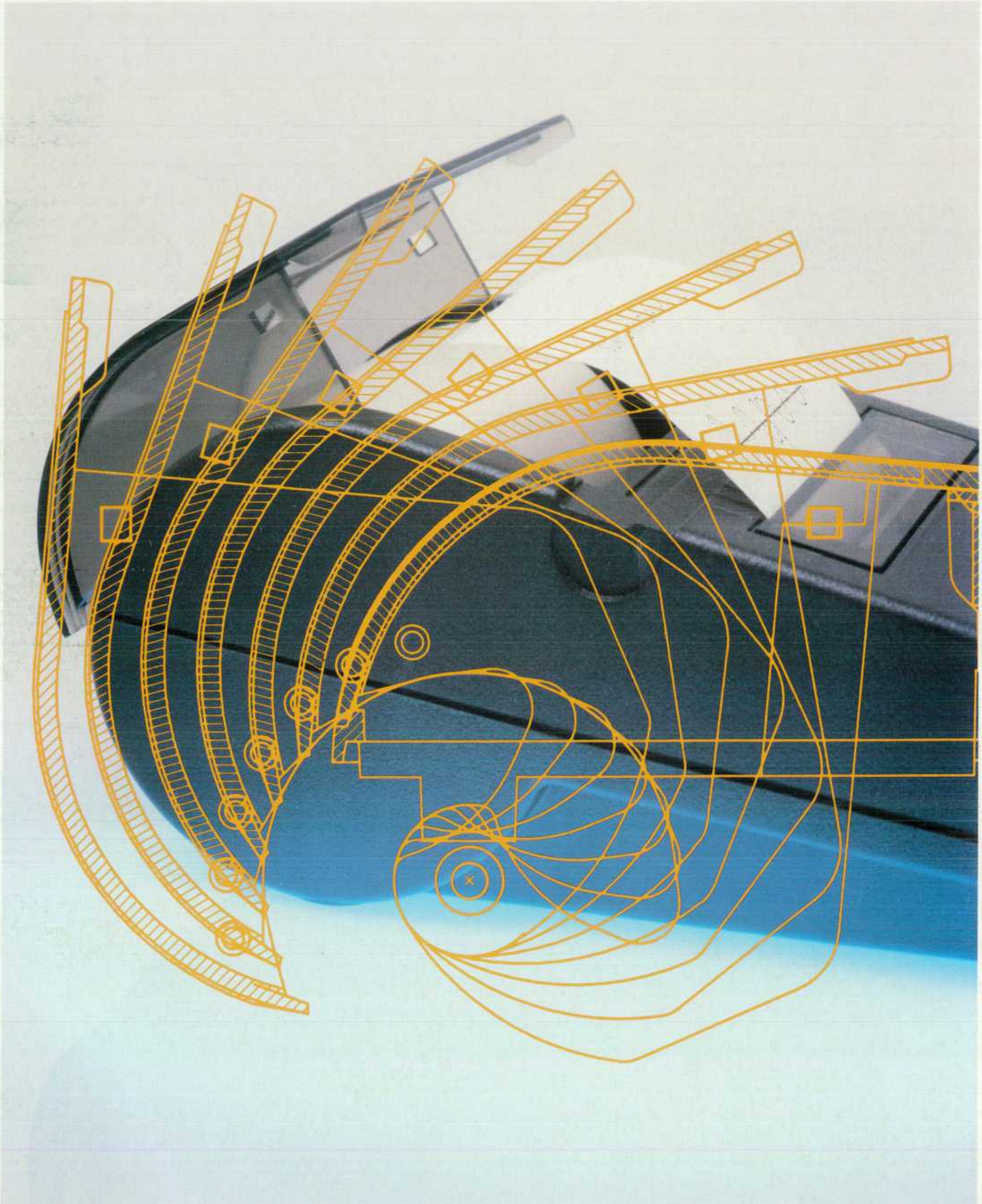
HEWLETT·PACKARD JOURNAL

OCTOBER 1987



HEWLETT-PACKARD JOURNAL

October 1987 Volume 38 • Number 10

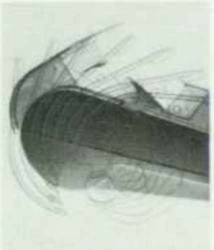
Articles

Research Report

Departments

Editor, Richard P. Dolan • Associate Editor, Business Manager, Kenneth A. Shaw • Art Director, Photographer, Arvid A. Danielson • Support Supervisor, Susan E. Wright
Administrative Services, Typography, Anne S. LoPresti • European Production Supervisor, Michael Zandwijken

In this Issue



In our computerized society, there's always a prodigious amount of data in motion, much of it on the voice-frequency circuits of the telephone network. To judge a voice-frequency circuit's suitability for carrying data, instruments called transmission impairment measuring sets (TIMS) transmit test signals and measure a circuit's effect on these signals according to standards set up by the CCITT or the Bell System, depending on the country. There's a trade-off in this picture—while a circuit is being tested, it isn't available for data transmission. The paper on page 4 describes the design and theory of operation of a new kind of TIMS, the HP 4948A In-Service TIMS, that does away with this test-or-use decision. The ITIMS attaches to any voice-frequency access point in a circuit, and without disturbing normal data transmission, measures the parameters that other TIMS measure. The measurements the ITIMS makes aren't the ones described in the standards, but the results are comparable. Its technique, which makes interesting reading, involves imitating a modem and setting up a model of the circuit in parallel with the modem.

Two months ago, in August, we presented the design story of two advanced handheld calculators, the HP-18C and the HP-28C. This month we have papers on the portable thermal printer for these calculators (page 21) and the infrared link between printer and calculator (page 16). The link is interesting because it's one-way only. The calculator sends data all the time, never knowing whether the printer is there. Link reliability is ensured partly by pacing the data transmissions so that even the slowest printer can keep up. As the paper on page 24 informs us, the production of the HP-18C and HP-28C called for technologies that were new to HP's calculator production operations, and for extensive use of computer-aided design (CAD) and manufacturing (CAM). The cover photo shows the printer with a CAD system simulation of its paper door rotation.

The explosion in medical knowledge has raised fears among medical educators that the amount of knowledge a medical student needs to absorb may be outstripping the time available to teach it. Harvard Medical School is attacking this problem with their New Pathway curriculum, which makes heavy use of HP computers, electronic mail, and computer-based training modules. On page 28, four Harvard staff members describe the new curriculum and some early results.

HP Laboratories' Bristol Research Centre is investigating formal methods of specifying system behavior, properties, and requirements, and is developing tools to provide machine support for rigorous software engineering. On page 37, two Bristol researchers report on their Framework for Program Development, a tool that facilitates the documented construction of correct programs.

-R.P. Dolan

What's Ahead

An Important Notice to Our Readers

After this month the Hewlett-Packard journal will become a bimonthly publication. Next month, November, there will be no issue. Subsequent issues will be in December, February, April, June, August, and October. Each issue will be about twice the present size. This change will allow us to take advantage of some economies of scale and put a greater variety of articles into each issue. Current subscribers will continue receiving their copies, and subscriptions will continue to be free. As always, your comments are welcome.

The December issue, in addition to the annual index, will have papers on the technical details of MPE XL, the commercial operating system for HP Precision Architecture computers such as the HP 3000 Series 930 and 950. Also featured will be the design of the HP 8780A Vector Signal Generator and the HP 8980A Vector Analyzer, which are stimulus/response instruments for testing digital microwave radio systems, and the HP 3065AT Board Test System.

The HP Journal encourages technical discussion of the topics presented in recent articles and will publish letters expected to be of interest to our readers. Letters must be brief and are subject to editing. Letters should be addressed to: Editor, Hewlett-Packard Journal, 3200 Hillview Avenue, Palo Alto, CA 94304, U.S.A.

In-Service Transmission Impairment Testing of Voice-Frequency Data Circuits

This transmission impairment measuring set operates on the live modem signal instead of special test signals. It warns of problems while the channel is in service, so preventive maintenance can be scheduled.

by Norman Carder, William I. Dunn, James H. Elliott, David W. Grieve, and W. Gordon Rhind

THE HP 4948A IN-SERVICE Transmission Impairment Measuring Set (ITIMS) advances the state of the art in analog data communications link testing by offering new possibilities for testing leased voice-frequency data circuits. Conventional test methods require that service be suspended while testing takes place. In-service testing allows the same measurements to be made directly on the modem signal without disrupting service. This means that if there is a problem, it can be investigated immediately without waiting until the circuit can be removed from service. The HP 4948A, Fig. 1, does not need a conventional test signal, but instead uses the live modem signal as its input.

To perform an in-service test, the HP 4948A is connected to any suitable voice-frequency access point in a network. Because of the high input impedance of the HP 4948A, the circuit is not disturbed. If the user does not know the type of modem in use, the instrument will try to identify it from a wide range of compatible modems. If the modem type is known, it can be entered manually to save time. After a short training time, the HP 4948A produces all its measurements simultaneously. This makes the instrument not only easy to use, but also very effective in troubleshooting.

The major benefits of the HP 4948A are obtained by using it to test circuits that are still in service. However, it can also be used on out-of-service circuits. The HP 4948A can transmit and receive a tone, so it can work with other tone sources or conventional test sets. The HP 4948A can also transmit a simulated modem signal to allow out-of-service testing with another instrument. This is useful for benchmarking a circuit with a modem-like signal.

Powerful features make the most of the HP 4948A's measurement capabilities. Extensive data logging makes long-term monitoring practical. The ITIMS can be left unattended for long periods, collecting and storing results in nonvolatile memory for later interrogation. To aid troubleshooting, the user can look back at intermittent faults after they have occurred. Because a circuit's performance can be characterized over a period of time, preventive maintenance is possible.

The HP 4948A can print current or logged results and instrument settings. It can also be included in a network monitoring system controlled by a computer. Circuits in one location can be monitored or results from a number of locations can be concentrated at a central site (see Fig. 2).

ITIMS Measurements

In the absence of channel impairments, the constellation of digital modulation states received by a modem consists of a number of well-defined points (see Fig. 3) and no data errors will occur. Various constellation patterns exist for the different modem modulation schemes. The number of

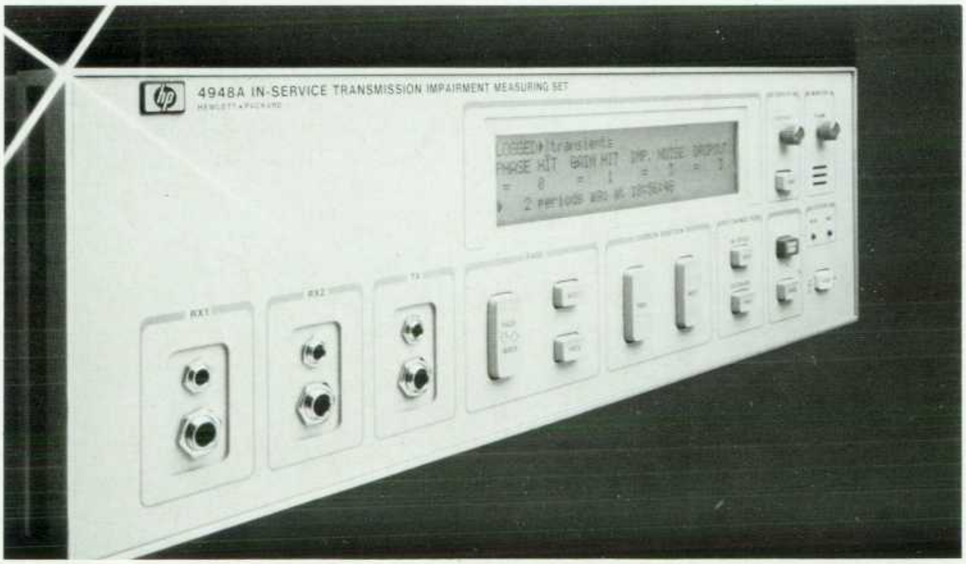


Fig. 1. *The HP 4948A In-Service Transmission Impairment Measuring Set connects to any suitable voice-frequency access point in a data communications network and makes measurements of amplitude and phase distortion, noise, jitter, gain and phase hits, impulse noise, and other impairments while the network is in use. Measurements are comparable to IEEE and CCITT standards.*

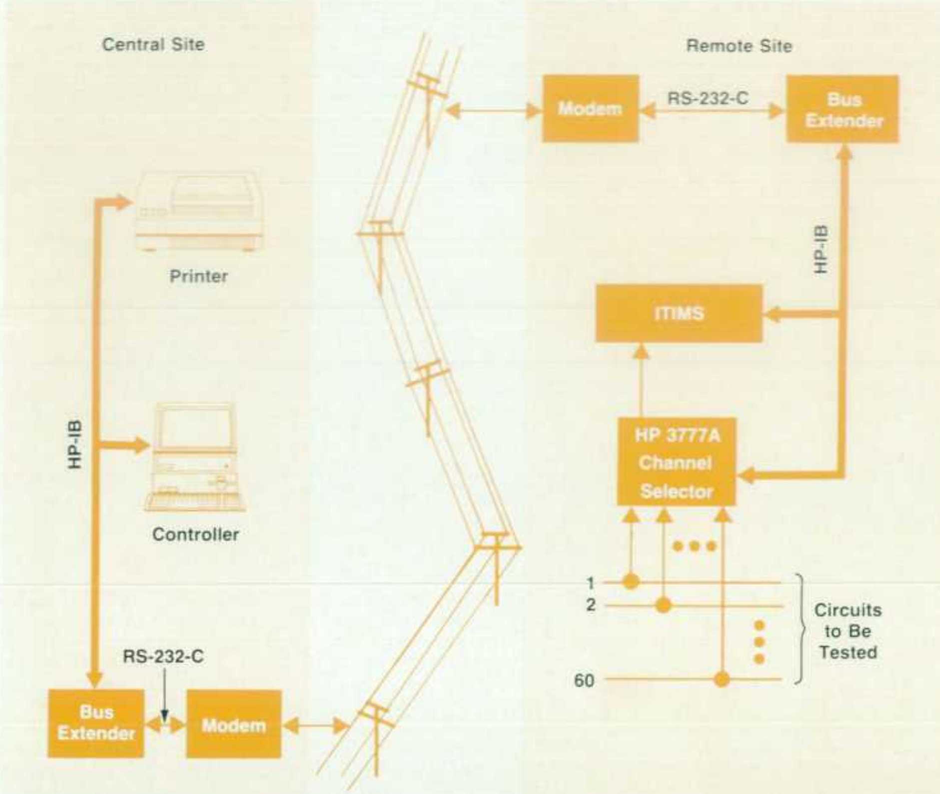


Fig. 2. *One or more HP 4948A ITIMS can be included in a network monitoring system controlled by a central computer.*

points in the constellation determines the number of data bits that can be transmitted per modulated symbol. For example, using a 16-point quadrature amplitude modulation (QAM) constellation, four data bits are transmitted per symbol, so a symbol rate of 2400 Hz allows a bit rate of 9600 bits per second.

In a real circuit, various impairments may be added to the modem signal, resulting in the possibility of data errors. These can be categorized as either steady-state or transient impairments. The HP 4948A measures the following steady-state impairments:

- Attenuation and delay distortion. This is caused by the nonideal frequency responses of cabling, filters, and equalizers in the circuit. The adaptive equalizer in the modem will remove these distortions, so they do not normally affect the modem constellation.
- Signal-to-noise ratio. Noise arises from circuit amplifiers, crosstalk from other circuits, or quantization in digital transmission facilities. Noise causes the received constellation points to become clouded rather than well-defined points (see Fig. 4a).
- Phase jitter and modulation. Phase jitter is normally considered to be the sum of phase modulation plus noise in the phase axis. The most commonly found modulation components are 20 Hz (ringing current), 60 Hz (commercial power), and up to the fifth harmonic of these (see Fig. 4b).
- Amplitude jitter and modulation. These are not as commonly observed as phase jitter, but may originate from the same sources (see Fig. 4c).
- Frequency offset. An offset between the transmitted carrier frequency and the received carrier frequency may exist because of differences in modulating and demodulating carriers in analog carrier systems. This is removed by the digital phase-locked loop in the modem receiver and so cannot be observed on the received constellation.

The HP 4948A measures the following transient impairments:

- Gain hits. These are sudden changes in signal power and may be caused by switching to standby facilities or carrier supplies. Fading or route changes in microwave facilities may also cause gain hits. The gain may return to its original level in a short time or remain at the new level.
- Phase hits. These are sudden changes in carrier phase and may be caused by route changes in the communications channel.
- Dropouts. These are considered to be negative gain hits greater than 12 dB.
- Impulse noise. These are short spikes of noise well above the background noise level. Impulse noise may be caused by switches and relays operating in the vicinity of the circuit. Impulses appear as abnormal points on the constellation.

Comparison with Standard Measurement Techniques

The measurement techniques used by the HP 4948A are not those described in CCITT and IEEE standards. However, the standards have been used as the basis for designing the measurements, and the results are broadly comparable with those produced using conventional techniques. Because the HP 4948A makes measurements on the live data signal rather than using artificial test tones, these measurements are in many cases a better indication of what is causing data errors in the modems being used. Because the HP 4948A makes

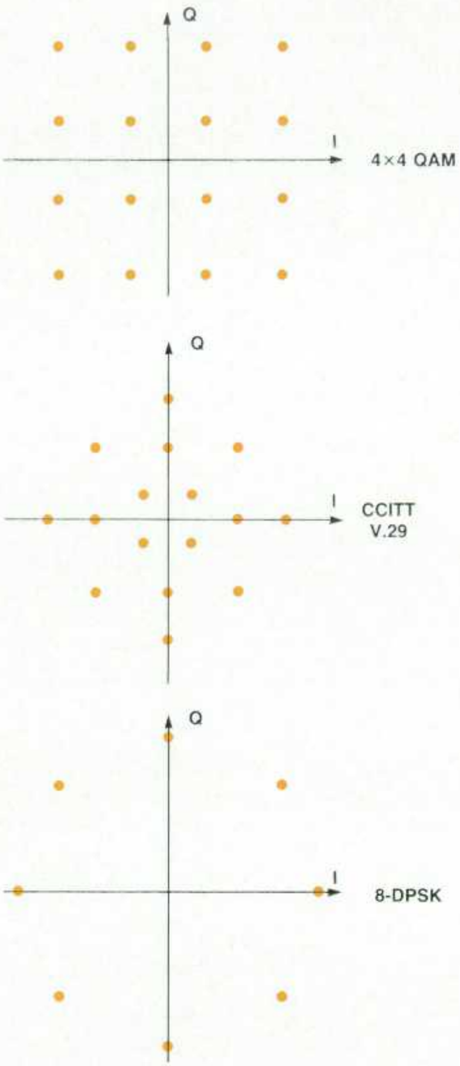


Fig. 3. *Typical impairment-free constellation diagrams for various digital modulation schemes. Each point represents one possible combination of in-phase (I) and quadrature (Q) modulation components.*

measurements in-service, it provides a new tool for maintaining and troubleshooting circuits without disrupting customers' data.

User Interface

On the front panel of the HP 4948A are a 4-line-by-40-character liquid crystal display (LCD) and ten keys. The LCD is used to display measurement results, setups, and status information. Functionally similar information is grouped together into pages for display purposes. With the addition of two index pages, the structure is similar to a book. Three front-panel keys are provided for the user to browse through this "book." **NEXT** and **PREV PAGE** keys allow pages to be turned, and an **INDEX/PAGE** key allows the user to select a page directly from one of the index pages (see Fig. 5).

