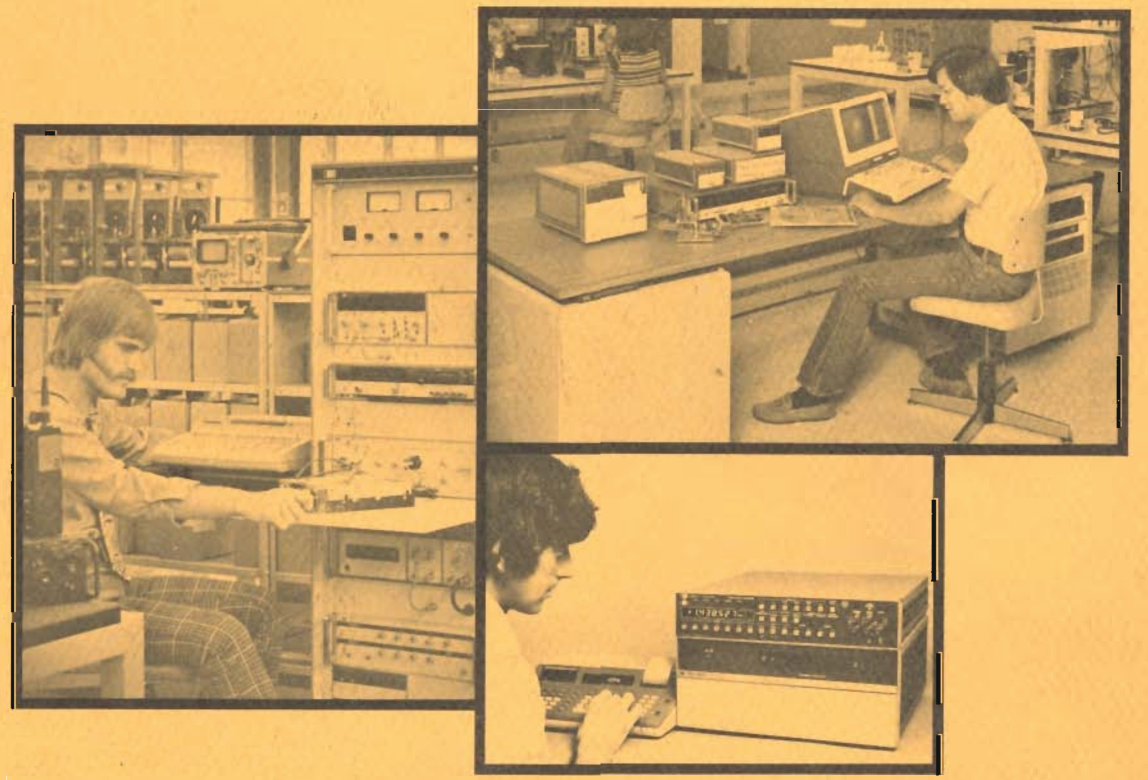
OLSEN

HP-IB

IMPROVING MEASUREMENTS IN ENGINEERING AND MANUFACTURING

...a collection of useful technical information



HEWLETT hp PACKARD

5952-0058

FOREWORD

The Hewlett-Packard Interface Bus (HP-IB) is Hewlett-Packard Company's implementation of IEEE Standard 488-1975, "Digital Interface for Programmable Instrumentation." This interface system concept represents a cooperative effort among many engineers working on the standardization effort. The basic concepts have been developed and refined significantly since the inception of the first proposals considered in early 1972. Standardizing the interface among various types of programmable measuring equipment manufactured by different companies was taken up by the International Electrotechnical Commission on the initiative of the German national committee. Hewlett-Packard's interface concepts served as the model for further elaboration at both national and international discussions. IEEE Standard 488 and the current draft document now under active development among IEC member nations represents the work of many individuals, many companies and nations.

The Hewlett-Packard Interface Bus is an easy-to-use, high-performance concept that links various combinations of instruments, calculators or computers and peripheral devices to function as automated instrumentation systems. The interface definition contains the mechanical, electrical, and functional specifications. Hewlett-Packard has contributed many of its interface design concepts to the industry for the sake of standardization and the ultimate benefit of equipment users. The sharing of this interface capability will allow customers to tailor solutions to their measurement and computational problems. This is a contribution that has long-range advantages for both Hewlett-Packard and its customers. The Hewlett-Packard Interface Bus makes it possible to put together more new and versatile instrumentation systems.

ACKNOWLEDGMENTS

The first technical article in this brochure, INTERFACE STANDARD 488 IN ACTION: CONCEPTS AND CAPABILITIES, Donald C. Loughry, has been reprinted by permission of ELECTRO '76, Professional Program Copyright 1976, and Institute of Electrical and Electronic Engineers, Inc., Boston, 1976. The second technical article, STANDARD INSTRUMENT INTERFACE SIMPLIFIES SYSTEM DESIGN, David W. Ricci and Gerald E. Nelson, has been reprinted by permission of ELECTRONICS, copyright McGraw-Hill, Inc., 1974. The third technical article, A STANDARD INTERFACE APPLIED TO MEASUREMENT SYSTEMS, Jane G. Evans and Howard Merrill, has been reprinted from ISA Proceedings of 22nd International Instrumentation Symposium, Fundamentals of Test Measurement, Volume 3, pp 17-27, 1976 with permission of the copyright holder: Instrument Society of America © 1976.

IEEE STANDARD 488 - 1975

If you wish to obtain a copy of IEEE Standard 488, it may be ordered from the IEEE Standards Office, 345 East 47th Street, New York, N. Y. 10017.

HP-IB

General Information

- Relationship to interface standards
- How HP-IB operates
- List of products available with HP-IB
- Computing controllers
- Applications information
- Preassembled systems

HEWLETT-PACKARD INTERFACE BUS

HP-IB Versatile interconnect system for instruments and controllers

- An easy-to-use, high performance concept that links instruments, desktop computers, minicomputers, and peripheral devices into automated measurement systems
- Very broad selection of HP-IB compatible instruments and accessory devices
- Wide choice of computers for the reduction, analysis, storage and management of measurement systems and resulting data
- Useful over wide range of problems from simple to very complex—add capabilities as your system requirements grow

There are many applications where the measurement power of interactive instruments can be further enhanced by coupling them to desktop or minicomputer. Operating in a remote mode can provide more exact, error-corrected results as compared with conventional manual operation techniques.

Presently, three major parameters combine to reduce significantly the engineering development costs of configuring measurement systems:

1) The Hewlett-Packard Interface Bus, also known as "HP-IB";
2) Distributed computing through the growing number of "smart" instruments with internal microprocessors;
3) The broad choice of computers, ranging from "friendly" easy-to-program desktop computers to more sophisticated computer systems capable of managing multi-station instrument clusters and complex data bases.

Relationship Between HP-IB and Other Interface Standards

Hewlett-Packard historically has been committed to the overall advancement of measurement technology, and has for some time been working on the problems of simplifying and standardizing means of instrumentation interfacing. An example of such an effort is the intimate involvement with the HP-IB from its conception at HP to its present status as a world instrumentation interface standard (IEEE 488–1978).

In mid-1972, Hewlett-Packard began to participate in various international standardization bodies. The U.S. Advisory Committee, composed of diverse interests represented by both users and manufacturers, first established initial goals—and then adopted the interface concept utilized by the HP Interface Bus as an appropriate starting point. A draft document was subsequently written and evaluated by members of the Committee, and then submitted as the U.S. Proposal to the IEC (International Electrotechnical Commission) Working Group in the autumn of 1972. Since then, the interface definition has undergone a number of minor changes to accommodate various needs at the international level.

In September 1974, the parent technical committee IEC TC76, approved the main interface draft document for a formal ballot among the member nations of the IEC. Balloting took place in 1976, and IEC recommendation 625–1 was adopted. The IEC recommendation, using a different connector, is totally compatible with the present definition of the HP-IB.

Meanwhile, the IEEE Standards Board approved IEEE Standard 488-1975 "Digital Interface for Programmable Instrumentation", first published in 1975 and again published in 1978 with minor editorial changes as IEEE Standard 488-1978. The IEEE standard is also fully compatible with the HP-IB. In January 1976, the American National Standards Institute adopted the IEEE Standard and published it as ANSI Standard MC 1.1.

The standardized interface concept is now well accepted, and more than 500 products utilizing the HP-IB concepts articulated in IEEE 488 are today available from more than 150 different manufacturers. As additional instrumentation interface standards evolve from the HP-IB, we will clearly indicate the relationship of the Hewlett-Packard Interface Bus to those standards—just as we have done with IEEE Standard 488, ANSI Standard MC 1.1 and IEC Publication 625.1.

Why The HP Interface Bus Name?

As the list of HP products available with the "new digital interface" grew, our customers sought a convenient way to identify those products having the interface capability. In response, in 1974 we adopted the name "Hewlett-Packard Interface Bus" or simply "HP-IB". We will continue to use the identifying name and this symbol:

HP-IB

Both will be used with appropriate HP products so that their interface capabilities may be readily identified.

The Hewlett-Packard Interface Bus fully complies with IEEE Standard 488. As such, it incorporates the mechanical, electrical and functional specifications of the Standard. A fourth and vital element of any interface system is the operational aspect of a product at both the human-machine interface and machine-machine interface at the HP-IB port. HP-IB capability provides additional user benefits that are beyond the scope of IEEE Standard 488. Typical user conveniences such as underscored program codes on the front panel of the instruments for easy programming, convenient data output formats, designed-in "Learn Mode" capabilities, complete support documentation in the form of programming and interfacing guides, application notes and operation manuals illustrate the added benefits for users of products with HP-IB capability.

Single Source Systems Approach

The decision to use a "system" instead of conventional manual methods must be based on an engineering evaluation of benefits versus costs. Among the many benefits associated with a systems approach:

- More consistent results in repeated measurements—a system is not subject to operator fatigue.
- Greater throughput because systems are generally faster.
- More thorough testing because system speed allows more parameters to be measured in a shorter time.
- Results expressed in engineering or scientific units, since many systems controllers are capable of on-line data manipulation.

HEWLETT-PACKARD INTERFACE BUS

Versatile interconnect system for instruments and controllers

- Greater accuracy because system errors can be measured automatically, stored and accounted for in the results.
- "Adaptive" data acquisition wherein a system can be programmed to branch to other measurements to help pinpoint when it senses an abnormal condition.
- Measurement results can be stored in computer memory or on hard copy.

It is our objective to facilitate the integration of instrumentation systems by providing users with instruments and computers designed for systems applications. Computers are designed with HP-IB options that allow easy hook-up to the bus and incorporate easy-to-use bus commands in their software. HP's policy when designing HP-IB compatible instruments is to eliminate interfacing ambiguities associated with controllers and instruments operating per the IEEE, ANSI and IEC standards by adopting consistent interface design guidelines.

Proper training on system components is very important for efficient utilization of any interface sytem. Therefore, we offer training at sales and service offices worldwide on HP desktop computers, computer systems and instruments as they relate to the HP-IB. (Refer to HP-IB Training, Page 26). In the area of HP-IB support documentation, we offer Operating and Service Manuals with programming information, Instrument/Controller Introductory Operating Guides, Quick Reference Guides and Application Notes. Technical assistance during system development is available from resident systems engineers specialized in desktop computers, computer systems and instruments at most local sales and service offices.

How The HP Interface Bus Operates

All active interface circuitry is contained within the various HP-IB devices, and the interconnecting cable (containing 16 signal lines) is entirely passive. The cable's role is limited to that of interconnecting all devices in parallel, whereby any one device may transfer data to one or more other participating devices.

Every participating device (instrument, controller, accessory module) must be able to perform at least one of the roles of TALKER, LISTENER or CONTROLLER. A TALKER can transmit data to other devices via the bus, and a LISTENER can receive data from other devices via the bus. Some devices can perform both roles (e.g., a programmable instrument can LISTEN to receive its control instructions and TALK to send its measurement).

A CONTROLLER manages the operation of the bus system primarily by designating which devices are to send and receive data, and it may also command specific actions within other devices.

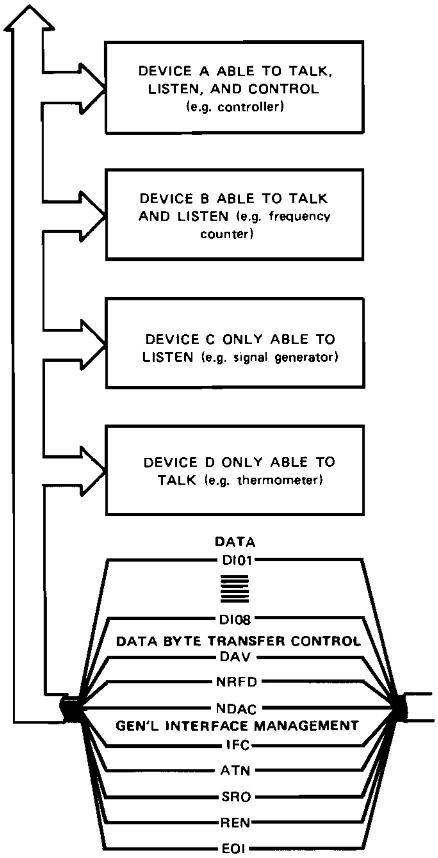
A minimum HP-IB system configuration consists of one TALKER and LISTENER, without a CONTROLLER. In this configuration, data transfer is limited to direct transfer between one device manually set to "talk only" and one or more devices manually set to "listen only" (e.g., a measuring instrument talking to a printer, for semi-automatic data logging).

The full flexibility and power of the HP-IB become more apparent, however, when one device which can serve as CONTROLLER/TALKER/LISTENER (e.g., calculator or computer) is interconnected with other devices which may be either TALKERS or LISTENERS, or both (e.g., frequency synthesizers, counters, power meters, relay actuators, displays, printers, etc.), depending on the application. An HP-IB *controller* participates in the measurement by being programmed to schedule measurement tasks, set up individual devices so that they can perform these tasks, monitor the progress of the measurement as it proceeds, and interpret the results of the measurement. HP offers controllers which can be programmed in higher level languages such as BASIC, FORTRAN and HPL.

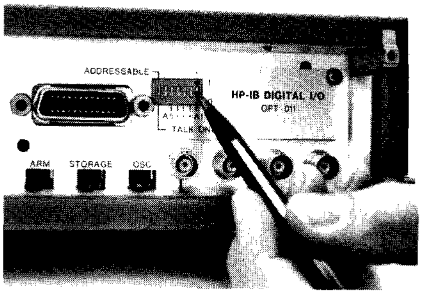
HP-IB Connections and Structure

The HP-IB has a party line structure where all devices on the bus are connected in parallel. The 16 signal lines within the passive interconnecting HP-IB cable are grouped into three clusters according to their function as follows:

1) Data Bus (8 signal lines)
2) Data Byte Transfer Control Bus (3 signal lines)
3) General Interface Management Bus (5 signal lines)



Interface connections and bus structure.



Rear panel switches are set so instrument will either be addressable by controller in a multi-device system, or will simply "talk only" to another device such as a printer.

The DATA BUS consists of eight signal lines which carry data in bit parallel, byte serial format across the interface. These signal lines carry addresses, program data, measurement data, universal commands and status bytes to and from devices interconnected in a system. Identification of the type of data present on the DIO signal lines is indicated by the ATN (attention) signal. When the ATN signal is true (asserted) either addresses or universal commands are present on the data bus and all connected devices are required to monitor the DIO lines. When the ATN message is false, then device dependent data (e.g., programming data) is carried between devices previously addresses to talk and listen.

Transfer of each byte on the Data Bus is accomplished via a set of three signal lines: DAV (data valid), NRFD (not ready for data), and NDAC (not data accepted). These signals operate in an interlocked handshake mode. Two signal lines, NRFD and NDAC, are each connected in a logical AND (wired OR) to all devices connected to the interface. The DAV signal is sent by the talker and received by potential listeners whereas the NRFD and NDAC signals are sent by potential listeners and received by the talker.

The General Interface Management Lines manage the bus to effect an orderly flow of messages. The IFC (interface clear) message places the interface sytem in a known quiescent state. SRQ (service request) is used by a device to indicate the need for attention or service and to request an interruption of the current sequence of events. REN (remote enable) is used to select between two alternate sources of device program data. EOI (end or identify) is used to indicate the end of a multiple byte transfer sequence or, in conjunction with ATN, to execute a polling sequence.

It is not possible in this limited space to go into detail on each signal line's role. But you should note that every HP-IB device need not be able to respond to all the lines. As a practical and cost-effective matter, each HP-IB device will usually be designed to respond only to those lines that are pertinent to its typical function on the bus. (Details appear in each device's operation manual).

Products For "Do-It-Yourself" Unbundled HP-IB Systems

Hewlett-Packard has an extremely broad range of HP-IB instruments and controller capabilities, as indicated on the table below—capabilities you can use to solve a wide variety of measurement problems via HP-IB tailored system solutions.

Each bench instrument is, by itself, an exceptional performer in terms of providing signals, making measurements, or recording results. Each has the additional capability

Individual Hewlett-Packard Products Available With HP-IB Capability

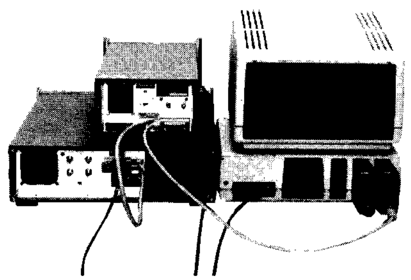
Products related to	Model	Product name/characteristics
Stimulus	3320B Option 007	Frequency Synthesizer: 0.01 Hz to 13 MHz
	3325A	Synthesizer/Function generator/Sweeper: 1 μHz to 22 MHz
	3330B	Automatic Synthesizer/Sweeper: 0.1 Hz to 13 MHz
	3335A	Synthesizer/Level Generator: 200 Hz to 80 MHz
	3336A	Synthesizer/Level Generator: 10 Hz to 20.9 MHz
	4140A	PA Meter/DC Voltage Source
	5359A	Time Synthesizer: 1 ns accuracy
	6002A Option 001	DC Power Supply: 200 W autoranging
	6129C Option J99	Digital Voltage Sources: ± 50 Vdc at 5 A (requires 59301A Converter)
	6130C Option J99	Digital Voltage Source: ±50 Vdc at 1A (requires 59301A Converter)
	6131C Option J99	Digital Voltage Source: ±100 Vdc at 0.5 A (requires 59301A Converter)
	6140A Option J99	Digital Current Source: ±100 mA at 100 Vdc (requires 59301A Converter)
	6940B	Multiprogrammer (requires 59500A interface)
	6942A	Multiprogrammer
	8016A Option 001	Word Generator: 9 x 32 bit
	8018A Option 001	Serial Data Generator: 50 MHz, 2048-bit memory
	8160A	Programmable Pulse Generator: 20 ns to 999 ms period
	8165A	Programmable Signal Source: 0.001 Hz to 50 MHz
	8170A	Logic Pattern Generator: 8 x 1024/16 x 512 bit
	8620C Option 011	Sweep Oscillator: 10 MHz to 22 GHz
	8660A Option 005	Synthesized Signal Generator: 10 kHz to 2.6 GHz
	8660C Option 005	Synthesized Signal Generator: 10 kHz to 2.6 GHz
	8662A	Synthesized Signal Generator: 10 kHz to 1280 GHz
	8671A	Microwave Frequency Synthesizer: 2 to 6.2 GHz
	8672A	Synthesized Signal Generator: 2 to 18 GHz
Display	1350S	Graphics Display System
	5150A Option 001	Alphanumeric Thermal Printer: 20 Columns
	9871A Option 001	Character-Impact Printer: 132 columns
	7225B	Graphic Plotter: ISO A4 and 8½ X 11 inch chart size
	7245A	Thermal Plotter/Printer: Vector graphics, matrix printing
	9872B, 9872S	Graphics Plotter: multicolor (4 colors) programmable
	59304A	Numeric Display: 12 LED characters, decimal point
		Display also via Desktop Computers and Computer Systems
Switching Scanning Translation or Timing	2240A	Measurement and Control Subsystem
	3495A	Scanner: to 80 channels, low thermal; (up to 40 relay channels)
	3754A	25 MHz Access Switch (requires 3755A switch controller)
	3756A	90 MHz Switch (requires 3755A)
	3757A	8.5 MHz Access Switch (requires 3755A)
	3777A	Telecommunications Channel Selector: up to 30 channels; dc to 110 kHz
	6940B	Multiprogrammer (requires 59500A interface)
	6942A	Multiprogrammer
	9412A	Modular Switch (requires 9411A switch controller)
	9413A	VHF Switch (requires 9411A)
	9414A	Matrix Switch (requires 9411A)
	11713A	Attenuator/Switch Driver (controls coax switches and step attenuators)
	12050A	Fiber Optic HP-IB Link
	37201A	HP-IB Extender
	59301A	ASCII-to-Parallel Converter: string to 16 characters
	59303A	Digital-to-Analog Converter
	59306A	Relay Actuator: for programmable switches, attenuators
	59307A	VHF Switch: two 50 Ohm, bidirectional, dc to 500 MHz
	59308A	Timing Generator
	59309A	Digital Clock: month, day, hour, minute, second
	59313A	Analog-to-Digital Converter
	59403A	HP-IB/Common Carrier Interface: RS232C or CCITT V24
	59501A	Power Supply Programmer: isolated D-to-A converter ± 10 V dc at 10 mA
Control and Computation	9815A/S	Desktop Computer (uses 98135A Interface)
	9825A/S	Desktop Computer (uses 98034A Interface)
	9835A/B	Desktop Computer (uses 98034A Interface)
	9845B/T	Desktop Computer System 45 (uses 98034A Interface)
	HP1000 M-series	Computers (2105A, 2108M & 2112M; use 59310B Interface)
	HP1000 E-series	Computers (2109E & 2113E; use 59310B Interface)
	HP 1000 F-series	High-performance computers (2111F and 2117F use 59310B Interface)
Interface Cabling	10631A	HP-IB Interconnection Cable: 1 m (3.3 ft)
	10631B	HP-IB Interconnection Cable: 2 m (6.6 ft)
	10631C	HP-IB Interconnection Cable: 4 m (13.2 ft)
	10631D	HP-IB Interconnection Cable: 0.5 m (1.6 ft)
HP-IB Extension	12050A	Fiber Optic HP-IB Link
	37201A	HP-IB Extender
	59403A	HP-IB Common Carrier Interface
Design and Servicing	59401A	Bus System Analyzer

HEWLETT-PACKARD INTERFACE BUS

Versatile interconnect system for instruments and controllers

HP-IB

Individual products (cont.)



Rear view of 3-device HP-IB Bench System

which allows its use in HP-IB instrumentation systems — either in "do-it-yourself" systems configured and assembled by users themselves, or in some of the standard systems which are designed, preassembled and supported by HP. While the HP-IB Interface is optional in many instruments, it is increasingly becoming "standard" in some of the newer products.

Most principal functions on the instruments are HP-IB programmable. New instruments incorporate conveniences for the programmer such as underscored program codes on front panels to enable quick reference programming and complete HP-IB documentation to facilitate integration into a system.

Just as with electronic instruments, HP controllers for use with HP-IB are all proven performers. Regardless of the need for reducing, analyzing, storing or managing measurement systems data, HP has a controller that's right for your application.

Individual Hewlett-Packard Products Available with HP-IB Capability (cont.)

Products related to:	Model	Product name/characteristics
Measurement	436A Option 022	Power Meter: –70 dBm to +44 dBm, to 18 Ghz
	1602A Option 001	Logic State Analyzer: 64 x 16 bit memory
	1610A/B Option 003	Logic State Analyzer: 64 x 32 bit memory
	1615A Option 001	Logic Analyzer: 256 x 24 bit memory
	1640A Option 001	Serial Data Analyzer: 2048 bit memory
	2240A	Measurement & Control Subsystem
	2804A Option 010	Quartz Thermometer: 0.05°C accuracy
	3437A	System Digital Voltmeter: high speed, 3½ digits
	3438A	Digital Voltmeter: low-cost, 3½ digits
	3455A	Digital Voltmeter: 5½ or 6½ digits, auto calibration
	3490A Option 030	Digital Voltmeter: 5 digits, self test
	3582A	2-channel Real Time (FFT) Spectrum Analyzer: 20 mHz to 25.6 kHz
	3585A	Swept Spectrum Analyzer: 20 Hz to 40 MHz, 3 Hz BW, 0.5% amplitude accuracy
	3586A/B/C	Selective Level Meter: 50 Hz to 32.5 MHz
	3745A	25 MHz Selective Level Measuring Set: CCITT FDM systems
	3745B	25 MHz Selective Level Measuring Set: Bell FDM systems
	3747A	90 MHz Selective Level Measuring Set: CCITT FDM systems
	3747B	90 MHz Selective Level Measuring Set: Bell FDM systems
	3771A Option 005	Data Line Analyzer: CCITT measurement standards
	3771B Option 005	Data Line Analyzer: Bell measurement standards
	3779A	Primary Multiplex Analyzer: CEPT 2 Mb/s PCM systems
	3779B	Primary Multiplex Analyzer: Bell 1.5 Mb/s PCM systems
	4191A	RF Impedance Analyzer
	4262A Option 101	Automatic LCR Meter
	4270A Option 101	Automatic Capacitance Bridge
	4271B Option 101	1 MHz Digital LCR Meter
	4272A Option 101	1 MHz Preset C Meter
	4274A Option 101	Multifrequency LCR Meter: 10 Steps, 100 Hz to 100 KHz
	4274A Option 102	Multifrequency LCR Meter: as above, but with isolation
	4275A Option 101	Multifrequency LCR Meter: 10 steps, 10 kHz to 10 MHz
	4275A Option 102	Multifrequency LCR Meter: as above, but with isolation
	4282A Option 101	Digital High Capacitance Meter
	4943A Option 010	Transmission Impairment Measurement System (TIMS)
	4944A Option 010	Transmission Impairment Measurement System (TIMS)
	5312A	HP-IB interface (Talker) for 5300B Counter System
	5328A Option 011	Universal Counter: to 512 MHz, 10 ns time interval
	5340A Option 011	Automatic Microwave Counter: 10 Hz to 18 GHz
	5341A Option 011	Automatic Microwave Counter: high speed, to 4.5 GHz
	5342A Option 011	Automatic Microwave Counter: 10 Hz to 18 GHz
	5345A Option 011,012	General Purpose Plug-In Counter
	5353A	Channel C Plug-in for 5345A
	5354A	4 GHz Frequency Converter for 5345A
	5355A	Automatic Frequency Converter
	5363A/B	Time Interval Probes
	5370A	Time Interval Counter: ±20 ps single-shot resolution
	5420A	Digital Signal Analyzer (requires 10920A cards)
	5501A Option 251	Laser Transducer: for accurate positioning measurements
	8501A	Storage Normalizer for 8505A RF network analyzer
	8503A & 8503B	S-Parameter Test Set: 50 or 75 Ohm, for 8505A
	8505A	RF Network Analyzer: 500 kHz to 1.3 GHz
	8566A	Spectrum Analyzer: 100 Hz to 22 GHz
	8568A	Spectrum Analyzer: 100 Hz to 1.5 GHz
	8901A	Modulation Analyzer: 150 kHz to 1,3 GHz
		Also see models 2240A, 6940B, and 6942A.
Storage	3964A Option 007	Instrumentation Tape Recorder: 4 channel
	3968A Option 007	Instrumentation Tape Recorder: 8 channel
		Storage also via Desktop Computers and Computer Systems

Standard Bundled HP-IB Measurement Systems

Many application requirements can be satisfied with a standard HP-IB measurement system — already preassembled, tested, and documented by Hewlett-Packard. Preconfigured systems save you design and setup time, and HP guarantees overall specified system performance. Installation and service contracts are available from your local HP Sales and Service Office.

Standard HP-IB Measurement Systems

Application	Model	Controller	System name/characteristic
Data Logging and Acquisition	3052A	9825/35/45	Automatic Data Acquisition: fast and precise low-level measurements, powerful computation.
	5391A	9825	Frequency and Time Data Acquisition Systems: over 50,000 four-digit frequency and time interval measurements per second
	9875A	9825/35/45	Tape Cartridge Unit
Network Analysis	3040A	9825	Network Analyzer: complete amplitude and phase characterization, 50 Hz to 13 MHz. Group delay optional.
	3042A	9825	Automatic Network analyzer: same as 3040A, and includes the faster 9825A as computing controller.
	8409A	9825	Automatic Microwave Network Analyzer: measures transmission and reflection parameters, 110 MHz to 18 GHz.
	8507B	9825	Automatic RF Network Analyzer: measures complex impedance, transfer functions, group delay; 500 kHz to 1.3 GHz.
Spectrum Analysis	3044A	9825	Spectrum Analyzer: precise amplitude and frequency measurements, 10 Hz to 13 MHz.
	3045A	9825	Automatic Spectrum Analyzer: same as 3044A, and includes the faster 9825A as computing controller.
	8581A	9825	Automatic Spectrum Analyzer: covers 100 Hz to 1.5 GHz; exceptional frequency tuning accuracy and resolution.
	8582A	9825	Automatic Spectrum Analyzer: covers 100 Hz to 22 GHz; exceptional frequency tuning accuracy and resolution.
Frequency Stability Analysis	5390A	9825	Frequency Stability Analyzer: short and long-term characterization of precision frequency sources, 500 kHz to 18 GHz.
Transceiver Testing	8950B	9825	Automatic Transceiver Test System: for AM and FM transceivers, 2 to 1000 MHz, transmitters to 100 W.
Circuit Testing	DTS-70	1000	Digital Test System: fast, accurate fault location on loaded printed circuit boards.
	3060A	9825	Analog and Digital Test System: Fast, accurate fault location on loaded printed circuit boards
Digital IC Testing	5046A	9825	Digital IC Test System: Reduces production costs through the isolation of faulty components prior to printed circuit board loading.
FDM Network Surveillance	37013A	1000	Frequency Division Multiplex Network Surveillance System: remote capability based on HP 1000 Computer
	37014A	9835	Frequency Division Multiplex Network Surveillance System: remote capability based on 9835A Desktop Computer

HEWLETT PACKARD INTERFACE BUS

Versatile interconnect systems for instruments and controllers

HP-IB Training and Support Available from Hewlett-Packard

Hewlett-Packard has field sales people trained in electronic instruments, desktop computers and computer systems to assist users configure HP-IB measurement systems. Also available for technical consultation are computing controller systems engineers and HP-IB instrumentation specialists.

Listed below are training courses on HP-IB computing controllers and instruments available on a regular basis.

Computer Systems

Course Name	Duration
• Introduction to HP1000 Minicomputers	4 days
• Disc-Based RTE System Course	10 days
• Memory-Based RTE System Course	10 days
• HP-IB in a Minicomputer Environment	4 days

Desktop Computer Systems

Course Name	Duration
• HP Basic Programming	4 days
• 9825A Operation and Programming	4 days
• 9835A Operation and Programming	5 days
• 9845B Operating and Programming	5 days
• 9835A Assembly Level Programming	5 days

Electronic Instruments

Course Name	Duration
• HP-IB Seminar	2 days

Service and Warranty Considerations

Hewlett-Packard has dedicated Measurement System Service people who perform on-site maintenance on both customer configured systems as well as HP configured systems, irrespective of whether an HP desktop or minicomputer is used. Service contract coverage is available to meet your specific measurement system service needs and can be tailored to include extended warranty, calibration and extended hours of coverage. Contact your local sales and service office for further information on HP-IB service contract information.

Every HP-IB device and HP configured system carries a standard Hewlett-Packard warranty appropriate to that product. The warranty period for each product will be provided on request at the time of sale and is specified in documentation supplied with the product. HP takes responsibility for standard HP-IB systems performing as specified. However, software or interfacing which has not been provided by Hewlett-Packard as part of a standard system delivered by HP is not covered by this warranty.

In all cases, overall operational responsibility for those HP-IB *systems assembled by a customer* from individual HP-IB devices shall rest with the customer.

HP-IB Specifications Summary

Interconnect Devices:

Up to 15 maximum on one contiguous bus.

Interconnection Path:

Star or linear bus network; total transmission path length 2 metres times number of devices or 20 metres, whichever is less (See HP-IB Extension Capabilities for extending operating distance.)

Message Transfer Scheme:

Byte-serial, bit-parallel asynchronous data transfer using locked 3-wire handshake technique.

Data Rate:

One megabyte per second maximum over limited distance; 250-500 kilobytes per second typical over full transmission path (actual data rate depends on individual device characteristics).

Address Capability:

Primary addresses, 31 TALK and 31 LISTEN; secondary (2-byte) addresses, 961 TALK and 961 LISTEN. Maximum of 1 TALKER and up to 14 LISTENERS at a time.

Control Shift:

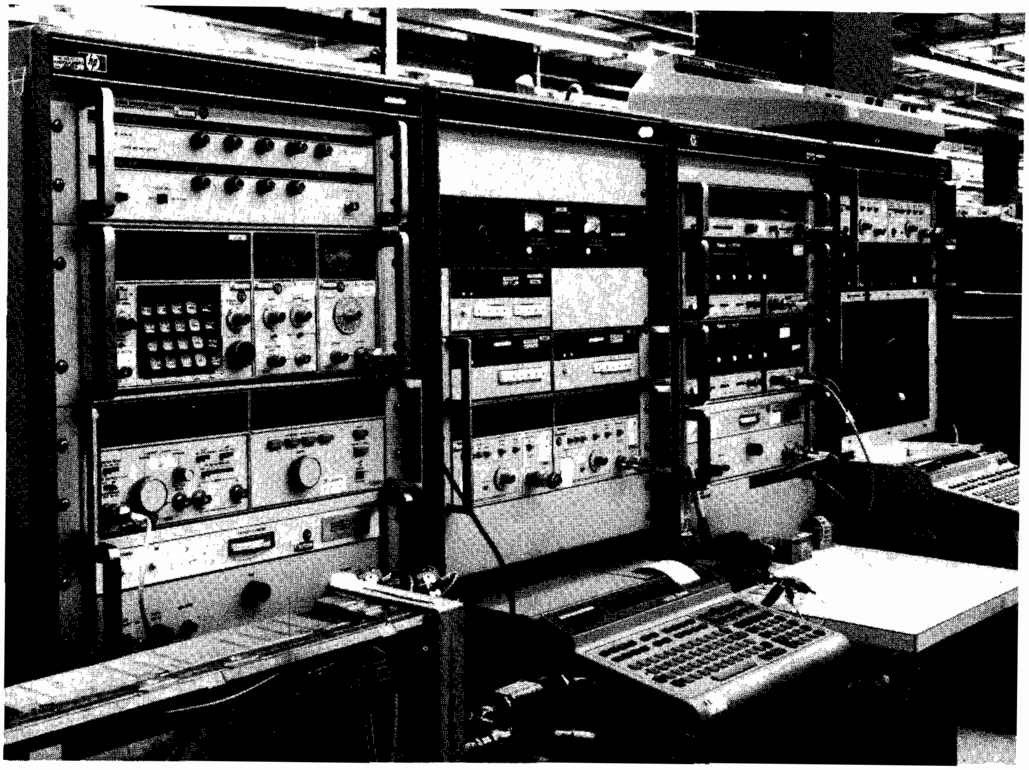
In systems with more than one controller, only one can be active at a time. A currently active controller can pass control to another, but only designated system controller can assume control over others.

Interface Circuits:

Driver and receiver circuits are TTL-compatible.

Connector Lock Screw Compatibility

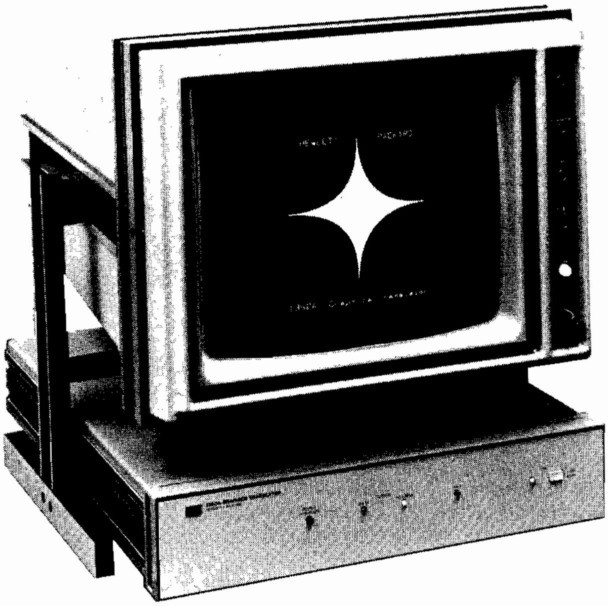
HP-IB products delivered now and in recent years are equipped with connectors having ISO *metric-threaded* lock screws, and stud mounts. (Very early HP-IB products have non-metric parts, but are readily distinguished from the metric by color: metric threaded parts are black and stamped with the letter "M" whereas non-metric parts have a shiny nickel finish). HP-IB Metric Conversion Kit (P/N 5060-0138) is available to convert these early instruments.



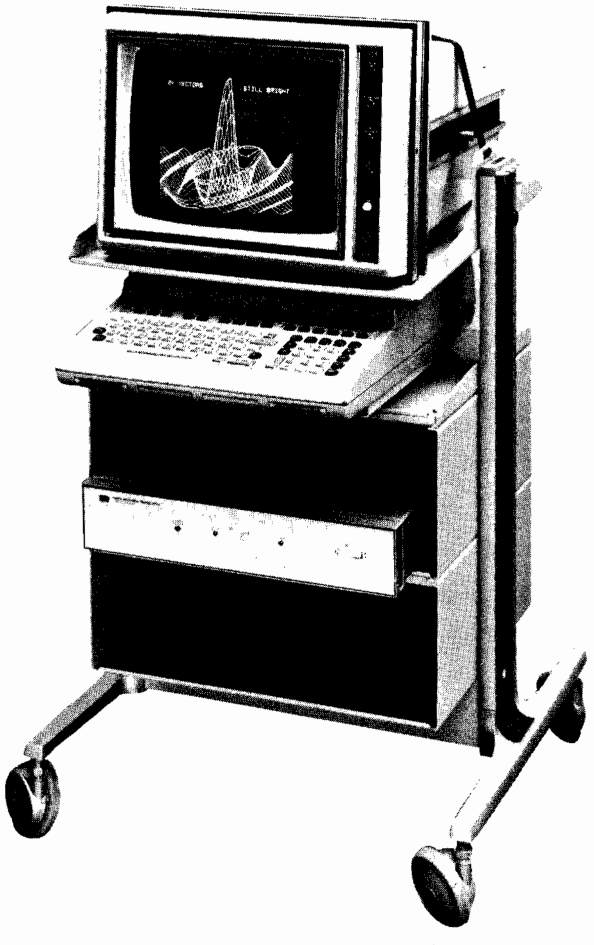
HEWLETT-PACKARD INTERFACE BUS

Versatile interconnect system for instruments, computers, and controllers

Model 1350S



The 1350S Display System includes a 1311 X-Y Display, a 1350A Graphics Translator, an interconnecting cable, and a binder for instruction manuals.



A complete, mobile softcopy graphics test station can be assembled using HP's Model 1008A C01 Testmobile with the options required to fit your application. With the configuration shown, HP small screen displays or the 1311 Large Screen Display may be used, and a slide-out tray is provided for the 9825A Computing Controller.

1350S Display System

- High speed CRT display for HP-IB systems
- Operates with 9825A or 9835A
- System 1000 graphics accessory
- Remote display using RS-232C (Option 001)

Model 1350S is a high resolution display system that generates bright, sharp vectors and alphanumerics at high writing speeds. The system includes a graphics translator with the high speed HP-IB (IEEE 488-1975) interface that accepts digital data from desktop computers and minicomputers or microprocessor controlled systems. Data is stored in a digital memory which continually refreshes the display, without placing a load on the controller or computers.

High Speed Graphics

The digital memory of the 1350S can be addressed in random fashion. Thus, any number of vectors or characters can be entered without erasing or rewriting all 2048 memory locations. For example, one curve on a graph can be updated while other picture elements remain unchanged. Random-access memory also increases the speed at which the graphics portion of a system can be operated. The 1350S is ideally suited for real time applications.

Versatile Operation

Up to 32 files are available for storing text, graticules, or other segments of the picture. A file can be repetitively flashed to alert an operator to abnormal system operation such as an out-of-tolerance measurement. File Management capability allows the 1350S to display different information on up to three additional CRT's.

Binary Tape Option

A 10184A Binary Tape option simplifies programming the 1350S when it is used with the 9825A Desktop Computer. It uses most of the same program commands as the 9872A or 9862A plotters. Additional commands are provided to blank and view individual files as well as flash segments of the picture. The 10184A binary program resides in 9825A memory, occupying 3806 bytes.

RS-232C Interface Option

An RS-232C interface option (001) can be substituted for the standard HP-IB interface. Option 001 is a teletypewriter interface (standard EIA RS-232C/CCITTV-24).

Option 001 operates in an asynchronous, receive only mode. It provides a system clock at standard baud rates from 110 to 9600 that can be used to clock the teletypewriter interface in the controller or computer.

NOTES

For complete description and specifications, request technical data sheets for the 1350S Display System, the 1311 Display, and the 1350A Graphics Translator. An HP-IB cable is not supplied with the 1350S, and must be ordered separately.

System Options	**Price**
001: RS-232C interface instead of standard HP-IB	$100
010: tilt stand for 1311 only	add $200
184: 10184 binary tape	add $100
510: 1310A 19 in. X-Y display instead of 1311	add $300
517: 1317A 17 in. X-Y display (rack mounting configuration) instead of 1311	add $250
521: 1321A 21 in. X-Y display instead of 1311	add $900
604: P-4 phosphor display, without graticule	add $30
908: rack mounting parts for 1311 and 1350A	add $75

Ordering Information

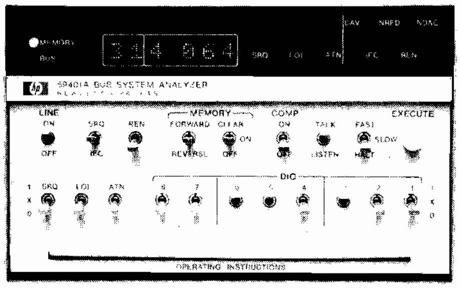
1350S Display System	$7500
1008A Option C01 Testmobile	$550

HEWLETT-PACKARD INTERFACE BUS

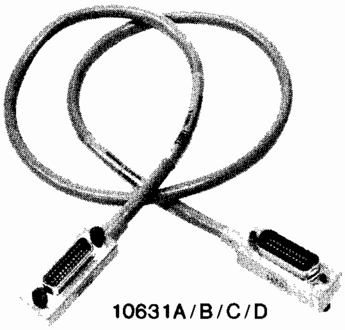
Versatile interconnect system for instruments and controllers

HP-IB

Bus system analyzer, cables & accessory modules



59401A



10631A/B/C/D

59401A Bus System Analyzer

The HP-IB (IEEE 488) concept has greatly simplified many of those things which have in the past made instrument interfacing a burdensome task. Even so, software errors can occur if the system designer does not completely understand the bus system or the capabilities of the instruments and other devices being interfaced. Hardware problems can occur if the instruments/devices are not functioning properly, or if they are not completely compatible with the bus standard.

The 59401 Bus System Analyzer is especially useful in design and service work. It simplifies and speeds up the diagnosis of software and hardware problems by allowing the user to see the status of all bus lines, including the actual characters on the bus data lines. Because the 59401A can also drive all bus lines, it can completely exercise another Talker, Listener or Controller—which is especially useful in verifying compatibility of new or user-designed products with the HP-IB.

There are several choices of analyzer operating speed. It may be operated at one character at a time (useful for software debugging), at 2 characters per second, or at regular bus speed. It may also be operated at a variable rate as determined by the external clock input.

The analyzer's 32 character memory can be used to store bus characters in the Listen mode, or to output characters to the bus in the Talk mode. When the analyzer is in the Compare mode, a stream of bus traffic may be stopped on a pre-selected character—and at that time, a trigger pulse is available, which is very useful when analyzing transient or timing problems related to the bus.

59401A Specifications

Display: monitors all bus lines. Represents data lines, any memory location, or DIO front panel switch settings; in octal code and ASCII character.

Listen mode: stores up to 32 characters of bus traffic in memory for real time and repetitive testing. In compare mode, halts bus traffic when a selected character is present, and user can display any one of the previous 31 characters stored in memory.

Timing: accept <750 ns; ready <750 ns.

Talk mode: bus lines can be driven directly from front panel switches; memory can be loaded from front panel switches for driving bus with a 32 character sequence.

Timing: (1) data changed >500 ns before DAV pulled low; (2) ATN driven low >1 μs before DAV pulled low; (3) DAV driven high <700 ns after NDAC is false; (4) DAV driven low <700 ns after NRFD is false, if conditions 1 and 2 are met.

Operating speeds: one character at a time, 2 characters per second, regular bus speed, or variable rate determined by external clock input; in either Listen or Talk mode.

External clock input: 1 standard power TTL gate input; ≤10 MHz repetition rate.

Compare output: provides 1 standard power TTL gate output (LOW TRUE) sync pulse when bus character is same as front panel switches.

HP-IB load: 1 bus load (capable of driving 14 other bus devices).

General

Temperature ranges: operating, 0 to 50°C; storage, −40 to +75°C.

Humidity: 95% relative, 0 to 40°C.

Power requirements: 100, 120, 220 or 240 V +5%, −10%; 48 to 66 Hz; ≤42 VA.

Size: 145.5 H, 205.1 W, 495.3 mm D (5.730" x 8.075" x 19.500")

Weight: net, 5.64 kg (12.44 lb).

Options and Accessories	Price
5061-0089 front handle kit	$15
10631B 2 m (6.6 ft) bus cable, furnished	N/C
59401A Bus System Analyzer	**$2800**

HP-IB Interconnection Cables

Cables for interconnecting HP-IB devices are available in four different lengths. The connector block at both ends of each HP-IB cable (photo above) has a plug on one side and a matching receptacle on the other, so that several cables may be conveniently connected in parallel, thus simplifying system interconnection. Lock screws provide for secure mounting of each connector block to an HP-IB instrument, or to another cable connector block.

SPECIAL NOTE: HP-IB cables are not *included with individual HP-IB devices, and must be ordered separately (exception: HP-IB computing controller interfaces include cable with connector).*

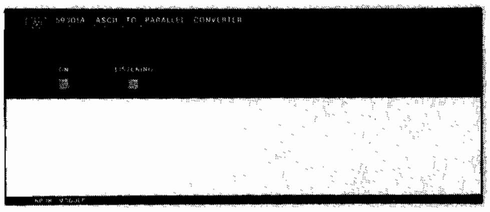
Ordering Information	Price
10631A HP-IB Cable, 1 m (3.3 ft)	$60
10631B HP-IB Cable, 2 m (6.6 ft)	$65
10631C HP-IB Cable, 4 m (13.2 ft)	$75
10631D HP-IB Cable, 0.5 m (1.6 ft)	$60

HP-IB Accessory Modules

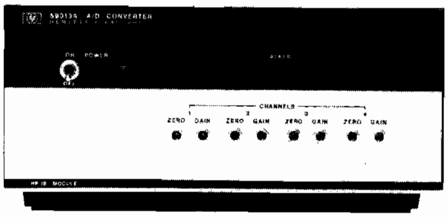
Modules in the HP 59300, 59400 and 59500-series are ideal building blocks for use with instruments to extend measurement capabilities. Modules listed here can be interconnected via the HP-IB to HP measuring instruments, signal sources and recording devices capable of operating directly on the HP-IB. In addition, these modules frequently serve as useful ways to interconnect with devices which are not themselves capable of direct HP-IB operation.

Instrument requirements differ. Some only output or accept data on the HP-IB. Others can be remotely programmed by ASCII characters sent along the HP-IB. These modules can work with instruments on any of these levels with or without a controller. Each module having controls can be operated stand-alone from its front panel, or it can be placed in automatic operation under program control.

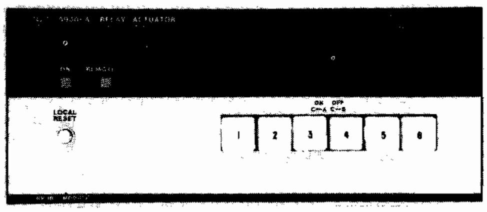
Module provision for stand-alone, local operation also has important system benefits. The operator can set up and check out the system under manual control, avoiding otherwise complex and time consuming error tracing. Each module has status indicator lights that make it easy to monitor operation.



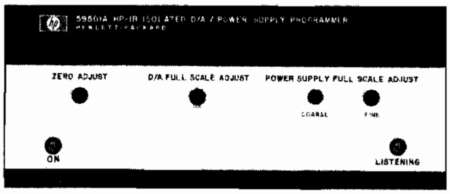
59301A



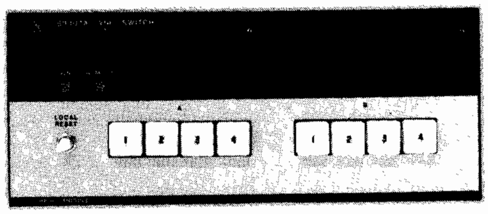
59313A



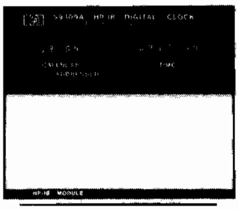
59306A



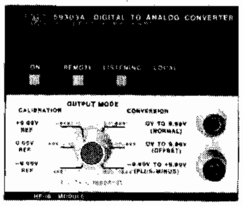
59501A



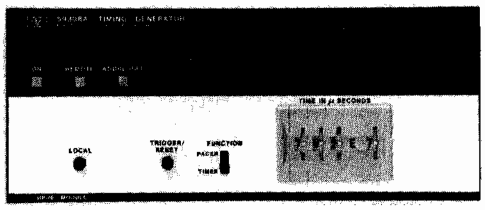
59307A



59309A



59303A



59308A

59301A ASCII-to-parallel Converter

Accepts byte-serial ASCII characters from the HP-IB and converts them to parallel output. A string of up to 16 characters terminated by linefeed is converted to 1-2-4-8 BCD and placed on the output lines; the ASCII linefeed character causes a print command (strobe) to be output by the 59301A.

