

An IEEE SPECTRUM applications report

A look at automatic testing

Automatic testing of electronic devices has been a major factor not only in the overall improvement of product quality and reliability, but also in the dramatic lowering of product costs

Harold T. McAleer *General Radio Company*

This survey reviews the current state of automation in the testing of electronic components, networks, and circuits. Elements and characteristics of typical test systems, both hardware-controlled and computer-controlled, are described. Paths to be followed and pitfalls to be avoided in achieving automation are discussed in an effort to help the reader toward a better understanding of the subject and its broad applications. "How to automate successfully," a major theme in the report, places emphasis on economic justification.

The need for and desirability of automatic testing have been well established and require little justification. The testing function itself, which used to be considered somewhat of a poor relation, a necessary evil, has gradually emerged as a well-recognized and essential part of any manufacturing operation—particularly in electronic manufacturing. This testing function appears in many areas and in many different forms. It may be called incoming inspection, line inspection, final test, or quality control, and it may involve sampling tests or 100 percent inspection. From the standpoint of product quality, 100 percent inspection at every recognizable step in production would be ideal. This ideal was seldom possible because of the high costs involved, both in time and money. As the inspection process becomes automated, however, 100 percent inspection becomes economically feasible and an attainable goal, especially when one realizes how rapidly the cost of repair rises as a faulty item gets buried in the final product.

Experience with transistors offers an example. It has been calculated¹ that the cost of *not* subjecting transistors to 100 percent testing as received from the manufacturer, but rather sample testing them to an acceptable quality level (AQL) of 0.65 percent, can amount to 8½ cents per transistor purchased. For a user of a million transistors per year, this amounts to an annual *loss* of \$87 500!

Integrated circuits provide another example. Back in 1969 when the price of integrated circuits varied from 85 cents to \$3.50 each, depending on complexity, the typical failure rate, as received from the IC manufacturer, was around 1.5 percent. Now that the price is down to 20–85 cents each for the same units, the failure rate is up to 2–3 percent.* With an average IC count of 20 to 30 per board and an average time to diagnose and repair of 20 to 30 minutes, we can see the importance of culling out bad units as early in the manufacturing process as possible.

Since electronic equipment is used more and more in products involving safety considerations—spaceships, aircraft, automobiles, medical instrumentation—and the cost of failures may be measured not only in dollars but in human lives, the need for thorough testing (as well as sound design) becomes even more important.

Levels of testing

In general, five distinct levels of testing can be recognized, one in the engineering-design phase of a product, three in the manufacturing phase, and one in the post-shipment phase. In the design phase, testing is required not only to prove out or modify the design, but also to evaluate and select the materials, components, or techniques used in the product. In the manufacturing phase, testing is required at the input to the production process to qualify the materials used in the product (incoming inspection). It is also required during the process to assure that all the steps have been successful (line inspection) and at the end of the process to qualify the final product (final inspection). In the postshipment phase, (service) testing is required, to establish that operation is proper or to diagnose failures and aid in repair.

Different test information is required at these different levels and different types of test equipment are often used. Engineering and incoming-inspection tests may require the acquisition and analysis of detailed measurement data, whereas line-inspection and final-inspection tests are more concerned with “go” or “no-go” decisions to maximize throughput.² Testing and test equipment used during the production process is sometimes separated into two types—equipment and techniques for the rapid sorting of items into “good” and “bad” categories and equipment and techniques for the slower diagnosis and repair of the bad items. Diagnosis and repair equipment is also used in the servicing of delivered products.

Distinctions between laboratory, production, and servicing equipment are disappearing, however, as design engineers are required to design and specify not only the product but also the test equipment and techniques.

Measuring systems

The block diagram of Fig. 1 shows the variety of items or functions to be found in any measuring system, manual or automatic. The major functions are within the blocks, and the words around the blocks show examples of hardware items that accomplish the functions. Although not perfect, the diagram provides a useful basis for discussion. The tinted blocks show the essential items: a device to be tested, a stimulus source to provide a signal for the device, a measurement instrument to quantify the response, and an operator to make the whole thing go.

The other blocks show the functions that can be added to ease the operator's task and automate the process. As equipment for these functions is added, the operator acts less as a mechanical part of the measurement process and can devote his attention to the results rather than the details of the measurement. If the results are used for automatic control of other equipment operating on the device under test, a process-control system can be achieved.

Device under test. It is useful to categorize the device under test according to the hierarchy shown in Fig. 1. Thus, a component is made out of materials, a network is made out of components, a circuit is made out of networks, etc. The categories can be further subdivided: components and networks can be linear or nonlinear, active or passive; circuits can be analog or digital, discrete or integrated. The hierarchy can be shown to be imperfect (for example, is a transformer a component, a network, or a circuit?) but in general any electronic device can be fitted into one of the categories.

Adaptor. The adaptor block is used to define items that interface the device under test into the measurement system. This category includes test boards, fixtures, or sockets that contact the leads or terminals of a component or network, or a multipoint probe that contacts the nodes of a microcircuit.

Input. Input equipment facilitates the insertion of the device under test into the adaptor. Examples are vibratory-bowl component feeders, rotary and linear transports, and other mechanical devices. Such items are usually associated with a sorting or binning device to deposit the tested unit into a useful location.

Condition. Conditioning equipment applies to the device under test a secondary stimulus, such as power,

* Typical user experience. However, for a premium of 5–10 cents each the customer can buy units with a guaranteed failure rate of less than 0.1 percent; i.e., the IC manufacturer will test them more thoroughly.

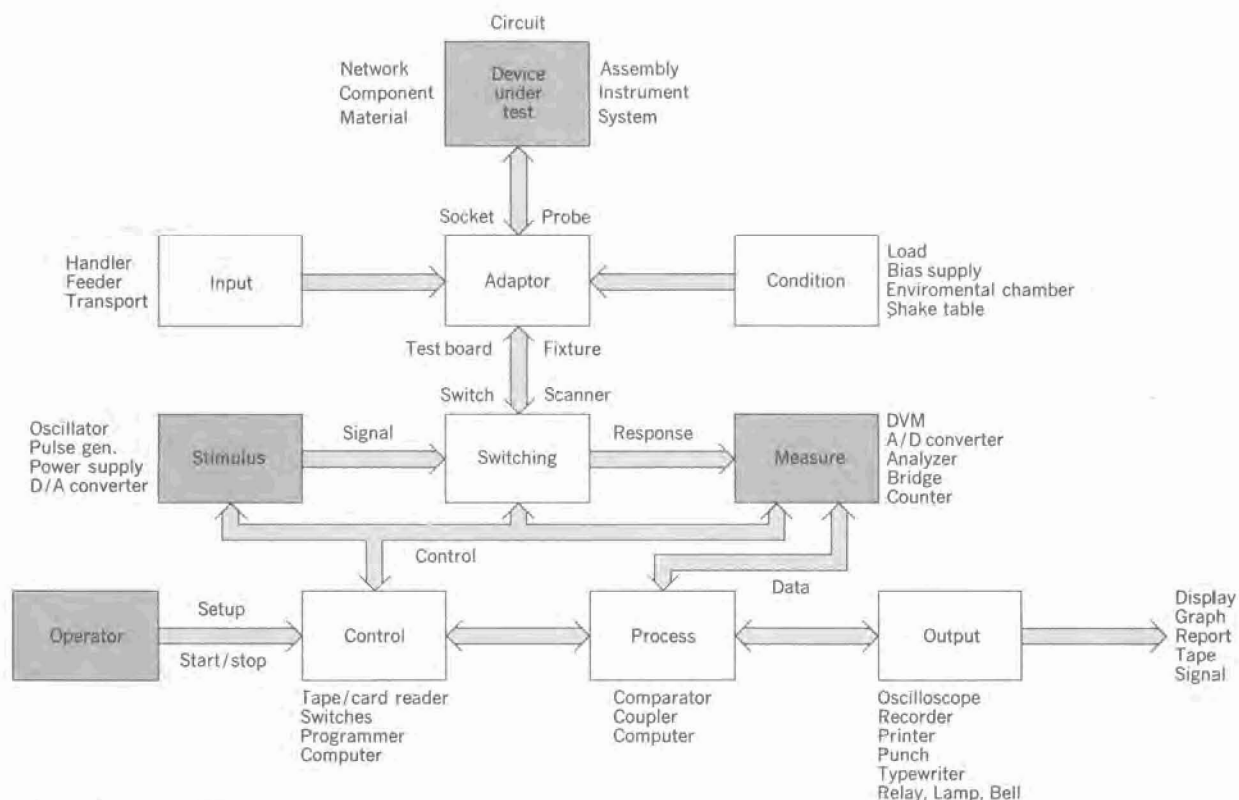


FIGURE 1. Functional block diagram of a manual or automatic measuring system.

bias, heat, cold, or shock, or sets up terminal load conditions.

Switching. Switching equipment is used to connect the device or device adaptor to the test system and to vary the connections of the device terminals. Examples are solid-state scanners (multiplexers), relay trees, crossbar scanners, and reed-relay scanners.

Stimulus. Stimulus sources for measurement systems are legion, and include dc supplies, oscillators, synthesizers, function generators, ramp, pulse, burst, and word generators, and D/A converters, among others. For use in automatic test systems such sources are often required to be programmable; that is, all their functions should be controllable by electric signals instead of (or in addition to) manual controls.

Measure. Measuring instruments are likewise legion; they include voltmeters, current meters, phase meters, impedance bridges, frequency counters, A/D converters, etc. These instruments should also be programmable. Data output in digital form is a useful, but not always necessary, characteristic of such instruments. The signal and measure functions are often combined in a single instrument, as for example in an automatic capacitance bridge or a digital ohmmeter.

