OCTOBER 1972

HEWLETT-PACKARD JOURNAL

© Copr. 1949-1998 Hewlett-Packard Co.

A Practical Interface System for Electronic Instruments

Connecting instruments into a digitally-controlled system now becomes a matter of plugging in cables. This article describes the interface system that makes this possible.

By Gerald E. Nelson and David W. Ricci

AS MODERN-DAY TECHNOLOGY gives rise to increasingly complex electronic devices, engineers are finding that to test these devices economically, they must resort to automatic systems. And, in automating their test procedures, engineers become more and more involved in instrument interconnection.

To overcome many of the problems experienced in interconnecting instruments and digital devices, a new interface system has been defined. This system gives new ease and flexibility in system interconnections. Interconnecting instruments for use on the lab bench, as well as in large systems, now becomes practical from the economic point of view.

The interface system evolved as part of the trend towards designing digital processing circuits into low-cost instruments. Now the functions formerly performed by the I/O circuit cards used in computers, calculators, and couplers are performed by the instruments themselves. Passive cabling ties the system components all in parallel into a common communications structure in much the same way that a computer or coupler backplane bus ties I/O cards together. The cabling is the only external part of the system. For reasons to be described later, the system is informally referred to as an ASCII-compatible interface bus system.

Interconnecting a variety of instruments, recorders, display devices, and calculators or computers with this new system is simply a matter of plugging in cables. Cable connectors can be stacked in parallel (Fig. 1) or the cables can be daisy-chained in any way that best fits the physical arrangement. System "overhead" is drastically reduced. Changing the system configuration is reduced to adding or replacing devices and reconnecting the cables; no modification of the existing instrument hardware is required.

A significant feature of the system is that there is complete flexibility of information flow. With all devices connected in parallel, any device can talk directly to any other (Fig. 2). A computer is not a necessity for managing the flow of information. One instrument can control another, or an

Cover: *The time is coming when a group of test instruments can easily be made to work as one by linking them together digitally through a new interface system. How the interface works and how it evolved are described in the first two articles of this issue. Except for cables needed to transfer analog signals, only one cable is needed to interface each instrument digitally to the rest of the system, as shown by the snapshot rear view of the instruments on the cover. The third article describes this particular measurement system.*

In this issue:

© HEWLETT-PACKARD COMPANY, 1972

© Copr. 1949-1998 Hewlett-Packard Co.

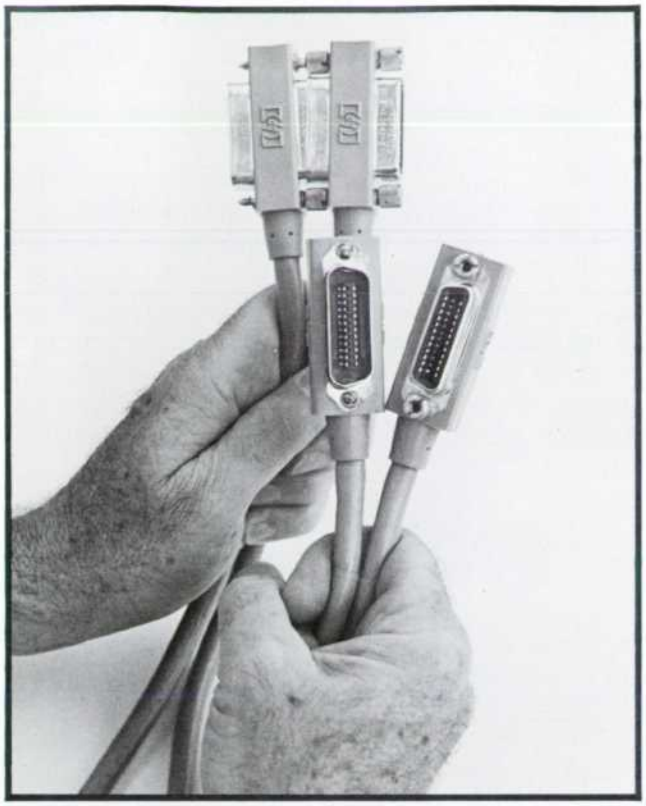


Fig. 1. *Interconnecting cables used with interface system have dual connectors. These can be stacked to accommodate variety of physical layouts by allowing more than one cable to be attached to any device.*

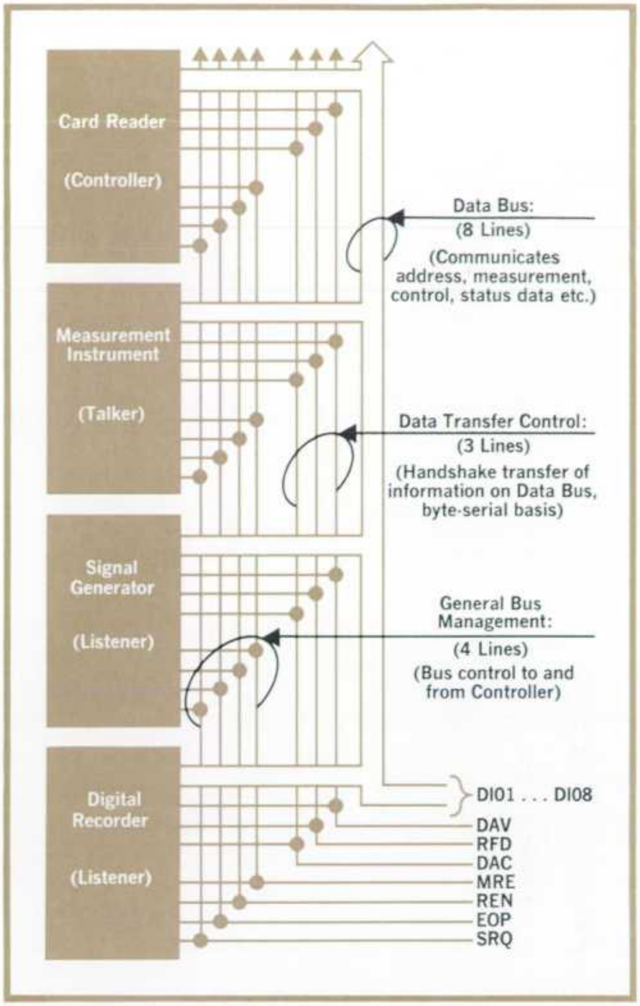


Fig. 2. *Typical interface bus system communications network. Information flow is bi-directional. Because of parallel connection, any device is potentially able to communicate directly with any other.*

inexpensive card reader can control a group of instruments, making it economically feasible to automate instruments for bench or production use where cost has been a deterrent. Incorporating a calculator or computer into the system for more complex operations is also much simpler than ever before.

How Many Lines?

The interface system uses fifteen lines in the bus. Only one line might have been used with all data and control information in serial form, but this would have been far too complex to implement. On the other hand, enough lines to transmit everything in parallel, besides being incompatible with character-serial devices like teleprinters, would have been too cumbersome. For example, the HP Model 3330B Automatic Synthesizer would require about 100 lines for full parallel control.

The new interface system uses the byte-serial approach in which each multiple-bit byte of information is transmitted in parallel, and bytes are transmitted serially. The number of lines is not large enough to make the cabling cumbersome while the circuit requirements for byte-serial data transfer are modest.

The byte-serial approach has other advantages too. For example, with only 15 lines dc isolation of the lines to avoid ground loops is not an expensive undertaking. Furthermore, translation hardware and software costs are minimized since many devices, particularly those involved with the human interface, process data on a character-serial basis.

Line Assignments

The lines are summarized in Table I. Eight of the 15 lines are reserved for data input/output (DIO1 through DIO8), allowing data to be transferred in 8-bit bytes. Eight lines accommodate the 8 data bits of the standard teletype 11-unit code and they accommodate the widely-used 7-bit ASCII code, leaving one bit that can be used as a parity bit if

© Copr. 1949-1998 Hewlett-Packard Co.

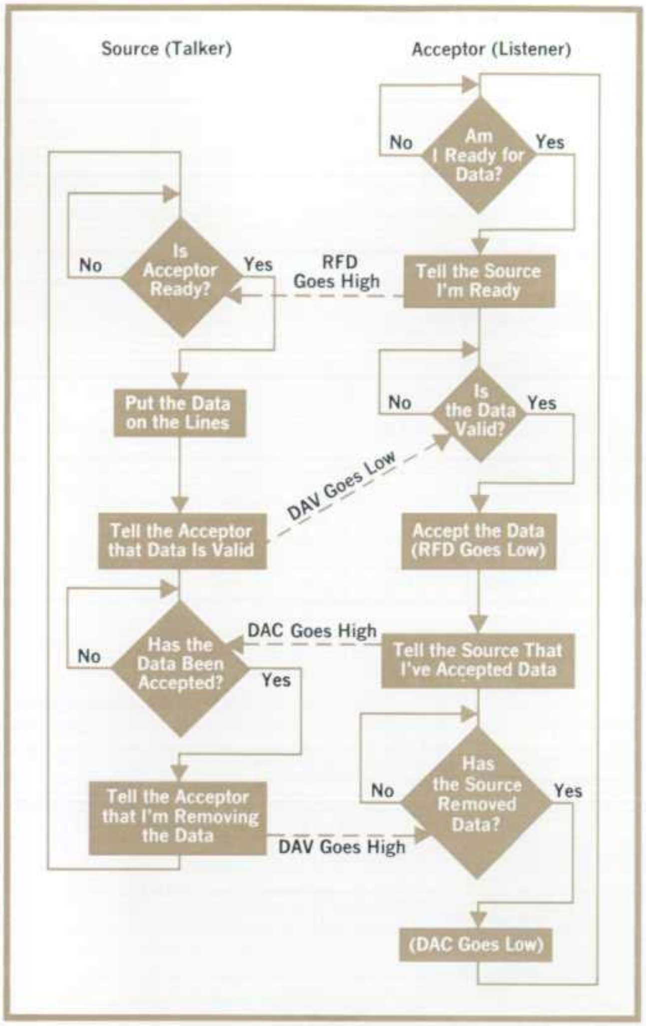


Fig. 3. *Flow diagram outlines sequence of events during transfer of data byte. More than one listener at a time can accept data because of logical-AND connection of RFD and DAC lines (see Fig. 4).*

it should be necessary to transmit data through a noisy channel. Also, most computers use word lengths that are multiples of 8 and there are software advantages in having bus bytes and computer words related in an integral ratio.