With the 59301A, instruments with the HP-IB interface can be operated with HP 5050B/5055A Printers (requires two output cables, HP 562-16C, not furnished). Or, the 59301A can be used with HP 6129C thru 6131C and 6140A (Option J99) digitally-controlled power supplies for HP-IB programmable voltage and current. The 59301A can additionally be used to control other functions using its hexadecimal format.

59303A Digital-to-analog Converter

Accepts an ASCII string and converts any three consecutive digits to a dc voltage accurate to 0.1% in 30 μs. Fully programmable via the HP-IB or operates stand-alone from the front panel. Offers three output modes for conversion: normal, offset, or plus-minus (9.99 volts to −9.99 volts) to make it convenient for operating strip chart recorders.

A primary application for the HP 59303A is to present on a logging device the data points being taken during a measurement, such as with the HP 5345A Counter. No controller is required for operation. Compatible logging devices include strip chart recorders, X-Y plotters, and displays.

59306A Relay Actuator

Has six Form-C relays that provide for control of external devices either manually from front panel pushbuttons or remotely from the HP-IB. Relay contacts are specified to switch 24 V dc or 115 V ac @ 0.5 A. Use the 59306A with HP 8761A/B SPDT switches for HP-IB programmable microwave switching dc-18 GHz; use it with HP 8494 thru 8496G/H attenuators for HP-IB programmable attenuation dc-18 GHz (external power supply required).

59307A Dual VHF Switch

This module offers a pair of single throw 4-pole switches (dc to 500 MHz, 50 ohm) optimized for fast risetime (1 ns) pulse waveforms. Switches are independent and bidirectional, and can be operated either from front panel pushbuttons or remotely from the HP-IB.

59308A Timing Generator

Has two modes of operation—a pacing function which provides output at a specified rate, and a timing function which provides a delay with respect to a trigger for a specified period of time. Timed intervals can be selected by thumbwheel switches on the front panel, or can be programmed remotely from the HP-IB. Times from 1 μs to more than a day are available. Trigger inputs are available via HP-IB commands and rear panel connector. Timing outputs are available for both TTL and ECL levels, with switch selection of a squarewave or pulse output positive or negative-going edge. Output pulses are 500 ns ± 100 ns wide, and rise time is <50 ns.

HEWLETT-PACKARD INTERFACE BUS

Versatile interconnect system for instruments and controllers

HP-IB Modules

59309A Digital Clock

Displays month, day, hour, minute and seconds, and upon command will output time via the interface bus. Time can be set into the clock by local control, or by remote commands received from the HP-IB. The clock accepts a small internal battery which can provide more than a day's standby during short power interruptions. Alternately, an external source such as the K10-59992 can sustain the clock for up to one year.

59313A Analog-to-Digital Converter

This medium-speed 4-channel unit can accept a full scale input of ± 10 V dc on each channel, individually selectable in four ranges. It also has a program-controlled reverse channel for driving small signal lamps, relays, or TTL circuits. An HP-IB controller can command this unit to perform a single conversion, or initiate a series of internally-paced conversions at one of six selectable rates (up to 200/s on one channel; up to 50/s on each of four channels). Sampling can also be initiated externally by TTL transition or contact closure to ground.

59501A Power Supply Programmer (Isolated DAC)

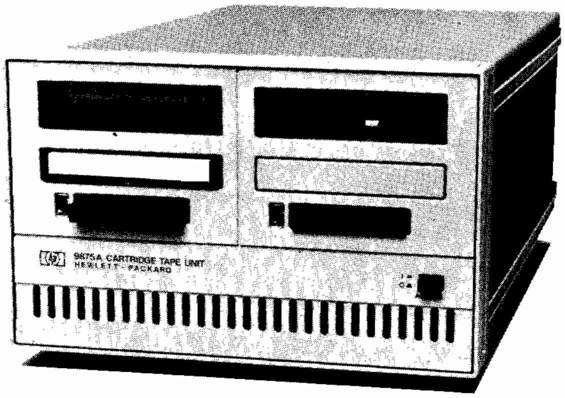
This single-channel digital-to-analog converter can control a wide range of power supplies (output voltage, or current), as well as other analog programmable devices. It may also be used as a low level signal source, depending on the speed of the controller. It has two output ranges (0−1 and 0−10V dc in unipolar mode; −1 to +1 and −10 to +10 V dc in bipolar mode), as well as photo-isolators which electrically separate HP-IB control and data lines from power supply circuitry by up to 600 V dc. (Additional details on page 232.)

9875A Cartridge Tape Unit

Provides a standard for data interchange among HP Series 9800 Desktop Computers via the Hewlett-Packard Interface Bus and also provides remote data acquisition capabilities. Any desktop computer in the series can store data on the 9875 tape unit, which can then read the data into any other desktop computer in the series. The tape unit stores data in HP's Standard Interchange Format.

An internal microprocessor enables the 9875 to become a stand-alone data logger in a simple HP-IB system. In the LISTEN-only mode the 9875 will automatically record data on the bus from another HP-IB device without a controller. When it's in the TALK-only mode, the 9875 will automatically output directly to another HP-IB device without a controller. Using a built-in programmable time interval (1 second to 18 hours) allows automatic delays between successive inputs or outputs.

The 9875 is rack mountable and is available as either a single or double tape drive unit. Each cartridge has 225k byte capacity.

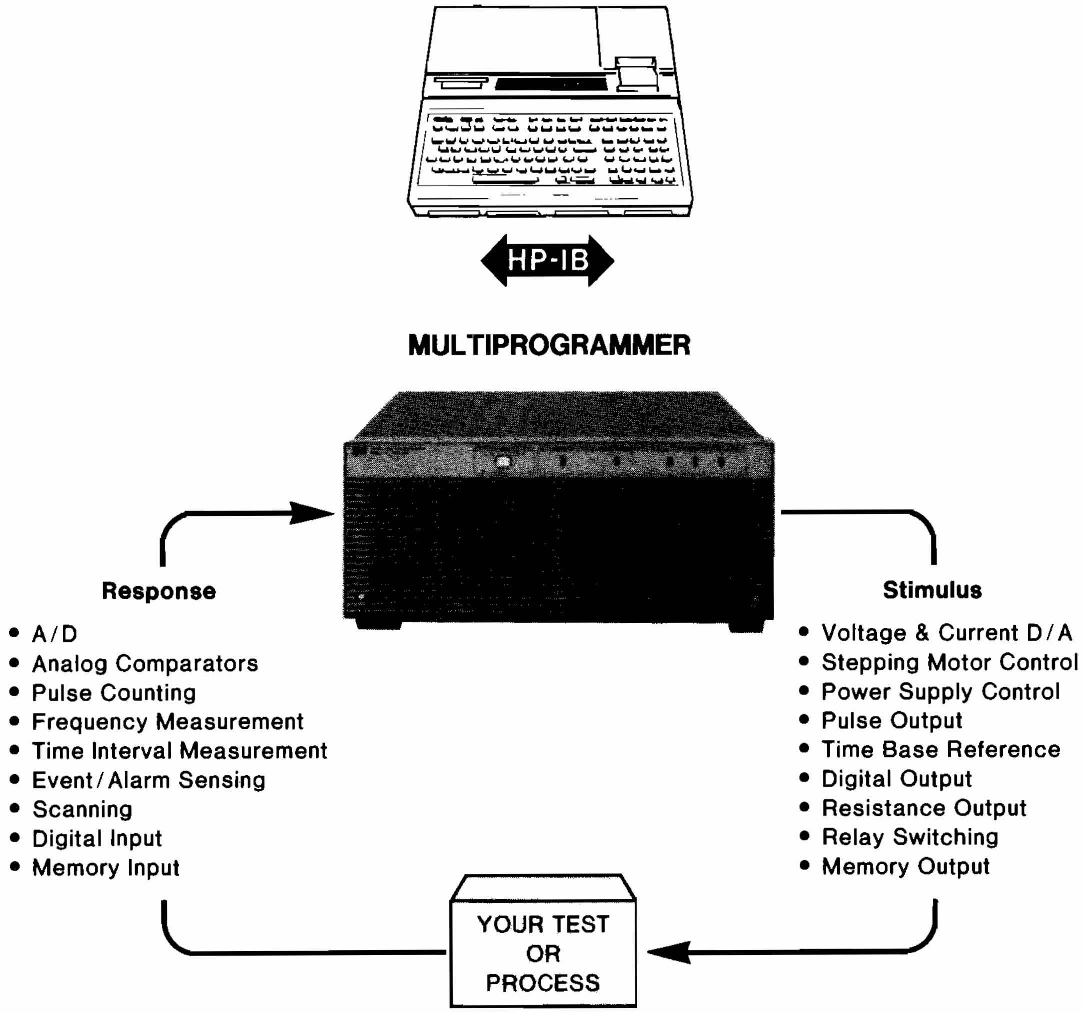


9875A Cartridge Tape Unit

Model	Description	Dimensions—max. height[1] x width x depth mm (inches)	Net Weight kg (lb)	Shipping Weight kg (lb)	Price
59301A	ASCII-to-parallel Converter	101.6 x 212.9 x 294.6 (4 x 8.38 x 11.6)	1.70 (3.78)	2.32 (5.16)	$ 575
59303A	Digital-to-analog Converter	101.6 x 105.9 x 294.6 (4 x 4.17 x 11.6)	2.61 (5.80)	3.17 (7.04)	$ 950
59306A	Relay Actuator	101.6 x 212.9 x 294.6 (4 x 8.38 x 11.6)	2.64 (5.87)	3.23 (7.18)	$ 700
59307A	VHF Switch	101.6 x 212.9 x 294.6 (4 x 8.38 x 11.6)	2.64 (5.87)	3.23 (7.18)	$ 750
59308A	Timing Generator	101.6 x 212.9 x 294.6 (4 x 8.38 x 11.6)	2.10 (4.67)	3.83 (8.51)	$1150
59309A	HP-IB Digital Clock	101.6 x 105.9 x 294.6 (4 x 4.17 x 11.6)	1.70 (3.78)	2.84 (6.31)	$1025
59313A	Analog-to-digital Converter	101.6 x 212.9 x 345.4 (4 x 8.38 x 13.6)	5.45 (12.0)	6.36 (14.0)	$1500
59401A	Bus System Analyzer	145.5 x 205.1 x 495.3 (5.73 x 8.08 x 19.5)	5.64 (12.44)	9.1 (20)	$2800
59403A	HP-IB/Common Carrier Interface	101.6 x 212.9 x 430.0 (4 x 8.38 x 16.9)	4.50 (10.0)	6.10 (13.5)	$1575
59501A	Power Supply Programmer	101.6 x 212.9 x 194.6 (4 x 8.38 x 11.6)	2.61 (5.80)	3.17 (7.04)	$ 550

[1] Height above includes feet, with feet removed height is 88.1 mm (3.47").

With a Multiprogrammer Your HP Desktop or Minicomputer Becomes a Reliable Easy-to-use Automatic Test or Control System



Benefit from the Multiprogrammer Functional Card System

You can quickly design and implement a control system using the HP-IB and one of the HP Multiprogrammers. Choose from the wide selection of functional plug-in cards and assemble them into a Multiprogrammer mainframe to economically interface your analog and digital input/output signals. The Multiprogrammer provides the interface between your HP-IB controller and the physical world. Thousands of Multiprogrammers are in everyday use as the nucleus of user defined and assembled systems for production testing and control, data acquisition, process monitoring, laboratory experiment control, life testing, quality control, and component evaluation.

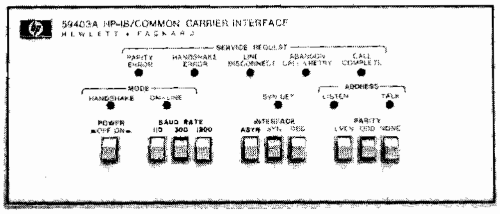
Start building your system with one of the HP Multiprogrammers combined via the HP-IB with a computing controller. To help you, HP offers a variety of proven design aids. These include the *Multiprogrammer Technical Brochures* complete with capabilities, typical system layouts, specifications, and more; . . . A *User's Guide* that gives you sample programs, test routines, and I/O interface data for all 38 Multiprogrammer plug-in cards. . . There is also a *Utility Cartridge* with a recorded program ready to use in the HP 9825A, 9835A and 9845A computing controllers, to aid in writing your own application . . . and a *System Throughput Analysis* that allows you to accurately determine the measurement and control speed you can expect before you build your system.

Refer to pages 658 and 662 for more details on the HP 6940B and 6942A Multiprogrammers and how they are used with the HP-IB.

HEWLETT-PACKARD INTERFACE BUS

Versatile interconnect system for instruments and controllers

HP-IB Over Longer Distances



59403A

The distance between HP-IB devices may be extended by up to 1000 metres, using two 59403A's; even further with modems.

The total transmission path length for the HP-IB is specified as 20 metres. To extend this, HP has developed these techniques:

Product	59403A Common Carrier Interface	37201A HP-IB Extender	12050A Fiber Optic Link
Application	Gen Purpose Inter-Intra Facility	Gen Purpose Inter/Intra Facility	Fast Intra Facility
Transmission medium	Dual Twisted Pair or Modem Link	Dual Twisted Pair or Modem Link	Dual Fiber Optic Cable
Operating range	Twisted Pair: 1000 metres Modem Link: Unlimited	Twisted Pair 1000 metres Modem Link: Unlimited	100 metres
Modem data rates	Asynchronous: 110/300/1200 bits/s	Asynchronous: 150/300/600/1200 bits/s Synchronous: Up to 19,200 bits/s	—
Hardwired speed	1760 Bytes/s	775 Bytes/s	20,000 Bytes/s
Error Checking	Parity Only No Retransmission	Block Check With Auto Retransmission	Checksum Byte with Auto Retransmission
Electrical Noise Isolation	—	Balanced Coupling on Hardwired Links	Optical
Programming Transparency	No	Yes, except Parallel Poll and Pass Control	Yes, except Parallel Poll and Pass Control

59403A HP-IB/Common Carrier Interface

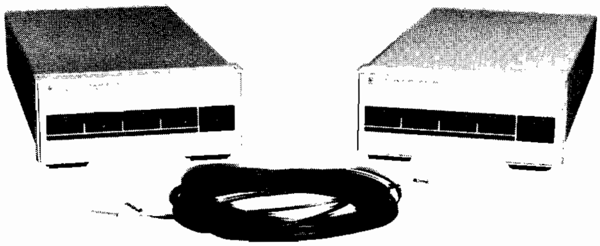
Provides a way to extend the separation of component parts in an HP-IB system by more than the 20 metre maximum transmission path length specified in various interface standards, and it is especially useful for production or remote site applications. Distances up to 1000 metres are possible by using two 59403A modules (one at each location) interconnected by a dedicated and shielded two-twisted-pair cable. And even longer distances can be achieved by using a telephone line (with appropriate modems) instead of the dedicated cable.

Each 59403A module converts HP-IB data and control lines to a serial bit stream of digital information for transmission over the dedicated or telephone lines, and vice versa in the reverse direction. In both cases, operation is full duplex, so that (for example) one HP-IB device at a remote location can request service from the controller at the same time the controller is sending data to another HP-IB device at the remote location.

The recommended dedicated cable is available from HP as Part Number 8120-1197 (Belden type 8723). The 59403A is designed to operate with 110, 300, and 1200 baud asynchronous or synchronous full duplex modems which are EIA RS 232C or CCITT V.24 compatible. In the U.S., Bell 103A modems with "soft carrier turn-off" are recommended for the direct dial (DDD) network. (Check your local telephone authorities regarding data communication regulations.)

59403A HP-IB/Common Carrier Interface **$1579**

12050A 12050A



39200 series Fiber Optic Cable

HP 12050A Fiber Optic HP-IB Link With 39200 Series Cable

- Extends HP-IB Length Up to 100 Metres Via Fiber Optic Cable
- 20 KBytes/s Data Rate
- Excellent Electromagnetic Noise Immunity
- Electrical Isolation Between Distant Sites
- Built-in Self Test and Error Correction

A single point-to-point Fiber Optic HP-IB Link consists of two HP 12050A Fiber Optic HP-IB Link units, one at the local controller site and the other at the remote instrumentation site. The 12050A units are connected using a single length of 39200 Series Fiber Optic Duplex Cable or two Simplex Cables. Data transfer rate is up to 20 KBytes/s regardless of cable length. If a remote device requests service, the service request (SRQ) will be asserted at the local end of the Link typically within 100 μs of its occurrence. Thus for many HP-IB applications, no system performance degradation will be observed when extending the bus length with the Fiber Optic HP-IB link. HP-IB devices communicate programmatically via the 12050A units just as they would in local operation. Since information is transmitted using light pulses rather than electrical signals, it is impossible for large electromagnetic fields to interfere with data being sent over the Fiber Optic Cable.

Specifications

HP 12050A Fiber Optic HP-IB Link

Power Requirements: 86 to 127 V ac; 172 to 254 V ac. 48 to 66 Hz. 15 W.

Operating Temperature/Humidity: 0 to 55°C. 10 to 95% RH; non-condensing at 40°C

Size: 9 H x 21 W x 44 cm D (3.5″ x 8.4″ x 17.4″)

Weight: 2.75 kg (6 lb. 1 oz.)

39200 Series Fiber Optic Cables

Operating temperature: 0 to 70°C.

Storage temperature: − 40 to 85°C.

Relative Humidity: 95% at 70°C.

Max. tensile force on Cable: 60 kg (132 lbs.).

Max. tensile force on Connector/Cable: 5 kg (11 lbs.).

Min bend radius: 7 mm (0.3 in.)

Flexing: 50000 cycles (180° bending at minimum bend radius).

Crush load: 20 kg (44 lbs.)

Ordering Information

39200 Series Fiber Optic Cables

Length	Simplex (2 req'd/system)	Duplex (1 req'd/system)
10 m	39201A	39201B
25 m	39202A	29302B
50 m	39203A	39203B
75 m	39204A	39204B
100 m	39205A	39205B

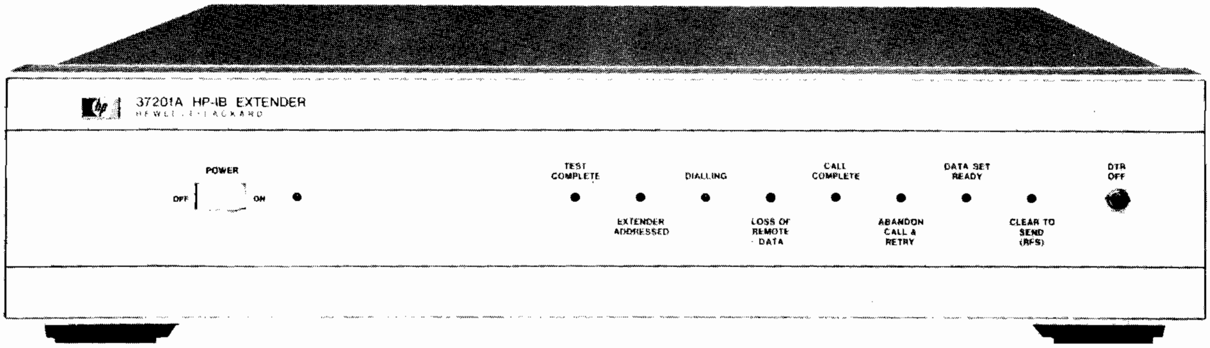
12050A Fiber Optic HP-IB Link **$1950 ea**
(Two required per system)

HEWLETT-PACKARD INTERFACE BUS

Versatile interconnect system for instruments and controllers

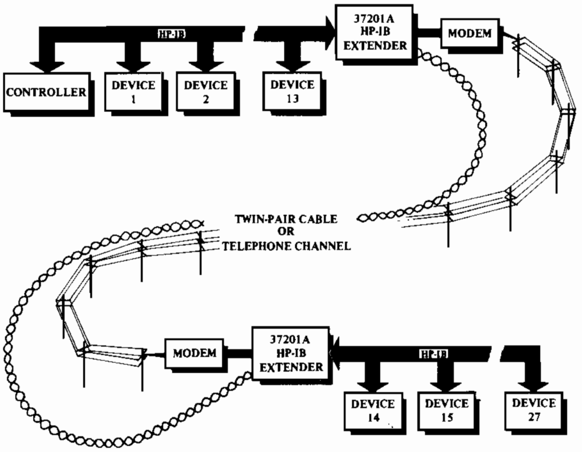
Model 37201A

- Transparent extension of HP-IB systems
- Operation over twin-pair cable or modems
- Automatic error detection and correction
- High immunity to electrical interference
- Multi-point (multi-drop) capability
- Auto-dialler interface



HP-IB

The 37201A HP-IB Extender overcomes the limited range available with direct HP-IB cable interconnections. Each 37201A converts parallel data from the interface bus into a serial bit stream, suitable for transmission to a remote site, and reconverts incoming serial data to bit-parallel HP-IB format. An HP-IB system can therefore be split into two or more discrete parts separated by HP-IB Extenders and a serial data link. A range of 1000 metres is obtainable if twin-pair cable is used for the transmission path, and virtually unlimited range is available if a modem link is used. Communication between Extenders is full duplex, allowing information to flow in both directions simultaneously.



Point-to-point connection using twin twisted pair cable or full duplex modem link.

A pair of HP-IB Extenders provides a transparent interface between local and remote HP-IB devices. Program control of the 37201A is seldom necessary. Consequently, HP-IB Extenders can be added to an HP-IB system usually without any modification of software and without writing special routines to control the Extenders.

Integrity of HP-IB data and control signals is assured by an automatic error-checking protocol, which retransmits any data corrupted in transmission.

Twin-Pair Cable Operation

Twin twisted-pair cable provides a simple inexpensive transmission medium for distances up to 1000 metres. The serial data rate is nominally 20 kbit/s. Suitable cable is available as an accessory (HP Part Number 8120–1187). Transformer coupling within the 37201A gives a high degree of immunity from the effects of common mode signals. This, combined with the automatic error correction capability, makes the 37201A suitable for use in an electrically hostile environment.

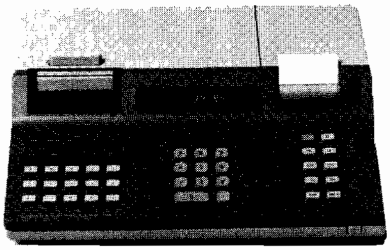
Modem Link Operation

The 37201A is designed to operate with a wide range of synchronous and asynchronous modems over private lines, leased lines, or the public switched (dial-up) telephone network. The data interface is compatible with EIA RS-232C and CCITT V.24 and V.28 standards. Asynchronous data rates provided are: 150, 300, 600, and 1200 bit/s. For synchronous modems, operation at any bit rate up to 19.2 kbit/s is possible. Besides operating in point-to-point mode, the 37201A can be used with modems in a multi-point (multi-drop) leased line configuration involving up to 31 remote sites. When operating over the public switched telephone network, connections may be dialled manually. Alternatively, an external auto-dialler may be used to make connections under program control. The 37201A has an RS-366/V.25 interface to permit operation with an auto-dialler.

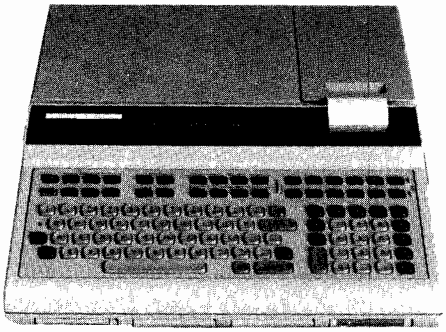
The error checking/correcting communications protocol used in the 37201A protects against errors introduced by poor quality data circuits. It even provides immunity to major interruptions in the data link, such as dropouts, line breaks and modem sync loss, and recovers automatically without loss of data.

The 37201A is in general compliance with each of the following standards and supports their major capabilities:
- IEEE Standard 488-1978
- ANSI Standard MC1.1
- IEC Standard 625-1

37201A HP-IB Extender	**$1840**



HP 9815 Desktop Computer (HP 98135A Interface)



HP 9825 Desktop Computer (HP 98034A Interface)

System 35 Desktop Computer (HP 98034A Interface)

A separate controller is not required for simple HP-IB configurations (e.g. data logging). However, the full flexibility and potential of the Hewlett-Packard Interface Bus are more obvious when used with HP Controllers.

Role of a Controller

In addition to managing the flow of information over the bus, the controller in an operating measurement system actively participates by scheduling measurement tasks, by setting up individual devices so they can perform the tasks, by monitoring the progress of the measurement as it proceeds, and by interpreting the results of the measurement.

HP controllers serve another important function by providing access to a large number of display, input/output and data storage peripherals. These include plotters, line printers, floppy discs, tape cassettes, etc. Additionally, HP controllers can perform the job of interfacing with other instrument subsystems or computer systems using serial communication links—thereby gaining access to common data bases, sharing results, etc.

Finally, a controller can provide the tools for program development. These will normally include an editor that can be used in generating source programs, debug aids that can be used in analyzing and modifying program flow, and a means of storing and recalling programs and/or results.

Wide choice of HP Controllers

Hewlett-Packard has a continuum of HP-IB (IEEE 488) controllers from which to select. If your interfaced-system application is of the "lab bench" variety (as in engineering design or metrology), you may prefer to use one of the desktop keyboard units such as the 9815A, 9825, System 35 or System 45. On the other hand, if your application calls for complex or high volume production testing at multiple locations, simultaneously, and in several programming languages, your choice will probably be one of the solutions offered by the HP 1000 Computer.

HP-IB interfaces for each controller are described below. For more comprehensive details on the controllers please consult pages 615–617.

98135A HP-IB Interface for 9815

HP's most economical controller using HP-IB is the 9815 desktop unit, for handling the less complex tasks associated with small systems. If you are familiar with HP's hand-held personal calculators, you'll feel at home with the 9815's Reverse Polish Notation (RPN) language. The keyboard has a 10-key numeric pad, 15 special function keys, program language and control keys, editing keys, and 28 scientific function keys. The 9815 has a 16-character numeric display, a thermal printer, and a high-speed bidirectional magnetic tape data cartridge system.

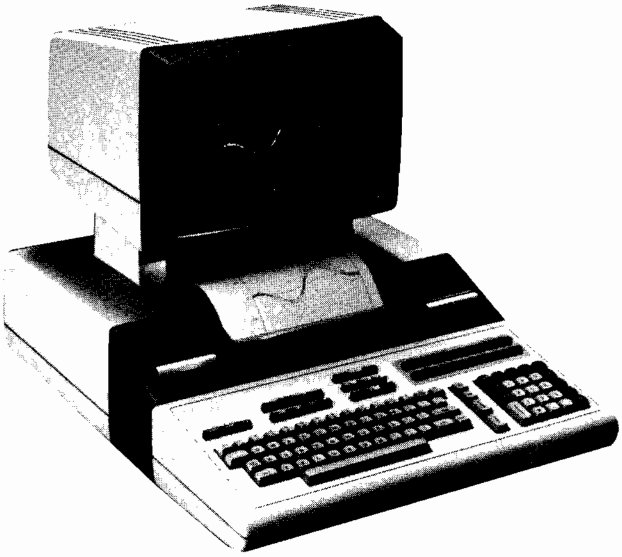
For HP-IB applications, the 9815 can accept one *HP 98135A Interface,* which allows the 9815 to communicate with up to 14 HP-IB instruments or peripheral devices. If your application requires an interrupt capability, please see other HP controllers, since interrupt is not available with the 9815/98135A.

98034A HP-IB Interface for 9825 or System 35 or 45

The 9825 Desktop Controller is an extremely flexible performer. It uses HPL, a high level, formula-oriented programming language which offers power and efficiency for handling equations, data manipulation, and input/output operations. HPL provides for subroutine nesting and flags, and allows 26 simple variables and 26 multidimensional array variables, limited only by the size of the 9825 memory. Plug-in option ROMs provide added power and flexibility for instrumentation control and data collection.

Significant capabilities of the 9825 include two-level priority interrupt (for controlling several instruments or peripherals requiring attention at unpredictable rates or times), live keyboard, direct memory access, multi-dimensional arrays, automatic memory record and load, and an extended range of internal computation. The 9825 offers up to 32 K bytes of memory. The computer includes a built-in 32-character alphanumeric display, a 16-character printer (upper/lower case), and a high-performance data cartridge system. Three I/O slots provide plug-in capability for standard desktop controller interfaces.

The Series 9800 System 35 (Models 9835A and B) is a powerful, integrated desktop computer ideal for many scientific and engineering applications involving computation, data acquisition or both. It offers large memory (64 K to 256 K bytes), built-in tape cartridge drive (217 K bytes) optional thermal printer (16-character) and an impressive range of interfacing capabilities including buffered I/O, Direct Memory Access (DMA), fast read/write, 15 levels of priority



System 45 Desktop Computer (HP 98034A interface)

HP 1000 Computer System (HP 59310B interface)

interrupt and built-in I/O drivers. System 35A has a 12-inch CRT (24 lines x 80 characters), and System 35B has a lower-cost 32-character single line display. Both can be programmed in HP's powerful, enhanced BASIC and in assembly language. Assembly-level programming can offer speed increases of two to 100 times to experienced programmers in specialized applications.

System 35's enhanced BASIC has many of the powerful features of FORTRAN while remaining easy to learn and use. ANSI BASIC programs as well as HP enhanced BASIC programs written for System 45 will run on System 35. With unified mass storage commands and unified graphics commands, the same programs work regardless of which mass storage device or plotter is used.

The Series 9800 System 45 is an integrated desktop computer for such applications as mathematical modeling, design analysis, production test control, text processing and linear programming. It provides fifteen levels of programmable priority interrupt and it includes a CRT display, an optional 80-character thermal line printer, enhanced BASIC language, and a unified mass storage system with two tape cartridge drives.

In the alpha mode, the CRT lists programs for viewing and editing, or displays data, keyboard inputs, user prompts and system messages. In the graphics mode, the CRT displays plots within a 560 x 455 dot matrix and allows dot-for-dot duplication of the graphic data in hard-copy form using the optional high-speed thermal printer.

System 45's language uses the same set of commands to address any selected storage medium, such as the HP 9885 Flexible Disc Drive, the HP 7900 Series large fixed disc drives, and the built-in 217 K byte tape cartridges.

The *HP 98034A Interface* is required for operating the 9825, System 35 or System 45 in HP-IB applications. A 9825 equipped with a General I/O ROM can handle fundamental HP-IB input/output operations. With an Extended I/O ROM, the 9825 is capable of complete HP-IB control. All these operations are available on the 9835A/B with just the General I/O ROM. Up to three interfaces can be plugged directly into the System 35's I/O slots and as many as 14 interfaces (with up to 14 devices on each) can be connected to System 35 using 9878A I/O expanders. System 45 has complete HP-IB capability with the Opt. 312 I/O ROM. Up to four interfaces can be plugged directly into System 45's I/O slots and as many as 12 interfaces (with up to 14 devices on each) can be connected using 9878A I/O expanders.

59310B HP-IB Interface for HP 1000

The HP 1000 computer system is especially well suited for broad measurement and data management requirements such as those found in quality assurance, production testing, etc. This is because the HP 1000 (combining an E-series or F- series computer and Real Time Executive Software) is capable of concurrently controlling multiple clusters of HP-IB test and measuring equipment which may be organized into separate physical or functional groupings, each of which may have up to 14 HP-IB devices per cluster. The HP 1000 also: (1) makes it possible to develop new programs while existing programs are actively controlling and communicating with the bus-interfaced devices; (2) can be programmed in HP Real Time BASIC, FORTRAN, and HP Assembly language; and (3) can be linked to distributed computer networks to achieve centralized test record maintenance, yield analysis, and work order scheduling and tracing.

Each separate bus cluster (of up to 14 HP-IB devices) connected to the HP 1000 requires one *59310B Interface*. The 59310B is supported by a driver, utility software and a manual supporting operation in HP's memory-based RTE-M and disc RTE-II and RTE-IV Real Time Executive systems. A diagnostic routine for quickly confirming correct operation is included with the interface, and each interface has a 4-metre cable terminated in an HP-IB connector with metric fasteners. Compatibilities between various HP computer systems, computers, and operating systems are indicated below. The E-series and F-series Computers include the HP 2170A, 2171A, 2172A, 2174A/B, 2175A/B, 2176A/B, and 2177A/B. Note that the 59310B interface may also be used with HP 2100A/S computers.

	HP 1000	HP 2105A	HP 2176A/B 2177A/B	HP 2100A/S
RTE-M:	Yes	Yes	Yes	No
RTE-II:	Yes	No	Yes	Yes
RTE-IV:	Yes	No	Yes	No

HP-IB Interface Ordering Information

	Price
59310B: Interface, RTE-II/IV for HP 1000	$600
98034A: Interface for 9825A, 9835A/B or System 45	$400
98135A: Interface for 9815A	$600

INTERFACE STANDARD 488 IN ACTION: CONCEPTS AND CAPABILITIES

Donald C. Loughry
Hewlett-Packard Company
1819 Page Mill Road, Palo Alto, CA 94304

SUMMARY

Instrumentation systems have undergone many changes over the last decade. Instruments today are highly programmable whereas earlier products were much less sophisticated. Over the last four years the development of an instrumentation interface standard has moved from the embryonic stage, recognition of the need for a stardard, to an approved and published document, IEEE Standard 488-1975 "Digital Interface for Programmable Instrumentation". Participation in this effort has been supported by U.S. instrumentation manufacturers and users. These national developments have taken place in parallel with similar standardization efforts at the international level within the IEC. Industry representatives in Germany were instrumental in initiating the international project. The resultant interface definition utilizes a byte-serial bit-parallel bus structure as the basic communication link among a group of instruments. Applicable to calculator and computer based systems the interface has been incorporated in excess of eighty different products being marketed today. This paper describes some of the major objectives and needs motivating the development of the interface concepts, the overall technical construct of the system, the capabilities available to design engineers, and the current status of the implementation of IEEE Standard 488.

EVOLUTIONARY EVENTS

The traditional approach to interface design in the field of bench-top instrumentation has been to provide each device (instrument) with a number of specialized control, data, and status signal lines. In this way the interface design itself was straight foward... relatively simple and direct control of an instrument's programmable functions was achieved. Also achieved, though unwanted, was a wide variety of interface solutions, almost as many different interface techniques as design engineers. The net result was a dedicated interface structure for each device or instrument integrated into a system. This situation led to many interface adapters designed to accommodate the wide variety of codes, formats, signal levels, logic conventions, and timing protocols, to name a few factors. Attempts to define a "universal interface" from the computer or controller perspective met with limited success. The bubble burst when more sophisticated and highly programmable instruments demanded in excess of fifty, sometimes one hundred, signal lines. The interface cost and complexity to achieve the necessary performance levels in complete instrumentation systems was out of line. The effort expended in accommodating the wide variety of hardware and implementing the necessary software was enormous. Thus, at the beginning of this decade the general interface problems became severe enough to warrant an all out effort toward a common solution.

Members of a German committee representing instrumentation manufacturers were instrumental in defining the needs within Europe and proposed that TC66 of the International Electrotechnical Commission (IEC) establish a specific project to define an instrumentation interface standard. Shortly thereafter, representatives of U.S. instrument manufacturers and users met to determine a reasonable set of objectives and consider what techniques or defacto standards were worthy of further consideration and elaboration. One such proposal, submitted by Hewlett-Packard Company, became the basis for further discussion within the U.S. Advisory Committee and was subsequently evaluated by the IEC working group and considered to be the basis of further discussion. Thus in late 1973 and early 1974 work began in earnest on the definition of what is now IEEE Standard 488 and the equivalent IEC draft document currently under ballot.

MANAGEABLE OBJECTIVES

An overview of instrumentation system components and requirements discloses a wide spectrum of needs. Some large systems contain fifty or more instruments while others contain fewer than five. Most systems, assembled by either manufacturers or users, remain intact for long periods of time whereas other system configurations are changed on a daily basis... today's trend in a cost conscious world. While the majority of instruments are located close to the central control unit there are definite needs for remote terminals and displays. Data rates among system components presents still another critical variable. Analog-to-digital converters and magnetic storage devices may well operate in the megabyte per second range. Counters, voltmeters, signal sources and printers, however, generally acquire and generate data at rates well below ten thousand bytes per second. Can one interface solution satisfy all these requirements?

The above situation underscores the necessity to focus on the most frequent needs, the must objectives, to the exclusion of those less frequent and costly demands. Since an interface system may be viewed as a communication link it is helpful to think in terms of communication parameters. How many devices need to be communicated among, at what distances and data rates, with what types of message requirements? In these terms a manageable set of objectives can be formulated around the most frequent interface system needs as applied to instrumentation systems:

- Data rates...under one megabyte per second
- Distances...less than twenty meters
- Number of devices...less than twenty
- Message lengths...ten to twenty characters typical

These parameters form the basis of IEEE 488's "scope". Given the scope, what specific objectives are worthy of attack? The questions of cost, flexibility, and compatibility standout as being significant in today's instrumentation systems. Stated more specifically it is the "object" of IEEE 488 to:

- Define a general purpose system for use in limited distance applications
- Specify the device-independent mechanical, electrical, and functional interface requirements
- Enable the interconnection of independently manufactured devices into a single operational system
- Permit apparatus with a wide range of capability, from the simple to the complex, to be interconnected simultaneously
- Permit direct communication between devices without requiring all messages to be routed through a control unit
- Permit asynchronous communication over a wide range of data rates
- Define a system that, of itself, may be relatively low cost
- Define a system that is easy to use

These considerations led to the interface system concepts and definitions set forth in IEEE 488. Stated simply, it seemed best to concentrate on a specific portion of the world's interface problems rather than focus on too broad a set of objectives in an ineffective way. In elaborating the standard several aspects of the definition were altered to make it more useful without compromising the original objectives.

INTERFACE SYSTEM CONCEPTS

System Elements

An interface system is a means to an end, a servant to enable efficient communications among the components of an instrumentation system. The overall goal of an instrumentation system, of course, is to test or monitor devices or processes in an efficient manner. In examining further the concepts of an interface system, it is helpful to group the major interface system elements (see Figure 1) into four broad categories; mechanical, electrical, functional, and operational. In general the extent to which these interface system elements are defined in a device independent manner determines the degree to which both manufacturers and users can affirm a common interface definition.

MECHANICAL	ELECTRICAL	FUNCTIONAL	OPERATIONAL
SCOPE OF INTERFACE SYSTEM DEFINITION DEVICE (SYSTEM) INDEPENDENT SPECIFICATIONS			DEVICE (SYSTEM) DEPENDENT SPECIFICATIONS

Fig. 1. Interface System Elements

The mechanical portion of an interface defines the physical configuration of connectors, cables, and indirectly the network topology (star or bus structure). The voltage and current limits at the connector node for each of the interface signal lines defines the electrical portion of an interface. The functional portion of an interface defines the precise use of each of the signal lines contained within the interface, the protocol used to transfer messages across the interface, timing relationships between signal lines, the repertoire of interface messages that may be carried between devices, and the basic communication sequences the interface is capable of supporting. It is this aspect of an interface system that determines, to a large extent, the ease with which independently designed devices can be interconnected in an instrumentation system. The operational portion of an interface deals with the way in which devices use the interface via application software, diagnostic routines, and unique program codes for specific devices. Operational elements of an interface system tend to be device and system dependent.

IEEE Standard 488 defines the mechanical, electrical, and functional specifications in relatively device and system independent terms. In this way the interface definition applies to a broad range of devices with few restrictions at the same time the additional engineering required to provide a complete operational system is reduced to a minimum.

Concepts and Capabilities

Given the overall objectives for an interface system, what are some of the specific capabilities needed and how are they to be met in instrumentation system environments? Table A identifies some of the functional needs required of the interface system and indicates how the results and capabilities are provided for. This list is by no means exhaustive. For example, the need to simplify the interconnection of a system is met, in part, by the decision to distribute equally the electrical load among all interconnected devices. This, in turn, eliminates both the need for special termination resistors at the end of a transmission line and the possible limitations of restricting the interconnection scheme to a linear bus. Similarly, the need to permit either a star or linear bus structure, in the physical sense, is satisfied, in part, with a piggyback connector cable assembly. Additional concepts and capabilities will be described in the interface function discussion.

BASIC NEED	CONCEPT	CAPABILITY PROVIDED BY
Unambiguous definition	Logical definition independent of implementation scheme	State diagram description of interface functions
Direct access to multiple asynchronous mesages	Dedicated signal lines	IFC, ATN, SRQ, REN, EOI signal lines
Cost/performance flexibility	Optionality	Ten interface functions with allowable subsets
Multiple listeners independent of position or response rate	Three-wire handshake	DAV, NRFD, NDAC signal lines
Minimal hardware cost	Bus structure with minimal signal line count	Bi-directional bus for address, command, data, status messages
Standard method for accessing devices	Common code, easily generated and used	Address and universal command structure based on ASCII code
Slow speed status reporting	Device initiated service request	Common SRQ signal line, Serial Pole Mode with status byte reporting
High speed status reporting	Controller initiated status request	Parallel Pole Mode, one status bit for each of eight devices
Accommodation of other interface techniques	Hierarchial partitions	Terminal unit dedicated to interface conversion for cluster of local devices

Table A. Key Functional Needs Accommodated By IEEE 488

Bus Structure

A party-line bus structure is considered the best choice to meet the overall objectives of cost, flexibility, and compatibility. Furthermore, a bi-directional and multi-purpose data bus within the overall structure serves as the major communication path. Sixteen signal lines comprise the complete bus structure, half assigned to the data bus function and the other half assigned to specific control and status functions. Figure 2 identifies the signal line names and clusters these functions into three main groups; Data Input Output Bus, Data Byte Transfer and Control, General Bus Management.

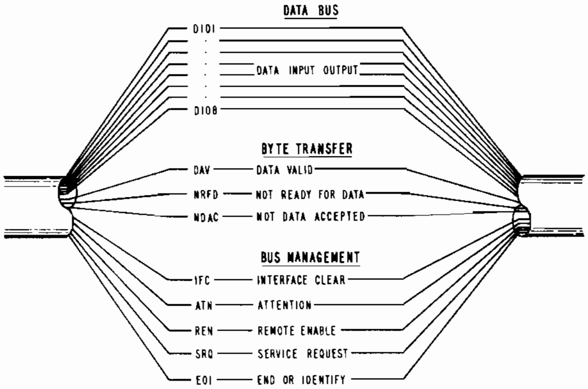


Figure 2. Party-Line Bus Structure Services Fifteen Devices

It is convenient to use terms considered appropriate for describing good communication between human beings, a talker and a listener, to define the bus structure. In the bus context the talker is that device enabled to send data over the bus and similarly listener(s) are those devices enabled to receive data over the bus. A third function, a controller, is also necessary. The controller designates which device is to talk and which is to listen. A controller may also send special types of data (i.e., addresses and commands) intended for all devices connected to the bus.

The DIO signal lines carry data, either seven or eight bits, in byte serial fashion across the interface. These signal lines carry addresses, program data, measurement data, universal commands and status bytes to and from as many as fifteen devices interconnected in a single system. Identification of the type of data present on the DIO signal lines is indicated by the ATN (attention) signal. When the ATN message (signal) is true (asserted) either addresses or universal commands are present on the data bus and all connected devices are required to monitor the DIO lines. When the ATN message is false then device-dependent data (e.g., program data) is carried between devices previously addressed to talk and listen.

Transfer of each data byte on the DIO signal lines is accomplished via a set of three signal lines; DAV (data valid), NRFD (not ready for data), and NDAC (not data accepted). These signals operate in an interlocked handshake mode. Two signal lines, NRFD and NDAC, are each connected in a logical AND (wire OR) to all devices connected to the interface. The DAV signal is sent by the talker and received by potential listeners whereas the NRFD and NDAC signals are sent by potential listeners and received by the talker. When all potential listeners have acknowledged they are "ready for data" via NRFD the talker initiates the data byte transfer by signaling data is valid on DAV. Upon receiving the DAV message the device removes its RFD message and asserts the "data accepted" message on NDAC. If only one device on the bus is an active listener the existence of the DAC message alerts the talker that the data byte has been received and may be removed from the bus. When two or more devices are "listening", the situation during the address operation, each device participates in the same handshake cycle. The major difference, however, is that each step of the handshake cycle is not completed until all devices have completed a particular step in the process. DAV is not sent until all devices are ready. A new data byte is not sent until all devices, from the fastest to the slowest, have accepted the previous data byte. Figure 3 illustrates the sequence of events and represents a composite waveform for both the NRFD and NDAC signal lines contributed to by more than one listener. This process enables devices with different data acceptance or response rates to co-exist on the same bus. Data is transferred asynchronously at the rate of the slowest listener participating in the conversation. High speed data transfer between two devices is not affected adversely by slow speed devices physically connected to the bus, when these slower devices are not addressed to participate in the conversation.

The remaining signal lines manage the bus to effect an orderly flow of messages. The IFC (interface clear) message places the interface system, portions of which are contained in all interconnected devices, in a known quiescent state. SRQ (service request) is used by a device to indicate the need for attention or service and to request an interruption of the current sequence of events. REN (remote enable) is used to select between two alternate sources of device program data. EOI (end or identify) is used to indicate the end of a multiple byte transfer sequence or, in conjunction with ATN, to execute a polling sequence. Some of these signal lines and messages will be described further in the next section.

INTERFACE FUNCTIONS FOR ALL REASONS

In general terms, a device or instrument designer is faced with the task of designing two broad categories of circuits; device functions and interface functions. The device functions encompass all of the analog and digital circuitry unique to the basic purpose of the device (e.g., voltage measurement circuitry, signal generaton circuitry). The interface functions bridge the gap between device functions and the physical interface bus. Their major purpose is to provide the communications link to the outside world and to provide an orderly flow of messages to and from the device. Of course, the specification and implementation of device functions is a matter left to the design engineer and is beyond the scope of IEEE 488. The interface functions themselves, however, are at the heart of the interface system. They provide the basic communication capability and, to a large extent, determine the degree of compatibility among independently designed devices. What then are the design choices available to the design engineer?

Design Choices

A repertoire of ten interface functions provides the essential capabilities of IEEE 488. Five interface functions (Source and Acceptor Handshake, Talk, Listen, Controller) provide primary communication capabilities and

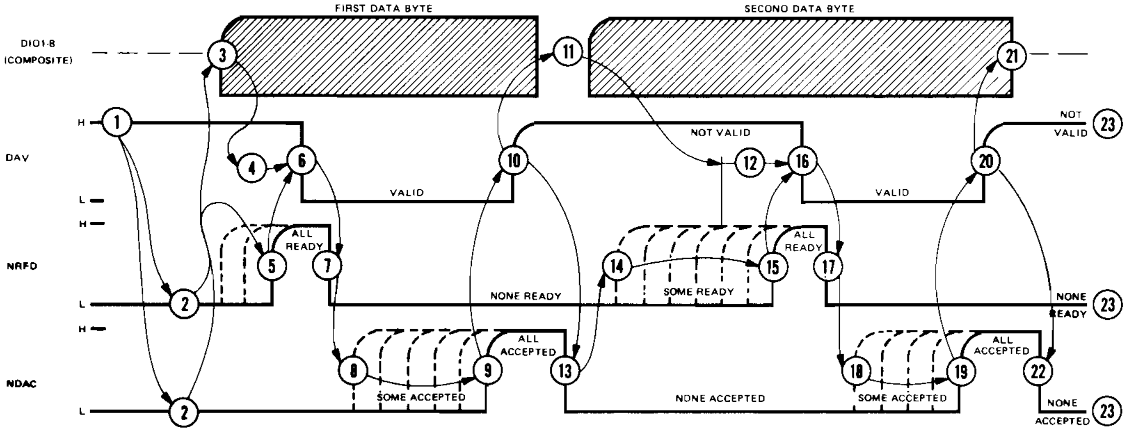


Figure 3. The Interlocked Handshake Transfers Data Bytes To One Or More Listeners...Asynchronously And Unambiguously.

five additional interface functions (Clear, Trigger, Remote/Local, Service Request, Parallel Poll) provide special purpose capabilities. The sixteen signal lines of the interface bus are carried to and from these ten interface functions in the manner shown in Figure 4, a simplified version of the actual definition. Only those functions important to a particular product's applications need be implemented.

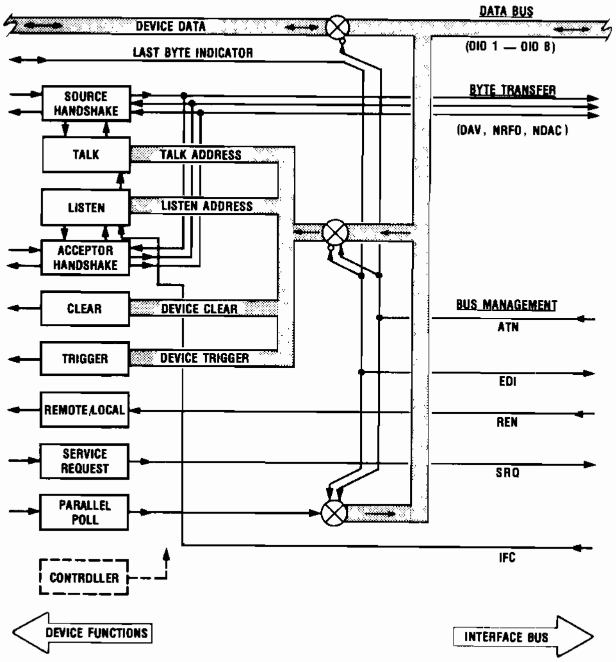


Figure 4. Ten Interface Functions Serve To Link The Bus Structure To The Device Functions.