Control. The control function can be accomplished by a variety of equipment in a variety of ways. Card readers and tape readers are often used to program and sequence automatic equipment with the actual program determined by the holes in the cards or tape. Patchboards, plug boards, and switches can be used as "memories" to set up test conditions and sequences. Cam-operated motor-driven switches are used as well as relay networks. Specialized digital-logic circuits are often applied to this

function. The most versatile programmer/controller of all, the computer, will be discussed in a later section.

Process. Data from the measurement instruments must often be "massaged" or processed in some way. One example of processing is limit comparison. The comparator may be an analog or digital circuit. It compares the measured data with preset or programmed limits and determines whether the data are above, below, or within the limits and provides appropriate output signals (lamp illumination, contact closures, digital-logic signals, etc.). Another form of processing is coupling. The coupler may store and convert parallel measurement data (all digits at once) to serial data (one digit at a time) to operate card punches or tape punches or strip printers, or it may perform the opposite function (serial-to-parallel conversion) to operate gang punches or line printers or displays. Yet another form of processing is computation or data processing. The processor may perform arithmetic computations on the data to provide corrections or conversions or to somehow convert the data to more useful form.

Output. The output function of an automatic test system is also performed by a variety of equipment and appears in a variety of ways. One typical output function is that of display. Measurement results or test conditions may be displayed by means of oscilloscopes, panel meters, indicator lamps, digital indicators (visual display), or by ringing a bell (auditory display). Hard-copy output can be provided in the form of graphs, printed lists, or typed reports by means of recorders, printers, or typewriters. Output can consist of condensed data in the form of punched cards, punched tape, or magnetic tape for off-line processing by other equipment. Output

can also consist of signals for on-line processing by local or remote equipment. Finally, output signals can be used to close the loop on the test process by activating sorters, automatic handlers, lasers, sandblasters, or other devices that affect the device under test.

Interface. One function not shown in the diagram is that of interface. This can be considered a miscellaneous or catchall category used to dignify any function that cannot be conveniently forced into one of the other categories. In general, it defines any item that is used to tie together some of the other items. This function becomes one of extreme importance in computer-controlled systems, as will be discussed later.

It should be noted that the diagram of Fig. 1 is a functional rather than a hardware diagram. The hardware diagram may be similar, but often it is not, since many hardware items combine several of the functions.

The computer

The largest single factor contributing to the revolution in the art of automatic testing has been the emergence of the computer, in particular the minicomputer, as an available and reliable item of hardware with immense, if not perfect, capability. It appears everywhere in automatic test systems, both for programming and controlling and for data processing. The computer however, is not always necessary, or even desirable, in some applications. How do you know when to use one? The following paragraphs discuss some considerations.

Operational flexibility. If the job of the test system is fixed, and well defined—for example, the sorting of resistors into prescribed categories—and the test system will be used for that one specific task forever, a computer may not be required. If simple and economical hardware—timers, programmers, bridges, comparators, etc.—

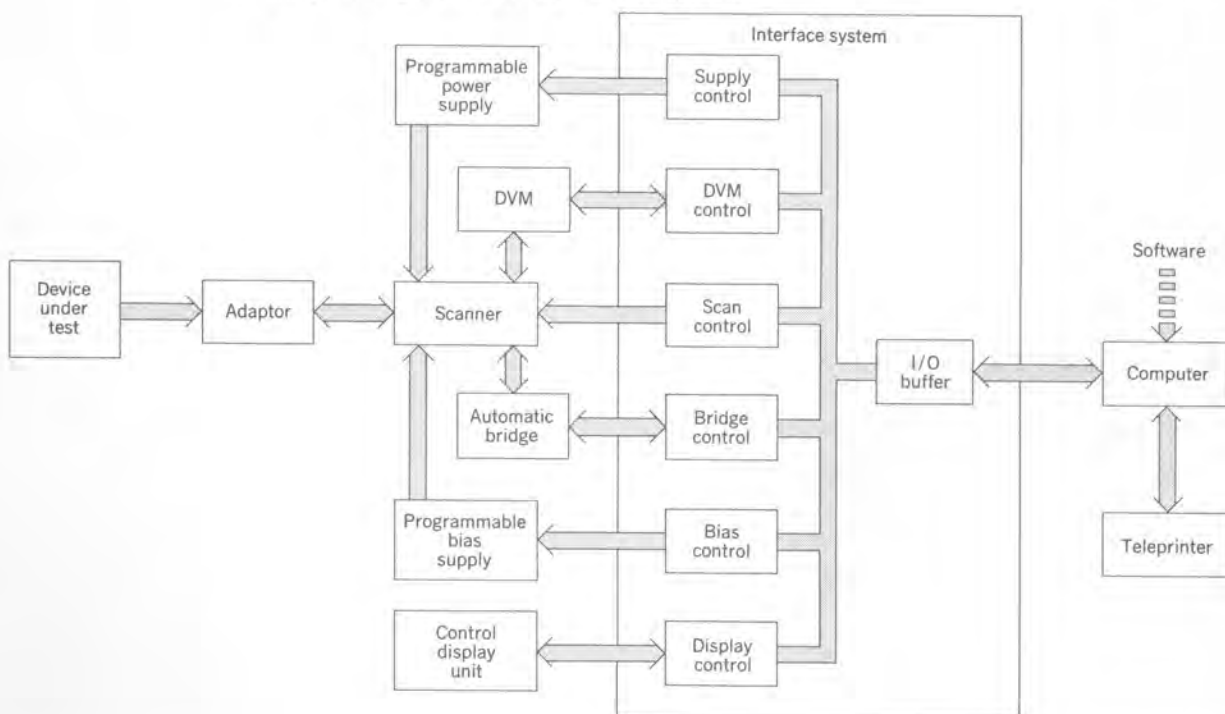
exists to do the job, it should be used. It won't take much, however, to warrant a computer. When the number of possible controls and settings becomes large—nominal values, upper and lower limits, bias voltage, test voltage, test frequency, advance, dwell and soak times, etc.—the sheer number of knobs, dials, and switches to adjust can get out of hand and some form of sophisticated programmer will be required. For this task alone, the computer may represent an economical solution.

Changing requirements. When the test system has to be rapidly converted from one job to another (and enough hardware can be combined in one system to do all the jobs) the computer may be the most economical way to provide this rapid conversion.

Data manipulation and output. If the measuring instruments provide data in the right form and format and processing equipment and output devices (couplers, printers, etc.) are available, the computer is not necessary. If, however, the data must be massaged or corrected and a complicated output format provided, the computer may offer the only solution. In this regard the computer provides a side benefit: the ability to correct measurement data for fixed or systematic errors and thus improve the accuracy of the basic measurement system. It also offers the possibility of providing data analysis in the desired form and eliminating the need for further off-line processing.

Computer functions. The computer can perform two of the basic functions shown in Fig. 1, the control and process functions. It can perform the control function by means of a program stored in its memory and the process function by means of its central processor. To make use of these capabilities, however, two other elements are required: an interface system and software.³

FIGURE 2. Simplified block diagram of a typical computer-controlled component-test system emphasizing the role of the interface system.



Interface system. Figure 2 is a simplified block diagram of a typical computer-controlled test system drawn purposely to emphasize the role of the interface system. The test system includes several programmable elements to perform the functions of Fig. 1. The interface system is required to wed these elements to the computer, since at this stage of the art neither computers nor test instruments are directly compatible.⁴ Cost considerations in the design of general-purpose test instruments do not allow the inclusion of expensive interface circuits. As we grope our way toward standardization, however, test instruments are becoming available that make the interfacing task, if not unnecessary, at least easier. Similarly, the design of computers cannot anticipate all the possible interface requirements. In general, the interface circuits perform the functions of I/O (input/output) bus expansion, level conversion, decoding, storage, transfer, and control. By storing data signals, both from and to the computer, the interface system allows the (fast) computer to keep busy while the (slow) test instruments are doing their job.⁵ When the test instruments are ready, either to receive instructions or give up their data, their interface control circuits can provide an "interrupt" signal and an identifying signal (flag) to the computer to request its attention.

Some minicomputers make provision for plug-in interface-circuit cards within their packages; others require an external unit. General Radio uses its 1761 Interface System in computer-controlled test systems. Hewlett-Packard provides plug-in interface cards for its computers and offers its 6936A Multiprogrammer System, a main-frame unit and a series of plug-in interface cards for its test instruments. Digital Equipment offers card-cage hardware and digital-circuit modules that can be used to construct interface systems, and other manufacturers offer similar items.

Expanding role of the computer. As the art progresses, it is becoming increasingly difficult to separate the elements of computer, interface, stimulus source, and response measurement, as well as to decide where one function leaves off and the other begins. Many automatic test sets of the future (and some of the present) will consist of card cages or main frames to house plug-in functional modules—memories, arithmetic elements, storage registers, stimulus sources, measurement modules, and peripheral-control circuits for input, control, and output devices.

Software. To make most efficient use of the limited core memory of the typical minicomputer, operating programs are often written in assembly language. (Such programming efforts are not trivial; it can take 4 to 6 man-months to produce 4000 words worth of debugged software and can cost \$10 000 to \$20 000, depending on how you value a man-month.) This approach works fine if the test set will be used with a few infrequently changed programs. If not, more memory, and perhaps the use of a high-level test-set language, may be needed.

Computer languages. Much has been written and said about computer languages,^{6,7} the terminology of communication between the programmer and the computer, and the level of such languages. There are no strict definitions of language level. In general, high-level languages allow the programmer to write program statements in terms that are familiar to him, such as English words or mathematical equations, and with few rules of

order or format, so that he can ignore the particular characteristics of the computer used to run the program. Low-level languages, on the other hand, force the programmer to adapt his problem to the characteristics of the machine to be used. They may use mnemonics or octal numbers and are related to the operation of the computer and its subsections. The use of higher-level languages may require several intermediate steps before a final operating program is prepared or run—that is, editing, debugging, punching tapes, loading tapes, etc. Given sufficient computer capability, however, the steps can be handled automatically (with little programmer action) by the use of a sophisticated operating-system program, and programs can be prepared and run at essentially the same time.