To allow the user to communicate information to the instrument, active fields are used. These are defined as fields on the display at which a cursor can be positioned and an instrument parameter value changed. The form of such a field is:

PARAMETER ▶ Value

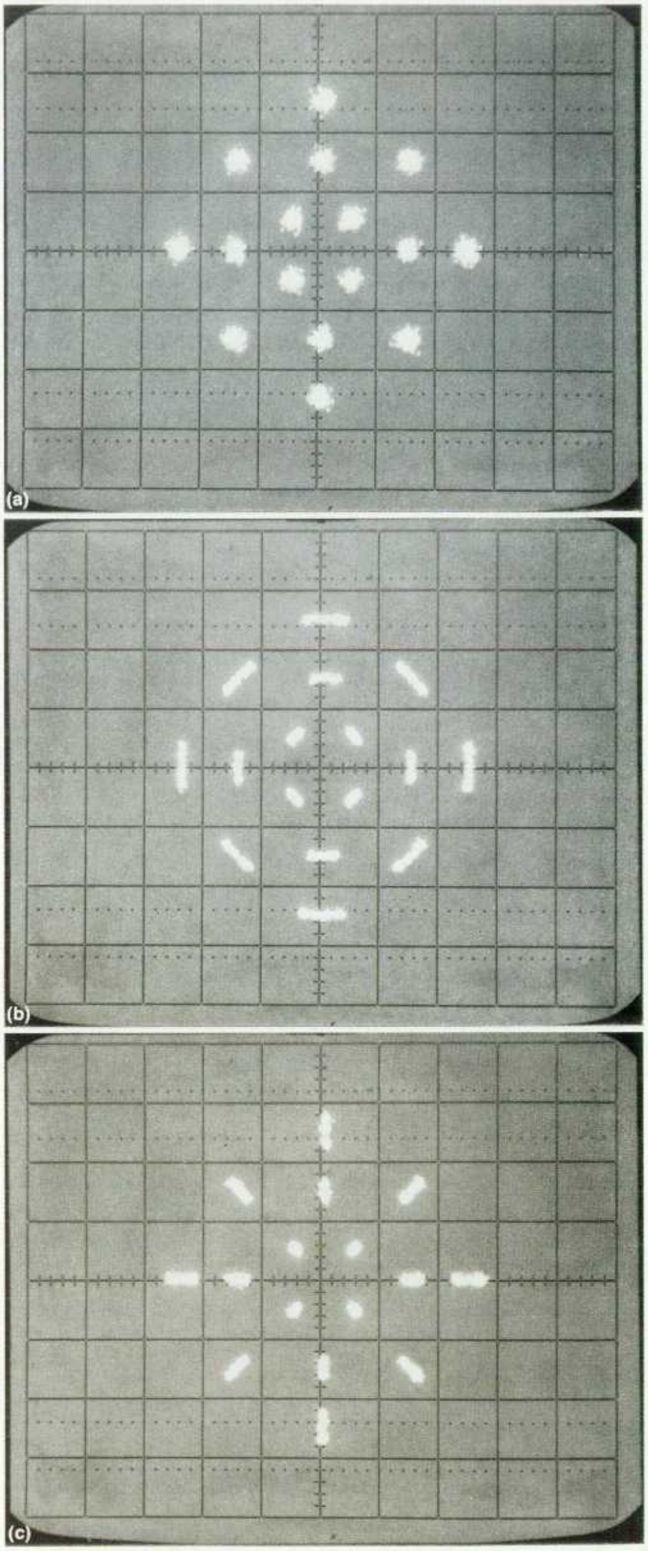


Fig. 4. *(a) Noise added to a CCITT V.29 constellation. (b) Phase jitter added to a V.29 constellation. (c) Amplitude jitter added to a V.29 constellation.*

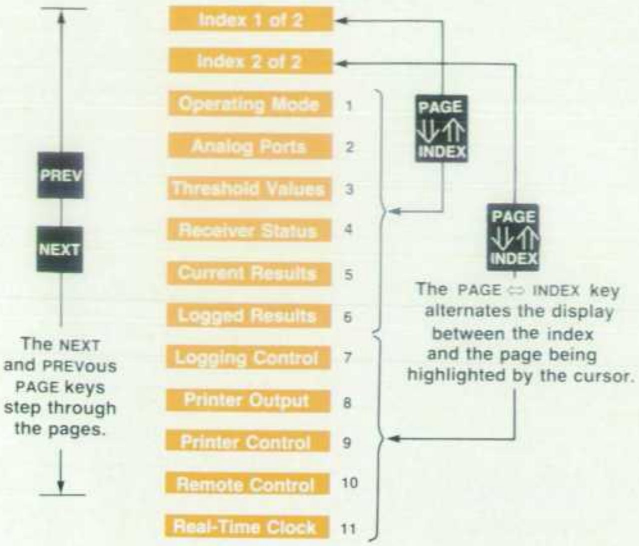


Fig. 5. *The HP 4948A user interface organizes functionally similar information into pages for display. The user can step through the pages or select one from the index pages.*

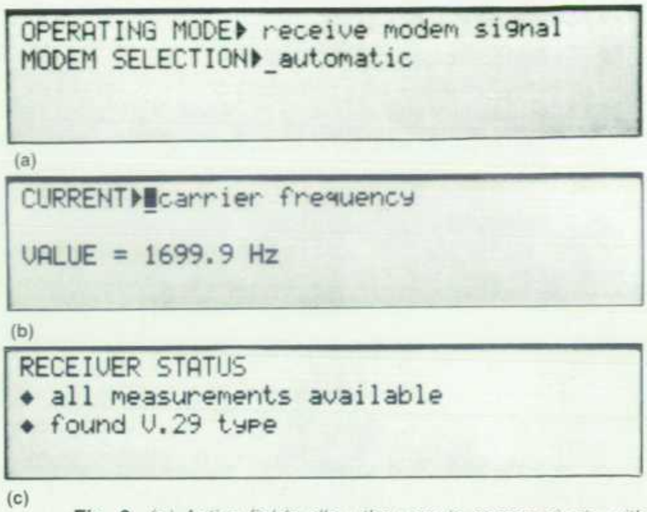


Fig. 6. *(a) Active fields allow the user to communicate with the instrument. Each field has a parameter name and a user-selected value. (b) Results pages are used by the instrument to return information to the user. (c) Status fields show various instrument conditions.*

NEXT and **PREV CURSOR** keys allow the cursor to be positioned at the field (denoted by a ▶) and the **NEXT** and **PREV CHANGE** keys allow the selection of parameter values (see Fig. 6a).

The instrument returns information to the user in the form of results and status reports. The results are given on the CURRENT (Fig. 6b) and LOGGED results pages. Each result can be displayed in turn by selecting the options on the results active field. Fig. 6c shows a status field, which is prefixed by the special character ◆. These special characters allow the user to identify ▶ as active fields and ◆ as status fields, and quickly distinguish between them.

Data Logging

The HP 4948A presents the steady-state measurements and transient counts as current results (see Fig. 7). When data logging is active, data reduction is performed on the current results and the receiver state to create log entries. The receiver state indicates whether all measurements are available and provides other information to the user.

The HP 4948A logs either periods or events. The logging store retains the most recent 1000 events or 200 periods for a single measurement session. Should a power failure occur during a logging session, logging is automatically resumed following power restoration.

The log contents can be displayed on the instrument front panel or printed on an HP-IB or RS-232-C printer. Alternatively, logged results can be extracted by either HP-IB or RS-232-C remote control.

Period Logging

Period logging provides regular measurement summaries at intervals of 5, 15, 30, or 60 minutes. The period average and maximum and minimum values of steady-state measurements are recorded, together with the total number of transients occurring during the period (see Fig. 8).

Warning messages are provided to indicate abnormal conditions (e.g., power failure) or to provide further information (e.g., fallback, receiver training). This extra information allows the user to assess the significance of the logged measurements and thus provides a complete picture of circuit performance during the period.

If a period is interrupted by a power failure, all results acquired during the active parts of the period are preserved

```
--------------------------------------------------------------------------------
28 Jan 1987 14:24:15

CURRENT STATUS
* all measurements available


           STEADY STATE MEASUREMENTS                        TRANSIENTS
 ---------------------------------------------      ------------------------
                value                               TIMER  00:08:35

 sig level      -14.9 dBm                                 phase hits      6
 s/n ratio       38.1 dB                                   gain hits      1
 carr freq     1700.0 Hz                                    impulses     35
 phase mod        0.0 deg                                   dropouts     11
 ph jitter        0.6 deg                           ------------------------
 ampl mod         0.0 %                                     retrains      0
 ampl jitt        0.5 %                            data rate changes      0
--------------------------------------------------------------------------------
```

Fig. 7. *An HP 4948A printout of current results.*

```
RESULTS FOR PERIOD STARTED: 17 Feb 15:59:13                            HP4948A
                                                    WARNINGS
                                          Pf/Cl Siglo Sighi NoSig WSig/ExIm IntSig
                                            0     0     0     0       0       0
                                          Train Fback NRnd TimeOff CarrOff  Stop
LOGGED FOR  83% OF PERIOD                   #     0     0     0       0       0

         STEADY STATE MEASUREMENTS                                 TRANSIENTS
-------------------------------------------------     ------------------------
            average      maximum      minimum

sig level    -15.5        -15.3        -15.7 dBm             phase hits      0
s/n ratio     26.9         27.2         26.8 dB               gain hits      0
carr freq   1700.0       1700.0       1700.0 Hz                impulses      0
phase mod      4.3          6.2          3.4 deg               dropouts      0
ph jitter      2.6          4.8          1.7 deg      ------------------------
ampl mod       0.0          0.1          0.0 %                 retrains      0
ampl jitt      1.7          2.6          1.2 %        data rate changes      0
```

Fig. 8. *Period log printout.*

in the log.

Event Logging

In event logging, the emphasis is on highlighting deviations from acceptable circuit performance and accurately time stamping their occurrence (see Fig. 9).

Once every second, each steady-state measurement is compared with limits set by the user and a specific alarm is registered for each limit violation. An event log entry is created corresponding to every second for which any of the following conditions is detected:

- Limit alarm
- Transient
- Change in receiver state
- Power failure
- Recovery after power failure.

How the ITIMS Makes Measurements

The HP 4948A can be considered to be a universal modem receiver that is able to receive signals from many commonly used modems and make measurements at the same time. It can train onto the modem signal without needing the synchronizing signal required by standard modems. Measurement processing is then performed to make measurements of the channel impairments. The HP 4948A only analyzes the line signal; it does not decode the re-

```
..HP4948A Event Log.........................................................

                                                                          rx
   event time              alarms                    transients          state
---------------   -------------------------   ----------------------    -----
                  LEV S/N FRQ PHM PHJ AMM AMJ   PHH GNH IMP DRO RTR DRC
---------------   -------------------------   ----------------------    -----

17 Feb 16:13:03    -   -   -   -   -   -   -     0   0   0   0   0   0      2
17 Feb 16:13:04    -   -   -   -   -   -   -     0   0   0   0   0   0      6
17 Feb 16:13:09    -   -   -   -   -   -   -     0   0   0   0   0   0      9
17 Feb 16:13:52                                  0   0   0   0   0   0     10
17 Feb 16:14:29                                  0   0   1   0   0   0     10
17 Feb 16:15:06                                  0   0   1   0   0   0     10
17 Feb 16:15:28                                  1   1   1   0   0   0     10
17 Feb 16:15:29    L                             0   0   2   0   0   0     10
17 Feb 16:15:35                                  0   0   3   0   0   0     10
17 Feb 16:15:36                                  1   0   3   0   0   0     10
17 Feb 16:15:37    U                             0   1   0   0   0   0     10
17 Feb 16:16:19                                  0   0   1   0   0   0     10

LOGGING STOPPED AT: 17 Feb 1987 16:16:39
```

Fig. 9. *Event log printout.*

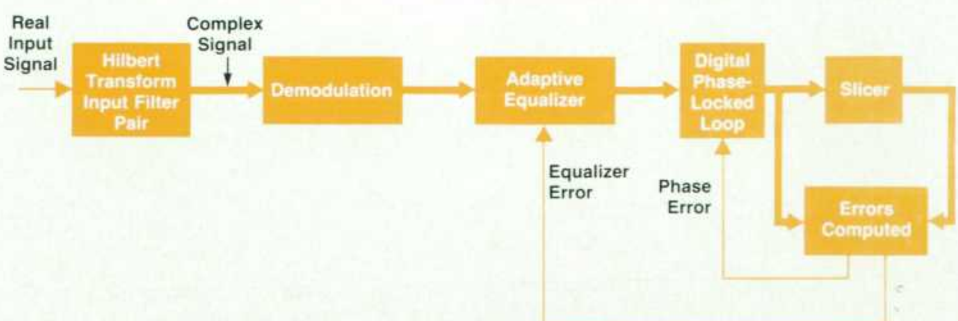


Fig. 10. *Block diagram of a modem receiver for a voice-frequency data communications channel. The HP 4948A performs all the same functions along with measurement functions.*

ceived data, so data security is not threatened.

A typical modem signal has a passband spectrum centered at its carrier frequency (typically 1700 or 1800 Hz) and a bandwidth equal to its symbol rate (also known as the baud rate).

The first stage of a modem receiver (see Fig. 10) consists of a Hilbert transform pair of passband input filters (see "Processing Passband Signals in Baseband," page 11). These filters band-limit the input to the bandwidth of the received signal and also transform it into a complex signal having in-phase (I) and quadrature (Q) components which are then demodulated by the expected carrier frequency.

Equalization removes attenuation or delay distortion present in the channel and provides an equivalent Nyquist frequency response which allows the equalizer output to be sampled at the symbol rate without intersymbol interference.

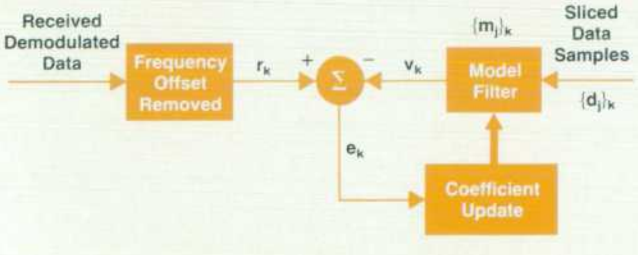
If the expected carrier frequency used in demodulation is not exactly the same as the carrier frequency of the transmit modem, or if some frequency offset has been introduced by the channel, a further demodulation is required. This is provided by a digital phase-locked loop (see "Digital Phase-Locked Loops," page 15).

The I and Q components at the output of the digital phase-locked loop can be plotted to give the modem's received constellation pattern. This is sliced to decide which point has been transmitted. (The slicer is basically a comparator network that determines which state on the constellation diagram the received symbol belongs to.) Various errors are computed and used to update the adaptive equalizer (see "LMS Algorithm for Equalizer Update," page 13) and the digital phase-locked loop. In this way, slow changes in the channel characteristics are tracked by the modem receiver.

The HP 4948A performs all of the modem receiver functions shown in Fig. 10, along with additional measurement functions.

Attenuation and Delay Distortion Measurements

In the modem receiver, attenuation and delay distortion are equalized to enable data to be received. To measure attenuation and delay distortion, the HP 4948A uses the received demodulated data at the input of its adaptive



$$v_k = \sum_{j=0}^{N} m_j d_{k-j}$$ Model Filter Computation

$$e_k = r_k - v_k$$ Error Computation

$$m_{k+1,j} = m_{k,j} + K_m e_k d^*_{k-j}$$ Coefficient Update

Fig. 11. *The HP 4948A sets up a channel model in parallel with the adaptive equalizer.*

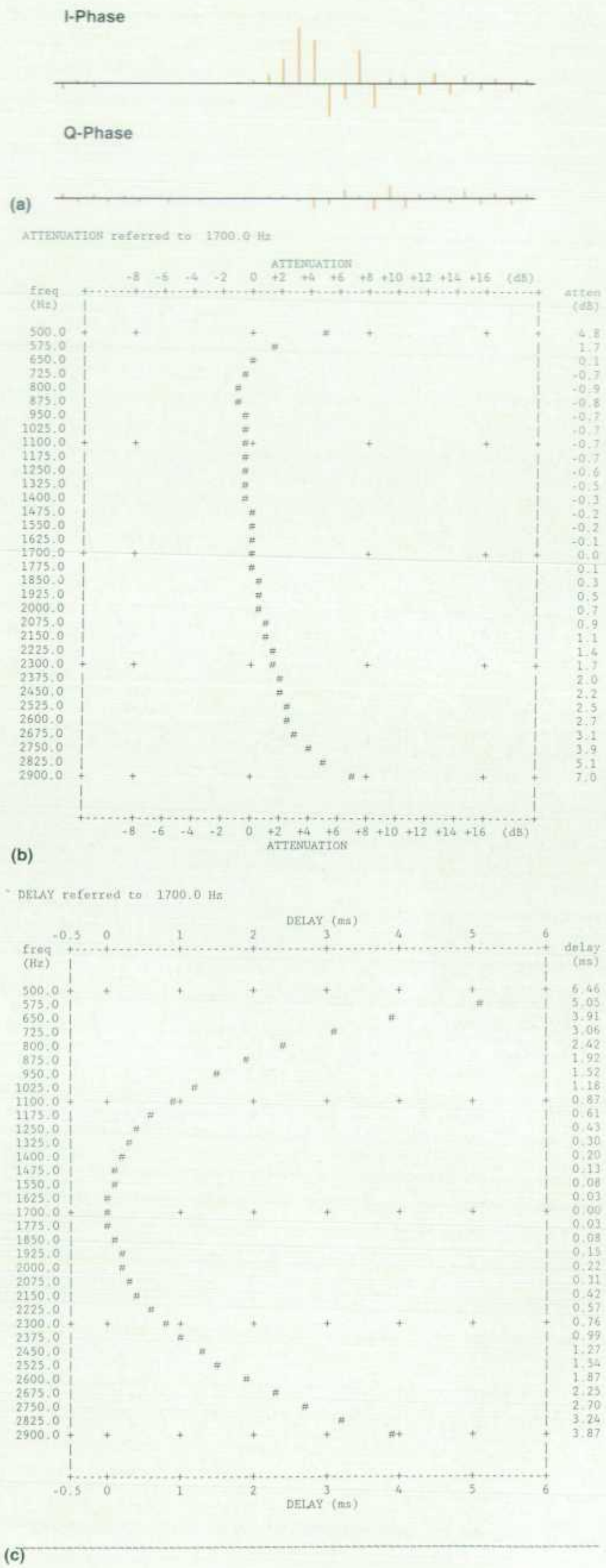


Fig. 12. *(a) Model tap coefficients for a channel having both attenuation and delay distortion. (b) Attenuation printout for the same channel. (c) Delay printout for the same channel.*

equalizer and the samples from its slicer. The HP 4948A sets up a channel model in parallel with the adaptive equalizer, taking its input from the sliced data samples in the modem receiver path (see Fig. 11). The model filter is similar in structure to the adaptive equalizer. The output of the model filter is compared to the received demodulated data (first rotated in phase to remove frequency offset) and the resulting error is used to update the model filter coefficients using the least mean squares (LMS) algorithm (see "LMS Algorithm for Equalizer Update," page 13.)

The model filter is a digital transversal filter having 128 complex coefficients. In the steady state, the coefficients represent the sampled equivalent baseband impulse response of the channel. By performing a 128-point fast Fourier transform (FFT) on these coefficients, the amplitude and phase responses of the channel are obtained. Differentiation of the phase response gives the delay response. These results are presented at 33 points centered on the modem carrier frequency and covering the data bandwidth of the modem signal. In this way only distortion likely to cause problems to the modem is measured.

Fig. 12a illustrates the steady-state model filter coefficients of a channel containing a combination of attenuation and delay distortion. Figs. 12b and 12c show the attenuation and delay of the channel printed out by the HP 4948A.

Phase and Amplitude Distortion Measurements

The HP 4948A measures phase jitter in the band 4 to 20 Hz and phase modulation in the band 20 to 300 Hz. The phase modulation measurement uses adaptive signal processing to suppress the background noise, and can extract modulating tones that may not be seen on the constellation display. Fig. 13 shows a V.29 constellation with a 24-dB signal-to-noise ratio and five degrees of phase modulation. The modulation is not easily recognized by visual inspection and cannot be measured by a conventional measuring set. The HP 4948A separates these two impairments and makes separate measurements of noise and phase modulation.

To reconstitute the input phase error without frequency shaping, the phase error value from the digital phase-locked loop is added to the output phase value. This provides a wideband phase error which is filtered to provide the different bandwidths required for the phase measurements (see Fig. 14). The phase error enhancer is an adaptive filter

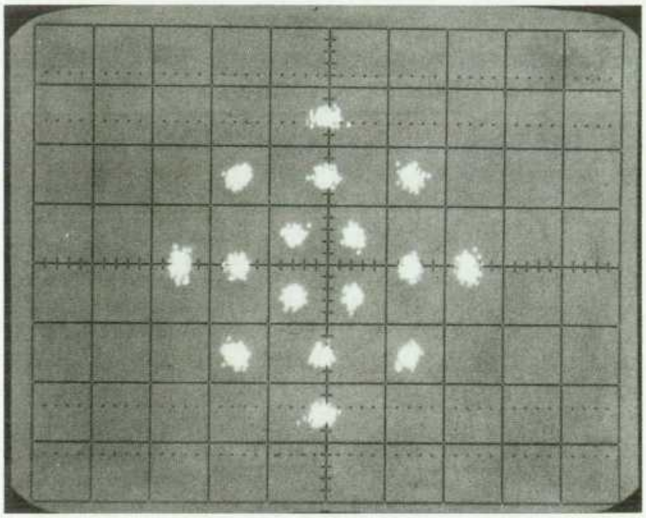


Fig. 13. *V.29 constellation with 24-dB signal-to-noise ratio and 5° of peak-to-peak phase modulation.*

configured to provide the modulation measurement described above. Because the phase error is a purely real signal, the enhancer filter is a real filter and not complex valued. The enhancer consists of a transversal filter preceded by a delay element (see Fig. 15). The error is computed as the difference between the input phase error signal and the filter output, and the filter coefficients are updated using the LMS algorithm. The coefficients settle to a response that enhances coherent signals present in the phase error. The frequency response of the filter is a bandpass filter (or set of bandpass filters) tuned to the modulating frequency or frequencies present and so the background noise is suppressed.