Basic communications are established with a device via the talk or listen interface functions. Three conditions must be met to enable an active talker or listener; the ATN message must be asserted to indicate that interface messages rather than device dependent messages are to be received, a specific talk or listen address must be present on the DIO lines, and the data byte transfer process must be executed via the DAV, NRFD, and NDAC signal lines. When either the talk or listen interface function has been enabled and the ATN message is received false the active function operates in conjunction with the source or acceptor handshake function to send or receive device dependent data. Note that the data bus is used to transfer either device dependent messages (e.g., data bytes) or interface messages (e.g., MLA, my listen address) as controlled by the assertion of the ATN message. In a similar way the special purpose interface functions of device clear and device trigger are enabled via the ATN message, the acceptor handshake function and universal or addressed commands encoded on the DIO lines. Once these messages are received and decoded by the interface function a local (single line) message is transferred to the device to either "clear" it and return the device functions to a predetermined initial state or "trigger" the device to perform its basic task (e.g., "measure").

In contrast, the remote/local interface function is enabled via a single dedicated line, REN. This function enables the operator of an instrumentation system to select the program data source for a given instrument. Will a specific instrument receive its program data from front panel controls (local) or via the interface bus? Instruments connected to the system may thus be switched back and forth between remote and local control concurrently or individually (via coded messages on the DIO lines) at the system operator's discretion.

The service request interface function provides a device with the capability to alert the system controller that additional status information is available from a device. This condition may be asserted asynchronously from other activity on the interface bus. For simple devices where only one reason exists for sending the SRQ message (e.g., out of paper) no further status information is needed once a device is identified as having "requested service". If, however, a device is capable of initiating the SRQ message for any one of a number reasons then a complete status byte may be sent to identify the specific reason(s) as part of a multi-bit status message. In some systems it is important for the controller to seek specific status data from individual devices. The parallel poll interface function is used for this purpose to send a single status bit, on demand, to the controlling device. Thus, status bits from up to eight different devices may be solicited concurrently whenever the controller asserts both the ATN and EOI messages. The EOI signal line is also used as a means of indicating the last byte of data to be sent by an active talker.

The controller interface function is not necessary in most instruments though it is shown in Figure 4 for completeness. When a controller function is contained within a device (e.g., calculator, minicomputer) it drives the ATN, REN IFC, and, at times, the EOI signal lines. These four signal lines provide overall bus management.

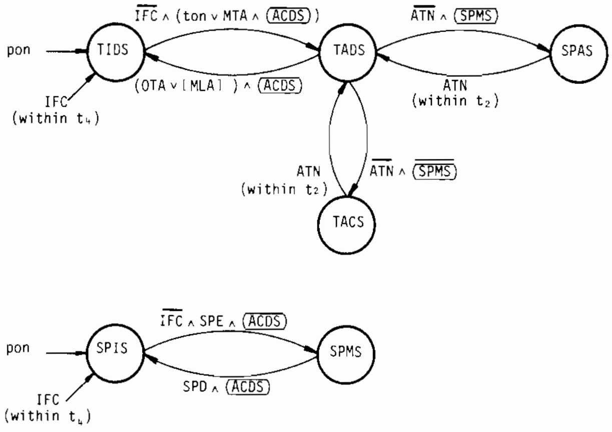
In summary, the design engineer is free to select and implement those interface functions required for the particular set of product applications envisioned. Given that the design engineer needs to implement one or more interface functions, what design concepts and interface capabilities are included in the typical interface function?

The Unambiguous State Diagram

The functional compatibility between two independently designed devices depends on the accuracy and clarity with which each of the ten interface functions is defined. Therefore, state diagram techniques, rather than timing or logic diagrams, are used to define each interface function for two critical reasons; (1) each interface function can be defined and then implemented in an unambiguous manner, (2) the resultant definition is independent of any particular type of logic circuitry or device technology.

The complete talker interface function, see Figure 5, serves to illustrate state diagram concepts. A state diagram represents the interrelationships between a set of mutually exclusive states. For the talker interface function it is possible to have two states active concurrently, one for each mutually exclusive set. As each state becomes active, it represents and defines the total message processing capability of that set of states; the messages that will cause a transition to another state, the messages that may be sent while that particular state is active. Since messages that cause a transition from one state to the next may come either from the interface bus or be generated internally (locally), it is convenient to adopt notation conventions. Upper and lower case three-letter mnemonics are used to identify remote and local messages respectively.

Initially the talker function resides in the idle state (TIDS) when the power is turned on (pon)or the IFC message is received. By definition the talker function will return to TIDS from any other active state whenever the IFC message is true. If the IFC message is false and either the talk only switch (ton) is on or the device receives its talk address (MTA) concurrent with the accept data state (ACDS) of the acceptor handshake function, then the talker function will shift to the talker addressed state (TADS). In this state the talker function has been designated to become active, however, it is not yet enabled to send data across the interface. Presumably the controller would address other devices in a similar way to become listener(s) before direct communication takes place between designated talker and listener. Once a specific set of devices has been addressed to talk and listen the ATN message sent true is no longer needed. The talker state diagram shifts from the TADS state to the talker active state (TACS) when the ATN message is received false. The device containing this function is now enabled to send data bytes across the interface according to the dictates of its own device related functions. This state of affairs continues until the controller both asserts the ATN message and sends the address of some other talker, other talk address (OTA). As soon as the controller sends the ATN true message the active talker reverts to the TADS state and when an OTA message is received the talker function shifts from the TADS state to the TIDS state. In this way the possibility of more than one talker on the interface bus at a given time is precluded.



MESSAGES

pon - power on
ton - talk only
IFC - INTERFACE CLEAR
ATN - ATTENTION
MTA - MY TALK ADDRESS
SPE - SERIAL POLL ENABLE
SPD - SERIAL POLL DISABLE
OTA - OTHER TALK ADDRESS
MLA - MY LISTEN ADDRESS

INTERFACE STATES

TIDS - TALKER IDLE STATE
TADS - TALKER ADDRESSED STATE
TACS - TALKER ACTIVE STATE
SPAS - SERIAL POLL ACTIVE STATE
SPIS - SERIAL POLL IDLE STATE
SPMS - SERIAL POLL MODE STATE
(ACDS)- ACCEPT DATA STATE (AH Function)

SYMBOLS

∧ - Logic AND
∨ - Logic OR
$\overline{\text{XYZ}}$ - Not XYZ
[ ] - Optional
(ABCS)- Active state on another diagram (state linkage)

Figure 5. Talker Function State Diagram

A device that has been designed to generate the service request, SRQ message, must also be capable of sending out an appropriate status byte when called upon to do so by the controller during a serial poll. Under these conditions, the second portion of the talker state diagram is implemented using the serial poll idle state (SPIS) and the serial poll mode state (SPMS). Under normal conditions the talker function remains in the SPIS state. Upon receipt of a universal command, serial poll enable (SPE), the function shifts to the SPMS state. When the device containing the talker function is next addressed to talk with its MTA and the ATN message is received false the talker function shifts from the TADS state to the serial poll active state (SPAS). The talker function is now capable of sending one and only one status byte to indicate that it was the requesting device and any further status bits required to identify the particular reason.

The specific set of messages that may be sent are defined for each active state. For example, the status byte (STB) may only be sent while the talker function is in the SPAS state. Data bytes (DAB) may only be sent in the TACS state. Similarly the END message may only be sent on the EOI signal line when the talker function is in the TACS state. IEEE Standard 488 defines the complete set of interface messages that may be generated by each interface function and the specific states or conditions under which these messages may be sent. The state diagram concept of defining the interface functions permits different devices to be designed by different engineers at different times for different applications and then assembled into one common system without undue concern for functional incompatibilities among the interconnected devices.

A design engineer has two sets of choices to make for most if not all of the interface functions. First, the basic decision as to whether or not to implement a given interface function. Second, a number of allowable subsets are available to further optimize the particular interface capabilities in a given product to the anticipated applications of that product. Table B indicates four variations for the states or transition expressions of the state diagram shown in Figure 5. The design engineer may choose any or all of the capabilities to be included in the talker interface function; basic talker, serial poll capability, talk only mode, and the special purpose "unaddress if MLA". The truth table contained on the left half of Table B leads the design engineer to the particular subset of states and transition expressions to be implemented in a given application. In addition this same table of allowable subsets indicates the other interface functions that must be included in a device in order to fully implement the selected capabilities of the talker function. Each of the interface functions is defined in a similar way with state diagram and two associated tables; a list of messages that may or must be sent for each state and the allowable subset table.

STANDARDIZATION STATUS

Standardization To Date

National and international related standards bodies worked cooperatively and in parallel over a three year period to define and refine at least nine draft documents and elaborate fully the present programmable instrument interface definition. The IEEE Technical Committee on Automated Instrumentation approved an IEEE draft document in the early Fall of 1974 and the IEEE Standards Board approved what was later printed as IEEE Standard 488 in December of 1974. By October 1975 the same Standard had been approved by the ANSI Committee on Programmable Test Measurement Systems and the ANSI Board of Standard Review as ANSI MC1.1-1975.

In September of 1974 IEC Technical Committee 66 approved the electrical and functional interface spedifications for a Six Month Rule document at its meeting in Bucharest, Romania. The mechanical specifications were left for further consideration. Bilingual editing of the approved draft has since been completed and the document was sent out for ballot in

Identification	Description				States Omitted	Other Requirements	Other Function Subsets Required
	Capabilities						
	Basic Talker	Serial Poll	Talk Only Mode	Unaddress If MLA			
T0	N	N	N	N	all	none	none
T1	Y	Y	Y	N	none	omit [MLA $\wedge$ $(\overline{\text{ACDS}})$]	SH1 and AH1
T2	Y	Y	N	N	none	omit [MLA $\wedge$ $(\overline{\text{ACDS}})$] ton always false	SH1 and AH1
T3	Y	N	Y	N	SPIS, SPMS, SPAS	omit [MLA $\wedge$ $(\overline{\text{ACDS}})$]	SH1 and AH1
T4	Y	N	N	N	SPIS, SPMS, SPAS	omit [MLA $\wedge$ $(\overline{\text{ACDS}})$] ton always false	SH1 and AH1
T5	Y	Y	Y	Y	none	include [MLA $\wedge$ $(\overline{\text{ACDS}})$]	SH1 and L1-L4 or LE1-LE4
T6	Y	Y	N	Y	none	include [MLA $\wedge$ $(\overline{\text{ACDS}})$] ton always false	SH1 and L1-L4 or LE1-LE4
T7	Y	N	N	Y	SPIS, SPMS, SPAS	include [MLA $\wedge$ $(\overline{\text{ACDS}})$]	SH1 and L1-L4 or LE1-LE4
T8	Y	N	N	Y	SPIS, SPMS, SPAS	include [MLA $\wedge$ $(\overline{\text{ACDS}})$] ton always false	SH1 and L1-L4 or LE1-LE4

Table B. Eight Allowable Subsets For The Talker Function Provide Optional Capability

December of 1975. The final vote should be complete by late May 1976. The mechanical specifications for the IEC document are still being considered under the Accelerated Procedure as of this writing. The remaining mechanical questions relate to potential incompatibilities with existing standards. Several European countries are proceeding to draft and circulate for approval their own national standards based on the IEC work as was done in the U.S.

Product Introductions

The interface concepts embodied in IEEE 488 clearly take into account the needs of measurement instrumentation. Many different device types are necessary in typical instrumentation systems and significant efforts were made to facilitate the use of additional device types in the design of the interface system. In fact, the standard enumerates at least five major device types to which the interface is intended to apply; measurement, stimulus, display, storage, and processor. A sixth device type, identified as a terminal unit, is that device by means of which a connection (and translation, if required) is made between IEEE 488 and some other external interface system. What products are available in the market place now that IEEE Standard 488 exists?

Any list of product types is certain to be out of date the moment it is printed. Table C identifies typical products, listed by generic name, introduced as of January 1976. This partial listing of products is intended to give a feeling for major areas of activity rather than a complete list. It is not surprising that there are more measurement devices listed than any other since this was the initial target area both in the U.S. and abroad. It is equally interesting to note that at least some products have been introduced in all six of the device categories.

A significant portion (about 40%) of the product types listed above are now manufactured by more than one company. At least the following U.S. based companies have utilized IEEE 488 with their products: Boonton Electronics Corp., Dana Labs. Inc., J. Fluke Mfg. Co. Inc., Hewlett-Packard Co., Interface Technology, Ithaco, Process Dynamics Inc., Systron-Donner Corp., Tektronix Inc., and Wavetek. In addition, Motorola Inc. has produced quad transceiver components to provide electrical compatibility with IEEE 488. Three European companies (Decca Comm. Ltd., N.V. Philips, and Rhode & Schwarz) have introduced products compatible with either IEEE 488 or the IEC draft document counterpart.

MEASUREMENT DEVICES

- Analyzer, Network
- Analyzer, Pulse
- Analyzer, Spectrum
- Analyzer, Waveform
- Bridge, Capacitance
- Clock, Digital
- Converter, D/A
- Counter, Electronic
- Counter, Frequency
- Counter, Universal
- Counter, Timer
- Filter, Programmable
- Level Measuring Set, Selective
- Meter, Power
- Multimeter, Digital
- Probe, Time Interval
- Receiver, Communications
- Scanner, Analog
- Switch, VHF

DISPLAY DEVICES

- Numeric
- Plotter, Digital
- Printer, Digital

STIMULUS DEVICES

- Generator, Function
- Generator, Signal
- Generator, Timing
- Generator, Waveform
- Generator, Word
- Synthesizer, Frequency

STORAGE DEVICES

- Disc
- Tape, Magnetic

PROCESSOR/CONTROLLER DEVICES

- Calculator, Programmable
- Minicomputer
- System, Graphic Computing

TERMINAL DEVICES

- Converter, Bus/Parallel
- Interface, Common Carrier

Table C. Typical Product Introductions With IEEE 488 Capability

The acceptance and utility of an interface standard is, in some measure, supported by the ease with which different products from different manufacturers can be configured into one operational system. This is particularly true for field installations where the opportunity to modify equipment is minimal or non-existant. Such was the case in the past six months for seminars and short courses where IEEE 488 compatible products were demonstrated. In each case, where two or more different manufacturers products were involved, the system integration went smoothly with only the addition of the normal application program. Though the number of such demonstrations is limited to date, the results indicate IEEE 488's major objectives are achievable.

Future Directions

IEEE Standard 488 specifically excluded the operational elements of an interface since they tend to be device or system dependent. System experience gained with the introduction of IEEE 488 compatible products points to both the need and feasibility of further effort in specific areas. The most fruitful possibilities center around the subject of codes and formats. Guidelines or preferred alternatives, rather than a single standard message format appears to be feasible. The net effect of such work in codes and formats would be to redefine a small portion of what is considered as operational specifications in a relatively device (system) independent way. Obviously, this would reduce some of the potential operational incompatibilities and improve the ease with which systems can be configured. Discussions are in progress along these lines at both the national and international levels.

It is presumptuous for any one individual or organization to state the extent to which IEEE 488 will be incorporated in future products. Standards are voluntary and as such stand on their own merit and solution to real world problems. The interest in and use of IEEE 488 to date suggests that it does indeed fulfill many of its objectives toward providing an easy-to-use concept that links various combinations of instruments, calculators or computers and peripheral devices to function effectively in instrumentation systems.

BIBLIOGRAPHY

Coates, Tom (Hewlett-Packard Co.) "Interfacing to the interface: practical considerations beyond the scope of IEEE standard 488". WESCON '75 Session 3.

Fisher, P. David and Welch, Stephen M. (Michigan State University) "Part two: how to build an automated system around a programmable calculator". ELECTRONICS, May 16, 1974.

Fluke, John M. Jr. (John Fluke Mfg. Co. Inc.) "System Considerations in using the IEEE digital instrument bus". WESCON '75 Session 3.

Knoblock, Daryl E. and Loughry, Donald C. (Hewlett-Packard Co.) Vissers, Chris A. (Twente Univ. of Technology) "Insight into interfacing". IEEE SPECTRUM p. 50-57, May 1975.

Knoblock, Daryl E. (Hewlett-Packard Co.) "Identifying, understanding, and selecting among the capabilities provided by IEEE standard 488". WESCON '75 Session 3.

Lee, Richard C. (Boonton Electronics Corp.) "Microprocessor implementation of a measurement instrument and its interface". WESCON '75 Session 3.

Loughry, Donald C. (Hewlett-Packard Co.) "A common digital interface for programmable instruments: the evolution of a system". HEWLETT-PACKARD JOURNAL, October 1972.

Loughry, Donald C. (Hewlett-Packard Co.) "Digital bus simplifies instrument system communication". EDN MAGZAINGE, September 1972.

Loughry, Donald C. (Hewlett-Packard Co.) "A common digital interface for programmable instruments: the evolution of a system". HEWLETT-PACKARD JOURNAL, October 1972.

Loughry, Donald C. (Hewlett-Packard Co.) "Instrument communications: a new interface system". INTERCON '73 PAPER.

Loughry, Donald C. (Hewlett-Packard Co.) "What makes a good interface?". IEEE SPECTRUM, p.52-57, November 1974.

Nelson, Gerald E. and Ricci, David W. "A practical interface system for electronic instruments". HEWLETT-PACKARD JOURNAL, October 1972

Pannach, Arndt (Hewlett-Packard Co.) "Interface-entwicklung für den IEC-bus" ELEKTRONIC HEFT 12, p.61-64, 1975.

Ricci, David W. and Nelson, Gerald E. (Hewlett-Packard Co.) "Standard instrument interface simplifies system design" ELECTRONICS, p. 95-106, November 14, 1974.

Ricci, David W. and Stone, Peter S. (Hewlett-Packard Co.) "Putting together instrumentation systems at minimum cost". HEWLETT-PACKARD JOURNAL, p. 5-11, January 1975.

Sugarman, Robert "IEEE interface beats microprocessor as most-wanted signal generator feature" ELECTRONIC ENGINEERING TIMES, February 16, 1976.

Reprinted from ELECTRO '76 Professional Program

Copyright 1976

Institute of Electrical and Electronics Engineers, Inc.

Boston, May, 1976

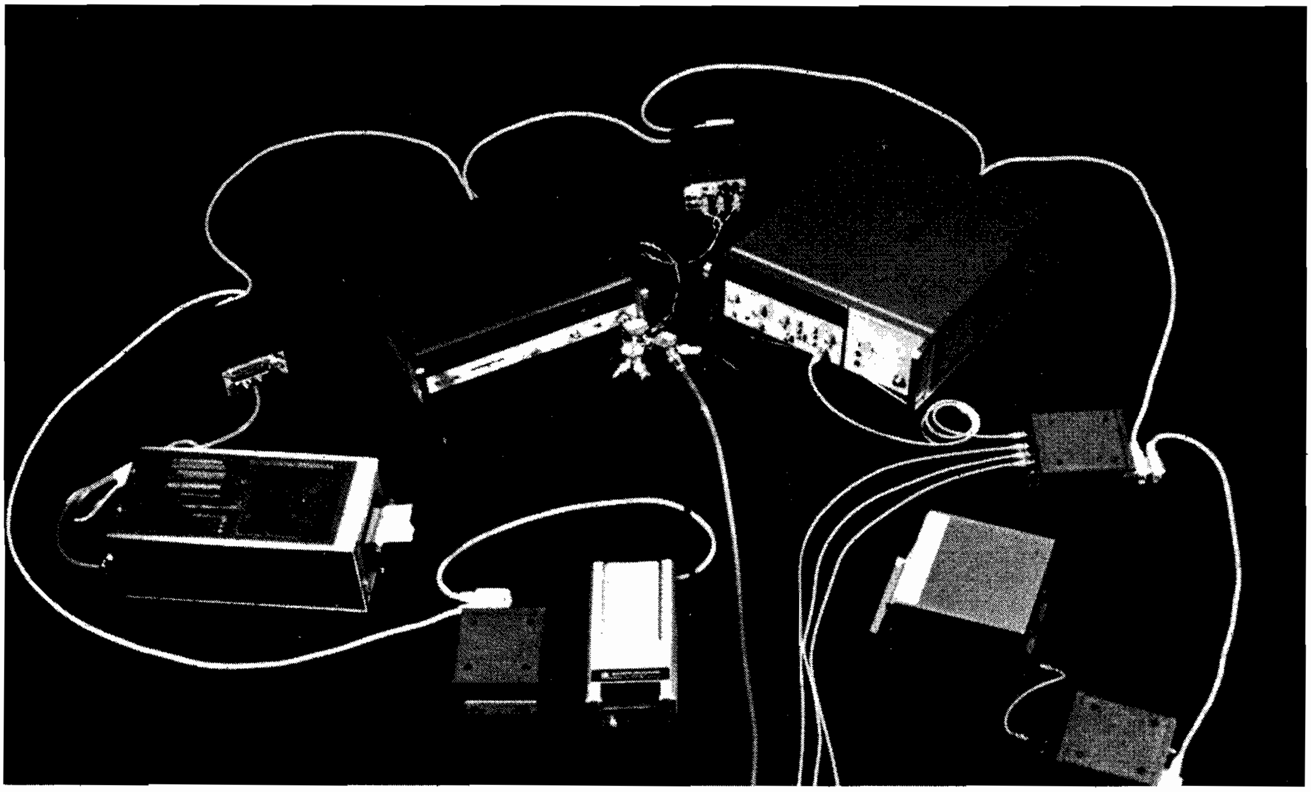
Reprinted from Electronics, November 14, 1974;

5952-2464

Standard instrument interface simplifies system design

Instruments made by any company anywhere in the world will be easy to link up into systems when all the elements conform to an international standard on interface circuitry and bus interconnection

by David W. Ricci and Gerald E. Nelson, *Hewlett-Packard Co., Santa Clara Division, Calif., and Loveland Division, Colo.*



Once the international standard for an instrument interface is agreed on, it will become quicker and more economical to construct automated instrumentation systems. Until now, sophisticated, cost-effective instruments have been readily available—but their generally incompatible inputs and outputs have forced the system designer to put a lot of effort into interface design. A standard interface is the rational solution, and for once the rational solution is being attempted.

The International Electrotechnical Commission recently met in Bucharest, Romania [*Electronics*, Sept. 19, p. 67], to discuss just such a standard. The proposal before the IEC's Technical Committee 66 was an instrument bus standard initiated by Hewlett-Packard Co., Palo Alto, Calif., and supported by many other instrument makers. The document defines the bus's physical connector, the roles of the interconnecting bus wires, the logic conventions, format, and timing of control and data signals, plus the other factors necessary in a communications link that will be capable of interconnecting instruments and peripherals—computers, voltmeters, card readers—made any place in the world.

The IEC technical committee voted to accept the draft standard for ballot by the Commission's member nations, but further changes, if any, should be minor. Final adoption is now about a year off [*Electronics*, Oct. 3, p. 56].

Details of the standard are presented here in a two-part article. The first part is aimed at the system designer, to help him make the most of the interface by understanding its capabilities and limitations. The second part covers the interface from the instrument designer's viewpoint and includes examples of how to use it in instruments.

Part 1: The standard interface and system design

□Building an automated system of instruments already equipped with standard interface circuitry is almost a matter of plugging them into a standard data bus—but not quite. In the case of the Hewlett-Packard interface, which soon may become the international standard, the system designer needs to know how the data bus is used to transfer commands and data between the attached instruments and why some methods of data coding are more useful than others. He also needs to be aware of the constraints that exist on the length of bus cables and the number of instruments that may be connected to it. In short, an understanding of the HP standard interface helps in configuring a system around it.

Figure 1 diagrams the basic interface structure. A set of 16 signal lines interconnects a number of instruments, each of which fulfills at least one interface function or role, depending on the interface capabilities designed into its circuitry.

At any one time, any particular instrument connected to the bus may be either idle, simply monitoring the activity on the bus, or it may be functioning as a talker or listener or controller. As a talker, it sends data over the bus to a listener or listeners. As a listener, it receives such data. As a controller, it directs the flow of data on the bus, mainly by designating which instruments are to send data and which are to receive data. An instrument may be equipped to serve in more than one of these interface roles, depending on the kind of system expected to be built around it.

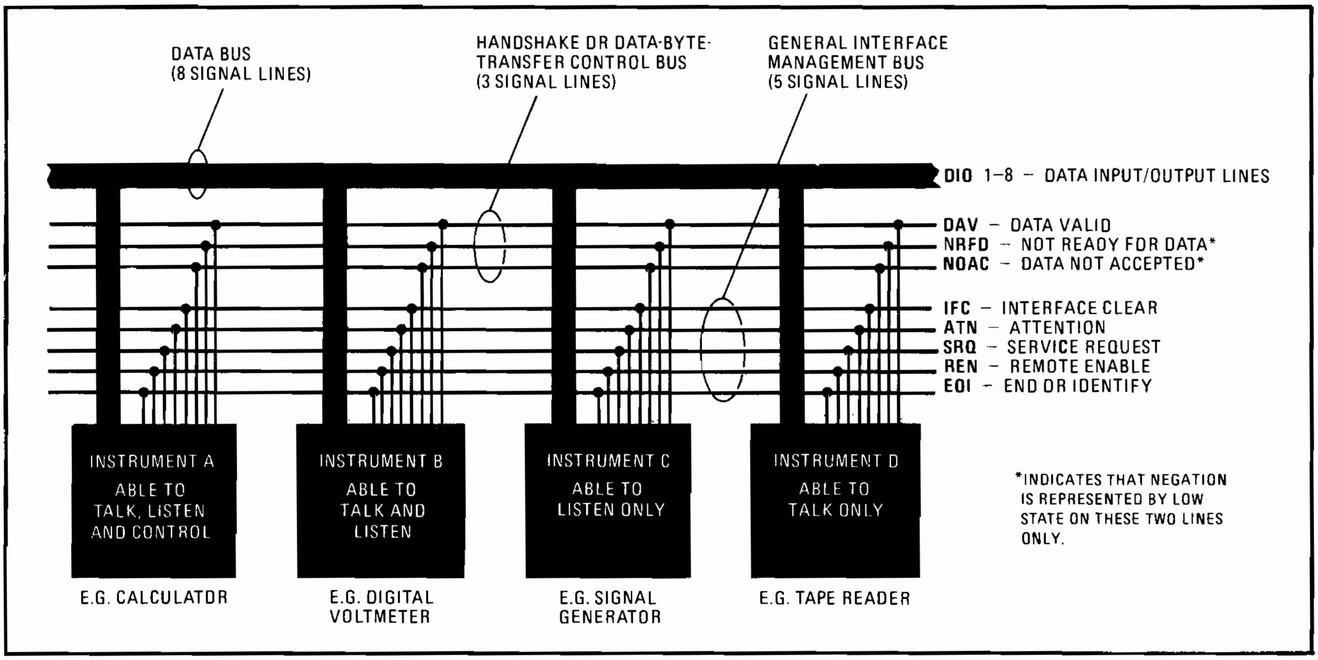
A minimum system need not contain a controller but may consist of just one talker and one listener—a counter and a printer, for example—provided that the two instruments have the interface options that allow a local control to assign them their interface functions. Otherwise, a system must include a controller to designate talkers and listeners. A typical system might include one element with a talker, listener and controller interface, such as a calculator or computer, and a variety of other elements that may be talkers or listeners or both, such as tape readers, signal generators, or digital voltmeters.

All of the active circuitry equipping an instrument to talk or listen or control and simply to monitor the bus is contained within that instrument. The interconnecting bus is entirely passive. Circuitry and bus together make up the interface.

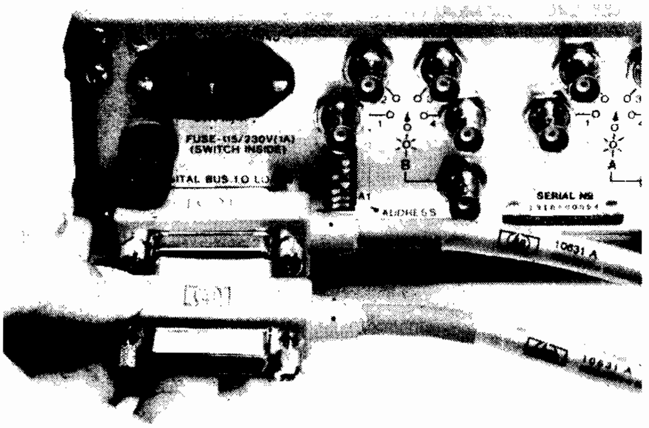
The bus itself consists of 16 signal lines, grouped functionally into three component buses. The data bus (eight lines) is used to transfer data in bit-parallel, byte-serial form from talkers to listeners; it also transfers certain commands from the controller to subordinate instruments. The transfer bus (three lines) is used for the handshaking process, by which a talker or controller can synchronize its readiness to transmit data with the listener's readiness to receive data. The general interface management bus (five lines), as its name suggests, is principally used by the controller.

The operation of the interface is generally controlled by the one member of the instrumentation system that's equipped to act as controller. It uses a group of commands, referred to as interface messages, to direct the other instruments on the bus in carrying out their functions of talking and listening.

The controller has two ways of sending interface messages. Multiline messages, which cannot exist concurrently with other multiline messages, are sent over



1. Data bus. The Hewlett-Packard bus uses a 16-line cable to quickly link up any instruments equipped with appropriate interface circuitry into a system. Data transfer is byte-serial, bit-parallel at rates as high as 1 megabyte per second.



2. Connector. Cables used with the interface system have dual male-female connectors at each end so that they can be stacked, thus allowing more than one cable to be attached to any instrument. This permits either star or daisy-chain configurations.

the eight data lines and the three transfer-bus lines. Uniline messages are transferred over the five individual lines of the management bus.

The commands serve several different purposes:

- Addresses, or talk and listen commands, select the instruments that will transmit and accept data. They are all multiline messages.
- Universal commands cause every instrument equipped to do so to perform a specific interface operation. They include multiline messages and three uniline commands, interface clear (IFC), remote enable (REN), and attention (ATN).
- Addressed commands are similar to universal commands, except that they affect only those devices that are addressed and are all multiline commands. An instrument responds to an addressed command, however, only after an address has already told it to be a talker or listener.
- Secondary commands are multiline messages that are always used in series with an address, universal command, or addressed command (also referred to as primary commands) to form a longer version of each. Thus they extend the code space when necessary.

To address an instrument, the controller uses seven of the eight data-bus lines. This allows instruments using the Ascii 7-bit code to act as controllers. As shown in Table 1, five data bits are available for addresses, so a total of 31 addresses is available in one byte. If all secondary commands are used to extend this into a two-byte addressing capability, 961 addresses become available (31 addresses in the second byte for each of the 31 in the first byte).

TABLE 1: COMMAND AND ADDRESS CODES

Code Form								Meaning
X	0	0	A_5	A_4	A_3	A_2	A_1	Universal Commands
X	0	1	A_5 except	A_4	A_3	A_2	A_1	Listen Addresses
X	0	1	1	1	1	1	1	Unlisten Command
X	1	0	A_5 except	A_4	A_3	A_2	A_1	Talk Addresses
X	1	0	1	1	1	1	1	Untalk Command
X	1	1	A_5 except	A_4	A_3	A_2	A_1	Secondary Commands
X	1	1	1	1	1	1	1	Ignored

Code used when attention (ATN) is true (low)

Addressing details

A talk address selects one instrument to send data and disables all the others with talker circuitry from sending data. That is, sending a talk address selects one and only one instrument to transmit on the data bus, preventing data errors due to wire OR'ing on the data bus. A listen address selects one instrument to receive data, but does not affect the others—they remain as they were, addressed or unaddressed. Thus, several instruments may listen at the same time. Sending a talk address does not affect listeners or vice versa.

Also shown in Table 1 are two other commands associated with the addressing process—the untalk and unlisten commands. They are called "un" commands because they perform exactly the opposite function of an address; that is, they disable an instrument from sending or receiving data. The unlisten command is used whenever a new listener or group of listeners is to be selected in order to disable all the previously selected listeners. The untalk command disables all previously selected talkers. In fact, the untalk command is merely a talk address using an address number to which no instrument may be assigned, and instruments make no distinction between an untalk command and a talk address to a different device.

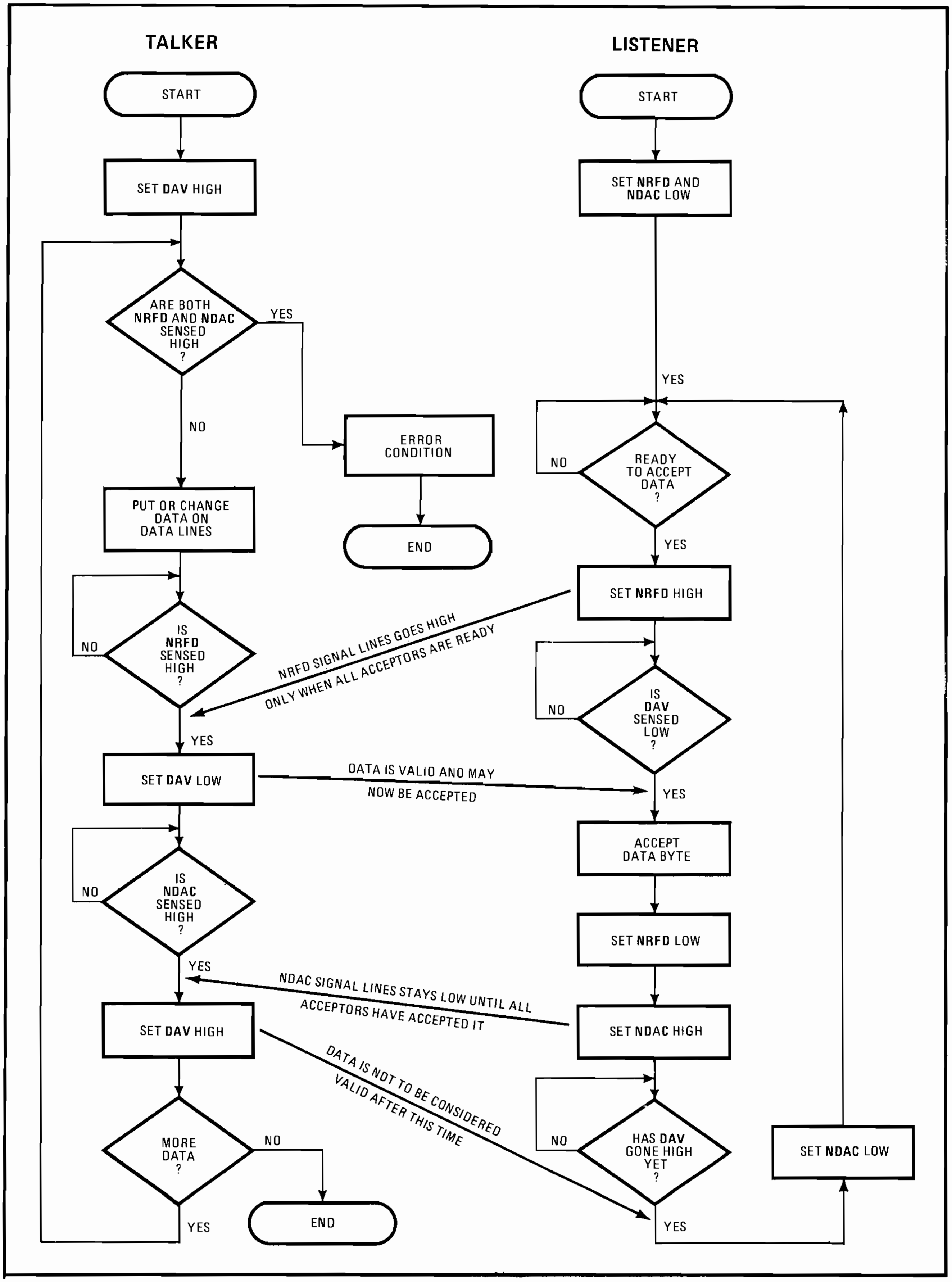
During the configuration of a system, each device must be assigned one or more addresses unique to it. However, two listeners may have the same address if they are always to receive the same data. An instrument is assigned its address by some convenient means such as switches on the rear panel or jumper wires on a printed-circuit board. Typically, there are five switches or jumpers to be set which specify the five bits of the talk or listen address (or both if the instrument is both a talker and a listener). The particular value of the address is the system designer's choice. In the case of instruments using more than one talk and/or listen address, only four bits of the address may be settable.

The actual bus

Instruments are connected into a system with a special piggyback cable (Fig. 2) which has a single male-and-female connector at either end and a lockscrew mechanism which allows one cable to be stacked on top of another and secured. This arrangement allows the user to assemble a system in any configuration he wishes—a line or a star or any combination that's convenient in terms of the space available. (Although the connectors are in theory infinitely stackable, in practice

Closing the loop

The authors will both be available on Nov. 26 and 27 to answer any questions readers may want to ask about their article. Call either David Ricci in California at (408) 246-4300, ext. 2192, or Gerald Nelson in Colorado at (303) 667-5000, ext. 2158, in office hours.



3. Handshake. To ensure that data is properly transferred, this handshaking procedure must be performed every time a talker sends data to a listener or listeners. The DAV line is controlled by the talker, NRFD and NDAC by the listeners.

TABLE 2: HOW TO USE ASCII IN CODING INTERFACE COMMANDS

Recommended	To be avoided
The upper-case alphabet: A – Z	The lower-case alphabet: a – z (some controllers cannot generate or display these)
The digits: 0 – 9	
Common punctuation, used for their standard meanings: (+) (-) (.) (,)	Punctuation used by computer languages for syntax control: (") (space) (_)
End of record: (CR) (LF) (carriage return and line feed)	Most nonprinting characters: EOT hangs up telephones; (NUL), (DEL) mean no information; (BEL) rings a bell; (SO), (ESC) alter the meanings of subsequent bytes; (X 11 11111) is sometimes used as a "rub-out" code.

only two to four are actually stacked to avoid creating a cantilevered structure that may damage a panel.)

Electrical considerations limit the total number of instruments to 15 and the maximum length of cable to 20 meters, although both of these limitations can be overcome by extender or terminal units (to be discussed in Part 2). The fact that the cable terminations are distributed (each device contains a termination) imposes another restriction—the maximum length of cable for any given configuration is two meters times the number of devices up to the maximum of 20 meters. This restriction only applies to the maximum length of cable, not to how it is distributed between the devices.

Data transfer and data rates

When a talker sends a listener data over the data bus, they coordinate their activities by a handshaking process, which is carried out over the three-line transfer bus. As a result, the rate at which data is transferred is determined only by the characteristics of the instruments involved. Moreover, several instruments can listen to the same data simultaneously.

There are essentially four phases to the data transfer cycle: the talker or source generates a new data byte; the states of the data bus's signal lines settle; the listener or acceptor instrument accepts the data (i.e., no longer requires it to be held on the data bus), and the acceptor becomes ready for the next byte. Figure 3 is a flow chart detailing the handshake process.

The time taken to generate data is determined by the characteristics of the source, and the times taken to accept data and to become ready for more depend on the acceptor's characteristics. The data settling time is determined by the characteristics of the transmission system (the drivers, receivers and terminations in the instruments, and the bus cable).

Between them, therefore, the source, acceptor, and transmission system set the upper limit for the data rate at which a bus system will operate. Note that acceptors can be designed to store data for a while, to permit the data-generation and data-settling times to overlap with the acceptor's ready-for-data time and thus increase the bus's data rate.

By careful design and configuration, a rate of 1 megabyte per second can be achieved on a burst basis.

The logic levels employed on the bus's signal lines are TTL levels (high at or above 2.4 volts, low at or below 0.8 V). Driver and receiver circuits must, therefore, be TTL or TTL-compatible devices. The driver circuits are open-collector types capable of sinking 48 milliamperes at 0.4 V. Three-state drivers may be employed on some of the signal lines to increase the speed of data transfers. The receiver circuits may be standard TTL gates or inverters, or equivalent. For higher noise immunity, Schmitt trigger-type gates or inverters are better.

Each signal line is terminated within each instrument with a 3-kilohm resistor to +5 V and a 6.2-kilohm resistor to logic common. Distributing the terminations among all the instruments (rather than having a lumped termination) is a compromise between an ideal design for a transmission-line termination and the desire to minimize the job of configuring a system (i.e., by not restricting the interconnection scheme or requiring an external load device).

Data coding

The only functional restriction on data transfer between a talker and listener imposed by the bus is that it must consist of a sequence of 8-bit bytes. Even though the interface imposes no restrictions on information coding, conventions obviously must exist between the talker and listener, and the system designer is free to choose whichever he prefers.

Still, he should bear in mind that the computers and calculators used as controllers in many systems have established coding and format conventions built into their software. If he can use these conventions, his task will be much easier than if he makes the machines generate and accept arbitrary coding.

The two most common of these conventions are Ascii coding of information bytes and Fortran-style number representations (formatted and free field). Though both of these are subject to ANSI (American National Standards Institute) standards, the American National Code for Information Interchange (Ascii) conforms to the ISO 7-bit code used internationally. In fact, though the kind of bus now proposed as the international standard can be used with other codes, it has used Ascii so often that it became commonly known as the "Ascii bus."

It still is very useful to implement instrument program codes and numeric data with Ascii, where possible, because of the ease with which most controllers read and write Ascii strings and numbers. However, a further restriction to a subset of these codes is also highly desirable, because certain codes have special meanings in these controllers and their software. Table 2 gives recommendations on code use.

Number representations

Note that a measurement instrument that reads out numbers with the least significant digit first causes severe headaches for the system programer. He must read the data as individual bytes and rearrange them to reconstruct the number. On the other hand, a simple read statement in the controller's language is all that's necessary if the meter puts out its data in a form acceptable to standard software. This form for numbers is largely identical to the conventional way numbers are written. Examples are [12976], [-42.67], [+1.00298 E-04]. Multiple values should be delimited by commas, and the most convenient end-of-record indication is [CR] [LF].

If numeric data must be mixed with alpha status information, the alpha characters can generally be made to precede the number and either be read as a string of binary bytes or optionally skipped. □

Part 2: The standard interface and instrument design

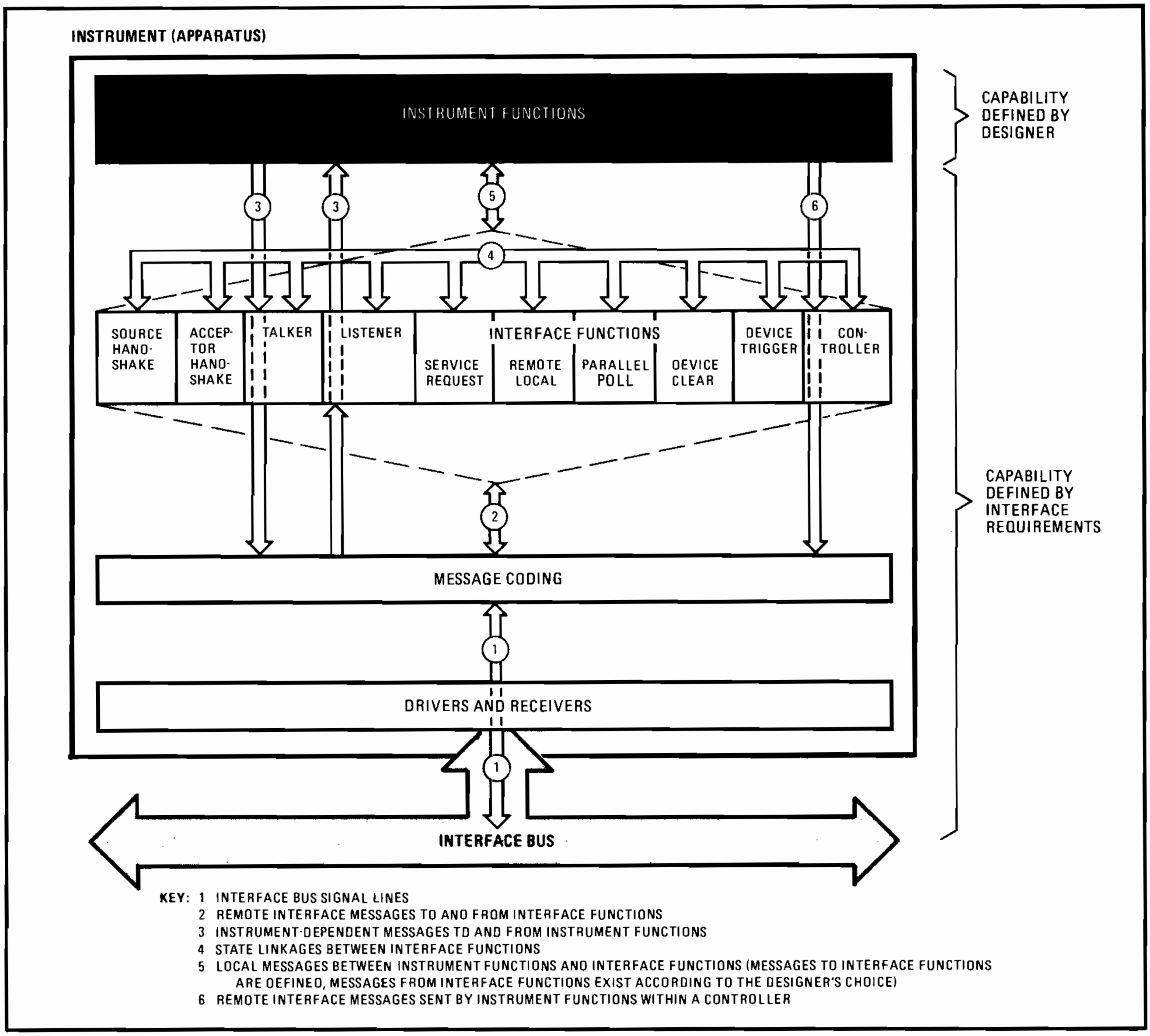
□While the system designer has merely to understand how the standard interface works, the instrument designer has to be able to build part of the standard interface into a piece of equipment. He therefore needs to specify how an instrument must behave if it's to be classified as conforming to the standard.

Writing a specification for this interface, however, presents a significant challenge. The requirements must be explicit enough to ensure compatibility yet flexible enough to allow the designer to tailor an instrument to his particular needs. To achieve this goal, the specification is written in terms of *interface functions* (as distinct from instrument functions), *messages* to and from the interface functions, and *state diagrams* describing the behavior of each of these functions (Fig. 5).

Interface functions and messages

Figure 4 shows how to conceptualize the interface functions and the messages.

Each instrument contains a set of driver and receiver circuits that serve as the electrical interface between the bus signal lines and the instrument's internal logic. To



4. Partition. Instrument designs can be conceptualized as being partitioned into two areas: instrument functions and interface functions. But this division does not necessarily imply two separate physical layouts within the instrument.

leave the designer free to decide what kind of logic to use for an instrument, the instrument's functions are distinguished conceptually from the interface functions and are independent of the needs of the data bus. The interface functions ensure that the instrument behaves correctly with respect to the bus signal lines and are constrained by the needs of the standard interface.

Note that this theoretical separation between the functions of interface and instrument need not imply that they are also physically separated. The aim is simply to make it easy to analyze each of the interface requirements in relation to the instrument's requirements.

Communication between the instrument, its interface functions, and the bus signal lines is described in terms of messages. Actually, every such message is coded into particular electrical states (high and low voltage levels) of the bus signal lines. But in writing a specification, it would be tedious to have to describe exact electrical values of the bus lines for each message, so instead the messages are treated simply as binary functions with values of true or false.

The total capabilities of the interface are grouped into 10 interface functions—five basic functions and five supplementary functions (Tables 3a and 3b).

The basic functions are the ones that appear most frequently in most systems. Almost all instruments will contain either a talker or a listener function or both. Although only a few will incorporate the controller function, an instrument with controller capabilities will almost always be included in a system, and in certain cases multiple controllers can be utilized. The source and acceptor handshake functions are always used in conjunction with the talker, listener and controller.

In Table 3b, the service request function must always be used in conjunction with a talker function, since the instrument has to identify itself as the source of a service request during a serial poll. The parallel poll function differs from serial polling in being initiated by the controller rather than requested by the instrument.

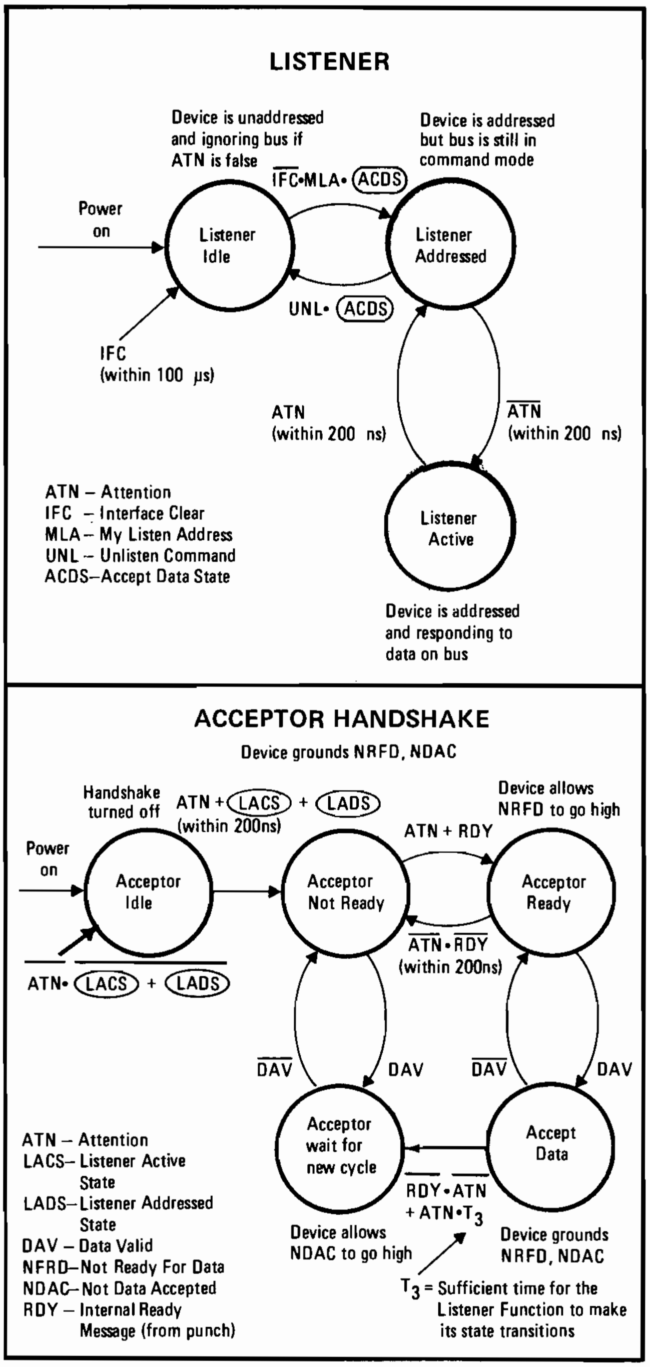
State diagrams are used to describe the sequence of states the interface goes through as it fulfills a particular function in relation to the instrument and the data bus. Two examples are given in Fig. 5.

