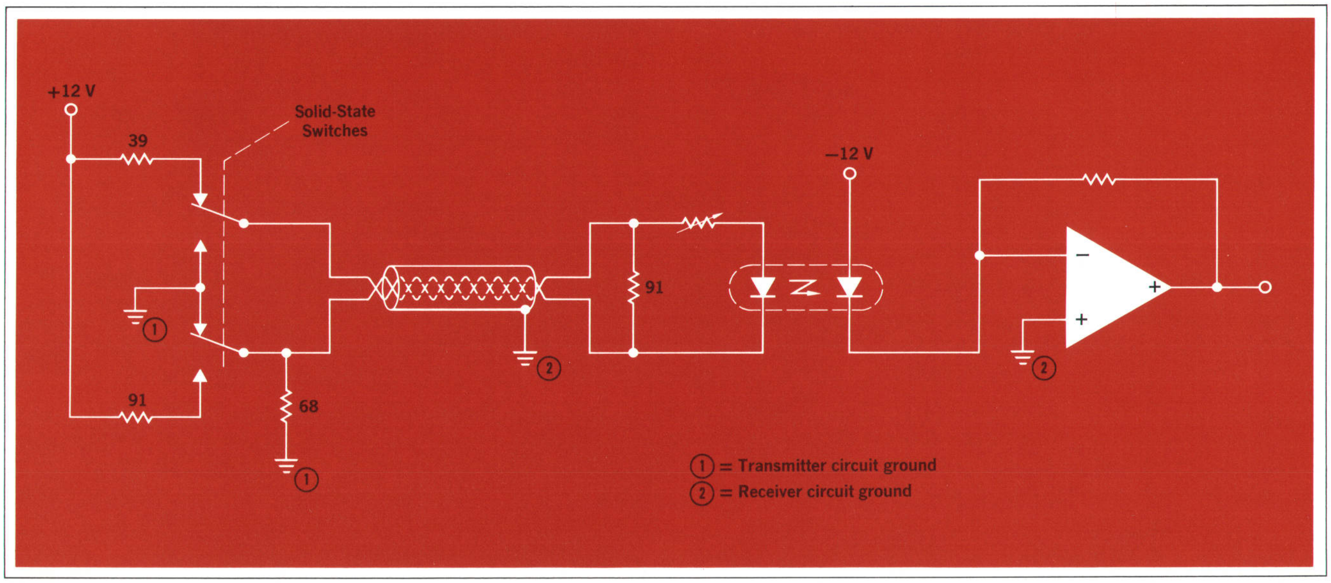


Fig. 2. *Noise and ground-loop currents can be minimized by proper interface and cable design. An individually shielded twisted pair for each signal effectively reduces noise pickup in cables. Photon-coupled isolators provide good ground isolation to break ground-loop currents.*

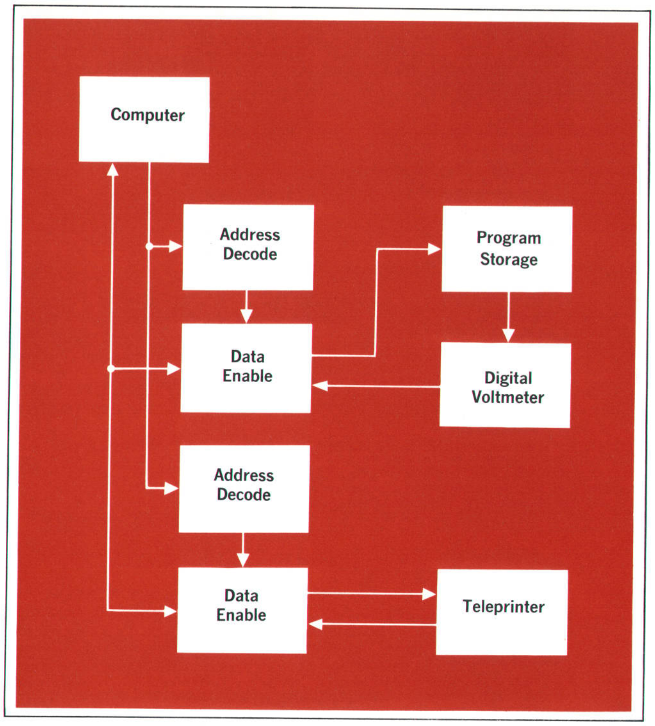


Fig. 3. *Data transfer between the computer and an instrument or peripheral requires a data bus and an addressing capability. Instrument control requires program storage, so the computer can do other things after telling the instrument what to do. In HP computers, the data bus is in the computer backplane and the other functions are on plug-in interface cards.*

to switch high-frequency signals. Providing this variety of switching in a system is often quite complex. One approach is to separate each switching function and provide a unique interface to the computer for that function. However, a more versatile approach is to provide a general-purpose switching unit. Although such a unit can't handle all of the switching requirements of a system, it can take care of most of them. A switch unit used in many HP systems is controlled through one computer I/O card. This card addresses and controls up to 16,384 switches, each of which is either a point in a switch matrix, a channel of an input scanner, or an independent relay or solid-state switch.