To illustrate the response of the adaptive enhancer, a signal having phase modulation at 60 Hz plus odd harmonics was applied to the HP 4948A. The steady-state coefficients of the phase error enhancer were recorded and an FFT performed on them by a computer (Fig. 16a). The resulting frequency response, Fig. 16b, shows how the adaptive filter tuned its response to let only the modulation components pass. In the absence of any true phase modulation, the enhancer taps will tend to zero. Hence, background noise alone will not cause any measurement of

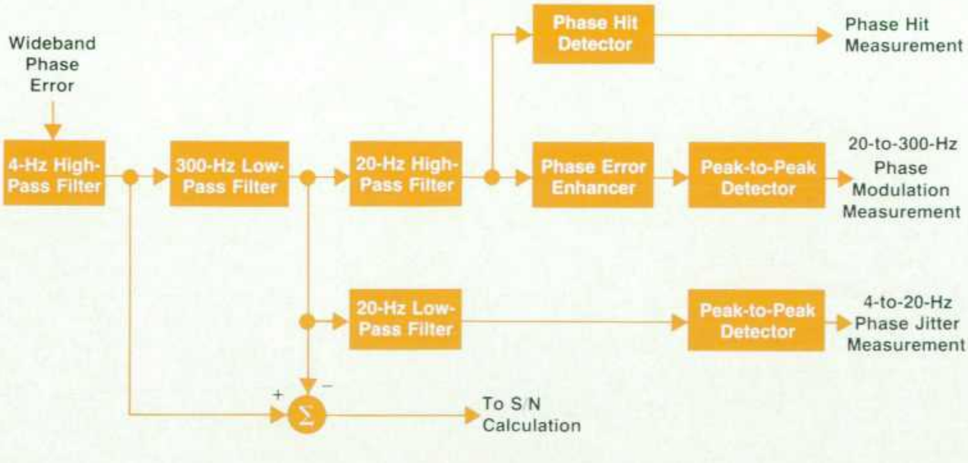
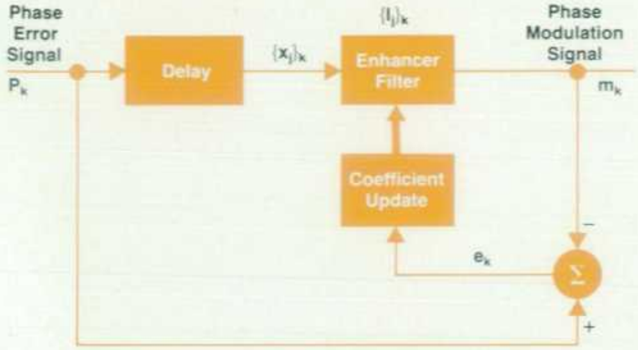


Fig. 14. *Phase measurement processing.*



$$m_k = \sum_{j=0}^{N} l_j x_{k-j} \quad \textbf{Enhancer Filter Computation}$$

$$e_k = P_k - m_k \quad \textbf{Error Computation}$$

$$l_{k+1,j} = l_{k,j} + K_l e_k x_{k-j} \quad \textbf{Coefficient Update}$$

Fig. 15. *Adaptive line enhancer enhances coherent signals present in the phase error while suppressing noise.*

phase modulation.

The time required to perform the coefficient update for the various adaptive filters (equalizer, line model, enhancers) averages about 2.5 μs per complex coefficient and about 1 μs per real coefficient. This gives a total time of about 600 μs to perform the coefficient update for all these filters. Where the data symbol rate is 2400 Hz, there is not enough real time available to update each filter once per symbol interval. A multiplexing procedure is therefore used, whereby each adaptive filter is updated once per n symbol intervals. To ensure that the n symbol intervals are not synchronous with any coherent modulation impairment present, the multiplexing is randomized.

The frequency offset from the expected carrier frequency can be derived from the second-order point of the digital phase-locked loop. This is then added to the expected carrier frequency to provide an accurate measurement of the received carrier frequency.

An amplitude error is also computed between the slicer output and the received signal. This is then used to provide measurements of amplitude jitter (4 to 20 Hz) and amplitude modulation (20 to 300 Hz) similar to the phase measurements. To make a measurement of the signal-to-noise ratio that will not be affected by the presence of phase or amplitude modulation (normally <300 Hz), the phase and amplitude errors in a bandwidth greater than 300 Hz are computed. These are then combined and scaled to compensate for the missing section of the signal bandwidth to give the signal-to-noise ratio measurement.

Transient Measurements

Making transient measurements in-service on a modem signal poses some problems not encountered when making out-of-service tests using a tone. Because normal data consists of random changes in phase and amplitude, a transient may not be distinguishable from normal data changes. However it is still possible to make useful measurements.

Impulse noise is recorded whenever the error power of

Processing Passband Signals in Baseband

A passband signal s(t) can be described by the time-domain notation:

$$s(t) = s_+(t) + s_-(t)$$

where $s_+(t)$ and $s_-(t)$ are the inverse Fourier transforms of the positive and negative spectra $S_+(f)$ and $S_-(f)$, respectively (see Fig. 1a).

The Hilbert transform $\bar{s}(t)$ of the passband signal s(t) can be written as:

$$\bar{s}(t) = -js_+(t) + js_-(t)$$

and can be produced by a −90° all-pass network. A Hilbert transform filter pair, therefore, consists of two filters, both having flat amplitude response across the data bandwidth, but with a −90° phase difference.

Combining s(t) and the output $\bar{s}(t)$ of this Hilbert transform filter pair results in the complex-valued analytic signal:

$$a(t) = s(t) + j\bar{s}(t) = 2s_+(t).$$

This isolates the positive-frequency spectral component of s(t).

A negative frequency shift of the spectrum of the analytic signal by an amount f_c produces r(t), the complex-valued baseband equivalent of s(t):

$$r(t) = a(t)\exp(-j2\pi f_c t) = r_i(t) + jr_q(t)$$

where $r_i(t)$ and $r_q(t)$ are, respectively, the in-phase and quadrature-phase components of r(t):

$$r_i(t) = s(t)\cos(2\pi f_c t) + \bar{s}(t)\sin(2\pi f_c t)$$

$$r_q(t) = \bar{s}(t)\cos(2\pi f_c t) - s(t)\sin(2\pi f_c t)$$

The Fourier transform R(f) of r(t) is shown in Fig. 1b. Digital processing of the sampled complex-valued baseband signal r(t) is possible, provided that the sampling frequency f_s is greater than or equal to f_b, the information bandwidth of the passband signal.

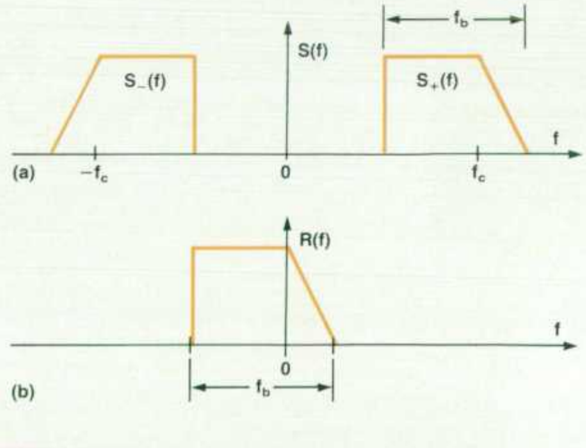


Fig. 1. *Demodulation of a passband modem signal.*

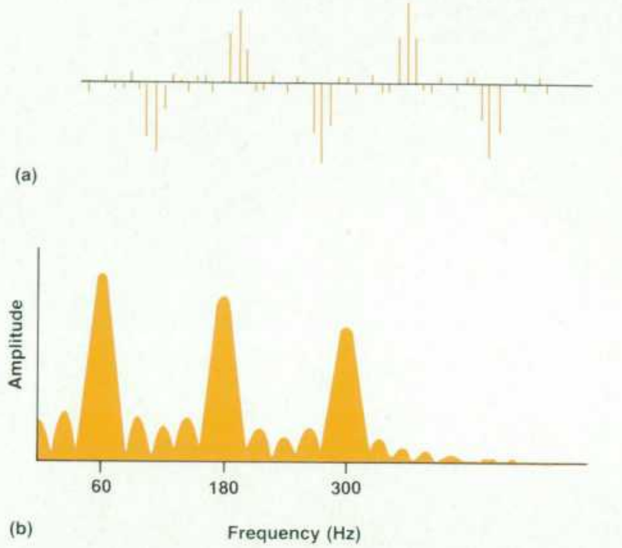


Fig. 16. *(a) Tap coefficients of the phase enhancer when 60-Hz phase modulation and odd harmonics are present. (b) Frequency response of the phase enhancer.*

the received point exceeds a threshold set by the user. The impulse noise thresholds available in the HP 4948A are with respect to the received signal level. Thresholds of −18, −12, −6, and 0 dB are available.

On the constellation diagram, these thresholds correspond to circles around each constellation point. An impulse will be recorded if a point is received outside all of these circles. Because an impulse is normally expected to be of short duration, it may be missed or reduced in size by the sampling process. It can be seen that the HP 4948A cannot guarantee to count all impulses of a given size appearing at its input. The impulse noise thresholds represent the level at which 50% of randomly spaced impulses will be counted.

To measure phase hits, the phase error is monitored. Any excursion of this error above the threshold set by the user and lasting longer than 4 ms will be counted as a phase hit. The HP 4948A allows threshold settings of 10, 15, or 20 degrees. These are sufficient to indicate if phase hits are likely to be causing data errors in the modem.

Gain hits are measured in two ways. First, a gain hit detector operates on the amplitude error in a similar manner to the phase hit detector, detecting gain hits exceeding the user-defined threshold for longer than 4 ms. The HP 4948A allows threshold settings of 1, 2, 3, or 6 dB. For QAM constellations, this may not detect every gain hit. A secondary gain hit detector monitors the average signal level for changes exceeding the gain hit thresholds. In this way, longer gain hits (greater than 200 ms) can be reliably detected.

The dropout detector also monitors signal level. If the signal level becomes reduced by more than 12 dB for longer than 4 ms, a dropout is recorded.

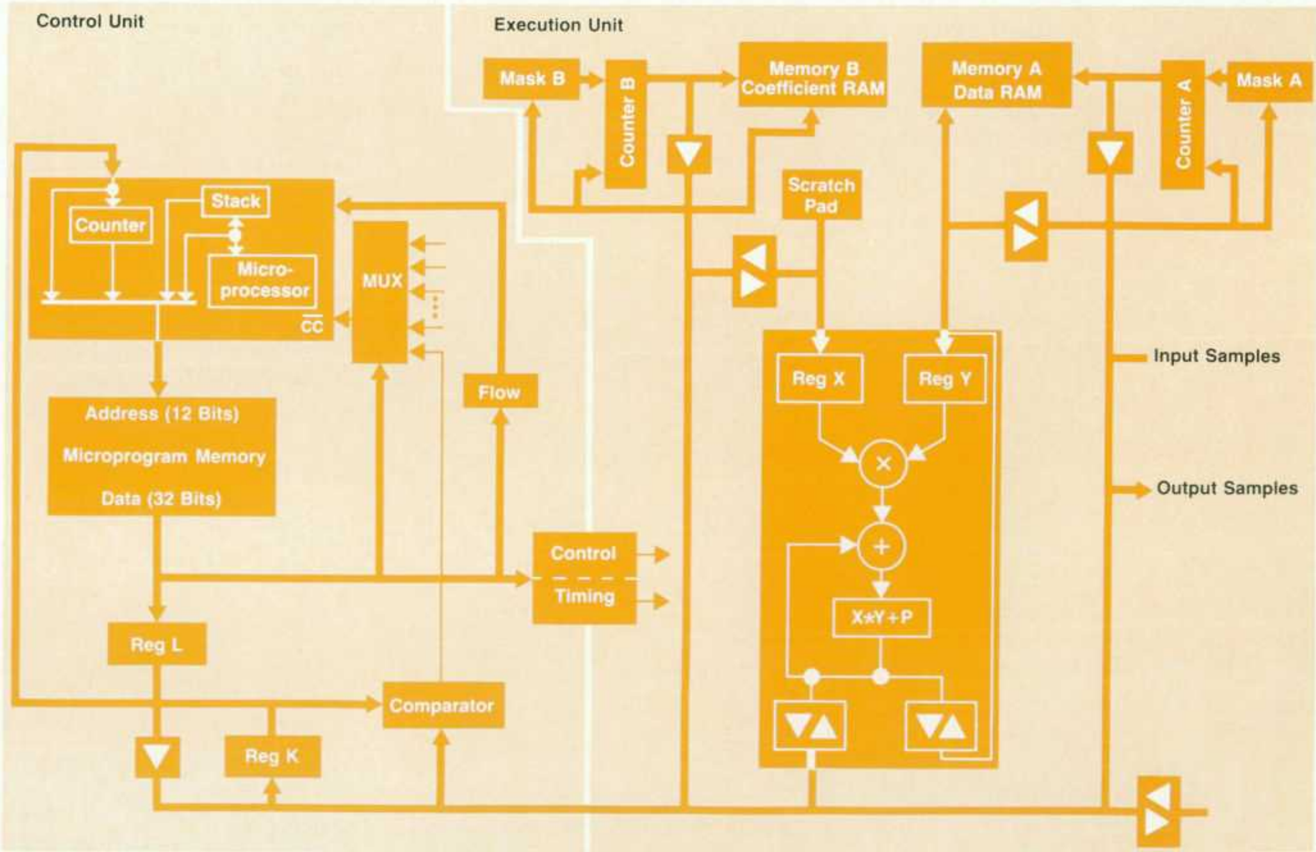


Fig. 17. *Signal processor architecture.*

LMS Algorithm for Equalizer Update

The most commonly used algorithm for adjusting the tap coefficients of an adaptive equalizer is the MSE (mean square error) or LMS (least mean squares) algorithm. As these names suggest, the algorithm operates to minimize the mean square error between the equalizer output y_k and its desired output d_k. The desired output will be one of n ideal constellation points produced by the slicing process.

$$\text{Mean square error} = E\,|\,d_k - y_k\,|$$

where E denotes the expected value.

The simplest form of equalizer filter is that of a linear transversal filter structure (see Fig. 1). The input samples to this filter must be provided at a minimum rate equal to the symbol rate. It can be shown that providing samples to the equalizer filter at twice the symbol rate yields an improvement in performance.[1] The equalizer output y_k is computed as follows:

$$y_k = \sum_{j=0}^{N} c_j x_{k-j}$$

where N is the total number of filter coefficients, c_j are the complex-valued filter coeffients ($j = 0,\ldots,N$), and x_j are the complex-valued equalizer input samples ($j = 0,\ldots,N$).

To update the equalizer, the error is computed at each symbol interval:

$$e_k = d_k - y_k$$

and used to modify the filter coefficients:

$$c_{k+1,j} = c_{k,j} + K_e e_k x^*_{k-j}, \qquad j = 0,\ldots,N$$

where * denotes the complex conjugate. K_e is the update constant, which is chosen as a compromise between stability and rate of convergence.

A further requirement to allow convergence is that the input signal x_j must be random.

See reference 2 for further information.

References

1. S.U.H. Qureshi and G.D. Forney, "Performance and properties of a T/2 equalizer," *National Telecommunications Conference*, 1977.
2. J.G. Proakis, *Digital Communications*, McGraw-Hill, Chapter 6.

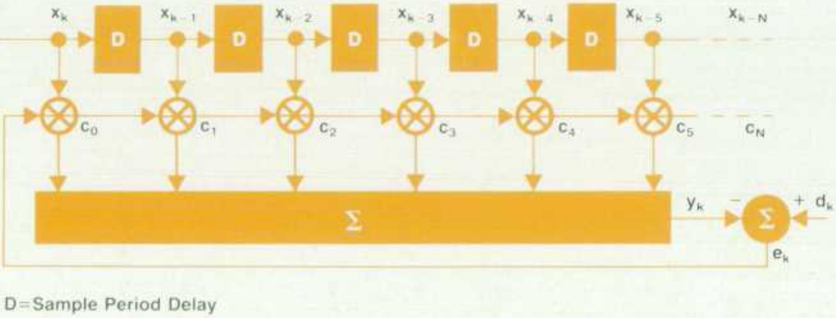


Fig. 1. *Adaptive equalizer structure.*

Architecture of the Real-Time Signal Processor

The HP 4948A makes measurements using sampled-data signal processing techniques. After some preliminary analog conditioning, the input signal is sampled and then digitized by a 12-bit analog-to-digital converter. The digital samples are passed to a specially designed signal processor which executes in real time the various algorithms required to produce the measurement results.

The signal processor consists of two parts: a control unit and an execution unit (see Fig. 17).

The control unit is microprogrammed, with the microinstructions stored in RAM. An instruction sequencer generates the memory address of the instruction to be carried out. Each microinstruction contains the signals to control both parts of the processor. Many microprograms are stored in ROM and the correct program is downloaded into the signal processor RAM whenever the instrument is run.

The processor execution unit is designed to operate as efficiently as possible in real time. It is optimized to perform digital convolution (the summation of products required when computing a digital transversal filter), since this function represents the largest requirement for real-time computation. The execution unit is designed around a high-speed 16-by-16-bit multiplier. This device is supplied with filter coefficients and input data samples simultaneously for fastest operation. The execution unit is split into two segments, the data memory A and the coefficient memory B. Memory A is connected directly to the Y input of the multiplier and memory B is connected to the X input of the multiplier via bidirectional bus drivers. This arrangement allows the accumulate and add function (X*Y+P) required for digital convolution to be performed in one instruction while also pipelining the next two factors.

Counters A and B are designed to have the capability to cycle on a variable length of memory, which minimizes the overhead required in data storage and access. The cycle depth of each counter is controlled by a mask register. Both counters may also be set to perform a double increment. This is useful when filter coefficients are stored in double precision (32 bits) but only the most significant 16 bits are used in computing the filter output. Also, the least-significant bit of counter A can be reversed to simplify the multiplication of two complex numbers whose real and imaginary parts are stored in adjacent memory locations.

32 scratchpad registers are used to hold parameters such as data pointers and constant values that are common to the activities of several subroutines, and to provide a working storage area.

An interface between the control unit and the execution unit is provided by registers K and L. Register L can be loaded with values from the microprogram, and register K can be loaded from register L or from the execution unit. Both registers can be compared with the execution unit data bus to determine sequencer branch instructions. By using register K, branching can be made dependent upon results obtained from the execution unit.

The input sampling clock is chosen to be a multiple of the symbol rate and so n input samples are required to process one received symbol. This sampling clock is adjusted under signal processor control to synchronize with the transmitted symbol rate. A software first-in, first-out (FIFO) structure is implemented. At convenient intervals during the program, a check is made to see if a new sample is available or a new transmit sample is required. Whenever sufficient samples to process a received symbol are not available, the program waits until these samples are received. In this way the microprogram repetition period is synchronized with the symbol interval.

Use of this FIFO structure allows the program cycle to exceed the symbol period on occasion without loss of data. This allows lengthy processes, for example FFT processing, to be performed on an occasional basis.

Signal Processor Development Tools

Since the HP 64000 Microprocessor Development System (MDS) did not support custom processors at the time, we had to develop the tools we needed. We identified four tools: a program assembler, a data assembler, a linker/ROM generator, and a program development utility. This program development utility we called the Software Development System (SDS).

SDS has two parts, a user interface and a set of execution routines which interact with the signal processor.

The user interface is a softkey-driven program modeled on the HP 64000 MDS. It is written in Pascal and runs on an HP 9000 Series 200 or 300 Computer. The execution routines are written in Z80 assembly language and are resident in the HP 4948A.

The SDS execution routines and the normal operational firmware of the HP 4948A share routines to configure and communicate with the signal processor. This simplifies the transfer of developed signal processor programs to firmware ROMs.

The major facilities provided by SDS are:

1) Load program and data from ROM in the HP 4948A or from an HP 1000 Computer via a serial link. The program and data assemblers are on the HP 1000.
2) List program and data memory.
3) Edit data memory.
4) Start and stop the signal processor, with or without interaction with the HP 4948A measurement firmware.
5) Set a breakpoint and display the signal processor status when the breakpoint is encountered. Fig. 18 shows a typical breakpoint screen display.
6) Select the point in the signal processing from which data is passed to the X and Y outputs on the HP 4948A rear panel. This allows an engineer to examine, for example, filter coefficient settling or raw FFT results. In normal operation the point is fixed to provide the constellation display.

It is worth noting that the breakpoint facility, the ability to display data while the signal processor is running, and the XY output display are the only facilities provided for debugging the real-time software. Without special hardware, other facilities would require the signal processor to be operated at less than full rate, which is not acceptable because the programs are time dependent.

```
BREAKPOINT: 0100 occurs 1 times ______________________________ HP4848A/SDS  3.1

register R: 0000    stack: 0024          register K: 0FCF
                           ****
                           ****          register P: DE8D 3440
                           ****
                           ****
                           ****
                           ****
                           ****          counter A: 0520
                           ****          counter B: 1525

SCRATCHPAD: _________________________________________________________

  0000: 0000 7FFF 8000 006C 0020 9000 0000 1000
  0008: 833C 5305 1F03 2B23 458D 4000 0117 06CE
  0010: 0FCF 0FCC 0300 0EF0 0D80 96C0 0D80 44EC
  0018: 09CE 094C C000 C313 1C9D 0016 04AA 1D24

STATUS: Breakpoint found ____________________________________________



__dump__  __load__  _modify_  __list__  _start__  __stop__  _break__  ___xy___
```

Fig. 18. *Breakpoint screen display of the software development system used in the HP 4948A project.*

Digital Phase-Locked Loops

A digital phase-locked loop consists of the same basic blocks as an analog phase-locked loop, but is implemented using data samples. These basic blocks are a phase detector, a loop filter, and an oscillator.

These three blocks in a modem receiver digital phase-locked loop are shown in Fig 1.

The phase detector consists of a rotation algorithm followed by the phase error calculation. The rotation algorithm rotates the equalizer complex output (x_i, x_q) by an angle θ determined by the digital phase-locked loop output.

$$y_i = x_i\cos\theta + x_q\sin\theta$$
$$y_q = x_q\cos\theta - x_i\sin\theta$$

The phase error is then computed as the phase difference between the rotator output (y_i, y_q) and the sliced point.

A second-order digital loop filter is shown. It consists of two update constants K_1 and K_2, chosen to give the desired loop bandwidth and damping factor:[1]

$$K_1 = 1 - \alpha^2 - \beta^2$$
$$K_2 = 1 + \alpha^2 - 2\alpha + \beta^2$$

where

$$\alpha = \cos(\omega_n T\sqrt{1-c^2})\exp(-c\omega_n T)$$
$$\beta = \sin(\omega_n T\sqrt{1-c^2})\exp(-c\omega_n T)$$

and ω_n is the loop natural frequency, c is a damping factor, and T is the sampling period.