How to select interface functions

In selecting interface functions, the instrument designer has two degrees of freedom: which functions to choose, and which capabilities of each to choose.

The interface functions are selected by being matched with the instrument's requirements. All the combinations allowed are compatible with each other. That is, a given set of interface functions is guaranteed always to work with other, appropriately equipped, instruments in the sense that the set will not limit the other instruments' operation. Thus, the only mistake a designer can make is omitting a function required for his instrument or the system designers' applications for his instrument.

Once the interface functions are selected, a second set of choices can be made. The various combinations of capabilities that form subsets of each function allow certain capabilities to be omitted if they are not relevant to the instrument's needs. Again, the omission of a capability does not limit the operation of the over-all system, just the particular instrument.



5. States. One way of looking at an instrument's interaction with the data bus is to use state diagrams like the two shown here for any listener tied to the bus (above) and for a listener in the process of receiving data and handshaking with a talker (below).

Four interface design steps

However, selecting the proper group of interface functions and capabilities is only the second step in designing an interface. The first and crucial step is to generate a detailed list of objectives for the remote input/output behavior of the instrument for which the interface is being designed.

These objectives, plus the interface functions chosen

TABLE 3A: BASIC INTERFACE FUNCTIONS

Function	Description
TALKER	
Basic Talker	To let an instrument send data to another instrument.
Talk Only	To let an instrument operate in a system without a controller.
Unaddress if my listen address (MLA)	To prevent an instrument capable of functioning as both a talker and a listener from talking to itself.
Extended Talker (TE)	Same as talker function with added addressing capability.
Serial Poll	To send a "status byte" to the controller and identify itself as the source of a service request.
LISTENER	
Basic Listener	To let an instrument receive data from another instrument.
Listen Only	To let an instrument operate in a system without a controller.
Unaddress if my talk address (MTA)	To prevent an instrument capable of functioning as both talker and listener from listening to itself.
Extended Listener (LE)	Same as listener function with added addressing capability.
SOURCE HANDSHAKE	To synchronize the transmission of information on the data bus by the talker when sending instrument-generated data and by the controller when sending interface messages.
ACCEPTOR HANDSHAKE	To synchronize the receipt of information on the data bus for all interface functions when receiving interface messages and for the listener function when receiving instrument-generated data.
CONTROLLER	
System Controller	To let an instrument send the interface clear (IFC) or remote enable (REN) messages.
Send Interface Clear (IFC)	To let a system controller take charge from another controller and/or initialize the bus.
Send Remote Enable (REN)	To let a system controller enable instruments to switch to remote control.
Respond to Service Requests (SRQ)	To let a controller respond to service requests.
Send Interface Messages	To let the controller send multiline interface messages
Receive Control	To let the controller accept control on the bus from another controller.
Pass Control	To let the controller pass control of the bus to another controller.
Parallel Poll	To let the control execute a parallel poll.
Take Control Synchronously	To let the controller take control of the bus without destroying a data transmission in progress.

to implement them, plus the rules in the interface standard applying to those functions, form the basis of the interface design. They must be detailed enough to spell out exactly the behavior of every wire on both the bus side of the interface and the instrument side.

The third step is to choose a method of logic implementation that fits one's knowledge, economic considerations, and so forth. This part of the procedure is the subject of an entire science and will necessarily be passed over rather lightly in this article, although it is undeniably the most difficult part of the design.

The fourth and final step is to verify hardware performance. If the interface is part of a very general-purpose piece of equipment, like a measuring instrument, calculator, or data peripheral, it is virtually impossible to check its performance in the same system configuration as the customer's. (The number of different system configurations is large and in all likelihood, some of the pieces of some of those systems are not invented yet.) It is very important, therefore, to check very thoroughly that the interface standard rules are adhered to.

Some examples of this design procedure follow. Each involves a different set of interface functions and illustrates a different combination of principles, but they are not completely detailed and do not necessarily represent designs now in development or production.

The first example, a paper-tape-punch interface, will be carried all the way to the schematic level. The other examples—interfaces for a digital voltmeter and a programable calculator, plus a serial terminal unit—will be carried only through the definition phase.

Interface for a paper-tape punch

Though this is a specific type of data peripheral, its list of objectives fits many other instruments—in fact, all those that take digital information from a remote input and convert it into some other form. Printers, programable power supplies, function generators, X-Y plotters, scanners, and programable attenuators all do this.

A block diagram for the tape-punch interface is given in Fig. 6. As for its objectives, it must be able to:

- Be told by the controller of the bus when and when not to punch received data.
- When told to punch, take data bytes from the bus and pass them to the punch data-input lines.
- Provide a punch-initiate signal to command the punch circuits to punch the data byte onto the tape.
- Accept a punch-ready signal from the punch and use it to control the timing of bytes received from the bus (to ensure the data remains stable while punching and notify the data source when ready for the next byte).
- Not punch data received in the bus command mode, and handshake as fast as possible to avoid slowing the bus down unnecessarily.

The most obvious interface function required is the listener function. This is needed to satisfy the first objective. The listener function is the one that basically remembers whether the instrument is addressed to listen or not. Of the subset of four capabilities, only the basic listener capability is needed here.

The only other interface function required for the punch is the acceptor handshake function. This function meets the third objective by indicating when the bus data is valid. It also allows control of byte timing for the fourth objective. Actually, the bus standard requires the acceptor handshake to be included in any interface having a listener function.

The standard rules for these two interface functions are most easily shown by means of state diagrams. The listener function is provided by a flip-flop that is set when the punch's listen address is received and cleared when the unlisten command or interface clear is received. In Fig. 5a, the set states are listener addressed and listener active, with the condition of attention on the bus distinguishing between the two states on the diagram. The cleared state is listener idle.

The acceptor handshake is also provided by one flip-flop plus some gating. In Fig. 5b, if attention is false and the listen flip-flop is not set, the handshake must be idle. The two not-ready states (acceptor not ready and acceptor wait for new cycle) can be one state of the handshake flip-flop, data valid being used to distinguish the wait-for-new-cycle from the not-ready state. The two ready states (acceptor ready and accept data) similarly can be the other state of the handshake flip-flop.

Because of the simplicity of this interface, standard TTL ICs were chosen to implement it. The complete schematic is shown in Fig. 7.

Interface for a DVM

Digital voltmeters represent the very large class of devices that must both listen and talk on the bus. Other measuring instruments that belong to this class include timers or clocks that must be set as well as read remotely, and terminals having both keyboard and display. The DVM interface must be able to:

- Provide two alternate means of controlling the voltmeter's program information—front-panel controls or remote commands.
- When in remote control and addressed, respond to program commands from the bus and pass them on to the voltmeter's control logic (range, function, etc.).
- Accept either a universal command from the bus or a normal program command to initiate a reading.
- Send measurement results and program status data to the bus when addressed to do so by the controller.
- Operate in a controller-less system with a printer or tape punch by being configured to talk without requiring an address.
- Asynchronously request the bus controller's attention for either of two reasons—programs not understood, or measurement complete—and, when polled by the controller, indicate to it which was the reason.

The first of these objectives is completely met by the remote local function in the bus standard. A subset of this function with no local lockout capability could be used for simplification, but full capability is preferable, especially with little foreknowledge of system usage.

The listener and acceptor handshake functions are needed for the second objective. The designer of a talker-listener combination such as a DVM might want to add the "unlisten if my talk address" capability.

For the third objective, the device trigger interface function should be included to allow response to the universal command, GET (group execute trigger). GET has the obvious meaning of "take a reading". The normal program command to take a reading can be brought in the same way as range and function information, under control of the listener function.

The talker function is required for data output to fulfill the fourth objective of the DVM interface. It is probably appropriate to include all the capabilities of the talker function (except perhaps the extended talker). Serial poll will be necessary to be able to indicate reasons for requesting service via a status byte to meet the last objective.

Talk-only mode is the means of achieving the second-to last objective. Unaddress-if-my-listen-address is a possible inclusion, providing the controller with a minor saving in software by not requiring an unlisten com-

TABLE 3B
SUPPLEMENTARY INTERFACE FUNCTIONS

Function	Description
SERVICE REQUEST	To let an instrument indicate to the controller that some event has occurred and request it to take some specific action asynchronously with respect to other bus operations (one SRQ function is required for each independent reason for requesting service).
REMOTE-LOCAL Basic Remote-Local	To let the control of the instrument be switched between its local (manual) controls and remote control (programing codes received while addressed as a listener).
Local Lock Out	To let the local control "return to local" be disabled.
PARALLEL POLL Basic Parallel Poll	To let instruments return one-bit status to the controller. Up to eight instruments may respond simultaneously. More than one instrument may respond on the same status line so that logical operations (AND, OR) may be performed on a group of instruments.
Parallel Poll Configure	To let the instrument be configured by the controller.
DEVICE CLEAR Basic Device Clear	To provide a means by which an instrument (device) may be initialized to a predefined state. All instruments are cleared concurrently.
Selective Device Clear	To clear individual instruments (devices) selectively.
DEVICE TRIGGER	To let instruments (devices), either singly or in a group, be triggered, or some action be started.

mand to be sent between programing the DVM and accepting its data. Moreover, since the DVM is a talker, it also requires the source handshake function.

Finally, the service request function is required, since, to meet the last objective, the DVM must be able to request asynchronous attention. Normally, an instrument having two reasons for requesting service would need two service request functions, but in this case the reasons are not independent. The DVM probably cannot take a reading and then request service if its program was not understood. Conversely, if the DVM is requesting service to indicate a measurement is complete, it is probably not working on program data or it is receiving erroneous program data.

In general, to minimize hardware logic costs in an instrument, it's best to consider the design as a unified whole (rather than assume hardware partitions from the start) and to make the circuitry operate time-serially (as far as is consistent with speed requirements). Application of these two principles maximizes the opportunity to share logic hardware among many tasks, though it may increase trouble-shooting time. It also enables the designer to take more advantage of available or custom LSI and MSI circuits.

This approach suggests that minimum cost to the customer would result from combining the interface functions described above with the voltmeter's functions (analog-to-digital conversion, program interpretation, etc.) and do one logic design for them all.

A calculator interface

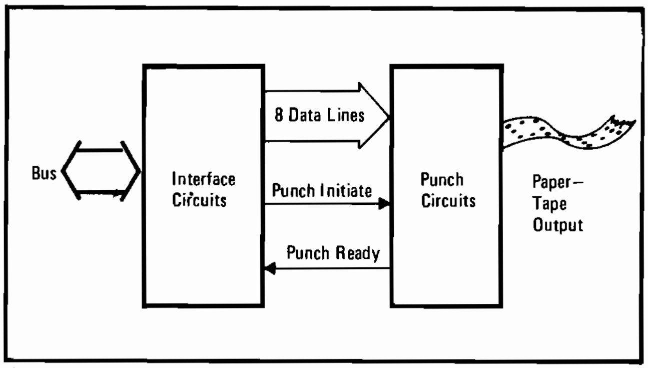
A programable calculator is representative of controllers of medium complexity. Modern calculators are powerful enough to replace minicomputers yet inexpensive and simple enough to use to replace paper-tape readers in many system control applications.

Let's suppose that this particular calculator is characterized by having a complete algebraic language (Basic, for example) and no interrupt capability (the calculator can execute only one program statement at a time). Moreover, it must be in control of the system it is interfaced to. There is no convenient way to treat the machine as a number-crunching peripheral to something else, or to wake it up and make it accept and execute programs from the I/O port.

The interface to be designed in this example is the combination of an I/O card, to handle specifics of communication between the bus and the calculator's internal I/O structure, and a "firmware" driver (software encoded into read-only memory), to handle special protocols and details relating to bus operation.

The calculator interface must be able to:

- Send data to instruments on the bus (numeric data and program data). This data could be binary bytes, Ascii-coded numbers, or Ascii strings.
- Receive similar data from the instruments on the bus.
- Send addresses and universal commands to the bus so as to control information flow.
- Control remote local functions in other instruments.
- Test the status of the service request line at any time, allowing software branching to service routines in lieu of true interrupt capability.
- Take charge of and initialize the bus at any time, so that erroneous operations can be halted and a program be run from a known initial condition for the system.



6. Punch. As shown by its block diagram, a tape punch can be partitioned into the punch circuits that will enable it to perform its functions as an instrument and the interface circuits that are necessary to connect it to the data bus.

The first of these objectives is met by the talker and source handshake functions. Even though the calculator is a controller, it is worth reemphasizing that the sending of instrument-generated data with attention false on the bus is done only by talkers, never by controllers The capabilities of talk only and unaddress on my listen address are not necessary for this type of instrument.

The second calculator-interface objective is realized by including listener and acceptor handshake functions. The listener function has one additional capability when combined with a controller; the calculator can, if it wishes, tell the listener function to listen or unlisten internally, without sending those commands to the bus.

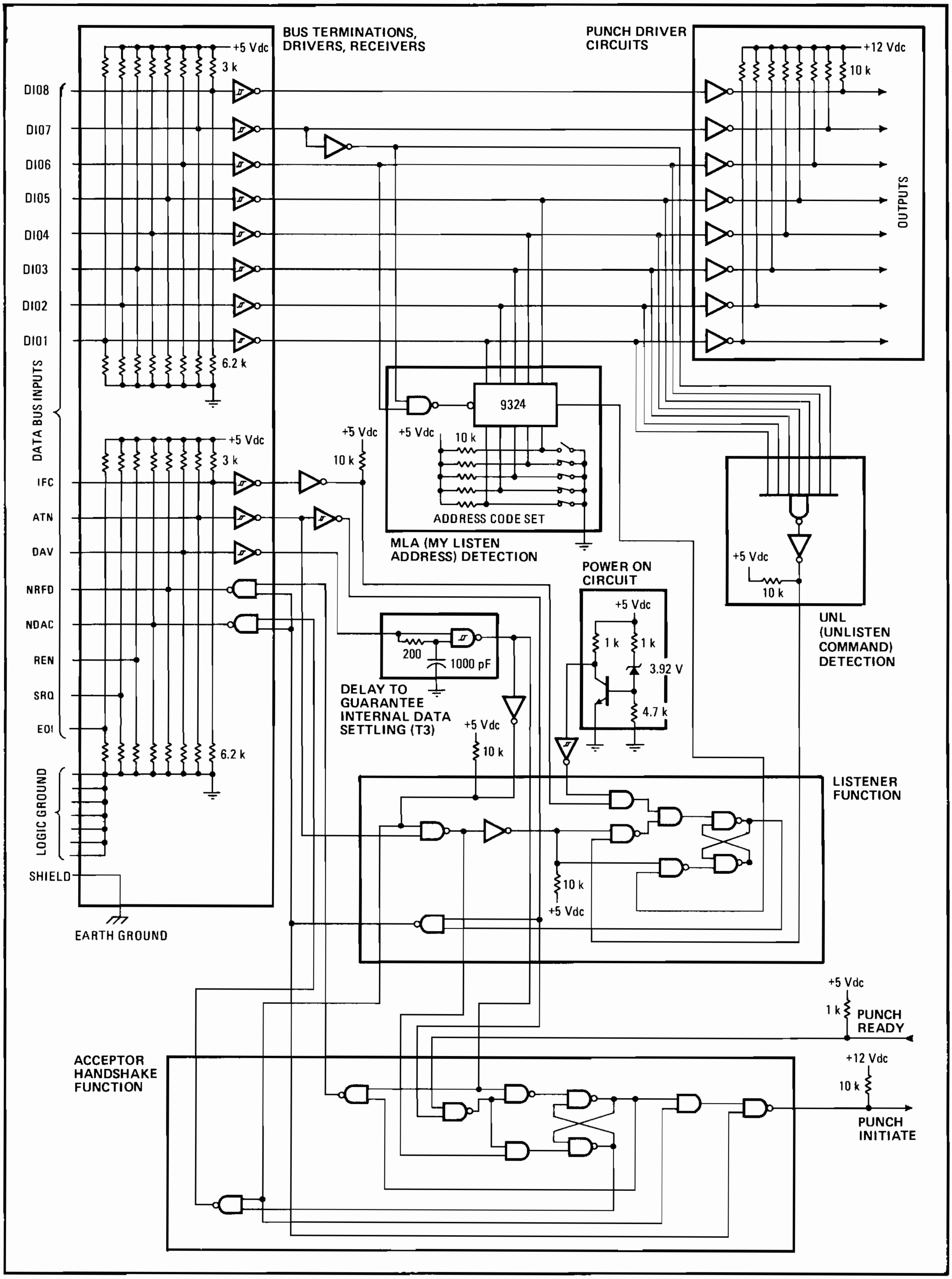
The next three objectives are implemented by various capabilities of the controller function—send interface messages, send remote enable, and respond to service requests, respectively. The send remote enable capability also requires the system controller function.

For convenience, the calculator could also include the "send interface clear and take charge" capability to allow bus initialization by pulling one line. But the last objective can also be implemented by the "send interface messages" capability since by sending attention true, all instruments must listen to the calculator.

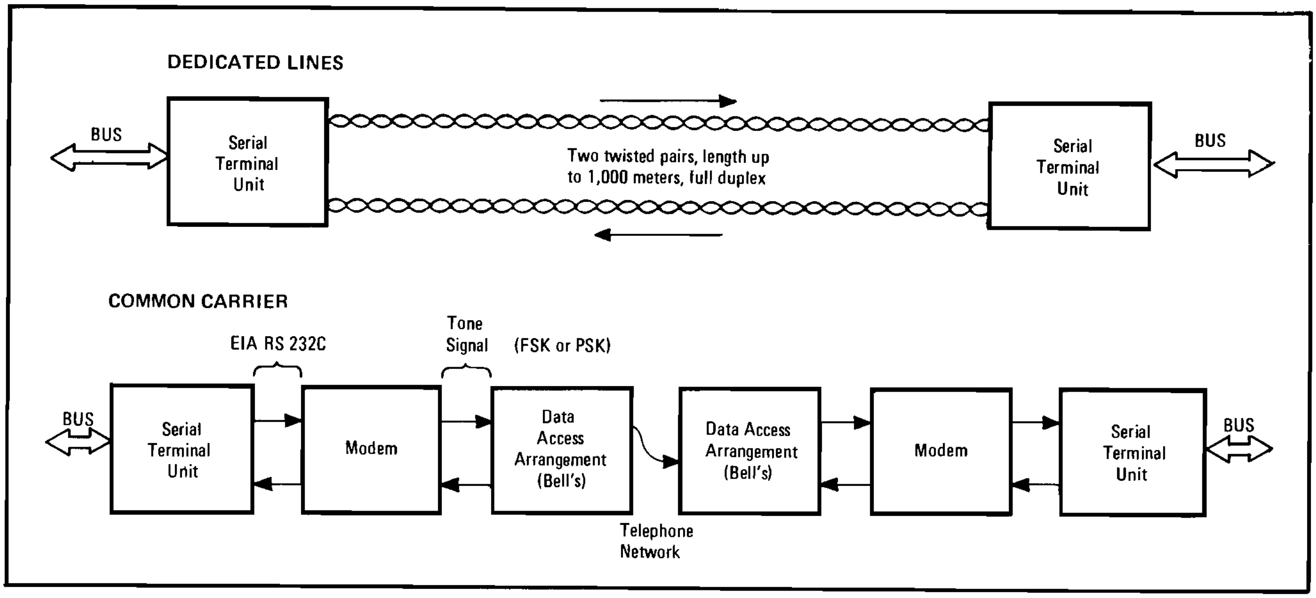
A serial terminal unit

Sometimes it is necessary to extend instrument-system interconnection length beyond the 20 meters provided by the bus. For instance, monitoring or test subsystems might have to be scattered throughout a manufacturing plant, so that a group of instruments might need to be several hundred feet from their controller. The technique that best meets this need is a totally serial transmission. This minimizes the cost and bulk of interconnection cables and makes it possible to transmit the information by telephone.

Use of a terminal unit can supply the designer with the advantages, where appropriate, of having two distinctly different interfaces. In this example, by designing a bus-to-serial interface, he could preserve the party-line communication and standardized instrument input/output of the bus yet still allow easy communi-



7. Design. Because of the simplicity of a tape-punch interface, standard TTL integrated circuits were chosen to implement the design. The class of devices that take digital data and convert it to some other form may make use of similar circuitry.



8. Extension. Serial terminal units may be used to extend the length of the data bus, either through dedicated lines or by common carrier. The units lower the cost of transmission by taking parallel data from the bus and converting it to serial form.

cations among instruments that may be separated by anything from a thousand feet to thousands of miles.

A serial terminal unit must be able to:

- Be used in two system modes (as in Fig. 8).
- Convert bus information on 16 lines into 8-bit serial words as efficiently as possible.
- Minimize software problems by acting in as "transparent" a fashion as possible. A controller on one end of the phone should be able to converse with an instrument at the other end exactly as if instrument and controller were on the same bus in the same room.
- Control an automatic dialer to allow communication to be established over the telephone network without a human at each end. (The information for the dialer will come from a bus talker.)
- Take into account the fact that modems used for voice-grade line communications greatly restrict the data rate achievable on the bus (details below).
- Provide a service request to the controller from either end for these five conditions: parity error in serial; handshake error (handshake overrun at receiver serial terminal unit when in on-line mode); line disconnect; and (if used with a dialer) busy, and call complete.

To elaborate on the fifth objective, these modems operate at 10, 30, and less commonly 120 bytes/second. Faster speeds can be achieved over dedicated private or leased lines, but require synchronous operation. If a complete asynchronous handshake is required by the system, data rates are at least halved again because a byte would have to be transmitted in each direction sequentially to insure a correct handshake cycle. Fortunately, most instrument and controller responses are rapid enough, compared to the above data rates, to allow information to be transmitted "open loop" without overrunning the receiving device handshake.

To allow bus systems to operate efficiently with these terminal units, the units must be provided with three modes of operation—off-line, on-line and handshake—all of which must be programable from a controller.

The off-line mode allows the controller to communicate with the bus devices at its end of the phone line at rapid bus speeds when information does not have to be exchanged with the remote part of the bus system. The on-line is an open-loop mode used when device responses are rapid enough for the byte rate of the modem not to overrun them. The handshake is a closed-loop mode that allows full asynchronous operation by having the receiving terminal unit re-transmit a byte to the sender as soon as the data is accepted.

What interface functions will meet these objectives? The transmitting terminal unit will operate by accepting data from the bus, using the acceptor handshake function, and transmitting the data bytes serially to the other end. Any changes in state of the five management bus lines (attention, remote enable, service request, end or identify, and interface clear) will cause a "cycle steal:" the transmitter will sense those changes and generate a two-character code sequence over the serial link. This sequence is decoded by the receiving terminal unit which drives the management bus at its end.

The prerogative of driving the three management bus lines called remote enable, attention, and interface normally belongs exclusively to the controller function when in active control. In this special instance, however, the terminal unit will drive those lines from instructions received by the system's active controller, but only on the bus side having no active controller of its own. In the strictest sense, the terminal unit has a controller function that from the bus's viewpoint behaves indistinguishably from a standard controller function. Functionally, however, the unit is only operating to keep the management bus lines in the same condition at both ends of the serial link.

The serial terminal unit does require these interface functions of its own: the source handshake, the acceptor handshake, the talker function (for responding to serial poll), and five service request functions (i.e., five independent reasons for requesting service). ☐

A STANDARD INTERFACE APPLIED TO MEASUREMENT SYSTEMS

Jane G. Evans
Hewlett-Packard Company
Product Manager
Palo Alto, California

Howard Merrill
Hewlett-Packard Company
Design Engineer
San Diego, California

ABSTRACT

A new interface standard has proven to be the means for implementing cost effective measurement and test systems based on benchtop instruments. An increasing number of vendors are offering devices incorporating this new standard, which has been accepted by the American National Standards Institute (ANSI) and by the Institute for Electrical and Electronics Engineers (IEEE) and is under active consideration by the International Electrotechnical Commission. This paper discusses the standard digital interface bus as the communications structure for a controller (desk top calculator or minicomputer), signal sources, measuring instruments such as voltmeters and frequency counters, and output devices such as plotters and printers. Applications are described that make visible the ways laboratory researchers and test engineers can benefit from systems which are easily assembled and reassembled rather than dedicated to one purpose. One specific application is environmental and production tests for a tape recorder. A system that can be operated automatically or manually will be discussed. This system repetitively measures 8 channels at 6 speeds, recording and plotting such measurements as playback frequency response and signal-to-noise ratio.

INTRODUCTION

The new instrumentation interface standard IEEE-488-1975 (1) issued a year ago has now amply demonstrated its value for instrumentation systems. As predicted, the number of instruments and controllers offered by various manufacturers using the standard interface has grown rapidly. With interest in the measurement community at a peak, it is appropriate for this paper to describe the interface in terms of its use in a practical system. This application, a production test system for recorders, is discussed not as an end in itself but to show how the interface serves for small systems combining benchtop instruments and a desktop calculator.

Optimized for instrument interfacing applications, the standard covers three of the four primary characteristics associated with complete interface systems: mechanical (connectors and cables), electrical (driver and receiver circuit parameters), and functional (signal line definitions, protocol and timing relationships, etc.) On the other hand, device-dependent operational characteristics are properly excluded. This focus toward specifications that are device and system independent leads to a broadly useful interface.

Data transfer among the up to 15 devices the interface can accommodate on one contiguous bus is byte-serial, bit parallel. Table I summarizes bus specifications.

The Interface Bus - A Digital Pathway

The purpose of the interface bus is to serve as a communications pathway for the system. This it fulfills by providing a way for controllers, measuring instruments, and many other devices to send and receive messages from each other. Messages include measured values and control information.

Use of the term "bus" implies a parallel connection scheme, in this case 16 signal lines. The interface functions for each system component are contained within it. Only passive cabling is needed to connect the system.

Eight of the bus lines transfer messages in a byte-serial, bit parallel manner. Each byte is sent under the control of three handshake lines. The handshake technique uses an interlocked sequence that lets a source and one or more acceptors transfer each byte securely and asynchronously. The remaining five bus lines are for control of bus activity.

Devices are described operationally as talkers, listeners, or controllers for purposes of discussing a network for a given communication over the bus. The controller in the systems to be described here is a desktop calculator. To set up the communications network, the controller establishes one device as a talker and one or more devices as listeners. This the controller accomplishes by setting a specified control line true and sending talk or listen addresses over the data lines.

Addresses are determined for each device at systems configuration time either by switches or by jumpers on a PC board.

With the network established, a talker can send a message to one or more listeners.

It is useful at this point to ask, "What does an instrument need to communicate over the bus?" One thing is its program string. For example, the interface bus option for the Hewlett-Packard Model 3490A Digital Multimeter allows the instrument to be controlled remotely. Remote programming of range, function, and operating mode is conveyed by ASCII character strings. For example, the string "FOR4TIM3E" sets up the multimeter to measure dc volts on the 10V range and triggers it. Another communication is its measured value. The multimeter outputs a prefix of two-character pairs signifying normal operation or overload, and function; followed by polarity, data, and terminator (carriage return-linefeed).

In the interchange just described, the multimeter first was addressed as a listener by the calculator and sent its program string. It then was addressed as a talker to output its measured value to the calculator.

The controller's communications over the bus are more extensive than an instrument's. To control information flow, the controller sends addresses as already described. The controller also selects remote or local operation. Under local control, an instrument responds to front panel controls set by an operator; under remote control, the instrument obeys codes sent over the bus. One of the bus's many useful features is this provision for letting a device be temporarily changed to local control to permit an operator to interact with it for set-up or troubleshooting purposes. The controller has one control line called interface clear that it uses to take control of the system. This is necessary when an operator wishes to intervene. Other communications the controller makes include status testing. The bus protocol has means for the controller's receiving a message that some device wants service and for polling to discover which one (serial polling). There is also provision for a parallel poll type of operation.

Applications

Instrumentation systems based on the IEEE-488 standard interface are now at work in a range of applications from a lab bench mini-system performing distortion analysis to a plant engineering system monitoring industrial effluent.

Environmental and production test areas are naturals for instruments and controllers joined with the interface bus because these systems lend themselves well to repetitive testing, yet are flexible to take on different tasks. For instance, a system used one month for reliability testing of prototype devices in a temperature chamber could be reassembled the next month to monitor temperature and airflow transducers on an air conditioning system.

The key reasons generally cited for automating production tests are three: (1) increase operator throughput, (2) minimize the chance for human error, and (3) obtain a hard copy of test results. A benefit that may outweigh the rest becomes apparent after automated testing replaces manual testing: tests have become more thorough. Each device is characterized more completely.

Hard copy records kept on file are useful later to document failures traced to, say, a vendor change in some component. Also records may be duplicated and sent to customers who want verification of critical specifications.

To help the reader further evaluate what potential an interface bus system could have to meet a particular need, the next sections will describe the development of a small, semi-automated production test system. The purpose of this system is to provide frequency response, linearity, and signal-to-noise testing of a portable 8-channel instrumentation tape recorder.

Automatic System Meets a Need

When an engineer faces a greatly increased test load he usually considers whether it might be possible to keep the number of test technicians about the same and increase their effectiveness with an automated or semi-automated system.

Hewlett-Packard's San Diego Division had been manually testing production runs of analog tape recorders having four record-reproduce channels and three operating speeds. With a new recorder having 8 channels and 6 speeds entering production, the repetitive testing load would be quadrupled.

A study to examine the merits of automated testing was begun by establishing that the system must measure (1) record-to-playback frequency response, (2) signal-to-noise ratio, and (3) linearity as recommended by IRIG (Inter-Range Instrumentation Group).

The next step in the study was to specify system components capable of meeting measurement needs. At this point, an engineer faces the risk of over simplifying the problems of automating complex measurements with consequent heavy costs and time lags to make the automatic system a working reality. The interface bus and benchtop instruments greatly reduce this risk because of their relatively low cost and flexibility. In addition the instruments comprising the system can be used as stand-alone equipment or in other systems. The situation can be very different where a system of components incapable of serving except within a dedicated system environment must be specified.

The next step in the study was programming the system (writing the software). The cost of software is, as a rule of thumb, estimated to be at least twice the cost of the hardware. In actuality, it turned out that the time to develop the initial software on a desktop calculator was one month. This time is remarkably short compared to what might be required for programming a minicomputer-based system. Also, programming skill at the assembly language level is not required, and the ease of program editing on a calculator is a real asset.

To anticipate the outcome of the system actually built, the incremental cost (above what would have been necessary even if all testing were to be done manually) was found to be relatively low. The entire job was kept relatively simple and quick by use of the interface bus, a desktop calculator, and benchtop instruments. Results have borne out the oft-repeated maxim that an automated system takes far more data - at lowered risk of operator error - resulting in more definitive tests of units than manual testing ever could.

PRODUCTION TESTS TO BE AUTOMATED

To evaluate production runs of the data tape recorder requires that two primary measurements be made for each of the 8 channels at each of the 6 speeds. These two measurements are frequency response and signal-to-noise ratio. In addition, linearity is measured for all FM channels. These measurements are treated briefly in turn to provide background for discussions of measurement problems and test data output. The first is record-to-playback frequency response.

Frequency Response

Frequency response is measured by recording selected frequencies onto tape, then playing them back and observing the amplitude. The output signals from the reproduce head are delayed with respect to their input counterparts by the travel distance around the capstan as shown by the tape path of Figure 1. The program must take this into account by delaying readings a like amount.

Response plots produced by the test system for a direct channel are shown in Figure 2. Measured points are concentrated in two areas of primary interest, the low frequency band edge region and the upper band edge region.

The lower band edge region is examined for "head bumps", an effect caused by signals picked up by the reproduce head pole-pieces rather than just the reproduce head gaps. The upper band edge region is examined for variations due to wavelength effects in the reproduce head gap and to tape-head separation losses. The midband region is merely sampled to check flatness (indicating correct equalization) and to provide a reference for the entire response (1/10 band edge).

The calculator program sets the number of points taken per decade appropriate to each region at each tape speed. For example, at the slowest speed, 15/32 ips (inches per second), the program plots the lower band edge region with a frequency resolution of 5 points per decade and plots the upper band edge region at 10 points per decade. For speeds of 3-3/4 ips and higher, the lower band edge is plotted at 20 points/decade; midband is plotted at 4 points/decade; and upper band edge is plotted at 10 points/decade. At these higher speeds, the maximum 20 points/decade resolution is needed to view "head bumps" clearly.

Figure 3 shows an FM plot. Here, 15 points per decade are taken in the upper band edge region. Only a few points are needed in the midband region, as FM response extends down to dc and is quite flat until the upper band edge is reached. This is because FM roll-off characteristics are determined primarily by electronic filtering rather than by tape head effects.

Signal-to-Noise Ratio Measurement

The second test to be automated is signal-to-noise ratio (S/N). In manual testing it had been customary to use a voltmeter of the true RMS type to measure S/N for the direct channels and a voltmeter of the average-reading type to measure S/N for FM channels (IRIG 106-73 and 118-73 respectively). For purposes of automating the system, it was desired to use a single voltmeter for all measurements. Having one voltmeter would reduce software complexity as well as cost. A way was found to use a single voltmeter for both types of measurements through the use of correction factors, as will be discussed.

The noise present in an FM channel owes mainly to flutter, a variation in the rate at which tape passes the heads. This variation produces an unwanted frequency modulating signal whose content is typically at lower frequencies. For instance, an 11 Hz (at 15 ips) signal is introduced as the tape passes the capstan due to a flutter component caused by capstan "once around".

On the other hand, the noise in a direct channel owes primarily to random orientation of the tape's magnetic particles. It is white noise and is relatively flat. (Electronic noise is also rather flat; it is several dB down in the direct channels and is negligible in FM channels).

The system voltmeter selected is the Hewlett-Packard Model 3490, an average-responding voltmeter. It was adapted to the direct channel measurement (RMS) by use of a conversion factor. It has been shown (4) that measurements of white noise made with an average-responding voltmeter can be converted to the true value measured by an RMS responding voltmeter by adding 1.05 dB to the average-responding measurements (reduced S/N ratio) so long as the bandwidths are the same for the two voltmeters.

Linearity

The third test to be automated is the linearity measurement for each FM channel. Input-output characteristics (voltages) are measured as recommended by IRIG at three points: zero carrier deviation,

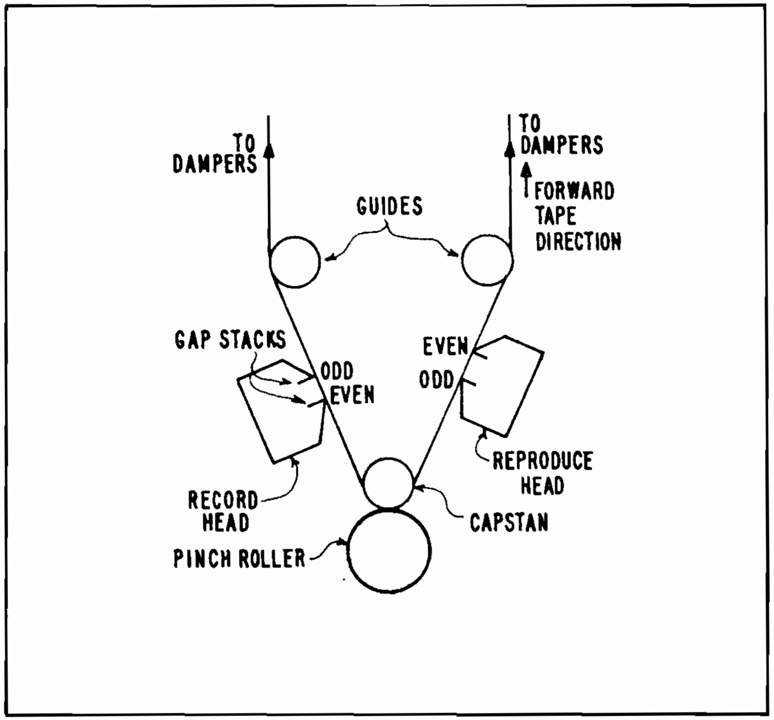


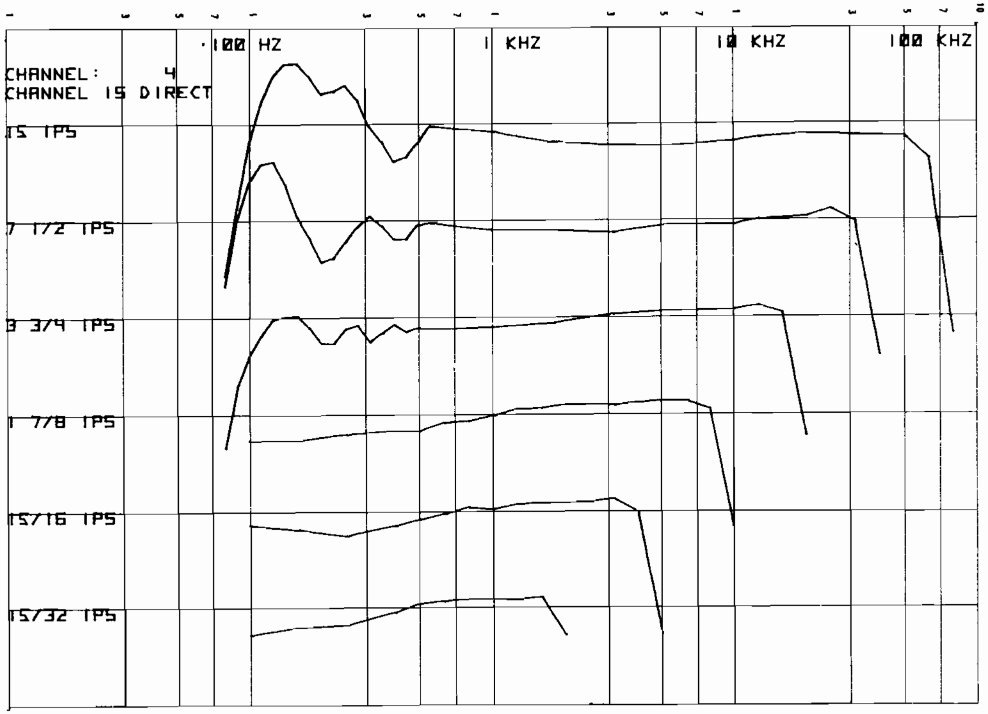
FIGURE 1. TAPE PATH; RECORD HEAD, REPRODUCE HEAD, AND CAPSTAN.

TABLE I

INTERFACE BUS SPECIFICATION

SUMMARY

Interconnected Devices:	Up to 15 maximum on one contiguous bus.
Interconnection Path:	Star or linear bus network up to 20 meters total transmission path length.
Signal Lines:	Sixteen active total; 8 data lines and 8 lines for critical control and status messages.
Message Transfer Scheme:	Byte-serial, bit-parallel, asynchronous data transfer using interlocked three-wire handshake technique.
Maximum Data Rate:	One megabyte per second over limited distances; 250-500 kilobytes per second typical over full transmission path.
Address Capability:	Primary addresses, 31 Talk and 31 Listen; secondary (2-byte) addresses, 961 Talk and 961 Listen. There can be a maximum of 1 Talker and up to 14 Listeners at a time.
Control Shift:	In systems with more than one controller, only one can be active at a time. The currently active controller can pass control to one of the others. Only the controller designated as system controller can assume control.
Interface Circuits:	Driver and Receiver circuits TTL-compatible.

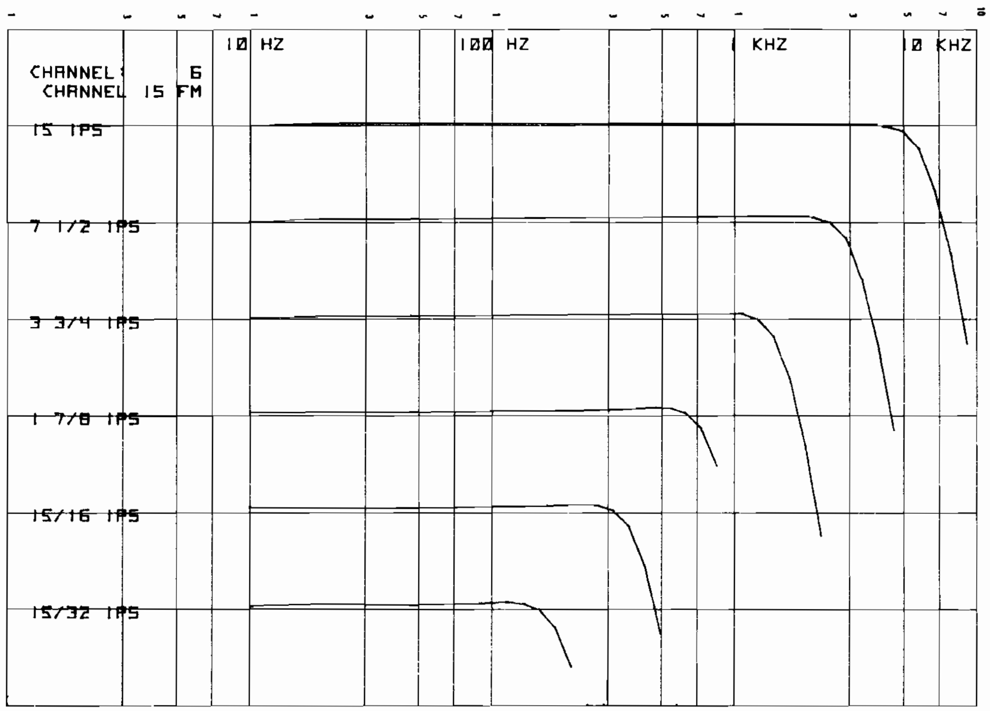


(A)

```
CHANNEL BEING TESTED ?

SET THE INPUT LEVEL ADJUSTMENT POT FOR 0 DB INDICATION
ON THE 'VU' METER (METER IS ON PEAK, MEASURING INPUT)
CHECK OUTPUT TO SEE THAT CHANNEL IS FUNCTIONAL
PRESS CONTINUE/EXECUTE

REFERENCE LEVEL IS     1.58260 VOLTS AC
PROGRAM WILL GIVE PLOTS AND SIGNAL TO NOISE RATIO
FOR EACH SPEED, STARTING WITH THE LOWEST INDICATED

ENTER STARTING SPEED IF DIFFERENT FROM 15/32 IPS
PUSH CONTINUE/EXECUTE IF STARTING FROM 15/32 IPS
USE FORMAT SHOWN BELOW
15/16 I.P.S.
1 7/8 I.P.S.
3 3/4 I.P.S.
7 1/2 I.P.S.
15 I.P.S.

CHANNEL IS NUMBER 4
STARTING SPEED = 15/32 I.P.S.,
  INDICATE IF CHANNEL IS FM OR DIRECT

CHANNEL IS DIRECT

NOISE LEVEL AND FM ZERO AT 15/32 I.P.S.
CORRECTED AVERAGE IS        37.00 DB
S/N NUMBERS ARE RELIABLE

NOISE LEVEL AND FM ZERO AT 15/16 I.P.S.
CORRECTED AVERAGE IS        37.45 DB
S/N NUMBERS ARE RELIABLE

NOISE LEVEL AND FM ZERO AT 1 7/8 I.P.S.
CORRECTED AVERAGE IS        38.54 DB
S/N NUMBERS ARE RELIABLE

NOISE LEVEL AND FM ZERO AT 3 3/4 I.P.S.
CORRECTED AVERAGE IS        38.51 DB
S/N NUMBERS ARE RELIABLE

NOISE LEVEL AND FM ZERO AT 7 1/2 I.P.S.
CORRECTED AVERAGE IS        38.71 DB
S/N NUMBERS ARE RELIABLE

NOISE LEVEL AND FM ZERO AT 15 I.P.S.
CORRECTED AVERAGE IS        39.46 DB
S/N NUMBERS ARE RELIABLE
```

(B)

FIGURE 2. SYSTEM-PRODUCED (A) PLOTS OF FREQUENCY RESPONSE AND (B) PRINTOUT OF RESULTS FOR DIRECT CHANNELS.



(A)

```
CHANNEL BEING TESTED ?

SET THE INPUT LEVEL ADJUSTMENT POT FOR 0 DB INDICATION
ON THE "VU" METER (METER IS ON PEAK, MEASURING INPUT)
CHECK OUTPUT TO SEE THAT CHANNEL IS FUNCTIONAL
PRESS CONTINUE,EXECUTE

REFERENCE LEVEL IS     1.70680 VOLTS AC
PROGRAM WILL GIVE PLOTS AND SIGNAL TO NOISE RATIO
FOR EACH SPEED, STARTING WITH THE LOWEST INDICATED

ENTER STARTING SPEED IF DIFFERENT FROM 15/32 IPS
PUSH CONTINUE,EXECUTE IF STARTING FROM 15/32 IPS
USE FORMAT SHOWN BELOW
15/16 I.P.S.
1 7/8 I.P.S.
3 3/4 I.P.S.
7 1/2 I.P.S.
15 I.P.S.

CHANNEL IS NUMBER 5
STARTING SPEED = 15/32 I.P.S.,
  INDICATE IF CHANNEL IS FM OR DIRECT

CHANNEL IS FM
NOISE LEVEL AND FM ZERO AT 15/32 I.P.S.
FM ZERO IS     0.00969 VOLTS AT THIS SPEED
CORRECTED AVERAGE IS        46.68 DB
S/N NUMBERS ARE RELIABLE

NOISE LEVEL AND FM ZERO AT 15/16 I.P.S.
FM ZERO IS     0.01020 VOLTS AT THIS SPEED
CORRECTED AVERAGE IS        46.47 DB
S/N NUMBERS ARE RELIABLE

NOISE LEVEL AND FM ZERO AT 1 7/8 I.P.S.
FM ZERO IS     0.01035 VOLTS AT THIS SPEED
CORRECTED AVERAGE IS        49.66 DB
S/N NUMBERS ARE RELIABLE

NOISE LEVEL AND FM ZERO AT 3 3/4 I.P.S.
FM ZERO IS     0.00947 VOLTS AT THIS SPEED
CORRECTED AVERAGE IS        49.70 DB
S/N NUMBERS ARE RELIABLE

NOISE LEVEL AND FM ZERO AT 7 1/2 I.P.S.
FM ZERO IS     0.01150 VOLTS AT THIS SPEED
CORRECTED AVERAGE IS        49.20 DB
S/N NUMBERS ARE RELIABLE

NOISE LEVEL AND FM ZERO AT 15 I.P.S.
FM ZERO IS    -0.00758 VOLTS AT THIS SPEED
CORRECTED AVERAGE IS        48.50 DB
S/N NUMBERS ARE RELIABLE
```

(B)

FIGURE 3. SYSTEM-PRODUCED PLOTS OF (A) FREQUENCY RESPONSE AND (B) PRINTOUT OF RESULTS FOR FM CHANNELS.

maximum positive deviation +40% and maximum negative deviation -40%.

A straight line through zero is constructed from the negative-most point and is extended in a positive direction. The error between the positive-most deviation predicted by the constructed line and the actual measured point is defined to be a measure of linearity.

SYSTEM DESCRIPTION

After the set of required measurements had been determined, the next task was to select measuring instruments and a controller capable of performing the measurements.

The system controller selected for this application is the Hewlett-Packard Model 30 Calculator and its Model 9866A Printer. This desktop calculator is programmed in BASIC and has a maximum mainframe memory of 8K words. (The final test program occupies some 6K words.)

Figure 4 is a block diagram of the system used to test data tape recorders. Figure 5 shows the system on a rolling cart in the production area. The Model 3320A Synthesizer is the input device for the response measurements, supplying test frequencies to be recorded and played back under program control.

The Model 3490A Digital Multimeter measures ac and dc voltages under program control. The manually operated switch is a simple device built to provide +2.5V and -2.5V inputs for the linearity measurement. Plots are produced by the 9862A X-Y Plotter.

A bus interface was built for the unit under test, the 3968A Portable Tape Recorder. The purpose of this digital bus interface is to permit remote control of tape speeds and operating modes. With reference to bus capabilities, this interface implements talker/listener functions, source/acceptor handshake, and service request/serial poll capability (ability to indicate to the controller the end-of-tape and to respond with a status character). This interface is now a product option.

PROGRAMMING AND SYSTEM OPERATION

Once interface bus compatible system elements have been selected, setup is relatively convenient. The dual-connector cables let instruments be arranged in a convenient stack. Each device is set to its unique system address by switches or jumpers. The next step is that of programming the system. The program tells when and how measurements are to be made and how the raw data points are to be processed for final presentation in plots and printouts. This same program, with minor changes, was used throughout the development of the instrument, beginning with bench testing of the recorder in the lab, continuing with the environmental testing of the recorder, and ending with the production line testing for which the system is now used.

Program Structure

Figure 6 shows the overall flowchart for the recorder testing system used. This flowchart states in words the steps to be followed. The program has two major branches, either to the direct channel measurements or to the FM measurements. It also has two subroutines to perform the S/N and linearity tests, and a plotting subroutine.