System Timing

One aspect of system design often overlooked by the newcomer to the field is the need for careful attention to system timing. In an automatic system, time delays must be inserted into the test program for several reasons. First, the outputs of signal or power sources take a finite time to stabilize after responding to a program command. Second, measurement instruments often receive transient disturbances on their inputs while the system is being automatically switched, and they need some time to recover from these disturbances. Also easily overlooked is the fact that there is often a considerable amount of cabling, and therefore shunt capacitance, in a system. Time must be allowed to let the signal sources stabilize the charge on this capacitance. The table below shows how much time it takes for a cable of shunt capacitance C and a source of resistance R to settle within various percentages of their final values.

Software

There are usually two levels of software in a system. There is the test program, which is usually written by the test engineer in a high-level language like BASIC, AuTest, or FORTRAN, and there is the system-control program, which must be written by whoever puts the system together. The person who builds the system must make sure that the software can control instruments, control timing, and perform any other system functions that the test engineer might call for. Software design is not a trivial part of system design; it usually requires an experienced programmer. More is said about software requirements in the preceding and following articles.

Checkout

A typical computer-controlled system has signal sources, switching, and measurement instruments. Therefore, with an appropriate test program for the computer, the system can check itself out to a great extent. This should be considered during the system design phase. It may be desirable to add some switching capability, or possibly a precision reference source, to make system checkout easier and more useful.

Cable Settling Time

System timing is affected by the time it takes cable capacitance to stabilize. A cable which has shunt capacitance C, driven by a source which has resistance R, will take the times shown here to settle within the indicated percentages of its final voltage.

Error	Settling Time
10%	2.3 RC
1%	4.6 RC
0.1%	6.9 RC
0.01%	9.2 RC

Hewlett-Packard Automatic Test Systems

HP uses and supplies small modular systems of all kinds: standard special-purpose systems, standard multipurpose systems, custom systems, and system components.

By Robert A. Grimm

HEWLETT-PACKARD'S INVOLVEMENT IN AUTOMATIC TESTING is both as a user and as a supplier. We use it extensively in our laboratories for testing components, circuits, and assemblies, and in our manufacturing departments for testing components, sub-assemblies, and products.

As a supplier, HP provides automatic test systems in three ways.

- **Standard systems.** Many test requirements are common to a number of users. HP has designed several general-purpose and special-purpose automatic systems, such as a microwave network analyzer,[1] a digital logic tester,[2] a dc-to-RF test system,[3, 4] an audio data processor,[5] an aircraft noise monitoring system,[6] a Fourier analyzer, and modular data acquisition and control systems. These are available as standard products.*
- **Custom configured systems.** Test requirements that can't be met by standard systems can often be met by a combination of standard products assembled by HP into a custom system. In such cases HP provides a complete system, including all instrument interfaces with the computer, and all necessary hardware and software.
- **Customer-assembled system.** During the past two years, more and more instruments and interface hardware have become available, and customers with some hardware and software design capability can often assemble systems themselves. The easiest HP systems to assemble are the computer-controlled data acquisition and control systems, for which a number of HP subsystems are available. These subsystems may consist of a single computer interface card or a collection of instruments with cards and cables. Software—drivers and verification routines—is also included.

The HP Approach

Hewlett-Packard's approach to automatic testing is the result of our experience as a manufacturer of general-purpose test equipment. The HP approach can be summarized as follows:

- Use commercial test instruments for building blocks.
- Use computers designed for easy instrument interfacing.
- Use simple programming languages.
- Make small free-standing modular test stations.

Standard Instruments

Because of the growing importance of automatic testing, new HP products have increasingly been designed with programmable controls and data readouts that are compatible with HP computers. Digital voltmeters, electronic counters, and frequency synthesizers were among the first of these; now there are also power supplies, switches, signal generators, X-Y displays, power meters, and others.

Standard HP instruments are used in HP systems where possible. If no HP instrument is made for the job, a standard instrument from another manufacturer is used. Only occasionally, when no instruments are available, are custom units designed and built.

HP Computers

Several years ago, HP estimated the future importance of automatic testing and developed small computers specifically for use in instrumentation systems. These computers have an input/output structure which makes it straightforward to interface them with test equipment; usually this can be done with a single card that fits inside the computer, and a cable to the instrument. The card accepts instructions in computer language and converts the information to the form which will control the instru-

*The seven systems mentioned are HP Models 8542A, 2060A, 9500A, 80501A, 80500A, 5450A, and 2300 Series, respectively.