The data lines are also used for establishing the role of each device in the activity that is to follow. Each device can play one of three active roles: (1) talker, (2) listener, and (3) controller. A digital voltmeter is a talker, for instance, when it transmits data to a recorder but it becomes a listener when a card reader, acting as a controller, instructs it to change ranges. Unless a device is assigned a role, it remains inactive.

Some means of "ringing up" a device therefore is needed to establish its role. Presumably, one could set aside some of the data line code words as role addresses but with 8 bits there are only 256 different words. Because data transmitted by some devices might include the same words set aside as address words, it is desirable to not place restrictions on the codes carried by the bus.

Hence the new interface system has a control line called Multiple Response Enable (MRE). This line is driven only by the device that is currently designated as the Controller. When the Controller pulls the MRE line low, all other devices must listen to the data lines and only the Controller may then transmit codes. When MRE goes high again, only those devices that were addressed while MRE was low can use the data lines. All others remain inactive.

Responding to the Call

The MRE line, then, is an address mode/data mode selector for the data lines. When MRE is low, the codes on the data lines have the same meanings for all devices connected to the bus. When it is high, there are no restrictions on the codes used.

The codes used when MRE is low, which include certain commands as well as addresses, fall into four classes:

Code form	Meaning
X 0 0 A_5 A_4 A_3 A_2 A_1	Universal commands
X 0 1 A_5 A_4 A_3 A_2 A_1 (except X01 11111)	Listen addresses
X 1 0 A_5 A_4 A_3 A_2 A_1 (except X10 11111)	Talk addresses
X 0 1 1 1 1 1 1 and	"Unlisten" command
X 1 0 1 1 1 1 1	"Untalk" command

Note that the 6th and 7th digits indicate the meaning to be assigned to the information in the first 5 digits. The 8th digits (X) is not used when MRE is low so devices that use the ASCII 7-bit code are able to control the bus.

To allow more than one device at a time to function as a listener, any device becomes a listener when its listen address is placed on the bus (while MRE is low) and it remains a listener until the "unlisten" command is transmitted. Talkers, on the other hand, stop functioning as talkers whenever another talk address is put on the data lines (with MRE low). This prevents more than one device from talking at a time. (Talking is also terminated by the "untalk" command.)

Data Transfer

Information—whether addresses, commands, measurement results, or other data—is transferred on the data lines under control of a technique

© Copr. 1949-1998 Hewlett-Packard Co.

Table I. Interface line assignments.

Line	Name	Abbreviation	Assertive State
1	Data Input/Output 1	DIO1	Low
↓	↓	↓	↓
8	Data Input/Output 8	DIO8	Low
9	Multiple Response Enable	MRE	Low
10	Data Valid	DAV	Low
11	Ready for Data	RFD	High
12	Data Accepted	DAC	High
13	End Output	EOP	Low
14	Service Request	SRQ	Low
15	Remote Enable	REN	Low

known as the three-wire handshake.

This technique gives a flexibility to the interface system that significantly simplifies the interconnection of a variety of instruments and digital devices. This flexibility results from three important characteristics. First of all, data transfer is asynchronous, that is to say there are no inherent timing restrictions. Data can be transferred at any rate suitable to the devices operating on the bus (up to a practical limit of 1 megabyte per second imposed by circuit response, cable delays, etc.).

Second, the handshake enables the bus to interconnect devices of different input/output speeds. Data transfer automatically adjusts to the slowest active device without interfering with the operation of faster devices.

Third, more than one device can accept data at the same time. Thus several devices can listen simultaneously. This capability enables all devices to accept addresses and commands when MRE is low.

The handshake sequence is diagrammed in Fig. 3. A listener indicates that it is ready to accept data by letting the Ready for Data (RFD) line go high. Listeners are connected to the RFD line in a logical-AND configuration, however, so that the line does not go high until all active listeners are ready for data (Fig. 4).

RFD going high signals the talker that all listeners are ready for data. The talker indicates that it has placed a data byte on the data lines by setting the Data Valid (DAV) line low, but it cannot do so unless RFD is high. When DAV does go low, listeners are enabled to accept the information on the data lines.

When a listener has accepted the data (stored it in a register), it lets the Data Accepted (DAC) line go high. Here again, all active listeners are connected in a logical-AND configuration so DAC does not go high until all listeners have accepted the information. DAC remains high until the talker sets DAV high again. The sequence may then repeat

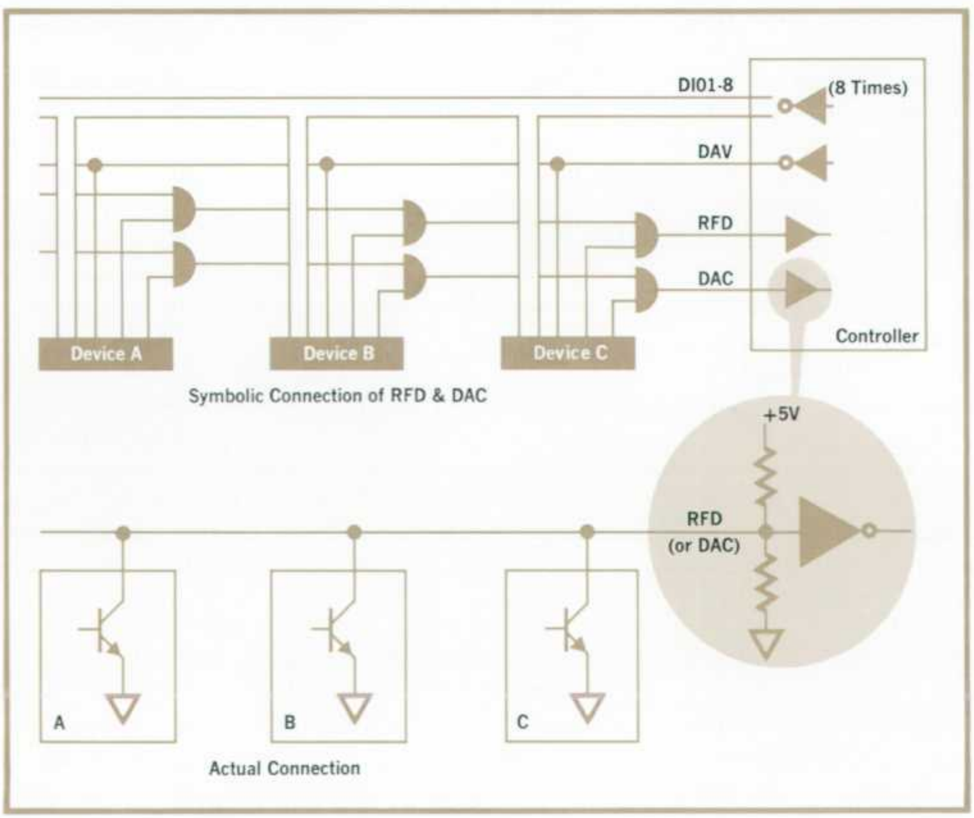


Fig. 4. *Logical-AND connection of RFD and DAC lines prevents these lines from going high until all active devices have set their outputs high (because of AND-gate configuration, assertive state for these two lines only is high).*

© Copr. 1949-1998 Hewlett-Packard Co.

after DAC goes low.

No step in this sequence can be initiated until the previous step is completed. Thus, information can proceed as fast as devices can respond, but no faster than the slowest device that is presently addressed as active.

Assuming Command

The interface allows a system to have more than one controller but only one controller at a time can manage the system. For example, two instruments may work together to perform certain measurements but they might be connected to a calculator to give more sophisticated results. The calculator could instruct one of the instruments, addressed as a listener, to execute a certain subroutine. The calculator would then pass control to that instrument by using an appropriate command word. When the subroutine was completed, the controlling instrument would then pass control back to the calculator with another command word. This arrangement reduces the number of programming steps needed for the calculator.

One of the controllers must be able to gain absolute control of the bus without being addressed itself, a capability that is needed to allow an operator to initiate action in the system. This controller, designated as System Controller at the time of system configuration, has control of another line called End Output (EOP), and it is the only device in the system to have control of this line. When the System Controller pulls EOP low, all bus activity stops immediately and all devices unaddress themselves. Besides being useful for system start-up, EOP can be used if the operator wishes to abort an automatic sequence.

Asking for Attention

Suppose a listener or a non-participant wants to become a talker or otherwise gain access to the bus —a voltmeter, for example, that wants to indicate an overrange, alarm, or error condition. A status line, Service Request (SRQ), gives this capability. Any device can indicate to the Controller that it wants attention by pulling the SRQ line low. The Controller then responds at its discretion, most likely permitting the current cycle of events to run to completion before acting on the service request. Devices are connected to the SRQ line in a logical OR configuration so that any number of devices can pull down on the line at the same time.

When the Controller responds to SRQ, it enters a service-request identification cycle. It begins by setting MRE low and then transmitting an identify-service-request command. All devices accept this command and when MRE goes high again (following the handshake cycle), those that have service requests pull down on one of the data lines, each device being assigned a particular line during system configuration. At the same time, the devices let the RFD line go high. When RFD goes high, the Controller reads the data lines to determine which devices asked for service and it takes whatever action it has been programmed to take.

