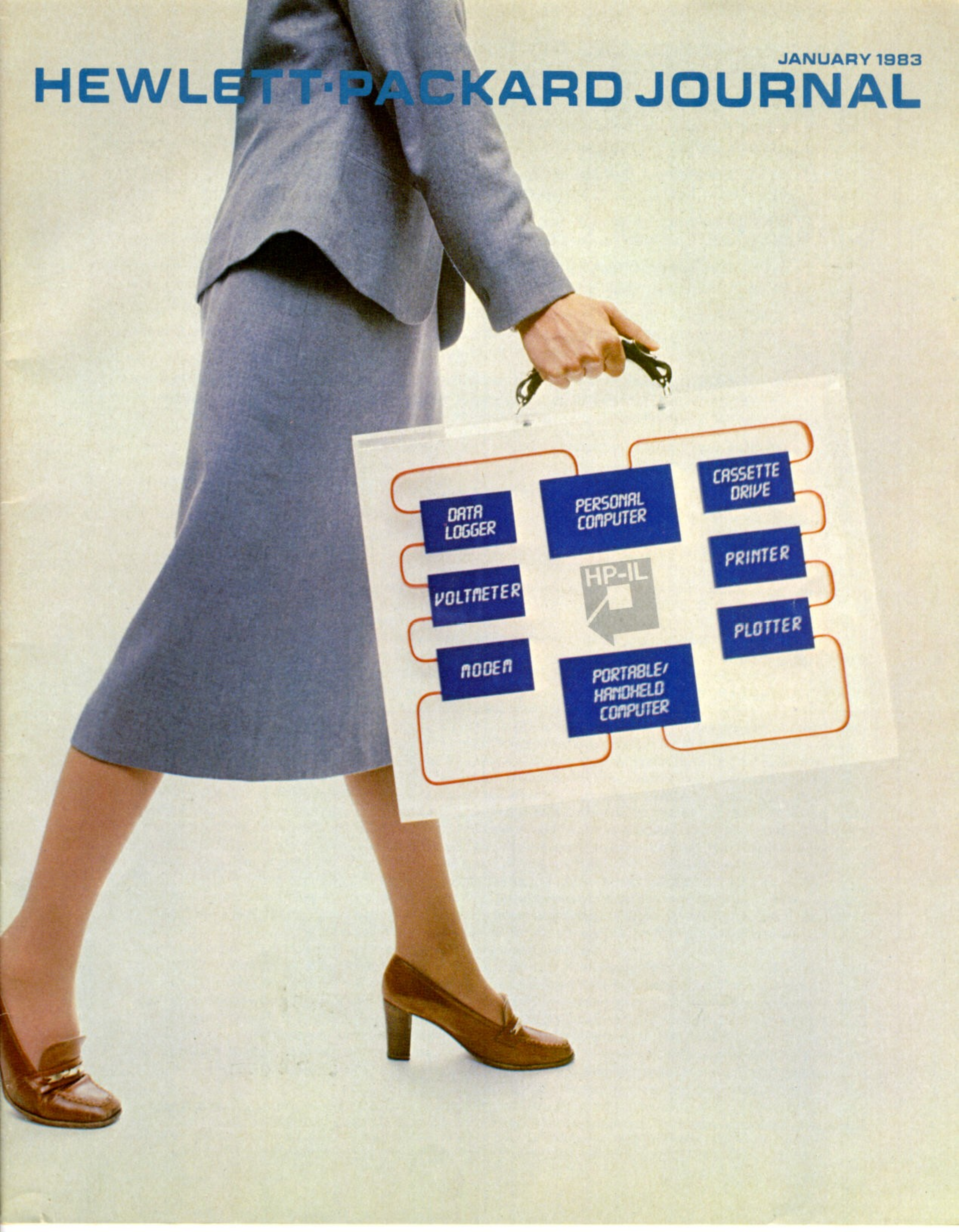
JANUARY 1983

HEWLETT-PACKARD JOURNAL



HEWLETT·PACKARD JOURNAL

Technical Information from the Laboratories of Hewlett-Packard Company

Contents:

JANUARY 1983 Volume 34 • Number 1

In this Issue:

Portable, low-cost, battery powered, plug-together systems, some small enough to fit in a briefcase, others larger and augmented with more conventional products such as personal computers and plotters—we can expect to see more and more of these systems, because thanks to a new interface system called the Hewlett-Packard Interface Loop, or HP-IL, they're going to be much easier to put together. When instruments, computers, and peripheral devices are designed according to the specifications of the HP-IL, they can exchange data, commands, and other messages using only two wires. The HP-IL is called a loop because that's the way the wires are connected—out of one device and into the second, then out of the second and into the third, and so on back to the first device.

The design of the HP-IL is based on the Hewlett-Packard Interface Bus, or HP-IB, an industrywide standard interface for higher-performance systems, also known as Standard 488 of the Institute of Electrical and Electronics Engineers. Because the HP-IB makes automated measurement and control so much more available, its impact on the ways such things are done has been tremendous. The HP-IL's impact is likely to be of similar magnitude. What the devices on the HP-IL do is similar to what the devices on the HP-IB do, but how they do it is very different. The differences between the HP-IL and the HP-IB and the details of the design and operation of the HP-IL are discussed on pages 3 through 22 of this issue. Our cover photograph shows our art director's conception of an HP-IL system in a briefcase. A little artistic license has been taken—while all of the devices shown can operate on the HP-IL, not all are battery powered and briefcase portable.

Another in our series of articles on processes used at HP to produce custom integrated circuits begins on page 23. This process is called CMOSC, for complementary metal-oxide-semiconductor version C. It's used by HP's Corvallis, Oregon Division to produce ICs for HP Series 10 handheld calculators.

This issue also carries two articles on gas chromatographs from HP's analytical instruments division in Avondale, Pennsylvania. When a sample is injected into one of these instruments, it is mixed with a stream of carrier gas and transported through a heated tube called a column. To ensure accurate results, the column temperature and the carrier gas flow rate and pressure all have to be accurately controlled. The article on page 30 describes a new gas chromatograph, Model 5790A, that gives much better control of column temperature than previous instruments. While the 5790A is designed for high-volume industrial and clinical use, the 5880A Gas Chromatograph is HP's top-of-the-line research-quality instrument. The article on page 35 discusses the design of a precise new electronic flow and pressure controller for the 5880A. On page 32, Fred Rowland gives us some basic information about gas chromatography.

-R. P. Dolan

Editor, Richard P. Dolan • Associate Editor, Kenneth A. Shaw • Art Director, Photographer, Arvid A. Danielson • Illustrator, Nancy S. Vanderbloom
Administrative Services, Typography, Anne S. LoPresti, Susan E. Wright • European Production Supervisor, Henk Van Lammeren

HP-IL: A Low-Cost Digital Interface for Portable Applications

The Hewlett-Packard Interface Loop is a bit-serial interface bringing many capabilities formerly reserved for much larger computer systems to the growing repertoire of portable computers and handheld calculators.

by Roger D. Quick and Steven L. Harper

IN 1976, SEVERAL DIVISIONS of Hewlett-Packard began to perceive the need for a low-cost, low-power, digital interface standard. Progress in integrated circuit technology had allowed development of small, low-cost, but fully functional instruments, calculators, and computers. The small size of these products allowed them to be considered portable and many were capable of being battery operated. The cost to add a standard interface to these devices could be small, but only if the interface were compatible in physical size and power consumption.

The existing applicable interface standards in 1976 were the HP-IB (Hewlett-Packard Interface Bus, HP's implementation of IEEE Standard 488) and RS-232-C/CCITT V.24. Both use many parallel conductors, and so their connectors are large, requiring 24 and 25 pins, respectively. Both of these interfaces use bipolar device technology, which because of inherently high current drain, consumes considerable power. The large connector size and power requirements are obviously not compatible with small, battery-operated devices such as the HP-41C Programmable Calculator.

Because size, cost, and power consumption are critical requirements for portable, low-power systems, a different interface design is required. Since many applications do not require the high-speed performance of the HP-IB, a slower, simple, bit-serial, two-wire link between devices was chosen. This choice and the design of a new miniature connector system provided the solution to the size problem and helped alleviate the power consumption and cost difficulties.

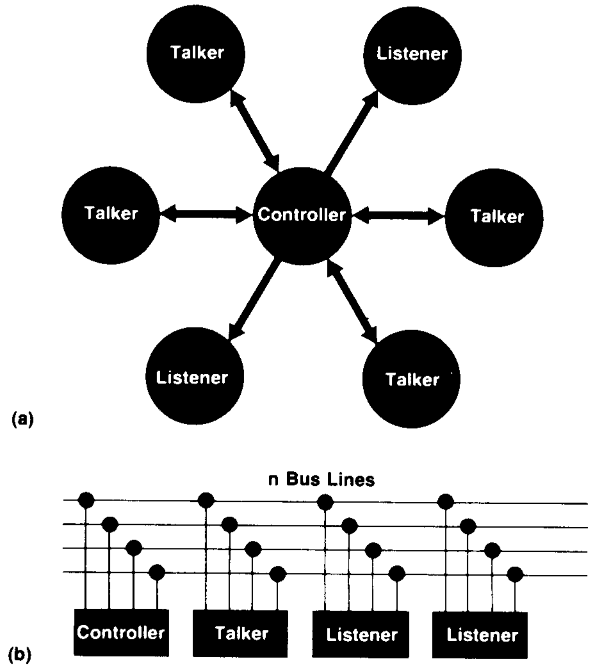


Fig. 1. *(a) Star network architecture. The controller located at the center must have an I/O port for each of the other devices on this interface system. (b) Parallel bus network architecture. Each device on the bus must be able to drive the combined load presented by the other devices on the bus.*

System Architecture

In any interface system, it is very important to be able to connect more than just two devices together. For example, the point-to-point structure of RS-232-C/CCITT V.24 suggests a star network (Fig. 1a). However, this structure is not practical for portable applications because it requires the system controller to have a separate connector for each device. In addition to the added cost, this would tie up all the I/O ports in a small device such as the HP-41C that users might want to use for other accessories.

The parallel bus structure (Fig. 1b) of the HP-IB is much more inviting since only one port or connector per device is required. However, the electrical design problems are more troublesome. Each device's interface output must be able to drive a large assortment of system configurations ranging from one other device connected by a short cable to as many as thirty devices located some distance away. The cost and power requirements are too great for portable, battery-powered devices.

The system architecture that resolves these difficulties is a unidirectional loop (Fig. 2). Each device has only one interface connector with two sockets: IN and OUT. Each output drives only one input regardless of the number of devices connected in the system. The electrical design problem becomes much simpler, longer distances between devices are possible, and the power to drive the interface is minimized and shared among all of the devices. Low-cost two-wire cable allows a distance of up to 10 meters between devices on the loop and work is progressing on a special

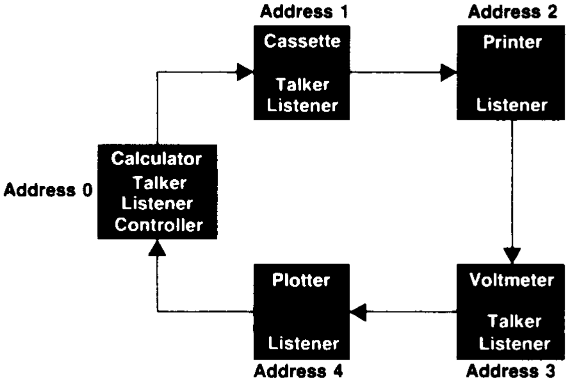


Fig. 2. *Typical Hewlett-Packard Interface Loop system. Messages are circulated around a loop. Only controllers and talkers can originate a message but all devices can retransmit a received message. Each device receives the message, acts upon it if required, and retransmits it to the next device until the message returns to its originator. Thus, each HP-IL device only has to drive one other device, regardless of the number of devices on the loop.*

cable to permit device-to-device spacing up to 100 meters.

The loop structure has one disadvantage. Because the data must pass through each device and return to its source, all devices in the system must be powered up and fully functional. If one device fails, it is likely that the entire system will not work.

Interface Electronics

Once the basic architecture was chosen, work proceeded to design a low-cost, low-power, electronic interface to the system. The combination of a custom CMOS (complementary metal-oxide-semiconductor) integrated circuit and miniature pulse transformers is the result. The electronic interface links each device to the next through a pair of wires using a floating, balanced, differential voltage mode of operation. This provides good noise immunity and reduces EMI (electromagnetic interference). Eliminating the need for a system ground avoids the problems often associated with ground loops and makes it easy for devices to float with respect to earth ground, a feature especially convenient for devices like voltmeters. The electronic design of the interface is discussed in the article on page 11. The design of the custom CMOS IC is described in the article on page 16.

Functions

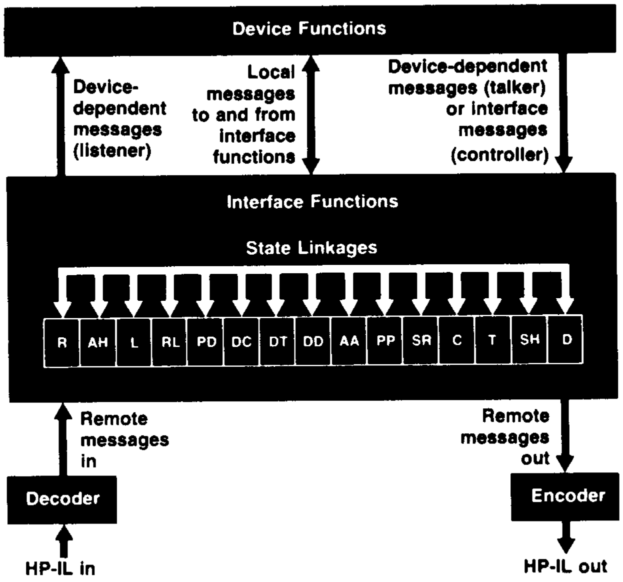
While size and power considerations required new interface electronics, the functionality of a new interface was a separate question. Experience with the HP-IB standard has shown it to be flexible but complete. Diverse products built to the HP-IB standard are able to communicate without anomalies. To perpetuate the HP-IB functionality in a lower-performance serial interface is obviously attractive, and it was with this objective that the Hewlett-Packard Interface Loop (HP-IL) was created.

The functions within a device divide logically into two categories; device functions and interface functions (Fig. 3). The device functions are special to each type of device. The designer can choose and implement these functions in any way considered appropriate. Therefore, the device functions in a voltmeter, for example, will have little similarity to those, say, of a flexible disc drive. The interface functions handle the communication of messages between the device and the rest of the interface system. Consequently, the interface functions in each device must be the same to maintain compatibility and the designer must implement these functions in strict accordance with the interface's protocol specification.

Like the HP-IB, the HP-IL is a master-slave system. One device is designated as the system controller and is responsible for bringing up the system when the power is turned on. This controller sends commands to configure and control the loop and initiate transmission of data from one device to another. The controller interface function is active in only one device at a time on the loop. Protocol is provided to allow passing of the controller responsibility to another device so that multiple controller devices can take turns being in charge of the interface system.

The talker interface function permits a device to supply data when instructed to do so by the active controller. There can be only one active talker on the loop at a time. The listener function allows devices to receive the talker's data, but only at the controller's command. There can be multiple listeners at one time on the loop. Often a device will have both talker and listener capabilities. Controllers almost always have the talker and listener functions.

In a typical loop operation, the controller might begin by sending an unlisten command to disable previously desig-


Device Functions
Device-dependent messages (listener)
Local messages to and from interface functions
Device-dependent messages (talker) or interface messages (controller)
Interface Functions
State Linkages
R
AH
L
RL
PD
DC
DT
DD
AA
PP
SR
C
T
SH
D
Remote messages in
Remote messages out
Decoder
Encoder
HP-IL in
HP-IL out

Interface Functions:

R	receiver	RL	remote local
D	driver	AA	automatic address
AH	acceptor handshake	PD	power down
SH	source handshake	PP	parallel poll
C	controller	DC	device clear
T	talker	DT	device trigger
L	listener	DD	device dependent command
SR	service request		

Fig. 3. *HP-IL functional partitions.*

nated listeners. This would be followed by a listen address command to make the device that has the matching address the active listener. Then the talk address command is sent to specify which device is to supply the data. This is followed by a special message to begin the data transmission. The talker intercepts this message and replaces it with the string of data bytes for the listener(s). After the last byte of data, the talker sends another special message back to the controller indicating completion and the controller takes over and replaces that message with its next command to the system.

In addition to the basic controller, talker, and listener capabilities, there are other interface functions to handle handshaking of messages into and out of each device. A device can use two different methods to notify the controller that the device needs attention. The device functions can also be cleared or triggered on command from the controller and can be instructed to respond to local device controls or to equivalent control messages on the loop.

Familiarity with the HP-IB is helpful in understanding the HP-IL, and many users will recognize the strong similarities in the preceding discussion. Logically, the two interfaces are very close. The serial loop structure of the HP-IL, however, causes significant differences at the implementation level.

The HP-IL interface functions are specified in the same manner as IEEE Standard 488, that is, by state diagrams which model the behavior of each function. While this method is perhaps difficult for the neophyte to understand, the precision and conciseness of this form of specification are very difficult to achieve in other ways.