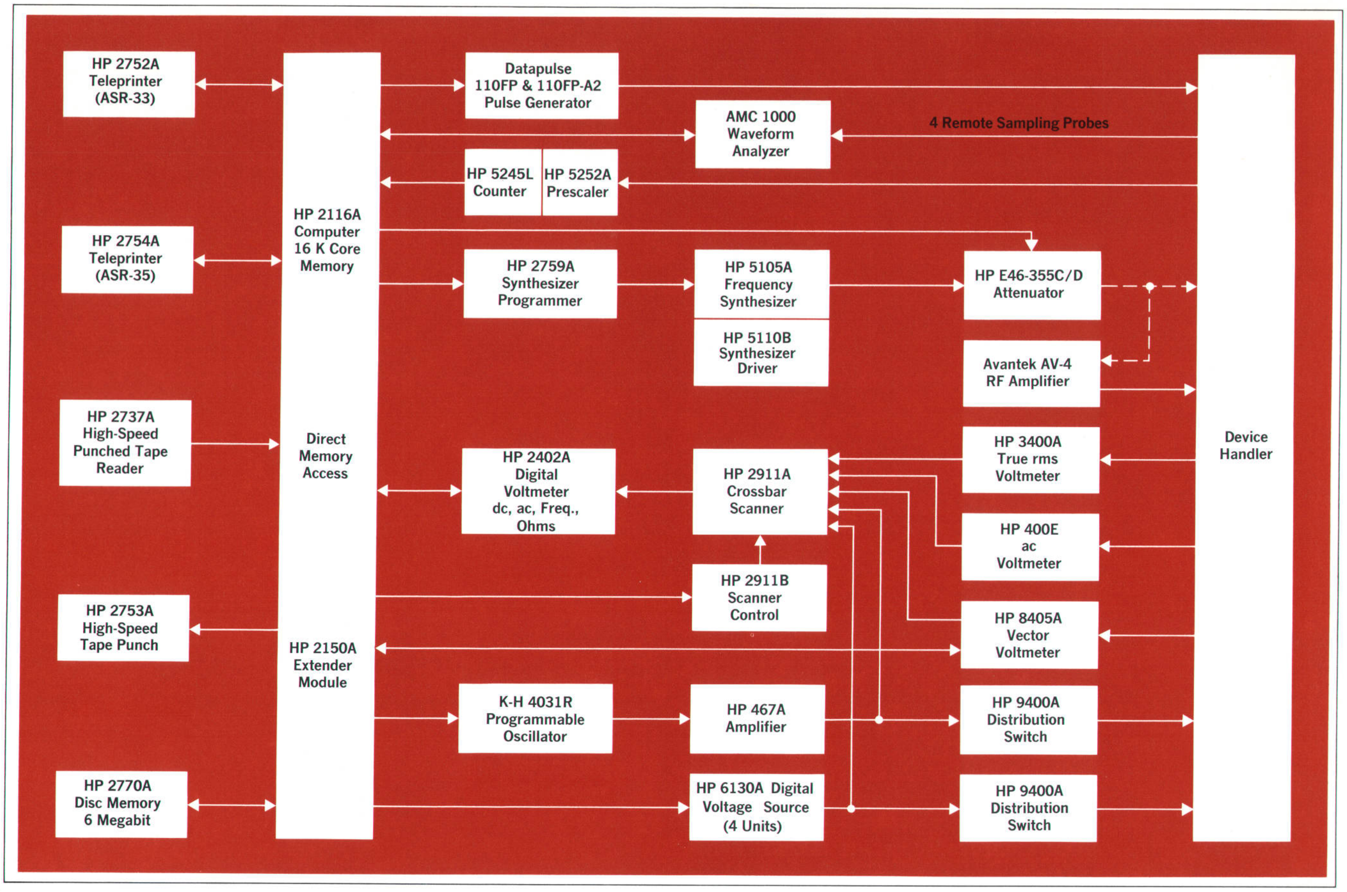

HP 2752A Teleprinter (ASR-33)
HP 2754A Teleprinter (ASR-35)
HP 2737A High-Speed Punched Tape Reader
HP 2753A High-Speed Tape Punch
HP 2770A Disc Memory 6 Megabit
HP 2116A Computer 16 K Core Memory
Direct Memory Access
HP 2150A Extender Module
Datapulse 110FP & 110FP-A2 Pulse Generator
AMC 1000 Waveform Analyzer
4 Remote Sampling Probes
HP 5245L Counter
HP 5252A Prescaler
HP 2759A Synthesizer Programmer
HP 5105A Frequency Synthesizer
HP 5110B Synthesizer Driver
HP E46-355C/D Attenuator
Avantek AV-4 RF Amplifier
HP 2402A Digital Voltmeter dc, ac, Freq., Ohms
HP 2911A Crossbar Scanner
HP 2911B Scanner Control
HP 3400A True rms Voltmeter
HP 400E ac Voltmeter
HP 8405A Vector Voltmeter
K-H 4031R Programmable Oscillator
HP 467A Amplifier
HP 9400A Distribution Switch
HP 6130A Digital Voltage Source (4 Units)
HP 9400A Distribution Switch
Device Handler

Fig. 1. *HP Model 7462 is used by the Magnavox Corporation to test radio receiver circuits. Typical test times are 3 to 6 seconds per circuit (amplifiers, local oscillators, etc.).*

ment. Thus an instrument can be added to the system simply by plugging in a card and making appropriate changes in the computer software. There are now three of these small general-purpose HP instrumentation computers: Models 2114A, 2115A, and 2116B.