The program starts with a setup section where the operator tells the program details of the test to be performed, such as channel number and channel type. For either branch, the next section sets the points-per-decade resolution for each selected region and keeps track of the plots. The measurement subroutines are entered to take data.

Figure 7 shows the flowcharts for (A) Plotting, (B) Signal-to-Noise, and (C) Linearity. The Plotting Subroutine accepts a specified starting frequency, frequency resolution, and total number of points from the main program. This subroutine also has a delay to allow time for the tape to traverse the distance from the record head to the reproduce head before telling the digital voltmeter to take a reading. For efficient operation, the subroutine avoids taking unwanted data beyond band edge by sensing the point where output falls more than 12 dB below midband value.

In all cases, the starting location of each plot has been adjusted as necessary to make an actual measured frequency coincide with the specified band edge frequency at a given speed.

The program used during the environmental test is identical to the program discussed here with the exception that entering the channel number automatically specifies whether the channel is an FM or direct channel since one unique mixture of channels existed in the units environmentally tested.

The signal-to-noise subroutine uses a criterion for valid readings requiring that the maximum noise reading be no greater than 6 dB above the minimum, a criterion determined by examining the behavior of a number of recorders operating properly. One of the advantages of an automatic system is that consistent tests for rejecting bad data are made, as operator judgment can very considerably from one operator to another. This advantage was found to be especially important to the recorder testing system. Another advantage is the ability to apply correction factors to data. An algorithm is included to compensate for the bandwidth difference between the average responding voltmeter and a true RMS voltmeter referred to earlier in the section, "Signal-to-Noise Ratio Measurement". This algorithm also examines the statistical distribution of the accepted readings to make sure that the result is still Gaussian, an indication that the procedure is indeed valid.

The linearity subroutine requires an operator to be available to change the manual switches that apply the dc input to the FM channel.

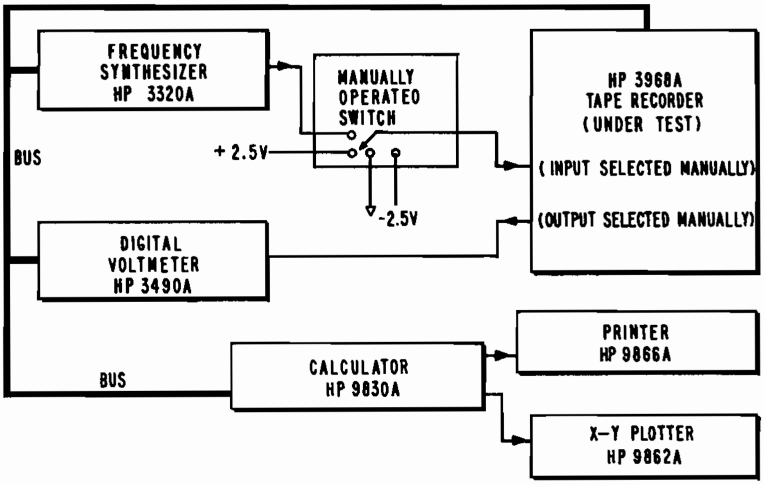


FIGURE 4. BLOCK DIAGRAM OF THE SEMI-AUTOMATIC SYSTEM INTERCONNECTED WITH THE IEEE-488 STANDARD BUS.

FIGURE 5. SEMI-AUTOMATIC SYSTEM FOR PRODUCTION AND ENVIRONMENTAL TESTING OF TAPE RECORDERS. THIS PORTABLE AND COMPACT SYSTEM INCLUDES A CALCULATOR AND PRINTER; A FREQUENCY SOURCE; A DIGITAL MULTIMETER; AND A PLOTTER. UNIT UNDER TEST (LEFT FOREGROUND) IS AN INSTRUMENTATION TAPE RECORDER.

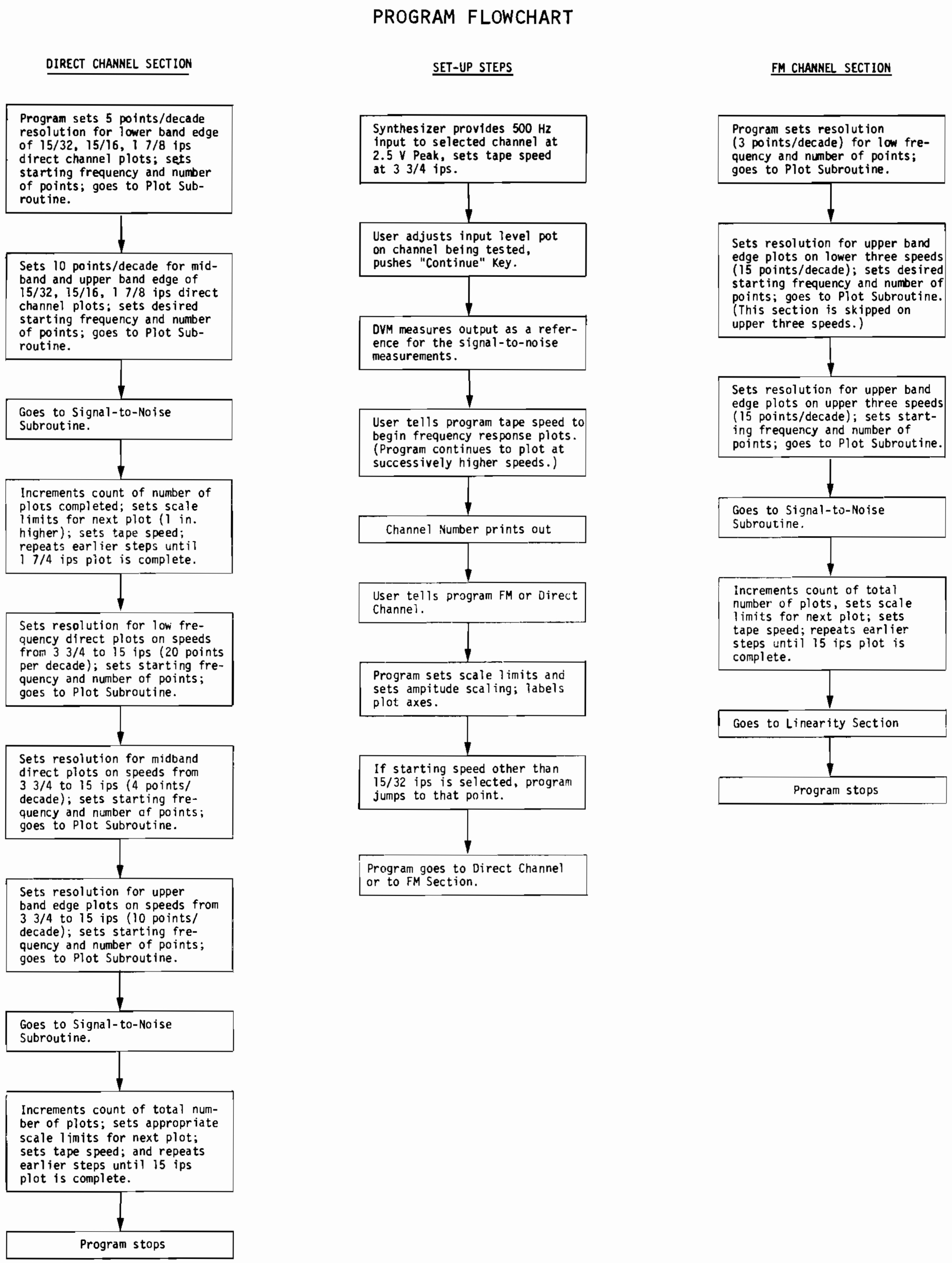


FIGURE 6. OVERVIEW FLOWCHART FOR THE SYSTEM PROGRAM.

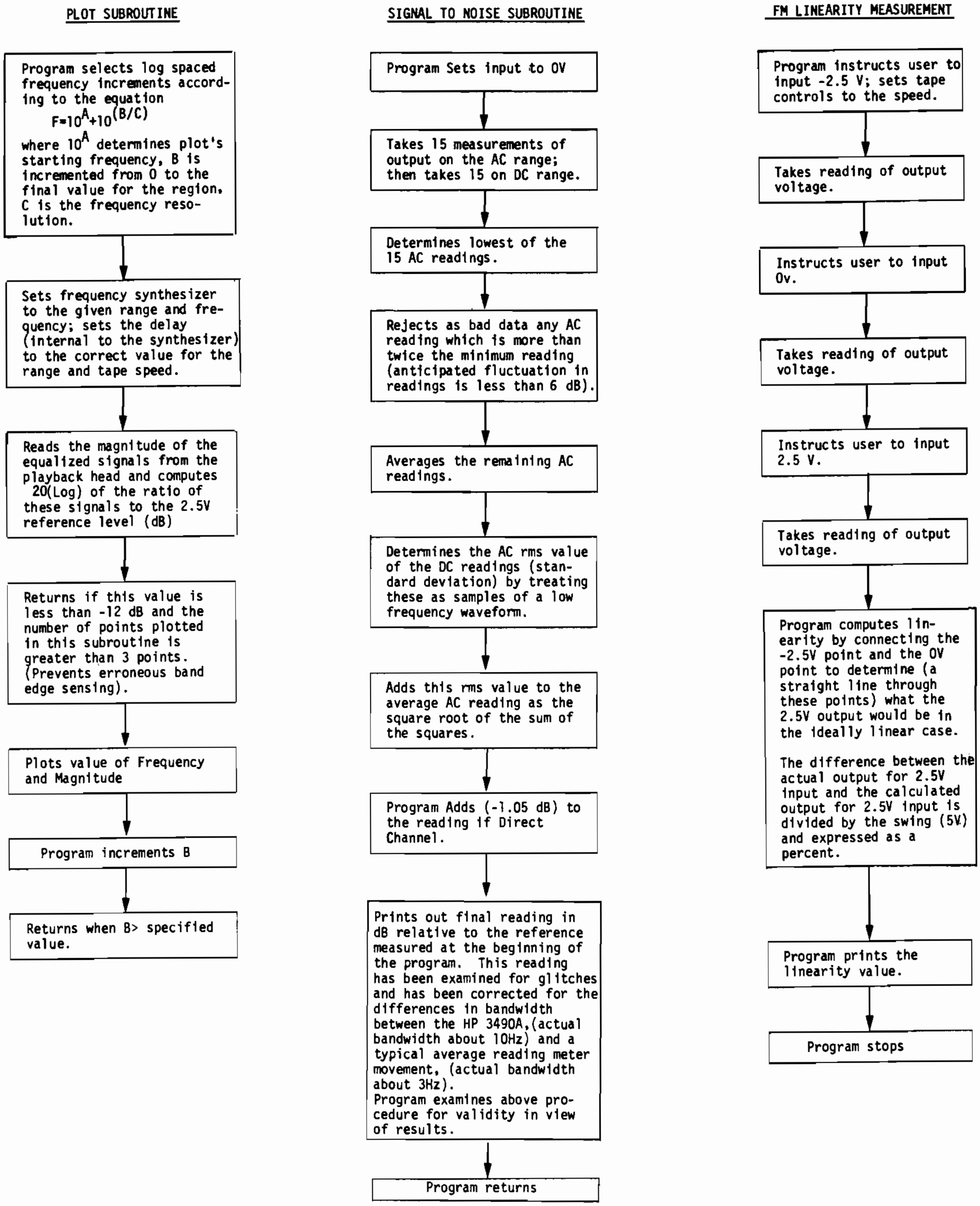


FIGURE 7. FLOWCHARTS FOR (A) PLOTTING SUBROUTINE, (B) SIGNAL-TO-NOISE MEASUREMENT SUBROUTINE, AND (C) FM LINEARITY MEASUREMENT SUBROUTINE.

System Operation

This system is semi-automatic in the sense that an operator attaches leads to the unit under test, loads the program, keys in the details of the desired measurement, and returns to the equipment to switch dc voltage inputs at the appropriate time. Since to further automate a system requires a heavier outlay for hardware and especially for software, a conservative decision is to approach the ultimate solution in steps, as was done here.

A programmable calculator does not make heavy specialized demands on the operator. The easy-to-learn keyboard has been found a benefit in making it possible for the test technician to contribute program improvements even though he may not have had prior programming experience. For example, the test technician who operates the recorder testing system was a mechanical assembler prior to his exposure to the system.

In operation, this system has made possible more thorough testing because the system speed lets measurements be made at less effort and in a shorter time. It takes about 10 minutes to plot one family of curves. Running the complete series of tests on a recorder takes about 90 minutes. The operator needs to be available once every ten minutes or so to reposition the input and output test leads in preparation for measuring a new channel.

Increased Automation

The system shown in Figure 4 could be further automated by the addition of two programmable devices: (1) a Model 3495A Scanner to switch multiple connections to the unit under test, and (2) a Model 59303A Digital to Analog Converter to apply the dc voltages necessary for the linearity test. These additions, possible because an interface bus system is easily expandable, would automate fully the measurement of linearity and would provide automatic switching from channel to channel. Data could then be recorded with a Model 9866A Printer or a Model 9862A X-Y Plotter as desired.

CONCLUSIONS

The test system for tape recorders has proven itself in the production area. Based on its success, two further applications of a Model 9830 Calculator have already been made. One uses the calculator as a tape controller for mean-time-between-failure tests; the other uses it to control a 3968A Recorder Mainframe adapted for "running in" tape transports.

Even though the semi-automatic recorder test system does not represent a full utilization of the potential in the controller and the interface, this system saves about 1-1/2 hours of test time per instrument. Where benchtop instruments would be required anyway for manual testing the incremental cost of an interface bus system is not great. Since this system has been used in the design lab, in environmental testing, and in production, the cost can be spread over a large number of applications. It is estimated that the system would pay for the cost of its acquisition in one year at a use level only one-third that contemplated.

REFERENCES

(1) IEEE-488-1975* "Standard Digital Interface for Programmable Instrumentation", April 4, 1975. Available from the Institute of Electrical and Electronics Engineers, Inc., 345 E. 47th Street, New York, N.Y. 10017.

(2) Ricci, D. W. and G. E. Nelson, "Standard Instrument Interface Simplifies Design", ELECTRONICS, November 14, 1974, 95-106.

(3) Loughry, D. C., "A New Instrument Interface: Needs and Progress Toward a Standard", ISA TRANSACTIONS, Vol. 14 (3) 1975, 225-230.

(4) Oliver, B. M., "Some Effects of Waveform on VTVM Readings", Hewlett-Packard Journal, April, 1955, Vol. 6 (8, 9, 10).

* American National Standard (ANSI MC1.1-1975)

Reprinted through the courtesy of MSN MICROWAVE SYSTEMS NEWS
JUNE/JULY 1976

Test Equipment: Mea$uring It$ Worth for Your Company

Want to know how to make a sound financial decision on your next instrument purchase? Just read this incisive analysis and, with the help of a pocket calculator, you can reach your own conclusions as to which item of test equipment should be bought for the laboratory or production line.

Brian G. Larson
Hewlett-Packard/Santa Rosa Division
Santa Rosa, CA 95404

As the capability, complexity and cost of test equipment climbs, engineers find the purchase justification process taking on new (and somewhat foreign) dimensions. No longer does the decision hinge solely on technical requirements and constraints as they relate to price. Instead, the accountants and financial specialists are much more in the act with their accompanying "business school investment analysis" syndrome. But take heart, engineers! You can present the fiscal analyses as persuasively and effectively as your technical evaluations. This article will attempt to remove much of the eerie mystique surrounding investment analyses that often frightens the engineer.

Electronic test equipment prices have traditionally ranged from a few hundred to a few thousand dollars per unit. In future years there will be a continuing upward trend in product prices due both to increasing user benefits and inflation. Many of those familiar with lower cost investments become uneasy when faced with large expenditures on single instruments and systems. Some negative first impressions have been: "That's more than I make in a whole year!," or, "I don't believe that price!!," or "My boss would never approve that expenditure!." Good investment alternatives can die of engineer frustration. After all, an engineer is a trained technical specialist, not a financial analyst, and who wants to risk more than a year's salary anyway? A $5000 bad investment is easier to recover from than a $50,000 fiasco.

There is a great deal more to making good investments than getting comfortable with the price tag. Large dollar purchases need to be justified both technically and financially. By applying a few techniques correctly, an engineer can calculate the return from a proposed investment to prove whether or not an idea appears financially sound.

Several different methods are being used by industry to evaluate the quality of potential investments. The most popular techniques are: a) PAYBACK PERIOD; b) AVERAGE RETURN ON INVESTMENT; and c) DISCOUNTED CASH FLOWS. Ideally, we should use investment worth indicators that assist us to make profitable investments. Unfortunately, the different investment worth indicators can lead to conflicting conclusions and poor investments. After eliminating the indicators that can prove faulty conclusions, focus of this piece will be on discounted cash flows, and an example investment analysis.

Investment Indicators. PAYBACK PERIOD determines the length of time required for the net cash savings (after taxes) generated by an investment to equal the initial investment. Consider the two equal investments of $10,000 with their net annual cash savings (see Figure 1). We may have investment opportunities that appear to be equally valuable, but which, in fact, are not.

CASH FLOWS

Years

	0	1	2	3	PAYBACK PERIOD
(A)	[$10,000]	$5,000	$3,000	$2,000	3 Years
(B)	[$10,000]	$2,000	$3,000	$5,000	3 Years

Figure 1.

According to the definition of payback period, we could accept either investment because both have equal payback periods of three years. However, there are two serious limitations with this analysis. First, product (A) returns more cash earlier in the investment cycle than Product (B). Cash can be reinvested in new opportunities only as soon as it is received. It would be very difficult to find a loan agency not concerned about "when" an outstanding loan is to be repaid. (The principle of the time-value of money will become clearer in the discounted cash flows discussion.) Second, most companies want to have information about their investment after the initial costs have been recovered. The payback method provides little useful information about overall profitability. But the indicator is valuable to the company having cash flow (liquidity) problems, looking for short-term investments of only one or two years.[1]

AVERAGE RETURN ON INVESTMENT eliminates one of the shortcomings of the payback method by considering the cash flows over the entire life of the investment. Figure 2 expands the previous example to include all net cash flows (after taxes) beyond the payoff of the initial investment.

Years

	0	1	2	3	4	5	CASH INFLOWS
(A)	[$10,000]	$5,000	$3,000	$2,000	$3,000	$2,000	$15,000
(B)	[$10,000]	$2,000	$3,000	$5,000	$1,000	$4,000	$15,000

Figure 2.

We will calculate the average annual returns based on the initial investments of $10,000. Note (in Figure 2) that the five-year cash inflows are $15,000 for each investment, or an average of $3,000 annually.

$$\text{AVG. RETURN} = \frac{\text{Avg. Annual Savings}}{\text{Initial Investment}} = \frac{\$3{,}000}{\$10{,}000} = 30\%$$

Identical average returns of 30% have failed to disclose significant differences in the timing of the cash flows, and the fact that Investment (A) is preferred to Investment (B). Corporations use a version of this same indicator called RETURN ON ASSETS to compare yearly net income with the book value of all their assets, thereby indicating the combined worth of all investments. For the purpose of comparing investment alternatives, use of the average return on investment indicator and the payback period should be left to investment specialists, who understand the several limitations.

Comparing the value of cash flows received today with cash flows to be received in the distant future is as reasonable as making comparisons between apples and oranges. Through the method of "discounting" we calculate today's value, or the present value of all future cash flows. We are then able to make valid comparisons of investment worth at a common point in time . . . now.

DISCOUNTED CASH FLOWS give a true picture of investment worth by taking into account the "timing" of the cash flows over the estimated life of the investment. To illustrate, consider an investment of funds into a savings account offering a compounded interest rate for some period(s) of time. At the end of period n we could expect to receive from the savings institution both the cash invested today (present value) and also interest on our investment. Stated mathematically:

$FV_n = PV(1 + i)^n$ or $PV = FV_n/(1 + i)^n$

where, FV_n = future value of investment in period n

PV = present value of investment

i = interest rate per period

n = time periods at "i" interest

If $1.00 is invested today (present value) at 6% annual interest, what is the future value in one year?

$$FV_1 = \$1\,(1 + .06)^1 = \$1.06$$

This simple example also points out that $1.06 to be received one year from now is worth only $1.00 to us today, if the available interest rate is 6%. Our equation states that we would be no worse off to have $1.00 today with an offer of 6% interest than we would be to have an offer of $1.06 one year from now. On this minor principle rests the theory of present value, and discounted rate of return.

Every company should have a profit objective, or minimum desired rate of return to be used as a standard against which all investment proposals are measured. Although several different opinions and titles exist to describe the minimum acceptable interest rate (discount rate, opportunity cost, etc.), we will use the cost of capital. The cost of capital rate selected by a company should be at least equal to the rate of interest of borrowed funds, but will likely be even higher. Accounting, or the finance group, will usually be able to provide information on the company cost of capital. However, the decision of where to get the funds to make the initial investment—whether borrowed, or resulting from the sale of stocks or bonds, etc.—is a *separate* decision for the finance department, and has little bearing in proving an investment good, or bad.

NET PRESENT VALUE (NPV) is the discounted cash flow method best designed to compare investment worth. NPV is the algebraic sum of the present values of all cash flows:

$$\text{NPV} = \sum_{k=0}^{n} PV_k = -\frac{\text{INIT INV}}{(1+i)^0} + \frac{FV_1}{(1+i)^1} + \ldots + \frac{FV_n}{(1+i)^n}$$

Because an example is usually clearer than a definition, we will apply the above NPV equation to the two previous examples. A cost of capital rate of 10% will be used (see Table I). Note that the initial investment has a negative sign being a cash out flow.

If for a given cost of capital the NPV is positive then by company standards the investment is acceptable. Both investments in Table I are acceptable, but Investment (A) is preferred because it has the largest NPV. According to our calculations, the future cash savings generated by Investment (A) could pay off a $10,000 loan having 10% interest, with a remainder of $1,818.

Having selected Investment (A) one can calculate the INTERNAL RATE OF RETURN (IRR)—one more indication of the investment's worth. IRR solves for the interest rate that will cause the NPV of the investment to be zero. The IRR represents the maximum interest that we would be willing to pay if we borrowed the initial investment funds, and still be no worse off financially. For Investment (A) the IRR equation is shown in Table II.

Note that solving for the interest rate in the equation is an iterative process. As a first step to solve for the IRR, we "guess" a value for "i" and substitute it in the equation. If the NPV is still positive, we continue to increase the value of "i" until the NPV is zero. At this point i = IRR. Decrease the size of "i" if the NPV of the cash flow is negative. A calculator having the function Y^x is an indispensable tool for discounting cash flows. An HP-65 programmable calculator is an ideal tool for investment analysis, especially for the time-consuming iterative equations. All the needed equations for making investment decisions are available on pre-programmed magnetic cards.

Table I

$$\text{(A)}\quad \text{NPV} = \$1818 = -\frac{\$10{,}000}{(1.10)^0} + \frac{5{,}000}{(1.10)^1} + \frac{3{,}000}{(1.10)^2} + \frac{2{,}000}{(1.10)^3} + \frac{3{,}000}{(1.10)^4} + \frac{2{,}000}{(1.10)^5}$$

$$\text{(B)}\quad \text{NPV} = \$1221 = -\frac{\$10{,}000}{(1.10)^0} + \frac{2{,}000}{(1.10)^1} + \frac{3{,}000}{(1.10)^2} + \frac{5{,}000}{(1.10)^3} + \frac{1{,}000}{(1.10)^4} + \frac{4{,}000}{(1.10)^5}$$

Table II

$$\text{NPV} = 0 = -\frac{\$10{,}000}{(1+i)^0} + \frac{5{,}000}{(1+i)^1} + \frac{3{,}000}{(1+i)^2} + \frac{2{,}000}{(1+i)^3} + \frac{3{,}000}{(1+i)^4} + \frac{2{,}000}{(1+i)^5}$$

IRR = i = 18.16% (HP-65 Calculator)

Example Cash Flow. Thus far, we have assumed the existence of a CASH FLOW CHART, showing inflows and outflows versus *time* see (Figure 3). For an investment analysis cash flow, we record only the cash changes that would result from actually making the investment. Sunk costs, or historical cash payments already committed by a company, should not affect the cash flows of a new investment alternative.[2]

SYSTEM PURCHASE CASH FLOW

	YEAR					
	0	1	2	3	4	5
INITIAL INV.	[$50,000]					
INV. CREDITS		3,333				
SELL OLD EQUIP	3,000					
EMPLOYEE TRAIN.	[500]					
MAINTENANCE		2,350	750	750	750	750
SALARY + OH		13,500	13,500	13,500	13,500	13,500
DEPRECIATION		10,000	6,000	3,600	2,160	740
SALVAGE						2,500
NET CASH FLOWS	[$47,500]	29,183	20,250	17,850	16,410	17,490

NPV = $23,003 (for i = 15%)
IRR = 36.75% After Taxes

Figure 3.

Previous investment examples (A, B) had physically separate cash flows. We were able to analyze them individually and then make comparisons. If a choice must be made between the best of two investments, much time can be saved by using an "incremental" cash flow that directly determines which investment is superior. The incremental cash flow is a chart showing only the cash differences between two investment alternatives. Most engineering decisions involving the purchase of new equipment ask the question: "Should I purchase newer technology, or continue to test my products using the same type of instruments now in the lab, and in production?" This example will use an incremental cash flow addressing the above question and will discuss the necessary ingredients and considerations of a good analysis.

Cash Flow Components. Paying required government taxes on profits is a legal obligation much like the bills for parts and materials. For a true picture of profitability, the return on investment analysis should include the effect of taxes. All example cash flow values will be calculated on an after-tax basis and a summary tax conversion table will be provided.

Determining the sales price or initial investment is as easy as getting a quote from the instrument sales representative. Let's assume that we desire to buy one of the new "smart" test instruments available on the market. Technically, our proposed controller-system will perform the measurements now requiring many different instruments and test set-ups. The improved speed of the system will allow us to test more products in less time, and do it more thoroughly. We need to show management that the resulting benefits and time savings will justify the expenditure of $50,000. On the cash flow chart of Figure 3 the initial investment of $50,000 is entered at time 0 when the expenditure actually occurs. (Cash outflows on the chart are bracketed [$]). Because the equipment being replaced has been purchased already, its initial cost represents a sunk cost, and therefore has no influence on the present cash flow.

Next, we must estimate the productive life span of a new investment, and the times when significant cash flows occur. Electronic test instruments have normal life spans of 7–10 years, and more, if designed and maintained properly. If an asset is generating net cash savings, the longer the usable life, the better the return on each invested dollar. However, for the sake of being ultra conservative in estimating a worst-case cash flow, this example will use a five-year asset life span. It's important to note when comparing investments, that all the alternatives should be compared over the *same life span* (which generally is the practice with electronic test equipment.)[3] (For discussion of unequal life spans, refer to Reference No. 3).

Investment Tax Credit is a tax relief mechanism established to encourage businesses to invest in certain kinds of assets like electronic instruments. If the asset useful life is:

- At least three but less than five years, 1/3 of the investment amount is eligible for the 7% investment credit.
- At least five but less than seven years, 2/3 of the investment amount is eligible for the 7% investment credit.
- Seven years or more, the full investment amount is eligible for the 7% investment credit.

During the periods of January 22, 1975 to December 31, 1976 the investment credit is 10%; this credit is deducted from the total tax due on the corporate tax return.

For our investment example having five plus years' life, we can expect a tax credit of 2/3 of 10% of $50,000, representing an after-tax savings of $3,333 for the investment. Investment credits are normally claimed in the first year and will be recorded as a cash inflow at the end of year 1 (see Figure 3).

If as the result of buying new equipment the old, fully-depreciated test gear can be disposed of for $6,000, this cash savings will also benefit the new equipment purchase. Your first reaction may be: "But the old equipment is a sunk cost, and shouldn't be included". The fact that the new equipment purchase causes the sale of the old gear, causing cash to flow in *now*, makes the cash savings a valid one. However, we are not allowed to keep the entire sum of $6,000 because the Internal Revenue Service (IRS) will want a share. The amount of the sale income that we pay to the the IRS is equal to the TR X Sale Price and the remainder (1-TR) X Sale Price can be placed in the company treasury. Our investment has a net after-tax benefit of $3,000 (assuming TR=50%) attributable to the sale of old equipment. This savings is claimed at time 0 when we dispose of the old gear.

One way to analyze labor savings is to perform a reasonable estimate of "measurement times" for individual products comparing the newer instruments with the old. Every hour saved by the "smart" instruments will create a cash saving. Conservatively estimating that our new controller system can perform the work of two manual test stations and operators, net savings will be the company cost of one technician. We should view the investment opportunity as a means of eliminating some of the frequent hunts for good technicians. To quantify the labor savings we must add the salary and the technician overhead (OH) cost. Electronics industry overhead rates average between 200 and 600%, depending on job function. One should be able to get this OH number from the accounting department. The total loaded labor rate is equal to (1 + OH) X Wage. If we pay $4.50 per hour and our OH rate is 200%, then our actual loaded labor cost is (1 + 2.0) X $4.50, or $13.50 per hour. A full-time year of 2,000 hours saved at $13.50 per hour will provide savings of $27,000. After-tax labor savings, using the factor (1-TR), will add $13,500 per annum to our cash inflows, when compared with the old way of testing.

Salvage is the estimated market value remaining at the end of the asset life. A fair value for test equipment salvage is 10% of list price after five years and 5% of list price at the end of seven years. The proposed investment of $50,000 will have a salvage value of $5,000, with after-tax savings of $5,000 X (1-TR) = $2,500 at the end of the fifth year (see Figure 3).

Depreciation is that part of an aging asset that we charge against product sales each year, thereby protecting part of our income from taxation. Note that the value of depreciation is subtracted from the same periods income *before* taxes are calculated. The net result is equivalent to an actual cash savings, because without depreciation we would have been required to pay more in taxes. It is therefore logical that the greater the value of depreciation write-off, the greater the tax savings. Depreciation creates an after-tax cash inflow equal to Depreciation X TR.

As already mentioned, a dollar received today, if wisely invested, can be worth more than a dollar received tomorrow. If our objective is to maximize the return on each invested dollar, then we would do well to adopt the depreciation method that would provide the earliest return of tax savings. The professional motive of a good tax accountant is to *legally* avoid the payment of as much tax as possible. Of the several acceptable depreciation methods available, three are most common.

STRAIGHT LINE—This depreciation method provides for equal yearly increments of write-off over the life of the investment.

Yearly Depreciation = D/n
D = Depreciable base (asset price minus salvage)
n = Asset Life

Consider a $50,000 capital investment with $5,000 salvage value at the end of a five-year life.

$$\text{Yearly Depreciation} = \frac{\$50{,}000 - \$5{,}000}{5} = \$9{,}000/\text{yr.}$$

Straight Line Depreciation Schedule

	D		TR		AFTER TAX
Year 1	$9,000	X	.5	=	$4,500
2	9,000	X	.5	=	4,500
3	9,000	X	.5	=	4,500
4	9,000	X	.5	=	4,500
5	9,000	X	.5	=	4,500
	$45,000				

In spite of the simplicity and ease of use, straight line depreciation is not used as frequently as "other" methods of depreciation for investment analysis, and has been presented mainly for reader awareness.

SUM-OF-THE-YEARS-DIGITS (SOYD)—This is an accelerated depreciation method formulated to write off a larger fraction of dollars in the early years of the investment. Tax savings are returned soon in the investment, giving us a significant advantage over straight line depreciation. A depreciation schedule using the SOYD method is calculated in the following manner.

$$S = \sum_{K=1}^{n} K = 1 + 2 + 3 + \ldots + n = \frac{n\,(n+1)}{2}$$

S = denominator of SOYD fraction
n = investment life
D = depreciable base (asset price minus salvage)
TR = tax rate

YEAR 1	n/S	X	D	X	TR	=	After-Tax Dep'n
2	(n-1)/S	X	D	X	TR	=	After-Tax Dep'n
3	(n-2)/S	X	D	X	TR	=	After-Tax Dep'n
.	.		.		.		
.	.		.		.		
.	.		.		.		
n	1/S	X	D	X	TR	=	After-Tax Dep'n

For the same $50,000 investment with five-year life, and $5,000 salvage value, the SOYD schedule is:

$$S = \frac{5\,(5+1)}{2} = 15$$

	Factor		D		TR		After Tax
YEAR 1	5/15	X	$45,000	X	.5	=	$7,500
2	4/15	X	45,000	X	.5	=	6,000
3	3/15	X	45,000	X	.5	=	4,500
4	2/15	X	45,000	X	.5	=	3,000
5	1/15	X	45,000	X	.5	=	1,500

DOUBLE-DECLINING-BALANCE (DDB)—Also sometimes called twice-straight-line declining-balance, DDB is another example of accelerated depreciation. One unique difference of the DDB method is that we ignore the salvage value for the first few years of the depreciation schedule, but do not allow the undepreciated asset value to decrease below the estimated salvage. The annual depreciation is calculated by applying twice the straight-line rate (2/n) to the remaining undepreciated value of the asset.

Undepreciated Balance X 2/n X (TR) = After Tax Depreciation

n = asset life
TR = tax rate

For the same example as before:

DDB DEPRECIATION SCHEDULE

	Undep Bal		Factor				TR		After-Tax
YEAR 1	$50,000	X	2/5	=	20,000	X	.5	=	10,000
2	30,000	X	.4	=	12,000	X	.5	=	6,000
3	18,000	X	.4	=	7,200	X	.5	=	3,600
4	10,800	X	.4	=	4,320	X	.5	=	2,160
5	6,480-Salvage			=	1,480	X	.5	=	740

Notice in Year 5 that depreciation is limited to only $1,480 due to the fact that we are not allowed to depreciate below the salvage value of $5,000. Depending on the magnitude of salvage estimated, it is possible with DDB to fully depreciate an asset before the end of the investment life. Under normal circumstances DDB accelerates depreciation faster than SOYD and thus improves investment worth, but the proof of which depreciation type is best lies in trying each method in the cash flow. DDB was used in the example cash flow of Figure 3.

Another depreciation method, called DDB to straight line, is used frequently in industry and usually provides the fastest acceleration of tax savings. A "crossover" point must be calculated to provide the optimum tax return, and the time during the asset life when a switch occurs from DDB to straight line depreciation. Large companies will usually revert to SOYD or DDB rather than keeping track of hundreds of assets, each having a unique crossover point. Use of the HP-65 calculator will greatly simplify the evaluation of any of the above depreciation methods, including DDB to straight line.

Comparisons of yearly maintenance costs need to be made between the new system and the equivalent amount of test equipment being replaced. Reputable electronic instrument vendors maintain files of information on the average annual repair and maintenance costs on each instrument manufactured. This information is usually provided as a customer service, and is used also to establish the cost of instrument repair contracts between the vendor and the user. Companies operating in-house repair facilities will likely have files of actual instrument repair costs and could be a source of much of the needed information.

Smart instruments having microprocessors and memories are, in most cases, assumed to be costly and difficult to repair. This assumption may not be valid; especially when system support costs are compared with those of numerous individual instruments. Some smart instruments can diagnose many of their own problems, and can interact with the technician to speed the repair.

The research for our example showed that the old equipment being replaced was costing $5,000 annually to provide repair, preventive maintenance, and needed calibrations. Our $50,000 investment alternative would cost $3,500 for the same services, and would have a basic warranty of one year. Incrementally, the new system would save $1,500 per year over the maintenance costs of the older style equipment. We therefore record $1500 X (1-TR) = $750 in after-tax maintenance savings per year in our cash flow. Figure 3 shows $2,350 in maintenance savings at the end of Year 1. During the first year of our investment, the warranty is in effect, and we experience no repair costs on the new system. Calibration charges, however, will cost $300 during Year 1, but these are offset by the $5,000 that would have been spent on the old instruments (the difference is equal to $2,350 after taxes).

Many ingredients that we would like to include in our cash flows are difficult to quantify. Products that are more thoroughly tested to better accuracies ought to be worth something to the customer. Reducing instrument operator errors by going automatic should be worth something. But short of having a crystal ball, one must be satisfied to use these kinds of benefits verbally to enhance the investment.

CASH EXPENSES that will result from the investment decision also need to be recorded on the cash flow chart. Only those expenses necessary for the successful operation of the system should be recorded. One such example might be special employee training if it is desired to maintain and calibrate the system in-house. Cash outflows may result from employees being trained to write the needed software for the system. We have estimated total employee training expenditures at $1,000. Expenses are also deducted from the income before taxes are calculated, causing a reduction in the tax obligation. The after-tax remainder of the bill that we pay in cash is Expense X(1-TR). So our employee training expenses are reduced to $500 after taxes.

A tax table has been provided (Table III) as an aid to convert pre-tax cash flows to after-tax cash flows. We first identify the nature of the cash flow, such as depreciation. The table tells us to multiply by (TR) to obtain the after-tax depreciation. We are also told that depreciation is a cash inflow.

Table III.

PRE-TAX	MULTIPLY BY	AFTER-TAX
Purchase Price	1	[Cash Outflow]
Savings	(1-TR)	Cash Inflow
Expenses	(1-TR)	[Cash Outflow]
Salvage	(1-TR)	Cash Inflow
Depreciation	TR	Cash Inflow

Having identified all the elements in the cash flow of Figure 3 we are ready to draw the important conclusions. The NET CASH FLOWS on the chart represent the algebraic sum of the cash flows at the end of each time period. Using this information we are able to calculate the NPV of $23,003, assuming our company cost of capital is 15%, and the IRR of 36.75% after taxes. If we are satisfied with investments that return at least 15% after taxes, then this new investment is clearly superior to the old method of testing.

The RISK that future projected cash flows will not materialize can be handled without the use of complex statistical models that are usually more cumbersome than the investment decision warrants. A successful technique is to estimate cash flows based on three projected levels of business: a) PESSIMISTIC, or worst-case; b) TYPICAL, or most likely; and c) OPTIMISTIC. The extra effort required to change basic assumptions and make the required calculations is minimal. Obviously, if an investment looks good using worst-case estimates, then the risk would be low for the "most likely" cash flows. Comparing the IRR with the cost of capital provides us with another built-in risk indicator. The fact that the actual investment return of 36.75% (IRR of Figure 3) greatly exceeds the 15% cost of capital rate should indicate a large "safety zone" for our cash estimates. The system investment opportunity looks excellent.

INFLATION is an ever present influence in the world's economy and yet it appears that we have ignored inflation's effects in our cash flows. The greatest inflationary pressure will be felt on future cash flows like salaries, overhead and maintenance. Remember that Figure 3 is an incremental cash flow comparing a new "smart" instrument with older technology equipment. If future technician labor and overhead costs increase due to inflation, then our future cash savings will *increase*, because the system is still doing the work of two technicians. Inflation will also cause future expenses to increase. If we apply the same inflation rate to the future cash expenses and cash savings of a good investment, the NPV and the IRR can only improve. To say otherwise is to assume a condition where future cash expenses exceed cash savings causing the investment to be rejected at the cash flow stage, never being a good investment alternative.

Another very useful technique using discounted cash flows is to work in reverse by calculating the yearly labor savings required to obtain a desired rate of return. Suppose that we desire to buy this smart instrument for the R & D lab or the calibration lab. All the cash flows in Figure 3 remain the same except for Salary + OH and this yearly number will be replaced with the variable "L" representing labor savings.

Years					
0	1	2	3	4	5
[$47,500]	15,683+L	6,750+L	4,350+L	2,910+L	3,990+L

If our company cost-of-captial is 15% (i = IRR = 15%) then we can solve for "L".

$$0 = -\frac{47{,}500}{(1.15)^0} + \frac{15{,}683+L}{(1.15)^1} + \frac{6{,}750+L}{(1.15)^2} + \frac{4{,}350+L}{(1.15)^3} + \frac{2{,}910+L}{(1.15)^4} + \frac{3{,}990+L}{(1.15)^5}$$

L = $6,642 annual labor savings (after-tax)

The pre-tax savings need to justify this investment are $6642/(1-TR) = $13,284. An experienced engineer would earn $10 per hour, and with 200% overhead the company cost is (1+2.0) X $10 = $30 per hour. Converting the pre-tax savings to equivalent hours worked by an engineer: $13,284/$30 = 443 hours annually, 37 hours monthly and less than 2 hours daily. If all the engineers using the $50,000 instrument system can save a total of 2 hours daily, the investment return will exceed 15% after taxes.

Summary. Discounted cash flow calculations may seem a little tedious, but they are straight-forward. And the conclusions are valid! Using these powerful techniques to correctly analyze and justify a new investment opportunity can be a satisfying challenge to the engineer for it will help ensure that the engineer's organization will make a profitable decision.■

References

1. Robert W. Johnson, *Financial Management*, 3rd Edition, pp 183–184.
2. Charles T. Horngren, *Accounting for Management Control*, 2nd Edition, pp 394–396.
3. Harold Bierman, Jr. & Seymour Smidt, *The Capital Budgeting Decision*, 3rd Edition, pp 76–87.

HEWLETT hp PACKARD

Sales and service from 172 offices in 65 countries.
1501 Page Mill Road, Palo Alto, California 94304

JANUARY 1975

HEWLETT-PACKARD JOURNAL

The Hewlett-Packard Interface Bus: Current Perspectives

First announced over two years ago, the Hewlett-Packard Interface Bus has undergone refinements that make it suitable as a model for a proposed international standard.

by Donald C. Loughry

THE GOAL OF AN INSTRUMENTATION system is to monitor some process or perform measurements on a device. An interface system, a means toward this goal, provides the essential communications link between the components of the system. No single interface method is a panacea for all of the world's interface requirements and the HP Interface Bus system is no exception, but it does fulfill major needs for a wide range of calculator and computer controlled instrumentation systems.

The HP Interface Bus is a definition of an inter-device connection scheme that is optimized for programmable bench instruments. It is applicable as well to other components essential in instrumentation systems.

An interface system definition has three primary elements: mechanical specifications (cables, connectors, etc.), electrical specifications (voltage and current levels for transfer of signals), and functional specifications (a precise definition of all the signal lines, the protocol and timing relationships for using the lines, and the repertoire of messages that may be exchanged).

A fourth interface system element involves operational characteristics and specifications. These tend to be device-dependent characteristics, such as specific programming codes unique to each instrument, and perhaps system-dependent characteristics such as the application software. The primary focus of an interface definition, however, is away from device-dependent characteristics and toward mechanical, electrical, and functional specifications that are device and system independent. This approach leads to a feasible and broadly useful interface system definition.

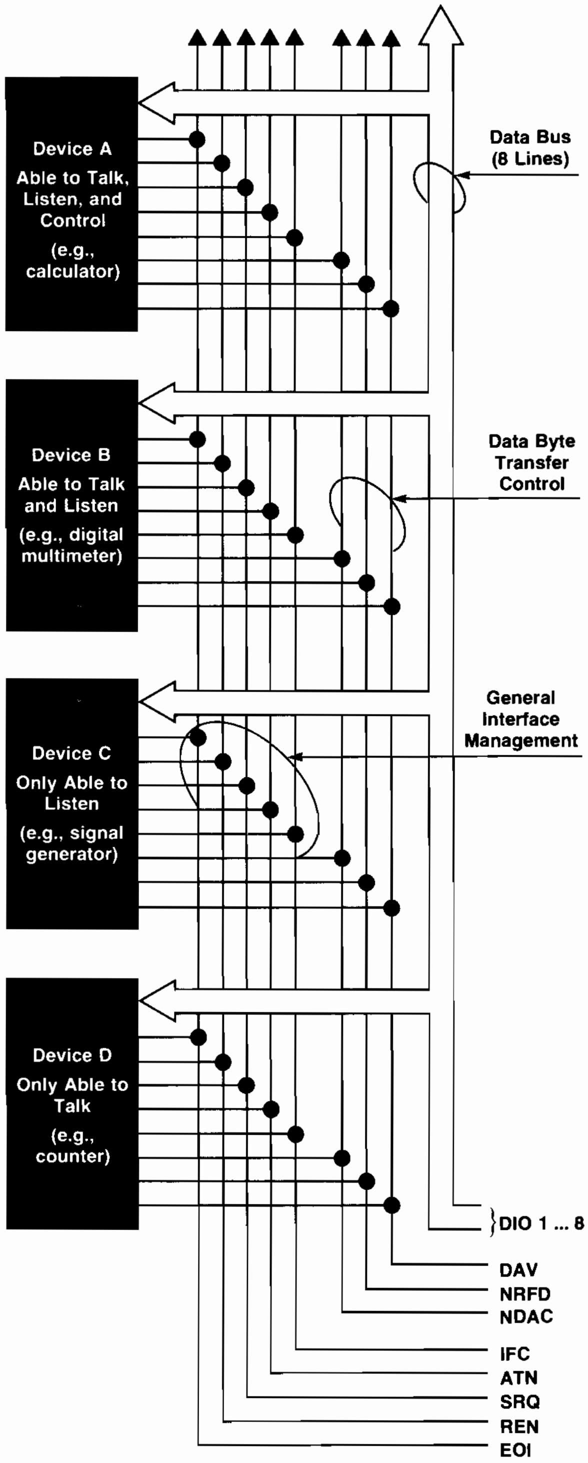
The HP Interface Bus is outlined briefly in the box on the opposite page. It is a byte-serial, bit-parallel, partyline bus structure organized to provide communication among a group of up to fifteen instruments and system components. Messages are exchanged over the bus asynchronously and flow both to and from a given instrument.

Cover: *As more devices are designed to work with the HP Interface Bus, it becomes easier to assemble low-cost systems, as HP product manager Jane Evans is doing here. Jane (BS Chem, BSEE, MBA) has been bringing ten years of HP instrumentation and data systems experience to bear on use of the HP Interface Bus in solving problems in automatic measurement systems.*

In this Issue:

Printed in U S A

The HP Interface Bus



The HP Interface Bus transfers data and commands between the components of an instrumentation system on 16 signal lines. The interface functions for each system component are performed within the component so only passive cabling is needed to connect the system. The cables connect all instruments, controllers, and other components of the system in parallel to the signal lines.

Eight of the lines (DIO1-DIO8) are reserved for the transfer of data and other messages in a byte-serial, bit-parallel manner. Data and message transfer is asynchronous, coordinated by the three handshake lines (DAV, NRFD, NDAC). The other five lines are for control of bus activity.

Devices connected to the bus may be talkers, listeners, or controllers. The controller dictates the role of each of the other devices by setting the ATN (attention) line true and sending talk or listen addresses on the data lines (D101-D108). Addresses are set into each device at the time of system configuration either by switches built into the device or by jumpers on a PC board. While the ATN line is true, all devices must listen to the data lines. When the ATN line is false, only devices that have been addressed will actively send or receive data. All others ignore the data lines.

Several listeners can be active simultaneously but only one talker can be active at a time. Whenever a talk address is put on the data lines (while ATN is true), all other talkers will be automatically unaddressed.

Information is transmitted on the data lines under sequential control of the three handshake lines. No step in the sequence can be initiated until the previous step is completed. Information transfer can proceed as fast as devices can respond, but no faster than allowed by the slowest device presently addressed as active. This permits several devices to receive the same message byte concurrently.

The ATN line is one of the five control lines. When ATN is true, addresses and universal commands are transmitted on only seven of the data lines using the ASCII code*. When ATN is false, any code of 8 bits or less understood by both talker and listener(s) may be used.

The other control lines are IFC, REN, SRQ, EOI.

IFC (interface clear) places the interface system in a known quiescent state.

REN (remote enable) is used with other coded messages to select either local or remote control of each device.

Any active device can set the SRQ (service request) line true. This indicates to the controller that some device on the bus wants attention, say a counter that has just completed a time-interval measurement and wants to transmit the reading to a printer.

EOI (end or identify) is used by a device to indicate the end of a multiple-byte transfer sequence. When a controller sets both the ATN and EOI lines true, each device capable of a parallel poll indicates its current status on the DIO line assigned to it.

In the interest of cost-effectiveness it is not necessary for every device to be capable of responding to all the lines. Each can be designed to respond only to those lines that are pertinent to its function on the bus.

*American Standard Code for Information Interchange

To ensure a high degree of compatibility among products that are independently designed and manufactured at HP in widely scattered locations, the HP Interface Bus goes much farther than previous interface definitions in its scope and content. It provides Hewlett-Packard design engineers with the tools needed to interconnect a wide range of products from which systems can be configured with a minimal amount of additional engineering. Although it facilitates the assembly of systems, it does not guarantee the assembly of "instant" systems. Configuring a complete operating system demands detailed attention to all the device-dependent characteristics beyond the scope of the HP Interface Bus definition.

What's New Since '72

The HP Interface Bus was first described in the October 1972 issue of the Hewlett-Packard Journal. Since that time, several aspects of the interface definition have been refined to make it more useful without compromising the original objectives. Typical of these refinements are the following areas of change:

- Signal line name changes made in response to the needs of international standardization, e.g. the MRE (multiple response enable) line is now called ATN (attention).
- Address extension (optional) to two bytes to permit a maximum of 961 talk and 961 listen addresses rather than 31 each.
- Physical extension of the maximum total transmission path to 20 meters, rather than 15.
- Enhancement of the service request protocol.
- Refinement of the parallel-poll capability.
- Addition of the EOI (end or identify) signal line.
- Addition of more addressed commands, e.g. DEVICE CLEAR.
- Specification of control shift capabilities.
- Remote-local protocol upgrade.

It is not the purpose of this article to describe these changes in detail but to alert the reader to the nature of the changes. In general, the bus structure remains basically as originally described. (More details about the bus are included in the operating manuals of some bus-compatible products).

Relationship to Proposed Standards

Interest in an international interface standard applicable to programmable measuring apparatus has grown substantially during the past few years, paralleling the growing need for instrumentation systems In addition, there is an increasing desire to configure these systems from products made by different manufacturers. European organizations, particularly in Germany, were instrumental in initiating the standardization effort.