Once the "serial HP-IB" concept was established, other attributes of the HP-IL were filled in. The loop structure chosen for the HP-IL allows an asynchronous handshake similar to that used for the HP-IB. An eleven-bit message frame provides for retention of the attention (command mode when true, data mode when false) and service-request functions of the HP-IB. Thus the HP-IB provided the nucleus for the HP-IL and allowed rapid but safe definition of the functions of the new interface.

Message Encoding

The HP-IB has an eight-bit data bus and several control lines whereas the HP-IL must handle these functions over a single bit-serial link. To do this, each HP-IL message is contained in an eleven-bit frame. The first three bits (C2, C1, and C0) carry the control information and the following eight bits (D7 through D0) specify the particular message or data.

Because much of the same information flows on the HP-IL as on the HP-IB, the device address range is the same. However, since the HP-IL does not have the electrical loading limitations of the HP-IB, the loop can handle up to 31 devices at one time using simple addresses. If devices capable of accepting double addresses are used, as many as 961 devices can be on the loop at the same time.

The bits are encoded using a three-level code sometimes known as the pulse bipolar code (see Fig. 2 and Fig. 3 on page 12). A logical one consists of a positive pulse followed by a negative pulse and a short idle period. A logical zero simply uses the reverse order of the pulses. This code trades off some density for simple timing requirements, good noise immunity, and no dc component (necessary for transformer coupling). The first bit (C2) of each frame is coded with double pulses. This is useful since slightly different clock rates in devices around the loop require that even the bits within a message frame be asynchronous with respect to each other. The start-of-frame reference provided by the first bit helps keep the bit count in synchronization in all system devices.

The three control bits indicate one of four major classes of messages (see Table I) and provide end-of-record and service-request information. If the first bit is a zero, a data message is indicated. This permits the earliest possible decoding so that idle devices on the loop can immediately begin to retransmit a data frame to the next device on the loop, thus maximizing loop speed for data.

The other three message classes are command messages, ready messages, and identify messages. The purposes of data and command messages are obvious and identical to those for the HP-IB. Ready messages are a special class for handling certain handshake tasks on the serial loop. Identify messages are the means for performing the parallel poll function on the HP-IL.

Generally there is only one message frame on the loop at a time. The handshake sequence for most messages is quite simple. The originating device, whether it be a controller or talker, is required to wait until the current frame returns to its source before initiating the next message. The designated receiving device simply stores the current frame until the device is ready for the next frame. Then the receiving device sends the current frame back to its originator. In this manner, slow and fast devices can coexist quite nicely on the HP-IL.

This handshake sequence works well when there are only one or two receiving devices since the idle devices on the loop pass on the message frame very rapidly. However, in situations where all of the devices on the loop must respond to a message, loop speed suffers because of the serial nature of this handshake process. This is commonly the case for command messages. To alleviate this problem, the sequence for commands is modified somewhat.

Commands are required to be immediately passed on by all devices. This permits the loop devices to execute a command more or less in parallel. The return of the command message to the controller does not indicate that the devices are ready for another command in this case. Consequently, after each command the controller sends a ready-for-command (RFC) message which must be held by

Table I
HP-IL Control Bit Coding

C2	C1	C0	Message Class
0	End of Record	Service Request	Data
1	0	0	Command
1	0	1	Ready
1	1	Service Request	Identify

each device in turn until that device is ready. When the RFC message returns to the controller, the next command can be sent.

Functional Definition

It is unusual for a development project to begin with a fully formed functional definition. It is even more unusual for that definition to remain relatively unchanged throughout the development. Yet the HP-IL development followed just such a stable path. The reason for this stability was the HP-IB standard. The use of the HP-IB model for the HP-IL allowed several different groups to proceed in an orderly and efficient manner while developing products that would use the new interface.

While most of the HP-IL functional definition relied on the HP-IB model, two areas required further definition. The first of these centered on the concept of friendliness which differs according to the type of user. To a laboratory engineer, friendly typically means the ability to tailor the operation of an instrument or interface to the engineer's specific needs using as few commands as possible. However, if necessary, the engineer wants access to the system's lowest control level, to be able to "twiddle bits" to achieve a desired result. Most HP-IB implementations allow this level of control.

At the other end of the user spectrum is the average operator of the consumer calculator. This user often does not want to learn anything about the operation of the system that is providing solutions. Forcing this user to learn about address and control codes to achieve a desired result is a deterrent to the use of a system.

Added Interface Functions

To allow this kind of friendly use, some other interface functions which have no counterpart in the HP-IB were added to the HP-IL definition. Portable systems are often battery powered and conserving that power is usually very important. HP-IL devices can implement a power-down function if needed so that the HP-IL controller can command a device to enter the powered-down state and cause it to return to full power. With a real-time clock and a wakeup function added to the controller, the power-down function can permit unattended remote operation of an HP-IL system for extended periods.

Another of the added functions is the autoaddress function. The HP-IB requires the user to set device addresses manually by means of switches on each HP-IB device. It is the HP-IB user's responsibility to ensure that no two devices have the same address. The loop structure of the HP-IL, on the other hand, allows the controller to assign sequential addresses automatically to the devices on the loop.

The controller can find a particular device on the loop with the device ID function. The controller sends the talk address command followed by a special message that causes the device to send a string of characters containing, for example, the device's model number. The accessory ID function is similar, but it allows the device to return a single-byte code indicating the type of device and its capabilities.

Combined with the autoaddress function, the accessory ID function allows the implementation of HP-IL systems that relieve the user and the user's programs of the problems associated with fixed device addresses and fixed device assignments. For example, an HP-IL device can inform the controller that it is an ASCII* printer or a 16-sector flexible disc. Thus, an HP-IL user can command the controller to "Print a File" and the controller software will find an appropriate device and perform the operation. The user is not involved in any specific system details of the operation and can concentrate on the problem rather than on the system that is solving the problem.

Local Area Network Operation

The second functional area of the HP-IL that required further definition was the relationship between the HP-IL and the emerging concept of a local area network. The local area network typically has a number of equally capable devices contending for access to the physical media, while the HP-IB and the HP-IL use one of these devices (the controller) to direct traffic on the interface. For some applications, the local area network operation offers simplicity of software support and operation. An example is a system of several smart terminals connected to one mass-storage device. In a local area network environment, all terminals have the same support software, but in an HP-IL environment, one terminal must keep track of I/O operations for the others.

Rather than change the master-slave nature of interface control on the HP-IL, capabilities are present to allow systems like the local area terminal network to be implemented easily. In particular, devices on the HP-IL may asynchronously generate service-request messages. With this capability in effect, a controller may totally ignore the interface, tending its own applications, until some device on the loop notifies it that an I/O operation is desired. The controller then must get involved, but only for a temporary period to do a specified transfer. Thus, for many applications, the HP-IL can model the operation of a local area network with only a small amount of overhead time and software.

Interface Components

Of course, the HP-IL will never accomplish its potential unless there are many different kinds of HP-IL devices on the market. It is unreasonable to think that Hewlett-Packard Company would be either able or willing to supply all of them. Therefore, an important thrust of the HP-IL program is to allow others to design the HP-IL into their products.

By using the HP 82166A HP-IL Converter—a complete HP-IL interface component—manufacturers can design devices for the loop with very little engineering effort. The converter is a good solution for devices with relatively low sales volume and where small size, low power, and low cost are not major concerns. When these factors are important, however, the converter is not the optimum solution.

Though it requires a much greater engineering investment, primarily in the design of the device's microprocessor firmware, a component-level interface to the HP-IL is necessary when the size, power, and cost factors are critical to the product. There are three components needed for component-level implementation: the panel connector, the pulse transformer set, and the HP-IL integrated circuit.

*American Standard Code for Information Interchange

How Fast Is the HP-IL?

Determining the data rate for an interface is often difficult and misleading since it is usually dependent on highly variable hardware and software delays in the devices connected to the interface, rather than the maximum specifications of the interface itself. For any statement of data rate to be at all useful, the conditions under which the measurement is made must be carefully and completely specified.

The first important question is, "How fast could the HP-IL be if the devices did not limit the speed?" Assume a loop with only two devices: a controller/talker and a listener. The talker is continuously sending data frames to the listener. Because the first frame bit determines whether or not a frame is a data frame (as opposed to a command or ready frame), the listener will immediately retransmit the frame after only a one-bit delay and simultaneously load the frame into its buffer. Similarly, the talker can begin transmitting its next data frame after receiving only the first bit of the previous data frame. Error checking of the received frame is either not done or is done in parallel with transmission of the following frame.

Under these conditions, a continuous stream of data would fill the loop, limited only by the HP-IL timing specifications. These specifications say that a frame can be sent every 49 μs. The maximum data rate that could possibly be achieved is then slightly more than 20 kilobytes per second (each frame contains one byte of data). Furthermore, the loop could contain as many as ten of these "no-extra-delay" devices before the data rate would begin to suffer.

While such devices could certainly be built with existing technology, the HP-IL interface integrated circuit (see article on page 16) trades off this maximum performance for somewhat lower cost. This leads immediately to the next important question. "How fast would the HP-IL be if it were limited only by the existing interface IC and not by any added software delays?" Once again, the two-device loop example is useful, except that the devices now use the real HP-IL interface IC as opposed to a hypothetical one.

The sequence of events would proceed more or less as follows. After the data byte is written to the talker's HP-IL chip, there is a 4-μs delay before transmission of the bits starts. The frame takes 46 μs to be sent, and then the listener delays 7 μs before passing the byte to its microprocessor. The listener now retransmits the frame, which again has a 4-μs delay followed by a 46-μs transmit time. After the frame is received at the talker, there is a final 34-μs wait while error-checking is done before the following frame can be sent.

The grand total is 141 microseconds per frame, which translates to just over 7 kilobytes per second. Because the active listener does not retransmit the frame before it has been completely received, any additional devices will add to the time and degrade the data rate somewhat. An idle device using the present HP-IL interface IC delays a data frame by 13 μs so that three devices on the loop will reduce the speed to about 6.4 kilobytes per second, four devices will only support 5.9 kilobytes per second, and five will run at 5.5 kilobytes per second.

As the analysis gets closer and closer to the real world, software delays must also be accounted for. With a reasonably fast microprocessor and time-efficient (not necessarily ROM-efficient) code, an extra delay of no more than 50 μs could probably be achieved in each active HP-IL device. At a little less than 250 μs per frame, the two-device loop would have a speed of just over 4 kilobytes per second, three devices would operate at 3.8 kilobytes per second, four at 3.6, and five at 3.5.

Probably the easiest way to determine the data rate of a real system is simply to total the delay times of the individual devices involved. These times must be measured from the end of the received frame to the end of the frame transmitted through the device. Naturally, this value will vary somewhat depending on whether the device is acting as a talker or a listener and what type of data is being transmitted, but an average value is still useful.

For the HP-IL interface integrated circuit without extra software delays, the delay is around 70 μs. When the assumed software delay is added, the number goes to 120 μs. An HP-85 Personal Computer with the I/O ROM and HP-IL interface can achieve a frame delay of roughly 300 μs. The HP 82161A Digital Cassette Drive takes 600 μs per frame for a transfer of less than one 256-byte record, but increases to an average of 2600 μs per frame for very large blocks of data because of the record gaps on the tape. The slowest device is the HP-41C Programmable Calculator with its bit-serial microprocessor at about 5000-μs delay per frame.

A little computation then indicates that the combination of the HP-41C and the cassette drive can achieve a data rate of 175 bytes per second for short transfers and about 130 bytes per second for longer blocks of data. An HP-41C talking to an HP-85 can run at a little less than 190 bytes per second. If a cassette drive were used with an HP-85, the rate would be 350 bytes per second for long transfers, and about 1100 bytes per second for short ones. Two HP-85 Computers could communicate with each other at a little less than 1700 bytes per second across the HP-IL.

Clearly, the HP-IL is fairly slow when compared with the HP-IB. However, the data rate and other features of this interface system are well suited to the primary area of intended application: low-cost, battery-powered, portable systems.

-Steve Harper

While it is rare for a product division of Hewlett-Packard to sell the components that go into its products separately, that is exactly what is happening because of the special need to make the HP-IL available to other manufacturers. Customers are able to buy not only the connector, the transformer, and the integrated circuit (Fig. 4), but also the HP 82166C HP-IL Interface Kit complete with documentation and the HP-IL Development Module for the HP-41C Programmable Calculator. The components for the HP-IL converter are included in the kit. Hand in hand with this program of selling the HP-IL components directly goes the development of alternate sources for these components. This strategy should make the HP-IL and devices using it widely available relatively soon.

Documentation

One of the most important parts of the component sales effort is the supporting documentation. There are three publications that contain the needed information. For a first exposure to the HP-IL, a prospective user or designer can read *The HP-IL System: An Introductory Guide to the Hewlett-Packard Interface Loop*. With Hewlett-Packard's help, this book was written and published by Osborne/McGraw-Hill. *The HP-IL Interface Specification* provides

(continued on page 9)

HP-IL Interconnect System

by James H. Fleming

In most projects, connectors and interconnection systems do not typically receive the level of concentrated design effort that the other electrical and mechanical sections do, even though a study of service and reliability records shows interconnection problems rank among the major contributors to failure. This project was no different until consideration was given to the impact that a poor design or an incorrect choice of contact system could have on the success of this new interface standard.

This design problem was attacked by several engineers as the interface system was being defined. Prototype connectors ranged from reworked charger plugs to fiber-optic links. Finally, the connector system came into the critical development path of the seven or so HP-IL products. The most recent prototype connectors had provided a means to interconnect the products under development but did not satisfy the rapidly developing list of wants from the HP divisions designing HP-IL products. These wants included:

- Cable plugs to fully engage each other with tactile snap
- Cable end plugs to fully engage the device panel receptacle with tactile snap
- Both genders of contacts to have maximum recess for mechanical and electrostatic discharge protection
- Ability to add a third contact set to the center position for future ground/shield requirements
- Mechanical tolerances to allow compatibility between at least two full sets of multicavity injection molding tools, including consistent tactile snap
- At least two fabrication sources for the contact system
- Capability of enduring at least 2,000 insertion/withdrawal cycles with no mechanical or electrical failures
- Plug retention to be great enough to allow normal HP-IL system handling without fallout, but low enough to prevent damage if pulled out sharply by the cord (typically withstanding about two pounds of force).

The connector system that was ultimately chosen (Fig. 1) is manufactured by two major companies. These are interchangable in production and are of comparable quality. The contact systems were machine tested and no failures were found after over one million insertion/withdrawal cycles. It should be pointed out to other potential users of this type of contact that the design and mechanical tolerances of the plastic housings are critical to the reliability of the system.

HP-IL interconnect cables were initially made of the same material used in charger cords, but it was felt that these kinked too easily and the material did not perform well during the flex testing. Therefore, we changed to a vinyl with a lower Shore A durometer hardness of 60-70 and changed the wire stranding from nineteen strands of 36-AWG copper to twenty-six strands of 38-AWG tinned copper.

An interesting problem arose after switching to the softer durometer hardness cable jacket. We had been attempting to maintain a characteristic impedance of 100 ohms in the two conductor cables, but when the Shore A hardness was lowered, the dielectric constant increased, causing the characteristic impedance to be closer to 120 ohms. This was corrected by changing the conductor center-to-center distance from 1.98 mm to 1.52 mm.

The strain relief varies slightly between the cable end plug and the HP-IL module body because of differing loads. Both receive bending stress, but the cable end plugs have the additional load of insertion and withdrawal and are therefore thicker. Most of our charger cord failure data showed that the wires broke right at the end of the strain relief on the plug end and at the transformer case on the other end. By adjusting the wire stranding, cord material durometer hardness, and strain relief material durometer hardness, we were able to achieve the ideal situation of ultimate breakage near the middle of the strain relief. The module end of the HP-IL cable had the first failure after 4,700 90-degree flexes while the plug end survived 7,100 90-degree flexes. The strain reliefs are molded in vinyl with a Shore A-65 durometer hardness.

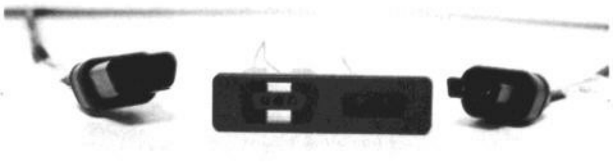
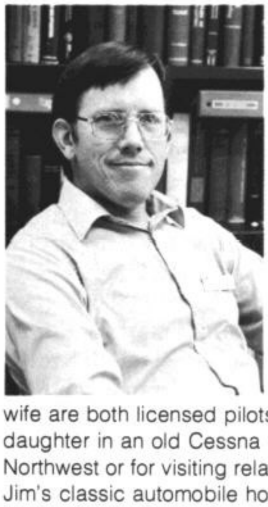


Fig. 1. *Connectors on the ends of each HP-IL cable are different and can only be inserted in their proper receptacles on an HP-IL device.*

Acknowledgments

I would like to thank Gene Frederick for his tool engineering support in bringing the vendors up to speed, Dave Hamilton for quality assurance engineering support, and Deby Birdsell for documentation and prototype parts organization.