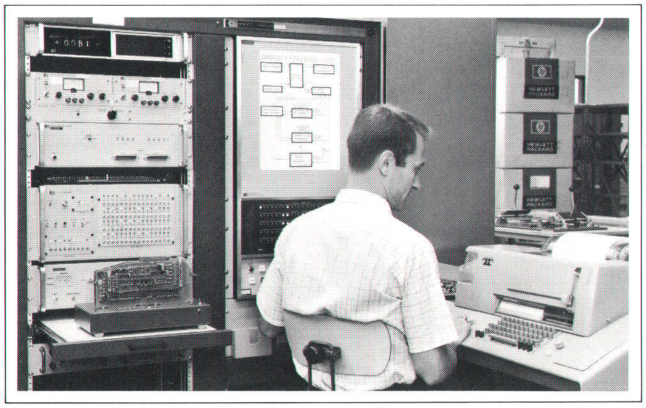


Fig. 2. *This system is used for production testing the read-only memories and logic cards in HP Model 9100A Calculators.*

Programming Languages

Easy programming is an important requirement in automatic test systems. The person who knows the tests that have to be performed is the best person to write the test program, but he is usually not a programmer. HP favors easy-to-learn, conversational, English-like languages. Many HP systems are programmed in an augmented version of the BASIC language—HP BASIC—which is standard BASIC with two additional statements.[7,8] The CALL statement initiates instrument-control subroutines, and the WAIT statement puts time delays into the program to allow for the settling times of instruments and devices being tested. HP BASIC can be learned in a day.

Table I is an example of an HP BASIC program for testing an audio amplifier. It includes instructions for setting all power supplies, checking the gain at five frequencies from 1 kHz to 5 kHz, and generating appropriate output messages, including the amplifier serial number.

Some special-purpose HP systems are programmed in special languages which are also quite simple to learn.

For example, the digital logic module test system, HP 2060A,[2] is programmed in AuTest, a special language optimized for testing binary logic circuits. It includes a PERMUTE instruction which can generate thousands of tests with only two program statements. Table II is an example of an AuTest program.

Many HP systems can also be programmed in FORTRAN and ALGOL. However, these languages are not interpretive, so they have some disadvantages for automatic testing. For example, a FORTRAN program written in English-like statements must be compiled, that is, totally converted into a binary machine-language program on paper tape or some other similar medium. The binary tape is then used to operate the system. Any changes or errors require re-compiling, and this can be tedious and time-consuming for one new to programming. HP BASIC, on the other hand, is an interpretive system which operates on each line of the English-like program as it is received, converting one line at a time into machine language. Errors are immediately presented to the programmer and may be corrected by simply retyping the line that contains the error. Similarly, program changes may be made at any time by retyping the English-like program statements.

An exception to all this is the computer-controlled data acquisition or control system. Most of these systems operate unattended for long periods of time, so conversational programming is not as important as it is in an automatic test system. HP data acquisition systems are programmed in FORTRAN or lower-level languages because, once these programs are compiled and loaded into the computer, they take less time to execute than equivalent BASIC programs.

A new software package for these data acquisition and control systems is DACE, the data acquisition and control executive program. With DACE, the measurement, computation, and control functions of a system are separated into tasks of varying priorities, and the executive

Table I. Test program for an audio amplifier, written in HP BASIC

Program	Comment
100 CALL (8, 1, 0, 1)	Sets power supply 1 to zero
110 CALL (8, 2, 0, 1)	Sets power supply 2 to zero
120 PRINT "PLUG IN AMPLIFIER"	Instructs operator
125 PRINT "SERIAL NUMBER IS"	Asks operator for information
130 INPUT S	Operator types in Serial No.
140 CALL (6, 4, 3, 0, 0)	Connects supply #1 to output 4, supply #2 to output 3
150 CALL (8, 1, −12, 100)	Sets supply #1 to −12, 100mA max
160 CALL (8, 2, 12, 100)	Sets supply #2 to +12, 100mA max
170 FOR F = 1000 TO 5000 STEP 1000	Establishes loop for changing frequency
180 CALL (5, F, .10)	Sets oscillator to 1000 (then 2000, 3000, 4000, 5000) Hz, 0.1 V
190 CALL (7, 7, 0, 0, 0)	Connects oscillator to output 7
200 CALL (9, 5)	Connects oscillator output to DVM
210 WAIT (30)	Delays 30 ms to allow settling
220 CALL (10, 2, .1, I)	Measures input ac voltage on 0.1 range
230 CALL (9, 23)	Connects amplifier output to DVM
240 WAIT (30)	Delays 30 ms to allow settling
250 CALL (10, 2, 10, V)	Measures amplifier output
260 LET G = V/I	Calculates gain
270 IF G < 5 THEN 320	Checks for low gain
280 IF G > 10 THEN 340	Checks for high gain
290 NEXT F	Return to 170 for next frequency
300 PRINT "AMPLIFIER SERIAL" S "GAIN OK, GAIN =" G "AT 5000 HZ"	
310 GO TO 290	
320 PRINT "AMPL SERIAL" S "GAIN LOW, GAIN =" G "AT" F "HZ"	
330 GO TO 290	
340 PRINT "AMPL SERIAL" S "GAIN HIGH, GAIN =" G "AT" F "HZ"	
350 GO TO 290	
360 END	