Table I shows some common types of programming languages. For further details, see Appendix A.

Test-set languages. One also can speak of programming and operating languages for automatic test systems. Test-set languages can be high level or low level, depending on their relationship to the application (problem) or to the test set (equipment). When you set the function knob of a digital counter to FREQUENCY, you are using a high-level language, related strongly to the application but of little flexibility. Similarly, setting the power switch to OFF uses a lower-level language. This concept has given rise to a host of test-set languages, Pol (Problem-Oriented Language), Tool (Test-Oriented Operator Language), Atlas (Abbreviated Test Language for Avionics Systems), and many others that have no names but do their jobs just as well. If the test set is dedicated to a specific application (capacitor testing, transformer testing, logic-circuit testing), the language can be made high level and use terms relating to the device under test (nominal value, transformer terminals, logic-drive pattern). If the test set is for more universal use, its language should be lower level and use terms relating to the instruments within the set (function, range, connection).

The program of a dedicated test set can provide two levels of language for the users of the set, a low-level language for the programmer, which will enable him to change certain operating constants or test sequences, and a higher-level language for the operator, which will enable him to control the set for a specific test. A partic-

I. Common programming languages

Type	Example	Typical Statement	Comment
Machine	PDP-8	100011101101	Machine code, object code, run code
Assembler	Pal III	HLT	Can make optimum use of memory and minimize run time. Error prone. Expensive. Great variation between programmers. Difficult to document
Compiler	Fortran Algol	D = A + B + C	Off-line compiling. Object program requires more memory and runs slower. Less variation between programmers. Easier to document. Compiler program may be expensive
Interpreter	Basic	IF E > 2 PRINT GO	Interactive for easy changes. On-line run. Requires even more memory and runs even slower

ularly useful mode of operation with such languages is an interactive or conversational mode, wherein programming questions are automatically displayed, printed, or typed by the set and replies are typed by the operator. The main resident program—the operating system—interprets the operator's instructions and runs the test.

Typical test systems

To illustrate the concepts described earlier, the following paragraphs discuss a few examples of commercially available automatic test systems. Excellent surveys are available in trade publications and commercial reports.⁸⁻¹⁰ The examples will be discussed in order of the device under test in the hierarchy previously described.

Component tests. Although threatened by film technology, classical discrete components (resistors, capacitors, inductors, transistors, diodes, etc.) remain with us. Their form is changing, but their function remains the same. Many automatic systems are used in the manufacture of these components—some for measurement only and some for process control.

Diode test. Figure 3 shows a hardware-controlled system for the automatic sorting of semiconductor diodes. The system includes a handler/sorter, which includes a vibratory-bowl feeder, a rotary test table with three test positions, and a five-bin sorting mechanism. The instrumentation includes three test instruments; a capacitance comparator, which compares the junction capacitance of the diode to a preset limit; a stored-charge (switching-time) detector; and a diode classifier, which tests for short and open circuits, reversed polarity, peak inverse voltage, reverse current, and forward voltage drop. The results from the three instruments are combined in the diode classifier to provide bin-sort signals to the handler. These systems can test and sort up to 10 000 diodes per hour.

Capacitor tests. Figure 4 shows an automatic system for the testing and sorting of capacitors. The input, adaptor, and switching functions are performed by a manually loaded transport unit that grips the capacitor

terminals and connects them to various soaking, measurement, and discharge busses. The transport is stepped by signals from a control unit, which also houses other interface circuits. Signals from the control unit also enable the transport to perform an output function: dropping the tested capacitors into sort bins. Programmable power supplies provide stimulus signals for the measurement of dielectric strength and leakage current, and also provide a conditioning bias voltage for capacitance measurement. An automatic bridge provides both stimulus and measure functions for the measurement of capacitance and loss, and a digital voltmeter measures leakage current. The digital output data of the automatic bridge and digital voltmeter are processed by digital limit comparators to provide sort decisions. The instruments are controlled by a program unit, which includes a card reader to interpret holes in a prepunched card to program voltages, soak time, instrument ranges, and sort limits, and an overriding series of thumbwheel switches to allow manual programming. A further output function is provided by a series of electromechanical counters, which display a running tally of the sort decisions. This system provides a good example of a hardware-controlled system for a specific inflexible purpose.

Another capacitor-testing system, shown in Fig. 5, is used for evaluation and life studies rather than physical sorting. Components are loaded on test boards, which are inserted into a test socket. A reed-relay scanner is used to connect the components to the stimulus and measurement instruments. A minicomputer and interface system control the stimulus and response instruments, process the measurement data, and provide a typewritten output report on a teletypewriter. The system software provides an initial test sequence that produces a printout of parameter values and indicates out-of-limit units, and a final test sequence that provides parameter values, deviations from initial values, and a statistical summary.

Network tests. One passive network of extreme importance is the connection network, which has the

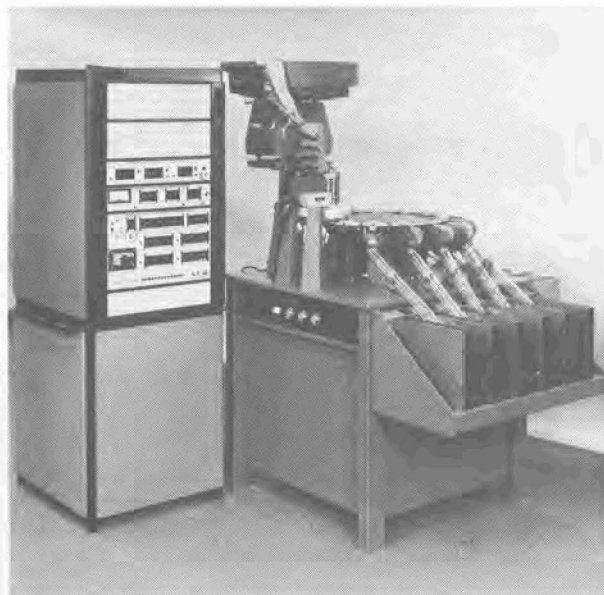
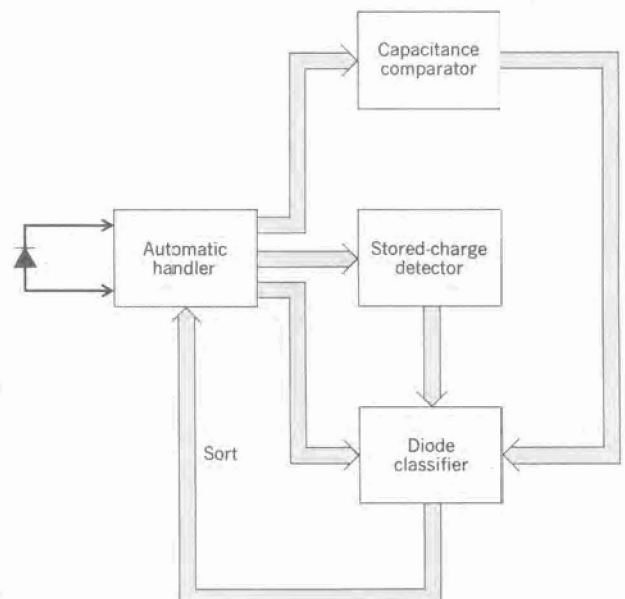


FIGURE 3. Automatic diode-sorting system. (Courtesy Teradyne)



mundane task of connecting some circuit nodes together and isolating others. This type of network appears in many forms—a cable, a wiring harness, a wired backplane or motherboard, a bare printed-circuit board, or a hybrid-circuit substrate before the components are attached. These networks are perhaps the most critical elements in electronic hardware and the ones most prone to error in manufacture, and therefore considerable effort has gone into the design of automatic test equipment to check them out. The stimulus and measurement units are relatively simple, since the measurements required are usually dc continuity, dc or ac dielectric strength (hipot), and dc leakage resistance; however, the control and switching units can get quite complicated. The less expensive systems (\$20 000) use paper-tape readers and bar relays and can handle networks with up to 500 nodes. The more expensive systems (\$100 000) use computers and random-access mercury-reed scanners and can handle networks with up to 100 000 nodes.

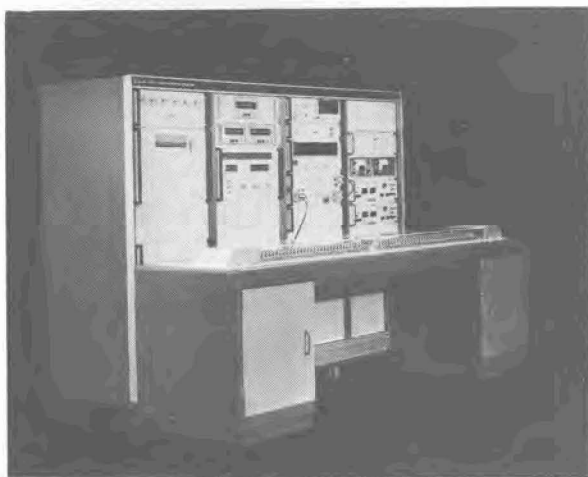


FIGURE 4. Automatic capacitor-sorting system, which measures and sorts capacitors to pre-programmed specifications. (Courtesy General Radio)

Another important passive network is the multipair communications cable. Figure 6 shows two versions of a computer-controlled test set for the measurement and evaluation of the capacitance parameters of such cables.¹¹ (Most of the important transmission characteristics are determined by these parameters.) A functional block diagram is shown in Fig. 7. The input function is provided by a large “fanning fixture” and the switching function is performed by a special-purpose reed-relay scanner. Stimulus and measurement functions are performed by an automatic capacitance bridge and the control and process functions are provided by an interface system and a minicomputer. A teletypewriter provides the output report. The set uses a high-level language to enable an interactive dialogue with the operator to determine test limits and desired output data. Although individual measurements can be printed, in the usual mode of operation all the data are condensed into a compact statistical summary. A typical output report is shown in Fig. 8.