James H. Fleming

Jim Fleming received two BS degrees from Oregon State University, one in engineering (1961) and one in business administration (1962). With the exception of one and one-half years in his own engineering consulting business, Jim has worked for HP since 1963 at five different divisions on a variety of products, the latest being the HP-41C and HP-IL peripherals interconnection. Born in Oakland, California, Jim served in the U.S. Navy, and before joining HP did mechanical design for a toy-making firm. He and his wife are both licensed pilots and enjoy packing their son and daughter in an old Cessna 172 for sightseeing trips around the Northwest or for visiting relatives in the San Francisco Bay area. Jim's classic automobile hobbies are going "on the shelf" for several years to allow time to design and build the family dream home in Albany, Oregon.

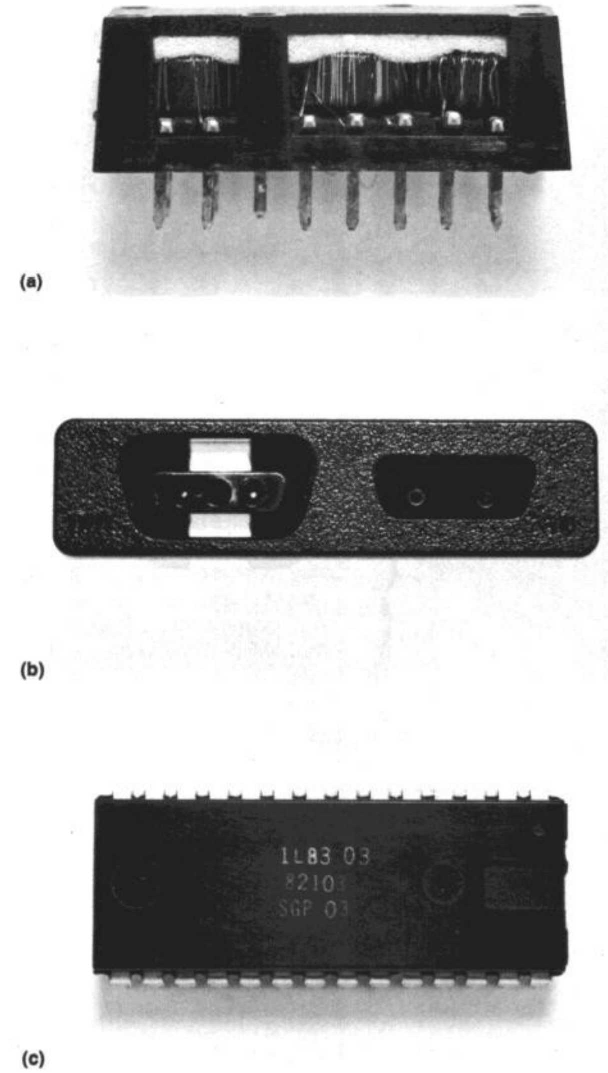


Fig. 4. *HP-IL connector (a), pulse transformer set (b), and interface IC (c).*

the designer with the HP-IL definition. This document contains all the necessary information to determine that a device is truly compatible. Format and content are very similar to IEEE Standard 488. The description, specification, and use of the HP-IL CMOS IC are given in *The HP-IL Integrated Circuit User's Manual*. This last document provides fairly detailed electrical and firmware design examples as an implementation guide. All three documents are included in the HP-IL Interface Kit.

In the near future, the major extension for the HP-IL will be the design of implementations with substantially higher transfer rates, yet compatible with current designs. Future implementations will also become even more friendly in both the low-level control and the transparent operation senses.

Today, even the smaller computers require the user to plan activities taking into account the computer and its transportability. Future computers and computer extensions will be used as casually as today's calculators. People take a calculator everywhere, not because they know they need it, but because they know they might need it. As this casualness enters the computing area, the HP-IL, with its small size, low power, and low cost, will be a desirable interface for small systems.

Acknowledgments

The HP-IL interface program was a joint venture of two product groups of Hewlett-Packard: the Instrument Group and the Personal Computer Group. This program was a good example of two groups cooperating effectively to produce an outstanding new product. The HP-IL standard is a credit to all who worked on it, and special thanks go to Bill Kay, Roy Barker, and Chung Tung for perceiving the need for the HP-IL and for supporting the joint venture at the

Steven L. Harper

Steve Harper is a graduate of Brigham Young University. He received the BSEE degree in 1971 and the MSEE degree in 1972, then joined HP. His contributions have included work on instrument calibration software and firmware for the HP-01 Calculator/Watch, and serving as a project manager for the HP-IL. Steve is co-author of a book, *The HP-IL System: An Introductory Guide to the Hewlett-Packard Interface Loop*, and is named co-inventor on a patent related to the HP-IL protocol. Born in Medford, Oregon, he is married, has four children, is expecting a fifth child, and lives in Corvallis, Oregon. He is the coordinator for the local Boy Scout troop and is involved in church activities, which have included a two-year mission in Brazil. His other interests are sports, hunting, folk guitar, science fiction, and dining out.

Roger D. Quick

Roger Quick joined HP in 1975 with ten years' experience in CAD and MOS IC design. His work at HP has included development of the electronics for the HP-10 and HP-19C Calculators and serving as project manager for the HP-41C software and electronics, the 82143A Printer, and the HP-IL system and components. Roger is a member of the Association for Computing Machinery, co-inventor on a patent related to the HP-IL, and co-author of two other articles related to his contributions at HP, one appearing earlier in the HP Journal. Born in Berkeley, California, he attended the University of California at Berkeley, earning a BA degree in mathematics in 1964. He is married, lives in Corvallis, Oregon, and owns an assortment of chickens, ducks, cats, two dogs, and a goat. His outside activities include being the campus manager of the site interaction committee at Oregon State University, breeding yellow Labradors, and collecting old Lotus sports cars.

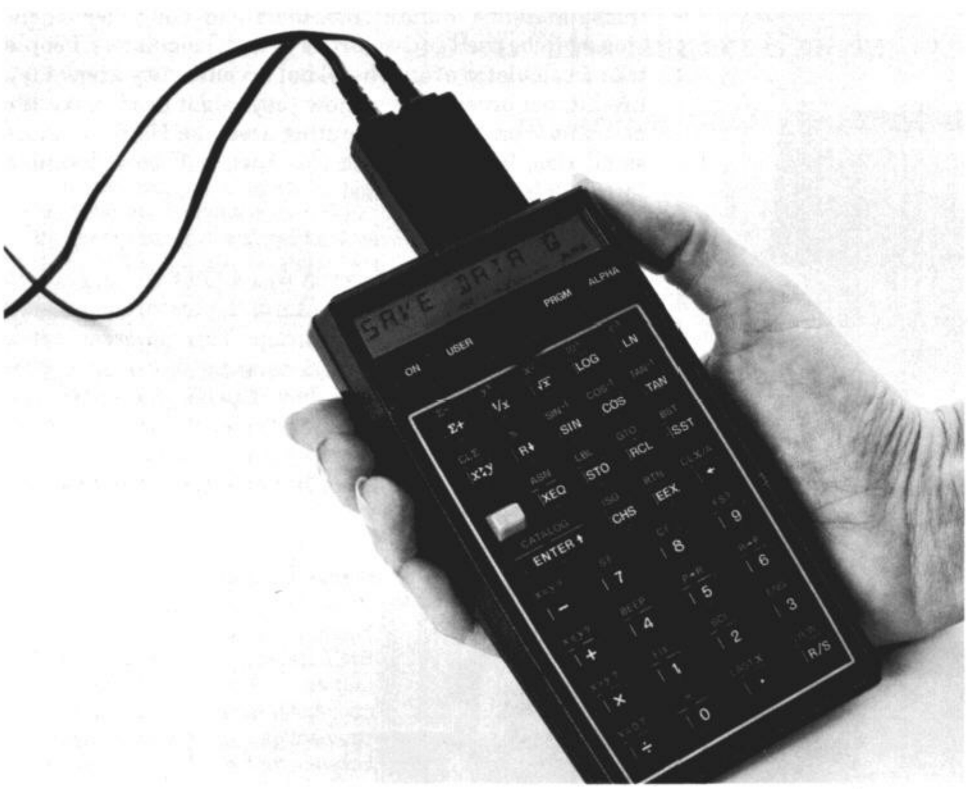


Fig. 5. *When a user plugs the HP 82160A HP-IL Module into one of the four I/O ports in the HP-41CV Programmable Calculator, the calculator becomes a portable system controller, able to interface with a variety of peripherals. Through the HP-IL, the HP-41CV can control a graphics plotter (see last month's issue, page 16), display data on a video monitor, transmit data over telephone lines to a larger Series 80 Computer, and collect measurements in the field.*

beginning, Joe Marriott and Dave Palermo for setting the initial architectural and functional foundations which allowed the HP-IL to carry the HP-IB experience into a totally new environment, Tom Heger and Dave Sweetser for establishing the HP-IL protocol and integrated circuit specifications and for the effective education they provided, Bernie Musch for supporting the HP-IL in the handheld computer area, Gary Stadele and Carl Landsness for doing the first implementations and finishing the job, and Dave Ricci for his guidance all along the way.

Further Reading about the HP-IL

1. S.Harper, C. Landsness, and R. Quick, "Interface System Weds Instruments to Small Computers," Electronic Design, December 24, 1981, pp. 78-87.

2. R. Katz, "The Hewlett-Packard Interface Loop—HPIL," Byte, April 1982, pp. 76-92.

3. P. Snigier, "Designer's Guide to the GPIB/HPIL," Digital Design, June 1982, pp. 48-52.

The Electronics Interface for the Hewlett-Packard Interface Loop

This low-cost, low-power serial interface uses two-wire cables, a three-level code, a CMOS IC, and small pulse transformers.

by Carl J. Landsness

FROM THE INITIAL CONCEPTION of the HP-IL, the driving objective was that it be compatible with small battery-operated devices such as an HP-41C Programmable Calculator. This dictated cost, size, and power requirements far below those of any other interfaces presently offered in the marketplace. The only significant tradeoff was speed, and as it turns out, the primary limit to speed is not the HP-IL electronics, but the microprocessor-based devices talking to the HP-IL (see box on page 7).

The choice of interface media was a very critical decision, with conventional cable and fiber optics leading the race and wireless radio control a distant third possibility. Although a wireless interface was appealing, practical considerations eliminated it early. After a thorough investigation, fiber optics was removed from contention, primarily because of cost and power consumption limitations. The final product is a careful blend of balanced two-wire cable (inexpensive zip cord for short distances, shielded twisted pair for long distances), transformer coupling (in most HP-IL products), an innovative encoding method (three-level, self-clocking), and maximum use of a low-power, large-scale integrated circuit technology (CMOS). Shown in Fig. 1 is the HP 82160A interface to the HP-41C, an example of an HP-IL product where the cost, size, and power objectives become visibly obvious.

Encoding

Without the luxury of extra interface lines for clocks or handshake signals, most interfaces and mass storage devices use some form of self-clocking codes. Some common examples are Manchester, Miller, and delta distance codes. Their common trait is encoding logical information as a function of time using two signal levels.

The HP-IL uses a three-level code in which logical information is encoded as a function of signal level transitions. Four distinct logical bits, 1, 0, 1S, and 0S, are defined as shown in Fig. 2. 1S and 0S are specially encoded versions of 1 and 0, respectively, and are used only within the three-control-bit segment at the start of each eleven-bit message frame (Fig. 3). The first control bit also serves as a sync bit for the frame. This provides an unequivocal start-of-frame reference if the loop gets lost for any reason (e.g., power failure or external interference). The particular code defined here is optimized not for density but for reliability. While a single-level transition per bit is possible (i.e., 1=positive transition, 0=negative transition), that approach would be more sensitive to glitches than the HP-IL code where a sequence of two transitions is necessary before decoding a bit. Another advantage to this particular code is the absence of any dc level. This is an advantage for transceiver pulse transformers, which don't operate very

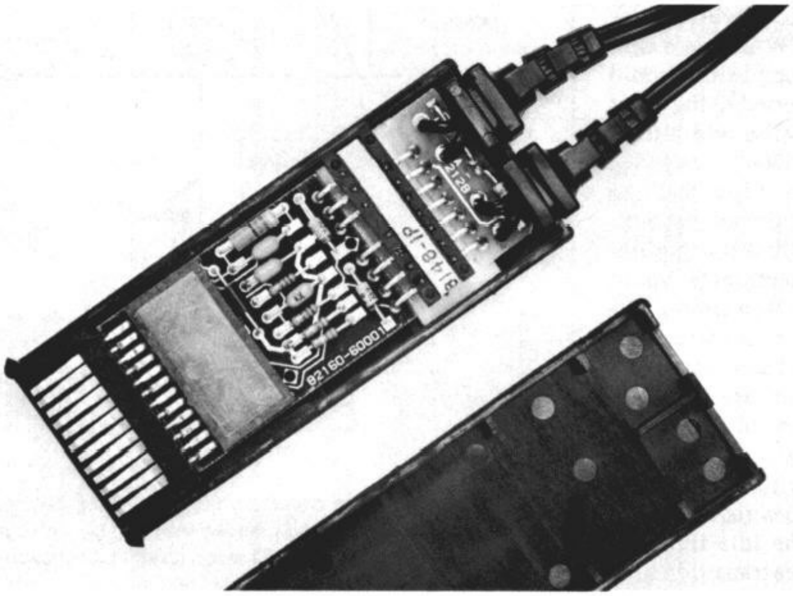


Fig. 1. *Opened HP 82160A HP-IL Interface Module showing internal electronic components.*

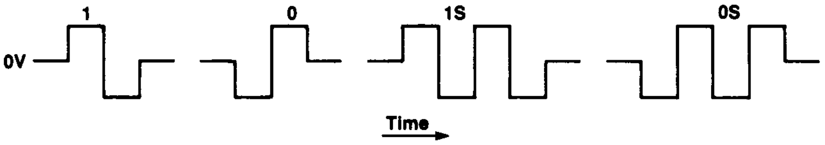


Fig. 2. *HP-IL definitions for logical bits 1, 0, 1S, and 0S.*

efficiently with dc bias (more on this later).

A very important feature of the three-level code is that the bits within a message frame are completely asynchronous with respect to each other. This makes it possible for devices of widely different speeds to communicate on the HP-IL without difficulty. For example, many types of frames may be retransmitted by a device after it receives only one bit. The remaining frame bits are retransmitted as soon as they are received, but at the speed of the device's transmitter electronics, not at the speed of the received frame, which could be slower or faster. Without asynchronous bits, the device would have to wait for all eleven bits in a frame before beginning retransmission to guarantee proper bit timing. If every device on the loop did this, the HP-IL speed degradation would be phenomenal. Two-level codes generally require one or more start bits to establish a timing reference. For bit-asynchronous operation, the start bit(s) would have to be tacked on to every information bit. The resulting bit-density degradation would be clearly undesirable.

Perhaps the biggest advantage of a three-level code for the HP-IL is the ability to detect significantly distorted signals using fairly simple and inexpensive receiver circuits while maintaining a high degree of reliability. To illustrate just one example of this, let's investigate the effect of detector threshold voltage variations in a situation where pulse rise and fall times are a significant percentage of pulse width. Fig. 4 shows the waveform of a two-level code with a pulse width TW and pulse spacing TS (TS=2TW in this example). In a delta distance code, an error condition would occur if the comparator's output were distorted to the point where TW′=TS′ (unable to distinguish pulse width from pulse space). This would occur if $TW/TR=2-4(V_T/V_M)$. What does this mean? If V_T is much different from half the signal amplitude, or if the rise time is a significant percentage of pulse width, detection becomes difficult if not impossible. In other words, the detector electronics would have to contain very accurate high-resolution timing circuits. In the HP-IL, we are dealing with pulse widths of about 1 μs. The circuits required to handle these fast pulses with good timing accuracy and resolution are generally beyond the cost, power, and size objectives of the HP-IL.

Using a three-level code, it is generally only necessary to detect signal levels. Timing is not critical. However, the particular code used in the HP-IL does require some timing of pulse widths to distinguish the idle time TS′ following a bit from the positive-to-negative transition time TR′ within a bit. From Fig. 5, it is obvious that even if V_T is equal to V_M and transition time TR is equal to pulse width TW (the worst case), no error condition can exist. What this says is that neither threshold level nor timing is very critical in detecting the three-level code. This makes it ideally suited for the HP-IL because the detector circuits can be very simple and inexpensive. Moreover, reliability may actually be improved by intentionally increasing the transition times at the detector (using a low-pass filter) to enhance noise immunity.

All of these advantages of the three-level code are countered by only one major drawback—generating and detecting a three-level signal in a basically digital (two-level) system. As we will see later, the answer lay in the use of simple pulse transformers.