If a frequency offset exists in the equalizer output, the second-order point of the digital phase-locked loop assumes a dc value proportional to this frequency offset.

The oscillator consists of a digital integrator followed by a sin/cos conversion. A one-sample delay is added to reflect the delay around the loop (equivalent to one symbol interval).

The frequency response Y/X of the digital phase-locked loop is as follows:

$$Y/X = 1 - \frac{(1-z^{-1})^2}{1 + z^{-1}(K_1 - K_2 - 2) + z^{-2}(1-K_1)}$$

This is a high-pass characteristic. Therefore, frequency offset and low-frequency phase jitter are removed from the signal at Y. The loop is most effective at removing phase jitter whenever a high loop bandwidth is chosen. However, it can be shown that the noise bandwidth of the digital phase-locked loop increases with loop bandwidth. Since it is not desirable to inject extra noise at this point, the choice of a loop bandwidth is a compromise between phase jitter removal and noise minimization.

Reference

1. W.C. Lindsey and Chak Ming Chie, "A Survey of Digital Phase-Locked Loops," *Proceedings of the IEEE*, Vol. 69, no. 4, April 1981, pp. 410-431.

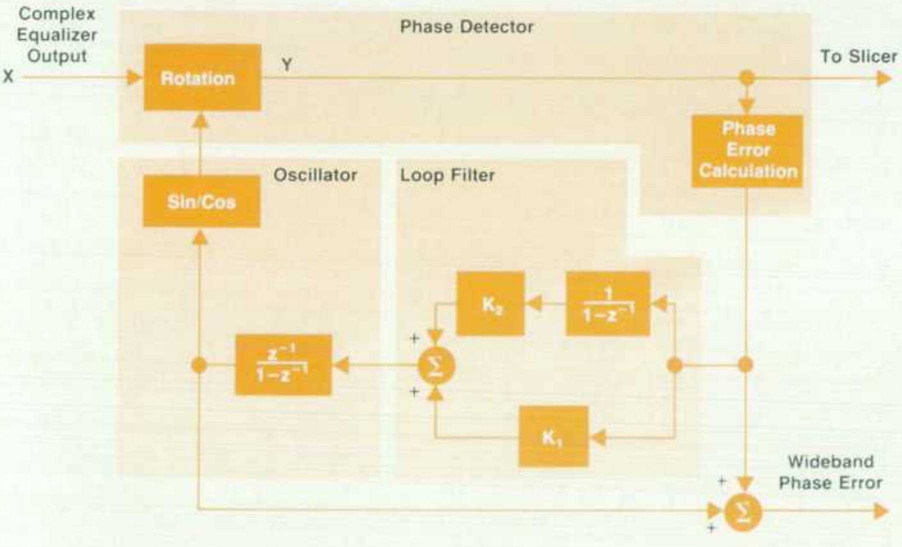


Fig. 1. *Digital phase-locked loop.*

Acknowledgments

We would like to thank Ian Buckner, Jacqui Dixon, Sam McMillan, Bill Savage, John Struthers, and Howard Tang, who were valuable members of the R&D team, and Gerald Dobbie, who in addition acted as project manager during the final stages of development. Special thanks are also due Allan Sturgeon and Mark Dykes in product marketing for their work in helping to define the product.

An Infrared Link for Low-Cost Calculators and Printers

by Steven L. Harper, Robert S. Worsley, and Bruce A. Stephens

MANY OF HP'S HANDHELD CALCULATOR customers have told us of their need to create a permanent record of their calculations easily. The need seems to be particularly acute for users of financial calculators. A banker or real-estate agent often wants to give a client a printed record of loan information or perhaps an amortization table showing interest paid and remaining balance. Because of this customer feedback, printing capability was high on the list of design priorities for the HP-18C Business Consultant.[1]

There are two approaches to providing this feature. One method is to design another model of the calculator with a built-in printer. The earlier HP-91 and HP-97 Calculators[2] are examples of this approach. Although the portability of the unit suffers somewhat, this is a good solution for the customer who uses the printer a lot and knows of the need for this capability before buying the calculator. It is less ideal for the person who really wants full portability or didn't realize the need for a printing capability until after the calculator was purchased. For this individual the other approach, that of designing a separate accessory printer which connects to the calculator in some way, is the better solution. This approach also yields somewhat lower development cost and a shorter design schedule. The HP-41C Calculator uses this method,[3,4] and the decision was made to design an accessory printer for the HP-18C also.

Interconnection Method

One very critical design area for such a printer is the means of interconnection to the calculator. The HP-IL interface[5] is used in the HP-41C for connection to a number of peripherals, including an accessory printer. This would have provided much more capability than was needed for a simple printer-only interface, and would have cost more. In addition, we had received a number of complaints from customers about the inconvenience of the cables for the interface and battery recharger with our portable products. With this in mind, we began to investigate the possibility of a wireless interface for a printer powered by disposable batteries.

Infrared transmission seemed to be the only wireless technology that allowed the use of low-cost, low-power, and readily available components. Infrared remote-control units for television sets and videocassette recorders have been in use for years. Their transmitting element usually consists of one or more infrared light-emitting diodes driven so as to produce short bursts, each containing a few pulses of invisible infrared light at a wavelength of 940 nm. The pulses within each burst have a repetition frequency of about 40 kHz and their intensity is proportional to the current through the diode(s). Information is encoded by varying the time between bursts.

The receiver uses a photodiode which acts as a weak current source with an amplitude proportional to the incident light intensity. A sensitive gain-controlled preamplifier provides frequency selectivity, demodulation, and conversion of the photodiode current to logic voltage levels. Since the HP-18C Calculator already has a 32.768-kHz oscillator for its time and alarm functions, our infrared link uses 32.768 kHz as the infrared modulation frequency, rather than the slightly higher value used by conventional remote-control units.

Silicon photodiodes respond to a broad range of wavelengths, including most of the visible spectrum. To reduce unwanted signals, the photodiode in the receiver is often encapsulated in a material that is opaque to visible light, but transparent to the longer infrared wavelengths. Various kinds of optical filter materials are also used as windows in front of the photodiode to reduce extraneous light further by moving the cutoff wavelength still closer to 940 nm. The infrared receiver in the printer for the HP-18C uses both of these techniques. However, incandescent lights and sunlight still have strong components in the infrared range. The modulation of the bursts allows some frequency selectivity that reduces this dc and low-frequency optical interference still further to provide a higher signal-to-noise ratio and an improvement in the distance over which the link will operate reliably.

Implementing the Link

The cost of adding the infrared transmitting circuitry in the calculator is minimal. This is especially important since it allows us to include that part of the link in every calculator without penalizing the customer who does not want the printing capability. Otherwise it would be necessary to resort to a more expensive and complex plug-in module system, which would likely add as much or more cost to the calculator, even without the module.

Unfortunately, the receiver end of the link is substantially more expensive. The infrared preamplifier chip requires several discrete components and a regulated supply. The additional printed circuit board area needed was much more than could be accommodated in the calculator. Because of this, it was necessary to make the infrared link a one-way-only interface—the calculator is the transmitter and the printer is the receiver.

While cost and size considerations made this decision obvious, it nevertheless represented a trade-off in link performance. The printer cannot send handshake signals to the calculator, indicating readiness to receive more data. Because of this, after enough characters are transmitted to fill the printer's buffer, the calculator must carefully pace subsequent transmissions so that even the slowest printer can keep up. When the printer has fresh batteries, it can print at a rate of slightly more than one line per second. When the batteries are near the end of their useful life, the rate is reduced to just over one half line per second. It would have been nice to be able to run the printer from a regulated supply rather than directly from the batteries so that the speed would be more constant. Unfortunately, the current requirements are quite high (up to 1.5A average, 3A peak while printing) and the inclusion of such a supply would have escalated the cost of the printer far beyond its design objectives. In any case, the user will seldom notice this trade-off since most printing with the calculator is done in segments smaller than the size of the 200-character print buffer. Under these conditions, the printer performs as fast as it is able, given the current state of the batteries. For longer print segments, the first eight lines or so print as fast as the printer can go, and then the system slows to a rate of slightly more than one half line per second.

One of the most important specifications for any interface is the maximum data rate. For this infrared link, the rate is slightly less than 80 characters per second. Other important parameters are distance and directional sensitivity. Commercial infrared remote-control units typically have a range of thirty feet or more. This is neither necessary nor desirable for calculator-to-printer communication. The infrared link obeys a square-law response with respect to distance, that is, doubling the range requires four times the transmit drive current for a given receiver sensitivity. This quickly becomes a problem for the calculator's power supply. In addition, a long-range capability in this application creates a potential problem where one individual's calculator might interfere with someone else's printer nearby. The minimum distance is about 18 inches to allow comfortable desktop operation, but the maximum range is no more than a few feet to avoid such problems.

The infrared light-emitting diode in the calculator has an integral molded plastic lens which forms a somewhat directional radiation pattern. At an angle of slightly less than thirty degrees from the directional axis of the pattern, the intensity falls to half its maximum value along the axis. The photodiode in the printer has no lens and its sensitivity is much less directional. Its response is proportional to the perpendicular area facing the source and is therefore a cosine function. At sixty degrees the response will be half the maximum. If the calculator's batteries are fresh, and the calculator and printer are lined up facing each other, the range is about four feet. On the other hand, if the batteries are about ready for replacement and the calculator faces thirty degrees away from the printer, the maximum distance over which the link will operate is about 18 inches. Naturally, obstructions in the path and reflections can significantly change these values.

How the Interface Works

Fig. 1 diagrams the circuitry used for the infrared link. A combination of hardware and firmware in the calculator generates the proper gated pulse waveform to drive the infrared light-emitting diode. A resistor connects the microprocessor port to the base of a bipolar transistor that acts as the driver device to handle the relatively high current peaks required. In the HP-18C, the series resistor limits the light-emitting diode current to about 80 to 160 milliamperes, depending on the state of the batteries. While these currents are fairly high, it may be possible in the future to integrate the transistor driver on the microprocessor chip, and thus reduce the size and cost of the transmit end of the link in the calculator even further.

The transmitted infrared signal is converted into a current by the photodiode in the printer, and then into a very small differential voltage by the pull-up and pull-down resistors. Two coupling capacitors feed this signal into the preamplifier IC. In the preamplifier are two externally compensated gain stages with automatic gain control. The pulse bursts then pass through a tuned synchronous demodulator stage and an output integrator and pulse shaper before

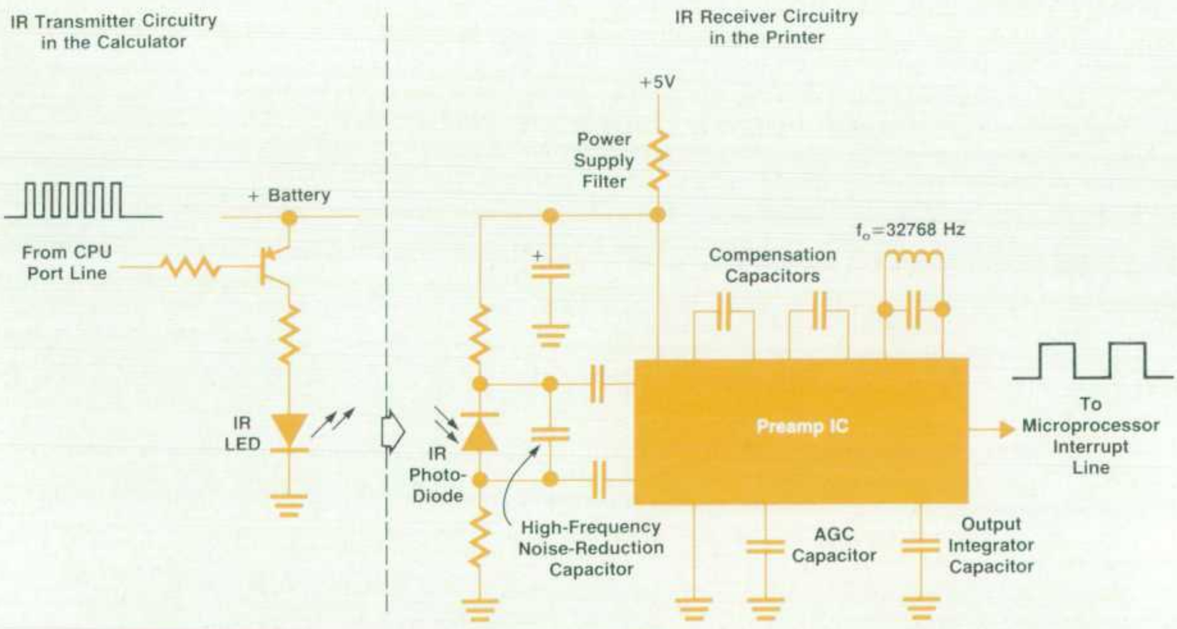


Fig. 1. *Infrared link schematic diagram.*

reaching the logic signal levels required to drive the interrupt line of the printer's microprocessor.

Electrical interference is a special problem for the extremely high-gain preamplifier. Careful printed circuit trace routing for power supply and ground lines and power supply filtering are essential. The typical solution to this problem is shielding, but this is particularly difficult for a small portable product with a nonmetallic case and no connection to earth ground. Several circuit configurations were tried before an acceptable solution was found. A capacitor is added in parallel with the photodiode, its value carefully chosen such that high-frequency noise is shorted to ground and prevented from getting into the first stage of the preamplifier without attenuating the 32.768-kHz signal frequency appreciably. This increases the link range substantially.

The calculator sends bursts of infrared light that are modulated with a 32.768-kHz square wave to produce 6 to 8 pulses of light in each burst. Fig. 2 shows two typical bursts. Each 32.768-kHz cycle consists of a 15.26-μs pulse of light followed by a 15.26-μs pause.

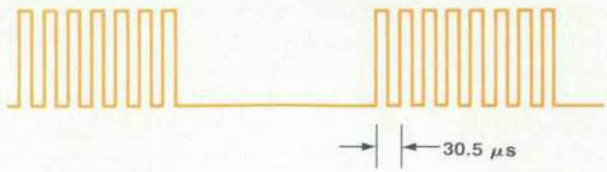


Fig. 2. *Data is transmitted as a series of bursts of infrared light, each burst formed by six to eight pulses of energy at a pulse repetition rate of 32.768 kHz.*

Coding

The data is encoded in bit times which are subdivided into half-bit times. A bit time is defined as 28 periods of a 32.768-kHz waveform (approximately 854.5 μs). Time intervals are measured from the leading edge of the bursts. There are three kinds of bits (see Fig. 3 for an example):

- One bit. A one bit is defined as a burst at the beginning of the first half-bit time with no burst in the second half-bit time.
- Zero bit. A zero bit is defined as no burst in the first half-bit time and a burst at the beginning of the second half-bit time.
- Start bit. A start bit is defined as a burst at the beginning of three consecutive half bit times, an otherwise illegal sequence. Start-bit bursts can have six to nine 32.768-kHz pulses of infrared light.

Each frame consists of a start bit followed by 12 data bits. The first four data bits are the error correction bits and the remaining eight bits are the byte being transmitted. There must be a delay of at least three half-bit times between frames, measured from the end of the last bit time of a frame to the leading edge of the start bit of the next frame. This gives a maximum data rate of about 78 bytes/s. An example of a complete frame is shown in Fig. 3.

Error Correction

Two kinds of errors are expected to be the most likely: bits that are missed entirely and noise bursts introduced in addition to the correct data bursts. The bit decoding routine treats bit times with extra bursts (noise) as missed bits since it does not know which burst was data and which burst was noise. Therefore, the error correction code only has to correct one kind of error—missed bits. Flipped bits ($1 \rightsquigarrow 0$ or $0 \rightsquigarrow 1$) are much less likely since this requires a noise burst to occur in the opposite half-bit time of a missed burst, so these errors are not corrected.

Each error correction bit encodes the parity of a subset of the data bits, allowing correction of up to two missed bits by checking the parity of separate sets of bits. The correction bits (H1 to H4) are set as the even parity of the data byte ANDed with the following masks:

Bit	Mask
H1	01111000
H2	11100110
H3	11010101
H4	10001011

Unidirectional Communication

The unidirectional infrared interface permits the calculator to talk to the printer, but the printer cannot communicate back. This eliminates the need for any receiver circuitry in the calculator. Conversely, the printer is saved the circuitry needed to transmit back to the calculator.

As a result of the unidirectional nature of the communication, the calculator has no direct information on the status (or even the existence) of the printer. It can merely transmit bursts of infrared light in such a manner that the printer can handle them. The most critical piece of information that is missing is whether the printer's buffer is full. The printer has a 200-character buffer. The calculator must make sure that this buffer never overflows. The printer is capable of emptying the buffer at a rate of so many lines per second, but the number of lines in the buffer depends on how many characters there are per line.

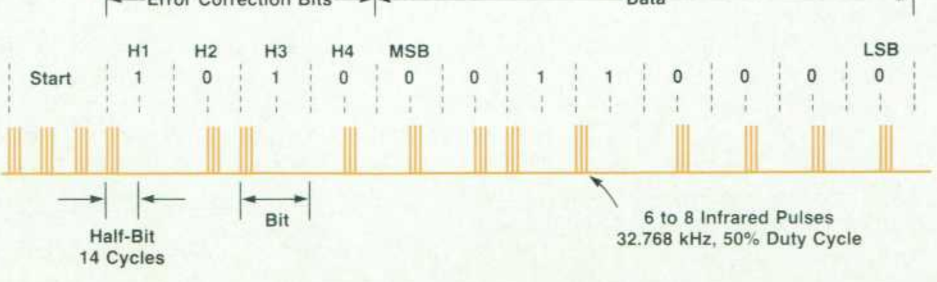


Fig. 3. *Example of infrared message frame.*

A simple solution would be simply to wait long enough following each transmission of a line to be sure that the printer has finished. This doesn't take much advantage of the printer's buffer and causes the calculator to remain inactive for prolonged periods of time. What we would really like is to have the calculator strive to keep the printer's buffer full. After a long pause without printing the printer's buffer is certain to be empty, so several lines can be transmitted immediately and the printer should be able to buffer them and print them as fast as possible. Only when the calculator has sent enough lines to fill the printer's buffer should the calculator have to wait to transmit more data.

A truly perfect algorithm would have the calculator keep track of when each of the lines was transmitted and how many bytes were in each line. However, this requires too much memory. Instead, some simplifying assumptions are made. All lines are assumed to contain 25 bytes (24 characters plus an end-of-line byte), which means that the printer's buffer will hold up to eight lines. Rather than keeping track of when each line was sent, a simple line counter and a record of the time when the last line was sent are used. The counter is incremented each time a line is sent and the time of that transmission is saved. Before sending a line, the calculator checks how much time has elapsed since the last line was transmitted. It then calculates how many lines the printer should have printed during that time and subtracts that number from the line count. If the count indicates that the buffer can now hold another full line, the calculator sends the line and saves the current time as the last transmission time and updates the line count to include this newly transmitted line. If the buffer cannot hold another line yet, the calculator waits and repeats the above process until the buffer empties enough to accept another line.

Assuming that the printer can print at the specified rate, this algorithm is foolproof. Since not all printers print at the same rate and they tend to slow down as their batteries deteriorate, the calculator must never send lines faster than the slowest printer can print while running at the lowest acceptable battery level. Hence, since most printers empty their buffers more quickly than the calculator is allowed to transmit, they will often pause while the calculator is waiting to send the next line.

Critical Timing in the Link

Another challenge in the printer transmission firmware was the timing of the bursts of infrared energy needed to transmit a character. As the development of the printer progressed along with the calculator, the need for accuracy of the burst timing became quite apparent. Each character is transmitted as a series of bursts of infrared energy modulated at 32.768 kHz. Each transmitted byte (frame) is divided into 27 subparts which include the start bits, redundantly encoded data, and error correction information. This scheme is designed to maximize the printer's ability to recover garbled frames.

The most common problem with infrared transmission is dropout of the infrared signal. Some of the transmitted bursts may not be detected by the printer if the calculator and printer are at the limits of their range or if something momentarily blocks the transmission path between them. The encoding scheme used works well for recovering from lost bursts, but only if the bursts are timed accurately enough so that the printer does not get completely out of synchronization with the calculator. This would cause the rest of the bits in the frame to be incorrect.

To achieve the required timing accuracy, the calculator's microprocessor needs to start each infrared burst within a 4-μs window. Since the nominal clock rate is 617 kHz, the microprocessor must not be more than one cycle away from the perfect time. The clock is generated by an LC oscillator whose frequency varies slightly from unit to unit and with changes in temperature and battery voltage. The calculator also contains a 32.768-kHz crystal oscillator that the firmware can use to calibrate the loops used to time infrared bursts. The exact number of processor cycles required between bursts is calculated, and by using variable cycle count instructions, this exact number of cycles is achieved. This calculation is performed before each line is sent to guarantee that the clock has not drifted significantly. The flowchart in Fig. 4 shows what is done.