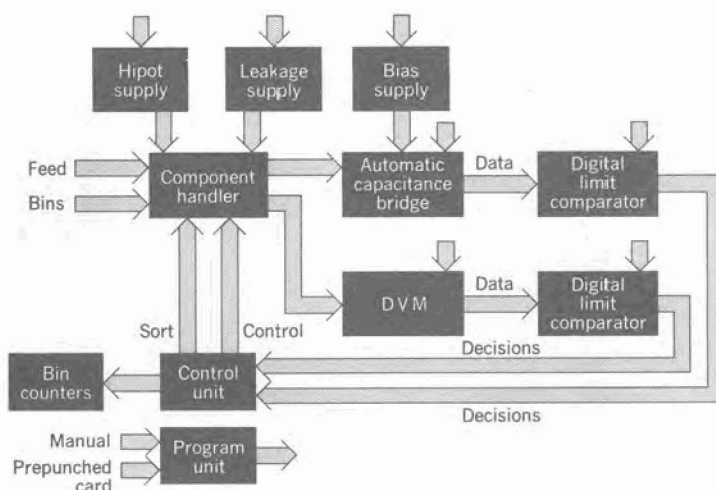
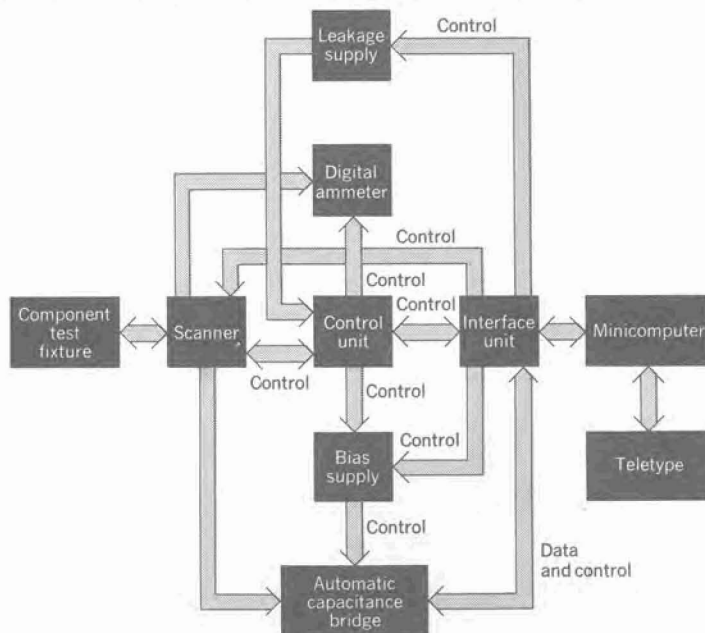


FIGURE 5. Computer-controlled system for life evaluation of capacitors. (Courtesy General Radio)



Circuit tests. Circuit testers usually fall into three categories, depending on the type of circuit to be tested—that is, digital, analog, or hybrid (combination).

Digital circuits. The explosion in the use of digital circuits, both discrete and integrated, has created an entire industry in test equipment with its own evolving terminology and jargon.¹² Digital-logic engineers speak of three basic types of testing¹⁰:

1. Functional testing, to check truth-table or “Boolean” operation of digital circuits by applying and sensing patterns of 1’s and 0’s on the pins of the circuit.
2. Parametric testing, to check dc or static parameters, such as forward and reverse currents, saturation and offset voltages, etc.
3. Dynamic testing, to check time-domain parameters, such as rise and fall times, propagation delay, etc.

FIGURE 6. Automatic systems for the measurement of capacitance parameters of multipair communication cable. (Courtesy General Radio)



Inexpensive (\$10 000) functional testers now available apply a sequence of bit patterns to the input pins of both a known good circuit and the circuit under test and compare the output patterns. Only one input signal is altered between consecutive tests to avoid indeterminate states. As one author points out, however,¹³ since it takes at least 2^n tests to provide a complete input sequence for a unit with n interdependent inputs, it might take 40 000 years to test a 60-input device, even at a 2.5-MHz test rate! The answer, of course, is to bring out internal logic nodes as test points to reduce the number of interdependent points to ten or so and the test time to milliseconds. Such testers are useful in the rapid sorting of bad from good units but are of little help in diagnostic repair.

The more expensive (\$20 000 to \$100 000) testers use tape readers or computers to exercise a predetermined test program for both sorting and diagnostic information.

Figure 9(A) is a photograph of the General Radio 1790 Logic-Circuit Analyzer, a high-speed functional tester for digital circuits.¹⁴ This system, shown in block-diagram

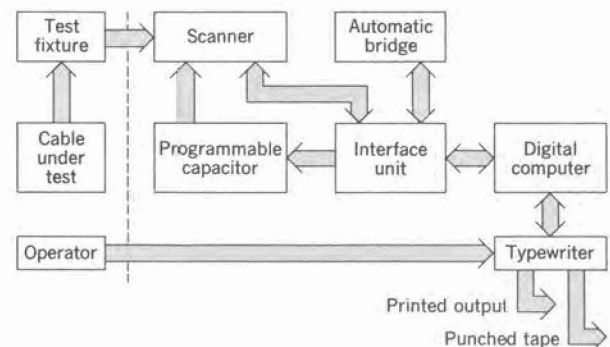


FIGURE 7. Block diagram of cable capacitance-measuring system shown in Fig. 6.

FIGURE 8. Statistical summary of results from cable capacitance-measuring system of Fig. 6.

```

IDENTIFICATION INFORMATION:
135382 ADMC-1500 L2 RD-WH UNIT #4 6/28/69

PAIRS: 100
LENGTH (FEET): 750
SELF TEST: OK

MUTUAL
ERROR THRESHOLD: 3.020 NF
MUTUAL DISTRIBUTION - CELL NOS ARE NF/MI +/-1
  L   71   73   75   77   79   81   83   85   87   89
    0   0   0   0   0   1   13   16   29   13   19
  91   93   95   97   99  101  103  105  107  109  H
    5   2   1   1   0   0   0   0   0   0   0

NUMBER OF PAIRS TESTED: 100      COEFF. OF DEVIATION: 3.987%
AVERAGE: 85.777 NF/MI
STD. DEVIATION: 3.32 NF/MI      LIMITS: PULP XXX PIC XXX

UNBALANCE TO GROUND DISTRIBUTION - CELLS ARE 100 PF/1500 FT WIDE
  1   2   3   4   5   6   7   8   9   10  11  12
  21  17  17  13  8   6   8   5   3   0   2   0
  13  14  15  16  17  18  19  20
    0   0   0   0   0   0   0   0

STD. DEVIATION FROM BAL.: 430 PF/1500 FT
AVERAGE UNBAL.: 372 PF/1500 FT
PAIR 32: +1910 PF/1500 FT +980 PF
PAIR 47: +1960 PF/1500 FT +915 PF

PAIR-TO-PAIR UNBALANCE
CAP. UNBAL. LIMIT: 124.207 PF SKIP LIMIT: 13.799 PF (11.11%)
CAP. UNBAL. DISTRIBUTION--CELLNOS ARE CELL MAX. % OF CAP. UNBAL. LIMIT
  2.5  5   7.5  10  12.5  15  17.5  20  22.5  25
4597 135  79  38  36  18  17   9   6   6
  27.5 30  32.5 35  37.5 40  42.5 45  47.5 50  H
    2   3   2   0   2   0   0   0   0   0

STD. DEVIATION FROM BAL.: 3.644% OF CAP. UNBAL. LIMIT
NUMBER OF TESTS: 4950      COMBINATIONS SKIPPED: 4358

SELF TEST: OK
END OF TEST

```

form in Fig. 9(B), is an example of a test set developed by a manufacturer out of necessity for in-house use and then offered for sale after prove-in. The set includes a mini-computer for control and processing, a control panel, tape reader, display oscilloscope, and teletypewriter for operator communication and printout, a device adaptor for connecting the circuit under test, and an interface system that provides logic-level drive and sense circuitry for digital circuits with up to 240 pins.

The system communicates with the operator in four ways. The operator control panel contains control buttons and switches to control the system operation, and lamps to indicate the go/no-go results of a test. A display oscilloscope serves as alphanumeric output of test conditions and results, and is also used to display instructions to the system operator. A teletypewriter and paper-tape reader provide keyboard input, paper-tape input and output, and printed output where a permanent record of test results is desired.

A key feature of the system is the use of a high-level test-set language that allows test programs to be prepared by technicians familiar with the terms of digital-logic circuits. The language contains English-like commands,

which perform functions such as the following:

1. Set specific inputs or outputs high or low.
2. Check specified outputs against output patterns specified in the program.
3. Generate program "loops" to allow selective repetition of part of a program.
4. Branch to another place in the program if specified output patterns exist.
5. Print instructions or error comments to the operator.
6. Insert specified delays in the program.