Transceivers

The HP-IL is a two-wire balanced differential interface with both transmitter and receiver electrically isolated from the interface cable. The isolated operation (there is no ground on the wires) eliminates many system noise problems such as electromagnetic interference (EMI) susceptibility and radiation. This is especially important because

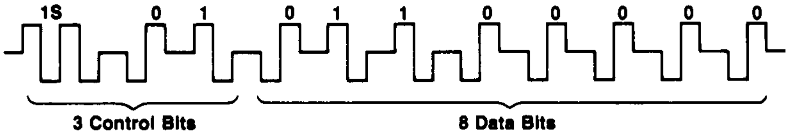


Fig. 3. *The HP-IL uses an eleven-bit message frame consisting of three control bits and eight data bits. The first control bit also serves as a sync bit.*

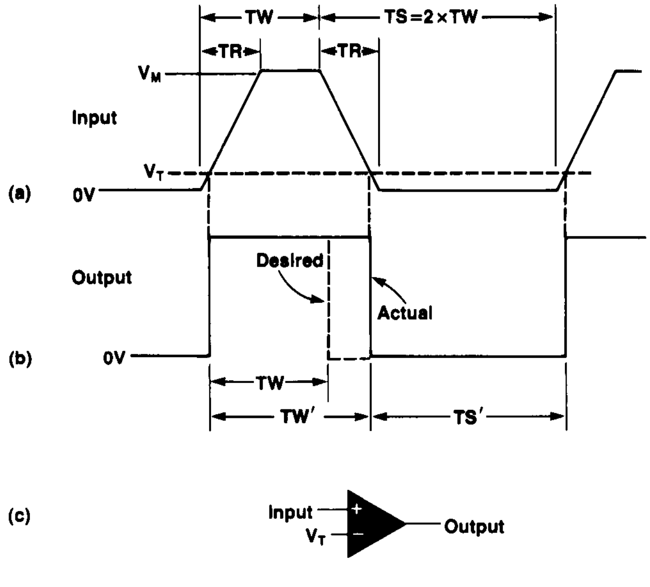


Fig. 4. *Rise-time distortion, two-level code. (a) Input pulse waveform. (b) Desired (dashed line) and actual output pulse waveform of circuit shown in (c), given input shown in (a). (c) Circuit block diagram.*

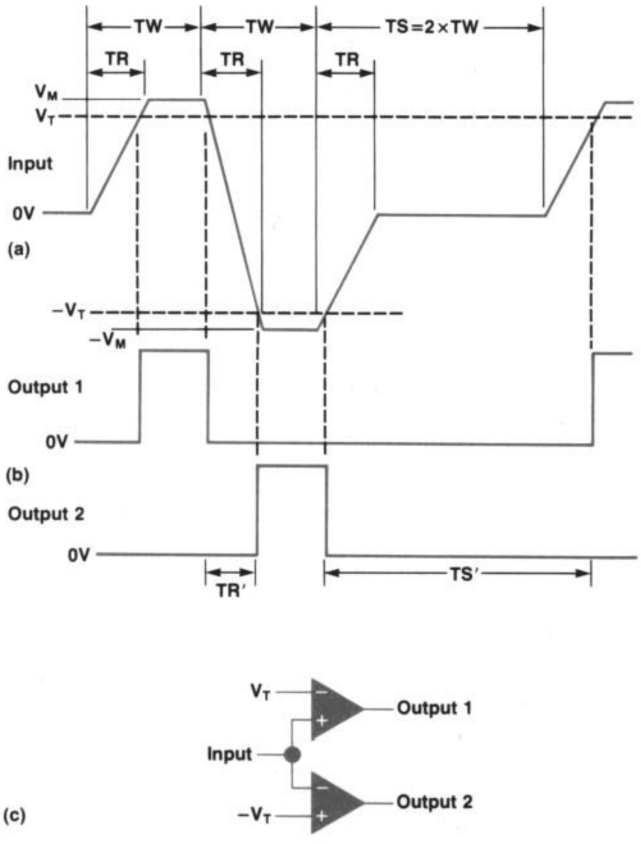


Fig. 5. *Rise-time distortion, three-level code. (a) Input pulse waveform. (b) Output pulse waveforms. (c) Circuit block diagram.*

the HP-IL may contain both floating (i.e., battery operated) and earth-grounded devices. The minimum isolation from interface cable to device common or earth ground is specified as 100 pF and 10 MΩ at 500Vdc. This eliminates any ground loop hazards and allows devices to be operated at different potentials. For instance, an HP-IL voltmeter may make small differential voltage measurements with its common terminal several hundred volts above earth ground and the measurement common the same as the circuit common within the voltmeter.

Each HP-IL device transmitter drives only one device receiver. This offers certain advantages over a bus type structure. The receiver may modify the transmitted message frame waveform (e.g., by filtering, attenuating, or clamping the signal) with no danger of interfering with any other receiver. The HP-IL takes advantage of this by thoroughly specifying the transmitter, while putting almost no restrictions on the receiver except that it reliably detect transmitted signals. This is accomplished by specifying several worst-case transmitter characteristics that the receiver must be able to detect.

The transmitter is designed to drive a 100-ohm balanced transmission line up to 100 meters in length which may be terminated in any impedance equal to or greater than 100 ohms. Therefore, the transmitter specification is given as a Thévenin circuit (to make it independent of load) with a nominal 100-ohm source impedance to absorb any transmission line reflections and a voltage waveform as shown in Fig. 6. Great effort was made to make the voltage and timing limits as loose as possible to allow for inexpensive electronics.

Shown in Fig. 7 is a schematic of one implementation of the HP-IL. The design combines some very simple digital circuits with small pulse transformers. Throughout the project design cycle, the typical reaction to the use of transformers was something less than favorable. Transformers simply don't have the glamour of the state-of-the-art

Parameter		Minimum	Maximum
TR	Rise Time	30 ns	120 ns (200 ns*)
TW	Pulse Width	950 ns	1500 ns
TI	Idle Time	1.9 μs	10 μs
TO	Overshoot Time		1 μs

*When tested with a 20-meter HP-IL cable

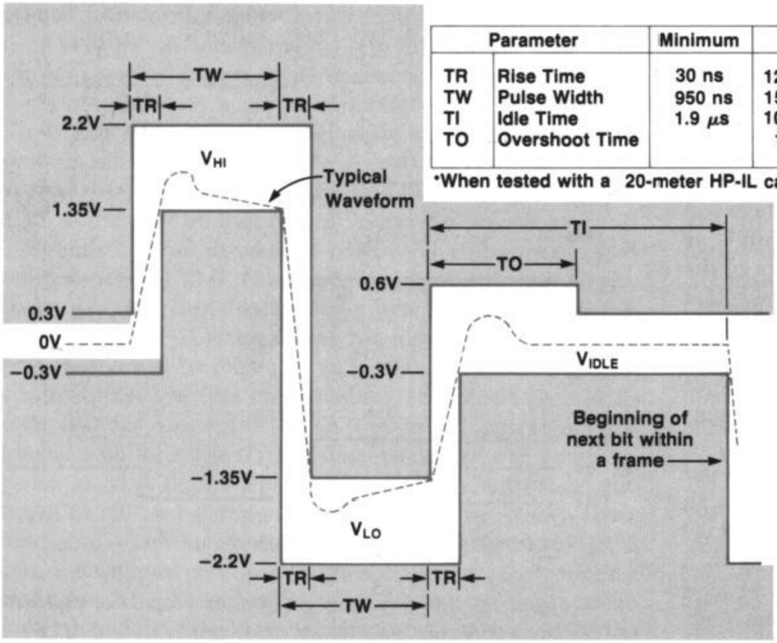


Fig. 6. *HP-IL transmitter waveform specification (open circuit). While only a logical-one bit is shown, all logical bits must conform to the same basic specifications with polarity changes made where appropriate.*

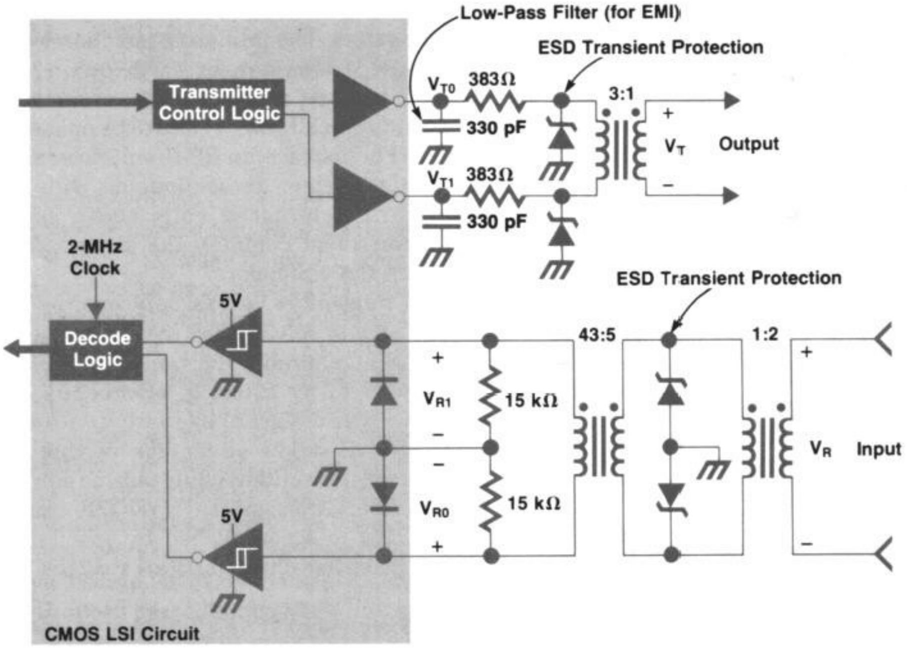
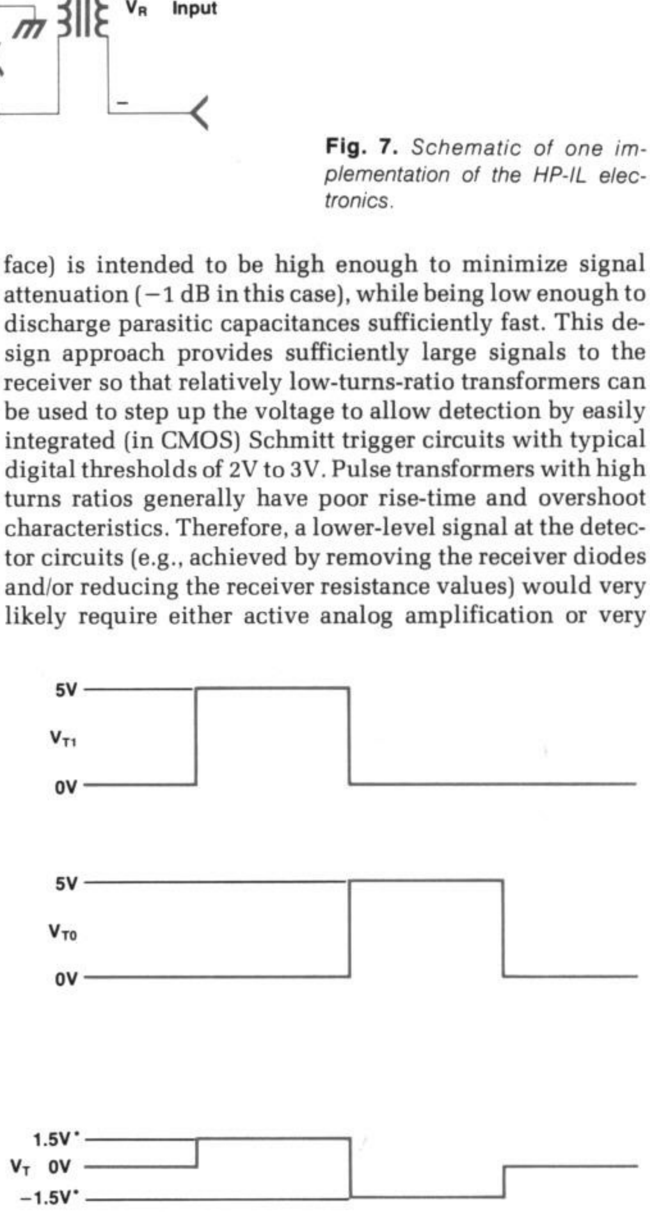


Fig. 7. *Schematic of one implementation of the HP-IL electronics.*

technologies like LSI (large-scale integration) and fiber optics. But the transformers have many redeeming qualities that made them always win in any direct confrontation with any other design approach. For one thing, they easily provide the electrical balance and isolation necessary for good noise performance (discussed earlier). Second, they provide a very easy way of generating the three-level code for the HP-IL and require no standby power, unlike most active circuits. Another advantage is the ability to shift voltage levels with near 100% power efficiency. The HP-IL nominal voltage levels were chosen at a fairly low level (1.5V typical) partly to conserve power. For example, a 100-ohm transmission line can be driven with about ten times less power at 1.5V than at a typical logic level of 5V. Fig. 8 and Fig. 9 show transmitter and receiver waveforms for the circuit of Fig. 7.

Still another big plus in using the transformers is easy impedance matching. It was a strong desire to minimize the circuit parts count by integrating most of the electronics using an LSI process like CMOS. The HP-IL specification calls for a transmitter impedance of 100 ohms +5%, −10%. It is moderately difficult to achieve on-chip impedances lower than 100 ohms in MOS drivers, and the tolerances can easily be ±30%. By using a 3:1 step-down transformer as shown in Fig. 7, the impedance on the device side of the transformer can be high enough (900 ohms in this case) that inexpensive discrete resistors can be used to swamp out the on-chip impedance. The on-chip impedance of 116 ohms ±30% (58 ohms per inverter) adds to the discrete resistance of 766 ohms ±1% to generate a loop impedance of 98 ohms ±5%, sufficient to satisfy the specification.

As shown above, transformers make it possible to use very simple digital circuits for the transceivers. The design shown uses diodes in the receiver to convert the three-level signal to two ground-referenced digital signals with a minimum of signal loss. The relatively high receiver load impedance of 15 kilohms (reflects to 810 ohms on the interface) is intended to be high enough to minimize signal attenuation (−1 dB in this case), while being low enough to discharge parasitic capacitances sufficiently fast. This design approach provides sufficiently large signals to the receiver so that relatively low-turns-ratio transformers can be used to step up the voltage to allow detection by easily integrated (in CMOS) Schmitt trigger circuits with typical digital thresholds of 2V to 3V. Pulse transformers with high turns ratios generally have poor rise-time and overshoot characteristics. Therefore, a lower-level signal at the detector circuits (e.g., achieved by removing the receiver diodes and/or reducing the receiver resistance values) would very likely require either active analog amplification or very



*Amplitude depends on load impedence

Fig. 8. *HP-IL transmitter waveforms for a logical one bit in the circuit of Fig. 7.*

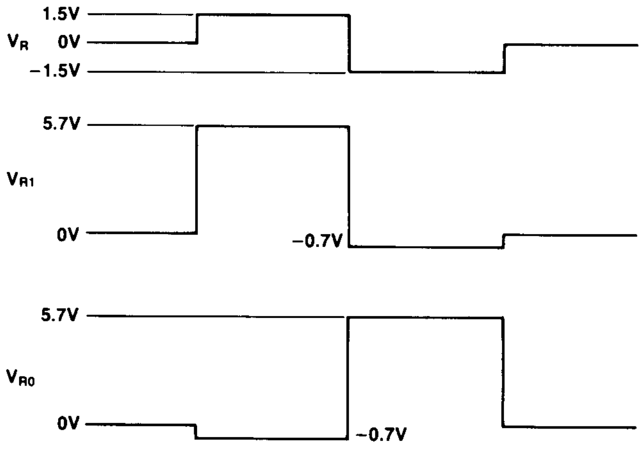


Fig. 9. *HP-IL receiver waveforms for a logical one bit in the circuit of Fig. 7.*

low-level comparator circuits. Both options are generally higher in cost and use more power at the pulse frequencies used in the HP-IL.

Electromagnetic Interference

From the HP-IL project's inception, a major concern was the radiation of EMI. The relatively long interface cables (great antennas), the lack of any chassis ground, and our experience with other products all supported this concern. Of five common noise-reduction techniques (grounding, shielding, filtering, isolation, and balancing) we chose to concentrate on the latter three, mainly because the HP-IL does not have the luxury of earth grounds in most products. The objective was to minimize the level of any common-mode signal (relative to earth ground) on the two-wire interface and assume that a differential signal would generate electromagnetic fields that would cancel at any significant distance away from the cable (more than a few centimeters). By their isolating nature, the pulse transformers do an excellent job of rejecting frequencies below about 10 MHz. For the higher frequencies where parasitic capacitance degrades transformer isolation, a simple low-pass filter is used (330-pF capacitors in Fig. 7) to filter the outputs of the digital inverter drivers. This is necessary because the two inverters generate a common-mode pulse of amplitude $V_{CC}/2$ and a fundamental frequency of 500 kHz with harmonics out to very high frequencies. A true differential analog driver would eliminate this need for filtering, but also would require more complexity and power.

Equally important was the need to be immune to externally generated EMI. Our practical experience has shown that power line transients along power cords in close proximity to the HP-IL cables were the most severe sources of EMI that could be expected to be encountered in typical applications. For this environment, the HP-IL defines its own test for susceptibility. A two-meter power cord is placed parallel to the HP-IL cable at a distance of 1 cm. The power cord is then subjected to transients with short rise times (5 ns) and damped sinusoids. The test setup is intended to represent a rather severe application where HP-IL cables and power cords from calculators, instruments, small appliances, and other devices are grouped together on a lab bench or desktop. Much data exists[1] regarding the amplitude, rise time, and frequency of occurrence of transients, but this is dwarfed by the number of opinions regarding acceptable performance criteria. After much study, we concluded that rejection of 500V transients would provide reliable operation, and that error checking and recovery techniques in the system software could handle the very rare occurrence of transients of higher amplitude.