Remote/local

The bus has one more control line: Remote Enable (REN). When this line goes low, control of each instrument's functions is transferred from the front-panel controls to the interface bus. This line can be pulled low only by the System Controller and gives the operator the ability to switch all instruments to remote control from a central location. Setting REN high restores local control.

The System Controller can also switch instruments to bus control by transmitting a switch-to-remote code, one of the universal commands. With this arrangement, a front-panel pushbutton, if the instrument is so equipped, can restore local control to that instrument, allowing the operator to take over control at any time. The Controller can return all devices to local control with a switch-to-local code.

Preferred Codes and Formats

Although any code of 8 bits or less can be used on the data lines when MRE is high, it is highly desirable that devices which input or output decimal numbers do so in a manner compatible with calculators or computers. The use of a common code also minimizes the need for code translation.

An example of the problems that arise with non-standard codes or formats concerns one of the early instruments designed for use with the new interface system. To simplify internal logic, this instrument used the ASCII colon (:) for a radix point instead of the usual period (.). Then it was found that a calculator would have to output a string such as <12:62> by breaking the number into an integer part and two separate digits, increasing the programming time. The instrument has now been modified to accept a period as the radix point.

A code set easily generated and understood by standard software in computers and calculators is the "printing ASCII," so-called because the 64 characters used, out of the possible 128 words available with 7 bits, are those used by teletypes. Hewlett-Packard instruments designed for use with the new

© Copr. 1949-1998 Hewlett-Packard Co.

interface system use this code, hence the reference to the bus as ASCII-compatible.

Hardware

Driver and receiver circuits connected to the bus operate at typical TTL voltage levels (high state greater than 2.4 V and low state less than 0.4 V). Driver circuits are typically open-collector NPN transistors capable of sinking more than 48 mA. Tri-state drivers may be used on the data input/output lines where high data rates are desired. Receiver circuits are standard TTL gates.

A 24-pin connector accommodates all lines, using single conductors for data input/output lines, twisted pairs for control lines, and a single shield surrounding all lines.

Interface System Summary

Number of Devices:
15 per system maximum

Signal Lines:
15 assigned (8 data, 7 control)

Data Rate:
1 Megabyte per second maximum

Data Transfer:
Byte serial, bidirectional using interlocked handshake technique

Transmission Path:
50 feet total accumulated cable length maximum

Address Codes:
31 Listen addresses
31 Talk addresses

Control:
May be delegated, never assumed; maximum of 1 talker and 14 listeners.

Hewlett-Packard instruments that can be used with the new interface system include (although these instruments are available now, in some cases interfaces will not be available until dates shown):

Signal Sources:	3320A/B Frequency Synthesizers 3330A/B Automatic Synthesizers 8660A Synthesized Signal Generators (Spring '73)
Measuring Instruments:	3490A Digital Voltmeter (Winter '72) 3570A Network Analyzer 5340A Automatic Counter
Controllers:	3260A Marked Card Reader 9820A Calculator (with 11144A Interface Kit) 2100A Computer (with 59310A Interface Kit—Summer '73)

Gerald E. Nelson

Jerry Nelson, fresh out of the University of Denver with a BSEE, joined the HP Loveland Division in 1964. Initially he worked on the 675A Sweeping Signal Generator. He then became project leader for the 3330A/B Automatic Synthesizer for a time before moving on as group leader for the 3570A Network Analyzer and related systems. These activities involved him in the definition of the new interface system.

Along the way, Jerry earned his MSEE degree at Colorado State University in the HP Honors Co-op Program. During leisure hours, he likes to sail but he also cares for a 2½-acre cherry orchard. What to do with all those cherries? He makes wine.

David W. Ricci

Dave Ricci has been involved in digitally interfacing instruments almost from the moment he joined HP's Santa Clara Division in early 1965. Once that work on his first project, the 5431A Display Plug-in for HP's Multichannel Analyzer, was completed Dave started on digital peripheral interfaces for the Multichannel Analyzer. Of late he has been coordinating interface efforts for all of HP's Counters.

A third-generation Californian, Dave obtained his BSEE at California State Polytechnical College and, following a stint with another company working in communications, earned an MSEE degree at the University of California at Berkeley. In his spare time, Dave and his wife like to ski and sail. They also like to backpack, taking along their dog, a friendly Samoyed, who carries her own pack.

© Copr. 1949-1998 Hewlett-Packard Co.

A Common Digital Interface for Programmable Instruments: The Evolution of a System

HP's corporate interface engineer describes the trends, philosophy, and ancestors that have helped define the new HP instrument interface system.

By Donald C. Loughry

INSTRUMENTATION SYSTEMS CAN'T WORK WELL UNLESS their separate elements—instruments, computers, peripherals—can communicate effectively with one another. There must be a way to tell an instrument (perhaps a digital voltmeter) what to do (go to the 10 Vdc range) and when to do it. There must also be a way for the instrument to tell what it has accomplished (DVM reading is +9.765 Vdc). That these mainly digital messages be communicated clearly and conveniently serves the common interests of design engineers, instrument manufacturers, and system users alike.

Recognizing this, representatives of various Hewlett-Packard divisions began about eight years ago to discuss what techniques might be appropriate for a digital interface system that would be applicable to most HP programmable instruments. One such general-purpose system has now been developed. It is described in detail in the article beginning on page 2. In this article I will summarize what has happened in the last few years to influence the evolution of this system.

Where We've Been

The earliest programmable instruments were programmed by contact closures. Frequently this was accomplished on a line-per-function basis where each program input line was dedicated to a specific task (i.e., one program line for each range or function). Program input lines typically did not have storage. Inputs were usually generated by means of simple, economical, mechanical switches, or later, solid-state switches. In either case, the logic convention or assertion state was "ground true."

Unlike program inputs, measurements results were usually output in coded form (e.g., two's complement binary or binary-coded decimal). The mathematical perspective of engineers, along with ease of interpretation, usually led to the "positive true" assertion state for measurement data. Thus, input and output logic conventions were frequently dissimilar. This presents no particular problem when separate input and output lines are used, but the interface circuitry becomes costly when there is a direct connection between inputs and outputs and each uses a different logic convention.

Separate unidirectional input and output lines for each instrument, the absence of program storage capability, inconsistent codes and formats, unique control techniques for each instrument, and liberal use of uncoded line-per-function program control signals are a few of the parameters that characterize the instrument interface environment from which we are now emerging.

Tracing the Evolution

Looking back at specific hardware, one can see the changes in interface techniques that were implemented as problems were recognized and experience was gained.

The history of today's digital interface methods at Hewlett-Packard begins with the "Adam and Eve" of instrumentation systems, a frequency counter output coupled to a digital recorder input. Digital information was presented in parallel form, that is, all data bits for a complete frequency measurement were presented simultaneously. Information flow was unidirectional: the first frequency counters did not have programmable inputs, and it wasn't necessary to output data from the digital recorder. The marriage of these two products back in the mid-1950's proved highly successful. The

© Copr. 1949-1998 Hewlett-Packard Co.

event sparked much activity, and digitally coupled instrumentation systems were born at HP.

Within five years digital interfaces were commonplace. Digital voltmeters, scanners, power supplies, remote displays, and other products were added to the growing family. By this time some products, digital voltmeters in particular, required both digital inputs and outputs. The voltmeter not only generated measurement data but was programmable as well. This was another significant event in the interface world.

Along with this new capability came the need to program many products within one system. The solution was to create a special purpose controller, backed up by a paper tape or punched card memory. The interface network formed a radial or star structure as shown in Fig. 1. Most of these systems were custom-designed mixtures of standard instruments and special products. While the number of system elements expanded rapidly, each digital interface remained unique with little similarity to others.

The next interface milestone saw the introduction of small, stand-alone, relatively low-priced, standard instrument systems with program control capability internal to one of the products. These systems forced the start of more commonality among the interface characteristics. Programmable inputs needed to have common signal levels and logic conventions. Attention was focused on common code structure and data format.

Fig. 2 illustrates the typical standard data acquisition system in vogue at this time. Information flow was still essentially unidirectional and each signal line was dedicated to carry only one type of information.

To correct some of the format and code translation problems, additional coupler units were developed to interact with various recording media. A different coupler was required for each output recording medium (e.g., magnetic tape, punched cards, punched paper tape, hard copy, displays, etc.).

A breakthrough occurred in the late 1960's as the marriage of computers and instrumentation systems became a reality. Overall system capability soared, data rates climbed significantly, and message traffic volume increased dramatically over the system interfaces.