The printer receiver decodes the incoming infrared bursts by measuring the time intervals between successive bursts. This time-interval measurement is subject to four sources of error:

1. After being detected by the photodiode, the incoming bursts go through a preamplifier, which introduces additional timing error. The input to the preamplifier consists of a series of 15.26-μs-wide pulses of current separated by pauses. The preamplifier may not respond to the bursts of pulses in exactly the same way every time. Its output might go true on the nth pulse in one burst and on the (n+1)th pulse in the next burst, which would make the interval between bursts at the preamplifier output look 30.5 μs too long. Similarly, the output could go true on the (n+1)th pulse in one burst and on the nth pulse in the next burst, which would make the interval look 30.5 μs too short. Therefore the error from the preamplifier is approximately $\pm$30.5 μs.
2. The output of the preamplifier goes to the interrupt pin of the printer's microprocessor. Since there are places in the printer's firmware where the interrupt must be temporarily disabled, the interrupt can be delayed up to 13 processor cycles (about 34 μs). Depending on whether this happens on the first or last pulse of an interval, the interval can look longer or shorter by 13 cycles.
3. Once the interrupt occurs, the interval between interrupts is measured using a timer in the CPU. Since the timer ticks only every 32 cycles (about 84 μs), additional error is introduced by this granularity. For example, an interval that is really 11.3 timer ticks long will be measured as either 11 or 12 ticks long depending on when it occurs relative to the timer ticks.
4. The final error comes from the printer's oscillator speed variations. Since an LC oscillator is used, the frequency varies somewhat from printer to printer. Therefore, the actual time interval represented by the measured number of timer ticks varies with processor timing. A fast processor will show more timer ticks for a given interval than a slow processor.

The receiver code uses the time interval since the last interrupt as an index into a fixed timing table to determine whether the current interrupt is for a one bit or a zero bit. The time interval must be adjusted by five ticks if the last bit was a zero since the interrupt for a zero occurs at the midpoint of the bit time rather than at its beginning. The ability to correct up to two missed bits means that the table must be seven half-bit times long to cover the bit sequence 1XX0 where X indicates a missed bit. Table I shows the range of timer ticks calculated for each bit-to-bit interval, taking into account the above errors as well as the number of ticks actually used in the timing table.

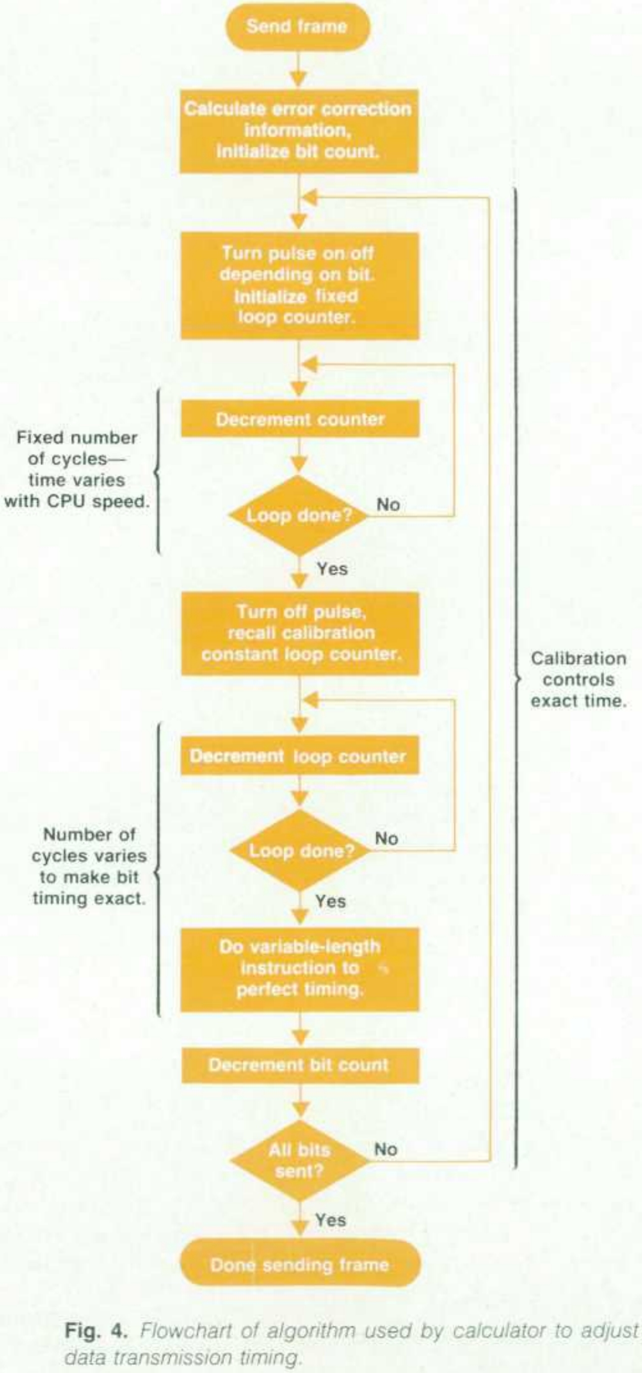


Fig. 4. *Flowchart of algorithm used by calculator to adjust data transmission timing.*

Table I
Timing Table

Processor Ticks	Table Ticks	Bit Received	Comments
0-2			Too short, ignore it
3-7	3-7	X	Two pulses in same bit time, treat as missed bit
8-12	8-12	1	
13-17	13-17	0	
18-23	18-22	X-1	Missed bit followed by a one bit
23-28	23-28	X-0	Missed bit followed by a zero bit
28-34	29-33	X-X-1	Two missed bits, then a one bit
33-39	34-39	X-X-0	Two missed bits, then a zero bit

The burst timing resyncs on each received burst since the time of each interrupt is saved as the starting point of the time interval between it and the next interrupt. Start-bit timing is handled as a special case. The start bit must be received correctly before the code looks for the 12 data bits. After the frame is received, the byte is checked using the error correction bits and corrected if necessary.

Looking at an earlier version of the calculated timing table, it became apparent that the calculator must transmit the infrared bursts very accurately to get reliable decoding of received frames since the intervals overlap when bits are missed. For example, if 34 processor ticks occur between bursts, the cause is two missed bits followed by either a one or zero bit, but the printer would interpret it as the latter. Table I includes the maximum allowed error in the calculator transmission as well as the printer errors. The overlap of intervals leaves a possibility of improper bit decoding, but a statistical simulation showed that the probability of this occurring is very low.

Acknowledgments

A number of people deserve special mention for the contributions to the infrared printer project. Dave Rabinowitz was the project manager. Mechanical design was handled by Dave Smith and Jack Muranami, who explain some of the mechanical design decisions in the article on page 21. Gary Podwalny did the industrial design. Theresa Gibney's efforts in manufacturing were especially important. Grant Salmonson did the production test tooling. With the help of Ng Say Ban, David Shum, and Tan Zing Chiou at Singapore, a difficult transfer to production went very smoothly. Herbert Ting provided strong support from the QA department.

References

1. S.L. Wechsler, "A Handheld Business Consultant," *Hewlett-Packard Journal*, Vol. 38, no. 8, August 1987.
2. B.E. Musch and R.B. Taggart, "Portable Scientific Calculator Has Built-In Printer," *Hewlett-Packard Journal*, Vol. 28, no. 3, November 1976.
3. B.E. Musch, "Powerful Personal Calculator System Sets New Standards," *Hewlett-Packard Journal*, Vol. 31, no. 3, March 1980.

4. R.D. Quick and D.L. Morris, "Evolutionary Printer Provides Significantly Better Performance," *ibid.*
5. R.D. Quick and S.L. Harper, "HP-IL: A Low-Cost Digital Interface for Portable Applications," *Hewlett-Packard Journal*, Vol. 34, no. 1, January 1983.

A Low-Cost Wireless Portable Printer

Based on a unidirectional infrared transmission path, this small thermal printer can provide hard copy of HP-18C and HP-28C calculations.

by David L. Smith and Masahiko Muranami

THE HP 82240A INFRARED PRINTER (Fig. 1) is a portable battery-powered thermal printer capable of printing a maximum of 24 columns of alphanumeric characters or 166 columns of continuous graphics per line. Designed for use with an HP-18C or HP-28C handheld calculator,[1,2] the information to be printed is transmitted to the printer by the calculator using an infrared beam. This transmission method is discussed in detail in the article on page 16.

The printer uses HP's standard 58-mm-wide black-printing thermal paper. The 2-inch-diameter, 80-foot-long roll will provide about 6000 lines of print. User controls include power, print intensity, and paper advance switches. The HP Roman8 character set is provided.

Power is supplied by four commercially available AA-size batteries and can be supplemented by an ac adapter with a common barrel-shaped plug. The unit can accept adapters with ac or dc output. With full power the printer is capable of printing 0.8 line per second. One set of fresh batteries will print up to one roll.

The HP 82240A measures approximately 7.25 inches long by 3.5 inches wide by 2.5 inches tall. It weighs about

Fig. 1. *The HP 82240A Infrared Printer is a battery-operated printer designed for use with the HP-18C and HP-28C Calculators. The need for connecting cables is eliminated by using an infrared beam for data transmission.*

one pound when loaded with a full paper roll and batteries. A manual, paper roll, and batteries are included with the printer.

Product Design

The components of the HP 82240A can be divided into several general categories (see Fig. 2): printer mechanism, printed circuit assembly, battery contacts, electrostatic discharge protection, and plastic parts.

The development time for the HP 82240A was rather short because of its scheduled announcement along with the HP-18C. One factor that enabled quick development was the decision to purchase an OEM printer mechanism. The mechanism was chosen for compactness, quiet operation, and graphics printing capability. Its cost accounts for one third of the total part cost of the printer. Testing was conducted to ensure acceptable life, environmental, and drop-survival performance. Based on qualification test results, the manufacturer agreed to modify the mechanism to meet our drop test and package drop test requirements.

Other parts such as switches, CPU, and interconnects were chosen for low cost and savings in development time.

Two battery springs were designed to connect the batteries to the printed circuit board. Nickel-plated beryllium copper was chosen for high strength and corrosion resistance. The 3.3-by-4.5-inch single-sided printed circuit board is a departure from recent double-sided and/or hybrid board technology used in HP's handheld products. Lower cost was the main reason for this decision.

Attention was paid to efficiency and manufacturability throughout the design process. A CAD/CAM system was used from initial layout to drawing generation. Components of the printer such as plastic parts, metal parts, and the printed circuit board were designed as wireframe models. To ensure proper interaction among the parts, purchased components such as the paper roll, batteries, printer mechanism, switches, photodiode, and ac socket were also recreated in the data bases. Where necessary, these models could be manipulated to determine feature locations and to check for fit and interferences. The rotating motion of the paper door, for example, was simulated to determine its pin location in the bottom case and its snap detail in the top case (see Fig. 3). Application of CAD/CAM methods to the design process greatly increased confidence that all components would perform together as intended.

Early in the design process, manufacturing engineering advised that parts allowing layered assembly were more desirable. An effort was made to design plastic parts that held components in place during assembly, thus eliminating as many two-handed operations as possible. The solution chosen uses snap fits and slip fits throughout the product and allows assembly from the bottom case up. The battery shorting bars and contacts snap into the bottom case to allow manipulation of the assembly without dislocating them. The ESD protection components and printed circuit assembly are located by screw bosses. The keycaps slip onto the switch actuators and the printer mechanism is secured. Two screws locate the mechanism within a small tolerance band to eliminate the possibility of paper jams. The ac adapter receptacle and infrared window snap into the bottom case, after which the top case is slipped over the whole assembly. Six screws hold the cases together to ensure acceptable drop performance. The cosmetically sensitive paper door and paper tear-off window are snapped into place at the end of the assembly process.

Custom tooling for plastic parts was another area in

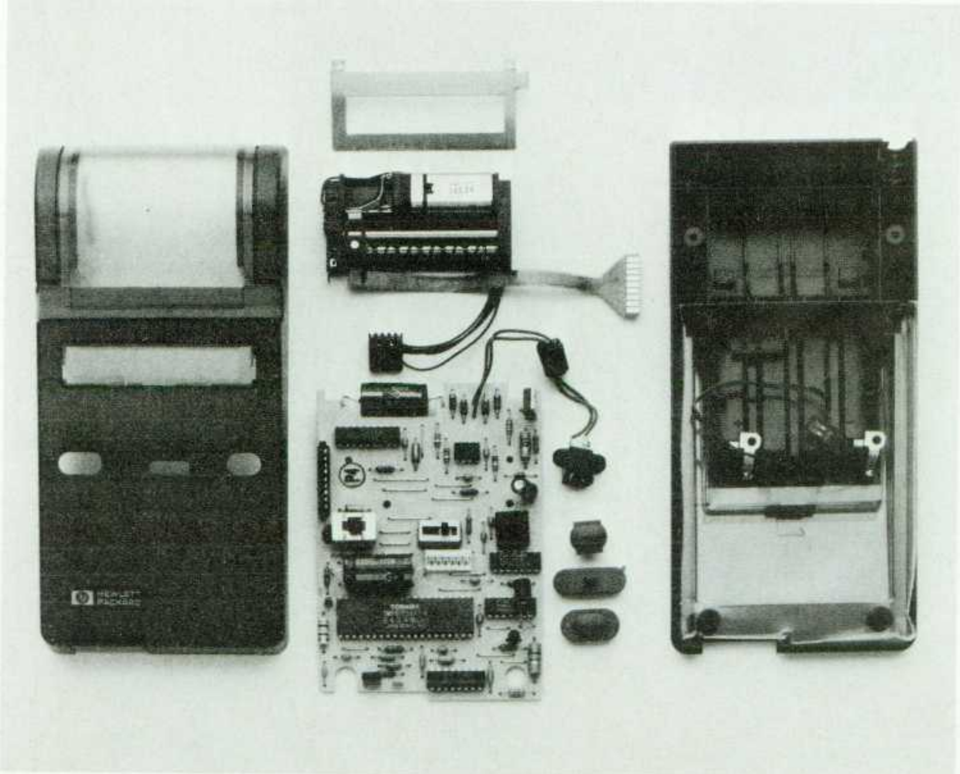


Fig. 2. *Components used in HP 82240A.*

which CAD/CAM played a major role. The plastic injection molds for the top and bottom case were built entirely using computer-aided-manufacturing techniques. The wireframe data base from R&D was transferred to a similar system in manufacturing. The mold designers used the part data base to create the mold data base. No part drawings were necessary, since the data base contained the required tooling information. Waterline locations, inserts, ejector plates and all electrode designs were completed on the system. The complex shape of the case parts would have made conventional calculations time consuming and difficult, but the system made number-checking trivial. Details such as the external radii along the intersection of a curved surface and a drafted plane were programmed and cut on a CNC (computerized numerical control) milling machine with great accuracy. With conventional techniques, milling such details exactly as the print specifies is nearly impossible and very time consuming. Complete tooling for the top and bottom case was manufactured in seven weeks, compared to quotes of nine to twelve weeks from outside vendors without CAD/CAM capability. Design modifications were also conveyed to the mold designers through the part data base. The hard-copy documentation for the top and bottom case was done after the molds were completed for production parts.

Because we lacked sufficient experience with single-sided printed circuit boards, manufacturing engineering gave considerable attention to the solderability and testing of the board. Design guidelines were first obtained from Roseville Terminal Division. The pad and trace design incorporates features designed for optimum solderability and strength in the finished product. Solder defect data was collected by conducting wave soldering experiments at Vancouver Division, from which design modifications were determined.

During development, the decision was made to transfer production to Singapore Manufacturing Division. A coordinated effort for a smooth transition was accomplished in several stages. First, assembly tooling was designed and built in Corvallis with the aid of Singapore engineers. Parts were sourced domestically, whether custom or commercially available. Lab prototypes were assembled in Corvallis. The results gave information upon which design improvements, tooling debug, and process refinement were based. Complete QA testing was conducted in Corvallis with a Singapore engineer, while some of the tests were duplicated in Singapore. Singapore then began procuring parts locally and initiated tooling for custom parts except plastic molds. Tooling and custom parts were shipped to Singapore for the production prototype build, for which Corvallis engineers were at hand. Finally, for production, the plastic molds were sent to Singapore.

Acknowledgments

Theresa Gibney conducted solderability experiments, coordinated assembly tooling, and designed the line both in Corvallis and Singapore. She also acted as liaison between the two sites. Marc Baldwin coordinated plastic tooling, texturing, and tool transfer. Other members of John Mitchell's manufacturing team contributed to assembly tooling. Gary Watts and Bill Peters designed the molds for the top case and bottom case. Burl Smith programmed the CNC milling machine. Singapore engineers David Shum and Tan Zing Chiou came to Corvallis to aid assembly tooling, line design, and procurement. Herbert Ting came to conduct product qualification. Other members of Ng Say Ban's manufacturing team contributed to the smooth transfer of production to Singapore. Procurement engineers both in Corvallis and Singapore were essential to the success of protoype builds.

References

1. S.L. Wechsler, "A Handheld Business Consultant," *Hewlett-Packard Journal*, Vol. 38, no. 8, August 1987.
2. W.C. Wickes, "An Evolutionary RPN Calculator for Technical Professionals," *ibid.*

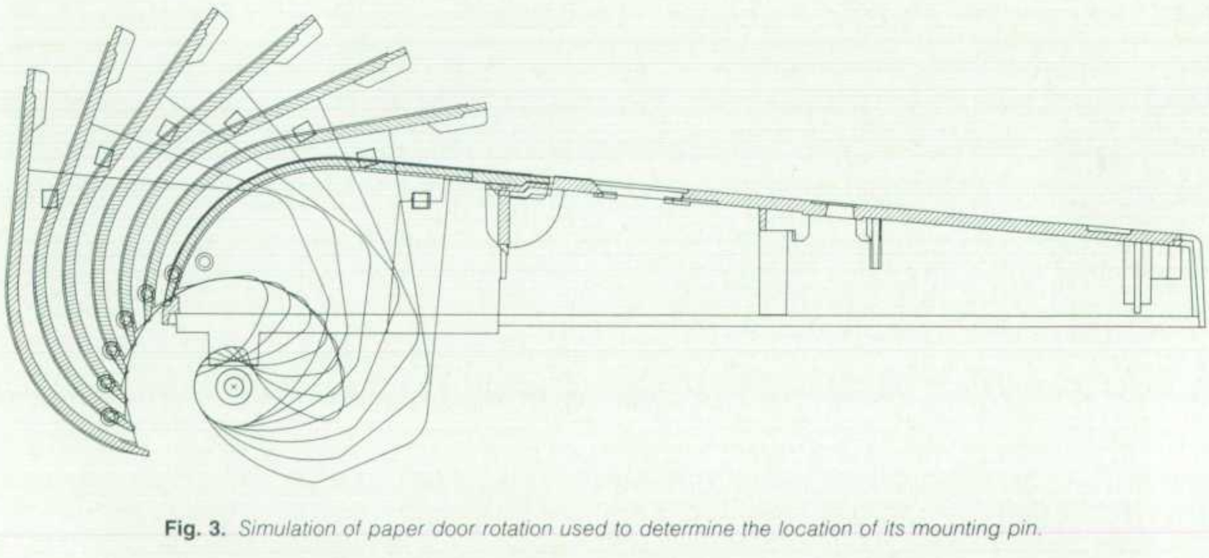


Fig. 3. *Simulation of paper door rotation used to determine the location of its mounting pin.*

Manufacturing State-of-the-Art Handheld Calculators

by Richard W. Riper

HANDHELD CALCULATOR users are demanding more functions in smaller, less-expensive packages. At the same time, new products must be brought out in shorter times. This requires greater cooperation between R&D and manufacturing. On the HP-18C and HP-28C project,[1] the lab and manufacturing teams began working together from the very beginning, long before the mechanical design was firmed up. This helped ensure that the designs coming out of the lab could be built easily on the production line. Also, on this "fast-track" project, much of the tooling work had to begin before all of the design details were worked out.

One form of this cooperation was to have R&D design into manufacturing's strengths—using existing processes and technologies where possible. This meant getting manufacturing personnel involved in the design process, for example by giving the printed circuit board designers guidelines for the size and arrangement of solder pads for the surface mount components and specifying the amount of clearance required for these components for loading. Guidelines for the design of the display assembly were also given so that technologies already developed for the automatic assembly of Series 10 calculator displays could be used.

There were a number of areas of the design, however, that required new manufacturing technologies. A pallet conveyor for final assembly, automated key trim and load, and robotic placement of RTV sealants were new areas for us, and our existing manufacturing capabilities had to be extended in other areas.