The design of a receiver circuit to handle these transients turned out to be a major undertaking. Testing verified that most of the interference is coupled to the HP-IL cable as common-mode. Therefore, the obvious solution was to design the receiver electronics with good common-mode rejection over broadband frequencies. However, because of parasitic capacitance between the transformer primary and secondary, this is not an easy task at higher frequencies (2 to 50 MHz in our case). Two design solutions have been used in HP-IL products to date. One is an electrostatic shield between transformer primary and secondary; the shield shunts common-mode currents to device low rather than allowing them to flow to the load. The second approach (Fig. 7) uses two very carefully balanced transformers in the receiver that operate on a principle very similar to a single-shielded transformer except that the parasitic capacitance from each signal lead to device low must be well balanced. Because of the difficulties in manufacturing very small transformers with shields, it was found that the two-transformer approach is more cost-effective. There are many other approaches to improved balancing, but most require replacing the simple digital detector circuits, which by themselves are not well balanced, with analog differential amplifiers/comparators. Concentrating on the transformers has proved more cost-effective for the present.

The design results have been very rewarding. With the design approach outlined here, prototype systems that failed the susceptibility test at 100V transient amplitudes were able to handle 500V to 1000V transients and did not exhibit any transmission failures below a field strength of 5V/meter.

Electrostatic Discharge

With the proliferation of rather sensitive MOS circuits in today's electronic products, susceptibility to electrostatic discharge (ESD) is a major reliability concern. All HP-IL products are required to exhibit no permanent failures, and most will exhibit no temporary failure, when a 300-pF capacitor charged to 15 kV is discharged through a 500-ohm resistor to any product surface. This is a very high-energy pulse compared to those typically generated by a user shuffling across a carpet.

Products interfacing to the HP-IL will generally follow one of three general ESD protection approaches: an earth-grounded Faraday shield, a floating Faraday shield, and insulation of circuits. This creates an interesting problem when a device with a floating Faraday shield is connected in the system. The original interconnect design went to great efforts to insulate and otherwise protect the interface conductors so that a discharge could not occur to the wires and couple directly into the product electronics. After we tested many prototype products with great success, the first system test found that a discharge to a product with a floating Faraday shield caused failures in an adjacent product. The discharge to the shielded device raised the potential of all internal electronics to 15 kV, which broke down the HP-IL transformers in that device, thus putting 15 kV on the cable conductors. This broke down the HP-IL transformers in an adjacent device. As a result, it is now required that HP-IL devices withstand discharges made directly to the interface conductors, even if they are not externally accessible. The design approaches used to date to solve this problem have been very device dependent, but have generally employed a combination of clamping, filtering, good grounding, and transformer primary-to-secondary shielding.

Acknowledgments

Special thanks goes to Ron Swerlein of Loveland Instrument Division for much assistance in defining the electrical specifications.

Reference

1. *Transient Voltages Suppression Manual*, General Electric, 1978, pp. 1-8

Carl J. Landsness

Carl Landsness was born in Madison, Wisconsin. He received the BSEE degree from the University of Wisconsin in 1973 and the MSEE degree from Stanford University in 1976. With HP since 1973, Carl has worked on a variety of products including the HP 3000 Computer, the HP-91/97 Calculator power supply, the optical wand for the HP-41C, and most recently, the HP-IL. He is married, has a son, and lives in Corvallis, Oregon. He is interested in whitewater kayaking, nordic and alpine skiing, and boardsailing, and plays soccer during his lunch hour.

A CMOS Integrated Circuit for the HP-IL Interface

by Steven L. Harper

UNDOUBTEDLY THE MOST IMPORTANT judge of the Hewlett-Packard Interface Loop (HP-IL) system is the end user. However, the device designer is probably very near this level of importance also. Unless the designer sees the HP-IL system as capable and friendly from a design viewpoint, the designer is not very likely to create a product that shares these attributes.

This requirement, together with other critical needs of very low power consumption, low cost, and small physical size, make the hardware design decision for the HP-IL interface a fairly straightforward one. A CMOS LSI circuit (Fig. 1) is the only technology that effectively satisfies all of the above objectives.

Hardware Architecture

The eight-bit microprocessor has become almost a universal controller for the tiny digital systems used in small instruments or peripherals. For this reason, it made sense to us to design the HP-IL interface IC with an eight-bit data bus that can mate directly with most common microprocessors. In this way, the HP-IL interface simply becomes another component in a device's microprocessor system.

This approach provides two important advantages. The abilities of the device's microprocessor can be effectively shared between the device functions and the interface functions for lower cost, less power, and smaller size. Perhaps equally important is the flexibility of this design. While the time-critical portions of the HP-IL protocol can be executed quickly by the logic on the interface IC, most of the protocol can be contained in the microprocessor's firmware. This approach reduces cost and at the same time makes it relatively easy to incorporate changes to enhance capability or speed or to correct problems. Somewhere in the neighborhood of 1000 bytes of microprocessor code is required to support the HP-IL functions for most typical devices having the ability to send or receive data.

In addition to the eight-bit bidirectional data bus common to microprocessors, the interface IC also has the more or less standard complement of control lines. A RESET input allows external power-on circuitry to set the entire integrated circuit to a predetermined state when power is first applied. There are three address (REGISTER SELECT) inputs which select one of eight control and data registers to send or receive on the data bus. External address decoding circuitry drives the CHIP SELECT input so that the HP-IL interface circuit appears as a block of eight memory addresses or input/output ports to the device's microprocessor. A WRITE or READ input gates the contents of the data bus into or out of the interface IC. An INTERRUPT REQUEST line indicates to the microprocessor that the HP-IL interface circuit needs

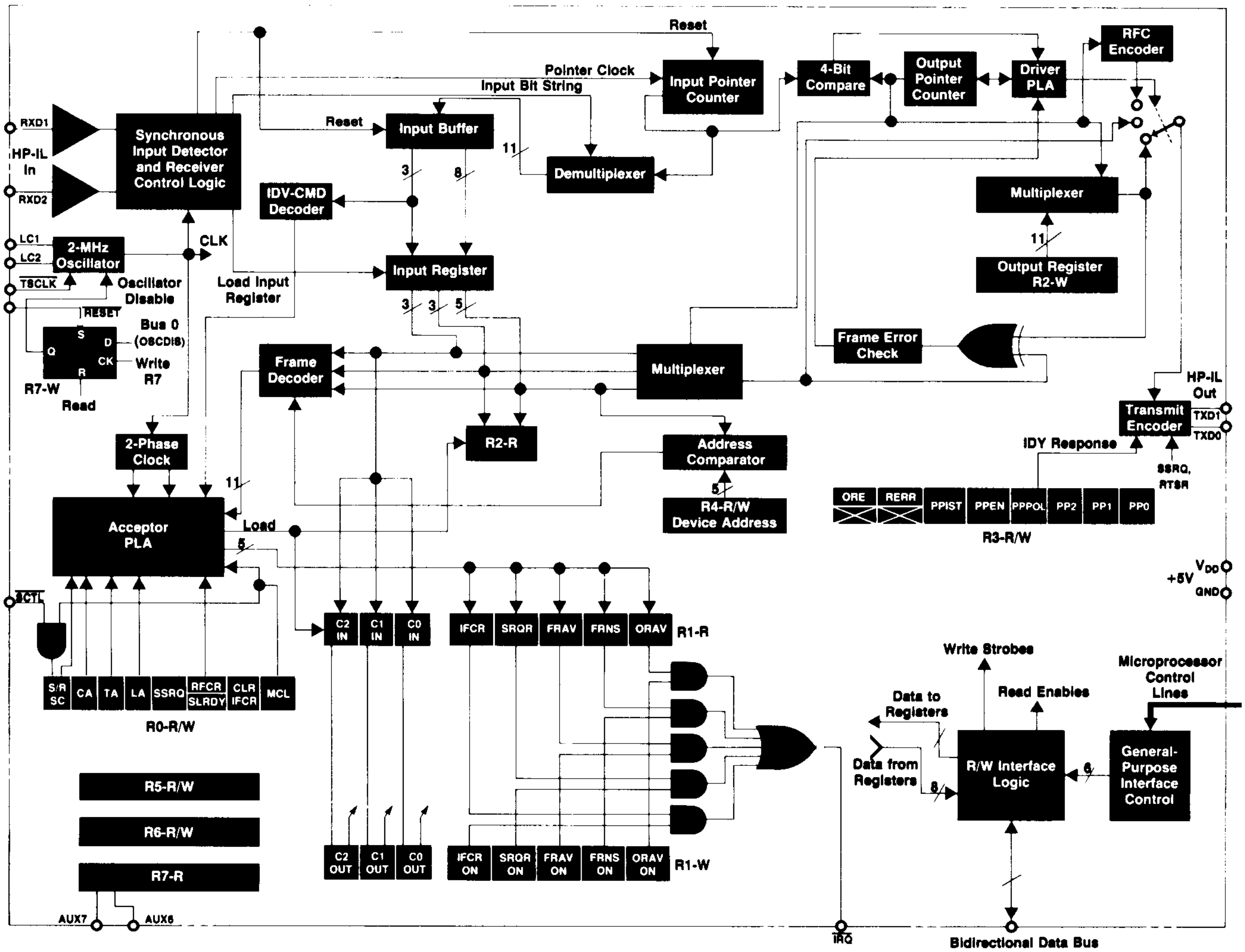


Fig. 1. *Block diagram of HP-IL interface CMOS integrated circuit. This IC has an eight-bit bidirectional data bus for easy connection to most common microprocessors.*

attention.

Beyond those connections that interface to the device's microprocessor are a number of pins necessary to support the HP-IL interface IC. The power and ground lines need a standard 5V supply. There are two loop data inputs and two loop data outputs, which connect to the pulse transformers and other discrete components that adapt the logic level signals to what is required on the loop. There are also two connections for a parallel LC circuit to control the frequency of the on-board oscillator at 2 MHz. Two general-purpose flag inputs are provided along with a special flag that is sampled only when power is applied and indicates to the integrated circuit that it either is or is not in charge of the entire HP-IL system (SYSTEM CONTROLLER). The last pin on the integrated circuit's 28-pin dual in-line plastic package is an external oscillator input, primarily used for testing purposes.

Various parts of the eight registers that are the main means of communication through the data bus to the device's microprocessor are prominent in the block diagram of Fig. 1. Note that some of these registers are really two separate registers, one that can only receive data from the bus and one that can only send its data out on the bus. The read-only portions are loaded within the interface IC and the write-only sections send their data to or control other internal logic only. This technique saves address space for the microprocessor and eliminates the need for an extra address pin. R1-W refers to the write-only part of register one, for example, and R2-R similarly indicates the read-only half of register two.

Virtually all communication from the device to the HP-IL and vice versa is initiated by read or write operations executed by the device's microprocessor to these eight registers. No other control lines are necessary to perform this function. For example, when the microprocessor writes a byte to register two (R2-W), the interface IC automatically transmits that byte on the loop.

The interface logic and interface control blocks connect the registers to the external microprocessor. This link functions asynchronously from the loop and the rest of the integrated circuit. These blocks provide a simple, standard interface to the microprocessor, but require an extra interlock and synchronization circuit. If the microprocessor were to read a register at the same instant it was being loaded internally, only part of the data might be correctly read, causing an error which would be nonrepeatable and very difficult to trace. The extra logic eliminates this possibility.

The on-board oscillator provides the basic timing source for the entire interface IC. This 2-MHz signal is used to sample the incoming HP-IL data directly. Since the nominal pulse width on the loop is 1 μs, each pulse can be sampled at least three times for increased noise immunity. The clock is also divided down to form a 500-kHz two-phase signal for use in the rest of the integrated circuit. By writing the proper control register and bit, the oscillator can be turned off or on. When it is off, the circuit draws less than 1 μA, something very important for small, battery-powered devices.

The incoming HP-IL lines enter the detector and receiver control block where the pulse sequences are decoded. Noise spikes are ignored. The presence of a sync bit in the middle of the eleven-bit message frame sets an error bit (see pages 5 and 12 for sync bit and frame definitions). Detection of a sync bit also resets the input pointer so that it and the ensuing bits of the frame are correctly loaded through the demultiplexer into the input buffer.

The input buffer, the input register, and R2-R form a three-level, first-in, first-out (FIFO) buffer that is really only necessary in those rare instances when there is more than one message frame in transit around the loop at the same time. Usually, when a frame is being received, the input register is empty and the frame is gated through the input buffer directly into the input register. Most of the frame decoding is done in the input register. Depending on the type of frame and whether the interface IC is configured as HP-IL controller, talker, or listener, the frame might be immediately sent out again through the input register multiplexer and the transmit encoder or it could be loaded into R2-R.

There is a third important possibility for the disposition of the frame in the input register. If the interface IC happens to be the source of the received frame (a talker getting back its own transmitted data frames, for example), the received frame can be compared to what was transmitted for error-checking purposes. This is done by shifting out the input register and the output register (which contains a copy of the transmitted frame) simultaneously through their multiplexers into an XOR gate.

The input and output pointer counters control the three multiplexers that provide the serial-to-parallel and parallel-to-serial conversion of HP-IL frames. Oscillators on the HP-IL interface ICs in loop devices are not synchronized. Comparison logic prevents the possibility of the output pointer overrunning the input pointer during an automatic retransmission. This requires that the individual bits within a frame be asynchronous with each other.

The transmit encoder receives data to be sent on the HP-IL from several sources. The input register multiplexer passes frames that must be immediately retransmitted with as little delay as possible. The output register multiplexer provides the frames that originate locally at the device. Other inputs to the transmit encoder provide modification of frames "on the fly" for service request and parallel poll responses and for regenerating a received handshake frame when the device has finished executing the previous command frame.

Certain HP-IL commands contain a device address. A device's address comparator checks to see if the address in the received command is the same as the address assigned to the device.

The acceptor and driver programmed logic arrays (PLA) provide the real intelligence of the interface IC. The frame type and state of the device (controller, talker, or listener) are the primary inputs to the PLA. Its outputs control virtually the entire integrated circuit except for the interface circuitry to the device's microprocessor.

Register Map

Fig. 2 is the programmer's model of the HP-IL interface IC which shows the detailed functions of the various registers.

Register 0 is the status register. Its bits encode the controller, talker, or listener state of the integrated circuit. When the system controller bit SC is set, an incoming interface clear (IFC) command is presumed to have been generated by this device and is stopped and error-checked. If the controller active bit CA is set, all other commands are stopped and checked since this device must have generated them. When the talker active bit TA is set, data frames are stopped and checked. The listener active bit LA causes incoming data frames to be loaded into R2-R and passed to the device's microprocessor. More than one of these bits can be set at once and the combined functions are performed as would be expected.

		BUS7	BUS6	BUS5	BUS4	BUS3	BUS2	BUS1	BUS0
Register 0: Status	Read	SC	CA	TA	LA	SSRQ	RFC	CLIFCR	MCL
	Write						SLRDY		
Register 1: Interrupt	Read	C2IN	C1IN	C0IN	IFCR	SRQR	FRAV	FRNS	ORAV
	Write	C2OUT	C1OUT	C0OUT	Interrupt Enable Bits				
Register 2: Data	Read	D7IN	D6IN	D5IN	D4IN	D3IN	D2IN	D1IN	D0IN
	Write	D7OUT	D6OUT	D5OUT	D4OUT	D3OUT	D2OUT	D1OUT	D0OUT
Register 3: Parallel Poll	Read	ORE	RERR	PPST	PPEN	PPPOL	P2	P1	P0
	Write	-	-						
Register 4: Loop Address	Read Write	Scratchpad Bits			ADD4	ADD3	ADD2	ADD1	ADD0
Register 5: Scratchpad	Read Write	Scratchpad Bits							
Register 6: Scratchpad	Read Write	Scratchpad Bits							
Register 7: Auxiliary Input	Read	AUX7	AUX6	1	1	1	1	1	1
	Write	-	-	-	-	-	-	-	OSCDIS

Fig. 2. *Register map for HP-IL interface IC.*

The four least-significant bits of register 0 perform certain special functions. The send service request bit SSRQ causes the service request bit SRQ to be set in any HP-IL data or identify message frames that are transmitted through a device so the controller can be aware that the device needs attention. Bit two is the only split bit in register 0. The read-only bit indicates that a ready for command (RFC) frame has been received. The write-only bit, set local ready (SLRDY), indicates to the interface IC that the device has finished executing the previous command and can pass on the ensuing RFC. If it has already been received, the RFC encoder automatically regenerates the RFC frame when SLRDY is set. SLRDY automatically resets itself in 2μs. The clear interface clear received bit CLIFCR provides the only means of resetting the interface clear interrupt bit IFCR in register 1. CLIFCR also automatically resets in 2μs. Master clear MCL resets the integrated circuit to its power-on state except for shutting off the oscillator.