In mid-1972, Hewlett-Packard began to participate in various national and international standardization bodies to help develop a suitable interface standard. After initial goals were established by the U.S. Advisory Committee, the techniques used by the HP Interface Bus were adopted as an appropriate starting point for a draft document. An initial draft was written, evaluated by the Committee, and submitted as the U.S. proposal to an IEC (International Electrotechnical Commission) Working Group in the fall of 1972. Since then, the interface definition has undergone a number of minor changes to accommodate various needs at the international level.

In September 1974, the parent technical committee IEC TC66 approved the latest draft document for a formal ballot among the member nations of the IEC. The final results of the ballot will not be known until the end of 1975. Concurrently, a similar draft document is being evaluated as a potential IEEE Standard. The present definition of the Hewlett-Packard Interface Bus is compatible with the current IEC and IEEE draft documents.

It would be presumptuous for the Hewlett-Packard Company to forecast the eventual outcome of the draft document ballot, but it is worth pointing out that the widespread interest in this particular interface system outside of HP suggests that it satisfies many interface needs, that it simplifies the interface challenge for designers, manufacturers, and users alike, and that it does make a significant contribution toward providing more versatile and lower cost instrumentation systems.

HP Interface Bus Specification Summary

Interconnected Devices:	Up to 15 maximum on one contiguous bus.
Interconnection Path:	Star or linear bus network up to 20 meters total transmission path length.
Signal Lines:	Sixteen active total; 8 data lines and 8 lines for critical control and status messages.
Message Transfer Scheme:	Byte-serial, bit-parallel, asynchronous data transfer using interlocked three-wire handshake technique.
Maximum Data Rate:	One megabyte per second over limited distances; 250-500 kilobytes per second typical over full transmission path.
Address Capability:	Primary addresses, 31 Talk and 31 Listen; secondary (2-byte) addresses, 961 Talk and 961 Listen. There can be a maximum of 1 Talker and up to 14 Listeners at a time.
Control Shift:	In systems with more than one controller, only one can be active at a time. The currently active controller can pass control to one of the others. Only the controller designated as system controller can assume control.
Interface Circuits:	Driver and Receiver circuits TTL-compatible.

Putting Together Instrumentation Systems at Minimum Cost

Instrumentation systems that do useful work can be assembled around the HP Interface Bus at costs in the $15k to $25k range. Here is an approach to assembling such systems with a minimum amount of engineering time.

by David W. Ricci and Peter S. Stone

INSTRUMENTATION SYSTEMS CAN NOW be applied to a wide range of applications where system solutions were previously not justifiable on an economic basis. This is the result of recent developments that are making systems easier to design, build, and use, and thus cost-effective for many small-scale, low-volume measurements

In this article, we would like to discuss some of the techniques of assembling instrumentation systems based on the HP Interface Bus for use in various kinds of measurements. Some of the tasks for which these systems are particularly well suited are: (1) multiple or often-repeated measurements; (2) measurements needing real-time data reduction and/or decision making; (3) stimulus-response measurements; and (4) measurements requiring repeatability and accuracy.

Why a System

The question, "do I need a system" has no clear-cut answer but must be based on an engineering evaluation of benefits versus costs. There are many benefits in using a system rather than a manual operation, some of which are:

- More consistent results in repeated measurements—a system is not subject to operator fatigue.
- Greater throughput because systems are generally faster.
- More thorough testing because system speed allows many more parameters to be measured in a shorter time.
- Results expressed in appropriate units since many systems controllers are capable of on-line data manipulation. A measurement of a thermistor's resistance, for example, may be converted directly to temperature.
- Greater accuracy; system errors can be measured automatically, stored, and accounted for in results.
- "Adaptive" data acquisition; a system can be programmed to branch to other measurements to help pinpoint the problem when it senses an abnormal condition.

The principal reason for *not* using a system is cost, not only the cost of the individual instruments used, but also the cost of special hardware needed, such as test fixtures, and the cost of preparing and debugging the software. It is possible that a thorough investigation of alternative ways of doing a job may point to an approach that can do the job reasonably well at less cost than a system. A system is really only a tool with which to solve a problem and regardless of how powerful the tool may be, it is nevertheless advantageous to select the right tool for the job.

A New System Technology

The engineering costs of putting a system together, however, have been reduced significantly by three recent developments: (1) the HP Interface Bus; (2) the growing number of "smart" instruments with internal microprocessors; and (3) the advent of highly agile, "friendly" controllers that have a high degree of operator interaction.

The HP Interface Bus (HP-IB) has been the prime energizer in making systems a more attractive alternative. Its direct impact has been to simplify or eliminate many of the steps involved in system design and implementation. Its indirect influence has been as a catalyst during the design of new instruments to make them more useful in systems applications—more thought now goes into the design of a laboratory bench instrument in terms of its potential for systems applications. It has also sparked the development of a number of useful system accessories (see page 12).

The new "smart" instruments make it easier to apply the accuracy and versatility of the lab bench instrument to a system environment. Previously, lab instruments were seldom adaptable to systems work so special purpose system components of limited ability had to be used. The wider use of digital techniques made feasible by advancing semiconductor technology has made it easier to include the interface functions within an instrument so it can work more effectively in a system. It has also given the lab bench instrument the ability to process data and execute more complicated measurement algorithms, thus relieving some of the burden placed on the controller and the interfacing. It is not necessary to program each discrete action for these instruments—the interface now needs to handle only processed data and programming instructions that occur at a relatively low rate.

Another component contributing to the rise of the new system technology is the highly agile programmable calculator. These provide a "friendly" controller, useful for simple to moderately complex systems, with a high degree of operator interaction that greatly simplifies program generation and debugging.

Another benefit of the HP-IB capability is that a minisystem can be assembled for a one-time test and, once the test is performed, disassembled to allow return of the components to normal bench use—it is not necessary to have a lot of hardware sitting around idle between systems. These "one-shot" systems are usually put together in the engineering lab to evaluate or characterize certain devices and usually use a calculator as the controller because of the ease of developing programs. Engineers who have had experience in assembling these systems are able to plan, configure, write programs for and get results from a new system in only two to five days.

The interface bus also makes it easier to service a system. Operation of each device can be verified by testing it alone with the controller; the others are removed simply by disconnecting the cables.

Recreatable Automatic Systems for the Lab

In the course of doing their own work, HP engineers have assembled a number of minisystems to quickly perform tests in the lab that otherwise would require considerable time to do manually. These are systems that are assembled around the HP Interface Bus for specific tests and then disassembled or reconfigured for others. Because of their wide applicability, many of these "one-shot" systems have been documented in a series of application notes (series 174-0) to help others assemble similar systems with minimum loss of time.

Typical of these is a system for measuring the transfer characteristic of voltage-controlled oscillators. It uses the Model 59303A Digital-to-Analog Converter (page 12) to derive a dc control voltage for the oscillator in response to calculator commands, a counter to measure the oscillator frequency, and a plotter to trace graphs of voltage versus frequency.

After the program is keyed into the calculator, all the engineer has to do is enter the voltage range and the voltage step size and then press RUN PROGRAM. The system then traces a plot in about 10 seconds.

The application note (AN 174-1) describes the equipment needed and how it is connected, and it gives the program listing. Other application notes in the series describe measurements of non-linearity in VCO's, short-term stability of oscillators, FM peak-to-peak deviation, and determination of probability densities.

First Steps

Although the advent of intelligent instruments and controllers on a standardized interface has brought the systems approach within the reach of a much wider range of users and applications, it has not altered the fundamental process of designing and building a system. All the considerations that go into building a system still exist and must be evaluated. We propose the following procedure:

1. Define the problem
2. Select the instruments
3. Select the controller
4. Interface the devices
5. Integrate the system
6. Write utility software
7. Write applications software
8. Document the system.

Although this appears to be a step-by-step procedure, the design process is not that orderly—there is a good deal of iteration back and forth between steps. The reader should also be cautioned that this is not a magic formula that guarantees instant success. Use of the HP Interface Bus has not eliminated all the pitfalls—it just makes it easier to cope with them. This list merely outlines the considerations that must be evaluated.

Defining the Problem

Defining the problem in terms of the results to be achieved is the most critical step. Without a clear defination at this point, it is difficult to make good decisions throughout the process, and subsequent effort may be wasted in backtracking. In fact, one cannot really determine whether or not a system approach is the best way until a precise statement of requirements has been established.

An instrumentation system can be thought of as an instrument in its own right, but the measurements it performs are generally broader and more complex than any single instrument can perform. Unlike a purchased instrument, however, the instrument system is partially designed by the ultimate user. The system designer must thoroughly understand the measurements he is trying to make, the trade-offs involved, and the techniques required to get the desired information. As with instrument design, failure to recognize the real needs will often get the system designer locked into a technique that may not be the best one for solving the problem. One of the biggest pitfalls in system design is trying to fit the solution to the problem, rather than the other way around.

How does one go about defining the problem? The best way is to make clear statements of the results to be achieved. This should ultimately result in a fairly detailed list of basic requirements and system features, stated as objectives. The trick here is to state the objectives in terms of results so a particular solution is not automatically indicated.

Objectives should be classified into two groups: Musts and Wants. Musts distinguish those requirements that are absolutely necessary; Wants are those that are desirable, but expendable. Want objectives may be weighted in importance to help in making trade-offs.

After the list of must and want objectives is established, a decision can be made as to whether or not a system represents the optimum approach. A system is indicated if it can do the job better, faster, more accurately, more economically or a combination of these. Other factors to consider are consistency of results and the need for data reduction.

Selecting the Instruments

Selection of the proper instruments is not very difficult since measurement needs are generally quite specific once the problem is well defined. Care must be taken, however, to insure interfacing compatibility. Although the HP-IB solves the mechanical, electrical, and functional compatibility problems of interfacing, there are various operational differences. Each instrument generally has a different set of programming commands and/or data output formats. Failure to understand the syntax needed for each instrument can sometimes cause readings to be taken at the wrong time, cause a controller to interpret data incorrectly, prevent instruments from triggering when they're expected to, and so on.

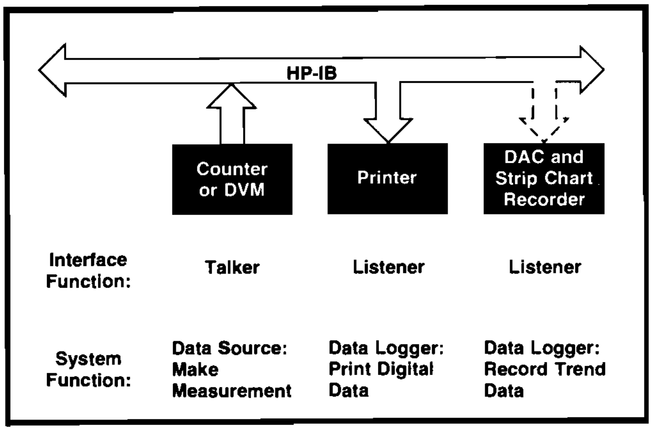


Fig. 1. *Elemental systems has one measurement device supplying data to one or more recording devices.*

At some point, the system designer may be faced with making a choice between using a bench instrument, either already designed for systems use or adaptable to it through various bus-compatible accessories such as code or D-to-A converters, or using a system component (a stimulus or measurement module designed specifically for systems use but not usable outside a systems environment). Although a systems component may be more cost-effective, it often lacks the high-performance capabilities of the lab-bench instrument. Besides, the lab-bench instrument is capable of manual operation as well as remote control, useful in debugging a new system and in system maintenance. Manual operation is also useful for diagnosing problems in a unit under test that is found to be faulty.

Selecting the Controller

Traditionally, a system controller is a device that controls all the other devices in the system, performing such tasks as programming instrument modes, collecting and processing data, and so on. The HP-IB, however, defines a controller as the device that manages the operation and flow of data on the bus, a subset of the operations performed by a system controller. It is important to understand that an HP-IB controller does not necessarily program instruments or process data, although it may perform these functions also.

The various levels of controller complexity that are possible with the HP-IB are shown in the drawings. An important feature of the HP-IB is that the controller can be chosen independently of the instruments, enabling instruments assembled into a system at one level of system complexity to be operated at another level without changing the interface (provided the instruments' capabilities are adequate for a higher level).

The simplest configuration is shown in Fig. 1. This has a single measurement device, such as a counter or a DVM, outputting its data to a printer and, optionally, to a strip-chart recorder. The measurement device is operated in the "talk only" mode while the printer and/or strip-chart recorder are in the "listen only" mode. This means that whenever the measuring device places data on the bus, the listeners accept it without being addressed. There is no separate controller—the measurement rate is established by the measuring device.

The next level of complexity is shown in Fig. 2. Here a scanner functions as a simple controller. It addresses the talkers one at a time in sequence, and each transmits its data to the printer.

Both of the above examples are concerned primarily with data logging. No programming information is placed on the bus, so instrument functions and ranges must be set manually.

By replacing the scanner with a more complex controller that can address devices to talk and listen and that can send programming codes, the same collection of instruments can perform a wider variety of tasks, especially with the addition of a stimulus instrument. The higher-level programming can be done with a card or tape reader as the controller.

Once arriving at a stimulus/response situation,

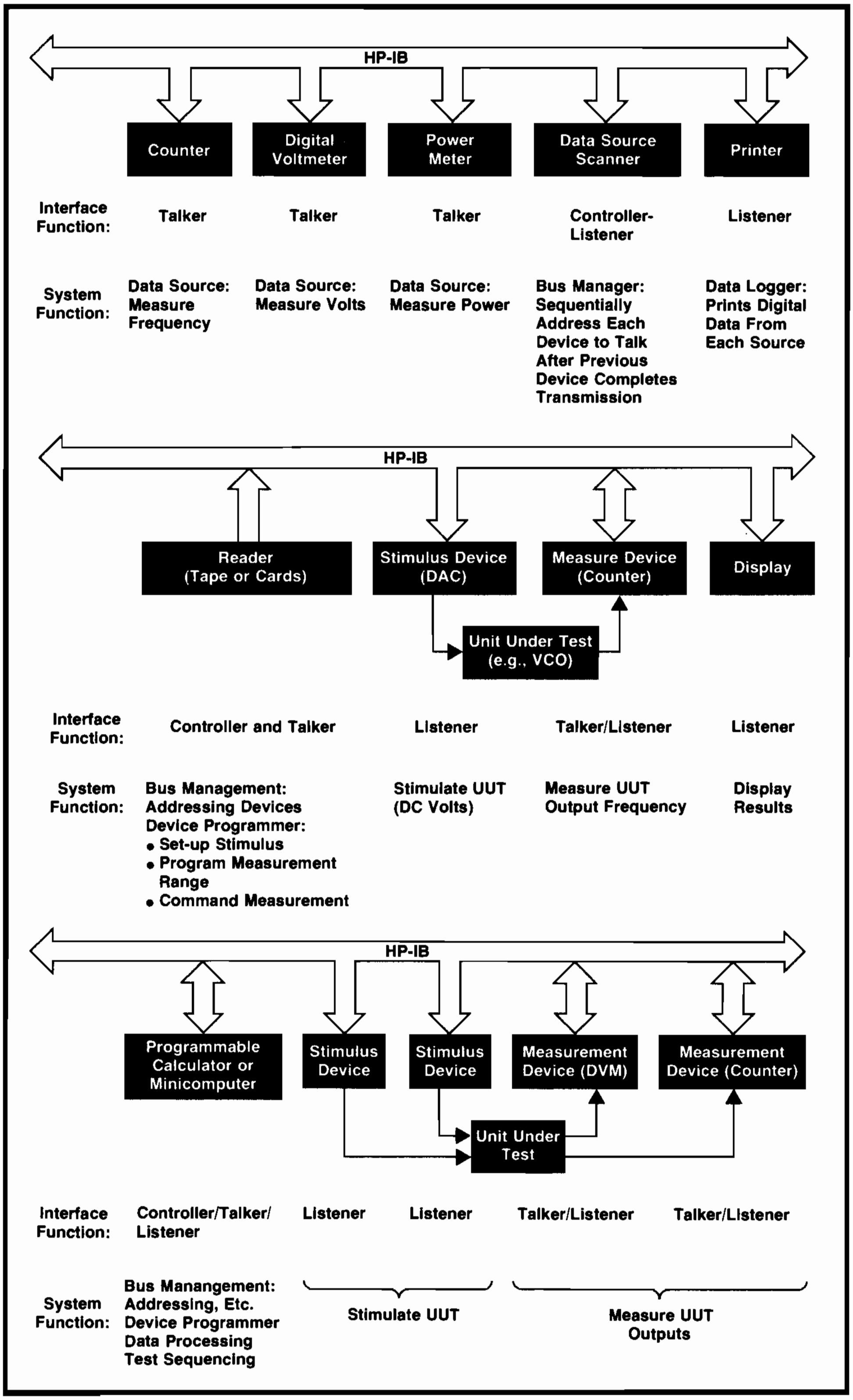


Fig. 2. *Addition of a scanner enables several measurement devices to take turns supplying data to a recording device.*
Fig. 3. *A tape or card reader as a controller enables ranges and functions to be programmed.*
Fig. 4. *With a calculator or computer as a controller, automatic date manipulation and decision-making can be included in the test program.*

Developing a One-of-a-Kind Automatic Test System

One of the first systems built around the HP Interface Bus was a production test system for the HP Model 5340A Microwave Counter.

The goal was to shorten test time. During the early production of this instrument it became apparent that the limiting factor in the quantity that could be produced was the capacity of the test station, several hours being required to test each instrument. The most economical way to break this bottleneck, it was decided, would be to automate the procedure.

The problem could be stated very simply: to completely verify the operation of the Model 5340A and check all its specifications it must be tested at many frequencies between 10 Hz and 18 GHz at several power levels between −45 dBm and +10 dBm, and in all operating modes. This was a "must" objective. A "want" objective was to make the test unattended so the test technician could spend his time troubleshooting units that failed the test.

Selecting the instruments for this system posed a problem. At that time, there were no HP-IB-compatible signal sources that operated above 1.3 GHz. So, for the higher frequencies HP 8690-series sweepers were used, monitored with a 5340A Counter to give precise frequency information. A programmable attenuator and a 432C Power Meter provided control of the power level.

The lower range of frequencies was covered by a 3320A Synthesizer for the 10 Hz-to-13 MHz range and an 8660A Synthesizer for the 13-to-1300 MHz range, both HP-IB-compatible.

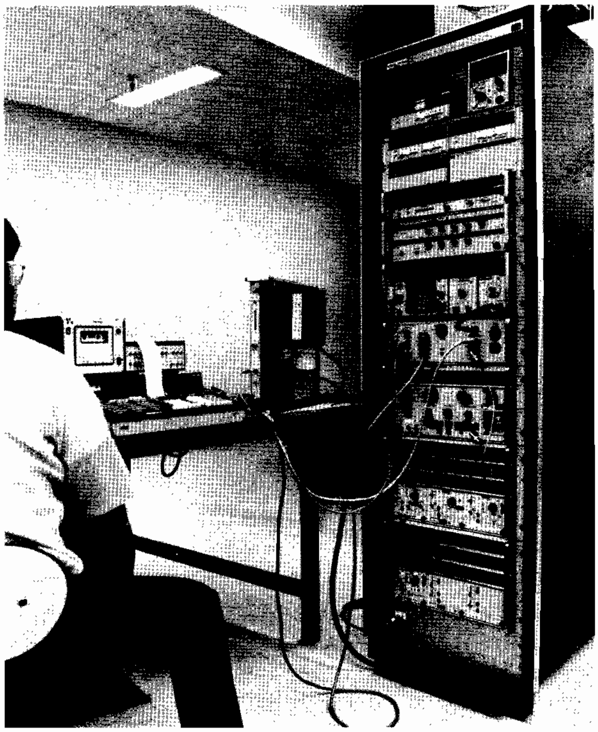
Selection of a controller was dictated by the need for programmability and the need for some computation capability, such as finding the logarithm of power readings and comparing readings to determine whether or not a reading is within tolerance. The Model 9820A Calculator was selected for this task but the desire to make the system run unattended required the addition of a tape cassette memory to accommodate long programs (the Model 9821A with its built-in cassette memory was not available at the time).

There was no problem, of course, interfacing the HP-IB-compatible instruments. The programmable attenuator was easily interfaced by way of the Model 59306A Relay Actuator (page 12).

Interfacing the non-HP-IB instruments, however, required some effort. This was accomplished by use of Model 59301A ASCII-to-Parallel Converters (page 12) controlling versions of the sweepers and power meter that had been adapted for computer control through a BCD interface. Additional circuits had to be designed, however, to match the logic levels of the sweeper interface to those of the ASCII-to-Parallel Converter.

Integrating the system largely involved careful consideration of how to run the RF cables to minimize VSWR and resultant losses.

There was no problem with the utility software since all the instruments were controlled through the interface bus one way or another. Writing the applications software was straightforward. It involved test strategy and some diagnostic programs that prepared failure reports to help the test technician locate troubles in the instrument under test.



After the system was assembled, debugged, and running, another "want" objective came to light. This was a desire to give the test technician some indication that a test was completed or otherwise stopped. Because the system did not use all six relays in the Model 59306A Relay Actuator, one was available for ringing a chime. A program loop was written to ring the chime five times whenever the test program was stopped.

A complete test of the Model 5340A Counter is now completed within 40 minutes, and this is an unattended test.

Because of the success of this test system, there was little argument about whether or not to use an automatic test system for the next high-performance instrument to come along, the Model 5345A. Testing requirements for this instrument were similar to the 5340A, but the frequency range goes only to 500 MHz. Thus, all but one of the instruments needed were bus-compatible (the exception was a pulse generator, but since it would be used in one mode only it did not require programmability). As a result, the system was assembled and up and running in less than 20% of the time that had been taken for the 5340A test system.

The 5345A test system was developed by Tom Coates. Al Foster designed the automatic test system for the 5340A.

some sort of feedback to the operator may be needed to indicate whether or not test results are within specified limits. Such a situation is diagrammed in Fig. 3.

As soon as this level of complexity is reached, however, there will likely be other requirements for data manipulation or automatic decision making. Thus, the use of a programmable calculator or a computer as the controller is indicated, as shown in Fig. 4. Here, all the potential advantages of programming instruments and accessories can be brought to bear on the problem.

The question of which level of controller to use is largely answered by the level of decision making and data manipulation required. The decision to use a lower-level controller is not a binding one, however, since a system can be upgraded to a higher level, assuming that the other system components have the necessary capabilities. Upgrading is simplified by the fact that no changes in the interface are required to do so.

The choice between a calculator and a computer is not so straightforward. For systems of simple to medium complexity we, as design engineers, have found the programmable calculator to be a powerful controller that is especially easy to use for program generation and editing. Where a great deal of on-line storage may be needed, a computer is indicated. The computer also offers more flexibility in terms of language and software operating systems, and it offers the potential for higher speed.

Interfacing the Devices

Interfacing used to be the major problem in assembling a system—each device required a separate piece of interface hardware and, very often, a separate software driver as well.

Now, if all the components selected for a system are compatible with the HP Interface Bus, the hardware interfacing is already done. Each instrument has its own I/O facilities for communicating on the bus, and they are linked together simply by connecting them with passive cables.

If a required instrument function is not available with the HP-IB , then one has to decide if it is possible to obtain the desired function by using some of the bus accessories (page 12) to drive a standard instrument. If that is not possible, then the engineering effort to interface the instrument must be evaluated. As a greater variety of instruments are developed for the HP-IB, this should become less and less of a problem.

Integrating the System

Integrating the system is simply a matter of assigning addresses to all the devices and connecting them with standard HP-IB cables. Again, if all the system components are bus-compatible, there is no real problem.

One aspect of system design that often is overlooked until too late is that of "fixturing"—connecting the system to the unit under test or to measurement points. This may involve the switching of low-level analog signals with resulting cross-talk and accuracy problems and can require extensive development time, especially in fully automatic systems. This problem is highly application dependent, so it is difficult to characterize generally other than it must be considered in the overall system design. Failure to do so may negate the effectiveness of the remainder of the effort.

Writing Utility Software

At this point, a means of controlling the communications to and from each device in the system must be developed. Usually, this is in the form of software. For convenience we have divided the software into two parts: the utility software, which is the instrument-dependent driver software for handling the I/O requirements of each device in the system, and the applications software which is concerned with the measurement algorithms. Applications software is largely device independent.

The utility software is greatly simplified by the nature of the HP-IB. The HP-IB addressing structure allows all bus instruments to share common driver routines thus reducing the amount of specialized software that needs to be written. Because the HP-IB is basically a communications structure that does not require an understanding of content to function, the utility software can be developed to provide the communications to and from a device without regard for the particular device's characteristics.

The simplest form of software for an HP-IB system would be that required of a card reader to manage the bus, i.e. send addresses and bus commands. Here, the binary code for each data line on the bus and for the ATN line must be marked on the card for each byte of information the card reader is to place on the bus. Where an HP 9800-series Calculator is to be used as the controller, this kind of detail is handled by a plug-in ROM block so the operator can control the devices through the higher-level language of the keyboard.

Writing the Applications Software

The availability of desk-top calculators with their readily grasped program generation, editing, and debugging techniques combined with the standard communication techniques used on the HP Interface Bus and system-oriented bench instruments makes the generation of application software much easier than it has been. But, if any one step in building a system could be called the most significant, writing the applications software is it! This details when and how measurements are to be taken and how the raw measurement data is to be processed.

Too often a system builder underestimates the extent of the effort required to achieve desired results—there can be a large discrepancy between having the capability to make a measurement and actually making it. In stimulus/response testing, for example, applications software is heavily involved in test strategies—where to start, what sequence of tests, how many tests are needed, tests limits and guard banding, failure analysis, troubleshooting aids, etc., etc. Even with a packaged system, software development is still required to solve specific problems the user has in mind because applications are too diverse to lump into one universal package.

For very simple systems, such as one using a data source scanner as the controller, applications software is practically non-existent as the user sets all ranges and functions manually. But as soon as the controller requires the specification of program steps, then the applications software assumes major proportions.

The system designer should keep in mind who will use the software. We have found that with small-scale production test systems, the use of a programmable calculator with its easy-to-learn keyboard makes it possible for the test technician who runs the test to contribute valuable software improvements even though he may not have had prior programming experience. If the software-user interactions are planned properly, the program can allow the test technician to decide what course of action to take when a failure is encountered in the device under test—for example, he may want to loop on the failure while he probes with an oscilloscope or he may simply want an error message printed and the test continued. The point is, good software should be flexible; it should allow the system user to select reasonable variations of the procedure depending upon the particular conditions encountered during a test.

Good system software should also take into account the mistakes a human operator is likely to make when inputting system information. Many systems require a fairly large number of input parameters before the actual test or measurement cycle begins. The entire system is much more usable if the software is written to allow the user to correct isolated input errors without having to restart the entire program. In general, it is much more advantageous to tailor the software to human characteristics than to try to train the human to cater to the software.

Documentation

Sometimes overlooked is the need to write down operating procedures so others who need to use the system can know what to do. System performance verification and servicing procedures also need to be worked out for maintaining the system. It often helps to also have verification procedures written for each individual device in the system as troubleshooting aids.

Often overlooked is the need to document the system configuration so it can be readily rebuilt if it should happen to be disassembled, or if the need should arise to duplicate it.

Packaged Calculator-Based Measurement Systems

Because one user's requirements are generally not quite the same as another's, most applications of automatic systems are unique in one way or another.

Nevertheless, HP engineers have been able to identify a number of applications that use the same hardware, and very often some common software. Because of the hardware commonality, equipment for a number of these applications is now offered in packaged systems that have all the hardware complete and tested as a unit.

These are not "turnkey" systems, however, in the sense that they can be put to work as soon as installed—software needs to be prepared for each particular application. The package, though, does include useful subroutines and examples of small applications programs that help in getting the system "on the air". Very often, these programs can be modified for particular purposes.

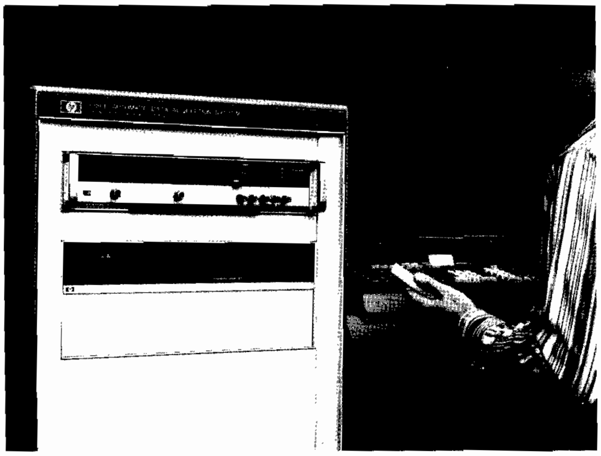
The Model 3050B Automatic Data Acquisition System is an example of the kind of calculator-based system offered. This system includes the Model 3490A Digital Multimeter (1-μV resolution, 120-dB system common-mode rejection), the Model 3495A Scanner (see page 17), one of three calculators (Models 9820A, 9821A, or 9830A), an equipment rack and cabling.

The basic version measures dc volts, ac volts, and resistance through as many as 40 channels. Besides logging data, it can do simple go/no-go limit testing. However, it can also process data, such as compensate readings for transducer nonlinearities, convert readings to engineering units, and do statistcal analyses (determine average values and standard deviations, and do trend analyses).

Many peripherals, such as a plotter and a timing generator, are available to broaden the system's capabilities. Because it uses the HP Interface Bus, it is easily expandable.

Other available calculator-based systems include the Model 3045A Automatic Spectrum Analyzer that performs spectral analysis, distortion analysis, and wave analysis over a frequency range of 10 Hz to 13 MHz with selectivity as fine as 3 Hz, and the Model 3042A Network Analyzer that measures phase response along with amplitude response over a 50 Hz to 13 MHz frequency range.

Typical prices in the United States for basic systems are $21,950 for a Model 3042A Network Analyzer, $22,400 for a Model 3045A Automatic Spectrum Analyzer, and $14,100 for a Model 3050B Data Acquisition System with ten low-thermal channels.



Filling in the Gaps—Modular Accessories for Instrument Systems

These programmable modules provide such accessory functions as remote display, switching, digital-to-analog conversion, and measurement pacing and timing. They are useful both with single instruments and as components of automated systems.

by Steven E. Schultz and Charles R. Trimble

EVERY DAY, ENGINEERS APPLY bench instruments such as counters, voltmeters, and synthesizers to the solutions of measurement problems. But very often something more than the basic instruments is needed to complete the job. This something more might be a signal switch, a digital-to-analog converter, a pacer, a relay actuator, or some other accessory.

This need for something more led to the development of a series of programmable modular accessories for instrument applications. The modules were designed with a dual purpose in mind: to work as accessories for stand-alone instruments, or as components of an HP Interface Bus-connected automated system where they fill the gaps in the system. A description of the modules presently available is given here to help the potential system builder envision how he might implement a solution to his measurement problem.

Digital-to-Analog Converter

The Model 59303A Digital-to-Analog Converter accepts up to 15 ASCII-coded digits serially, stores them, and on receipt of the line-feed character produces an analog voltage within a range of ±10V equivalent to three consecutive digits selected from the string. Among other uses it can be used to convert the digital output of a counter or a DVM to an analog voltage for driving a strip-chart recorder or an X-Y plotter.

The new D-to-A Converter converts any three consecutive digits within the received character string to the equivalent analog voltage with an accuracy of 0.1%. The relationship between the output and the three digits (D_1, D_2, D_3) is as follows:

$$\text{Analog output voltage} = (D_1 \times 1\text{V} + D_2 \times 0.1\text{V} + D_3 \times 0.01\text{V}) \times \text{the polarity sign}$$

Ignoring decimal points, the converter selects three digits in any data format from fixed point to scientific notation. When under remote control, it can be programmed to select the digits from either of two data words sent in the same character string. This ability is useful when the measuring instrument has two outputs, such as phase and amplitude, transmitted on the interface bus. One 59303A can convert the information in the first number for a plot of phase versus frequency, and a second 59303A can convert the second number for a plot of amplitude.

A block diagram is shown in Fig. 1. A "listen-only" mode, selected by a rear-panel switch, causes the D-to-A Converter to respond to all inputs. It may thus be used directly to convert the output of any bus-compatible "talking" instrument without use of a bus controller. In the "addressable" mode, it responds to inputs only when addressed to do so by the system controller.

Other uses for the D-to-A converter are as a programmable voltage source for testing other devices (it can sink or source up to 10 mA), to program trigger levels on counters, and to program analog-controlled devices such as sweepers and voltage-controlled oscillators. A precision frequency source can be obtained by using a calculator to set a number into a D-to-A converter controlling a VCO, and a counter to report the VCO frequency back to the calculator. The calculator compares the counter reading to the selected number and then adjusts the digital input to the D-to-A converter to minimize the difference. Only a few iterations are required to reduce the difference to the required low level.

Three output formats are provided: −9.99 to +9.99V, 0 to +9.99V ignoring the sign of the input data or 0 to +9.99V offset where a zero input produces 5.00V out. This last mode is useful for avoiding jumps from full scale to zero when plotting

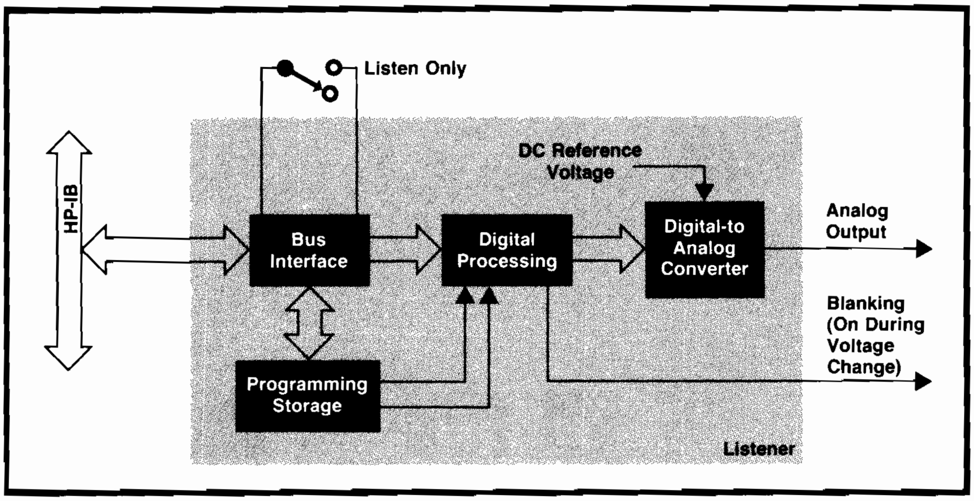


Fig. 1. *Model 59303A Digital-to-Analog Converter derives dc voltage equivalent to three consecutive digits out of a string received serially in ASCII code.*

bipolar inputs on a unipolar strip-chart recorder, as when plotting the output of a counter that is monitoring the long-term stability of a 10-MHz oscillator.

A Timing Family

Two modules—a timing generator and a clock—were developed for control of time functions.

The Model 59308A Timing Generator functions either as a digital delay generator (timer) or as a precision time marker generator (pacer). A block diagram is shown in Fig. 2.

As a timer, the Generator counts down crystal-controlled 1-MHz pulses and, following the receipt of an input trigger, generates an output pulse when the selected number of 1-μs time increments has elapsed. It is, in effect, a programmable one-shot with a range of 1×10^0 μs to 999×10^8 μs. This mode is useful for programming a delay, for example to allow a power supply to slew to a new voltage level before it is used in a measurement.

The time delay is programmed through the HP Interface Bus or, for local control, it is set on front-panel switches. In either case, three digits and a power-of-ten multiplier are entered ($\text{DDD} \times 10^{\text{D}}$) in units of microseconds.

In the pacer mode, the generator produces a pulse every Δt on and following the receipt of a trigger, where Δt is the number set into the front-panel switches or programmed through the HP Interface Bus. In this mode it may be used as a precision sample rate generator, say, to trigger a voltmeter reading every 10 seconds.

It may also be used for time interval measurements. It has an internal six-digit (decimal) counter that totals the number of output pulses produced since a trigger was received. It could be pro-

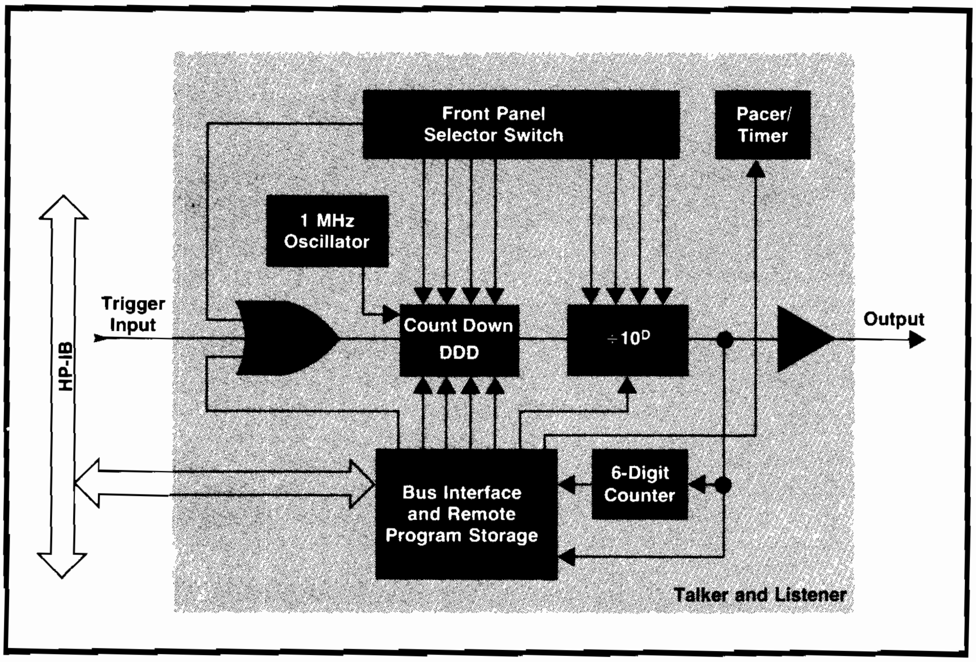


Fig. 2. *Model 59308A Timing Generator functions as a digital delay, a time mark generator, and a time-interval counter.*

A Quiet, HP-IB Compatible Printer that Listens to Both ASCII and BCD

by Hans-Jürg Nadig

The need for a low-cost, bus-compatible printer coupled with a desire for quieter operation has led to the development of a new printer, the Model 5150A (Fig. 1).

This printer is not limited to recording data in systems using the HP Interface Bus, however. It works as well with the older BCD instrument interface. It was designed with an internal data bus to give it a flexible, option-based architecture that provides the versatility needed for a wide variety of applications. The user can "design" his own printer according to his needs without paying for capabilities he doesn't need, and he can update it to a different configuration whenever he wishes.

Besides accepting the widely-used BCD data from a variety of instruments or ASCII-coded characters from the HP-IB or other ASCII-coded data sources, the new printer can also control the timing of data acquisition and direct the sequential acquisition of data from several sources. It can also print the time of day on the data record.

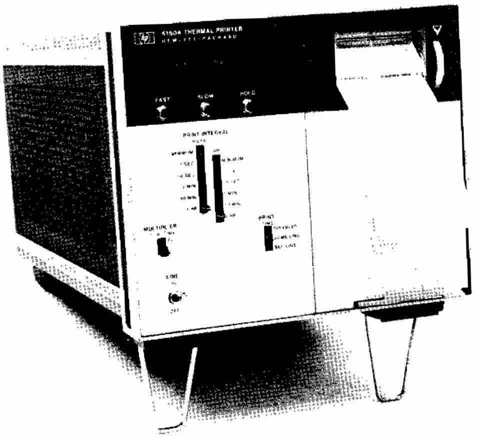


Fig. 1. *Model 5150A Thermal Printer.*

Plug-In Options

In the interest of keeping the printer as simple as possible, the basic instrument contains only the print mechanism, a power supply, and internal control circuits (Fig. 2). There are no functions in the basic unit that are not needed by all options. As a consequence, one of the plug-in interface options must be installed for the printer to become a functional unit.

Four options are presently available: (1) an ASCII interface for communicating on the HP Interface Bus; (2) a BCD parallel 10-column input (two can be installed for 20-column print-out); (3) a scanner for controlling the HP Interface Bus, for sequentially addressing a number of instruments and printing their readings; and (4) a digital clock for printing time with the data and for controlling the rate of data acquisition (data output of the clock to the interface bus is not provided). Three of the options are on plug-in circuit cards that include the necessary external connectors. The clock option is mounted behind the front panel and does not occupy either of the two plug-in slots.

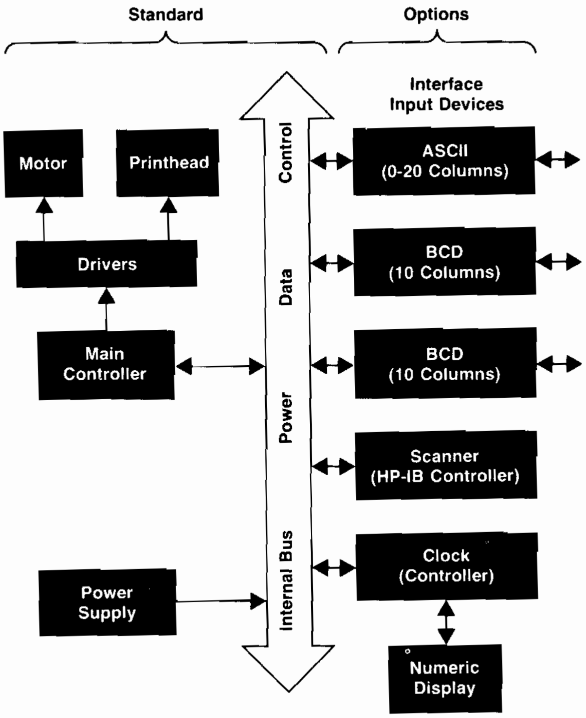


Fig. 2. *Basic organization of Thermal Printer.*

More Than a Printer

In its simplest application, the new Model 5150A can do as earlier printers have done; log data from an instrument. When used with the HP Interface Bus and operated in the "listen-only" mode, it accepts all data appearing on the bus. When it senses the LF command, it prints the most recent 20 characters (or fewer if less than 20 have been received since the last LF command). In the "addressable" mode, it accepts data only when addressed.

The combination of the clock, scanner, and HP-IB options places the printer beyond the realm of a simple data logger. The clock can initiate a scan at intervals selected on the front-panel DATA PRINT INTERVAL control. The scanner will then address the lowest numbered instrument on the bus, wait for the instrument to send its data, print that data, then address the next instrument. The time required to complete a scan thus depends on the instrument response times.

The clock can initiate scans at intervals shorter than one second and as long as two hours(if a scan is still in progress when a clock trigger occurs, the scanner ignores the trigger). It can thus be used for short-term tests in the R and D lab, the quality assurance lab, and the production test stand, or it can be used for unattended monitoring over long periods of time, as in industrial processes or checking conformance to FCC regulations. When the scanner is used without the clock, the scan cycle repeats continuously.

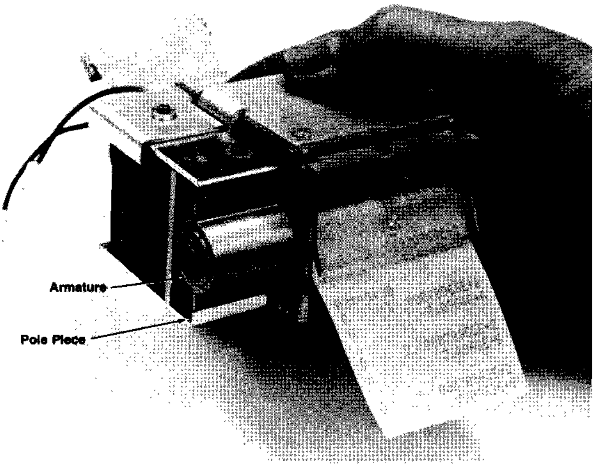


Fig. 3. *Print mechanism is uncomplicated.*

Few Moving Parts

The new printer uses the thermal-print technology developed for the HP-9800-series Calculators*. Alphanumeric characters are printed on a 5 × 7 dot matrix as the heat-sensitive paper is stepped past the thermal print-head. With the ASCII Interface, it prints characters from the ASCII 64-character upper case printing set. With the BCD interface, the printer has a repertoire of 16 characters, normally the digits 0 through 9, +, −, V, A, R, and blank. Any other set of 16 characters may be printed by changing ROM's within the instrument. It is even possible to configure the printer to print entire words in response to a single BCD input. It prints up to 20 columns at a maximum rate of 3 lines per second.

In the interest of achieving long-term reliability at minimum cost, the paper drive uses a very simple stepping motor. The motor armature is a cylinder with two winged projections (see (Fig. 3). Whenever the field magnet is pulsed, the wings are drawn into alignment with the pole pieces. Between pulses, a spring (hidden behind the armature in the photo) rotates the armature about 5° backwards against a stop. The resulting oscillating movement of the armature drives the paper-drive shaft through an overrunning clutch. A data line is written in 7 increments with 3 more provided for interline spacing.

Drive pulses occur at a 30-Hz rate and because the incremental movement is very small, motor operation is barely audible. The motor is the only mechanical motion in the printer—there are no print wheels, hammers, or inking systems.

The paper drive was developed by Ron Jensen, who also contributed a large portion of the total design. Product design was by Bill Anson and Keith Leslie.

*D.B. Barney and J.R. Drehle, "A Quiet, Low-Cost, High-Speed Line Printer", Hewlett-Packard Journal, May 1973.

grammed, for example, to generate a pulse every millisecond (100E1) so the stored count would give a reading of the number of milliseconds since a trigger occurred. The stored count is output to the interface bus by addressing the generator to talk.

The timing generator functions either as part of an interface-bus-connected system, receiving and sending trigger indications through the bus or through rear-panel connectors, or in stand-alone applications using the rear-panel trigger input and output connectors. It could be used, for example, to establish a precision data rate in a voltmeter-printer system that does not have a controller.

Two timing generators can be used as a programmable pulse generator. One establishes the repetition rate and serves as a trigger generator for the second one. The second is operated in the square-wave mode, in which it generates a rectangular pulse equal to one-half the time value set into the front-panel switches, thereby establishing pulse width.

The Clock

The Model 59309A ASCII Digital Clock gives absolute time in seconds, minutes, hours, days, and months. When connected to the HP interface bus and asked to talk, it outputs the time on the bus. A block diagram is shown in Fig. 3.

The Digital Clock is a precision instrument, using a 1-MHz quartz crystal resonator in its master oscillator. The aging rate of the crystal is 5 parts in 10^6 per year. The clock can also be driven by an external frequency standard of 1, 5, or 10 MHz.

It has other features that make it more than an ordinary digital clock. With a standard 9-volt battery installed for standby power, it becomes immune to powerline transients and it can operate on the battery for as long as a full day when there is a power-line interruption (the display will be turned off, however). It can operate with any other 8-10V dc power source through a rear-panel connector (it draws 2 mA at 8V with the display off). A companion unit, Model K10-59992 Standby Power Supply provides up to a year of standby power using size D flashlight cells.

Another useful feature is an internal memory that stores the time on command for later output. This would be used to store the time of a voltmeter reading at the instant the reading is taken, for later printout.

The clock can be set by codes sent on the HP interface bus. It is thus possible to create a subroutine that automatically sets the clock on system start-up. It can also be set manually with switches that are behind a front-panel lift-up door.

A Code Converter

Pre HP-IB instruments such as HP's 5050A/B Digital Recorder and 580A Digital-to-Analog Converter can be operated on the HP Interface Bus using the Model 59301A ASCII-to-Parallel Converter. This "interface to the interface" accepts ASCII-coded data from the HP interface bus and converts it to the BCD code used by these and many other instruments. It is thus possible to obtain a 20 line/second hard-copy log from an HP-IB system using the Model 5050B Printer with the ASCII-to-Parallel Converter.

ASCII-coded characters are first converted to the equivalent 4-bit BCD or hexadecimal equivalent, then fed to four 16-bit shift registers. The contents of the shift registers are output in parallel on two rear-panel 50-pin connectors. On receipt of the line-feed character, a "strobe" command is sent to the parallel interface to indicate that the data is complete. Up to 16 ASCII-coded characters can be handled per byte.

The converter has a "listen only" mode that enables its use in controllerless systems. In the "addressable" mode, it processes characters only when addressed by a system controller.

Supplemental Display

For display of data transported on the interface bus, the Model 59304A Numeric Display includes a memory for temporary storage and display of up to 12 digits and a decimal point. This module is useful for storage and display of intermediate results, for example so the program does not have to be slowed for display refresh in a calculator-based system.

The numeric display has a "listen only" mode so it can be used as a remote display for a measuring instrument. This is useful in RF or microwave systems where the actual measurement must be performed at an inconvenient place—it is much easier to transmit the data digitally on the interface bus then it is to reroute signal-carrying cables.