One new technology used on this project was CAD/CAM. The HP-18C and HP-28C were the first products at our

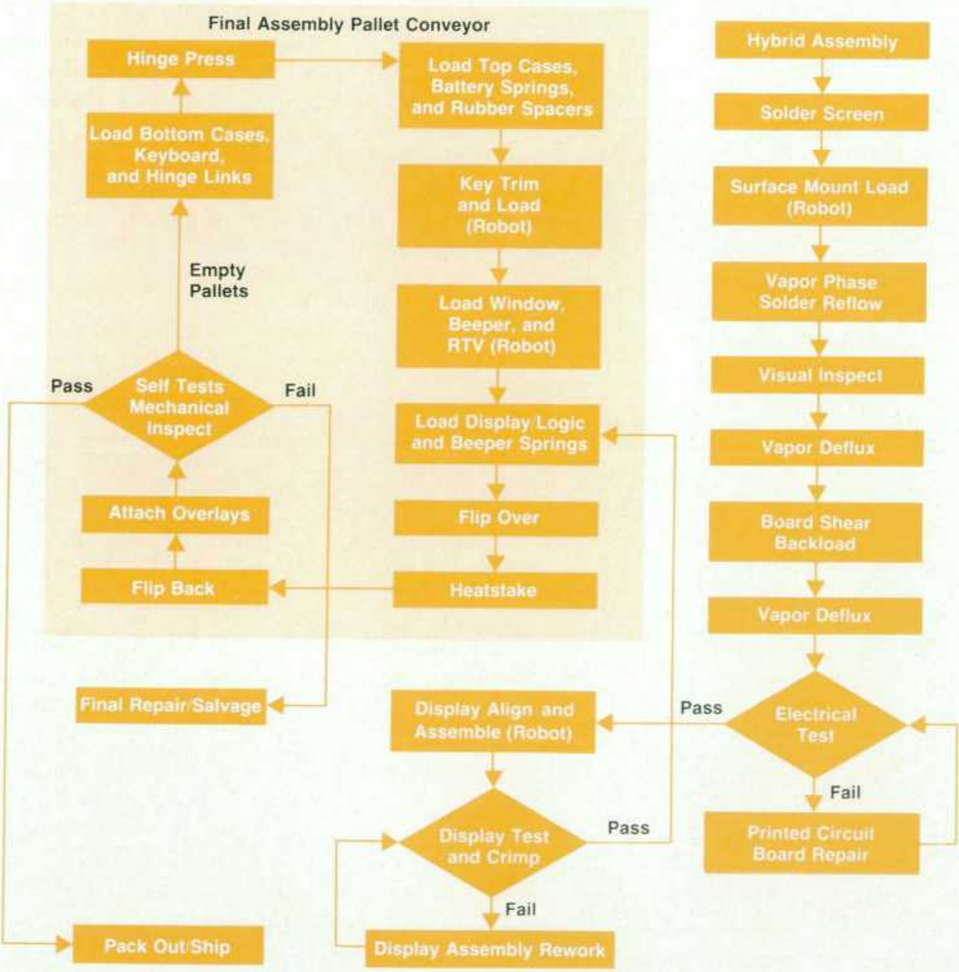


Fig. 1. *Flow chart of the assembly process for the HP-18C and HP-28C handheld calculators.*

division to be designed primarily on our CAD system, which consists of local workstations connected to HP 9000 Computers. This resulted in easier sharing of design information, which also was more accurate. Our CAD/CAM system is linked directly to several computer-controlled milling machines in our model shop, so that parts can be machined without having to reenter part geometry. Some of the tooling was also designed and built using CAD/CAM techniques, which resulted in tools that are more accurate and have better repeatability. Some of the machining on the plastic molds was done using the CAD/CAM system, such as the nomenclature for the keys, saving shop time compared to traditional methods. In addition, the layout of the final assembly line was done on the CAD system, which made it easy to try different alternatives and move assembly stations around for the best work flow.

Production Flow

The production flow (Fig. 1) begins with the electronic assembly. The circuit boards for the HP-18C and HP-28C are produced in a subpanel of four boards to reduce handling of individual boards and to fit the processing equipment. Three custom ICs, the two display drivers and the CPU, are attached and wire-bonded directly to the boards during the hybrid assembly. The panels then move to the printed circuit assembly area for other components. Here the subpanels are screened with solder paste and fixtured in a robot workcell for loading of the surface-mount components (Fig. 2). The robot arm contains a reflective-light sensor, which is used to find the exact position of the etched conductor traces on the subpanels. This ensures that the two ROMs and 12 discrete components are loaded accurately. This is an example of designing into manufacturing's strengths, since similar robots load boards for our established products such as the Series 10 Calculators and the HP-71B Handheld Computer. It is also an extension of our existing capability in that this is the first time we have added surface-mount components to a hybrid circuit.

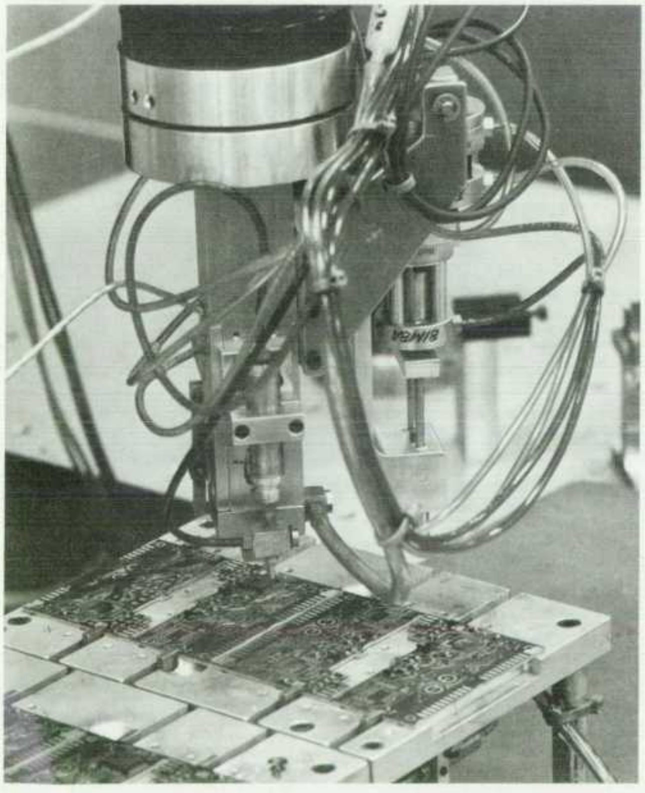


Fig. 2. *SMT robot.*

The HP-18C Business Consultant is produced in five languages: English, French, German, Spanish, and Italian. Currently, the HP-28C is produced as a single-language product. Each HP-18C language variation has its own ROM set so that softkey labels and messages appear in the localized language. The six different printed circuit assembly variations are produced by just changing the ROM set. This greatly reduces the number of subassemblies that must be stocked and reduces the changeover time to build a different model to minutes.

After robot loading, the subpanels enter an automated soldering system. They are fed on a conveyor through a predrying oven to drive off the solvent in the solder paste, then into a vapor phase reflow machine where the balls of solder in the paste liquefy, soldering the components onto the board. The subpanels are then inspected for solder joint quality, washed, and sheared apart into individual boards. Four components not currently available in surface-mount packages are then hand soldered to the boards before they are washed again and sent to electrical test.

The boards are electrically tested on an HP 3065 Board Test System using a fixture that was designed and built just for this product. The test fixture probes both sides of the board simultaneously, including all of the pads for connection to the liquid-crystal display (LCD). This fixture is also used for testing the subpanels during the hybrid assembly, which gets more leverage out of one tooling design. The tested assemblies then move to the final assembly area, where they are mated with the LCD before being placed into the calculator.

Final Assembly Conveyor

The final assembly of the HP-18C and HP-28C represents a departure from the way we have fabricated previous products. Instead of a rigid, one-piece package, the calculator has two main cases connected by a double-jointed hinge. This makes the product hard to hold during manufacturing, especially before all of the parts have been joined together. For this reason, we chose a pallet conveyor for moving the calculator through the assembly steps. Another benefit of using this type of conveyor is that the amount of work in progress (WIP) can be tightly controlled. This leads to better control of inventories and more efficient production.

The pallet conveyor assembly line (Fig. 3) is controlled by an HP Series 9000 Model 236 Computer. This computer is interfaced to the hardware through three HP 3488A Data Acquisition and Control Units and its control software is based on a program written to control a similar conveyor used in the assembly of the HP Vectra Personal Computer.

The pallets (Fig. 4) consist of a metal plate held in a plastic frame by four precision bushings. These bushings allow accurate alignment of the pallets at each of the automated stations on the line. The plates are fixtured to hold the parts of the calculator, with one calculator per pallet.

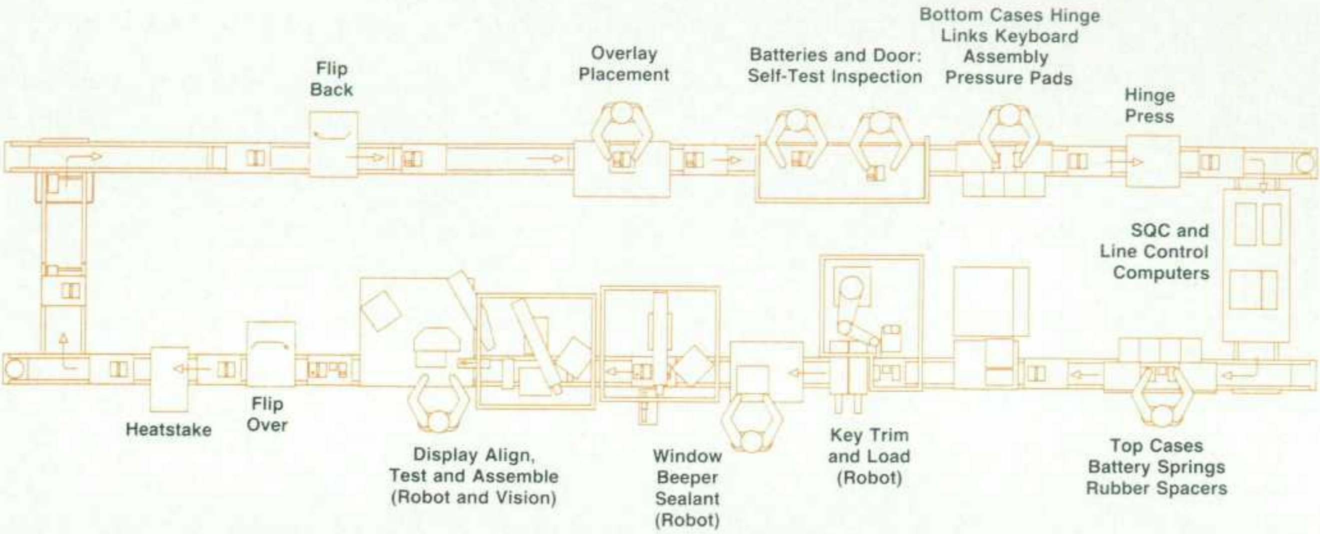


Fig. 3. *Final pallet conveyor assembly line layout.*

Metal plates embedded in the plastic frame of the pallet activate inductive switches mounted on the conveyor so that the control computer will know when the pallets are in position. The computer controls all pallet movement around the rectangular conveyor, signaling the automatic stations to start operation once a pallet is in position. The computer also watches for signals from the operators at the manual stations that they have completed their operations and the pallets can be moved on. The conveyor is asynchronous, so each pallet can move on once an operation is finished as long as there is space available for it downstream. This pallet conveyor then forms a mechanized demand-pull production line.

The first station on the final assembly line is a manual station where an operator places the two bottom cases and the hinge halves on the pallet. The polyester-film keyboard assembly is also placed in the cases and routed through the hollow hinge at this station. The second station is an automated hinge press which completes the snap-together assembly of the hinge halves. After moving across the end of the conveyor, the operator at the third station places the top cases into the pallet after snapping in the battery contacts. These formed wire battery springs give reliable contact to the printed circuit board without the need for hand-soldered wires. Soldered wires not only take a long time to assemble, but are very difficult to automate.

The next station on the assembly line is an automated key trim and load system. The keys are two-shot molded in clusters (the key label is molded from a different color of plastic, not just printed on). These clusters must be trimmed apart, leaving the individual keys. The keys cannot be molded in the same sequence as they appear in the product, as is done in our other calculators. In addition, the keys are molded from three different colors. This required the development of a robotic system capable of taking the trimmed keys and placing them in the proper locations in the top cases.

The controller for this robot has two parallel processors, which allows it to operate the key trim machines at the same time that the robot arm is loading keys onto a fixture that flips them down into the top cases waiting on the pallet. The clusters of keys are stacked in metal magazines after plastic molding, allowing automatic loading of up to 100 calculators without attention from an operator. The five different language variations of the HP-18C use the same set of keys, while the HP-28C uses a different set of clusters. Again, changeover can occur in a few minutes.

At the next station, an alcohol-cure RTV compound is dispensed into the cases by a robot to bond the display window and the piezoelectric beeper. RTV is also used to seal the electronics against damage if the batteries should leak and to provide additional protection against electrostatic discharges. A robot was chosen for this operation because it can apply a smooth, uniform, and continuous bead of RTV, an operation difficult for an operator to do by hand. At this station, as well as at the key-load station before it, the pallet is positioned by the bushings for accuracy. The windows are inspected by an operator for cosmetic defects and placed in a tray that is presented to the robot. The beepers are stack loaded in a magazine and fed automatically to the robot.

The next operation attaches the LCD to the printed circuit assembly (Fig. 5). This operation uses another robotic system, chosen for the high accuracy required. The liquid-crystal displays, as well as the metal clips that hold them to the printed circuit assembly, are tray loaded for the assembly robot. The robot places the clip in a holding fixture and passes the LCD under a reflective-light sensor to find the interconnection pads. The robot then takes the LCD to an automated tape dispenser and places strips of double-sided adhesive tape on each long side of the LCD. The robot then places the LCD into the metal clip, based on the position data sensed earlier. The LCD and clip assembly is then placed on a ramp that slides the assembly to an operator who places two elastomeric connectors over the LCD contact pads and mates the assembly with the hybrid printed circuit board.

The resulting assembly is then placed in a tester/crimper,

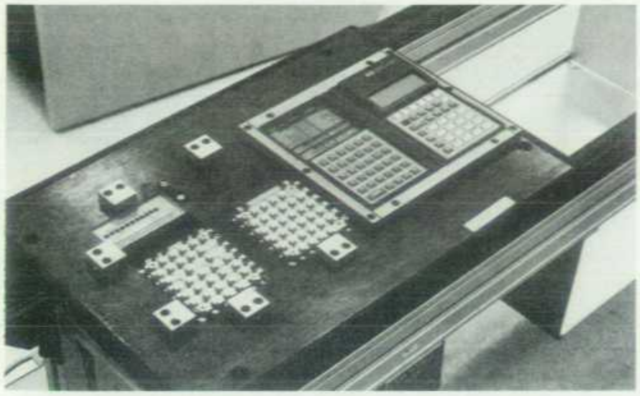


Fig. 4. *Final assembly pallet.*

where the keylines and several test points on the printed circuit board are probed and a set of diagnostic tests is run. A vision system uses two cameras to look at the LCD during this test to make sure that all of the pixels are operating. With over 4,000 pixels per display, this is not an operation that can be reliably done by eye. After the assembly passes the test, the tool automatically folds over the tabs on the metal clip, holding the printed circuit board and the LCD tightly together. The operator places this assembly into the calculator and places two conical beeper springs

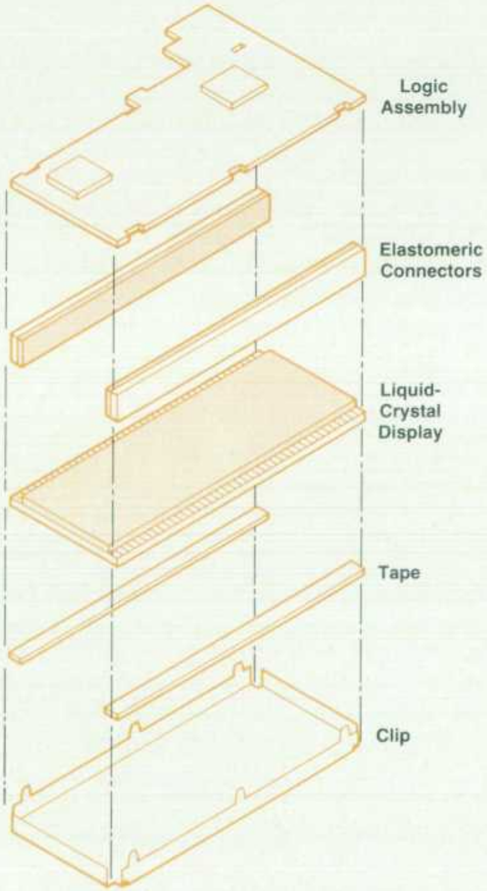


Fig. 5. *Exploded view of display assembly attachment to logic board.*

into holes in the printed circuit board. Like the battery contacts, these give reliable contact without the need for hand-soldered wires.

The pallet then moves to the first of two flip stations, which is an automated tool that picks up the back case assembly and mates it with the top cases. There are 101 plastic bosses molded in the bottom cases that have to fit through holes in the top cases, with little room for misalignment. The next station is an automated heatstaker, which uses heated pins to form rivet-like heads on the plastic bosses. These hold the calculator together—no screws are used. Once heatstaked together, there is no way of getting a unit apart without destroying the plastic cases. After passing across the end of the conveyor, the pallet stops at the second flip station, where the assembled calculator is turned back over into the other side of the pallet.

The next station is a manual station where the overlay labels are placed on the calculator. These four labels cover the heads formed on the plastic heatstake bosses and provide user information. The overlay set is different for each model and language variation being produced, just as the ROM set is. The pallet then passes on to the last station, where the batteries are placed in the calculator and a number of self-tests are run. The calculator is also inspected for cosmetic defects and sample printing is done on the separate printer to make sure that the infrared LED transmitter for sending data to the optional printer operates correctly. The empty pallet continues on the conveyor to the first station to start the cycle all over again. The finished calculators are carted to the pack-out area, where they are boxed with the owner's manual. The owner's manual is another part of the product that is peculiar to the model and language.

Acknowledgments

The other members of the manufacturing engineering team that worked on this project included Dirk Bodily, George Custer, Ken Frazier, Jerry Hackett, Horst Irmscher, John Liljeberg, Martin Marino, Ralph Sebers, and Bob Walsh. The final assembly line controller was developed by Bob Clark, Roger Quick, Kathy Shelby, and Carl Johnson. A great number of people in tool build, the model and NC shops, and elsewhere contributed a lot of time and effort to getting the tools ready on time. A special thanks to the production workers on the HP-18C and HP-28C, especially those who were involved from the earliest prototype builds.

Reference

1. Complete issue, *Hewlett-Packard Journal*, Vol. 38, no. 8, August 1987.

Information Technology and Medical Education

This paper discusses the use of information technology in an experimental curriculum at Harvard Medical School and describes several of the computer-based educational modules that have been developed for the program.

by G. Octo Barnett, M.D., Judith L. Piggins, Gordon T. Moore, M.D., and Ethan A. Foster

WITHIN THE LAST FEW DECADES, powerful forces have radically changed the scope and complexity of the health sciences and the delivery of health care. The explosion of knowledge in the basic medical sciences makes it a formidable task to keep abreast of the medical knowledge base. It has been estimated that over 600,000 articles are published in the biomedical literature each year. If physicians were to attempt to keep up with the literature by reading two articles per day, in one year they would be more than 820 years behind. If physicians were to read everything of possible biomedical relevance, they would need to read 1640 articles each day.

In addition to the problem introduced by the exponential growth of the medical knowledge base, the aging of the population, the shift from acute illness toward chronic disease, the emphasis on cost containment, the increasingly corporate nature of health care delivery, and the availability of information processing technology are radically changing the way that health professionals function today. These factors will surely shape the way that health professionals of the twenty-first century provide medical care.

Despite the major advances in the science and technology of health care, and despite the new challenges to health care, the training of physicians today differs little from what it was a half century ago. For all of the health disciplines, the structure of education still consists primarily of lectures in which a procession of teachers relate large quantities of scientific material to a passive student audience. This has placed impossible time demands on the curriculum, and has far outstripped the ability of students to absorb the quantity and complexity of scientific knowledge. It is not practical to increase the duration of professional education, it is educationally and medically undesirable to increase the fragmentation that results from narrow specialization, and it is not possible to depend on continuing education to fill the gap.

There is considerable reason to believe that computer-based educational applications can facilitate acquisition of essential knowledge and mastery of problem-solving skills. Comprehensive training and experience with modern methods of information management during the students' formative years may greatly enhance their effective functioning as health care practitioners and as professionals committed to life-long learning and teaching.

We recognize the inherent tension between the technological and the personal aspects of health care. Information technology has the potential to address the ever-changing and ever-broadening mass of knowledge concerning the etiology, prevention, and treatment of disease as well as the maintenance of health. This use of technology, however important, must not detract from the fundamental human aspect of care: the relationship of an individual health professional to an individual patient.

The potential application of information technology involves content, but more important, involves the method of education. Students should be given fewer "answers" and more "tools"—tools for self-teaching and for synthesizing, framing, and revising knowledge. They should have the opportunity to practice from the earliest days of professional education the skills of seeking information, testing hypotheses, and solving problems. The underlying objective is in part the transfer of current information, but more important, the creation of an environment where the student takes increasing responsibility as an independent learner.

The New Pathway

Harvard Medical School has initiated a new curriculum, called the New Pathway, which involves a basic restructuring of medical education and a greater emphasis on problem solving and independent learning. In September 1985, the New Pathway program started with a group of 24 students selected from the entering class of 165 students. The 24 students will remain in this separate curriculum throughout the four years of medical school. In September, 1986, 38 new first-year students also began the New Pathway program.

The New Pathway program makes extensive use of active educational methods such as problem-solving and information management, self-paced learning, and small group discussions. There are few lectures; instead there is more emphasis on the student assuming individual responsibility for his/her own education. One of the key elements of the New Pathway curriculum is the intensive use of information technology as a primary educational and information management resource to assist the student in the mastery of the scientific basis of medicine, and in the development of problem-solving skills. Each student and each faculty member has access to a computer for personal use. In addition, there is an electronic mail network that supports sharing of information among students and faculty. This computer technology is used to provide increased access to the knowledge base of medicine, both through an automated bibliographic reference capability and through the development of innovative computer-based programs to access medical knowledge bases.

Technical Issues

In 1984, Harvard Medical School received a five-year grant from Hewlett-Packard to provide an integrated computer network to support the communication and information processing needs of the New Pathway students and faculty members. As of September 1986, the New Pathway community is using the following hardware to support its information processing functions:

- An HP 3000 Model 48 Computer serves as the central computer for electronic mail communication.
- Eighty HP Touchscreen (HP 150) Personal Computers,[1] each with 640K bytes of memory, 15 or 20 megabytes of hard disc storage, a ThinkJet printer,[2] and a modem, are assigned to individual students and faculty.
- Fifty HP Vectra Personal Computers, each with 640K bytes of memory, 20 megabytes of hard disc storage, a ThinkJet printer, and a modem, are also allocated for individual student and faculty use.
- Thirty HP 2392 Terminals, each equipped with a modem and a ThinkJet printer, are assigned to faculty members who need access to electronic mail communication but do not need the other functions supported by the personal computers.
- Five HP Portable Computers[3] equipped with internal modems are used for word processing and accessing the electronic mail system by faculty on vacation or while traveling.