Register 1, the interrupt register, is totally split. Since the incoming HP-IL message frame has eleven bits, more than one eight-bit register is needed to contain it. The three control bits of the received frame are loaded into the three most significant bits of R1-R. Likewise, the control bits of the frame to be transmitted to the next device on the HP-IL must be written to R1-W before sending the frame out. Often the incoming frame is simply sent on without change after it is read. To save steps, the control bits of the received frame in R1-R are automatically copied to R1-W whenever a frame is read by the device's microprocessor.

The other five bits in R1R represent the various interrupt conditions. Each of these five bits has a corresponding enable bit in R1-W. If the enable bit is set and the proper condition occurs to cause the corresponding interrupt bit in R1-R to be set, then an interrupt will be generated on the interrupt request line to the device's microprocessor. If the enable bit is not set, the interrupt bit functions the same way except that no interrupt is transmitted to the microprocessor.

The interface clear received bit IFCR is set whenever an interface clear (IFC) command is received. Service request received SRQR is set only in the active controller when a data or identify message frame comes in with the service request (SRQ) bit set. This interrupt resets itself when a frame comes in without the SRQ bit set. The frame available interrupt FRAV is generally used to indicate to the active listener that a data frame has been received, though there are some other situations where this bit is used. When the frame is read from R2-R, the bit resets. Frame received not as sent (FRNS) tells the active HP-IL controller or talker that the frame that was sent returned incorrectly. When this happens, the bad frame is loaded into R2-R so the device's microprocessor can read it. Like FRAV, this bit resets when the frame is read. Output register available ORAV indicates to the HP-IL talker or controller that its frame has returned correctly and the next frame can be sent. ORAV resets when the next frame is written to R2-W.

The interaction of the interface IC status, the received frame and the interrupts is quite complex. Table I is a complete summary. Taken as a whole, this appears rather formidable. However, when the microprocessor firmware designer starts to design code for the talker function, for example, it will be found that a relatively friendly subset emerges from the complexity of Table I. The protocol used in the HP-IL is based on the HP-IB* and so the mnemonics used in Table I are generally familiar to HP-IB users.

Register 2 is the data register. The eight data bits of the received message frame are loaded from the HP-IL into the read-only part of the register. When the frame is transferred to the device's microprocessor, another frame can be loaded from the input register into R2-R. Also, after the data bits of the frame to be sent by the talker or controller are placed in R2-W, the interface IC transmits the frame over the loop.

The information to allow the device to respond automatically to parallel poll is contained in register 3. A positive response is enabled only when both the parallel poll status bit PPST and the parallel poll enable bit PPEN are set. If the parallel poll polarity bit PPPOL is set to one when a positive response is enabled, the proper bit in the identify frame is set to one as the frame is retransmitted. If PPPOL is zero, a negative response (PPST=0,PPEN=1) will set the bit in the identify frame. This allows the HP-IL controller to receive the logical AND or the logical OR of the responses of multiple devices in a single bit in the identify frame. The three low-order bits in register 3 determine which bit will be modified in the identify frame.

Also in register 3 are two read-only bits that indicate when the output register is empty (ORE) and when a receiver error has occurred (RERR). This is caused by the presence of a sync bit in the middle of a message frame.

Register 4 is the loop address register. The five least-significant bits contain the HP-IL device's assigned loop address. When an idle device receives a talk or listen command frame, for example, and the bits in the command frame match the bits in register 4, the device's HP-IL interface integrated circuit interrupts the device's microprocessor so that it can set the appropriate status bit (TA or LA, in this case). If the bits do not match, the command does not disturb the device's microprocessor.

The three high-order bits of register 4 and all bits of registers 5 and 6 are scratchpad registers for the device's firmware designer to use for any purpose. This data is preserved, even when the master clear bit is set and the oscillator is shut off, as long as power is continuously applied to the interface IC.

Register 7 contains the two input flag bits (AUX6 and

*Hewlett-Packard Interface Bus, HP's implementation of IEEE Standard 488 (1978)

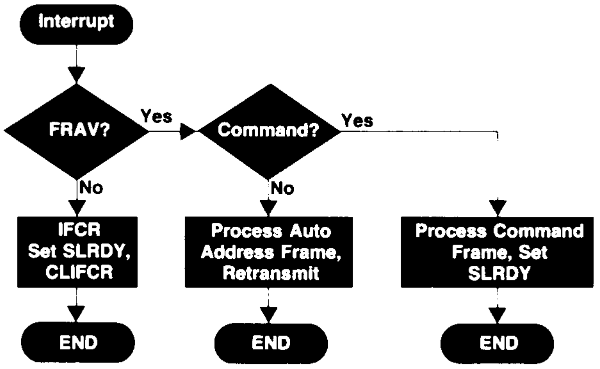


Fig. 3. *Simplified flowchart of the response of an idle device on the HP-IL.*

Table I
HP-IL IC Interrupt Flag Response

Received Frame	Chip Status				Interrupt Flags			Notes
	SC	CA	TA	LA	FRAV	FRNS	ORAV	
DOE	X	X	0	0	-	-	-	Automatic retransmission
	X	0	0	1	S	-	-	CPU retransmission
	X	X	1	0	-	E	S	Source, automatic error check
	X	1	0	1	S	-	S	CPU retransmission
IFC	0	0	X	X	-	-	-	Automatic retransmission
	0	1	X	X	-	-	S	CPU retransmission
	1	1	X	X	-	E	E	Source, automatic error check and automatic source RFC if okay.
UCG • $\overline{\text{IFC}}$+SAG +(TAG+LAG) • match	X	0	X	X	S	-	-	Automatic retransmission
ACG+(TAG+LAG) • $\overline{\text{match}}$	X	0	0	0	-	-	-	Automatic retransmission & SLRDY
	X	0	X	X	S	-	-	Automatic retransmission
CMD • $\overline{\text{IFC}}$	X	1	X	X	-	E	E	Source, automatic error check and automatic source RFC if okay.
RFC	X	0	X	X	-	-	-	Automatic retransmission after SLRDY
	X	1	X	X	-	-	S	Source, decoded
ARG	X	0	0	0	-	-	-	Automatic retransmission
	X	*	*	*	S	-	S	CPU retransmission or source, (no automatic error check)
AAG	X	0	X	X	S	-	S	CPU retransmission
	X	1	X	X	-	E	S	Source, automatic error check
IDY	X	0	X	X	-	-	-	Automatic retransmission
	X	1	X	X	-	E	S	Source, automatic error check

* CA+TA+LA=1
X Don't Care. Some combinations are explained in the text, so the table does not necessarily represent all possible don't care states.
S The bit is set high when the specified frame is received and the chip status is as shown.
E The bit is set high only if automatic error-checking detects an error.
- This combination has no effect on this bit.

AUX7). These are general-purpose bits and may be used to sense the state of switches, for example. The other read-only bits always return ones. The write-only part of register 7 has only one bit, the oscillator disable bit OSCDIS. After the master clear bit is set, this bit turns the oscillator off or on. The other bits are "don't cares"; they can be written as one or zero with no effect on the interface IC.

Device Interaction with the Interface IC

To understand fully the place of the interface IC in the HP-IL portion of the microprocessor system of a device, it is necessary to delve into the firmware and look at some specific situations. A prospective designer will then be able to see how Table I reduces to something more manageable.

In the case of an idle device, the possibilities are few. Many types of message frames are automatically retransmitted on the loop without disturbing the device's microprocessor at all. Some other frames do cause an interrupt, but are simply retransmitted without any other response.

If an interface clear (IFC) is received, the IFCR bit is set, causing an interrupt. Since the device is already in an idle state, however, nothing needs to be done to execute this command. The device's microprocessor simply writes the CLIFCR bit to clear the interrupt, and then writes the SLRDY bit so that the following ready-for-command (RFC) frame will be passed on.

Any other message frames that require a response from an idle device will cause the FRAV bit to be set. The device's microprocessor reads the control bits from R1-R and the data bits from R2-R. If the frame is a command, the device executes it and then writes SLRDY. The only other type of frame that causes a FRAV interrupt in an idle device is an auto-address-group (AAG) message. The device's microprocessor may respond by reading the frame, storing its address bits in register 4, incrementing the frame address bit, and retransmitting the modified frame by writing it to

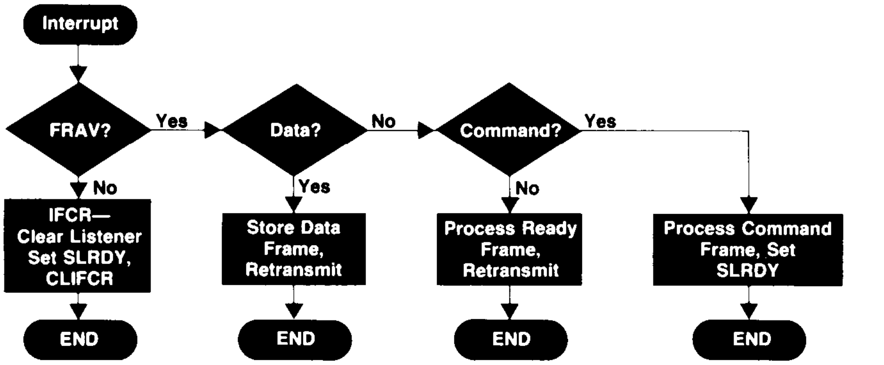

Interrupt
FRAV?
Yes
No
IFCR— Clear Listener Set SLRDY, CLIFCR
END
Data?
No
Yes
Store Data Frame, Retransmit
END
Command?
Yes
No
Process Ready Frame, Retransmit
END
Process Command Frame, Set SLRDY
END

Fig. 4. *Simplified flowchart of the response of a listener on the HP-IL.*

R2-W. Fig. 3 illustrates in flowchart form the response of an idle device.

When a device becomes an active listener, the additional capability needed is minimal. A FRAV interrupt is received when a data frame comes in. The listener's microprocessor reads it, may store it in a buffer, and then retransmits it. An addressed-ready-group (ARG) frame also may cause an interrupt. These frames are merely retransmitted by the listener's microprocessor. Fig. 4 shows the additional flowchart necessary for the listener.

For devices that are normally the destination of loop messages, the primary interaction is with the frame-available interrupt. The previous discussion regarding idle devices and listeners illustrates this. With talkers and controllers, however, which are usually the sources of loop messages, the emphasis shifts to the output register available interrupt ORAV. This interrupt does not merely indicate that the message frame has been transmitted. It is used to notify the device's microprocessor that the full loop handshake has been completed. In the case of a data frame sent by a talker, this means that the frame has been transmitted, has been received and retransmitted by the active listeners (so that they are all ready to receive another data frame), has returned to the talker, and has been error-checked and found to be correct. Only at the completion of this sequence does ORAV go true. When an HP-IL controller generates a command, the frame is sent and, in turn, is automatically retransmitted by all devices (each retains a copy of the command and begins executing it), the command returns to the controller where it is error-checked, the controller's interface IC automatically sends the ready-for-command frame, which is retransmitted by each device when it finishes executing the previous command, and the RFC returns to the controller's interface IC, which then sets ORAV.

When a frame is garbled so that it does not error-check properly, the frame received not as sent (FRNS) and ORAV bits are both set and the bad frame is loaded into R2-R so the controller's microprocessor can read it. The firmware can then choose to retry the transmission or notify the user of an error condition.

Fig. 5 and Fig. 6 flowchart the normal talker and controller responses to the interrupt bits. Note that although the flowcharts in Fig. 3 through Fig. 6 are basically correct, some of the details of the firmware interaction have been omitted for clarity.

HP-IL State Diagrams

The HP-IL protocol is defined in the same way as the HP-IB, that is, with state diagrams for the various interface functions such as controller, talker, listener, service request, et cetera. The state diagrams are all envisaged as independent asynchronous machines operating in parallel. However, the internal states of the sequential device microprocessor program and the sequential PLA-driven HP-IL interface IC are not at all similar to the state diagrams.

To explain this apparent difficulty, the purpose of the defining state diagrams must be understood. They are not

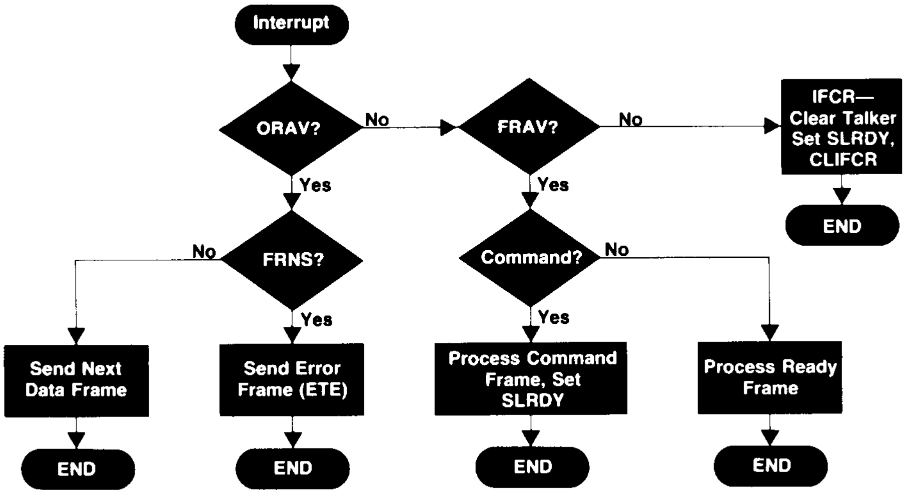


Fig. 5. *Simplified flowchart of the response of a talker on the HP-IL.*

intended as a set of internal design rules. They serve only as a precise model of the external behavior of a device on the HP-IL. The device is treated as a black box with access only through its HP-IL connector. If the device reacts in exactly the same way that a black box with the defining state diagrams inside would react, then the details of the internal design are not important. The device is functionally, electrically, and mechanically compatible.

To implement the HP-IL in a device the designer must first understand the state diagrams. Equally important is a thorough knowledge of what parts of the state diagrams are implemented automatically by the interface IC and what parts must be handled by the device's microprocessor firmware. The remote local interface function, for example, must be carried out totally by the program code in the microprocessor. The receiver interface function, on the other hand, is largely automatic.

A specific example may help clarify the situation. Fig. 7 shows the parallel poll interface function. The state diagram indicates that when power is applied, the parallel poll function must enter its idle state. In the device microprocessor's initialization routine for the interface IC, the firmware should make sure that bit four in register 3 (parallel poll enable PPEN) is cleared. The interface IC leaves this bit undefined at power-on and consequently, after the IC's oscillator is turned on, it might start responding to parallel polls when it should not unless the firmware prevents this.

The transition from the idle state (PPIS) to the standby state (PPSS) is also the firmware's responsibility. When a parallel-poll-enable command PPE is received, the device's microprocessor must make sure that the device is an active listener. If it is, then the least significant bits of the PPE command are decoded and put into register 3. Now the interface IC is ready to respond to parallel poll messages (identify (IDY) frames).

The actual response to the parallel poll, that is, entering the active state PPAS of the state diagram, is an automatic function of the interface IC. When an IDY frame is received, the appropriate bit in the frame is modified according to the information stored in register 3, the parallel poll register. The automatic retransmission of this frame by the interface IC represents the transition back to the standby state.

The move back to the idle state from the standby state is once again the responsibility of the device microprocessor's firmware. When a disable command PPD or an unconfigure command PPU is received, it must be decoded by the program and the proper bit cleared in the parallel poll register of the interface IC.

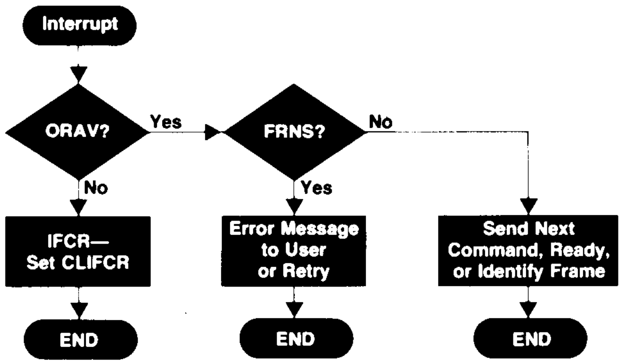
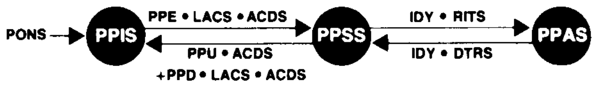


Fig. 6. *Simplified flowchart of the response of a system controller on the HP-IL.*

(a)



(b)

Messages:

IDY	Identify	PPE	Parallel Poll Enable
PPD	Parallel Poll Disable	PPU	Parallel Poll Unconfigure

(c)

Interface States:

PPAS	Parallel Poll Active State	ACDS	Acceptor Data State
PPIS	Parallel Poll Idle State	DTRS	Driver Transfer State
PPSS	Parallel Poll Standby State	LACS	Listener Active State
		PONS	Power on State
		RITS	Receiver Immediate Transfer State

Fig. 7. *(a) State diagram of parallel poll function. Mnemonic definitions for (b) messages and (c) interface states.*

Acknowledgments

The original architecture of the HP-IL interface IC and the sometimes tough job of making sure that the logic was capable of duplicating the external behavior of the state diagrams were done by Dave Sweetser and Tom Heger. Carl Landsness did the analog circuit design. Most of the digital logic and circuit design was done by Mike Pan. Project management was ably handled by Roger Quick.