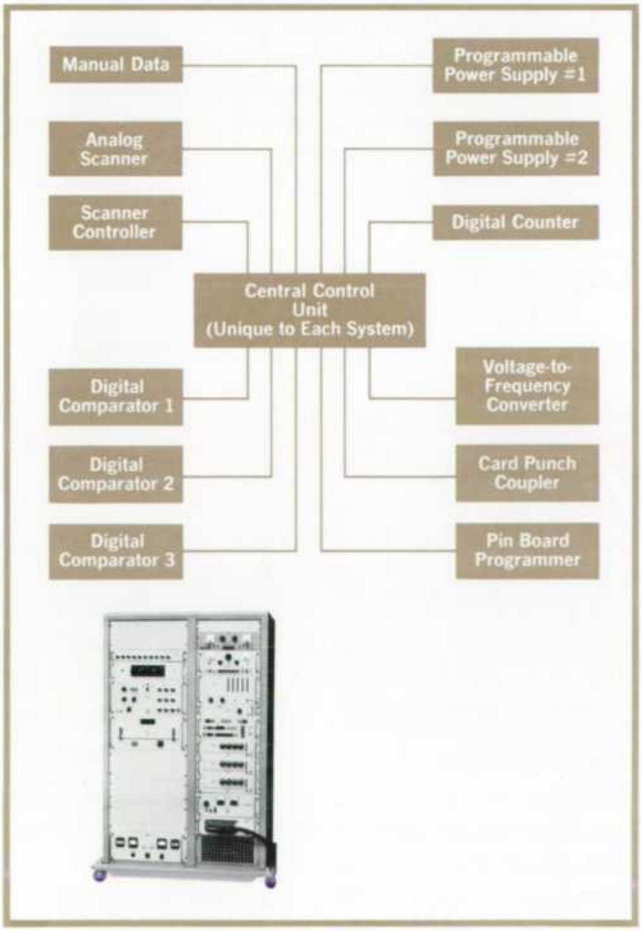


Fig. 1. *Early instrumentation systems had a special control unit and a "star" interface pattern.*

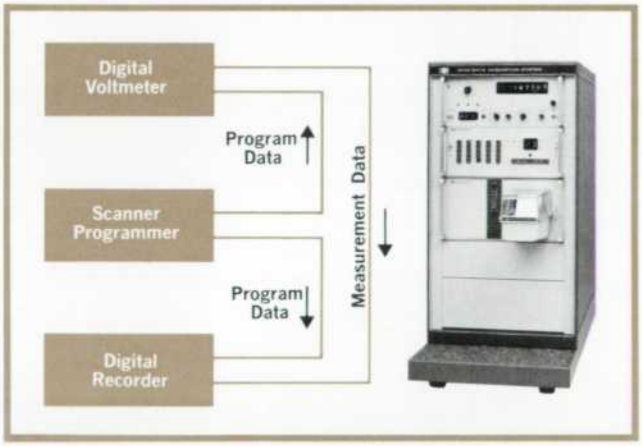


Fig. 2. *Later, small stand-alone systems had the control function within one of the instruments.*

In these systems, control was vested in one location, the central processor mainframe. The interface hardware, particularly the input/output circuits, approached uniformity as a result of the economic demands of the CPU design and the wide availability of solid-state microcircuits. The demands of system software led to a reduction in the variety of instrumentation program codes and data formats.

Bus techniques were frequently used to interconnect internal sections of the CPU mainframe, and

© Copr. 1949-1998 Hewlett-Packard Co.

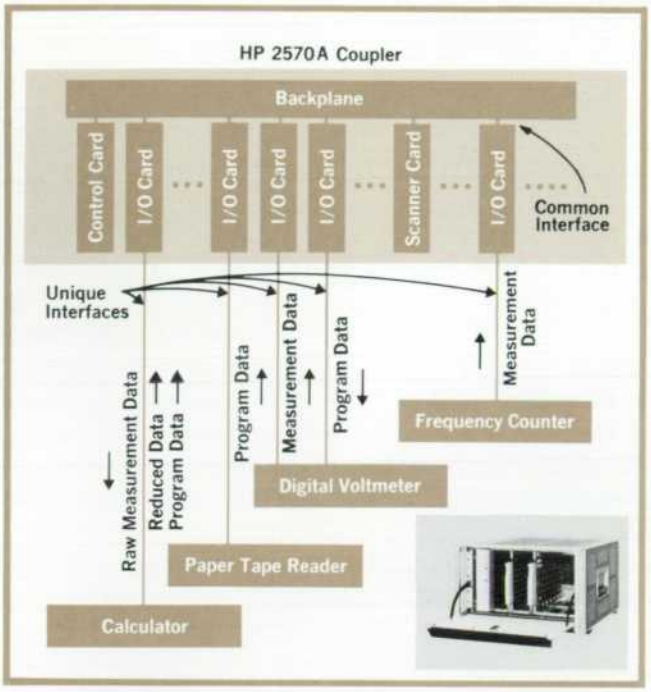


Fig. 3. *A recent instrumentation coupler is the forerunner of the new general-purpose interface system now being introduced into HP instruments. Individual instruments communicate in character-serial format via a bidirectional backplane bus. Control can be delegated to different devices at different times.*

some of these same capabilities were extended to the instrument interface in the form of word serial (16-bit parallel) data transfers and more bidirectional interfaces. The overall interface network, however, remained a star pattern with each instrument connected to the mainframe processor by its own cable and plug-in I/O card.

The plug-in I/O card and backplane structure used in processor mainframes gave rise to a new coupler philosophy that came into existence around 1970. Fig. 3 shows the coupler as the data interchange hub for a data acquisition system. The backplane of the coupler is a bus structure for a closely interrelated collection of I/O cards, one to each instrument or device.

This system concept represented the beginning of an instrument bus structure. Looking at the interface backplane, data rates were high, information flow was bidirectional, control could be delegated to different elements of the system, codes and formats were standardized. At the backplane side of the I/O cards the interface circuits were identical. However, the individual instruments still retained their unique interface characteristics.

In many respects this collected bus system was the immediate predecessor to what has been described in the preceding article: a distributed bus interface system where similar functions and capability exist without being concentrated on one backplane.

Where We're Headed

Turning from the past to the present and future, we can identify several trends. For one, serial interface techniques are in common use throughout the data processing field in devices such as teleprinters, line printers, and magnetic tape units. These products, as well as calculators and computers, are now frequently used in instrumentation systems. Programmable instruments are therefore required to interface with many devices which use some form of byte-, character-, or word-serial interface. Even simple program input devices like paper tape and mark-sense card readers are character-serial by nature.

Remote data collection stations, interaction with time-shared systems, and the frequent use of common-carrier communication facilities are also on the increase. All of these trends seem to point to a serial interface.

Another trend is toward intelligent instruments, for which a few bytes of input data define the limits for an entire series of events. No longer is it necessary to program each discrete action.

These trends help create an environment conducive to a common interface system. There are other trends that will influence the character of that system.

One of these is towards decreasing instrument response times as instrument capability responds to the need for greater data rates in automatic measurement systems. Greater instrument data rates tend to increase the amount of message traffic that must be carried by the interface.

While the trend toward smart instruments is decreasing the need for programming data, the number of programmable parameters is increasing. It is not unusual to have all front-panel controls available for digital remote programming. This means that any interface system must provide for an increasing repertoire of messages.

Now Is the Time

With these trends both demanding and helping to create more effective interproduct communications, is there reason to believe that a common interface system is feasible at this time? The answer is yes, there are many reasons.

One key fact is the availability, capability, and variety of inexpensive digital integrated circuits.

© Copr. 1949-1998 Hewlett-Packard Co.

Interface circuit costs no longer inhibit the addition of storage (20¢ per bit), conversion between serial and parallel data (10¢ per bit), interface control logic, and the driver/receiver circuits necessary in any interface system.

The advent of intelligent instruments has decreased interface message requirements, thereby increasing the feasibility of a serial interface system. General acceptance of a common message convention (polarity or logic assignment of binary data), now a virtual reality within HP, has appreciably aided interface feasibility.

Not so obvious is the experience gained over the last few years. The last five years in particular have taught us many hard lessons and made us better able to develop an effective interface system.

Perhaps the most significant factor that makes an effective interface system feasible today is acceptance of the fact that no single interface system either can or needs to solve all problems! Instrumentation systems tend to be in the middle of the spectrum in terms of such interface parameters as data rates and path widths, data transmission lengths, number of interconnected devices, and message traffic volume. Defining an interface system for this middle range now appears possible. Interfaces for special environments such as extremely long transmission paths and electrically noisy environments are indeed needed, but less frequently. Overall costs are reduced and system flexibility is increased if individual instruments are not required to meet all of these special needs.

Interface System Objectives

As a first step toward developing an effective interface system it is helpful to identify the most important objectives and design criteria. Of many objectives, four stand out. The interface system must be:

1) capable of interconnecting small, low-cost instrument systems by means of simple passive cable assemblies without restricting individual instrument performance and cost.
2) capable of interconnecting a wide range of products (measurement, stimulus, display, processor, storage) needed to solve real problems.
3) capable of operating where control and management of the message flow over the interface is not limited to one device but can be delegated in an orderly manner among several.
4) capable of interfacing easily with other more specialized interface environments.

Additional criteria to consider would include at least these: operation under asynchronous conditions; communication with two or more devices simultaneously; limitation of the total number of signal lines to no more than can be accommodated in one computer word; compatibility with the most widely used codes for information interchange; ability to alter codes and data rates to achieve system flexibility; ability to alter the communication network to optimize system performance.

The byte-serial, bus-oriented interface system described in the article on page 2 is one that meets these criteria. This interface system is now being used in many new HP products.

Acknowledgments

The present interface system developments are the result of corporate-wide vision, commitment, diligence, and enthusiasm. Though the list of engineers is long, Gerald E. Nelson and David W. Ricci made major contributions. The patience, understanding, and constructive commentary from customers has had its positive influence as well. The efforts of all are hereby acknowledged.

Donald C. Loughry

Don Loughry started out at HP in 1956 as a production test department manager. In 1958 he moved to the development laboratory to work on custom and standard systems, and in 1963 became engineering manager of the Dymec Division. He's held his present position of corporate interface engineer since 1968. In that capacity, he's responsible for companywide interface guidelines and services. He's a member of IEEE and is active in various groups working on interface standards, including IEC, ISO, and ANSI.

Don is a 1952 B.S.E.E. graduate of Union College. When he's not busy interfacing, his interests vary widely, they include gardening, photography, classical music, skiing, camping, and star gazing (he grinds his own telescope lenses). He's also active in his church and in interfaith groups.

© Copr. 1949-1998 Hewlett-Packard Co.

Faster Gain-Phase Measurements with New Automatic 50Hz-to-13MHz Network Analyzers

Complete characterization of networks in the frequency domain now becomes faster and more convenient than ever.

By Gerald E. Nelson, Paul L. Thomas, and Robert L. Atchley

ALTHOUGH THE BENEFITS GAINED FROM AUTOMATING GAIN-PHASE MEASUREMENTS have led to near universal use of this tool at microwave frequencies, widespread automation at lower frequencies has been inhibited by cost. Available manual techniques already provide good information at reasonable cost.