Automatic Control

The Model 59306A Relay Actuator has six form-C relays for control of equipment ranging from microwave switches to environmental chambers. The relays have both normally-open and normally-closed

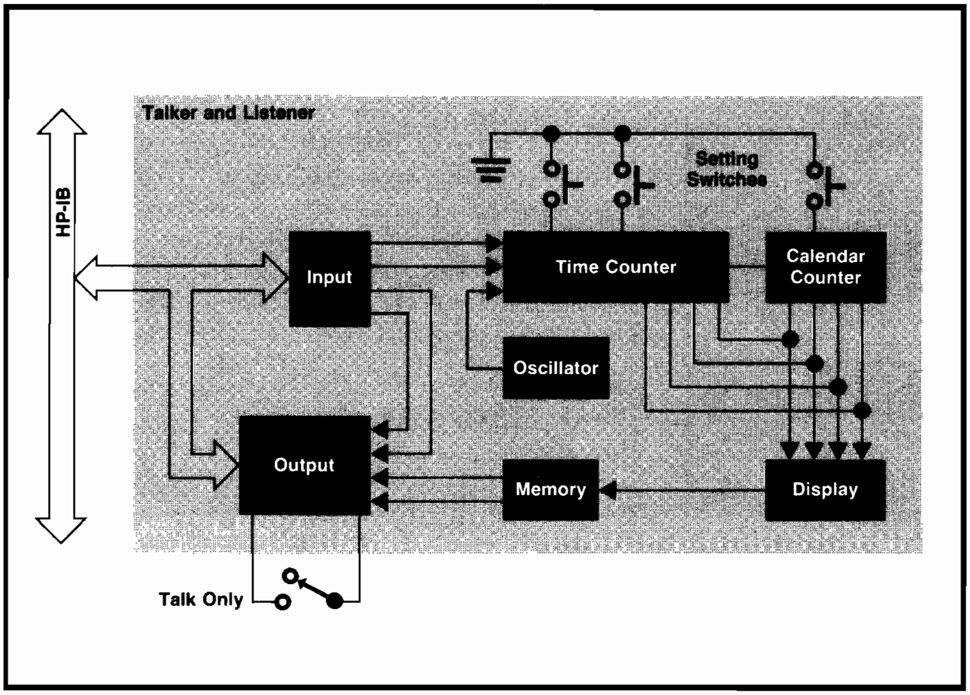


Fig. 3. *Model 59309A ASCII Digital Clock is settable by front-panel and remote control.*

contacts and can switch up to 25V at 0.5A.

Signal switching is performed by the Model 59307A VHF Switch. This module has two four-position, bidirectional switches, consisting of miniature relays that maintain a 50-ohm characteristic impedance for the signal path.

Designed to maintain fast pulse transition times, the VHF switch is useful for selecting trigger inputs

---

A Multifunction Scanner for Calculator-Based Data Acquisition Systems

by David L. Wolpert

In the usual data acquisition system, a scanner switches several inputs one by one to a single instrument for measurement and recording.

The new Hewlett-Packard Model 3495A Scanner does this and much more. For example, it can serve as a programmable switch for actuating external processes or for distributing power and/or signals. It can also connect more than one channel at a time so it is able to make the multiple connections needed for four-wire resistance or floating bridge measurements. Where there may be more than one measuring and/or stimulus channel to be connected to a multiport device, the new scanner can also be configured to make the multiple switch closures needed for matrix switching.

The scanner is designed to work with the HP Interface Bus in calculator-controlled systems. It is addressable and once addressed to listen, it accepts and stores channel addresses until it receives the EXECUTE command, which then causes the indicated channels to open or close. It switches up to 40 channels and up to five scanners can be connected to the bus to give break-before-make operation of up to 200 channels.

Channel switches are mounted on plug-in circuit modules, 10 channels to a module. The mainframe holds up to four modules in any combination desired and is easily reconfigured in the field to meet the needs of new measurement situations.

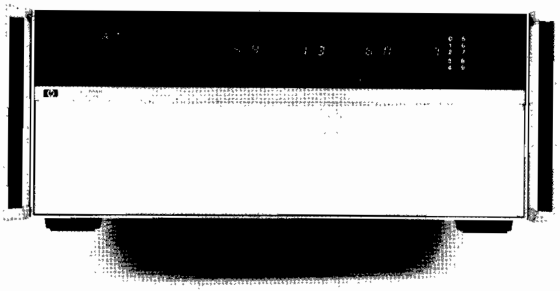
Two types of modules have been designed. One, known as the low-thermal assembly, was designed for minimum offset voltages ($<2\ \mu V$) and high impedance ($>10^{10}\Omega$ isolation) for use with thermocouples and other low-level transducers as well as high-level signals. It has three reed switches per channel. One switch in each channel connects the circuit guard and is programmed to close before and open after the HI and LO signal switches.

Only one low-thermal channel may be closed at a time on any one module but channels on adjacent modules may be closed simultaneously, making it possible to have up to four simultaneous closures if four modules are installed. Jumpered connections on each module enable channels on two or more modules to respond to the same channel address for simultaneous switching of channels.

The maximum terminal-to-terminal voltage permitted is 200V. Switching time is less than 10 ms with hardware insured break-before-make switching.

The other type of circuit module is known as the actuator assembly. Each of these modules has 10 general-purpose, double-pole, single-throw, armature-type relays with contacts rated at 100 V max, 2 A max. Any combination of channels on one of these boards may be closed simultaneously.

This card has a single unswitched guard for all 10 channels. Switching time is less than 40 ms, giving a maximum closure rate of about 30 channels per second.



Instruments for Use in HP Interface Bus Connected Systems

In addition to the instruments described in this issue, there are a number of Hewlett-Packard instruments that operate on the HP Interface Bus when equipped with the appropriate options. These include the following:

Signal Sources

3320B Frequency Synthesizer, 0.01 Hz to 13 MHz
3330A/B Automatic Synthesizer/Sweeper, 0.1 Hz to 13 MHz
8660A/B Synthesized Signal Generator, 10 kHz to 1.3 GHz
8016A Word Generator, 9 outputs, 32 bits each, 0.5 Hz to 50 MHz

Measuring Instruments

3490A Digital Multimeter, dc volts, ac volts, ohms; also ratio and sample/hold options
5340A Automatic Frequency Counter, 10 Hz to 18 GHz
5341A Automatic Frequency Counter, 10 Hz to 4.5 GHz
5345A Plug-in Electronic Counter, to 500 MHz direct and higher with plug-ins; time interval down to 2 ns

Calculators

9820A Algebraic Language Calculator, magnetic card programming
9821A Algebraic Language Calculator, tape cassette programming
9830A BASIC Language Calculator, typewriter-style keyboard

for time interval measurements, among other uses (signal path lengths are matched within 10 ps). It is also useful for routing the outputs of signal sources to various points in a measurement system.

Besides being actuated by instructions received over the HP interface bus, both the VHF switch and the relay actuator may be activated by front-panel pushbuttons. Local control facilitates system set-up.

Acknowledgments

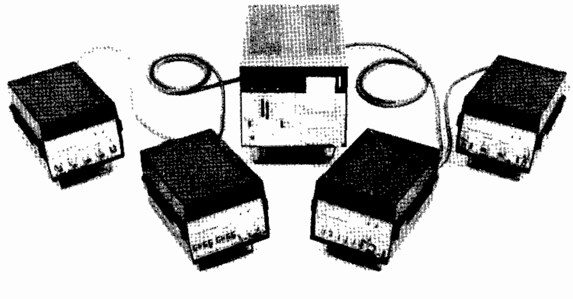
Special credit is due Eric Havstad and Carl Spalding for the product design. John Dukes and Jim Sorden fostered initial concepts of the modules with Jane Evans providing valuable marketing support.

Minimal Cost Measuring Instruments for Systems Use

by Gary D. Sasaki and Lawrence P. Johnson

Although the HP Interface Bus technology makes it possible to use very sophisticated laboratory instruments in automatic measuring systems, many requirements can be fulfilled with more modest instrumentation.

The HP 5300-Measuring System (counters, timers, and a multimeter) has the potential for providing a wide variety of capabilities for low-cost automatic systems. This system consists of mainframes that have basic counting capability, an LED display, and a time base. Signal conditioning circuits are in functional modules that mate with a mainframe to form a complete instrument. The transfer of signals between the functional module and the mainframe by way of a multi-pin connector provides an ideal means to gain access to measurement data for use on the HP Interface Bus.



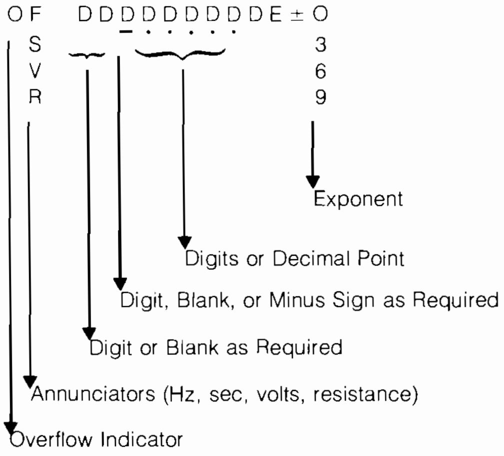
Accordingly, a new "snap-between" module, the Model 5312A ASCII Interface, was designed to make the measurement data available in ASCII code to the HP Interface Bus. This module works with the Model 5300B Mainframe to make any of the eight presently available plug-on functional modules a measurement entry port for systems using the HP Interface bus (the Model 5300A Mainframe does not make certain status signals available to the module connector so cannot be used with the ASCII Interface).

The measurement capabilities made available through various plug-on modules by this arrangement are frequency (up to 1.1 GHz), period, time interval, total count, frequency ratio, ac volts, dc volts, and resistance. Besides amplifiers and attenuators, the functional modules may have trigger circuits and the logic for making time-interval measurements, or voltage-to-frequency converters to enable measurements of voltage and resistance.

The 5312A ASCII Interface is addressable, meaning that a controller can direct any instrument using this module to place measurement data on the bus while the others remain silent. In keeping with the goal of providing accuracy and reliability at minimum cost, however, the 5300 instruments are not programmable, so functions and ranges must be set manually (except for those that are autoranging).

The modules are compatible with the various possible levels of system complexity. In the "talk only" mode they can be used with an HP-IB compatible printer (e.g., Model 5150A, page 14), to provide an economical data logger. In the "addressable" mode, up to 13 may be used with the Model 5150A Printer and its optional scanner to collect data from several sources that are making the same or completely different measurements from many inputs simultaneously, such as voltages from strain gauges and thermocouples, frequencies from carrier pre-amps and rotating machinery, or time intervals from photocells. Each instrument can gather data continuously, as it would in a stand-alone application, but it places data on the interface bus only when addressed by the controller. If the timing of a measurement is important, such as several inputs to be measured simultaneously, an initiate command directs each instrument to take a reading and hold it until addressed to put the data on the bus. When used with a desk-top calculator or a computer, the modules provide a low-cost approach to multiple-input data acquisition systems that can be modified any time with more complex instruments without modifying the interface.

Because the output of the ASCII Interface module is byte-serial, it is possible to output more information than is possible with the familiar BCD output. The module's output is 15 bytes in a format like the following, as it would appear on a Model 5150A Thermal Printer (it also transmits the LF and CR commands):



Inclusion of an indication of measurement units is especially useful with multifunction modules, such as the Model 5306A Multimeter (volts, ohms, frequency).

Visualizing Interface Bus Activity

Connecting to the HP Interface Bus, a new analyzer listens to and displays the status of all bus lines for easy study of bus activity. It also serves as a talker, using programs in its internal memory to exercise bus-compatible instruments and systems.

by Harold E. Dietrich

AS ANYONE WHO HAS ASSEMBLED instruments into a system well knows, software and hardware problems always seem to arise. Many of these problems are avoided when the HP Interface Bus is used but, even though the bus standardizes connectors, control lines, signal levels, and message transfer protocol, software errors can occur if the system designer does not completely understand the bus system or the capabilities of the instruments he's using. Hardware problems occur if the instruments are not functioning properly or if they are not completely compatible with the bus standard. Additional problems face the designer of a new instrument when he evaluates its compatibility to the interface bus system.

Solutions to these problems are found much more quickly with the help of the new Model 59401A Bus System Analyzer (Fig. 1). It connects to the bus in the same way as any other instrument. As a bus listener, it displays the status of all the lines in the bus. It enables the designer to go through his program step by step and, by making bus traffic observable, it makes software debugging relatively easy.

Besides serving as a listener, the analyzer can also

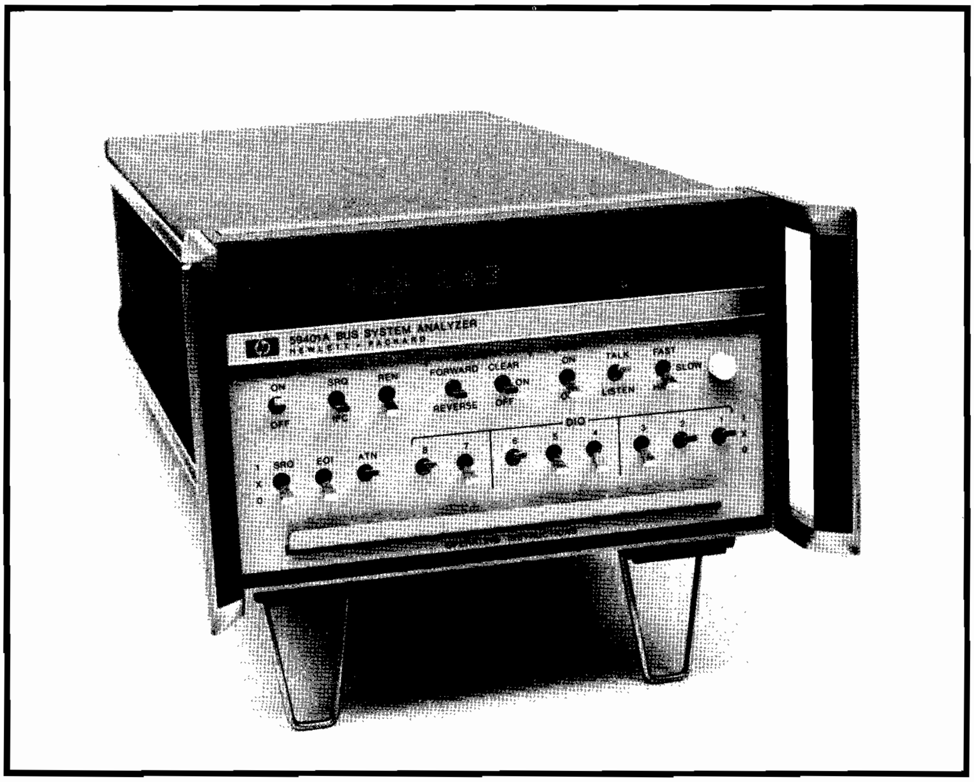


Fig. 1. *Model 59401A Bus System Analyzer monitors activity on the HP Interface Bus, displaying the instantaneous status of all bus signal lines. It can also be a bus controller using programs stored in its memory or program steps set up on the front-panel switch register.*

be a talker. It thus can completely exercise another talker, listener, or controller. It has an internal read-write memory that holds 32 program steps, loaded either from the front-panel switch register or from the bus itself. With a suitable program loaded, the Analyzer can exercise instruments at maximum bus speed, or step by step.

Visibility

Bus activity is made visible by an array of displays on the analyzer's front panel. Annunciator LEDs indicate the presence of the "true" state on the corresponding bus control lines. Information on the eight data lines, however, is converted to the equivalent octal number for numeric display, making it easier to read the instantaneous contents of the bus. As a further aid, the equivalent ASCII character is also displayed, which makes it easier to follow the steps in a calculator-based program.

The same front-panel alphanumeric display is also used for display of the contents of the internal memory. Two additional digits are then illuminated to show the address in memory of the character on display.

Monitoring Traffic

In the LISTEN mode, the bus system analyzer functions as an addressed listener to all bus traffic. Because data transfer on the bus is under control of the three "handshake" lines (see box, page 3), the analyzer can readily slow traffic to a speed convenient for visual monitoring. Three "speeds" are provided. In the FAST mode, bus traffic proceeds at the fastest rate allowed by the slowest instrument in the system. In the SLOW mode, the bus analyzer limits bus speed to two characters per second, a rate that enables visual monitoring of bus traffic. In the HALT mode, a character is accepted only when the analyzer's EXECUTE button is pressed, allowing the program to be stepped one character at a time. The operator can switch from SLOW to HALT when bus traffic approaches the place in the program where a software problem is known to exist, and then proceed one step at a time to find the problem. It is this ability to go through a program step by step while displaying bus activity that gives the analyzer its great usefulness.

The bus analyzer also has a COMPARE mode. It stops bus traffic when the character on the bus matches the settings of the front-panel switch register (the lower row of switches, Fig. 1). A program thus stopped may be resumed by pressing the EXECUTE button.

Whether or not the COMPARE switch is on, the analyzer outputs a pulse whenever a bus character matches the front-panel switch register. This can be used to trigger an oscilloscope to observe related analog signals or to examine transients on the bus (test points for all bus lines are available at the rear of the instrument).

Replay of Past Events

The bus analyzer can store data characters as they appear on the bus, continuously updating its memory so it always contains the most recent 32 characters. The COMPARE mode can be used to stop bus traffic where a problem is known to exist, then the previous 32 steps, now stored in memory, can be recalled one by one to determine what led to the problem. A front-panel switch steps the memory in either a forward or reverse direction.

For example, a DVM in a system occasionally outputted a negative reading when all readings were supposed to be positive. The analyzer's switch register was set to the ASCII code for "-" (00 101 101_2 or 055_8) and the system was operated with the analyzer in the COMPARE mode. The occurrence of the minus symbol on the bus data lines halted bus traffic, and the previous steps could then be recalled from the Analyzer's memory to find out what happened.

Step by Step

The TALK mode is useful in checking out a new interface design. The analyzer can determine whether or not an instrument can be addressed and unaddressed, whether or not it meets timing requirements (sufficiently short settling time, releases lines in time, etc.), and whether or not it can handshake with other instruments that are either faster or slower.

For example, a calculator-based system using several programmable multiposition switches was assembled. At one part of the program, the software designer wished to program two switches to the same position, so the program addressed both to LISTEN followed by programming information. The following calculator command statement was used:

CMD "?U#$", "A2B2"

The symbol "?" is the unlisten command and "U" designates the calculator as the talker. "#" is the listen address of the first switch, "$" is the listen address of the second, and A2B2 programs both for switch position 2. But, when the program was executed, only the second switch programmed correctly.

To trace the problem, the first switch was connected to the bus analyzer. The analyzer was set to TALK and the switch's listen address (# or X0 100 011) was set into the analyzer's front-panel switch register. With the analyzer's ATN switch set to true, DAV was driven true when the EXECUTE button was pressed, thus sending the address. The switch responded by driving NRFD true and NDAC false, as shown by the analyzer's front-panel annunciators.

When the EXECUTE button was released, DAV went

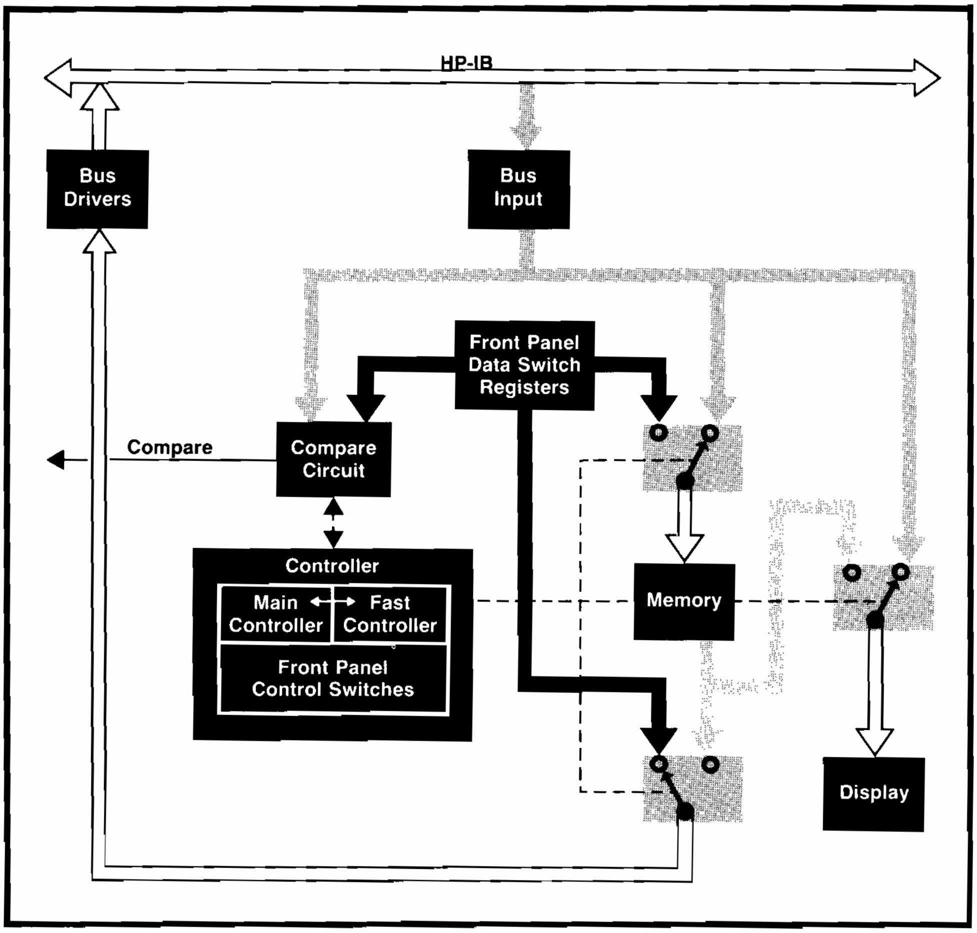


Fig. 3. *Simplified block diagram of the Model 59401A Bus System Analyzer. The switches shown represent gates operated in combinations by the internal controller in response to front-panel switch settings. The main controller manages most operations but a fast bipolar 'controller is used for the FAST TALK and FAST LISTEN functions.*

false. The switch then sent NDAC true and NRFD false, indicating that it had accepted the address and was ready for more inputs. ATN was then set false, and NDAC remained true, indicating that the switch was addressed to listen.

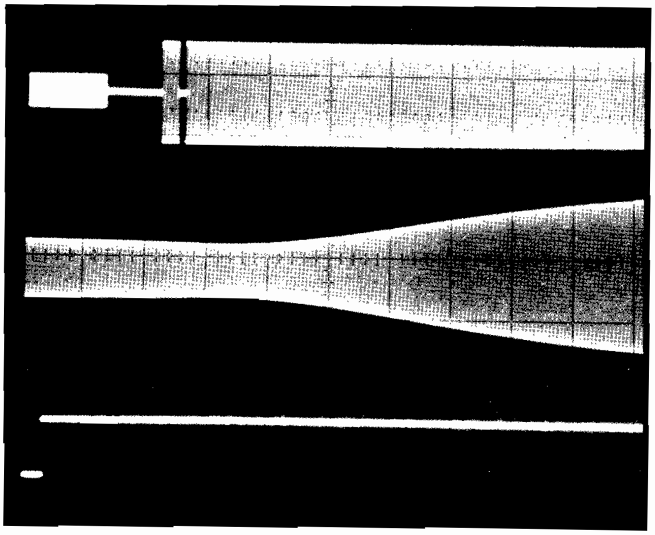


Fig. 2. *Oscillogram demonstrates use of the bus analyzer as an oscilloscope trigger (see text). The bottom trace shows the timing of the analyzer's "compare" pulse. The top trace shows the synthesizer output in response to the trigger event and the middle trace shows the output of the filter under test.*

The programmer then set the address of the second switch on the analyzer's front-panel register and repeated the procedure. This time, when ATN was set false, NDAC went false. This showed that the first switch became unaddressed when another listen address was placed on the bus. The problem was thus traced to the interface logic in the first switch.

Speed-Related Problems

A rear-panel input for an external clock enables the analyzer to slow the rate of bus activity to any rate below system maximum. This was used to track down a problem with a listener that would operate with some controllers but would cause the bus to hang up when used with others.

The analyzer was connected to the listener and the controller program was loaded into the analyzer's memory. This "minisystem" was then operated at various data rates by varying the external clock rate until the fault occurred. The problem was found in the handshake response and was quickly traced to an optical-isolator circuit.

Ancillary Activities

The photo in Fig. 2 illustrates how the Analyzer can be used for observing analog phenomena related to interface bus activity. The need was to determine what effect, if any, programming data for a frequency synthesizer had upon a test of a crystal filter. The test was to determine the response of the filter to a 10-dB change in signal amplitude. Filter center frequency was 100 kHz and bandwidth was 100 Hz.

Programming information for the synthesizer was stored in the analyzer's memory. To obtain a compare pulse for triggering the oscilloscope, a "1" was placed on data line DIO8 with the data byte that executes the change to the new output amplitude. The synthesizer ignores line DIO8 since it responds only to 7-line ASCII characters. The analyzer's switch register was set to 1XXXXXXX.

As the program executed, the oscilloscope triggered on the step that included the "1" on line DIO8. As the oscillogram of Fig. 2 shows, transients are caused by the switching of relays in the synthesizer's programmable step attenuator when a 10-dB or greater change in amplitude is called for.

Internal Organization

A simplified block diagram of the Model 59401A Bus System Analyzer is shown in Fig. 3. As a listener, the instrument accepts inputs from all the bus lines and displays their status on the annunciator LEDs and alphanumeric display. As a talker, characters and control signals are placed on the bus either from the analyzer's memory or from the front-panel switch register.

The memory consists of six 64-bit RAMs. In the listen mode, the memory is operated in push-down fashion so the oldest data is dropped as new data comes in. In the talk mode, characters set on the front-panel switches are entered in by the EXECUTE button. The contents of any memory location can be edited by using the FORWARD/REVERSE switch to step the memory to the address desired, then pressing the EXECUTE button to enter the new switch-register data.

As the block diagram shows, two controllers are used. The "fast" controller enables the analyzer to operate up to the maximum speed the bus may ever operate (1 megabyte/s). To accommodate the handshake operation, this speed requires the use of a bipolar controller. However, performing all operations within the instrument with a bipolar controller would have been excessively expensive. Hence, the bipolar controller is used only for the FAST LISTEN and TALK modes and an MOS controller is used for all the other functions.

To minimize the number of state times needed for processing a bus character, the fast controller can check up to four inputs (or qualifiers) in each state. For example, one of the states allows the controller to

ABRIDGED SPECIFICATIONS

Model 59301A ASCII/Parallel Converter

INPUT: From HP Interface Bus; 24-pin connector.

OUTPUT: Logic 0 = 0 4V; logic 1 = 5V. Two 50-pin connectors.
PRINT COMMAND. Negative pulse, +5V to +0.4V for minimum of 20 μs.
INHIBIT COMMAND INPUT: Voltage level within +2.4V to +20V into 5k ohms.

DIMENSIONS: ½-width module, 3 lb 12 oz (1.70 kg)

PRICE IN U.S.A.: $550.

Model 59303A Digital-to-Analog Converter

INPUT: From HP Interface Bus; 24-pin connector.

OUTPUT: −9.99V to +9 99V; front-panel banana plugs in parallel with rear-panel BNC.

ACCURACY: Output within ±0.1% (±½ LSD) over 0 to 50°C temperature range.

SETTLING TIME: <30 μs to ±½ LSB.

BLANKING: TTL signal that indicates digital information is changing. Duration, 25 μs after LF is accepted.

DIMENSIONS: ¼-width module; 5 lb 12 oz (2.61 kg).

PRICE IN U.S.A.: $850.

Model 59304A Numeric Display

INPUT: From HP Interface Bus; 24-pin connector.

DISPLAY: Gas discharge (orange) 0.4 in high; 12 characters and decimal point.

DIMENSIONS: ¼-width module; 2 lbs 11 5 oz (1 23 kg).

PRICE IN U.S.A.: $650.

Model 59306A Relay Actuator

OUTPUT TERMINALS: Banana jacks arranged on the rear panel in three rows, A, B, and C (common) Normally closed position is B to C and normally open position is A to C.

RELAY CONTACTS: 0.5A at 28V dc or 115V ac

RELAY SETTLING TIME: 20 μs.

DIMENSIONS: ½-width module; 5 lb 13 oz (2 64 kg)

PRICE IN U.S.A.: $650.

Model 59307A VHF Switch

IN/OUT TERMINALS: BNC's, one of four for each input (or vice-verse) selectable under front-panel or program control.

RELAY CONTACTS: 0.5A at 25V, <1 ns transition time.

RELAY SETTLING TIME: 20 μs

VSWR: <1.1

ISOLATION: >40 dB @ 100 MHz.

DIMENSIONS: ½-width module; 7 lb 2 oz (3.23 kg).

PRICE IN U.S.A.: $750.

Model 59308A Timing Generator

TIME INTERVAL (Δt) RANGE: 1 μs to 99,900s (001E0 to 999E8 μs), selected on thumbwheel switches or programmed through HP Interface Bus

TIMER MODE: Outputs one pulse one Δt following start trigger.

TRIGGER OUTPUT (rear panel): TTL or ECL logic levels (switch-selected), 50 ns transition time. Pulse width, 500 ns ± 100 ns (pulse mode) or ½Δt (square-wave mode)

TRIGGER INPUT (rear panel): Edge triggared et 0.5V or 2V, positive or negative slope, switch-selected. Input R, 10 kΩ

TRIGGER/RESET PUSHBUTTON (front panel): Output trigger occurs within 1 μs of time set after pushbutton is released.

TIME BASE: Crystal frequency, 10 MHz; 3 parts in 10^7 per month aging rate; temperature, ±5 parts in 10^6, 0° to 50°C Rear-panel BNC accepts 1, 5, or 10 MHz external standard.

DIMENSIONS: ½-width module; 4 lb 10 oz (2.10 kg).

PRICE IN U.S.A.: $875.

Model 59309A ASCII Digital Clock

DISPLAY: Month, day, hour, minute, second. LED numerals. } Programmable
RESET: Resets display to 01:01:00:00:00 and starts clock. }

SET DAY/TIME: Updates display (fast or slow) to arrive at desired time and date.

LEAP YEAR: Switch selects 365 or 366 days/year (non-programmable).

ERROR INDICATOR: All decimal points light to indicate possible error from power interruption or missed counts.

TIME BASE: 1-MHz room-temperature crystal; 5 parts in 10^6 aging rate; temperature, 5 parts in 10^6, 10° to 40°C (0.5 s/day). Accepts external 1, 5, or 10 MHz frequency standard (1V rms into 1 kΩ).

STANDBY POWER: Internal 9V battery (not supplied) maintains tima for 1 day with display off. Accepts 8 to 10V at 2 mA from external source.

DIMENSIONS: ¼-width module; 2 lb 10 oz (1.18 kg)

PRICE IN U.S.A.: $975.

All 59300 Series Units

DIMENSIONS:
WIDTH. Quarter-width modules, 4.17 in (105.9 mm). Half-width modules, 8.38 in (212.9 mm).
HEIGHT: 4 in (101.6 mm).
DEPTH: 11.4 in (289 6 mm).

POWER: 115/230V ± 10%, 50 to 400 Hz; 15 VA max.

Model 5312A ASCII Interface

SAMPLE RATE: Controlled by mainframe front panel control or by setting rate of reset command (when in listening mode, counter can be reset by sending the letter I).

TRANSFER TIME: 20 milliseconds (typical).

TRANSFER RATE: Maximum of 40 reading/sec depending on capabilities of plug-on.

SELF TEST MOOE: Checks functioning of interface.

NOTE: The 5312A is not compatible with the 5300A mainframe which contains its own BCD Digital Output.

PRICE IN U.S.A.: $350.

Model 5150A Thermal Printer

PRINTING TECHNIQUE: Thermal print, 5 × 7 dot matrix.

RATE AND SPACING: 3 lines/s, 6 lines/inch.

PAPER: Thermal sensitive in rolls or fan-folded.

POWER: 100, 120, 220, or 240V, 48 to 440 Hz (50 or 60 Hz only with clock option); 100 VA.

DIMENSIONS: 8 5 in W × 7 5 in H × 14.25 in D (216 × 176 × 356 mm).

WEIGHT (with one option): Approximately 16 lbs (7 kg).

ASCII Interface (opt 001)

COLUMNS: 20.

CHARACTER SET: 64 ASCII characters (columns 2, 3, 4, and 5 of ANS 3.4 − 1968 except "↑" in column 5, row 14)

BCD Interface (opt 002)

COLUMNS: 10 (20 with two option 002's installed).

CHARACTER SET: 0 through 9, +, −, V, A, R, and blank. Special character sets available.

INPUT: TTL levels, switch selects + or − true logic. Parallel BCD (8421) format.

PRINT COMMAND: TTL level, + or −

Scanner (opt 003)

INSTRUMENTS SCANNED: 1 to 13.

CYCLE TIME: Limited by slowest of (1) instrument response time, (2) 3 samples per second, or (3)DATA PRINT INTERVAL with optional clock.

COMPATIBILITY: HP Interface Bus.

Clock (opt 004)

DISPLAY: 6-digit LED display of hours, minutes, seconds; settable via front-panel switches.

TIME BASE: Line frequency, 50 or 60 Hz (selectable by internal jumper).

DATA PRINT INTERVAL: Minimum, 1, 2, 10, 20 sec; 10, 20 min; 1, 2 hours.

TIME PRINT INTERVAL: Same as data; interlock prevents intervals shorter than data interval.

PRICES IN U.S.A.: 5150A Printer, $800, Scanner, $175.
ASCII Interface, $175, Clock, $250.
BCD Interface, $110.

MANUFACTURING DIVISION: SANTA CLARA DIVISION
5301 Stevens Creek Boulevard
Santa Clara, California 95050

SPECIFICATIONS

HP Model 59401A Bus Systems Analyzer

LISTEN:
ACCEPT TIME: <750 ns
READY TIME: <750 ns

TALK:
1 Data changed >500 ns before DAV pulled low.
2 ATN driven low >1 μs before DAV pulled low.
3. DAV driven high < 700 ns after NDAC is false.
4. DAV driven low <700 ns after NRFD is false if conditions 1 and 2 are met.

EXTERNAL CLOCK INPUT: TTL gate input; ≤10 MHz repetition rate.

COMPARE OUTPUT: TTL gate output, low true.

POWER: 100/120/220/240V, +5%, −10%, 48 Hz to 86 Hz; 42 VA max

DIMENSIONS: 5 73 in H × 9 65 in W × 19.5 in D (145 × 245 × 495 mm).

WEIGHT: 12 lb 7 oz (5 64 kg).

PRICE IN U.S.A.: $4900

MANUFACTURING DIVISION: LOVELAND INSTRUMENT DIVISION
P.O. Box 301
815 Fourteenth Street, S.W.
Loveland, Colorado 80537

check whether listeners are ready for data (NRFD), whether sufficient time has elapsed since the data was changed and since ATN was pulled low, and whether there has been an interrupt from the main controller. If all conditions are met, data valid (DAV) can be driven low within one state time (200 ns). Thus, the fast controller needs only 16 states, and it operates with 1280 bits of ROM. The main controller has 256 states and uses 14,336 bits of ROM.

Donald C. Loughry

A 1952 graduate of Union College (BSEE), Don Loughry joined Hewlett-Packard as a production test manager in 1956. Two years later he moved to the Dymec Division lab to work on systems and later became division engineering manager. In 1968, Don was appointed corporate interface engineer. He's active in various groups working on interface standards, including IEC, ISO, ANSI, and IEEE. Don is also interested in skiing, photography, gardening, and church work.

David W. Ricci

One of the developers of the HP Interface Bus, Dave Ricci sometimes participates in the deliberations of various standards committees. He joined HP in 1965 after earning his MSEE degree at the University of California at Berkeley (he obtained his BSEE at the California State Polytechnical University). Initially he worked on Multichannel Analyzers but he is now developing applications for counters that use the HP-IB. Dave and his wife like to ski, sail, and backpack. They have one son, 2½ years old.

Peter S. Stone

Pete Stone did it all in one stretch: BSEE, MSEE, and PhD degrees at the University of Minnesota. He then worked for a medical equipment firm for a year before joining Hewlett-Packard in 1972 to work on automatic systems. He is presently group leader in the voltmeter lab. Married, but with no children yet, he still finds time for flying (he's 1/3-owner of a Cherokee 180) and ham radio (both fixed and mobile).

Steven E. Schultz

Steve Schultz joined HP in 1971, going to work for the 5345A Counter group before becoming project leader on some of the ASCII-Programmable Modules. A 1970 graduate of the University of California at Berkeley (BSEE), Steve is now finishing work on an MSEE at Stanford in the HP Honors Co-op program. Most of his spare time goes into his studies but he has many other interests, including dirt bikes and radio-controlled boats. He and his wife live in Menlo Park, California.

Charles R. Trimble

Graduating with honors from the California Institute of Technology in 1963, Charlie Trimble went on to get his MSEE there the following year. He then joined Hewlett-Packard where his initial project was the 5480A Signal Averager. Now an engineering section manager, he has a dual responsibility for precision timing measurements and for the Santa Clara Division's HP-IB activities. Charlie and his wife like sailing, backpacking, and bicycling. They live in Los Altos, California.

Hans-Jürg Nadig

Hans Nadig came to the United States and Hewlett-Packard in 1967. Initially he worked on Fourier Analyzers, then spent a year in IC production before taking on the thermal printer project. Hans graduated with a Diplom Ingenieur from the Federal Institute of Technology in Zurich, Switzerland, and designed automatic controls for mass transit systems before joining HP. He has a wife and two daughters and likes ski touring, mountain climbing, and photography.

Acknowledgments

Thanks are due Michael C. Williams for the product design and Skip Beatty for help with the breadboard. Gene Meismer and Al Boswell aided the transition into production. Special thanks are due Gerald E. Nelson for his guidance.

Harold E. Dietrich

A 1966 graduate of Purdue University (BSEE), Harry Dietrich came straight to work for Hewlett-Packard, contributing to the 3555A Telephone Test Meter, 653A Video Test Oscillator, and 3570A Network Analyzer before taking on the bus analyzer. Harry earned his MSEE degree at Colorado State University in 1970 in the HP Honors Co-op program. He likes hunting but he also takes his family, which includes three small sons, on camping trips in his pickup camper.

Gary D. Sasaki

Gary Sasaki graduated from the University of California (Berkeley) with a BSEE in 1973 and came straight to work on the ASCII Interface for the 5300 Measuring System. Gary actively promotes the engineering profession to high school students by giving talks at schools and holding informal "rap" sessions in the plant with students. For fun he plays volleyball and ping-pong. He and his wife build furniture by hand (their only power tool is a drill).

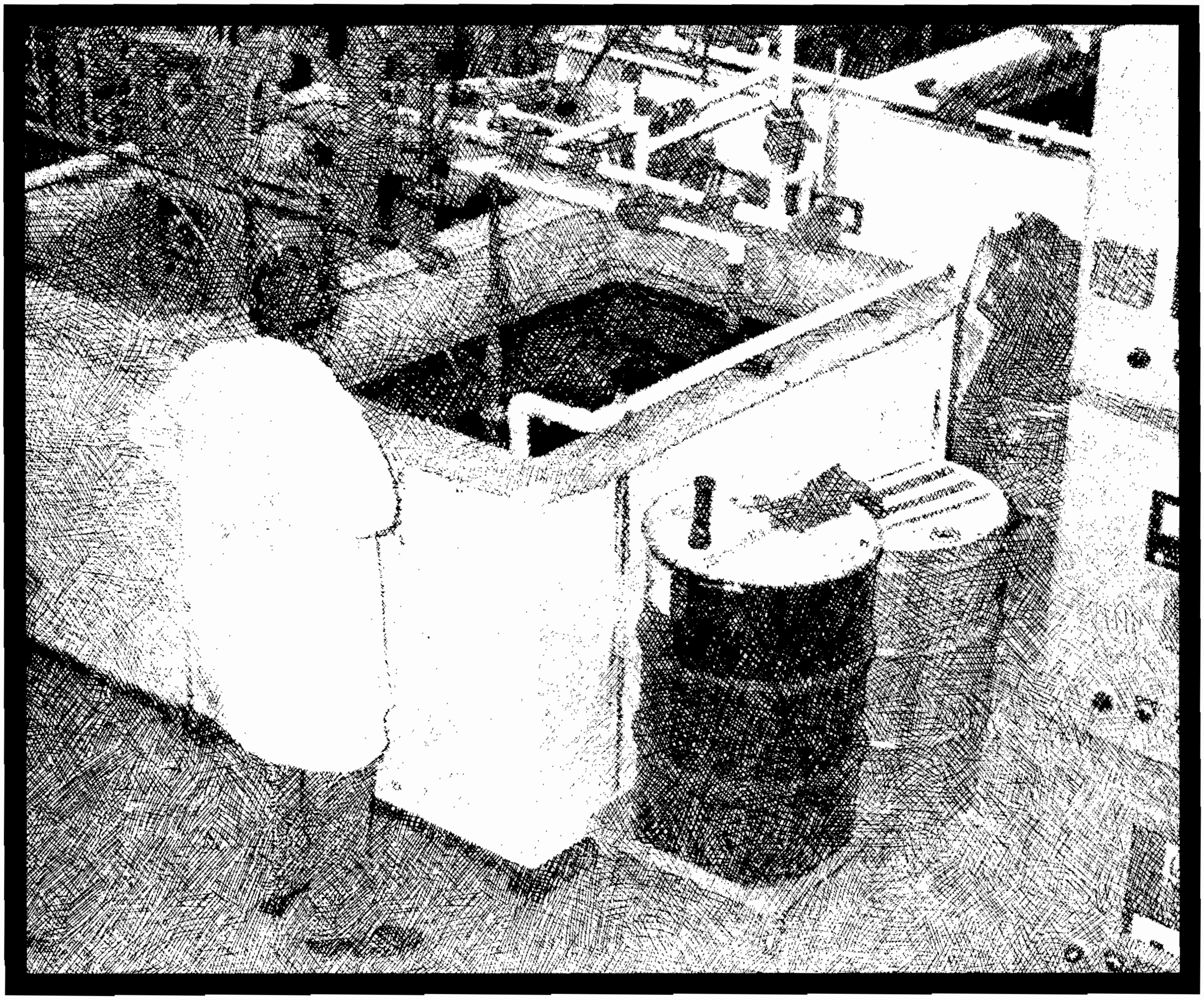
David L. Wolpert

Dave Wolpert joined the Hewlett-Packard Loveland Instrument Division in 1972 upon getting a BSEE degree at the Georgia Institute of Technology. Initially he worked on some investigatory projects, then went to the 3459A Scanner project. Dave's married and he gets involved in photography, doing his own darkroom work, and folk guitar.

Lawrence P. Johnson

Larry Johnson earned a BSME at Rensselaer Polytechnic Institute followed by a Master's degree in manufacturing engineering at Boston University (1968) and later an MBA. He joined HP's Medical Electronics Division in 1968 as a process engineer, spent some time at the San Diego Division, then moved to the Santa Clara Division as product manager for the 5300 Measurement System, which included work on the ASCII Interface. Larry and wife ski and dabble in real estate.

applications
bulletin L6



Monitoring and Controlling the pH* of Industrial Chemical Waste

The disposal of industrial chemical waste is a problem for many industries. The simplest and most economical method of disposal of chemicals that meet the EPA or local requirements for metallic content is to discharge the waste into the sewer lines. However, before this can be done, the chemical waste must be neutralized so that it will not upset the sewage treatment process. This article will cover a method by which this process can be performed automatically.

*See Appendix

HEWLETT hp PACKARD

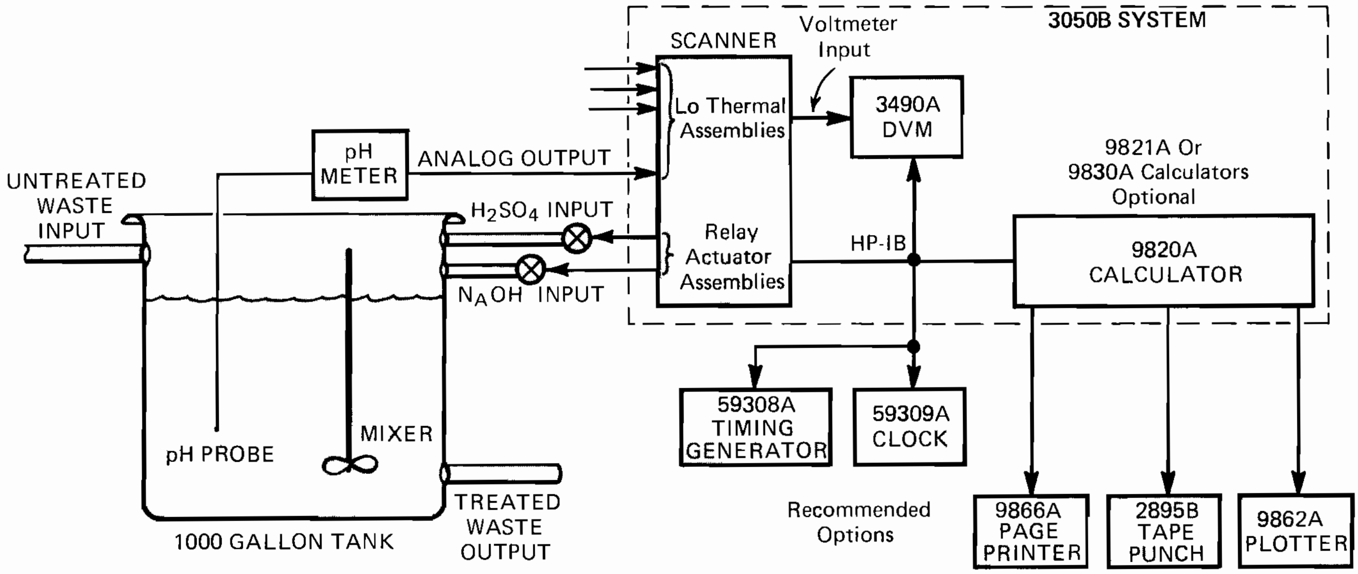


Figure 1

The pH of the industrial waste must be monitored and the proper amount of neutralizer added to make the pH acceptable for the sewage treatment plant. It may also be required that the pH of the final waste being dumped into the sewer be monitored and recorded.

The HP 3050B Automatic Data Acquisition System can both monitor the output of a pH Meter* as well as compute and control the amount of neutralizer added to the chemical waste. The system also has the capability of recording the pH of the waste. An example of a solution to this problem would be the treating of the waste chemicals from a printed circuit board manufacturing process.

The waste from this process may vary during particular portions of the process. The pH of this solution will be in the range from 2 to 12. This waste must be neutralized such that when it enters the sewer system, it has a pH of 7.5 to 8.5 The neutralization is achieved by adding either a solution of H_2SO_4 (sulfuric acid) or N_aOH (sodium hydroxide) depending on whether the waste is basic or acidic. Figure 1 shows a physical layout of the process. The time between samples of pH can be up to several minutes because the neutralizer must be uniformly mixed with the contents of the tank. The type and amount of neutralizer required will vary with pH of the waste and volume of the waste. Figure 2 shows the process flow chart.

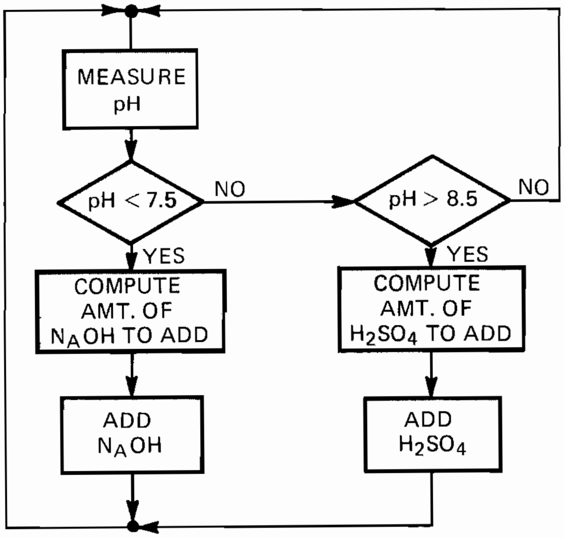


Figure 2.

Flow Chart of pH Monitoring and Control Process

To compute the volume of neutralizer required:

C_I - Concentration of Input Waste
C_N - Concentration of Neutralizer
V_I - Volume of Input Waste
V_N - Volume of Neutralizer Required
$C_I V_I = C_N V_N$ in order to neutralize the waste.

$$\text{Thus,} \quad V_N = \frac{C_I V_I}{C_N}$$

*See Appendix

The 3050B System can be expanded to operate in a system where the input waste is continuously mixed with the neutralizer and allowed to flow on through a holding tank. In this type of application, the chemical waste flow rate as well as pH must be monitored. The neutralizer flow rate must also be controlled in order to obtain the proper waste pH at the output.

The system can also record the pH on an internal cassette recorder if a 9821A or 9830A Calculator is selected as the system controller. The addition of an external clock, HP Model 59309A, can allow recording of second, minute, hour, day and month to be included in the record. The system is capable of printing information at selected intervals or the cassette can be used to plot or print data at some later time. The speed with which the 3050B can monitor and control the process would allow it to handle several processes simultaneously.

An additional application is to monitor the pH of the final waste output into the sewage system and sound an alarm if the pH goes beyond the acceptable limits. This would allow the sewage to be diverted from the main sewage treatment process, thus eliminating the possibility of upsetting the treatment process.

The 3050B Automatic Data Acquisition system is capable of solving a wide variety of measurement problems. These capabilities can be summarized as follows:

(1) Control the system instruments,
(2) Acquire and convert analog data from physical sensors to digital form,
(3) Correct the data for nonlinearity and offsets and convert it to meaningful scientific units,
(4) Determine test results,
(5) Control processes or set alarms,
(6) Perform high level statistical and historical analysis, and
(7) Log or display results.

APPENDIX

pH

A measure of the free hydrogen ion concentration of an aqueous solution. At 25°C, the pH scale hydrogen ion concentration, 7 is neutrality, and 14 is highly alkaline (very low free hydrogen ion concentration). Negative values and values greater than 14 are also possible.

What is a pH Meter?

A pH Meter typically consists of a probe, shown in Figure 1, and a high input impedance voltmeter calibrated to read pH directly on the meter and give an analog output voltage proportional to pH.

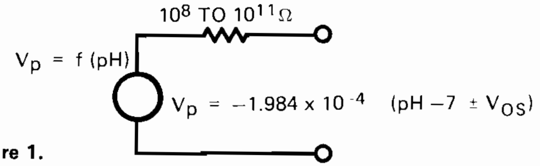


Figure 1.
Model of a glass calomel pH probe appears as a voltage source in series with a high resistance.