CMOSC: Low-Power Technology for Personal Computers

To meet the growing need for integrated circuits with more functions and lower power consumption, an improved CMOS process has been developed at HP's Corvallis Division

by David E. Hackleman, Norman L. Johnson, Craig S. Lage, John J. Vietor, and Robert L. Tillman

AN IMPROVED HIGH-VOLUME CMOS (complementary metal-oxide-semiconductor) process has been developed at Hewlett-Packard's facility in Corvallis, Oregon. Required by the increasing integrated circuit complexity of personal computers, CMOSC meets several objectives that affect all phases of IC design and development. The objectives include:

- Low power consumption
- High device density
- Low circuit cost
- High reliability
- Improved latch-up and electrostatic discharge (ESD) protection
- Standardized process models with design rule checks.

To accomplish these goals, the project team had to develop a new bulk CMOS technology and new concepts in IC facility design. Adding spice to the challenge was a requirement to merge the CMOSC technology into an existing IC clean-room facility without disturbing production of the parent bulk-CMOS process.

CMOSC Process

Cross sections of a basic CMOSC structure at various points during processing are shown in Fig. 1. First, a thin oxide layer is grown on the silicon wafer surface and a layer of silicon nitride is deposited upon it. Photoresist is applied and patterned, using mask 1, to define the active regions. This pattern is plasma etched into the nitride layer (Fig. 1a). Mask 2 is then applied to define the p-well regions (Fig. 1b). A high-energy ion implantation of boron is done to set the threshold voltages of the n-channel transistors. The p wells are diffused to the desired depth by a long high-temperature cycle in a dilute mixture of oxygen in nitrogen. During this operation, a 100-nm-thick layer of silicon dioxide is grown in the field regions and subsequently is chemically etched away.

Photoresist is reapplied to the wafer and patterned, using mask 2 again, to mask the lateral-channel-stop implant (Fig. 1c, similar to an n-channel-process field implant). Another dose of boron in this manner establishes the n-channel field thresholds. After the photoresist mask is stripped, the entire wafer is subjected to an arsenic implant (not shown). The patterned nitride layer prevents the active regions from being implanted.

The field oxide is grown in steam. Because of the slower diffusion rate of oxygen through silicon nitride, oxidation takes place much more slowly for the areas covered by nitride. The result is islands of silicon, in which active devices will be formed, surrounded by regions of thick field oxide (Fig. 1d). The remaining nitride is chemically removed and a gate oxide 50 nm thick is thermally grown in an ambient of oxygen containing a small percentage of TCE (trichloroethylene).

Polysilicon is deposited (Fig. 1e) by low-pressure chemical-vapor deposition (LPCVD). The polysilicon layer is then removed from the backside of the wafer. In this manner, during phosphorus doping of the polysilicon, the backside of the wafer also becomes highly n type, improving defect gettering. The polysilicon is patterned by mask 3 and the exposed regions are plasma etched away (Fig. 1f).

The p-channel device regions are patterned using mask 4 (Fig. 1g). Photoresist is again used as an implant mask. Boron is implanted through the exposed areas of the gate oxide layer to form the p+ diffused source/drain regions. The polysilicon pattern protects the active p-channel device gate region from the implant, creating self-aligned gates.

The n-channel device regions are defined by mask 5 in a very similar manner (Fig. 1h). Phosphorus is implanted using the photoresist as an implant mask.

A short oxidation cycle at this point produces a thin, but very defect-free oxide layer over the p+ regions, n+ regions, and polysilicon. A phosphorus-doped oxide film is deposited as the intermediate insulator. By using an elevated temperature, this oxide is softened and flowed to ensure smooth steps for subsequent metal coverage.

Photoresist is applied and mask 6 is used to define the contact areas. Contacts are chemically etched through the phosphorus-doped oxide to the p+ diffusion, n+ diffusion, and n+ polysilicon regions (Fig. 1i). A silicon-aluminum alloy is sputtered onto the wafer and patterned using mask 7 and wet etching (Fig. 1j). The wafer is then passivated by a layer of plasma-deposited silicon nitride. Finally, openings to the bond pads are plasma etched through the nitride using mask 8 (not shown).

Low Power Consumption

A calculator such as the HP-11C (Fig. 2) may operate for about one year on a single set of batteries. The calculator's standby current is so low that batteries may be replaced

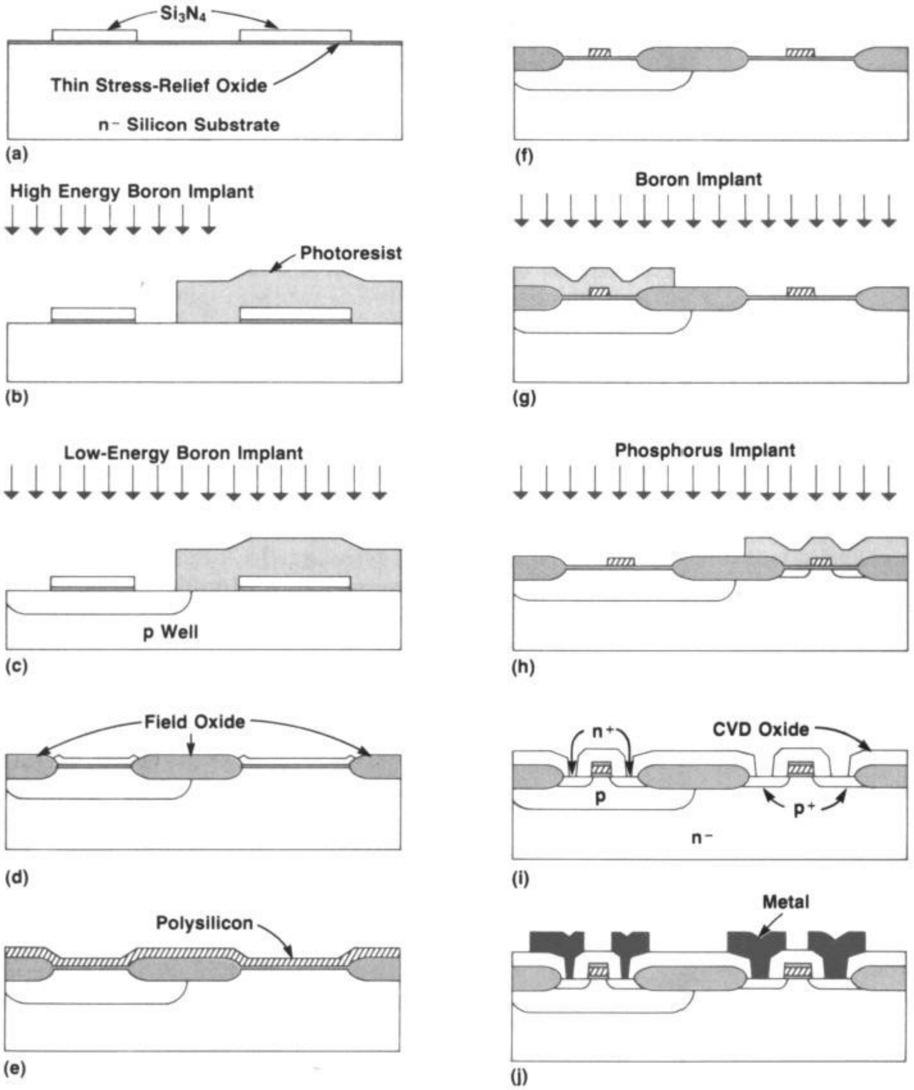


Fig. 1. *Cross-sectional views of the CMOSC process. (a) Active areas are defined by plasma etching a nitride layer (mask 1). (b) Photoresist is used to mask p-well implant (mask 2). (c) Mask 2 is used again to mask the lateral-channel-stop implant (to avoid subthreshold leakage current). (d) Field oxide is grown by local oxidation after the lateral-channel-stop implant. (e) Nitride layer is stripped after field oxidation and polysilicon layer is deposited (f) Polysilicon definition (mask 3) forms a self-aligned gate and is used as mask for source/drain implantations for p-channel devices and n-channel devices. (g) Photoresist (mask 4) is used to define p+ source/drain regions during boron implant. (h) Phosphorus is implanted for n source/drain regions (mask 5). (i) After contact etch (mask 6). (j) Metal deposition and patterning (mask 7). After this step, a passivating layer of silicon nitride is deposited over the circuit and bonding pad openings are etched using mask 8 (not shown).*

without fear of information loss during replacement. CMOSC circuits have an extremely low standby leakage current of 5 to 10 nA. To realize such a low leakage current, the CMOSC process has some critical design rules. For example, the distance from the edge of the contact hole to the edge of the active region (dimension a in Fig. 3a) is carefully controlled. Leakage characteristics were optimized by improving several process techniques through characterization of the growth of field oxide (Fig. 1d), improved control of the out-diffusion of the implanted source/drain junctions, and increased integrity of the gate oxide edge. Leakage current can be decreased by inhibiting the thinning of the field oxide "bird's beak" during the contact etch (dimension b in Fig. 3a). Thinning this oxide region below 50 nm can result in a nondestructive breakdown that increases leakage currents. Fig. 3b is an SEM (scanning electron microscope) photograph of an actual contact hole cross section.

High Device Density

The circuit density objective was to allow a typical calculator product to be designed with a minimum of IC chips. Specifically, we wanted a minimum three-fold increase in the functionality of a typical integrated circuit designed in the existing CMOS process. To achieve this increase, individual device dimensions were reduced so that three times as many devices can be fabricated in the same chip area. 1:1 projection scanning lithography is used to define the small patterns required. With this technique, a photomask consisting of an exact scale replica of the desired circuit pattern is imaged through a series of mirrors and projected on the photoresist-coated wafer as shown in Fig. 4a. The resulting image quality is shown in Fig. 4b. A 2.5-μm minimum linewidth with 0.4-μm tolerance can be registered to within 1.5 μm of previous layers across the entire 100-mm width of the wafers (all values are 3σ statistical range).[1]

The minimum linewidth and registration accuracy determine the minimum device spacing, while the width tolerance sets the variability of electrical device characteristics. During the photoresist exposure, partially coherent light predominantly composed of three wavelengths and emitted from a high-pressure mercury-arc lamp impinges on the multilayered structure of the wafer, photoresist, and any deposited or grown materials. The illumination inten-

Fig. 2. *The HP-11C: A calculator representative of the type using the CMOSC process.*

sity varies through the thickness of the photoresist because of standing-wave reflections of the three wavelengths between the top and bottom surfaces of the resist layer. The layer thickness is adjusted so that intensity minima lower than the contrast threshold of the photoresist do not occur at the bottom surface. The result is less scum (unexposed and therefore undeveloped resist) at the bottom of developed areas.

Photoresist is predominantly composed of large polymeric compounds. Proper pre-exposure treatment is necessary to retain a consistent photosensitivity. Careful control of coating and drying conditions is necessary. Positive photoresist acts unfavorably in an oven purged with pure nitrogen, forming a tough outer skin which can subsequently lift and redeposit in a different location during development. Therefore, clean processed air is used instead of pure nitrogen. Development of an image to 0.4-μm tolerance using a mask with 0.2-μm tolerance requires continuous control of the concentration and temperature of the developer and the time of development. Developer titration and adjustment to ±1% are performed frequently. The development is done in a batch process, an automatic operation that includes a nitrogen blanket and recirculating temperature control to better than 1°C.

Once the image is developed, the critical geometries of the silicon-nitride island before oxidation and the polysilicon gate pattern are plasma etched. A single-wafer parallel-plate plasma reactor using SF_6 as an etchant gas provides reproducible control of the transistor width (island) and length (gate). A typical cross section demonstrat-

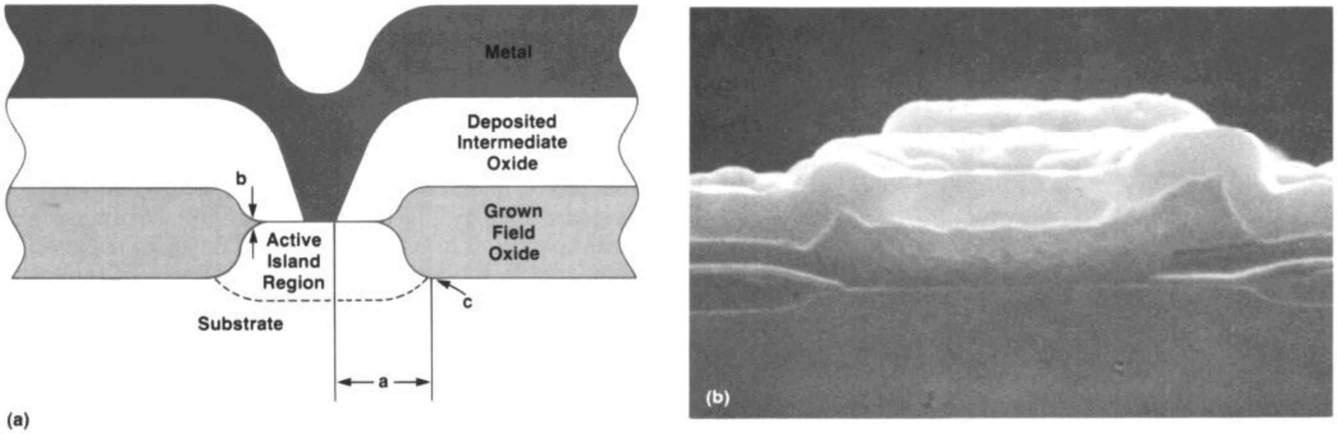


Fig. 3. *(a) Cross-sectional drawing of a typical CMOSC contact hole structure (scales distorted for clarity). (b) Photo of actual contact hole structure.*

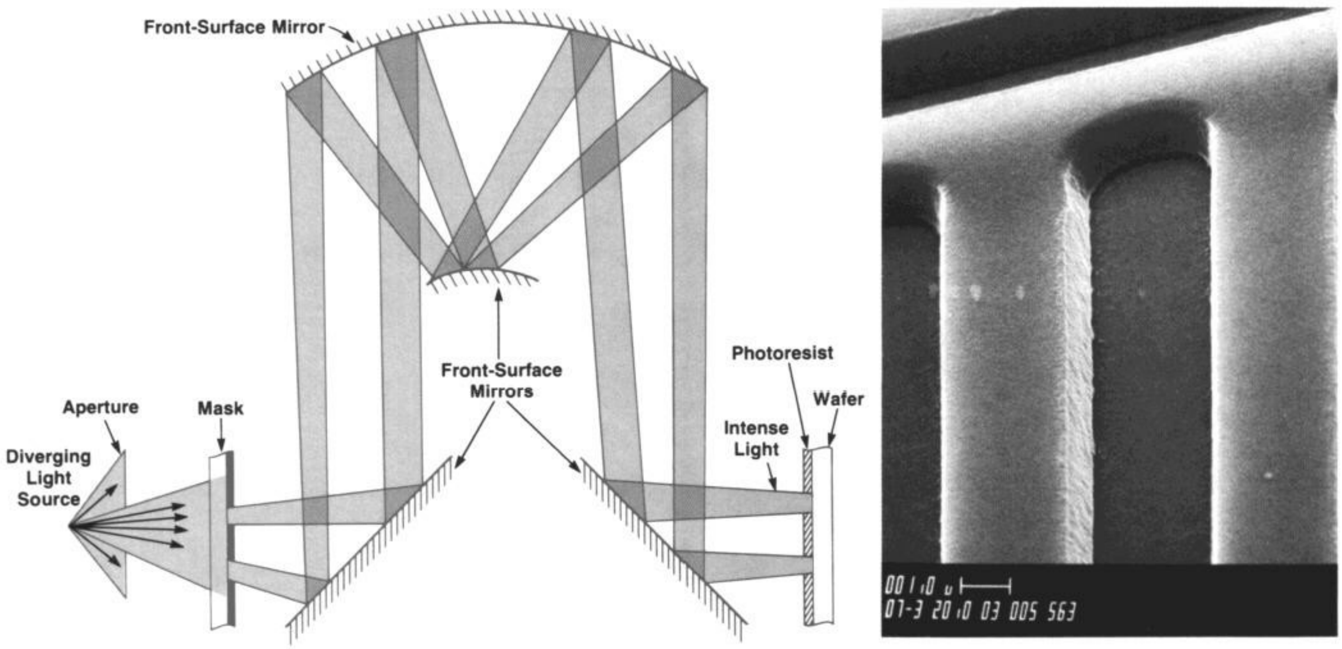


Fig. 4. *Outline of 1:1 projection printing method. (b) Photoresist images produced by 1:1 projection aligner.*

ing the reactor's etch performance for the doped polysilicon gate structure is shown in Fig. 5.