Table II. Test program for a logic module, written in AuTest

Program	Comment
IDENTIFIER, 02116-6274, MA6274	Name the program for module part number 02116-6274, used with module adapter 6274
CONNECTOR, BANK 1 (3, 4, 5, 6 . . . . 55, 81, 82, 83)	List all module signal connector pins
1 POWER, PS1 = 4.5V, PS2 = −1.8V,/ BANK 1 (H = 3V, L = −1.8V,/ P = 30MA, N = 50MA,/ +T = 0.5V, −T = 0.5V)	Specify minimum power supply and high/low logic level voltages as well as positive and negative current limits and test tolerances for first run-through of test
2 POWER, PS1 = 5V, PS2 = −2V,/ BANK 1 (H = 5V, L = −2V,/ P = 30MA, N = 50MA,/ +T = 0.5V, −T = 0.5V)	Specify maximum power supply and high/low logic level voltages and remaining parameters for second run-through of test
3 TEST, L( ), H(81)	Set all pins to low initial states and set pin 81 to high for check of thermal switch during all tests
4 DO, 5, P = 4 5 TEST, PERMUTE (3, 27, 29, 25)	Check memory module select logic by generating 16 (2^4) memory address input combinations for pins 3, 27, 29, and 25
6 TEST, H(3)	Set pin 3 (M12) for test of memory protect output AND gate
7 DO, 8, P = 8 8 TEST, PERMUTE/ (37, 55, 54, 53, 52, 51, 50, 49)	Check memory protect output AND gate by generating all 256 (2^8) possible input combinations
9 TEST, L(3) 10 TEST, H(55, 54, 53, 52, 51, 50, 49)	Set pin 3 (M12) low and pins 49 through 55 high for check of remaining memory protect logic
11 DO, 12, P = 8 12 TEST, PERMUTE/ (32, 31, 24, 33, 12, 35, 6, 37)	Check remaining memory protect logic by generating all 256 (2^8) possible input combinations
13 END	Terminate test program.

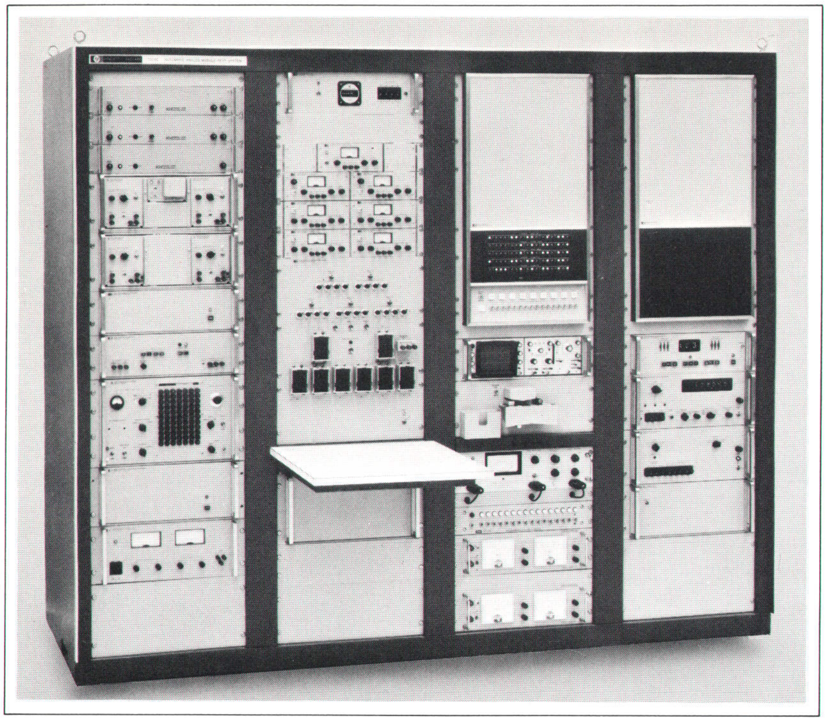
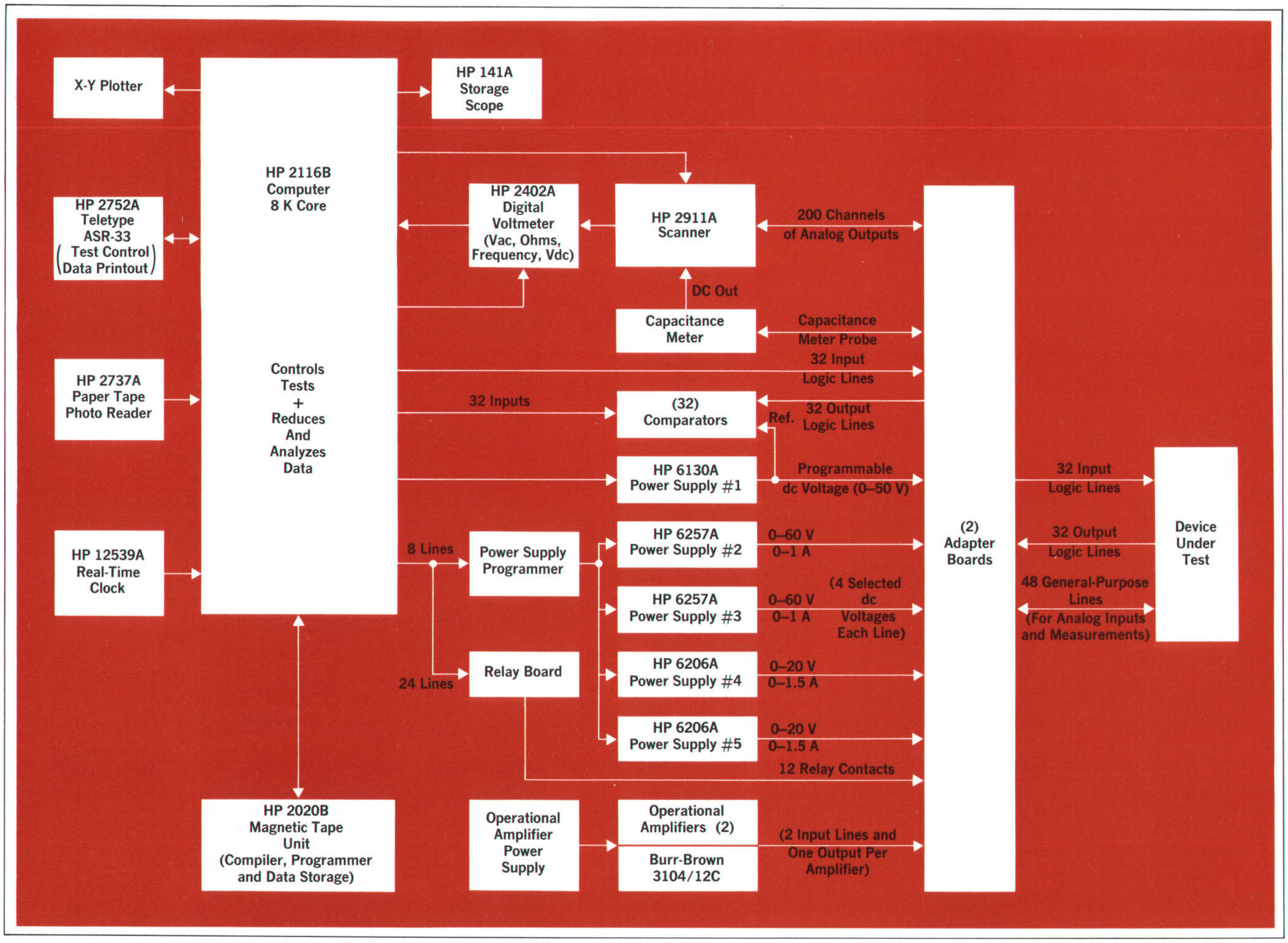