A number of HP-supplied software packages are used in the New Pathway program. These include:

- HP Desk[4,5] on the HP 3000 Computer for electronic mail communication.
- The MS™-DOS operating system for the personal computers.
- WordStar® on the Touchscreen Computers and HP Executive Memomaker on the Vectras for word processing. HP Slate is available on the HP 3000 for users of HP 2392 Terminals who have no stand-alone word processing capability.
- HP AdvanceLink on all personal computers for communications support.
- A selection of application packages, including more sophisticated word processors and data base managers, are used by some of the students and faculty on an *ad hoc* basis.

Training

The New Pathway users vary widely in their level of familiarity with computer technology—only five of the 24 first-year students had significant prior experience with computers (usually word processing). We provide a spectrum of support services that users can take advantage of as needed in their work with the personal computers.

Several hours of group instruction for students are held at the beginning of the academic year. Faculty members, who receive their computers at varying times during the year, can request individual training sessions. In addition, technical staff are available by telephone for specific questions about the computers on an on-going basis. These inquiries range from questions about how to accomplish a given task on the system to reports of hardware malfunctions. The staff attempt to diagnose problems over the telephone. If that is not possible, or if it is determined that there is a hardware difficulty, a technician is dispatched to the user site to investigate and replace any malfunctioning hardware modules.

One of the important goals of the New Pathway information technology effort is to provide computing power in a format that is easy to understand and use, and that is presented in a coherent and unified package. We wanted a system that would not intimidate those unfamiliar with technology, while still being convenient to use for those who had already had some experience with computers. We found HP's PAM (Personal Applications Manager[6]) screen to be an effective front end for allowing our users to select among the different software functions available on the personal computers.

In the modules developed for the New Pathway we have attempted to standardize the interface, including the way that function keys are used and the use of windows and their graphical presentation on the screen. We developed a set of utility programs to perform these and other common functions which we supply to new software developers as they join the project. This saves each developer from writing a new set of programs to perform these functions, and leads to more standardized and consistent software modules.

All application software developed for the New Pathway is written in the MUMPS® language.[7] MUMPS is an efficient data management and text manipulation language originally developed at Massachusetts General Hospital. It is now in use in a number of locations throughout the world for medically oriented software development as well as for business applications. MUMPS has been standardized by the American National Standards Institute (ANSI) so that all MUMPS systems support a common set of language features. MUMPS systems are currently available from commercial vendors for a large variety of computer hardware. Writing all of our software in MUMPS makes it easy for different applications to share files, use a common set of utilities, and call other applications as needed. It also simplifies the task of moving applications developed on the Touchscreen Computers to the Vectra Computers and vice versa. Finally, it enhances the transportability of the software we have produced to other medical schools, since these programs can be run on any computer for which a standard MUMPS system is available.

All communication between the HP 3000 and the personal computers is done over standard voice-grade telephone lines (Fig. 1). The decision to use telephone communications was based on the geographical distribution of the New Pathway user community. About half of the students live in the dormitory on the medical school campus; the others live in apartments scattered through the surrounding communities. Faculty members have offices located in almost a dozen different medical institutions separated by a considerable distance in the Boston metropolitan

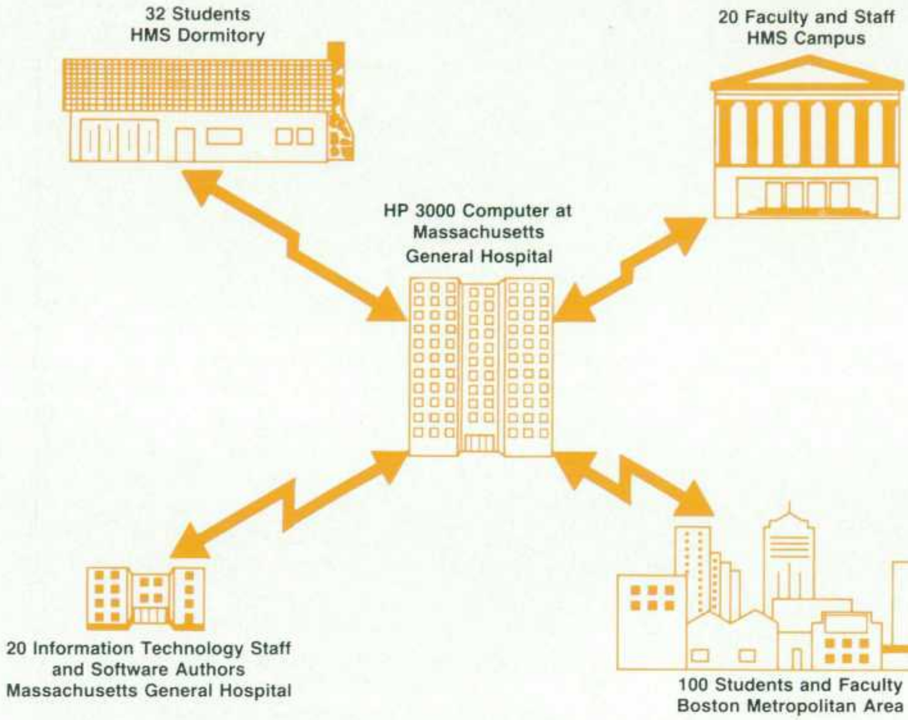


Fig. 1. *The various groups in the New Pathway program at Harvard Medical School (HMS) communicate with a central computer system via telephone lines.*

area, and a number of faculty prefer to have their computers located in their homes rather than in their offices. Telephone lines were the only feasible way of connecting such a widely scattered group of users.

To facilitate the communication between the personal computers and the HP 3000, we make extensive use of HP's AdvanceLink communications package on the personal computers. AdvanceLink command files have been written to dial the telephone automatically and log the user onto the HP 3000 and into HP Desk. Other command files are available to transfer text files created on the personal computers to HP Desk and bring HP Desk messages from the HP 3000 to be stored in files on the personal computers. This capability has been particularly useful for the development of curriculum materials and reports where a number of faculty members contribute to a common document. Similar command files have been written to access automatically the other computer services available to New Pathway users. The AdvanceLink capabilities of encrypting command files and turning the computer display on and off as required have made it possible for us to support access to these systems from New Pathway machines without publishing the passwords.

The use of telephone lines for communication has limited us to a communication rate to the HP 3000 of 1200 baud. This limitation has not proved to be a problem for electronic mail communication, which is the only application currently supported on the central HP 3000. The personal computers with their dedicated processors have been used thus far for all of our application development, which involves processor intensive activities such as graphics displays. An added benefit of equipping each personal computer with a modem for electronic mail is that it makes it possible for each user to access commercial timesharing services for New Pathway-related functions.

Distribution of new software modules as they become available is a time-consuming task in an environment such as ours where there are 130 different personal computers to be updated. Initially we accomplished this by duplicating the software on flexible discs and mailing a copy to each user. This proved to be quite expensive in terms of administrative time and supplies. We now use a set of AdvanceLink command files and related software to transfer programs and data files from the HP 3000 to the personal computers via telephone lines. When a new application is ready for distribution, a copy of all programs and related data for that application is placed on the HP 3000. Users are notified via electronic mail that a new application is available. Each user then calls the HP 3000 and issues a request to transfer that application to the user's personal computer. When the transfer is complete, software is available on the personal computers to transform the downloaded information into the appropriate format to be run as an application module. This new capability has considerably streamlined the task of making new applications available to our users.

Electronic Mail

We did not anticipate the degree of impact the ready availability of electronic mail would have on the educational experience of both faculty and students. Originally we believed that electronic mail was a relatively trivial application and that it would have little educational value. This view has proven to be a gross underestimation of the worth of electronic mail. We now consider this application one of the more important elements of the information technology support. In one typical four-month period during the first year the total number of accesses per student varied from 34 to 258. During one sample three-day period, all of the 24 students logged on at least once, and 15 students accessed the system on all three days.

The faculty uses the system for planning and developing the curriculum materials and cases, and for communicating with other faculty and with the students about specific course-related issues. Because of the geographical dispersion of the Harvard Medical School faculty, curriculum planning and development in the past was typically associated with many hours wasted in commuting to committee meetings and in trying to communicate by phone, often playing "telephone tag"—leaving messages to call, not being available when the return call is made, etc. The increased efficiency of the communication and information sharing among the New Pathway faculty made possible by electronic mail has come to play an important role in curriculum development and, more important, has strongly influenced how the teaching sessions are planned and how the different tutorial sessions are implemented.

The students use the system for sharing ideas about the curriculum assignments, for personal interactions, and for questioning the faculty. The students, as well as the faculty, believe that electronic mail facilitates faculty and student communication and cooperation in a community that may not have daily face-to-face contact. They feel that the intellectual "networking" made possible by the electronic mail system greatly improves the educational experience and makes it more enjoyable.

Medical Education Modules

A series of New Pathway medical education modules are incorporated into the curriculum as integral elements of the total educational experience. However, we emphasize that the New Pathway curriculum is *not* just a set of computer programs; the computer support is only one aspect of a much larger commitment to self-directed learning. The role of the computer is not to replace faculty contact but to improve faculty-student interaction and to enrich and extend the learning environment of the student.

The dominant challenge in the application of information technology to medical education is the development of software. There is a paucity of outstanding medical educational programs, and those that are available are often poorly documented, difficult to modify, and written in computer languages that we do not support.

Six major classes of software are being developed for the New Pathway curriculum:

- File management applications related to medical knowledge and patient records
- Programs that facilitate vocabulary acquisition
- Data base access applications
- Test bank questions
- Basic science teaching programs based on physiology simulations
- Clinical teaching programs based on case management simulations.

A number of programs developed for the New Pathway program are now in active use by the students; others are still in the prototype phase and are not yet integrated into the curriculum.

File Management Applications. An important goal in the New Pathway philosophy is to stimulate the students to manage their own learning process efficiently and effectively, including the recording of one's educational experiences. The typical medical student uses 3×5 cards and a pocket notebook to record key references, course notes, useful items of information, etc. The Personal Reference File is a support program that is now being developed to provide a computer-based solution that combines the functions of a personal notebook, a personal filing system, and a clinical case book.

An important component of the Personal Reference File is a standardized hierarchical vocabulary to index each record. This vocabulary is based on a terminology developed by the National Library of Medicine to index medical literature. We believe that this vocabulary, known as MeSH for Medical Subject Headings, is a good model for organizing the medical knowledge base. The 15,000-term vocabulary is implemented on each student's personal computer as a window-driven set of displays (Fig. 2). Within each window, the student can move up and down through the tree structure. Once in the tree, the user can call up at any time a map of the path from the base of the tree to the current term. The student can request information about other locations within the nomenclature that contain related terms. It is also possible to view all contexts of a term within the model and switch among these contexts. The use of this controlled vocabulary facilitates selective retrieval of information from the Personal Reference File, and allows students and faculty to share notes and references that are relevant to the same topic.

We plan to use this same strategy in the development of a clinical case book which can be used by the student to record a minimal description of each patient seen by the student. We will extend the MeSH vocabulary to provide the capability for the student to use a controlled nomenclature in recording these clinical terms. This primitive medical record provides practice for the student in recording and retrieving medical record data using a computer system, and allows review of the clinical experiences of each student to identify areas where there are gaps in the educational experience.

Vocabulary Acquisition. It is estimated that the medical student's vocabulary must double during the first year of medical school in learning the basic language of medicine. Particularly in the first two years, the student must engage in brute-force memorization of material, the relevance of which is not always apparent to the student. A major effort of New Pathway software development is devoted to providing more efficient and effective techniques for learning the concepts, nomenclature, and definitions of medicine. These computer applications provide dynamic and interactive experiences that use graphic and visual images and integrate each new term or concept with the knowledge base already acquired.

One such program developed for the New Pathway program teaches the student to recognize the visual patterns and accompanying nomenclature of normal and abnormal blood cells. This program uses a video disc controlled by a teaching program on the Touchscreen Computer to present images of different types of blood cells in an interactive game format to challenge the student with different pattern recognition problems. The computer program requires the student to classify and interpret the different images, and

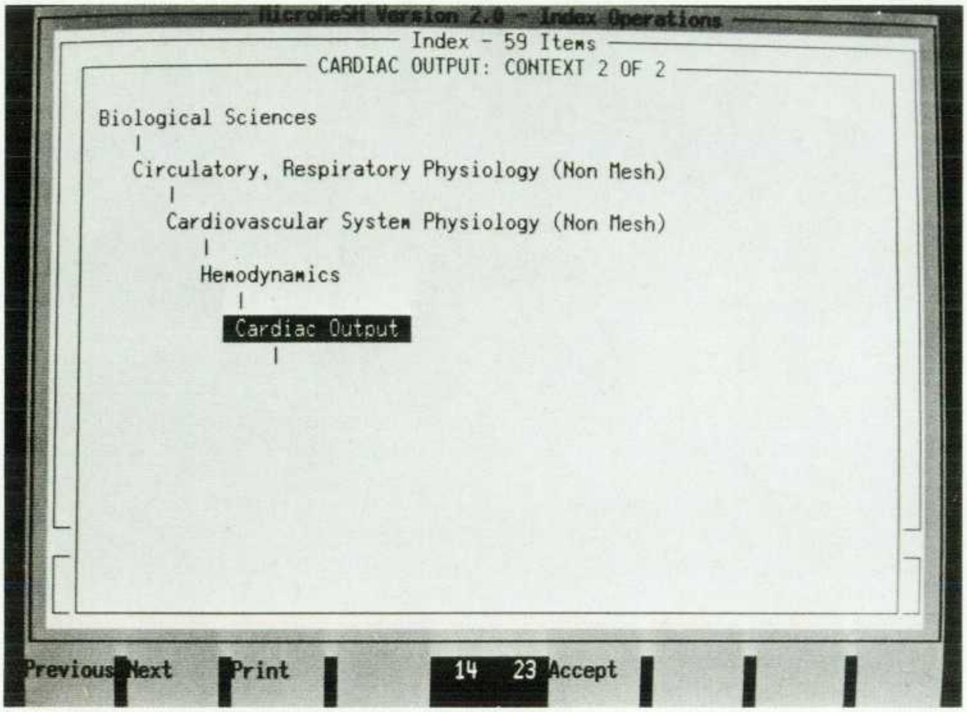


Fig. 2. *A window-driven set of displays allows students to browse easily through the MeSH vocabulary.*

provides feedback using other visual images to illustrate correct and incorrect interpretations by the student.

Data Base Access. A central theme of the New Pathway program is the promotion of student-directed learning using a variety of educational resources; thus it is expected that the student will become skilled in accessing the published medical literature. A major limiting factor is the sheer volume of published articles of interest. More than 3,000 journals containing more than 17,000 articles are indexed each month in the standard National Library of Medicine MEDLINE file of bibliographic references. MEDLINE now contains over six million references to articles published in the United States and seventy foreign countries since 1965. Even though Harvard Medical School has a large and easily accessible medical library, time limitations often make it difficult for the student to locate and read appropriate articles or textbooks.

All New Pathway students are provided access to a national, computer-based, on-line information retrieval system—BRS/Saunders COLLEAGUE—which contains both the complete MEDLINE file provided by the National Library of Medicine and also full text files of many of the more important journals and textbooks. Programs have been written to allow the students to use their own computer and modem to access these files through a national communications network. This provides them with an extraordinary opportunity for easily accessing the medical literature to supplement their learning in each aspect of the New Pathway curriculum.

Computer-Based Test Bank Questions. The educational strategies of the New Pathway curriculum are designed to promote independent learning and self-assessment. Student evaluation is based on competency and accomplished through regular formal evaluation procedures. In addition, all students are required to take the National Board of Medical Examiners (NBME) tests. As part of an informal self-assessment of knowledge acquisition on a continuing basis, and as part of the preparation for the NBME tests, a test bank of questions is available on each personal computer.

This test bank does not simply score on a right/wrong basis but provides active feedback and interpretation as to the validity of the different answers. The item bank includes questions that are multiple-choice assessments of factual recall and questions that challenge the student with problems of interpretation and integration of information in problem-solving situations. Questions on a variety of topics are currently available and additional sets of questions are being developed by faculty and students. Clinical simulations of patient cases will also be used to evaluate the student's ability to deal with complex clinical data. It is not intended that this test bank be used in formal evaluation of the student, but rather that it provide checkpoints to assist both the faculty and the students in assessing the successful acquisition of factual information and clinical problem-solving skills.

Physiology Simulations. In the basic medical sciences (e.g., biochemistry, physiology, pharmacology, etc.), it is often useful to classify biological phenomena according to body systems, and to describe their relationships using general system theory. In a system model, the student must learn how the different elements interact with each other, how the different parameters change with time, and how the parameters change as a result of external perturbations. In the medical school curriculum, laboratory exercises using animal experiments are used to demonstrate these systems, and to provide the opportunity for the student to learn the response of the system to different experimental interventions. However, such laboratory experiences are time-consuming and expensive to set up and supervise, and often do not allow the student opportunity for extensive experimental interventions.

In many situations, computer simulation of biological systems can simulate the laboratory environment so that the student can observe a realistic model of the system in a variety of states, introduce different interventions, and learn by trying "what-if" experiments. Computer models can provide the student the opportunity to manipulate various elements of the system and gain an understanding of the interactions and the feedback controls that occur among the elements of the model. Computer-simulated models allow the instructor to specify and control the experiment more explicitly. Each student, or group of students, can be presented with the same challenge on the same experiment.

Several computer-based physiological simulations have been developed for the New Pathway curriculum so far, including carbon dioxide regulation by the lungs and acid-base regulation. In the acid-base simulation (Fig. 3), for example, the student learns to characterize a variety of different acid-base abnormalities and to select appropriate therapies to correct each specific type of abnormality. In this program, extensive use is made of graphical presentation of standardized plots of the relations between the different biological variables.

Computer-Based Patient Simulations. Problem solving is a fundamental activity of medical practice. Important tasks include deciding what clinical information to collect, and determining when there is sufficient data to justify making a diagnosis. The nature of clinical problem solving has been the subject of considerable research in recent years, with a general consensus being that it is an iterative and sequential process. The physician must collect a minimal data set first, formulate a diagnostic hypothesis that might explain the clinical findings, and then decide what further information should be collected to refute or refine the hypothesis. Further information is then collected, the data base is extended, and the diagnostic hypothesis is modified in light of the new information. This iterative process of data collection, hypothesis generation, and hypothesis testing continues until all relevant diagnostic issues have been resolved. This iterative hypothesis testing has been labeled a clinical experiment (analogous to a scientific experiment) where a successful result depends on the judgment and skill of the investigator (in this case, the physician) in the design, implementation, and evaluation of the experiment.

During the last two decades, members of our group at the Laboratory of Computer Science at the Massachusetts General Hospital have been developing patient simulations of the generic class of case management problems. These patient simulation computer programs attempt to provide the student an educational experience that gives insight and practice in hypothesis formulation and data interpretation which would otherwise be difficult or impossible to provide with either lectures or assigned reading. The computer-based patient simulations allow the student to formulate hypotheses, decide what further information to collect, and improve skills in data interpretation and complex pattern recognition.

In constructing these problem-solving simulations, the author must construct a model of the underlying disease process and how this disease might respond to therapeutic management. The computer program takes advantage of this knowledge of the simulated patient to respond to the student's requests for information or therapeutic maneuvers by describing the patient or the patient's response to treatment.

The series of programs developed at the Laboratory of Computer Science include such diagnostic problems as coma, abdominal pain, anemia, bleeding disorders, meningitis, dyspnea, and joint pain. All of these patient simulation programs are available to New Pathway students via

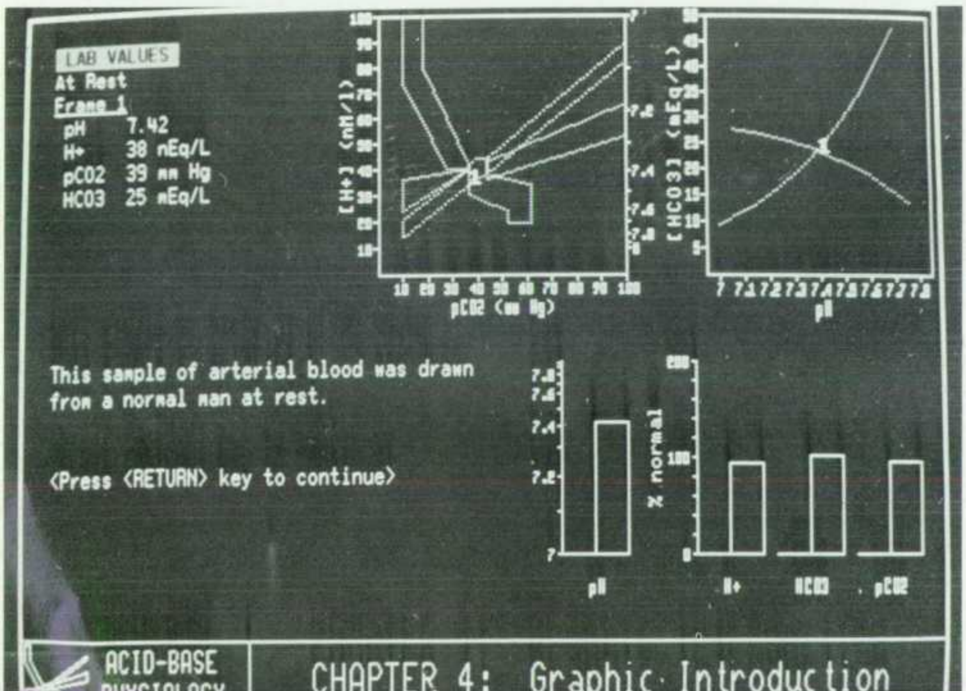


Fig. 3. *Example of acid-base simulation display.*

dial-up access to a computer located at the Massachusetts General Hospital, and a subset of them are available on each student's personal computer. The special features of this computer-based patient simulation series of programs are threefold:

- Each computer simulation emphasizes a probabilistic rather than a deterministic model of clinical problem solving, focusing on the uncertainty in the collection of clinical information and in the formulation of a differential diagnosis.
- The underlying diagnostic model allows the program to function in a professor mode, by critiquing the student's performance in terms of efficiency of data collection and appropriateness of diagnostic interpretations.
- At any point in the work-up of the simulated patient, the user can request advice as to the most appropriate data collection strategy or can request that the system provide the appropriate differential diagnosis hypothesis given the information then available.