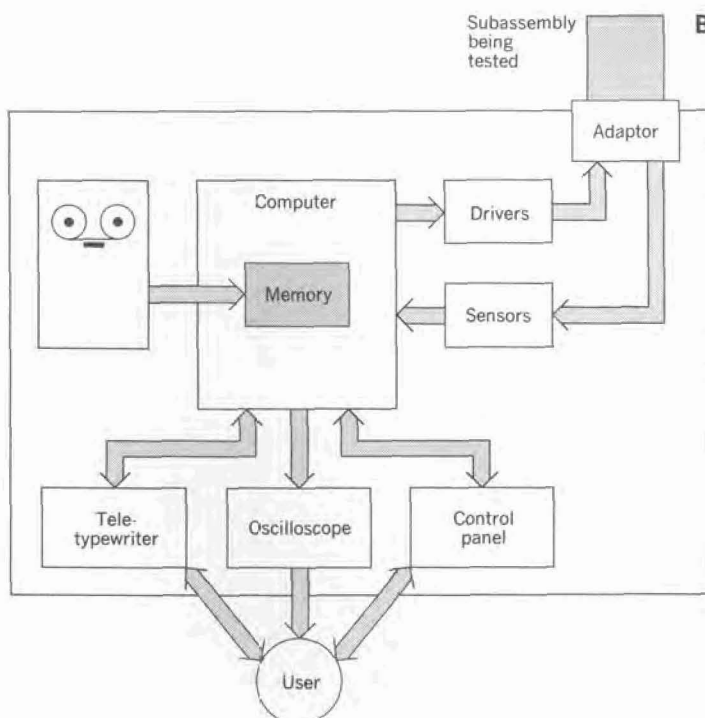
A typical program is given in Fig. 9(C). Two types of programs may be prepared: a simple truth-table program to allow quick sorting of good and bad circuits or a more complicated diagnostic program to facilitate troubleshooting and repair. Programs can be prepared either on line or off line on paper tape. An interactive interpretive mode provided ("combined interactive system") allows generation, translation, and execution of a test program on line. An "autoprogramming translator" permits the use of a known good circuit to ease program preparation by automatically recording output states.

LSI circuits. Continuous evolution in the technology of integrated circuits, particularly in the manufacture of large-scale integrated arrays, has sparked much interest in automatic test systems, as well as some controversy.^{16, 18} The development of the devices to be tested is running somewhat ahead of the development of the test equipment.

The title illustration shows a typical test system used in this area, the Teradyne J283 Circuit Test System, called the "slot machine" because of its application in the testing of sequential logic. The system consists of three free-standing racks or "kiosks," a teletypewriter, and one or more test decks. One rack contains a minicomputer and magnetic-tape transport. Another contains a control



A



B

FIGURE 9. Automatic system for functional testing of digital logic circuits. A—System. B—Block diagram. C—Typical program. (Courtesy General Radio)

C

```

/BOARD NUMBER 1790-4140
/USE ADAPTOR 4
/PROGRAM REVISION 1
*I (1, 2, 14, 15, 17)
*0 (20, 21, 22, 23, 30)
1; IH(1) OH (22, 23)
2; IL (14, 2)
PRINT MOVE S1 TO POSITION 5
AND PRESS CONTINUE
PAUSE 1
DO 5, 30
3; IH (2) IL (17) $
4; IL (1) OH (30)
5; IH (1) IL (2) OL (30)
DELAY 100
IGNORE (21, 22)
6; OL (20)
IF (23) 7
PRINT PROBE TEST POINT 1 AND IF HIGH,
THEN IC8 IS BAD
OTHERWISE CHECK IC6 !
PAUSE 2
.....
END

```

unit, a CRO display unit, and a dc parametric measurement unit. The third rack contains multiplex relays, buffer amplifiers, and the drive and sense circuits for functional testing.

The system can perform functional and parametric tests on digital circuits with up to 120 input/output terminals over a range of ± 30 volts. The functional test rate is 50 kHz. A clock-rate tester is available in a separate kiosk to permit functional testing of dynamic MOS circuits with clock rates of 10 Hz to 8 MHz. An independent dynamic test system can also be incorporated to provide measurements of propagation delay and rise and fall times on 48-pin devices. Programs are prepared on magnetic tape, using the teletypewriter and display unit and a high-level test-set language.

Analog circuits. Figure 10 shows an example of the Hewlett-Packard 9500 Series Automatic Test System,¹⁷ used to test analog circuits. A system is configured from a selection of building-block modules, including computers, stimulus sources, measurement units, processing units (called converting and conditioning units), switching units, and output units—most of which are commercial test instruments. Also shown is a block diagram of one of the simpler configurations. Figure 11 gives a program used to test the gain versus frequency of an audio amplifier. The program language is HP Basic, a modified version of the "Beginners All-Purpose Symbolic Instruction Code" developed at Dartmouth College in the mid-sixties. The software includes a 5500-word interpretive compiler and "software-driver" routines for the hardware elements.

Figure 12 shows the Instrumentation Engineering System 390, a computer-controlled system for the testing of printed-circuit boards—blank, digital, analog, or hybrid—or other forms of circuits. The system is another

example of a modular building-block approach that allows a system to be configured from a selection of stimulus, measurement, switching, control, and output units. The system software uses Atlas language and can be used in a compile-before-run mode or in an interpretive direct-run mode.

Military systems. As is true of almost all our technology, many significant advances in automatic test systems have been spurred by the requirements (and the money) of the military services. Many military contractors (Northrop, LTV, General Dynamics, etc.) have furthered the art of automatic testing. Although this article is not directed specifically to such equipment or such applications, a brief description of one program may be useful.

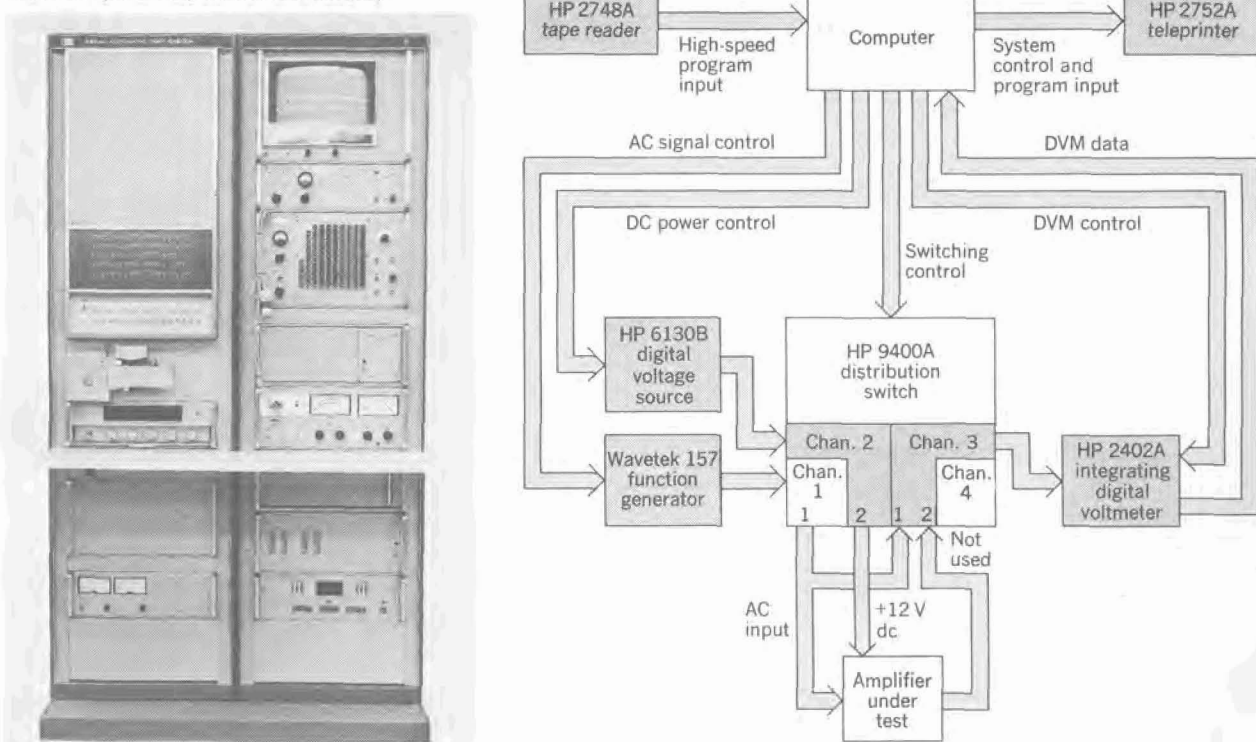
The VAST (Versatile Avionics Shop Test) system^{18,19} has been developed by PRD Electronics over the past ten years or so under several Navy contracts. About \$100 million has been spent to date.

The system is intended for maintenance tests, both aboard ship and at supporting shore sites, on the avionic equipment in Navy aircraft, particularly the new Lockheed S-3A, the Grumman E-2C and F-14A, and the LTV A-7E. This project influences not only the testing of the avionic equipment but also the basic design and packaging of the equipment itself, since the suppliers of the aircraft and avionics equipment will be required to provide interface devices (device adaptors) and compatible test programs, and demonstrate that the equipment can indeed be tested and diagnosed on the VAST system.

Figure 13 shows a typical VAST test station. The hardware contains three basic sections:

1. The computer subsystem, which includes a general-purpose digital computer and two magnetic-tape transports.

FIGURE 10. Automatic test system for analog circuits. (Courtesy Hewlett-Packard)



2. The data transfer unit, which represents the operator/machine interface. This unit includes a control panel with mode-select buttons, status indicators, and an ASCII keyboard; a display panel that provides a cathode-ray tube presentation of test data and instructions; and a maintenance panel to monitor autocheck results.

3. The stimulus and measurement section, which includes the basic building blocks used to configure a particular test station. These blocks are chosen from a list of about 50, which are divided into three categories: stimulus, measurement, and operational. The stimulus blocks include dc and ac power supplies, signal generators, function generators, pulse generators, delay generators, word generators, etc. The measurement blocks include digital voltmeters, frequency meters, power meters, impedance meters, spectrum analyzers, etc. The operational blocks include switching and interface modules.