Fig. 3. *(Left) HP Model 7404 is used by Litton Systems to test analog modules. Programmable stimuli are dc voltage and ac voltage with variable frequency, phase, amplitude, and waveshape (sine or square). Measurements include dc and ac voltages, resistance, and phase.*

Fig. 4. *(Below) This system is used by Hewlett-Packard Laboratories for testing solid-state dc and digital devices.*


X-Y Plotter
HP 141A Storage Scope
HP 2116B Computer 8 K Core
Controls Tests + Reduces And Analyzes Data
HP 2752A Teletype ASR-33 (Test Control Data Printout)
HP 2737A Paper Tape Photo Reader
HP 12539A Real-Time Clock
HP 2020B Magnetic Tape Unit (Compiler, Programmer and Data Storage)
HP 2402A Digital Voltmeter (Vac, Ohms, Frequency, Vdc)
HP 2911A Scanner
200 Channels of Analog Outputs
DC Out
Capacitance Meter
Capacitance Meter Probe
32 Input Logic Lines
32 Inputs
(32) Comparators
Ref.
32 Output Logic Lines
HP 6130A Power Supply #1
Programmable dc Voltage (0–50 V)
8 Lines
Power Supply Programmer
HP 6257A Power Supply #2
0–60 V 0–1 A
HP 6257A Power Supply #3
0–60 V 0–1 A
(4 Selected dc Voltages Each Line)
24 Lines
Relay Board
HP 6206A Power Supply #4
0–20 V 0–1.5 A
HP 6206A Power Supply #5
0–20 V 0–1.5 A
12 Relay Contacts
Operational Amplifier Power Supply
Operational Amplifiers (2)
Burr-Brown 3104/12C
(2 Input Lines and One Output Per Amplifier)
(2) Adapter Boards
32 Input Logic Lines
32 Output Logic Lines
48 General-Purpose Lines (For Analog Inputs and Measurements)
Device Under Test