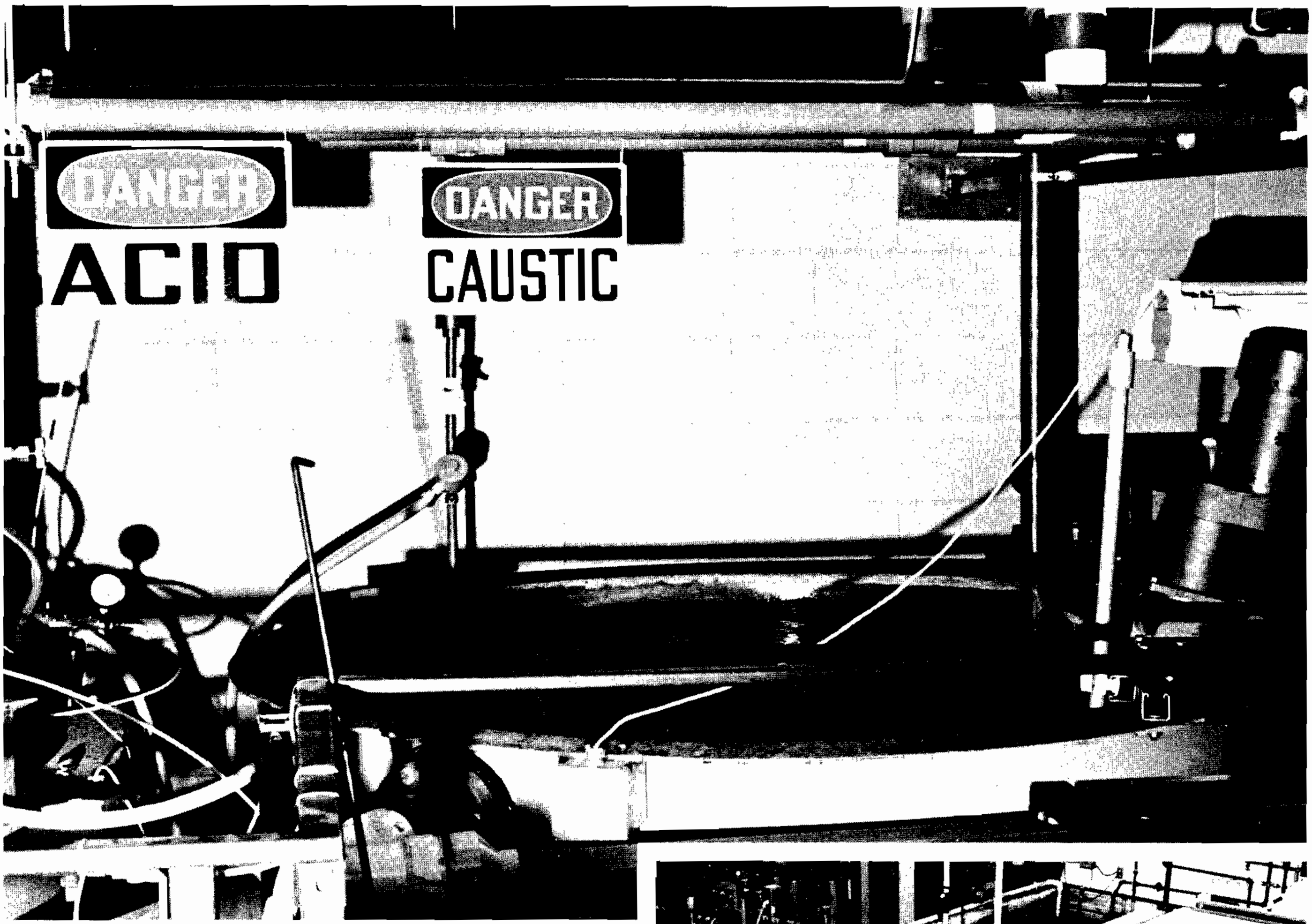
The voltage-pH relationship is described by the equation:

$$V = -\frac{RT\,(\ln 10)}{F}\quad(pH - 7 \pm V_{OS})$$

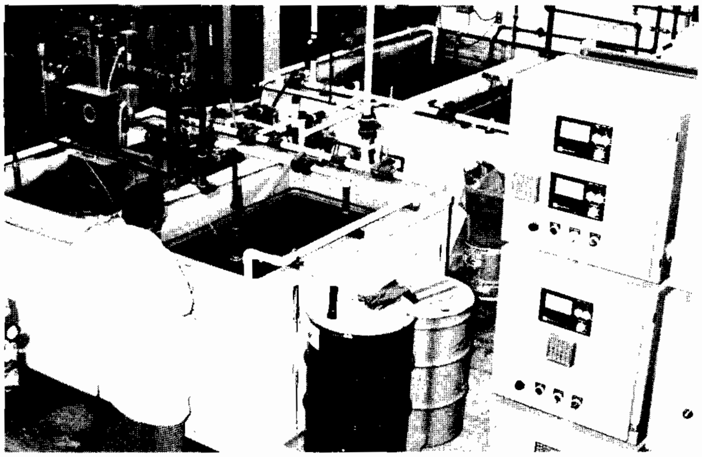
where R is the gas constant (8,314 J^oK^{-1}), F is the Faraday constant (96487 colulombs/mole), T is the probe temperature in degrees Kelvin, and V_{OS} is a small offset voltage which has a different magnitude for each probe.

With the constants inserted, the equation becomes:

$V = -1.984 \times 10^{-4} T\,(pH - 7 \pm V_{OS})$,
and when T = 25 C (298 K),
$V = -0.05913\,(pH - 7 \pm V_{OS})$.



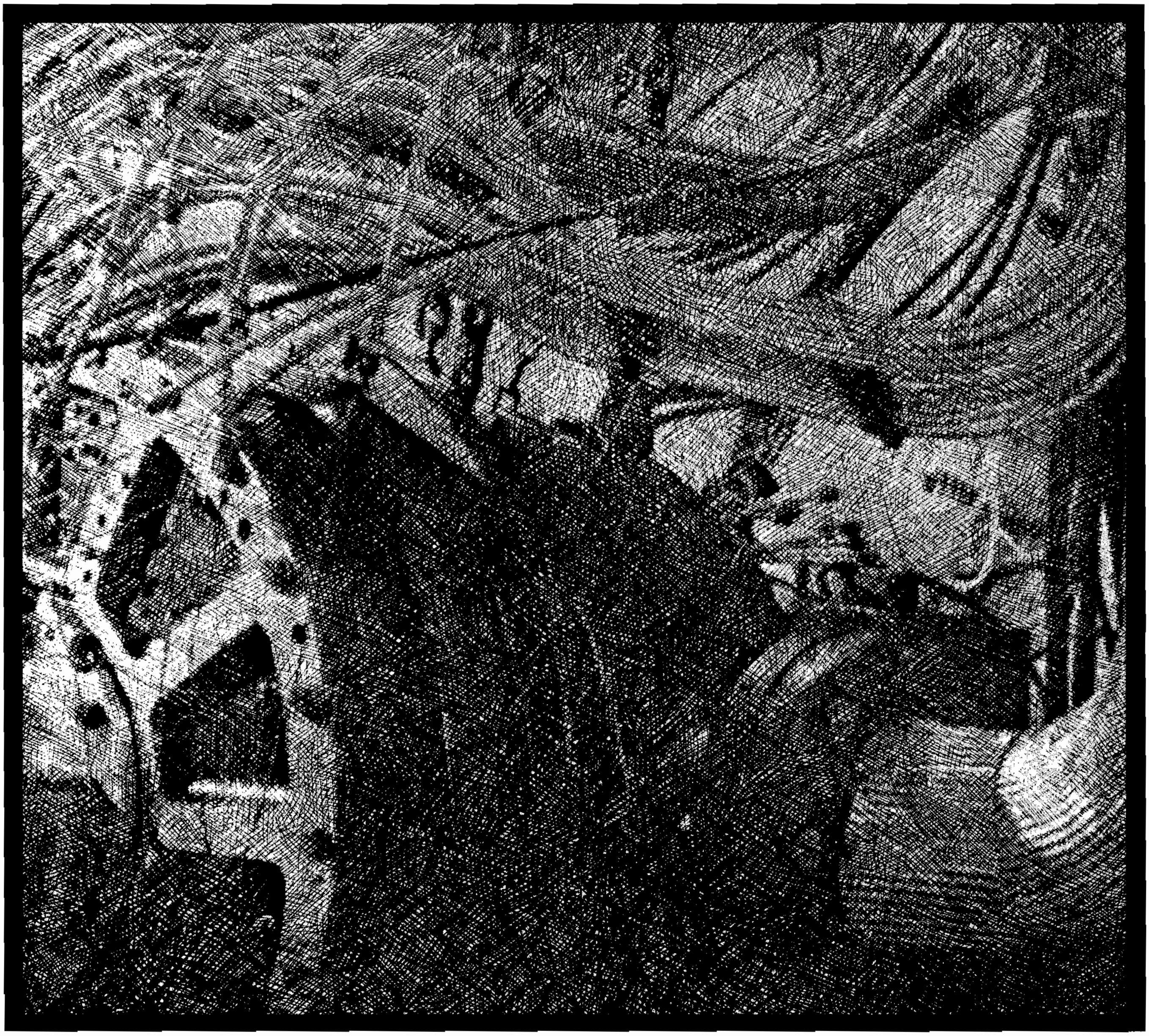
Typical chemical waste holding tanks with pH probe and neutralizer inputs.



HEWLETT hp PACKARD

For more information, call your local HP Sales Office or East (201) 265-5000 • Midwest (312) 677-0400 • South (404) 436-6181 • West (213) 877-1282. Or write: Hewlett-Packard, 1501 Page Mill Road, Palo Alto, California 94304. In Europe: P.O. Box 85, CH-1217 Meyrin 2, Geneva, Switzerland. In Japan: YHP, 1-59-1, Yoyogi, Shibuya-Ku, Tokyo, 151.
Printed in U.S.A. May 1975 5952-2474

applications
bulletin L5



3050B Turbine Measurements

At Chrysler Corporation's engineering center in Detroit an ongoing program for gas turbine development has made major contributions to the search for a low pollution alternative to the piston engine. As a part of this effort, researchers in the component test lab have increased the sophistication of their data gathering and analysis techniques.

HEWLETT hp PACKARD

Problem

By nature, turbine and compressor component tests require large amounts of data to properly characterize their operation. This data, primarily temperature and pressure, may include up to 18,000 individual measurements during the course of one test and is gathered in a test time of up to two weeks.

To manually gather test data from a component test cell formerly required that a test engineer and two technicians be present in the control room during a test; one technician to adjust and monitor test run parameters; the other to transfer temperature and pressure measurements onto "run sheets". After correction and approval the run sheets were sent to the Chrysler computer center for keypunching and processing through the data reduction program. The data processing turn around averaged one week.

As funding for additional tests increased, standard data collection techniques became far too slow to allow efficient use of the two component test cells. The problems cited by Chrysler engineers include:

1. The sheer volume of numbers required more hours in the test cells than were available.
2. With the manual system of acquisition, problems and measurement errors were not detected until after the keypunching and processing through the computer. Thus bad tests were determined to be invalid only after the completion of the test, resulting in two weeks of lost time.
3. The volume of keypunching was quite expensive and time consuming.
4. Test engineers were spending most of their time conducting test runs rather than planning new tests or evaluating data.

Solution

Several solutions to the data acquisition problem were considered, including data loggers and minicomputer systems. The final choice, however, was a Hewlett-Packard 3050A data acquisition system controlled by a 9820A calculator. The actual equipment includes:

9820A Calculator 5.8K user memory
9865A Cassette Memory unit
9862A Digital Plotter
3490A Digital Voltmeter
2895A Papertape Punch
Analog Scanner

In addition, an option to the 3050 was ordered to connect a 196 channel Scanivalve* pressure multiplexer to the system.

To multiplex signals from both test cells into the 3050, the scanner included some 300 channels, 196 pressure and 70 temperature.

Typical parameters measured by the 3050 system in the Chrysler system are in terms of voltages applied to the 3490 digital multimeter. Signal levels range from the 16 volt signal anticipated for a clearance transducer to the mV level signal from thermocouples.

The autoranging multimeter reduces the problems associated with a single-range analog to digital converter and downranges automatically to measure thermocouple output to 1 microvolt resolution.

Results

Since its installation, the 3050 has made several major changes in the compressor test lab. Test time has been reduced from two and one half weeks to three days with a 40% increase in data. Since at the conclusion of a successful run the 3050 system punches a standard format paper tape compatible with computer center readers, turnaround has been reduced to less than one day.

Although the 9820 does not do 100% of the data reduction on line, it does convert transducer inputs to engineering units, monitor test limits, output preliminary plots, and punch a tape after the run is verified successful.

The evaluation that the 9820 performs on line has been invaluable to insure that test data is valid during the test. The 3050 system is programmed to recalibrate all pressure channels twice a day, as well as flag leaky channels while the test is in progress. Among the parameters calculated by the 9820 during the test are pressure ratio, flow, efficiency, and slip factors. With this information test procedures can be adjusted during the test. A typical plot is included in figure 1:

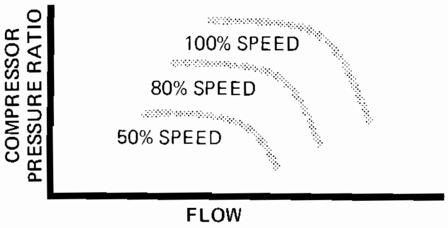
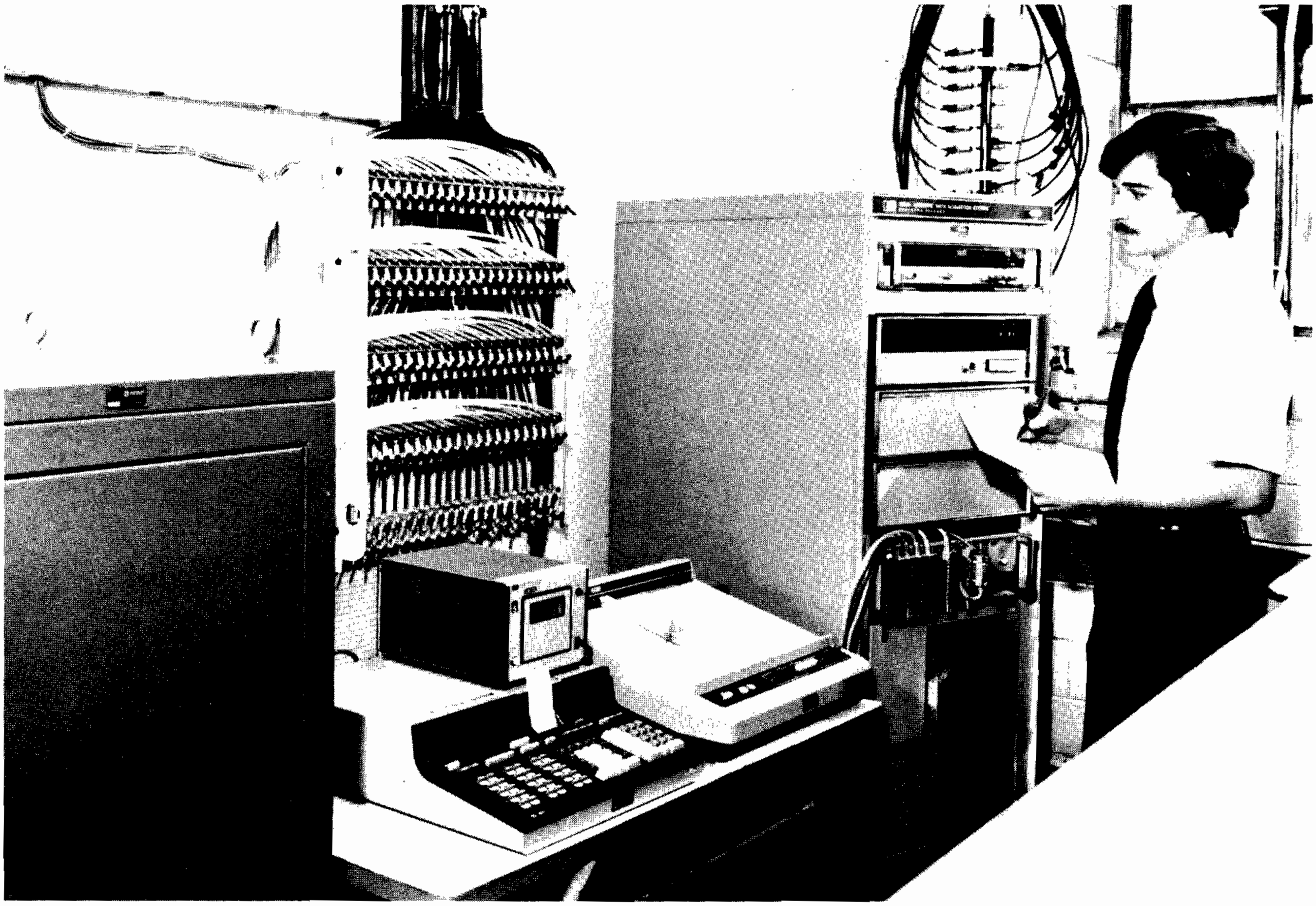


Figure 1. Efficiency vs. Work



Cost justification of the 3050 system was based on savings in keypunch time, operator time, as well as, the substantially increased output.

How do Chrysler engineers feel about a calculator based test system? According to test engineer Chris Mader, the system has reduced the time he has to spend in the control room and made one man operation a reality.

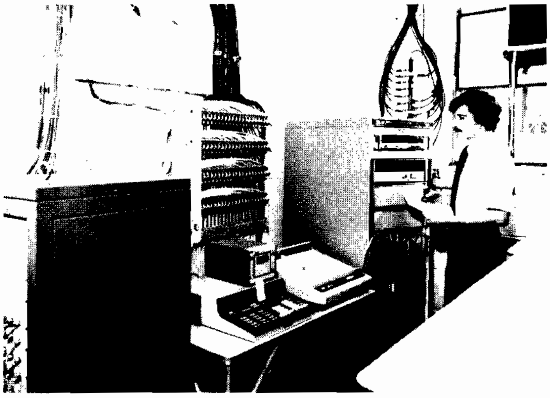
"The total program development time, from start to finish was less than six weeks," reports Mader. "It's very easy to program."

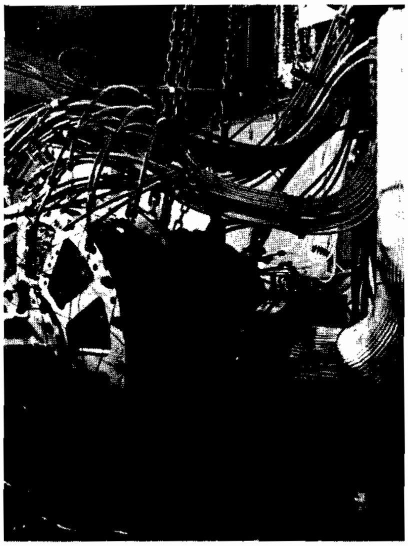
*WHAT IS A SCANIVALVE?

A Scanivalve is an electromechanical pressure multiplexer which allows one pressure transducer to be connected to many pressure lines. This avoids the use of an expensive pressure transducer for each data channel.

Scanivalve is a registered trademark of Scanivalve Incorporated, San Diego, California.

HEWLETT hp PACKARD

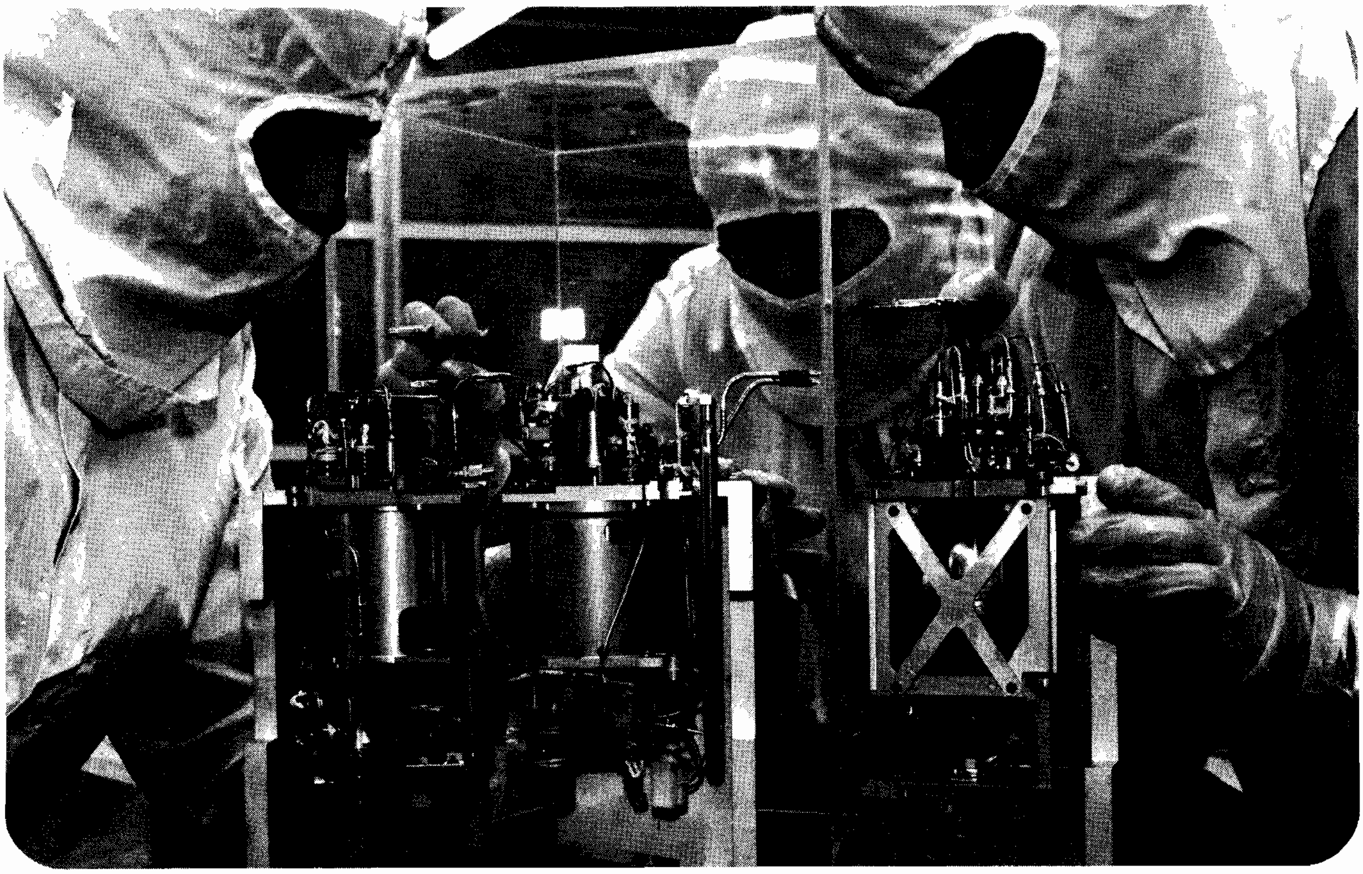




Photos courtesy of Chrysler Corporation

For more information, call your local HP Sales Office or East (201) 265-5000 • Midwest (312) 677-0400 • South (404) 436-6181 • West (213) 877-1282. Or write: Hewlett-Packard, 1501 Page Mill Road, Palo Alto, California 94304. In Europe: P.O. Box 85, CH-1217 Meyrin 2, Geneva, Switzerland. In Japan: YHP, 1-59-1, Yoyogi, Shibuya-Ku, Tokyo, 151.
Printed in U.S.A. May 1975 5952-2473

HP computers, calculators and the Interface Bus aid in the analysis of samples from Mars



HP-IB TRW's advanced technology is being applied this summer when a computerized life detection instrument package probes for life on Mars. The HP Interface Bus has contributed significantly to the development of testing programs at TRW.

At the TRW Space Park facility in Redondo Beach, California, precise measurement and testing is a critical aspect of the company's activities. As a designer and manufacturer of scientific military and commercial spacecraft systems, the Defense and Space Systems Group relies on 25 HP 2100/21MX computer-based Scientific and Measurement Systems for engineering and manufacturing testing.

Don Broutt, manager of TRW's automatic test systems department, adopted the Hewlett-Packard Interface Bus (HP-IB) to link numerous test instruments to his HP computers and HP calculators.

"Many of our programs involve extremely limited production runs . . . some products, for example, are one of a kind. Before adopting the interface bus, our efforts to reestablish test stations for each program was like reinventing the wheel. Now, when setting up a new test station with the HP-IB, we can easily add or reconfigure instruments in a computerized network with minimal set up time," explains Broutt.

With improved flexibility comes cost savings, according to Broutt. Prior to adopting the HP-IB, testing required specially engineered printed circuit board interfaces for each unique test device. If a device served as both a "listener" and "talker," it required two boards. Now, one board within a computer allows interfacing with up to 14 devices that meet the IEEE-488 standard. A single standard cable now replaces specially-engineered cables formerly required for each test instrument.

"In our pre-HP-IB testing, we wrote special driver software for each unique device. This consumed excessive amounts of computer memory. Now our engineers simply use a subroutine for each device to access an HP-IB standard driver. Gone is time-consuming reference to handbooks for device translation. Once a subroutine is written, the device interface is transparent to our engineers," relates Broutt.

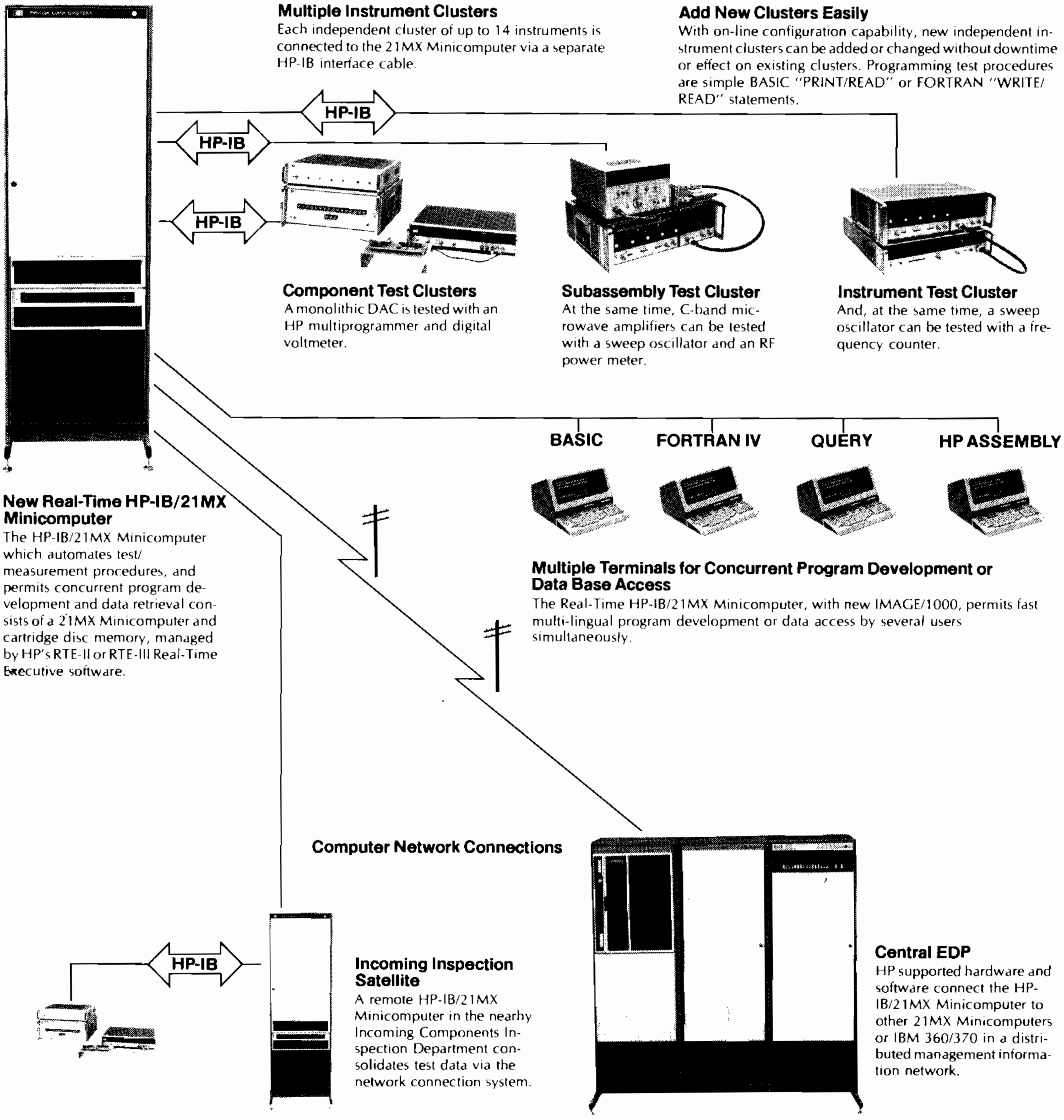
With the growing availability of test devices using IEEE-488 and the expanding use of the HP Interface Bus, TRW has reduced the cost of interface design. Manpower and resources formerly allocated to this function can now be applied to other priority projects.

● ● *COMPUTER ADVANCES from Hewlett-Packard* ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●

New HP-IB/21MX Minicomputer controls multiple instrument clusters, accesses data and develops new programs—all at the same time

In the past, interfacing instrumentation systems for measurement and test applications has been complex and costly. Not any more. With the HP-IB/21MX Minicomputer, automatic test systems for production, laboratory research plus automatic data acquisition systems can be implemented more easily and simply. HP offers for the first time a minicomputer with a multiprogramming operating system as a controller for instruments which conform to the IEEE-488 standard.

Take the simplicity of HP-IB* interfacing. Add HP's 21MX Minicomputer and Real-Time Executive (RTE) software for the power and control. Choose from over 100 HP and non-HP IEEE-488 compatible instruments, calculators and peripherals to handle test and measurement. Within hours, flexible and powerful measurement and test systems can be up and running. OEM's can focus resources on customer interfacing.



Multiple Instrument Clusters
Each independent cluster of up to 14 instruments is connected to the 21MX Minicomputer via a separate HP-IB interface cable.

Add New Clusters Easily
With on-line configuration capability, new independent instrument clusters can be added or changed without downtime or effect on existing clusters. Programming test procedures are simple BASIC "PRINT/READ" or FORTRAN "WRITE/READ" statements.

Component Test Clusters
A monolithic DAC is tested with an HP multiprogrammer and digital voltmeter.

Subassembly Test Cluster
At the same time, C-band microwave amplifiers can be tested with a sweep oscillator and an RF power meter.

Instrument Test Cluster
And, at the same time, a sweep oscillator can be tested with a frequency counter.

New Real-Time HP-IB/21MX Minicomputer
The HP-IB/21MX Minicomputer which automates test/measurement procedures, and permits concurrent program development and data retrieval consists of a 21MX Minicomputer and cartridge disc memory, managed by HP's RTE-II or RTE-III Real-Time Executive software.

Multiple Terminals for Concurrent Program Development or Data Base Access
The Real-Time HP-IB/21MX Minicomputer, with new IMAGE/1000, permits fast multi-lingual program development or data access by several users simultaneously.

Computer Network Connections

Incoming Inspection Satellite
A remote HP-IB/21MX Minicomputer in the nearby Incoming Components Inspection Department consolidates test data via the network connection system.

Central EDP
HP supported hardware and software connect the HP-IB/21MX Minicomputer to other 21MX Minicomputers or IBM 360/370 in a distributed management information network.

● ●

HEWLETT hp PACKARD

HP-IB

MINICOMPUTER
APPLICATIONS

AN201-3

A Multiple Station Electronic Test System

INTRODUCTION

This application note describes an HP 1000 Computer System with bus-connected instruments for performing component, subassembly, and final product tests at three different test stations. The HP 1000 system communicates with the respective test stations via individual Hewlett-Packard Interface Bus (HP-IB)* interface cards, each capable of interfacing up to 14 different instruments to the computer system. Tests are programmed in easy-to-use Multi-User Real-Time BASIC, which functions in Hewlett-Packard's powerful, disc-based RTE-III Real-Time Executive operating system. Although only a three test station system is described here, the RTE-III system is capable of coordinating several more test stations, each with more instruments, while supporting concurrent development of new test programs.

MEASUREMENT SETUP (Figure 1)

As noted above, the measurement setup consists of three individual clusters of test instruments or test stations, each performing a different test function or performing a different stage of testing. The individual test stations are described below. The setup also includes a color video monitor that provides graphic display capability for displaying, in turn, the test results from each of the setups. Because it is a color monitor, three 91200B TV Interface cards are required to interface it to the system. Black and white video displays require only one TV Interface card.

**The HP-IB (Hewlett-Packard Interface Bus) is Hewlett-Packard's implementation of IEEE Standard 488-1975 "Digital Interface for programmable instrumentation."*

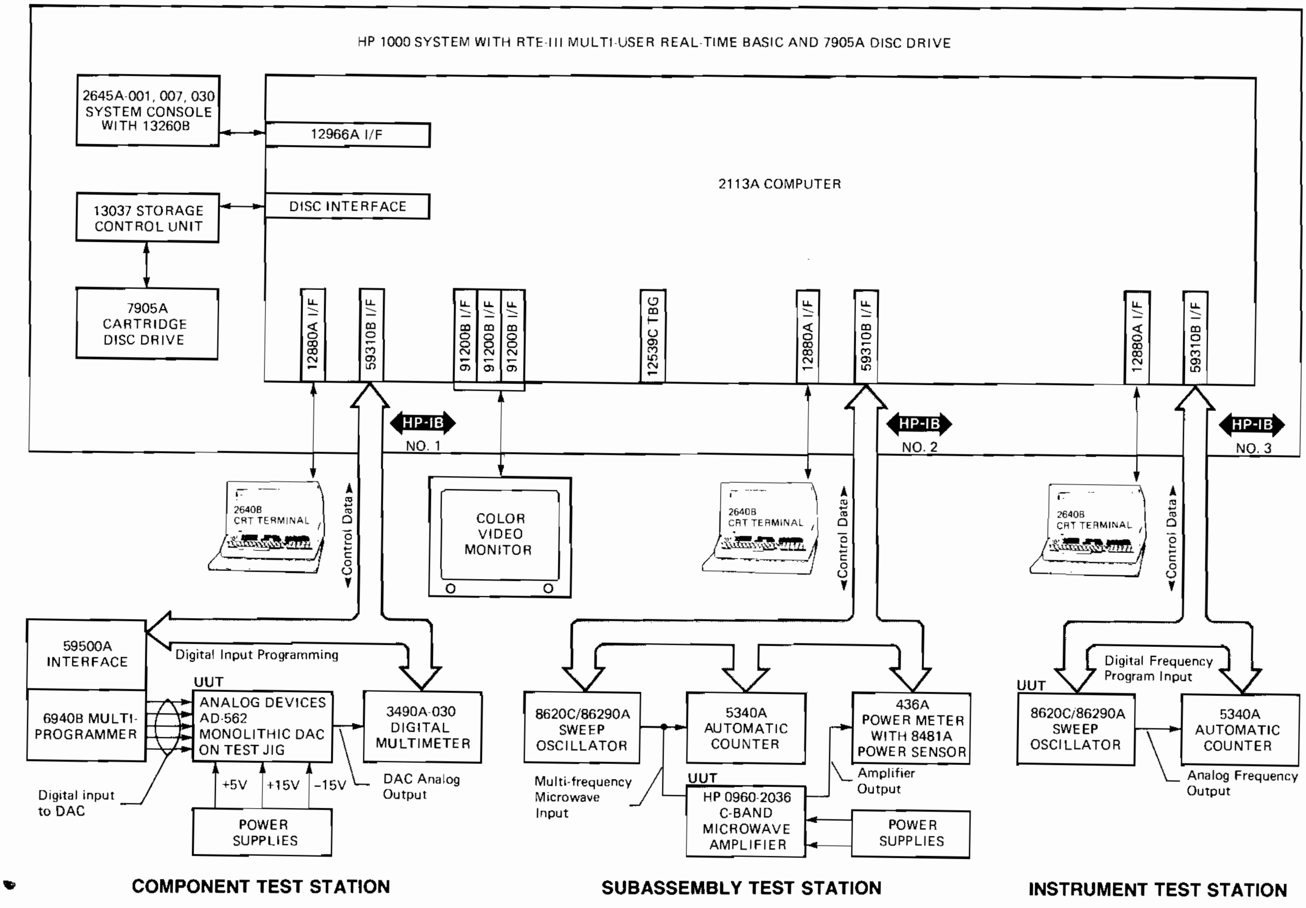


Figure 1. Multi-Station Test System

Component test station

The component test station tests the differential linearity of the analog output of a 12-bit monolithic Digital-to-Analog Converter (DAC) with respect to the digital input applied to it. The digital input is applied to the DAC via HP-IB No. 1, the 59500A HP-IB to Multiprogrammer Interface, and the 6940B Multiprogrammer as specified in the test program. The corresponding analog output is measured accurately by the 3490B Digital Multimeter. The digital result from the multimeter is returned via HP-IB No. 1 to the test program, which then outputs the data on the operator's keyboard-CRT console in numerical form and on the color video monitor as a graphic plot of the linearity of the DAC.

Subassembly test station

The subassembly test station checks the power-frequency output of a microwave amplifier with respect to its input. The amplifier input is generated by an 8620C Sweep Oscillator with 86290A RF Plug-In under control of test program commands issued via HP-IB No. 2. The amplifier input frequency is measured by a 5340A Automatic Counter and the amplifier output power is measured by a 436A Digital Power Meter with 8481A Power Sensor. Both of these measuring instruments also communicate with the test program via HP-IB No. 2. The measurement results they provide are displayed in tabular form on the operator's keyboard-CRT console and graphically on the color video monitor, ideally as a straight line at 45°

Instrument test station

The instrument test station tests the frequency accuracy of a completed product, the HP 8620C Sweep Oscillator. The 8620C is programmed via HP-IB No. 3 to a succession of frequencies as specified in the test program. These are measured individually by a 5340A Automatic Counter that also communicates with the test program via HP-IB No. 3. The results are displayed in tabular form on the operator's keyboard-CRT console. Actual frequency vs programmed frequency is plotted on the color video monitor, appearing normally as a straight line at 45°.

INTERFACING

The interfacing is shown in Figure 1. Three HP 59310B HP-IB interfaces are used to connect the three test stations to the system. The three HP 91200B TV Interface cards must be in physically-adjacent I/O slots to match the three-connector cable assembly used for connection to the color video monitor.

SYSTEM GENERATION

The computer I/O slots used and driver routines called by the user's programs are logically connected to the RTE-III operating system of the HP 1000 system during system generation. The system generation process is covered in the RTE-III Programming and Operating Manual (92060-93001) and the RTE-II/RTE-III On-Line Generator Reference Manual (92060-90020). The logical unit assignments (Table 1) can also be made at system generation time, or later if desired, and can be changed at any time without the necessity of regenerating the system. This maximizes flexibility for the user and minimizes the effort required to implement new or changed test setups.

Table 1. Logical Unit Assignments

Test Station	Unit	HP-IB Device Addresses			LU No.
		ASCII	Octal	Type	
1	HP-IB Bus No. 1				30
	59500A Multiprogrammer Interface	W 7	127 67	Talk Listen	27 27
	3490A Digital Multimeter	V 6	126 66	Talk Listen	
	2640B CRT Terminal	N.A.	N.A.	N.A.	7
2	8620C Sweep Oscillator	Z	132	Listen	40
	5340A Automatic Counter	C #	103 43	Talk Listen	48 48
	436A Power Meter	M --	135 55	Talk Listen	38 38
	2640B CRT Terminal	N.A.	N.A.	N.A.	9
3	8620C Sweep Oscillator (Unit Under Test)	Z	132	Listen	41
	5340A Automatic Counter	C #	103 43	Talk Listen	39 39
	2640B CRT Terminal	N.A.	N.A.	N.A.	1
Color Display	Three 91200B TV Interface cards	N.A.	N.A.	N.A.	12

OPERATION

After the system has been set up and the programs (facing page) have been entered, the various test stations are operated as follows:

1. The operator connects the UUT to the test setup.
2. In answer to the Multi-User Real-Time BASIC prompt of >, the operator types RUN TEST 1, RUN TEST 2, or RUN TEST 3, as appropriate.
3. When requested by the program, the operator types in the UUT serial number.
4. Entry of the UUT serial number starts the program, resulting in the display of numerical data on the keyboard-CRT and graphic display on the color video monitor.

TEST PROGRAM LISTINGS

The simple program statements shown highlighted in the listings of each test station's program below accomplish all HP-IB control and data input-output functions for the various tests.

NOTE: Subroutine SUBR, which processes data for graphic presentation on the color video display is not shown here because it is beyond the scope of this application note. DCODE, CMDW, and CMDR are standard calls to the HP-IB Driver Subroutine from Real-Time BASIC.

TEST 1 REAL-TIME BASIC PROGRAM

```
10   DIM A$[11],B$[14],D$[8],M$[6],Q$[4],T$[4],A[210],E$[10]
20   REM---------------THIS IS TEST1-------------------
30   PRINT "INPUT DAC SERIAL #"
40   INPUT A[210]
50   LET B=30
60   LET M=27
70   LET B$="@@@@@@@@@@@@@@"
80   LET M$="@0000T"
90   LET Q$="(I4)"
100   LET E$="(4X,E10.0)"
110   REM** SET BUS
120   CALL HPIB(B,14,0)
130   REM** SET SCANNER CHANNEL
140   PRINT #M;"00140T"
150   REM** INCREMENT DAC/READ DMM
160   LET K=1
170   FOR N=20 TO 4000 STEP 20
180   LET X=OCT(N)
190   CALL DCODE(X,M$[2,5],Q$)
200   PRINT #M;M$
210   CALL CMDW(B,"?6P","R4F0S0T1M3E")
220   CALL CMDR(B,"?0V",B$)
230   CALL DCODE(B$,G,E$)
240   PRINT "MEASURED",G,"VOLTS"
250   LET A[K]=(G*20)+40
260   LET K=K+1
270   NEXT N
280   REM** CLR BUS
290   CALL HPIB(B,15,0)
300   REM** SEND DATA TO DISPLAY
310   CALL SUBR(A[1],1)
320   END
```

TEST 2 REAL-TIME BASIC PROGRAM

```
10   DIM S$[11],T$[5],Q$[6],I$[7],C$[16],D$[10],P$[12],A[210]
20   REM---------THIS IS TEST2---------------
30   PRINT "*********** AMPLIFIER POWER OUTPUT TEST ***************"
40   PRINT
50   PRINT
60   PRINT
70   PRINT "INPUT SERIAL NUMBER OF UUT."
80   PRINT
90   PRINT
100   INPUT A[210]
110   REM** LU#'S B2=HPIB#2  S=SWEEPER
120   REM**         C=COUNTER  P=PWR METER
130   LET B2=31
140   REM*******LU40 IS SPARE FOR SWEEPER #2 *****************
150   REM**************CHANGE LU SUBCHANNEL PER HARDWARE SETUP****
160   LET S=40
170   LET C=48
180   LET P=38
190   REM** P$= PWR MTR INPUT DATA
200   LET P$="SRM+DDDDE-VV"
210   REM** S$= SWEEPER CONTROL DATA
220   LET S$="M1B1VNNNNNE"
230   LET D$="(3X.E9.0)"
240   LET Q$="(F5,3)"
250   REM** ENABLE HPIB
260   CALL HPIB(B2,14,0)
270   REM**  SET UP PWR MTR
280   PRINT #P;"9D+R"
290   REM**  DEFINE SWEEPER RANGE
300   LET F0=2
310   LET F9=6.2
320   REM** K= DATA COUNTER
330   LET K=1
340   REM**  SET UP FREQ COUNTER
350   PRINT #C;"3P@JLOH"
360   REM** FREQ SWEEP & MEAS PWR ROUTINE
370   FOR F=2 TO 6.2 STEP .025
380   REM**   DEFINE & SET FREQ ON SWEEPER
390   LET V=((F-F0)/(F9-F0))*10
400   CALL DCODE(V,S$[6,10],Q$)
410   PRINT #S;S$
420   WAIT (25)
430   REM**  READ PWR MTR
440   READ #P;P$
450   REM**  PRINT PWR LVL & FREQ
460   CALL DCODE(P$,G,D$)
470   PRINT G,"FREQ",F
480   REM**  FILL PLOTTING DATA ARRAY
490   LET A[K]=(G+.05)*10+80
500   REM**  INCREMENT DATA COUNTER
510   LET K=K+1
520   NEXT F
530   REM** HPIB DISABLE
540   CALL HPIB(B2,15,0)
550   REM**  XFER DATA TO DISPLAY
560   CALL SUBR(A[1],2)
570   END
```

TEST 3 REAL-TIME BASIC PROGRAM

```
10   DIM S$[11],T$[5],Q$[6],I$[7],C$[16],D$[10],A[210]
20   REM-----------THIS IS TEST3---------------
30   PRINT "************SWEEPER LINEARITY TEST*********************"
35   PRINT
40   PRINT
45   PRINT
50   PRINT "INPUT SERIAL NUMBER OF UUT."
55   PRINT
59   INPUT A[210]
60   REM**  B=HPIB LU#  C=COUNTER LU#  S=SWEEPER LU#
70   LET C=39
80   LET B=31
90   LET S=41
100   REM** S$=SWEEPER PARAMETERS  C$=COUNTER DATA
110   LET S$="M1B4VNNNNNE"
120   LET C$="NNNNNNNNNNNNNNNN"
130   REM** D$=SWEEPER DATA FORMAT  Q$=COUNTER DATA FORMAT
140   LET D$="(3X,E11.0)"
150   LET Q$="(F5.3)"
160   REM** ENABLE HPIB
170   CALL HPIB(B,14,0)
180   REM** SET UP COUNTER
190   PRINT #C;"4P@JLOH"
200   REM** K=COUNTER FOR DATA POINTS
210   LET K=1
220   REM** INITIALIZE SWEEPER TO FIRST SETTING
230   LET V=0
240   CALL DCODE(V,S$[6,10],Q$)
250   PRINT #S;S$
260   REM** INITIALIZE COUNTER
270   REM**  SET SWEEP FREQ'S & INCREMENT
280   FOR F=2 TO 18.05 STEP .1
290   REM**CALC SWEEPER FREQ PARAMETER
300   LET V=(F-2)/1.6
310   CALL DCODE(V,S$[6,10],Q$)
320   REM** SET SWEEPER FREQ
330   PRINT #S;S$
340   REM** ALLOW SWEEPER TO SETTLE
350   WAIT (35)
360   REM** 250-270,330 ALLOW LONGER SETTLING TIME IF REQD
370   LET Z=0
380   WAIT (10)
390   LET Z=Z+1
400   REM** READ COUNTER
410   READ #C;C$
420   READ #C;C$
430   CALL DCODE(C$,G,D$)
440   PRINT "PROGR. FREQ=",F,"MEASURED FREQ=",G
450   REM** FILL PLOTTING DATA ARRAY
460   REM**  Y0=PLOT OFFSET
470   LET Y0=30.5
480   LET A[K]=(G/1.00000E+08)+Y0
490   IF F=2 OR Z=5 GOTO 520
500   IF ABS(ABS(A[K]-A[K-1])-1)>1 GOTO 380
510   REM** INCREMENT DATA COUNTER
520   LET K=K+1
530   NEXT F
540   REM** DISABLE HPIB
550   CALL HPIB(B,15,0)
560   REM** DUMP DATA TO DISPLAY
570   CALL SUBR(A[1],3)
580   END
```

ADVANTAGES OF HP-IB MINICOMPUTER SYSTEMS

One interface serves up to 14 devices

Prior to HP-IB interfacing, systems like that described in this application note required at least one interface card per instrument communicating with the computer. The HP-IB interface, on the other hand, can serve as many as 14 instruments or devices, which are connected using standardized bus cables that provide for convenient piggyback connection.

Multi-bus operation

The 59310B HP-IB interface card is available with either BCS or RTE software. The RTE software supports multiprogramming operation in HP's disc-based RTE-II and RTE-III systems. Multiprogramming facilitates multi-bus operations like those described in this application note, in which one program controls and uses the instruments and devices on one bus while other programs are using the devices on other Hewlett-Packard Interface Buses. In this way, several different test or lab data acquisition functions can be controlled by a single disc-based computer system at the same time.

Choice of program languages

The RTE software package for HP-IB supports programming in Multi-User Real-Time BASIC* (see program listings above), FORTRAN IV, FORTRAN II, or HP Assembly language.

**In RTE-II/III system equipped with the 92101A Multi-User Real-Time BASIC System.*

Easy changeover from one setup to the next

Prior to HP-IB interfacing, the addition, deletion, or changing of instruments in a setup required not only a change of interface, but also regeneration of the operating system. The difficulty of changeover tended to limit the application of computer systems to longer-term jobs. Now, however, the HP-IB interface has made change-over of system setups easier than ever before, making it practical to apply the power of computer automated testing, data acquisition, and on-line data processing to many more jobs.

With the HP-IB interface, change of instruments in a set-up requires only temporary suspension of bus activity on the affected bus while the bus cable connections are made and the appropriate talk/listen addresses are set in the new instruments. Instead of stopping the computer system for system regeneration, all that is needed is re-definition of device addresses† for the RTE-II/III operating system, which is an on-line process that does not interrupt activity on other HP-IB buses or other system operations.

Of course the operations of the new setup must be programmed, but that would be required in any case. Moreover, because program development is an on-line capability in RTE-II/III systems, it can be carried on concurrently with existing bus operations prior to the changeover, so they need not be disturbed until the last minute.

†On-line redefinition of device addresses requires the pre-definition of spare logical unit numbers during system generation and the allocation of sufficient memory for extension of the equipment tables of the RTE-II/III operating system.

HEWLETT PACKARD

Sales and service from 172 offices in 65 countries.
1501 Page Mill Road, Palo Alto, California 94304

Printed in U.S.A. October 1976 5952-1686(22)

NOTES

NOTES