Several sophisticated software techniques have been developed for the VAST system.²⁰ A high-level language, Vital (Vast Interface Test Application Language), is an extension of Atlas, which was developed in 1968 by Aeronautical Radio, Inc. The new language includes several sublanguages, or "dialects," to obtain the ease of programming of a high-level language but still retain the flexibility of harder-to-use low-level languages.

Programs are prepared off line. Several compiler programs are used to convert the dialect programs into VOSC (Vast Operating System Code) at a compiling

station, which includes peripheral items and a PDP-8 minicomputer. The compiling station is in turn linked to a remote Univac 1108 computer. Source programs are entered on punched cards and are compiled into an object program (in VOSC) on magnetic tape. This object program is entered into the VAST test station along with an operating-system program that provides control of test-station operation and interprets the instructions of the object program during a test run.

Test complex. Most of the systems just described are "stand-alone" test systems. Some can include a few multiplexed test stations but in the main they are autonomous units. Another concept being considered, and in some cases implemented, is the test complex consisting of a central control station and remote test centers. Each test center can include a master test station and several "satellite" or "slave" test stations. The approach is intended to minimize cost by "putting all the computer in one place and developing the software once and for all." In practice, however, it has been found that the minicomputer itself is often the most economical item to use in a remote test center to communicate back to a central station. It also allows some degree of autonomy at the test center in case the central unit goes down.

Such a test complex is in operation at the Raytheon plant in Andover, Mass.²¹ It is called IHATS (Improved Hawk Automatic Test System) and is used for factory test of missile components. Each test station includes a

FIGURE 11. Programs for system shown in Fig. 10. A—Test result. B—Program example.

AMPLIFIER FREQUENCY RESPONSE TEST
SPECIFIED GAIN IS 10, TOLERANCE 1.7%

CONNECT AMPLIFIER
SERIAL NUMBER IS?12

FREQUENCY	GAIN	% ERROR	
1000	9.8424	-1.576	
2000	9.85134	-1.48661	
3000	9.85929	-1.40711	
4000	9.83948	-1.60521	
5000	9.84841	-1.5159	
6000	9.81771	-1.82291	OUT OF SPEC
7000	9.82665	-1.73349	OUT OF SPEC
8000	9.82665	-1.73349	OUT OF SPEC
9000	9.8247	-1.75301	OUT OF SPEC
10000	9.83363	-1.6637	

```

10 PRINT "AMPLIFIER FREQUENCY RESPONSE TEST"
20 PRINT "SPECIFIED GAIN IS 10, TOLERANCE 1.7%"
30 PRINT
40 G = 10
50 DVS (1, 0, 10)
60 PRINT "CONNECT AMPLIFIER"
70 PRINT "SERIAL NUMBER IS";
80 INPUT S
90 PRINT
100 PRINT
110 SWH (1, 2, 1, 0)
120 DVS (1, 12, 100)
130 PRINT "FREQUENCY", "GAIN", "% ERROR"
140 FOR F = 1000 TO 10000 STEP 1000
150 SYN (1, 1, F, .283)
160 WAIT (50)
170 DVM (2, 1, V1)
180 SWH (1, 2, 2, 0)
190 DVM (2, 10, V2)
200 E = ((V2/V1-G)/G)*100
210 PRINT F, V2/V1, E
220 IF ABS(E)>1.7 PRINT "OUT OF SPEC"
230 PRINT
240 NEXT F
250 END

```

A

B



FIGURE 12. Computer-controlled system for testing hybrid circuits. (Courtesy Instrumentation Engineering)

FIGURE 13. VAST test station. (Courtesy PRD)



collection of stimulus, measurement, conditioning, switching, adaptor units, etc. (usually commercial items) plus specially designed interface circuits for control and communication with a "line computer" on a multiplexed basis. Several line computers in turn communicate with a central "collection computer" for data collection on magnetic tape. Programs are prepared and data are analyzed on an isolated "utility computer" and associated peripheral units. The system uses a special compiler language, TCOMP, based on Fortran IV, which allows program preparation by test engineers rather than programmers.

Cost. The prices of commercial automatic test systems seem to cluster about decade and half-decade values (1:3:10), with, of course, a continuum in between. Table II shows what to expect.

How to automate successfully

There are two basic approaches to achieving automation: the do-it-yourself approach and the buy-it-outside approach, with several gradations in between. In the do-it-yourself approach you attempt to do as much of the job in house as your time, talent, and budget allow. (Of course you have to buy something—from wire and solder to ICs or complete instruments or subsystems—but the basic responsibilities for design, procurement, fabrication, assembly, test, documentation, installation, training, and maintenance are yours.) In the buy-it-outside approach you attempt to contract for as many of these services as possible. Either approach can succeed or either approach can fail, depending on how you go about it. The following paragraphs discuss techniques that can make the second approach successful.

Getting started. First, study your task. Try the "man-from-Mars" technique and mentally step back as far as you can to visualize your operation as part of an overall system. Then gradually narrow your view to specific operations. Where does the device to be tested come from? Where does it go? What paper work goes with it? Why? What happens to it? What tests are to be performed? Why? What accuracies are required? Why? What reports are required? Why? This exercise can sometimes disclose unnecessary or redundant steps or operations no longer required because of changes in other parts of the organization. Or it may disclose new or impending requirements.

The chickens in the other yard. Find out what other people with similar problems have done. Read, study, telephone, visit. Present your problem informally to a few potential suppliers to get their ideas. Don't tell them how to do it; tell them the basic task. You can suggest ways but don't, at this stage, demand them. Others may have different ideas based on previous experience; you're not committed to follow them.

Put it in writing. Prepare a written specification for your automatic system. This exercise will clarify your thinking and provide an essential tool for you and the supplier. Again, specify what the system must do rather than how to do it. Separate your specifications into "must have" and "would like." In other words, don't demand 0.1 percent accuracy if 1 percent will do; don't demand four-week delivery if 12 weeks will do. You can get almost anything you want, but cost and time will increase exponentially as specifications get tougher.

Get competitive bids. Send your specification to as many vendors as you think are qualified to do the job.

Give them time to absorb the request and invite them in to discuss it to make sure they understand the requirements. Keep prospective vendors separate unless you can handle a bidders' conference in a firm, unbiased manner.

In some ideal utopian world a prospective buyer would get a group of three or four prospective vendors together and say, "Look, fellows, I've got \$37 500 and this is the job I'd like done. What can you give me?" The vendors would leave and come back later with their best proposals—all priced at about \$35 000 (with optional extras to \$45 000).

Or, better yet, he might say, "If I can get a machine to do these things in four seconds it's worth \$37 500 to me. If you can get it down to two seconds it's worth \$75 000!"

In the real world this does not happen often. A buyer is afraid that the vendors will combine their efforts to milk him of his money for a machine of lesser value. This, of course, is a small danger, since most vendors are engaged in a competitive life-or-death struggle and have no intention of cooperating with each other.

Get it in writing. Request a written proposal from your prospective suppliers in addition to a quotation. The proposal should list and describe the equipment to be provided, both electrically and physically. It should define the specifications of the system as a whole. It should describe the operation of the system from the operator's point of view (rather than the designer's). It should describe the other services provided, such as installation, training, warranty, and repair. It should include a list of recommended spare parts. Get as many competitive bids as possible.

Evaluate the bids. When you open the bids you'll be surprised. The prices will be higher than you expected, the accuracies will be poorer, the speed will be slower, you'll get less equipment, the equipment will be harder to use, and you'll get fewer services. Don't be dismayed; the approach can still succeed.

There are four basic considerations in evaluating the bids: the equipment, the supplier, the economics, and yourself.

The equipment. Lean toward standard off-the-shelf items whenever possible. Such items are usually well debugged and conservatively specified, and will have a good backup of instruction and maintenance manuals, available spare parts, and trained service technicians. You may require some modified or specially designed items, but it is important to appreciate the differences between these items and standard items. In this area you are highly dependent on the integrity of the supplier.

The supplier. You should thoroughly evaluate the

II. Commercial automatic test systems

Price Range, dollars	Description
10 000–30 000	Hardware systems of dedicated capability. Sometimes with tape readers, patchboards, or other forms of semiautomatic control. Printed or machine-readable output
30 000–100 000	Computer-controlled systems. One or more test stations and processed output
100 000–300 000	Computer-controlled with several test stations. Flexible capability and completely processed output
300 000–1 000 000	Multicomputer installations. Central control with remote test centers and satellite test stations

supplier. Where is he? What are his local sales and service facilities? What's his reputation? What else has he done like this? How did it work out? If you are contemplating a major expenditure, visit the suppliers' facilities. Talk to the people who will be doing your job.

The economics. Although you have probably made some form of economic analysis to establish your budget, do it again based on the actual bids and the actual throughput promised. Appendix B describes some methods used to compare investments. All methods require an estimate of the initial investment and the first-year savings.

The cost of the initial investment includes not only the billed price of the equipment but also the additional costs of installation, training, program preparation, spare parts, etc. As a rule of thumb, these costs can add 10 to 15 percent to the cost of the equipment.

The computation of operational savings requires an estimate of the cost of doing a job with the new equipment versus doing it the old way. The following factors are involved:

1. Number of units to be tested.
2. Number of types to be tested.
3. Average failure rate.
4. Old labor costs, including overhead.
5. New labor costs, including overhead.
6. Preparation time for test procedures.
7. Sorting good units from bad.
8. Troubleshooting and repair of bad units.
9. Tooling and maintenance.

Table III shows a savings calculation for the testing of logic-circuit boards. The comparison is between the use of manual test equipment and an automatic system. The results illustrate two things. First, the automatic equipment saves \$20 742 in doing a year's work; second, it uses only 398 hours in doing it. This reduction in time is, of course, the reason for the cost savings, but it also shows that the machine will be sitting idle most of the time. It will be underutilized and has the capacity to provide greater savings during the year if the work volume increases.