To avoid generation of surface interface charge states late in the process, it was decided to do the contact etch with a solution of aqueous HF. An unexpected problem developed. The diode junctions produced before the contact etch step have such low leakage that a built-in electric field is established in the circuit during the etch. This field can have the effect of stopping the removal of silicon dioxide in certain contact holes, namely n-type contacts connected to p diffusions in p wells. A detailed explanation of this effect is presented in reference 2. After this anomaly was eliminated, the contact hole etch process was defined using temperature control, filtering, and automatic handling.

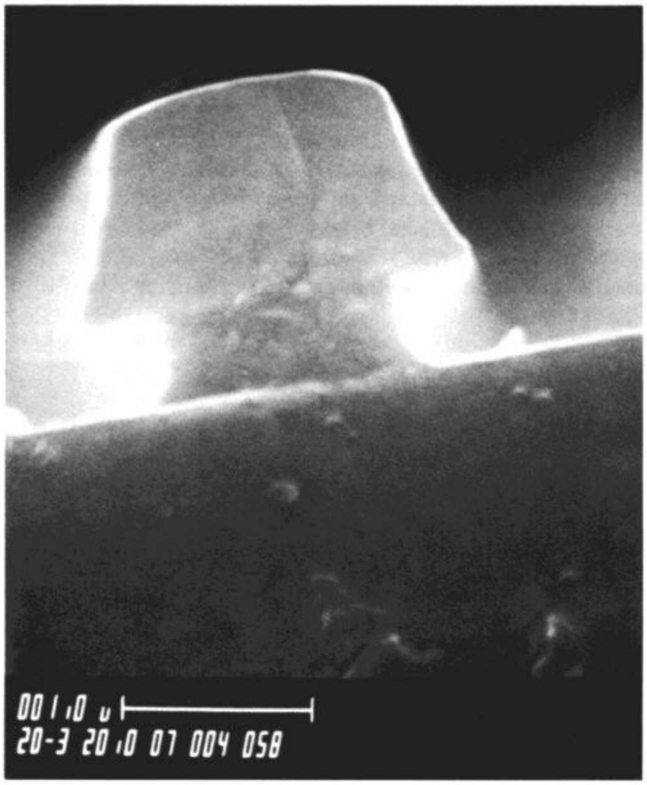


Fig. 5. *Plasma-etched polysilicon-gate cross section. Resist mask layer remains on top of the gate in this view.*

This lithographic development work for CMOSC has resulted in a typical ROM cell of 99 μm^2 area, a static RAM cell occupying 2020 μm^2 and a random logic density of 350 gates/mm^2.

Low Fabrication Cost

Low cost today means maximum use of resources. The development team began as a small group, increased in size to handle tasks as they were encountered, and is dissipating as the remaining problems are solved. The production force started as one operator, and now is a three-shift operation.

Before construction of the fabrication area began, scale models were used to find optimum locations for the equipment in the clean room. Work flow, safety, particle contamination, and supervisory feedback were all considered. The improvement in the efficiency of clean room space is demonstrated by Fig. 6. Fewer individuals can perform more operations in less space and less time for a net savings, compared to the older bulk CMOS process, of roughly a factor of 5.

Reliability

At each step of CMOSC, the acceptance criterion is no visible defects. Requiring complete coverage of all contact holes by metal (Fig. 1j) and locating passivation openings only above metal pads over field oxide removes several

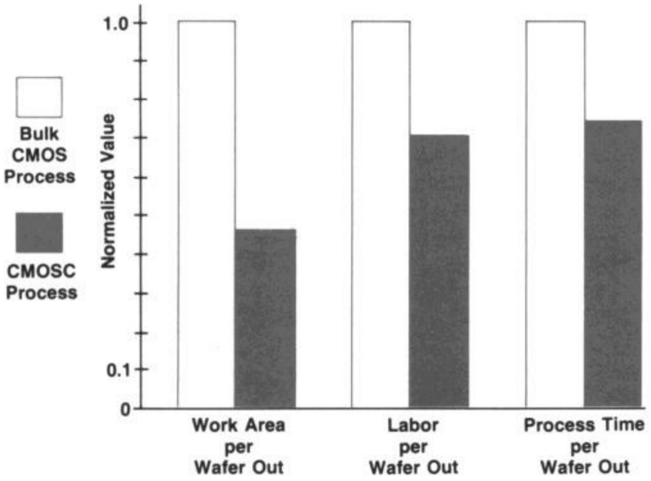


Fig. 6. *Efficiency of CMOSC clean-room use compared to previous bulk CMOS process. Data is based on a mature process in a facility running at full capacity. A wafer out means one wafer of devices fully processed into integrated circuits.*

pathways for ionic contamination. Special cleaning techniques at gate oxidation and polysilicon gate deposition help lower the defect density.

Oxide defects decrease reliability. A special intermediate oxide layer traps Group I metal contaminants (such as sodium) before they can enter the sensitive gate-oxide region. Use of such a layer is not common in most CMOS processes, giving CMOSC circuits an intrinsic reliability advantage. The effects of using these process steps have been characterized by more than 35 million device operating hours at 70°C. A continuing audit of CMOSC products at elevated temperatures insures against any change in device reliability.

Sixty monitors are used to help control the fabrication process. These range from distortion and focus monitors of the projection aligners[1] to etch-rate tests and charge inclusion measurements.[3]

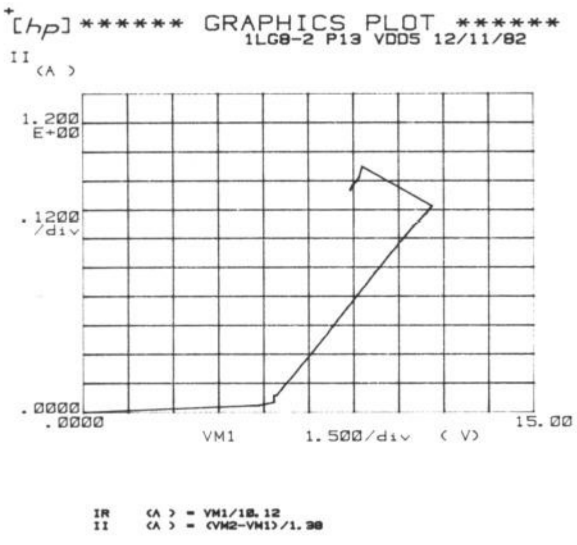


Fig. 7. *Latch-up current/voltage characteristic.*

Latch-up and ESD Protection

Latch-up is a fundamental problem of CMOS (see box). If latch-up is initiated, and the power-supply short-circuit current exceeds the latch-up sustaining current, a permanent low-impedance path between the supply and ground results. This causes a calculator system to lock up and rapidly drains its batteries. If the power supply cannot provide the required latch-up sustaining current, the latch condition dies away. In this case, a soft error may occur in the calculator system.

One design objective for CMOSC circuits was to force the latch-up sustaining current I1 to be greater than 200 mA. This would ensure that the calculator system batteries could not support a latched state. By combining a p-well architecture with a low-resistance substrate, and using conservative input/output circuit designs, a value of I1 greater than 500 mA was achieved (see Fig. 7). This level of latch-up hardness is sufficient to protect a calculator system from measurable electrical disturbances.

Electrostatic discharge (ESD) is another great danger to handheld systems. Fortunately, ESD susceptibility and latch-up hardness are linked. High values of I1 usually mean low susceptibility to ESD damage. (If oxide integrity is poor, this may not be the case.) The CMOSC input/output circuits are capable of withstanding ESD transients in excess of 4 kV. This is well above any level the circuit will normally experience once placed into the calculator system.

Summary

The first integrated circuit using the CMOSC process was produced in January 1981. With 85,000 transistors on a 0.27-cm^2 chip (Fig. 8) and a total operating power dissipation of 0.25 mW, it is truly low-power. This integrated circuit contains a 61K ROM, a 2.2K static RAM, a 100-

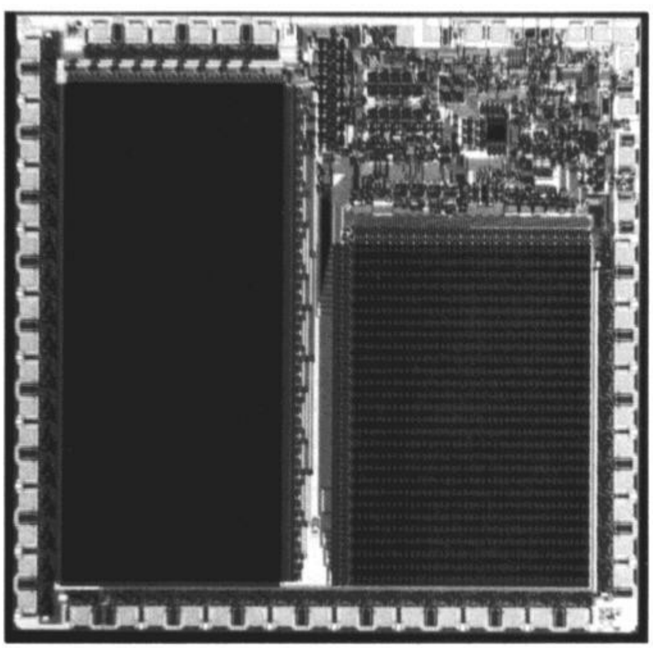


Fig. 8. *Photograph of a CMOSC circuit.*

What Is Latch-Up?

Given a standard CMOS device, as depicted by the inverter cross section in Fig. 1, one can draw a parasitic transistor-pair schematic as shown in Fig. 2.[1] These parasitic devices are not necessary for the functionality of the logic, but are a result of the structure obtained with standard CMOS processing. This circuit has the current-voltage relationship shown in Fig. 3. With no injected current I_J, the parasitic transistors have a high resistance. However, if enough current is injected into the n substrate and collected by the p well, the two transistors will switch to the low-resistance portion of the I-V curve. This stops functional operation of the circuit, and in a calculator will discharge the batteries. The circuit can sustain this low-resistance latch-up until the current drops below I1. An important objective in CMOS process design is to make V1 and I1 unattainable either during normal circuit operation or as a result of external stimuli.

Reference

[1] D.B. Esterich, "The Physics and Modeling of Latch-Up in CMOS Integrated Circuits," Technical Report No. G-201-9, Integrated Circuits Laboratory, Stanford Electronics Laboratories, Stanford University, Stanford, California, November 1980.

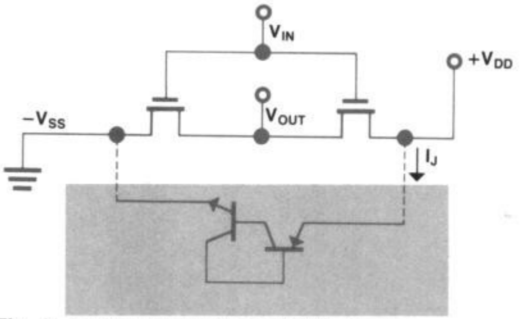


Fig. 2. *Equivalent circuit for CMOS inverter.*

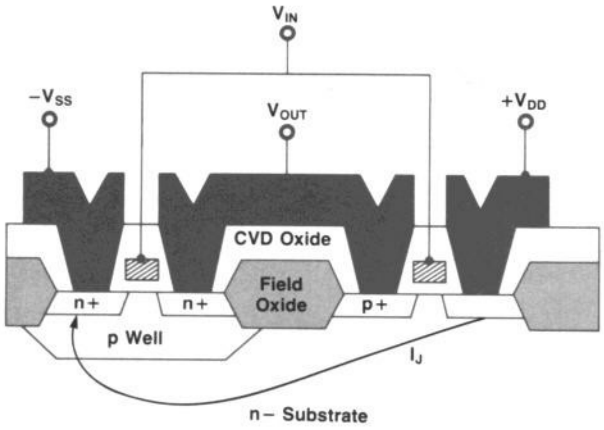


Fig. 1. *Cross section of CMOS inverter structure.*

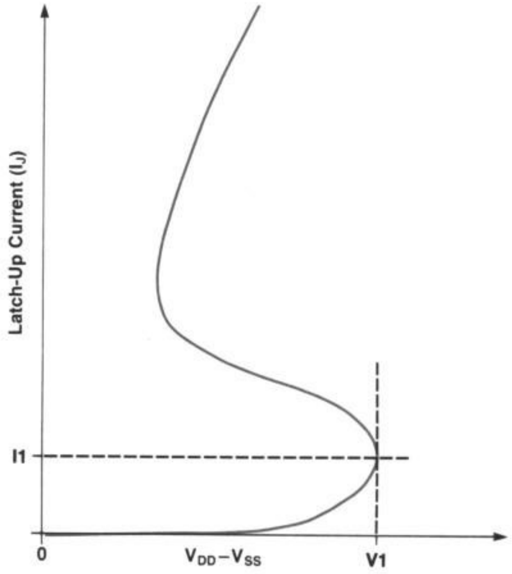


Fig. 3. *I-V characteristic for circuit in Fig. 2.*

segment liquid-crystal display driver, a clock, and an analog low-battery detector circuit.

Acknowledgments

Successful development of the CMOSC process would not have been possible without the dedicated efforts of the employees of HP's Corvallis Division, and discussions with people at other HP divisions. The CMOSC production staff, headed by Jim McMahon and the process engineering group under Gary Castleman are responsible for the dramatic increases in efficiency offered by CMOSC. Susan Swehosky extracted logical, meaningful phrases from our (very) rough draft. Without the rapid aid of Ellen Tappon and her associates in HP's Corvallis Components Operation Physical Analysis Lab, problems such as the contact etch effect discussed in the text would never have been solved as quickly and effectively. Rapid, effective maintenance was brought into the CMOSC area by Hugh Van der Huel and his staff. The quality assurance staff helped by performing long-term failure tests and are continuing to do a quality audit on the CMOSC process.

References

1. R. Hershel, et al, "Registration Monitor for 1:1 Aligners," Kodak Microelectronics Seminar Proceedings, October 20-21, 1980, San Diego, California.
2. H. Nielsen and D. Hackleman, "Some Illumination on the Mechanism of Silicon Dioxide Etching," presented at the Fall Electrochemical Society Meeting, Detroit, Michigan, October 1982, paper abstract #180.
3. K.H. Zaininger and F.P. Heiman, "The C-V Technique as an Analytical Tool," Solid-State Technology, May 1970, p. 49.

Norman L. Johnson

A native of Sioux Falls, South Dakota, Norm Johnson attended the South Dakota School of Mines and Technology and received a BSEE degree in 1966 and an MSEE degree in 1967. He continued his studies at Oregon State University and completed the requirements for a PhDEE degree in 1974. After three years working on MOS process development, Norm joined HP in 1977. He has worked on several of the CMOS circuits used in the HP-41C Calculator and presently is a project manager for part of the CMOSC process. Norm is married, has two sons, and lives in Corvallis, Oregon. His interests include woodworking and outdoor activities, particularly camping.

John J. Vietor

John Vietor joined HP in 1977 with a broad range of experience in silicon processing technology. His work at HP has included CVD process development and photofabrication for MOS. He currently is an MOS R&D equipment project leader at HP's Corvallis Component Operation. Born in Appleton, Wisconsin, John is a chemist, having received the AS degree in chemistry from Cabrillo College, California, in 1966. He is married, has two teenage daughters, and lives in Blodgett, Oregon where he has a wide variety of farm animals. Besides his interest in farming, he raises rainbow trout and serves as a volunteer fireman and on the citizens advisory board of the local county planning commission.

David E. Hackleman

A native of Coos Bay, Oregon, David Hackleman attended Oregon State University and received a BSEE degree in 1973. He continued his studies at the University of North Carolina and earned a PhD degree in chemistry in 1978. He then came to HP, and before assuming his current responsibility as a project manager of MOS lithography and plasma etching R&D, worked on the ICs for the HP-85 Computer and portions of the CMOSC process. His work has resulted in eighteen technical articles, papers, and invited lectures. David is a member of the IEEE, the American Chemical Society, and the American Radio Relay League and is a Member-at-Large for the Portland, Oregon section of the Electrochemical Society. Since 1979, he has given an annual lecture on semiconductor technology at the U.S. National Youth Science Camp. He also participates in the local amateur radio emergency service and is interested in astronomy and square dancing. David is married and building a home on 40 acres near Suver, Oregon.

Robert L. Tillman

A section manager for MOS R&D, Bob Tillman has also worked on GaAs FETs and silicon process technology since coming to HP in 1971. Before that, he worked on microwave IC design. He is the author or co-author of more than a dozen papers on silicon and gallium arsenide devices and technology. Bob received a BSEE degree from the Massachusetts Institute of Technology in 1969 and an MSEE degree from Stanford University in 1972. He is a member of the IEEE and Sigma Xi. Born in El Paso, Texas, he is married, has two children, and lives in Corvallis, Oregon. Bob's interests include reading mysteries, playing slow-pitch softball, and gardening when he is not busy working on his wine cellar and seeking the ultimate investment scheme.

Craig S. Lage

Craig Lage is a graduate of the University of Wisconsin at Madison, having received an MSEE degree in 1979 and an MS degree in nuclear engineering in l978. He also has a BS degree in physics awarded by the California Institute of Technology in 1976. Craig joined HP in 1979, worked on CMOSC, and now is a process enhancement project manager. Born in Hinsdale, Illinois, he is married and lives in Corvallis, Oregon. His outside activities are bicycling and playing handball.