This situation is now changing, a consequence of the fast-developing semiconductor technology that gives us digital processors at piggy-bank prices. Two developments are bringing about this changed situation: (1) instruments with built-in "intelligence," and (2) a new interface system that makes it possible to assemble a variety of systems largely by plugging in cables. The interface system is described in the article beginning on page 2 of this issue, and some of the new instruments have been described in recent issues.[1,2,3]

A new Network Analyzer for measuring gain and phase over a 50 Hz-to-13 MHz frequency range now joins the growing family of "intelligent" instruments that operate with the interface system, eco-

Fig. 1. *Model 3570A Network Analyzer (upper unit) functions as two-channel tracking detector with one of HP's 0–13 MHz Frequency Synthesizers (lower unit) to form complete network analysis system. Network Analyzer shown here has Delay / Limit Test / Offset option.*

© Copr. 1949-1998 Hewlett-Packard Co.

nomically giving the design engineer working at audio, video, and RF frequencies the speed, accuracy, and high information content that automatic gain-phase measurements can give.

The Network Analyzer, Hewlett-Packard Model 3570A, works with one of HP's new Frequency Synthesizers (Models 3320A/B or 3330A/B) to measure the gain or insertion loss of two-port devices, and to measure the phase shift between input and output ports. Such measurements, made over a range of frequencies, give the design engineer all the information he needs to predict the response of a device to any arbitrary signal. With the new Network Analyzer (Fig. 1), the engineer can analyze the performance of amplifiers, filters, and other analog devices quickly and in great detail. The Network Analyzer can also function as a comparator to determine how closely the gain and phase response of a device matches that of a reference.

Both the passband and the stopband of a device can be examined in detail because the Network Analyzer has both a wide amplitude range of 120 dB (1 μV to 1 V) and a high resolution display (0.01 dB increments). Display range on any one amplitude range is 100 dB. Accuracy is 0.2 to 1 dB depending on frequency and signal level (see page 20 for details).

The digital readout also displays phase readings with 0.01° resolution. Nominal phase accuracy is 1°

Beyond basic amplitude and phase measurements, the instrument has broader capabilities when equipped with an optional addition to its internal data processor. One is digital offset. Values of amplitude and/or phase measured at some reference frequency or on some reference device are stored in the instrument's memory at the push of a button. Future measurements can then be displayed with the stored values subtracted from the reading. This could be used, for example, to quickly find the −3 dB passband limits of a filter or amplifier by storing the amplitude reading obtained at the middle of the passband, and then finding the frequencies where the display reading drops to −3.00 dB.

Another capability of the option, for use in conjunction with the Model 3330A/B Automatic Synthesizer, is measurement of group delay (rate of change of phase shift with frequency). As the Synthesizer is stepped in frequency, the Network Analyzer compares the phase reading at each discrete frequency to the previous frequency's phase reading. The internal digital processor then calculates and displays group delay and it outputs a proportional analog voltage for presentation on an oscilloscope or X-Y recorder (Fig. 2).

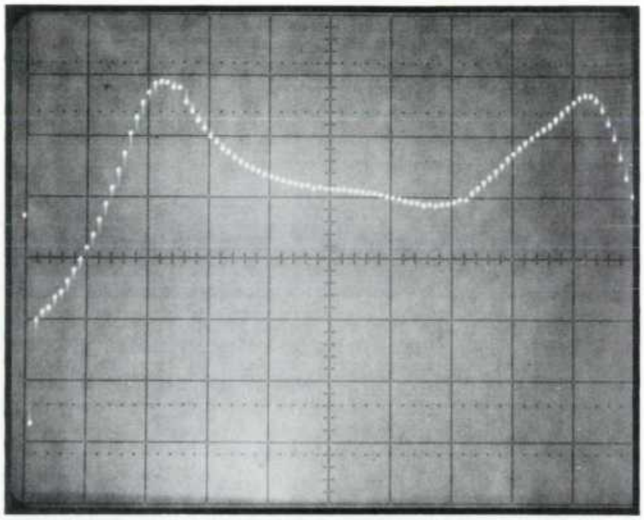


Fig. 2. *Group delay vs frequency of FM IF filter as measured by Network Analyzer equipped with Delay/Limit Test/Offset option. Center frequency is 10.7 MHz, horizontal scale is 50 kHz/div, vertical scale is 1 μs/div.*

A third capability is limit test. High and low limits can be entered as digital words from an external controller. The Network Analyzer, again working with the 3330A/B Automatic Synthesizer, can then be set to stop the frequency sweep when a limit is reached, or it can output a marker to the display device and continue the sweep. Limit test could be used, for example, to find the center frequency of a resonant circuit by stopping the frequency sweep when 0° phase is read, or it could find the gain crossover point of an amplifier by stopping the sweep at the 0-dB point. It would then indicate gain crossover on the frequency display and phase margin on the phase display.

Signal Paths

The new Network Analyzer is a synchronously tuned, two-channel receiver that functions much the same as earlier HP network analyzers.[4] It's the first to contain a digital processor, however, and the first to work with the new interface system.

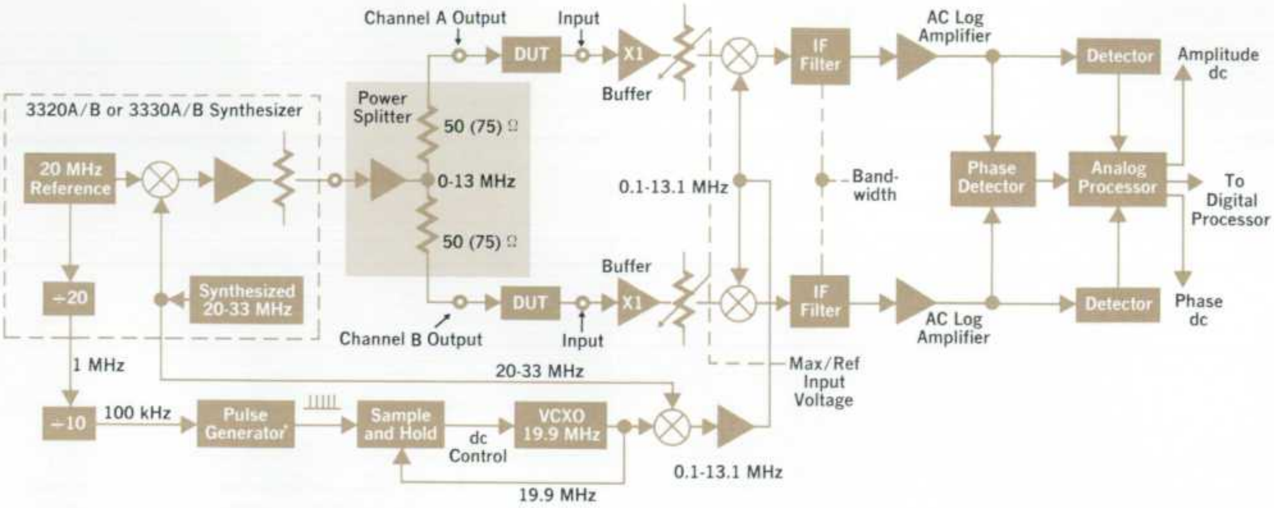
A block diagram is shown on page 14. The signal source input is split into two channels by an active power divider and supplied to two output ports. One port feeds the device under test and the other is used for the reference.

The output of the device under test is fed into one receiver channel. The reference signal goes to the other channel either directly or through a reference device. The signal in each channel, suitably buffered and attenuated, is heterodyned to 100 kHz in a double-balanced mixer.

Each IF signal passes through a narrowband filter

© Copr. 1949-1998 Hewlett-Packard Co.

The Analog Story



A. *Simplified block diagram of Network Analyzer includes pertinent sections of Frequency Synthesizer. Device under test (DUT) connects between channel output and input.*

Performance of both signal channels was made identical so the new Network Analyzer can be used over its full dynamic range as a comparator.

The power splitter has a driver amplifier with a low-impedance output that attenuates any signal reflections. The passive wye or delta network usually used for splitting the source signal equally into two channels has only 6-dB isolation between channels so reflections from a device in one channel could affect the amplitude and phase of the signal in the other channel. With the new arrangement, channel-to-channel isolation is greater than 50 dB up to 2 MHz, rising about 6 dB per octave above 2 MHz.

The input to each measuring channel goes to a high-input-impedance (1 MΩ/<30 pF) FET buffer amplifier. This allows a choice of terminating resistance (e.g., 50, 75, 600 ohms) and permits the use of standard scope probes for high-impedance probing or for measuring voltages up to 10 volts (normal maximum input is 1 V). To avoid phase-shift problems at the input, the buffer precedes the input attenuator.

IF Passband

Because image frequencies are not a problem with the Network Analyzer (there is only one frequency at its input at any one time) it can use a relatively low IF (100 kHz) to improve the resolution of phase measurements. For example, to get 1° phase resolution at 10 MHz requires time resolution of 278 picoseconds, but at 100 kHz, 1° is equivalent to 27.8 nanoseconds. Should the input signal itself be 100 kHz, good mixer balance suppresses the amount of signal that leaks through to the IF to more than 50 dB below the desired sideband.

Since filter response time is inversely proportional to bandwidth, the instrument has a 3-kHz IF bandwidth compatible with the maximum sweep speed (1 ms per step) of the Automatic Synthesizers. There could be some LO feedthrough at input frequencies below 2 kHz, however, and below 1.5 kHz, the upper sideband of the mixer output would pass through the IF channel so two more switch-selected bandwidths are provided: 10 Hz and 100 Hz.

All of the IF filters are 3-pole, synchronously-tuned filters (approximately Gaussian) that provide sufficient attenuation of the upper sideband to permit 80-dB dynamic range with signal frequencies 5 times the bandwidth and 100-dB range with signals higher than 30 times the bandwidth. The low end of the instrument's frequency range is thus 50 Hz with the 10-Hz filter switched in. The stability of the Synthesizers plus the stability of the crystal filters permits the 10-Hz bandwidth to be used over the entire 50-Hz-to-13-MHz range of the instrument when maximum accuracy is desired.