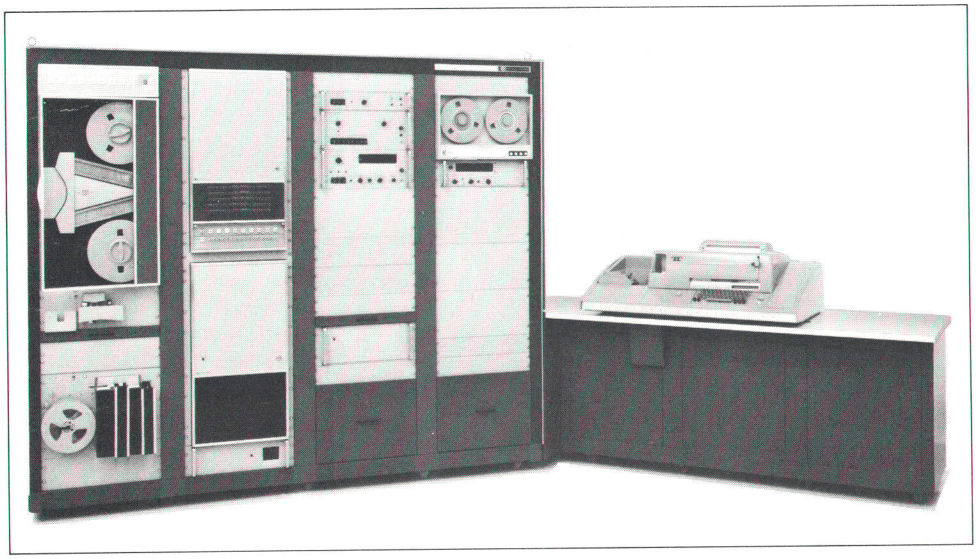


Fig. 5. *Hughes Aircraft Company uses the HP Model 7389 for testing missile subassemblies. It reduces test times to 15 seconds from 5 minutes per module.*

program schedules these tasks in real time to make most efficient use of the computer.

HP Computers can also be programmed in lower-level languages like machine code or assembly language. However, these are not English-like languages and are usually used only by experienced programmers. Table III shows the differences between machine language, assembly language, compilers, and interpreters.

Small Self-Contained Test Stations

With inexpensive computers it is practical to make individual test stations completely self-contained. The test equipment in each station is chosen for the devices to be tested. Separate test stations for different types of modules are practical. For example, HP makes a standard system for testing digital boards[2] and another one for testing microwave devices.[1] With two such systems, more testing can be accomplished, and program debugging, system modification, and maintenance will be easier than they would be with a single system which did both types of testing. And since the systems are modular, they can be expanded or modified to meet changing requirements.

A Few Examples

HP's small modular test stations are practical solutions to many testing problems in repair depots, in calibration laboratories, and on production lines. A few examples will illustrate the range of possibilities.

Table III. Types of Programming Languages

1. Machine Uses minimum memory Can be performed on line But difficult for human to read or write directly	**Machine Language** (2116B) 0110001100100001 0100001100100010 0100001100100011 0111001100100100
2. Assembly Easier for human to understand But uses many steps Requires off-line assembly Difficult to change	**Assembly Language** (2116B) LDA A ADA B ADA C STA D
3. Compiler Very easy to understand Very few steps But requires off-line compiling Also may use more memory	**FORTRAN** D = A + B + C
4. Interpreter Advantages of Compiler Is on line Easy to change But may require more memory	**HP BASIC** LET D = A + B + C

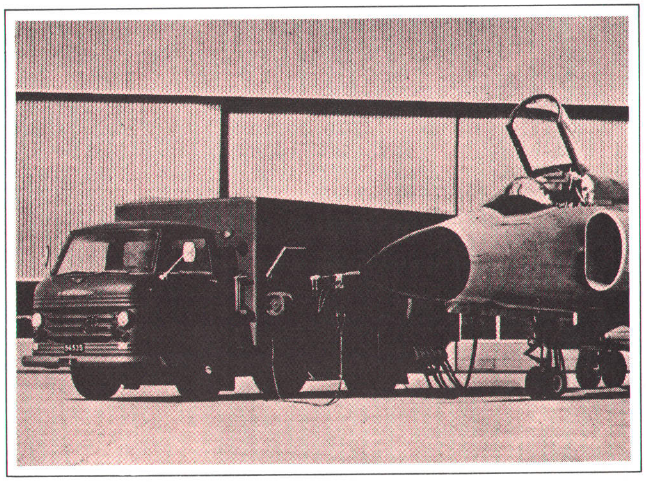


Fig. 6. *The Royal Swedish Air Force will use HP systems to check out avionics components.*