Armed with such an estimate, you can find an accounting expert who can calculate the payback period or discounted rate of return or net present value. You will usually find that you are losing money every day you are without the system—even a high-bid one. Amazingly enough, this is generally true (at least in the type of investment discussed here).

Yourself. You should evaluate yourself along with the bids. Can you assign a system "father" who will learn and master the system and nurture it as his own? (If it's foisted upon him by the "front office," it will never work.) Do you have, or can you develop, the capability to get the most out of the system and keep it running? Will you train new people as the present people get promoted or leave?

Follow-through. Once you've made up your mind and placed the order, get behind it solidly. Keep in touch with the supplier and work with him. Agree on an acceptance-test procedure—the simpler the better. If possible, visit his plant to look at and try the equipment before it is shipped; last-minute modifications are accomplished much easier at his plant than at yours.

Pilot operation. When your new system is installed, don't put it into full critical use at once; drive it slowly for the first "1000 miles." You'll find bugs, either in

your operation of the system (cockpit errors), or in the set itself—early-life failures or some combination of switch positions that the designer never thought of. Work these problems out patiently and smoothly and get your operators used to the system. Then you can put it into use with full confidence.

Service and maintenance. If the system is really to do a job for you, keeping it running is of prime importance. Most vendors provide free service at your plant for 90 days. This usually works well since most of your problems will occur early and, if the system is in trial use, the few days (or weeks) it takes them won't hurt you much. Beyond the 90 days, service at your plant will take time and will cost you—unless you've signed up for a service contract (for about 1/2 percent of the system price per month) or unless you're willing to diagnose the difficulty and send the defective item back to the vendor. If the system is in critical use and down time is disastrous, the best approach is to fix it yourself. This requires people who know the system inside out and a good stock of spare parts. The spare parts can range all the way from a few relays to spare circuit boards to spare instruments.

Conclusion

This article has attempted to survey the state of automation in electronic testing and to discuss some of the characteristics of current approaches. The message is simple: automation is desirable and everyone should have it. Get it any way you can. Rent it, lease it, buy it, make it. It's not as easy to achieve as it may seem (nothing worthwhile is), but neither is it as hard. The rewards can be startling and will be in direct proportion to the effort put in.

III. Operating savings analysis, logic-circuit boards, annual cost

	Manual	Automatic
Preparation:		
Number of types	10	10
× hours per type	× 40	× 16
Total hours	400	160
× cost per hour	× 15	× 15
Cost	\$6 000	\$2 400
Sorting:		
Number of boards	5 000	5 000
× minutes per board	× 10	× 0.25
× hours per minute	× 0.017	× 0.017
Total hours	833	20.8
× cost per hour	× 15	× 10
Cost	\$12 500	\$208
Troubleshooting and repair:		
Number of failures	1 000	1 000
× minutes per failure	× 30	× 10
× hours per minute	× 0.017	× 0.017
Total hours	500	167
× cost per hour	× 15	× 15
Cost	\$7 500	\$2 500
Tooling and maintenance:		
Total hours	100	50
× cost per hour	× 10	× 15
Labor cost	\$1 000	\$650
+ material cost	500	1 000
Cost	\$1 500	\$1 650
Total cost	\$27 500	\$6 758
Total hours	1 833	398
Cost savings		\$20 742

Appendix A.

Software languages; some definitions*

Figure 14 is a generic block diagram applicable to all computer systems. Any system can be considered as a subsystem consisting of some of the elements in the diagram. Let us consider the function of each element, beginning with the lowest.

Base machine. The base machine consists of registers and decision-making circuits that perform the basic data-processing operations. This machine is controlled by signals for opening or closing gates to allow the data to flow between the machine registers. These signals are sometimes called the *microprogram* of the machine, and the language that describes this microprogram is called the *register-transfer language*. Programs for basic machines are very complex and are usually determined only by the computer manufacturer; recently, however, machines that can be microprogrammed by the user have been introduced.

Emulator. The emulator block usually consists of hardware decoding circuits or a read-only memory, which decodes certain bit patterns, called the *machine code*, to operate the base machine. These circuits determine what is usually called the *instruction set* of the machine. Instructions consist of a pattern of bits in an instruction word. Some special machines use a string of ASCII characters as the instruction set, so they may be programmed directly from an I/O device, such as a teletypewriter. A few machines have been built whose machine language is a high-level language. The emulator and base machine together make up what can be called the *real machine*. This is what we generally buy from a computer manufacturer.

Interpreter. The interpreter block defines a program, sometimes called an *operating system*, or *simulator*, which can be thought of as a means for converting the real machine into a new machine, called a *pseudo-machine*, whose effective machine code is different from that provided by the real machine. The machine code for the pseudo-machine, called *interpretive code*, is usually chosen to make the pseudo-machine perform more directly those functions desired by the application. As a consequence, the program in interpretive code requires fewer bits of memory than the same program would require in machine code directly.

This efficiency in memory space is obtained at the expense of both speed and flexibility. The pseudo-machine usually executes the program at a slower rate than would occur with the same program coded in machine code. Also, the pseudo-machine will tend to be less general than the real machine. The interpreter is designed for the effective performance of certain functions needed by the class of programs for which it was intended. Consequently, it performs certain other functions poorly and, in fact, it may not allow the programmer to perform still other functions at all.

Program store. The *program store* block consists of the medium in which the program is stored. It may be either core memory or paper tape.

Translator. The *translator* takes the source code generated by the programmer, often performs some

rudimentary checking for errors, and translates the programmer's statements into interpretive code. Note that translation is not part of the real-time operating environment. It is performed once when the program is written, either on the machine in question or some other machine of a different type, or even manually by the programmer. If the translation is performed mostly by machine, the source language that the programmer uses can use terms very close to the terms in which he thinks about the problem. If this is the case, he is using *high-level language*.

Programmer. If the machine translation process is relatively simple (such as a basic assembler), then the *programmer* must first translate his programs into the language needed by the translator program. The closer the terms of the source language are to the terms in which the programmer considers the problem, the higher the level of the language. The further the programmer has to translate his thoughts into the terms of the machine, the lower the level of the language.

Degenerate cases. There are many practical systems where various parts of this hierarchy seem to be missing. In these cases the functions are actually absorbed by some other block. Some examples may illustrate this.

Missing interpreter. In many systems the interpreter or operating system block is missing. This occurs with most of the software provided by minicomputer manufacturers with their systems, including the following:

1. The assembler program requires the programmer to translate his problem into a language very close to machine code. It further translates his assembly-language code to machine code. When writing in assembly language the programmer has access to all the machine functions provided by the machine designers. For this reason, good assembly-language programmers can write efficient programs that use the machine resources efficiently. In small-computer systems most of the programs are usually written in assembly language, since efficient use of the limited machine resources is very important.

2. A Fortran compiler translates a higher-level language, Fortran, into machine code. This enables the programmer of algebraic problems to think more in terms of the problem and leave most of the translation to the Fortran translator or compiler. Good compilers generally use machine resources more efficiently than a poor assembly-language programmer and less efficiently than a good one.

Missing translator. There are several important software systems in which the translator appears to be missing. In these systems the program is stored directly in the source language of the programmer. Examples are most uses of Basic and Focal. In these systems the program will take much more memory space than a translated program, since the source language contains much redundant information. In addition, the system will usually run slower since each program statement must be processed each time it is executed. These systems have the advantage of being easy to change on line and are often more interactive than those in which translation is needed. There are several programming languages that are used in this way during program preparation and debugging. After prove-in, the program is translated into machine code for repetitive use.

Between these examples is a continuum of possibilities. The skill of the system designer determines which is the

* The thoughts in this Appendix were provided by R. G. Fulks of General Radio Company; the words were compiled by the author.

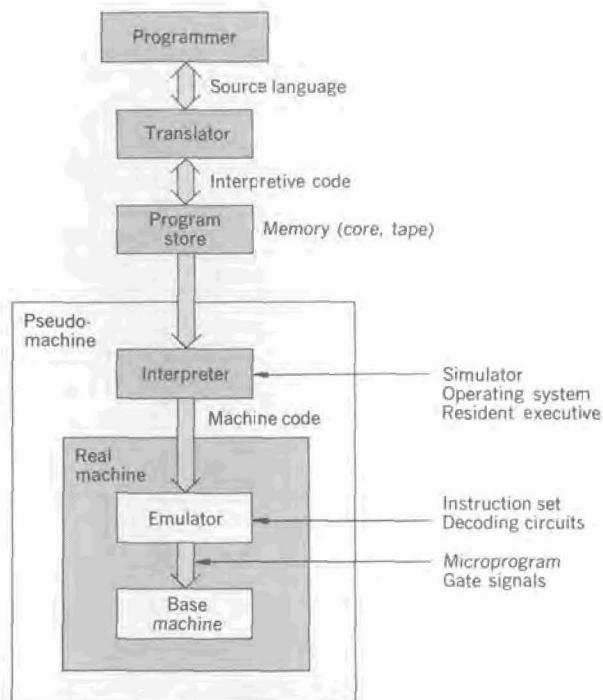


FIGURE 14. Block diagram of a computer system.

most appropriate for a given system problem. The right decision can result in the optimum use of program space and machine time.

Appendix B. Economic justification for automatic test equipment

Justifying an automatic test system from an economic standpoint is relatively easy. In fact, it can be said that automation, at almost any price, is worth it—or soon will be. Looking backward or forward ten years from any point of our lifetime we see a smooth increase (with time) of the cost per unit of production using skilled labor only, and a decrease of the cost per unit using a combination of semiskilled labor and automatic equipment. This seems a paradox, since the cost of automatic equipment itself might seem to be related to the increasing cost of the labor that goes into the equipment. That it is not is due to the insatiable demand for the products of technology, which causes, in turn, an increasing demand for automatic equipment and brings to bear the forces of “the economies of scale”; that is, the more items you make, the less their unit cost. Although the cost of the equipment goes up (after the initial decline from the first few developmental models), the efficiency of the equipment goes up even faster. This is further illustrated by the paradox of the equipment cost/product cost ratio. When diodes sold for \$1.50 each you could buy a diode tester for \$300. Now that diodes sell for a few pennies apiece, the test equipment costs \$100 000! Moreover, the \$100 000 testers have helped achieve the price reduction of the diodes.