Stable Local Oscillator

A highly stable local oscillator frequency that is always 100 kHz higher than the Synthesizer output is obtained by subtracting a fixed 19.9 MHz signal from the Synthesizer's offset signal (20 to 33 MHz). The 19.9 MHz is generated in a crystal-controlled oscillator (VCXO) that is phase-locked to the Synthesizer's fixed 1-MHz output.

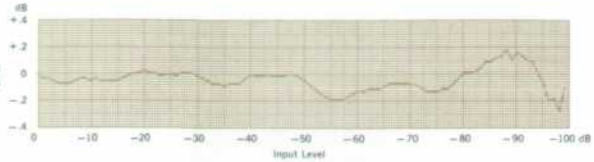
The VCXO thus has the same long-term stability as the Synthesizer. A source frequency change of 10 ppm, the worst that could be expected over a long period of time, would offset the intermediate frequency only 1 Hz. The small amplitude and phase errors that would result from a 1-Hz offset are easily calibrated out with the front-panel screwdriver controls that were provided to compensate for the effects of temperature on the zero levels of the analog processor.

Log Conversion

Wide dynamic range was obtained without serious degradation of resolution and accuracy by use of a thin-film, hybrid, log amplifier. This uses several differential transistor stages with inputs connected to separate taps on an attenuator and outputs driving a common load.

(continued)

© Copr. 1949-1998 Hewlett-Packard Co.



B. *Relative response of hybrid thin-film log amplifier shows exceptional linearity over 100-dB amplitude range.*

The relationship between the output current of a common-emitter stage and the logarithm of the input is linear over a range of about 10 dB. Taps on the attenuator are selected so that as the signal amplitude increases, a differential stage moves into its linear range as the stage on the next higher tap starts limiting. The amplifier has a total operating range of 100 dB using 12 stages separated by 10-dB attenuator taps (two extra stages are needed to take care of the range lost near the end points). All 12 transistor stages are fabricated on one monolithic chip, to assure stage matching, and the thin-film attenuator resistors are trimmed by a computer-controlled laser for accurate 10-dB spacing of the transistor inputs. Typical linearity is shown in the figure at left.

Phase Detection

The use of differential pairs in the log amplifier preserves zero crossings for accurate phase detection. Typically, less than 1° of phase change results from a 60-dB change in signal level and when there are equal changes in both channels, common-mode cancellation reduces this error even further.

The phase detector is the type that triggers a flip-flop "on" when the signal in one channel crosses zero and "off" when the other signal crosses. Averaging the resulting pulse train gives a dc that is accurately related to phase difference.

The phase detector works over a 1-359° range but by manipulation of the output, the 0° point is placed at the center of the range. Thus the instrument reads over a ±179.5° range. To read phase shifts around ±180°, the reference signal can be inverted to place the 180° point in the center of the range.

and is then amplified in a logarithmic amplifier before being applied to a detector. Each detector's output is thus a dc voltage proportional to the logarithm of the ac level at that channel's input.

The outputs of both logarithmic amplifiers are also fed to a phase detector that derives a dc voltage proportional to the phase difference between the two signals.

The dc voltages are fed to an analog processor that buffers them to output ports for use by external display equipment. The analog processor also subtracts the dc output of one channel from the dc output of the other to derive a voltage proportional to the difference, which can be used as a measure of gain.

Certain portions of a CRT display may be intensified for the user's convenience. He can choose to have either the center point or the major graticule marks intensified. If a limit test is being made, the transition points (i.e., going from GO to HI, etc.) are intensified.

Signal Processing

DC voltages resulting from a measurement are converted to digital numbers by an analog-to-digital converter in the digital processor (page 16).

The processor performs whatever operations may be called for on the numbers (offset, limiting, group delay) and encodes the results for use by the front-panel displays and for transfer to the interface bus. It also operates the instrument controls, such as changing input range or IF bandwidth, as called for by a system controller. All front-panel controls (except the on/off switch) are operable by the processor.

Group Delay Measurements

With the optional digital processing capability installed in the processor, the instrument can make group delay measurements. Group delay T_d is defined as:

$$T_d = \frac{-d\phi}{d\omega}$$

where ϕ is in radians and ω is in radians per second. By making phase shift measurements at two closely-spaced frequencies, the Network Analyzer can approximate a determination of group delay. It measures phase in degrees, however, and the frequency step is given in units of frequency. Therefore:

$$T_d \approx \frac{-\Delta\phi \text{ (degrees)}}{2\pi\Delta f \text{ (radians/s)}} \times \frac{\pi \text{ (radians)}}{180 \text{ (degrees)}}$$

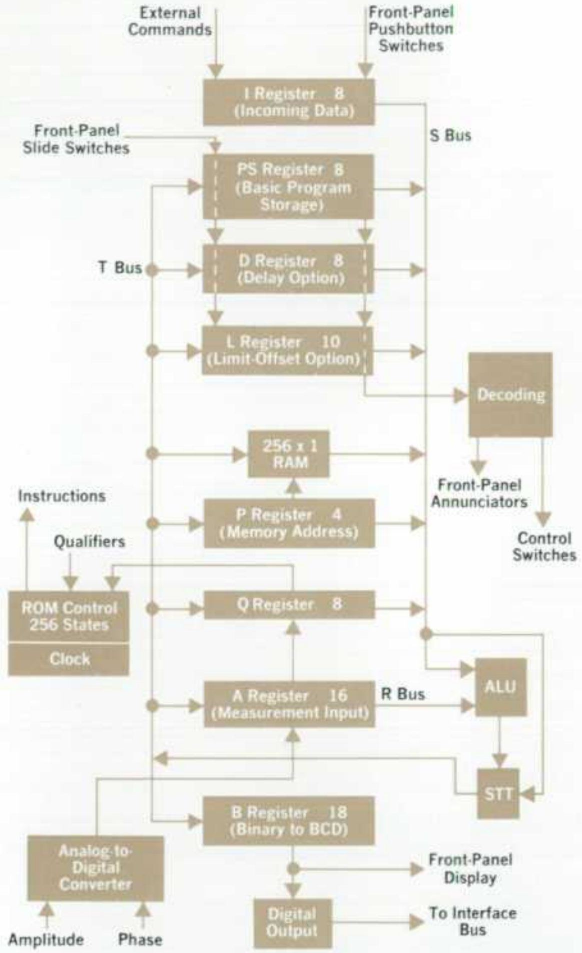
$$\approx \frac{\phi_2 - \phi_1}{360\Delta f} \text{ seconds}$$

The network analyzer stores the results of one phase measurement and then makes a second one at a different frequency to find $\phi_2 - \phi_1$. The frequency step is entered through the interface bus (or, in the absence of a system controller, entered on the front panel).

The Network Analyzer offers two modes with the delay option. One, called Swept Delay, calculates and outputs the group delay at every frequency step except the first. In the other mode, CW delay,

© Copr. 1949-1998 Hewlett-Packard Co.

The Digital Story



The digital processor in the new Network Analyzer is structured around the conventional "R-S-T" three-bus system, with the "R" and "S" busses carrying the two inputs to an arithmetical logic unit (ALU) whose output goes to the "T" bus. Because the basic instrument without its delay and limit offset options does not need the ALU, nor the read/write memory, nor the P, L, and D registers, the "S" to "T" (STT) bypass selects either the ALU output or the S bus for connection to the T bus. To make the systems operable without the ALU, the "Q" register was made parallel-loadable so the "A" register can be emptied without the use of the ALU.

Operation of the processor is completely serial with all computations done in binary arithmetic. The decision to use a binary machine was based primarily on the bit savings of binary as compared to BCD. A BCD machine needs 17 bits to enable the analog-to-digital converter (ADC) to accommodate numbers up to 100.00. A binary machine needs only 15 bits. Another advantage is the simplicity of the arithmetic software. The disadvantage is the necessity of converting from binary to BCD for the display and outgoing data.

The built-in programs have been partitioned into data routines and measurement routines. A data routine is entered if the data flag is set, which it is whenever the "I" (input) register is loaded (the numbers in the boxes indicate the numbers of bits in each register). The eight bits of information in the "I" register are transferred to the "Q" register for decoding and placement into the program storage registers. The "I" register also takes in the numerical data for offsets and limits. These numbers are prefaced with a code that tells the processor where to store them.

The processor enters a measurement routine only if the data flag is not set and if the measurement flag is set. The measurement flag may be set by a command from the external system controller, or by the Synthesizer changing either its frequency or amplitude, or by an internal multivibrator. In the absence of any external commands, the multivibrator repeatedly sets the flag so the instrument will make measurements at a rate of four per second.

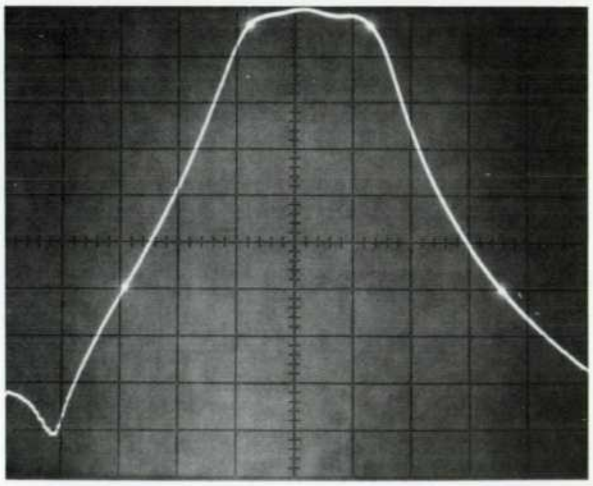
On entering a measurement routine, the processor commands the ADC to take an amplitude measurement. The ADC is a successive-approximation type that parallel-loads the "A" register, which is the entry point for measurement information.