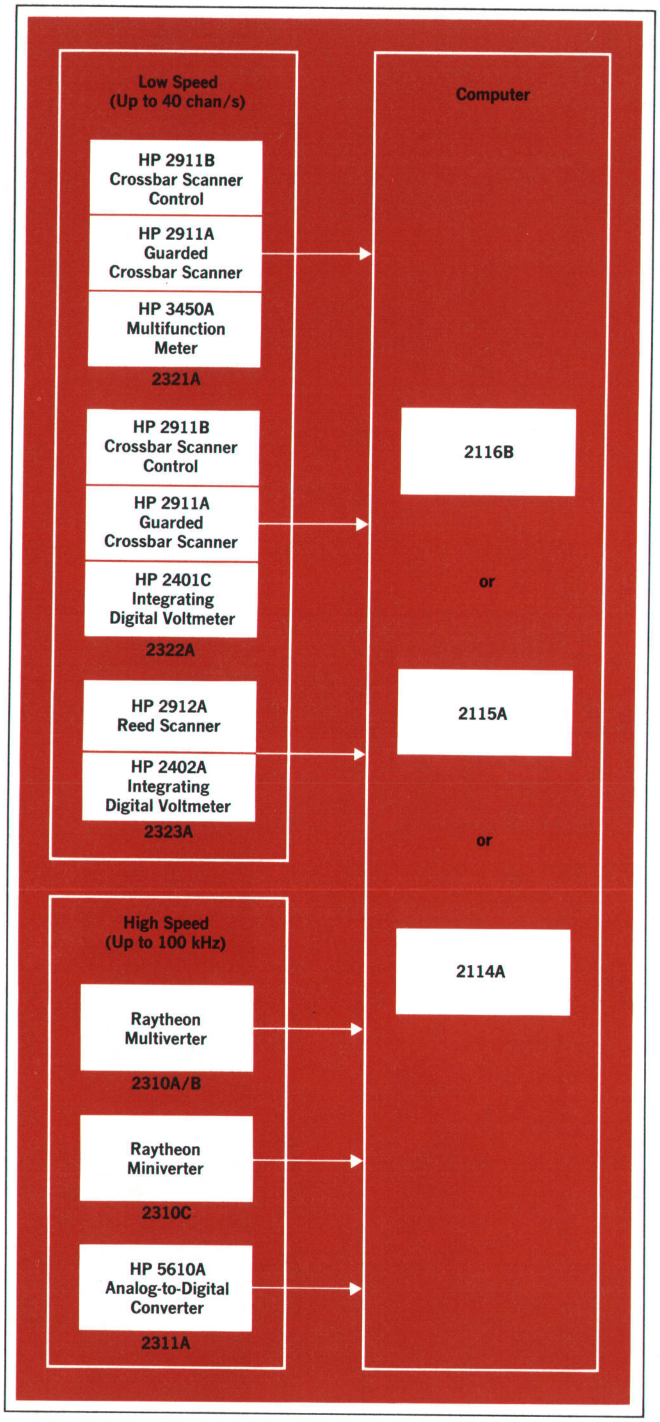

Low Speed
(Up to 40 chan/s)
HP 2911B Crossbar Scanner Control
HP 2911A Guarded Crossbar Scanner
HP 3450A Multifunction Meter
2321A
HP 2911B Crossbar Scanner Control
HP 2911A Guarded Crossbar Scanner
HP 2401C Integrating Digital Voltmeter
2322A
HP 2912A Reed Scanner
HP 2402A Integrating Digital Voltmeter
2323A
High Speed
(Up to 100 kHz)
Raytheon Multiverter
2310A/B
Raytheon Miniverter
2310C
HP 5610A Analog-to-Digital Converter
2311A
Computer
2116B
or
2115A
or
2114A

Fig. 7. *HP 2300-series subsystems include interface cards, cables, instruments, and software for use in modular computer-controlled data acquisition and control systems.*

Fig. 1 shows the HP 7462 Automatic Test System, (a version of the standard Model 9500A[3, 4]) which is being used by the Magnavox Company of Fort Wayne, Indiana, to test thick-film circuits for hand-held radio transceivers. This system is an excellent example of the building-block approach being used by Hewlett-Packard to produce automatic test systems for production line test stations. The HP 7462 uses commercially available catalog instruments, the HP 2116 computer, and the HP BASIC language. The high volume of Magnavox circuit production and the large number of required tests make this a clear example of the way automatic test systems can serve users. Individual steps in the Magnavox program being used with the HP 7462 take from a few milliseconds to a few hundred milliseconds. Typical complete test times for radio receiver circuits—e.g., RF, IF, and audio amplifiers, local oscillators, and so on—are in the range 3 to 6 seconds.

Other HP systems are illustrated in Figs. 2 through 6, and on pages 2, 3, 4, and 9. Some are in-house systems, some are standard special-purpose and multipurpose systems, and some are custom systems. Fig. 7 shows the subsystems that are available for data acquisition and control.

References

1. R. A. Hackborn, 'An Automatic Network Analyzer System,' microwave journal, Vol. 11, No. 5, May 1968.
2. W. P. Cargile, 'A Computer-Controlled System for Testing Digital Logic Modules,' **Hewlett-Packard Journal,** Mar. 1969.
3. 'Automatic Test Systems for Communications Equipment,' Telecommunications, Feb. 1969.
4. R. A. Grimm, 'Using Standard Instruments and an Abbreviated English-Language Program in a Computer-Based Automatic Test System,' Electronic Instrument Digest, May 1969.
5. W. T. Kapuskar and C. J. Balmforth, 'Real-Time Measurement and On-Line Processing of Acoustical and Other Audio-Frequency Spectra,' **Hewlett-Packard Journal,** July 1969.
6. W. T. Kapuskar and C. J. Balmforth, 'Monitoring Airport Noise,' **Hewlett-Packard Journal,** May 1969.
7. G. L. Peterson, 'BASIC—The Language of Time-Sharing,' **Hewlett-Packard Journal,** Nov. 1968.
8. R. M. Moley, 'BASIC at Hewlett-Packard,' **Hewlett-Packard Journal,** Nov. 1968.

HEWLETT-PACKARD JOURNAL *AUGUST 1969 Volume 20 • Number 12*

TECHNICAL INFORMATION FROM THE LABORATORIES OF THE HEWLETT-PACKARD COMPANY PUBLISHED AT 1501 PAGE MILL ROAD, PALO ALTO, CALIFORNIA 94304
Editor: R. H. Snyder Editorial Board: R. P. Dolan, L. D. Shergalis
Art Staff: John C. Allen, Clayton Associates, Director; Maridel Jordan, Assistant