The real problem in automatic testing is not in justifying automatic equipment, but in selecting which automatic equipment to buy. Even so, we are still called upon to justify such expenditures to our supervisors. The

following paragraphs may help.

The steps in justifying the purchase or construction of an automatic test system are identical to those in justifying any capital investment. Basically you have to compare the money saved by the new machine with the cost of the machine, and establish that the use of the money required to buy the machine is better than any other use you can think of. The preceding sentence makes it sound easy, but a rigorous analysis can get quite complicated, for here we leave the comfortable world of volts and ohms and enter the strange world of our professional colleagues in the accounting department, with its tax shields and discounted cash flows. We quickly find that what we thought was a dollar isn’t a dollar at all. It may be a half dollar, or 75 cents, or even \$1.25! This is the result of two aspects of our economic system: the corporate income tax and the time value of money. If you’re a profit-making corporation, you can view any additional dollar of operating expense or saving in its incremental effect on the profit of the corporation. If you “save” a dollar you must give half of it to the Federal Government as tax and you wind up with only half a dollar. Similarly, if you spend a dollar for operating expense, you reduce your tax by half of it and you’re only out half a dollar. Thus some of the operating savings and expenses in the use of capital equipment are viewed at half their value (tax shield).

The time value of money, on the other hand, shows us that a dollar to be saved during the next year is not worth a dollar today. In fact, if we can invest money and earn 10 percent on it, that dollar return spread over next year is worth only 95 cents today. Therefore, the time distribution of savings and expenses must be considered.

Three basic methods are currently used to estimate the relative profitability of capital investments: return on investment, payback period, and discounted cash flow.^{22, 23}

Return on investment. Several methods are used to estimate the percentage rate of return for a capital investment (in order to compare it with other possible investments). A simplified version is the MAPI formula,²⁴ wherein you use the estimated savings in the first year of operation to determine an “urgency rating,” the percentage return in the first year. The following calculation shows the urgency rating for a \$30 000 test system that will provide savings of \$15 000 a year (based on ten-year straight-line depreciation).

Operating savings:	\$15 000
Less depreciation:	3 000
Net operating advantage:	12 000
Less added tax:	6 000
After-tax return:	\$ 6 000
Urgency rating =	$\frac{6000}{30000} = 20 \text{ percent}$

Payback period. The payback period is the time necessary for the savings (after taxes) to pay back or make up for the initial investment. The shorter the period, the better the investment. Rather than savings, total cash flow is used; that is, depreciation charges are added to the actual savings, since these charges are an additional source of cash. (Depreciation dollars never leave the company; you’ve already paid for the machine.) Table IV compares two investments on this basis, a \$30 000 system that will save the company \$15 000 a year

IV. Payback-period investment approach

Investment	Operating Savings	Depreciation	Net Return After Tax	Cash Flow	Payback Period, years
\$30 000	\$15 000	\$3 000	\$6 000	\$ 9 000	3
50 000	20 000	5 000	7 500	12 500	4

V. Discounted cash flow technique

Year	Savings	Depreciation (sum of years)	Net Cash Flow After Taxes	Discounted Factor	Value at 20 Percent Return
1	\$15 000	\$10 000	\$12 500	0.909	\$11 363
2	15 000	8 000	11 500	0.751	8 637
3	15 000	6 000	10 500	0.621	6 520
4	15 000	4 000	9 500	0.513	4 873
5	15 000	2 000	8 500	0.424	3 604
Total:	\$75 000	\$30 000	\$52 500		\$34 997

and a \$50 000 system that will save \$20 000 a year.

Discounted cash flow. The discounted cash flow technique takes into account the smaller present value of future returns. The usual method is to tabulate all the future savings, select a desired interest rate (opportunity cost), and reduce the values of the future returns to present values by factors read from a discount table. You then add up all the discounted (reduced) returns. If the sum is greater than the investment, you'll do better than the selected interest rate; if it's less, you won't. Table V shows this calculation for a \$30 000 system that will save \$15 000 a year for five years and a desired interest rate of 20 percent (returns beyond five years and salvage value are ignored).

Since the total discounted value (net present value) of \$34 997 is greater than the investment of \$30 000, the actual rate of return is greater than the 20 percent selected. This technique can be refined by choosing different interest rates and interpolating until the net present value equals the investment, thus determining the actual interest rate.

The preceding methods are fairly straightforward and relatively simple. The procedure can be made a good deal more complicated, however, if all the fine points are accounted for, such as the growth of operating savings from year to year as the equipment is utilized more (more volume, more shifts) or as the cost of labor *not* paid for increases, the salvage value of the equipment if and when you sell it, the interest-compounding period to be used (continuous, monthly, annually), nondepreciated start-up costs, varying tax rates, etc. If your company requires such refined analysis you'd best consult a friend in the accounting department. Some equipment manufacturers provide forms, tables, charts, and even computer programs to simplify the analysis, but, if you use them, make sure you apply the same techniques to all choices.

REFERENCES

1. Lucey, R., and Newth, D., "When to inspect all incoming semiconductors," *Electron. Equipment Eng.*, Mar. 1968.
2. Holt, A., and Stoughton, A., "Guidelines for the purchase of memory testing equipment," *Evaluation Eng.*, July/Aug. 1970.
3. Bartik, J. J., "Common computer interfaces," IEEE Computer Society, Mideastern Area, Cherry Hill, N.J., Mar. 1971.
4. Litzinger, J., "Hardware/software characteristics of a computer-

controlled test system," IEEE Computer Society, Mideastern Area, Cherry Hill, N.J., Mar. 1971.

5. Bobroff, D. A., "Avoid pitfalls in computerized testing," *Electron. Design*, Aug. 16, 1969.
6. Wegner, P., *Programming Languages, Information Structures, and Machine Organization*. New York: McGraw-Hill, 1968.
7. Ross, D. T., "Fourth-generation software—a building-block science replaces hand-crafted art," *Computer Decisions*, Apr. 1970.
8. "IC test equipment buyer's guide," *Solid State Technol.*, Mar. 1970.
9. A.T.E. Reports, Box 746, Camden, N.J.
10. "Electronic industry manufacturing markets report," Theta Technology Corp., Wethersfield, Conn.
11. Fulks, R. G., and Lamont, J., "An automatic computer-controlled system for the measurement of cable capacitance," *IEEE Trans. Instrumentation and Measurement*, vol. IM-17, pp. 299-303, Dec. 1968.
12. *Glossary of Integrated Circuit and Related Terminology*. Motorola Semiconductor Products Inc., Phoenix, Ariz.
13. Johnson, W. R., Jr., "Proving out large PC boards," *Electronics*, Mar. 15, 1971.
14. Fichtenbaum, M. L., "A computer-controlled system for testing digital logic circuits," *NEREM Rec.*, pp. 38-39, 1969.
15. Robinton, M. A., "A critique of MOS/LSI testing," *Electronics*, pp. 62-64, Feb. 1, 1971.
16. Curran, L., "Readers reply on MOS/LSI testing," *Electronics*, Mar. 1, 1971.
17. *Hewlett-Packard J.*, Aug. 1969.
18. Loughlin, R. G., and McCoy, F., "The introduction of VAST," presented at WESCON, Los Angeles, Calif., Aug. 1970.
19. Fuller, E. J., and Proctor, J. A., "The evolution and development of a digital test capability for VAST," presented at the Automatic Support Systems Symposium for Advanced Maintainability, St. Louis, Mo., Oct. 1970.
20. Ellis, M. T., "Vital: A general-purpose multiple-dialect automatic test language," presented at the Automatic Support Systems Symposium for Advanced Maintainability, St. Louis, Mo., Oct. 1970.
21. Gundal, R. K., and King, J. F., "A central time-sharing mini-computer—provides data collection and automatic control over a variety of remote test systems," *NEREM 70 Tech. Appl. Papers*.
22. Ammer, D. S., *Manufacturing Management and Control*. New York: Appleton, 1968.
23. Helfert, E. A., *Techniques of Financial Analysis*. Homewood, Ill.: Richard D. Irwin, Inc., 1967.
24. Terborgh, G., *Business Investment Policy*, Machinery and Allied Products Institute, Washington, D.C., 1958.

Harold T. McAleer (M) received the B.S. and M.S. degrees in electrical engineering from the Massachusetts Institute of Technology in 1953. While in college he was employed as a cooperative student at the General Radio Company working on the design of high-frequency measuring and recording instruments. After two years' service as an engineer with the U.S. Army Signal Corps at Fort Monmouth, N.J., he returned to GR as a development engineer, in which capacity he was engaged in the design of frequency counters and associated instruments. In 1966 he transferred to

GR's Systems Group to work on the design of automatic measurement systems. In 1968 he became manager of custom products at General Radio and is responsible for marketing, design, and manufacture of custom-tailored test systems. He has written several technical articles and is a registered professional engineer and a member of Eta Kappa Nu, Tau Beta Pi, and Sigma Xi.



Reprinted by permission from IEEE SPECTRUM

Vol. 8, No. 5, May 1971

pp. 63-78

Copyright 1977 by The Institute of Electrical and Electronics Engineers, Inc.

PRINTED IN THE U.S.A.

A172

8/1/77

Vol. 14, No. 4 April 1977

PP. 36-40

JN 4047-478

JN581-473

CA172