When the "A" register is loaded, the processor interrogates the "L" register to see if there is to be any offset or limit testing. If these operations are programmed, the processor performs them. When all operations have been completed, information is shifted to the "B" register where it is converted to BCD, latched out to the display and, if so programmed, transferred to the interface bus in bit-parallel, character-serial form.

At the conclusion of amplitude data outputting, the processor tells the ADC to take a phase reading. It then interrogates the "D" register to see if it is to calculate delay. If it is to calculate delay, it stores the present phase reading and uses the preceding phase reading along with the value of the frequency step to calculate delay. Then the "L" register is interrogated to see if any offsetting or limit testing is needed.

Once all the data manipulation has been done, the phase/delay information is shifted into the "B" register, converted to BCD, latched out to the display, and transferred to the interface bus. If the instrument is in the limit test mode, it gives the limit conditions also.

Before the measurement flag is reset, time is given for the delay analog output to stabilize. The processor then issues an unblanking command for an oscilloscope.



Intensity markers indicate where measured quantity crosses high and low limits stored in processor.

© Copr. 1949-1998 Hewlett-Packard Co.

the Analyzer calculates the delay at one specific frequency only. To do this, the Synthesizer must be operated in the swept mode so that phase readings may be taken on either side of the center frequency.

System Configurations

The new Network Analyzer is not a stand-alone instrument but requires a signal source that can supply the appropriate offset frequencies needed for automatic tuning. It is therefore offered with one of the Synthesizers in packaged systems.

The basic system is the Model 3040A. It includes the Network Analyzer and a Model 3320A or B Frequency Synthesizer.[1] This gives the development engineer a bench system that makes gain-phase response measurements with the wide dynamic range and high resolution of the basic Network Analyzer and with the high accuracy and stability of the Synthesizer.

Next in capability are the 3040A Systems that use a Model 3330A or B Automatic Synthesizer.[2] These give the speed and convenience of swept-frequency response measurements but with the point-by-point measurement accuracy that synthesized incremental frequency sweeps give.

Residual FM, often a serious limitation to the frequency resolution of swept-frequency measurements, is so low with these systems ($<<1$ Hz) that narrowband sweeps can be made with accuracy (Fig. 3). For example, the frequency can be incremented in 100 steps over a sweep width of only 10 Hz around a high center frequency such as 10 MHz, providing accurate pictures of the frequency response of networks that have Q's of several million.

Fig. 3. *Passband response of crystal filter shows detail that low residual FM of incremental frequency sweep enables in measurements on narrowband devices. Sweep had 200 frequency steps.*

Through the Interface

Automating these systems is simply a matter of plugging in a controller, since both the Network Analyzer and the Synthesizers work with the new interface system. The first level of automation is provided by the 3041A Systems (Fig. 4), which include the Model 3260A Marked Card Programmer, the Network Analyzer with the group delay/offset/limit test option, a 3330B Automatic Synthesizer, cabling, and a 5-foot rack cabinet with filler panels (variable-persistence oscilloscope display and/or X-Y graphic recording are available as options).

Test parameters (function, frequency range, amplitude, etc.) are marked on standard size tab cards with an ordinary lead pencil and fed by hand to the Marked Card Programmer. The Programmer sets up the instruments' controls at the rate of 33 switches per second, saving considerable time when tests are repeated. The chances of human error are also greatly reduced since the Programmer locks out the front panel and sets up the instruments' controls exactly the same way each time the cards are run.

The Programmer can also call upon the routines built into the instruments to conduct a variety of tests. For example, with only two cards the system can be programmed to sweep through 1000

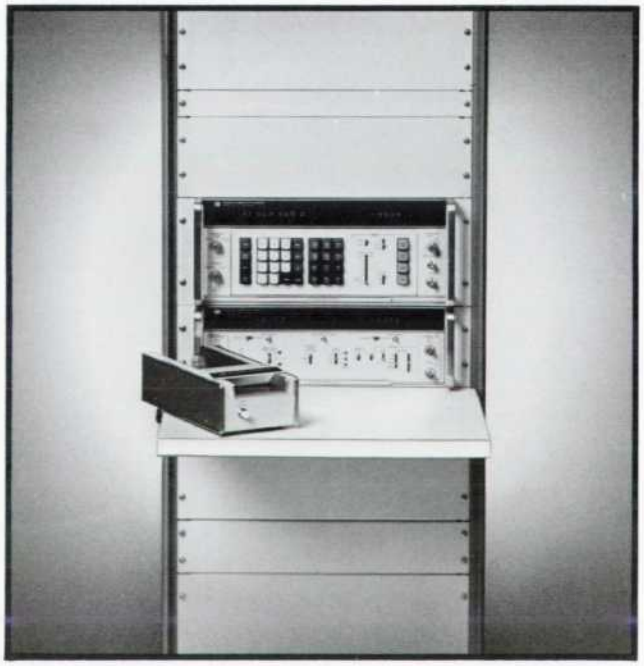


Fig. 4. *Model 3041A Network Analyzer System has Marked Card Programmer for fast set-up and execution of tests.*

© Copr. 1949-1998 Hewlett-Packard Co.

incremental frequencies, measure amplitude or phase at each frequency, and compare each measurement with programmed test limits. If a limit is exceeded, the system stops and displays the limit condition (Hi-Go-Lo), the measurement value that exceeded the limit, and the frequency at that point.

Full Automation

Using a Model 9820A Calculator as the controller gives the Network Analyzer the power of data reduction. The calculator-controlled system, Model 3042A (Fig. 5), has most of the capabilities of a computer-controlled network analyzer but at much lower cost. This is because the 9820A Calculator has problem-solving capability comparable to a 16-bit computer with 8k memory and a BASIC interpreter loaded in core, but it is much simpler to operate.

Fig. 5. *Model 3042A Network Analyzer System, shown here with optional oscilloscope display and X-Y recorder, has calculator that gives Analyzer capabilities of computer-controlled system.*

With the new interface system, the Calculator needs only one I/O card to drive both the Synthesizer and the Network Analyzer (equipped with the two-way interface option; the standard instrument accepts digital inputs but does not output). As a matter of fact, the Calculator can drive up to 15 instruments through the bus with the one I/O card. Similarly, only one software driver (a plug-in ROM block) is needed.

Among the many advantages of calculator-controlled Network Analyzers is a considerable gain in test speed. The system can run through a long series of tests on a device, checking performance at all specified points, and deliver a simple pass/fail answer in little more time than the sum total required for the device to reach steady-state behavior at each test frequency.

Increased measurement accuracy is another advantage. The system can measure its own errors and store them in the Calculator, later adjusting test readings to account for the inherent errors. In addition, the Calculator can store test data during production-line tests for later statistical analysis, a great help in quality control.

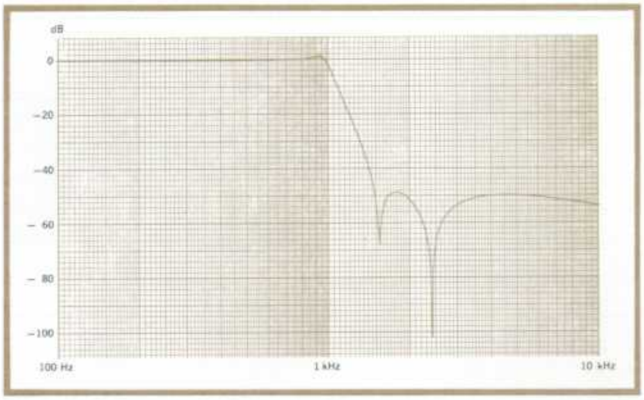


Fig. 6. *Response of audio-range elliptic low-pass filter, made with calculator-controlled system, demonstrates wide dynamic range of Network Analyzer.*

Acknowledgments

Initial product definition was greatly influenced by Bill Parzybok, who acted as the interface between marketing and engineering. Larry Whatley was the initial group leader and outlined the analog requirements. Bob Jeremiasen contributed to the log amp design and Harry Dietrich contributed the A-D converter, BiDec Register and the power supplies. Roger Cox designed the calculator-to-interface bus I/O card with valuable assistance from Frank Yockey, who designed the peripheral control ROM block. Product design was by Jay Jacobsen. Others who helped finalize the design and bring the project to a successful conclusion were Mike Aken, Virgil Leenerts, Chuck Platz, Dave Deaver, Jerry Daniels, and Marsh Faber.

References

1. Ronald K. Tuttle, "The Synthesized Test Oscillator—A New Signal Source for the 0.1 Hz-13 MHz Range," Hewlett-Packard Journal, July 1972.
2. Charles A. Kingsford-Smith, "The Incremental Sweep Generator—Point-by-Point Accuracy with Swept-Frequency Convenience," Hewlett-Packard Journal, July 1972.
3. John C. Shanahan, "Uniting Signal Generation and Signal Synthesis," Hewlett-Packard Journal, Dec. 1971.
4. Charles A. Kingsford-Smith, "Network Analysis at Low Frequencies," Hewlett-Packard Journal, July 1969.

© Copr. 1949-1998 Hewlett-Packard Co.

Programming Economy

Besides a substantial reduction in costs, the Model 9820A Calculator has another advantage over a computer as the Network Analyzer controller: it simplifies programming. All routines for controlling the instruments and all routines for mathematical manipulation of test results reside in plug-in ROM blocks supplied with the Calculator for these systems. Hence they do not occupy any of the main memory. These routines are called by special keys and fit in with the syntax of the source language as though integral parts of it. Debugging is easier too, since the Calculator's editing keys can insert or delete whole lines or individual characters from the calculator programs at will without disturbing the branching logic.

A brief routine for measuring complex impedance will illustrate the Calculator's language power. The measurement set-up is diagrammed in the figure below. The impedance (Z) to be measured is placed in shunt with the Analyzer's input. For maximum sensitivity, source and terminating impedances comparable to the unknown are inserted ($R_o \approx Z$).

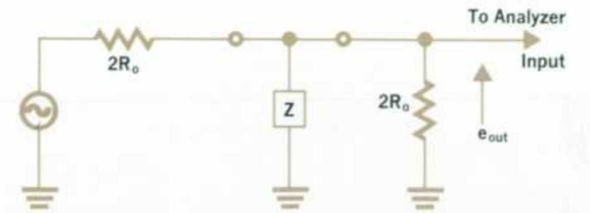
When the reference channel is equipped with the same source and terminating resistances, the "insertion gain" (G) read by the Analyzer, operating in the B-A mode, is:

$$G = \frac{Z}{R_o + Z} = A \quad \underline{/\theta}$$

The program statements that direct the Synthesizer to output the test frequency, command the Analyzer to measure and output a measure of insertion gain, and tell the Calculator to compute the real and imaginary parts of the unknown impedance Z are listed below.

	Program	
	0:	
	1E6→C;1E3→R0;	Specifies frequency = 1 MHz and R_o = 1kΩ.
	FMT "L",FXD *.0, "="⊢	Format specification for writing frequency to Synthesizer.
	1:	
Addresses Synthesizer to listen.	CMD "?U$";WRT 13 ,C⊢	Outputs frequency data to Synthesizer.
	2:	
0.2-second pause.	DSP "";CMD "?U!" ,"0,",",?5A"⊢	Commands Analyzer to take one reading and addresses it to talk.
	3:	
Specifies free-field read format.	FMT *;RED 13,A,B ⊢	Inputs amplitude and phase readings, storing them in A and B.
	4:	
	TN↑ (A/20)→Z;Z COS B→A;ZSIN B→B ⊢	Converts gain to real and imaginary components.
	5:	
	R0(A-AA-BB)/(1-2 A+AA+BB)→X⊢	Calculates real part of Z and stores it in X.
	6:	
	BR0/(1-2A+AA+BB) →Y⊢	Calculates imaginary part of Z and stores it in Y.

This program occupies only 80 words of memory, less than 5% of the total available to the user.



Robert L. Atchley

Bob Atchley, on obtaining his MSEE degree in 1967 from Colorado State University (where he had previously obtained his BSEE degree), took his family to Bangkok, Thailand, to work on a research study project for CSU. On returning to the U.S. eight months later he worked on range tracking instrumentation at a government laboratory and then joined HP Loveland in late 1968, going right to work on the 3570A Network Analyzer project where he eventually assumed responsibility for the digital processor.

Married, and with one son and one daughter, Bob dabbles in photography. He also enjoys fishing.

Paul L. Thomas

Paul Thomas earned both BSEE and MSEE degrees from Utah State University, joining the HP Loveland Division immediately thereafter (1966). Initially he worked on the 675A Sweeping Signal Generator and the 676A Tracking Detector and then moved on to the 3570A Network Analyzer project where he had final responsibility for the analog design.

Paul enjoys camping, taking the whole family along in his pickup camper. He does some hunting but looking ahead towards more vigorous activity, he has built two white-water canoes in his garage, one for himself and one for his three very active boys, 9, 6, and 2.

For a biography of Gerald E. Nelson, please see page 7.

© Copr. 1949-1998 Hewlett-Packard Co.

SPECIFICATIONS
HP Model 3570A Network Analyzer

FREQUENCY RANGE: 50 Hz to 13 MHz.
OUTPUTS, CHANNELS A AND B: each is equal in amplitude and frequency to signal source output.
OUTPUT IMPEDANCE: 50 Ω or 75 Ω, ±2%, as specified at time of purchase.
MAXIMUM OUTPUT: 1 V rms into 50 Ω (75 Ω).
INPUTS, CHANNELS A AND B (electrically identical):
INPUT IMPEDANCE: 1 M Ω shunted by <30 pF.
INPUT SIGNAL RANGE: 1 μV to 1 V rms.
INPUT SELECTIVITY (switch-selected): 10 Hz, 100 Hz and 3 kHz bandwidth.

AMPLITUDE MEASUREMENTS

DISPLAY RANGE:
A or B AMPLITUDE FUNCTION: 0 to −100 dB.
B − A AMPLITUDE FUNCTION: −100 to +100 dB.
0-dB REFERENCE (MAX/REF INPUT VOLTAGE): 1 V, 0.1 V, and 0 dBm (referred to 50 Ω or 75 Ω).
AMPLITUDE ACCURACY:
ABSOLUTE: not specified; can be calibrated to source with front-panel adjustments.
RELATIVE (A, B, or B − A):

0 dB	−20 dB	−80 dB	−100 dB
±0.2 dB	±0.5 dB	±1.5 dB	

FREQUENCY RESPONSE:
A or B AMPLITUDE FUNCTION: ≤0.5 dB p-p error.
B − A AMPLITUDE FUNCTION: ≤0.1 dB p-p error.

PHASE MEASUREMENTS

PHASE REFERENCE: CHANNEL A.
−A REFERENCE OFFSET: 180° ±0.1°.
DISPLAY RANGE: −179.5° to +179.5° (display recycles).
DISPLAY RESOLUTION: 0.01°.
PHASE ACCURACY (operating at 25°C, ±5°C, channel B within 6dB of channel A):
AMPLITUDE RESPONSE:

±.4°	±.6°	±1°	not specified
0 −20	−70	−80	−100 dB

PHASE LINEARITY: ±0.2°
FREQUENCY RESPONSE (channels at 0 dB):

±.8°	±.2°	±1°
50 Hz 100 Hz	1 MHz	13 MHz

ANALOG OUTPUTS (rear panel)

AMPLITUDE FUNCTION: DC voltage proportional to logarithmic amplitude response in channel A, channel B, or B − A; 0.1 V/dB.
PHASE/DELAY: DC voltage proportional to phase difference or group delay, as determined by front-panel switch.
PHASE: 0.05 V/degree.
DELAY: 0 to 10 V for 0 to full-scale delay (5 full-scale ranges from 20 μs to 200 ms in decade steps).
DELAY (X-Y RECORDER): Same voltage as above.
Z-AXIS OUTPUT: Provides markers during frequency or amplitude sweeps. Intensifies center frequency (amplitude) or intensifies 11 evenly spaced markers. During limit test, intensifies limit points (no other markers available during limit testing).

GENERAL

DIMENSIONS: 16¾ in W x 5¼ in H x 21⅝ in D (42,6 x 13,3 x 54,3 cm).
WEIGHT: 42 lb net (18,9 kg)
POWER: 100–120, 220–240 V, 48–63 Hz, 230 VA. (Fan cooling).

OPTIONS

DELAY/LIMIT TEST AND OFFSET MEASUREMENTS.
DELAY MEASUREMENT.
FREQUENCY STEPS: 5 Hz to 200 kHz in 5, 10, 16.6, 20 sequence.
MEASUREMENT RANGE (determined by Frequency Step): 19.999 μs full-scale to 199.99 ms full-scale in 5 decade steps.
DELAY ACCURACY (% of reading + % of range):

0 dB	−40 dB	−80 dB	−100 dB
±(0.2 + 0.2)	±(0.5 + 0.5)	Not specified	

LIMIT TEST. May be performed on Amplitude, Phase, and Delay measurements. Limits (HI and LO) are entered as digital words through interface connector.
LIMIT STOP MODE: Sweep stops when limit transition occurs (GO to LO, HI to GO, etc.). Sweep continues when "Limit Sweep Restart" button is pressed.
LIMIT NO-STOP MODE: Sweep does not stop when limit transition occurs. Marker at Z-axis output identifies limit transition.
OFFSET MEASUREMENTS. Amplitude, Phase, and Delay reading can have numerical constant added to or subtracted from measurement data.
Front-panel entry: present reading is entered as offset constant by pressing "Enter Offset" button and switching "Absolute/Relative" switch to Relative.
Remote entry: present reading can be entered by remote command or arbitrary value can be entered as digital word through interface connector.
ISOLATED TWO-WAY INTERFACE (field installable): Provides digital outputs to interface bus (standard instruments have digital inputs only). Electrically isolates digital input and output lines from signal ground.
PRICES IN U.S.A.
Model 3570A Network Analyzer (50 Ω or 75 Ω): $5000.
Delay/Limit Test and Offset Measurement option: $400.
Isolated Two-way Interface: $500.
Typical system prices (including cables, manuals, etc.):
Model 3040A opt 200 (Network Analyzer with 3320A Frequency Synthesizer): $6900.
Model 3040A opt 500 (Network Analyzer with 3330B Automatic Synthesizer): $11,000.
Model 3041A (Network Analyzer with Delay/Limit/Offset option, 3330B Synthesizer, 3260A Marked Card Programmer): $14,000.
Model 3042A (Network Analyzer with two-way interface, 3330B Synthesizer, 9820A Calculator): $22,900.

MANUFACTURING DIVISION: LOVELAND DIVISION
815 Fourteenth Street, S.W.
Loveland, Colorado 80537

HEWLETT-PACKARD JOURNAL
OCTOBER 1972 *Volume 24 • Number 2*

***Technical Information from the Laboratories of Hewlett-Packard Company,** 1501 Page Mill Road, Palo Alto, California 94304 U.S.A.*
Hewlett-Packard S.A., 1217 Meyrin—Geneva, Switzerland
Yokagawa-Hewlett-Packard Ltd., Shibuya-Ku, Tokyo 151 Japan

Editorial Director: Howard L. Roberts
Managing Editor: Richard P. Dolan
Contributing Editors: Ross H. Snyder, Laurence D. Shergalis
Art Director, Photographer: Arvid A. Danielson
Art Assistant: Erica R. Helstrom
Administrative Services: Ruth G. Ross

© Copr. 1949-1998 Hewlett-Packard Co.