Evaluation

The evaluation of any innovation in graduate education is difficult since the students involved in the innovation are heterogeneous in background and skills. In addition, their performance is strongly influenced by factors other than the specific details of any change in the curriculum. Evaluation of the information technology innovation in the New Pathway program is particularly difficult since the overall objectives of the new curriculum are not specifically concerned with student performance on any test of factual knowledge. Instead, the goal is to stimulate the students to acquire the total set of skills, attitudes, and knowledge that will result in their becoming competent and caring physicians in professional careers covering the next several decades.

Thus far, the only specific evaluation of the information technology aspects of the New Pathway program is a student questionnaire that was completed by the first class at the end of the first academic year of the new curriculum. The questionnaire requested the students to rate a number of the computer-based applications according to the following scale:

7—This application is of extraordinary value; my medical education is of significantly greater value because of the availability of this application.

6—This application is of great value; it should be supported and extended if possible.

5—This application is of value; I find it useful.

4—This application is of some value, but not high priority.

3—This application is of little use to me.

2—This application is of no use.

1—This application is distracting, a time sink. I recommend it be terminated immediately.

X—I have not used this application sufficiently to make a judgment.

The list of applications to which the rating scale was applied included electronic mail, BRS/Saunders COLLEAGUE access, word processing, test item data bank, physiology simulations, and patient simulations. Twenty-two of the 23 students returned the questionnaire. Table I summarizes the results.

Table I

Application	Number of Students Who Rated the Application	Range of Rating	Average Rating	Number Rating 6 or 7
COLLEAGUE	22	5-7	6.8	21
Word Processing	22	5-7	6.2	19
Electronic Mail	22	5-7	5.9	15
Physiology Simulations	19	3-7	5.4	10
Test Item Data Bank	16	2-7	5.4	9
Patient Simulations	15	3-7	5.1	5

We interpret these results to mean that most of the modules are felt by the students to have a very high educational value. We believe the evaluation by these students is relevant and important, for these are students deeply committed to their own education and very cognizant of the enormous amount of knowledge and skills that must be acquired in a very short period of time.

In the initial planning of the New Pathway program, there was concern that the use of the computer technology might interfere with personal communication among the students and between students and faculty. We asked the students to consider this issue in a separate question. Of the twenty students who responded, sixteen chose the response "The computer system and network capabilities have significantly improved and enhanced the communication I have with my fellow students and with the faculty," while the remaining four students chose the response "The computer system has somewhat improved and enhanced the communication." This strongly suggests that our initial concern was not only groundless, but that, in contrast, the students perceived that the computer system enhanced the level of communication.

Evaluation Questions Yet to Be Considered

There are a number of important issues that have not yet been addressed, but which must be considered before any final evaluation of the technology innovation in the New Pathway program can be made. Perhaps the most important issue deals with the cost of the technology—including hardware, software, and support personnel. Financial resources are finite and there are many other competing needs; it is not enough simply to claim that information technology is a popular and effective educational innovation.

At the present time all of the costs for the computer efforts of the New Pathway program are supported by a generous grant from Hewlett-Packard Company and by grants from several foundations. It is likely that the cost of the hardware will decrease in the coming years, although this decrease may be counterbalanced by a desire to use more expensive technology (e.g., higher resolution graphics, color displays, more powerful workstations, video discs, etc.). Costs for support and maintenance of the hardware and software as well as ongoing program development will remain substantial. There is a legitimate concern about any innovation that would increase the already high costs of medical education. We believe it will be most difficult to identify educational costs that can be replaced as a result of the introduction of information technology.

A second and related issue is to demonstrate the utility of students having their own personal computers, versus making available a smaller number of computers located in central teaching areas, a dormitory common room, and the library. We will explore this alternative approach in the next few years as we respond to a legitimate concern on the part of a growing number of students who are not in the New Pathway program, but who want equal access to the benefits of the information technology.

We are also concerned with the extent to which the information technology developments can and will be adopted by other medical schools. Obviously, if the technology innovation is successful, it would be foolish to expect that each of the 170 different medical schools should develop its own program and its own software. However, we do not know yet to what extent programs developed by our faculty for our students and for our educational environment will be acceptable to other faculties in other educational environments. We hope to address this question in the next two years by initiating a collaboration with several other medical schools whose educational objectives are similar.

References

1. S.Sukumar, "Touchscreen Personal Computer Offers Ease of Use and Flexibility," *Hewlett-Packard Journal*, Vol. 35, no. 8, August 1984, pp. 4-6.
2. C.V. Katen and T.R. Braun, "An Inexpensive, Portable Ink-Jet Printer Family," *Hewlett-Packard Journal*, Vol. 36, no. 5, May 1985, pp. 11-20.
3. J.T. Eaton, et al, "Design of HP's Portable Computer Family," *Hewlett-Packard Journal*, Vol. 37, no. 7, July 1986, pp. 4-13.
4. I.J. Fuller, "Electronic Mail for the Interactive Office," *Hewlett-Packard Journal*, Vol. 34, no. 2, February 1983, pp. 20-29.
5. L. Hurtado-Sanchez, et al, "Implementing a Worldwide Electronic Mail System," *Hewlett-Packard Journal*, Vol. 37, no. 9, September 1986, pp. 30-48.
6. P.S. Showman, et al, "Applications Software for the Touchscreen Personal Computer," *Hewlett-Packard Journal*, Vol. 35, no. 8, August 1984, pp. 15-24.
7. J. Bowie and G.O. Barnett, "MUMPS—An Economical and Efficient Time-Sharing System for Information Management," *Computer Programs in Biomedicine*, no. 6, 1976, pp. 11-22.

Authors

October 1987

4 — In-Service TIMS

Norman Carder

Norman Carder is a senior development engineer with HP's Queensferry Telecommunications Division. A native of Belfast, he received his BSc degree in electrical engineering and electronics from Queens University in 1978. He joined HP the same year, and has contributed to the design of high-speed modems, the HP 3764A Transmission Analyzer, and the HP 4948A ITIMS. His work on the HP 4948A resulted in a patent application. In 1984 he received an MSc degree in digital techniques from Heriot-Watt University. Now a resident of South Queensferry, Scotland, Norman is married, plays bass guitar, and enjoys camping, walking, and skiing.

William I. Dunn

Ian Dunn joined HP in 1984 with a background in superconductivity research, marine research, and electronic consulting. He has published papers on quantitative sonar techniques and a paper on energy dissipation in superconductors, and is named coinventor in a sonar-related patent application. He contributed to the control processor firmware design for the HP 4948A ITIMS. A native of Glasgow, he holds a BSc degree in electrical engineering from Glasgow University. Ian lives in Kinross, Scotland, is married, has three children, and is involved in politics. To relax, he favors mountaineering, skiing, photography, gardening, cooking, and taking his trailer to the Scottish Highlands or the continent.

David W. Grieve

David Grieve is a senior design engineer with HP's Queensferry Telecommunications Division, and has contributed to the design of the HP 37212A Modem and the HP 4948A ITIMS. He received his BSc degree in electronic engineering from the University of Manchester Institute of Science and Technology in 1979 and joined HP the same year. Born in Cuckfield, Sussex, England, David now lives in Edinburgh, and has made learning to speak Chinese one of his goals. He's married, and he and his wife are expecting their first child soon.

James H. Elliott

Jim Elliott is a specialist in microprocessor hardware and firmware design. He received his BSc degree in electronic engineering from Edinburgh University in 1982 and his MSc in digital techniques from Heriot-Watt University in 1986. A design engineer with HP's Queensferry Telecommunications Division since 1982, he contributed to the design of the HP 4948A ITIMS. Jim was born in Dunfermline, Scotland, and now lives in Edinburgh. He plays badminton, has a blue belt in karate, and likes to travel.

W. Gordon Rhind

4948/Elliott

Gordon Rhind received his BSc degree in electrical engineering in 1969 and his PhD in digital telemetry in 1972, both from Edinburgh University. He joined HP's Queensferry Telecommunications Division in 1979 with experience in the design of inertial navigation systems. He's been a senior design engineer for high-speed modems, was project manager for the HP 4948A ITIMS, and is now a section manager. A member of the IEE, he's the author of a paper on applications of inertial systems to land survey and is named a coinventor in a patent application on the HP 4948A. Gordon was born in Armadale, West Lothian, Scotland and lives in Linlithgow, West Lothian. He's married, has two sons, and has been known to participate in running, squash, and golf. He's also a connoisseur of "real" ales.

16 — Infrared Link

Robert S. Worsley

With HP since 1977, Bob Worsley has contributed to the design of a number of calculator and printer products. His past projects include the HP-41CX Calculator, HP 82143A Printer, HP 82240A Thermal Printer, and the HP-18C. Bob was born in Stamford, Connecticut and is a graduate of San Jose State University (BSEE 1977) and Oregon State University (MSEE 1982). He lives in Corvallis, Oregon, is active in his church, and likes hiking, cross-country skiing, and gardening.

Bruce A. Stephens

With HP since 1979, Bruce Stephens is an R&D engineer in the Handheld Computer and Calculator Operation. His first HP project was the HP-71B Computer and he worked on the hardware interface for the HP-18C and HP-28C Calculators. In addition to designing and developing calculators, he has played a major role in creating firmware development tools. Born in southern California, Bruce holds a 1979 BS degree in computer science from the University of California at Irvine. He's an avid runner and enjoys skiing, bicycling, birdwatching, and listening to all kinds of music.

Steven L. Harper

Steve Harper came to HP in 1972, the same year he graduated from Brigham Young University with an MSEE degree. He has worked on software for the HP 9551D Instrument Calibration System and on calculator microprocessor design for the HP-71B Computer and has been a project manager for HP-IL interface products. Most recently he contributed to the development of the HP 82240A Thermal Printer. He has written several articles on HP-IL, including two in the *HP Journal* (January 1983), and is coauthor of a book on the same subject. An Oregon native, Steve was born in Medford. He and his wife and five children currently live in Corvallis. He's active in his church and likes boardsailing, camping, softball, and tennis.

21 Portable Printer

Masahiko Muranami

Born in Tokyo, Japan, Jack Muranami earned his BSME degree from Rice University in 1985. He joined HP the same year and was a member of the R&D design team for the HP 82240A Thermal Printer. His speciality is the design of plastic and metal parts. Jack is married, has a young son, and lives in Corvallis, Oregon. His pastimes include gardening, cooking, and playing with his son.

David L. Smith

With HP since 1982, Dave Smith is a specialist in plastic part design. He has designed parts for the HP-71B Computer, HP-18C Calculator, HP 82240A Thermal Printer, and HP-94 Computer peripherals. He holds a 1982 BSME degree from California Polytechnic State University at San Luis Obispo. An Oregon native, he was born in Klamath Falls and now lives in Corvallis. He's married and is building a house this summer with the help of his family. His other interests center around outdoor sports, especially skiing and running.

24 Calculator Manufacturing

Richard W. Riper

With HP since 1982, Rick Riper coordinated manufacturing engineering for the HP-18C and HP-28C Calculators. Earlier, he was a manufacturing engineer for HP-10 Series Calculators and the HP-75D Computer. He also worked on a cost-reduction project for the HP 82161A Cassette Drive and on the HP 82162A Printer. Rick was born in Chester, Pennsylvania and is a 1982 graduate of Oregon State University (BSME). He's a certified professional engineer in the State of Oregon. A resident of Corvallis, Oregon, he's married and has two children. His outside interests include bicycling and photography.

28 New Pathway Curriculum

G. Octo Barnett

Octo Barnett was born in San Diego, California and earned a BA degree in chemistry at Vanderbilt University (1952) before completing work for his MD degree from Harvard University in 1956. He is a professor of medicine at Harvard Medical School and director of the Laboratory of Computer Science at Massachusetts General Hospital. Early in his career he served in the U.S. Public Health Service and was an established investigator for the American Heart Association. Later he specialized in medical information systems, medical informatics, and computer-based medical education. He has published over 100 papers in the field of medical informatics and is a member of several professional societies. Now a resident of Newton, Massachusetts, Octo is married and has three children, one of whom works for HP. For recreation he enjoys windsurfing, cross-country skiing, tennis, wilderness camping, and canoeing.

Gordon T. Moore

Gordon Moore received his undergraduate education and medical training from Harvard University and completed work for his MD degree in 1963. He served in the U.S. Public Health Service and later became chief operating officer and medical director for the Harvard Community Health Plan. He's now director of the New Pathway Project for Harvard Medical School. His publications include over 20 articles and papers in the fields of health services research and medical education. Born in Buffalo, New York, Gordon is a resident of Cambridge, Massachusetts. He's married and has two children and enjoys piano and sailing.

Judith L. Piggins

Judy Piggins holds BSEE and MSEE degrees from the Massachusetts Institute of Technology (both awarded in 1973) and an EdM degree in counseling and consulting psychology from Harvard University (1979). She designed and implemented medical software for Hewlett-Packard from 1973 to 1978 and has taught mathematics and computer science. She now manages the New Pathway Project for Harvard Medical School. She is author or coauthor of 13 papers on medical informatics and specializes in medical education and computerized medical record systems. Judy was born in Detroit, Michigan and lives in Watertown, Massachusetts. Scuba diving, hiking, aerobics, reading, and travel are her favorite leisure activities.

Ethan A. Foster

Ethan Foster is a Massachusetts native with a BS degree in computer science from Worcester Polytechnic Institute (1981). He spent three years as a programmer/analyst for medical data base systems at Massachusetts General Hospital before becoming lead programmer/analyst for the Harvard New Pathway Project. Ethan was born in Greenfield and he and his wife now live in Haverhill. He enjoys canoeing, camping, and skiing.

37 Program Development

Robin M. Gallimore

Robin Gallimore is software engineering department manager at the HP Laboratories Bristol Research Centre. With HP since 1985, he has also served as project manager for rigorous system specification and design. Before joining HP, he was a university lecturer in computation at the University of Manchester Institute of Science and Technology. He has published several papers on formal methods in software engineering, and is a member of the British Computer Society special interest group on formal aspects of computer science. He graduated from the University of Manchester Institute of Science and Technology with a BSc degree in computation in 1971. Robin is married and has a baby son. His interests include photography, jazz, and playing classical guitar.

Derek Coleman

Derek Coleman is a project manager at HP's Bristol Research Centre, working on tools to support rigorous software development using formal specification. Before coming to HP in 1985, he was a university lecturer at the University of Manchester Institute of Science and Technology and a visiting professor at the University of California at Berkeley. A member of the British Computer Society and a committee member of the special interest group on formal aspects of computing, he has published several papers on software engineering and a book on structured programming. He received his BSc degree in physics in 1965 and his MSc in computer science in 1970 from the University of London.

A Framework for Program Development

This paper presents a framework for recording a software design activity as a directed acyclic design graph, where each node denotes a fragment of the design and each arrow represents some kind of design decision that has been made.

by Derek Coleman and Robin M. Gallimore

SOFTWARE DESIGN is a complex process involving many decisions. The developer is faced with the task of formalizing the problem and reducing its level of abstraction to that of the programming language to be used.

It is on a clear understanding of the design that many life cycle activities such as maintenance, porting, and upgrading depend. However well a program is structured, its text is the result of the design process and is not to be confused with the design itself. A program text does not record the sequence of design steps that led to its construction. Traditionally the more abstract aspects of a design are documented *a posteriori* in natural language. This is as inadequate as it is imprecise.

This paper is an initial report on FPD, a Framework for Program Development. FPD supports an idealized history of a program development activity, and is similar to the inferential programming system suggested by Scherlis.[1] The goal is to permit the documented construction of correct programs. It is not intended to bind the designer to any particular methodology. Rather, the aim is to allow the exploration of design possibilities, with enhanced confidence that every avenue is safe.

As each decision is made there is an associated proof obligation,which must be satisfied to maintain correctness. The proof obligations provide the hooks for verification during design. Whether they are proved formally or justified informally determines the degree of certainty about the correctness of the resulting program. An important advantage of the framework embedded in FPD is that it can support rigorous software development[2] within a spectrum of formality.

The framework presented here uses a graphical notation to help record some aspects of the design of a program:

- The component structure of the program and relationships between components are shown.
- Explicit specifications, recorded separately from implementations of component bodies, define a component's global (interface) behavior in an implementation independent way. The graphical notation distinguishes statements of *what* a component does from *how* it does it.
- A component specification may have several alternative implementations (in a single design), supporting choice of implementations and component reuse.
- The definitions of, for example, the data types upon which a component specification has been based, are included in the design record in a hierarchic, incremental way.

Using the framework, component and program specifications are recorded during program design in a systematic, possibly formal, way. Significantly, the specifications can be related to both the definitions on which their intended meaning is based and the code modules that implement the component bodies using a simple graphical notation.

Structuring documentation in this way can aid the communication and comprehension of design decisions relating to program structure, the functionality of each component, representation of data, and the choice of component implementations. It can also drive the validation of design and implementation decisions by highlighting, graphically, the steps that have been taken and what criteria they must satisfy to produce an acceptable solution.

The current state of the art makes a totally formal approach to software development impractical. Currently FPD is used to combine formal specifications, both executable and nonexecutable, with design-time testing and semiformal reasoning. In this form it is being used at Hewlett-Packard Laboratories' Bristol Research Centre by a team constructing a suite of computer-aided software engineering tools. Further research will explore ways of increasing the formal content of practical software development.

Issues in Software Development

The quality of the development process is crucial in the production of software. Some of the criteria against which design practices should be judged are:

Correctness. How can the design process be organized to reduce the occurrence of bugs? It is especially important to avoid design errors, since these are much more expensive to locate and fix if they remain undiscovered until the software is in the field.

Reuse. Many software projects repeat in slightly different contexts work that has already been performed. Improved design methods should permit the reuse of both code and designs.

Maintenance. Maintaining and upgrading software is difficult since it requires the rediscovery of concepts used during the development of the implementation. The technical documentation for in-service software should include design blueprints similar to those used by maintenance engineers in other technologies.

Practicality. Formalism should be used wherever it is appropriate, but it should not be allowed to derail progress. A useful way to regard formalism is as an underpinning for informal arguments. If rigorous development methods are to gain widespread acceptance it is important that they be capable of modification to accommodate varying degrees of formality.

Correctness is the most fundamental issue since it makes no sense to reuse bug-ridden software. Similarily, maintenance costs of correct software will be much lower, since much maintenance is in reality bug-fixing.

A Framework for Program Development

Software design is a creative process. The designer navigates through the design, backtracking and aborting paths that turn out to be unpromising. These blind alleys and false starts are not useful in the postdesign life cycle. Only an idealized version of the design history is required.

It is important that the idealized design history be monotonic. Once a decision has been made, it must not be corrupted by later decisions. Preservation of correctness is thus a central concern that requires all design steps to be explicit and have associated proof obligations.

Large programs, incremental development, and reuse demand modularity. Designs and code must be modular to permit construction by teams of designers and to improve understandability. Modularity and specification go hand in hand since specifications serve as the interface definitions between components.

In FPD, program development is viewed as a directed acyclic graph, where nodes represent design fragments and the arrows represent the kind of derivation step that has been made (see Fig. 1). An entire design graph represents the history of a design. Each node denotes a program at a particular stage of its development.

A design can be factored into four kinds of components: program specifications, program code, data types, and

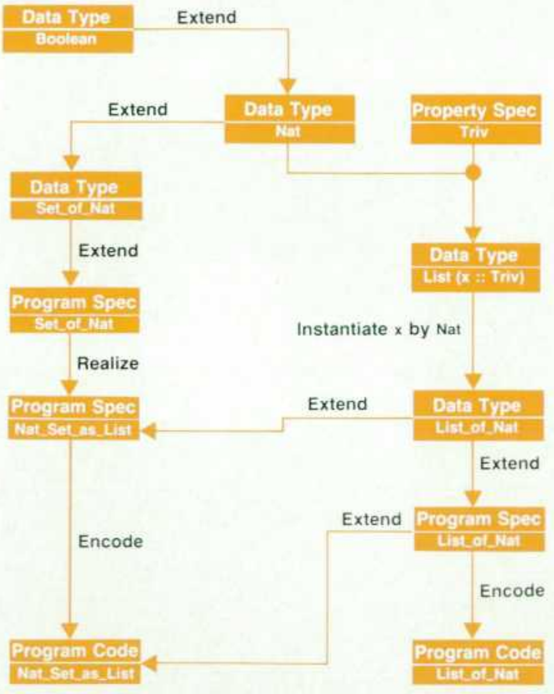


Fig. 1. *Implementing sets as lists.*

property specifications. In the examples that follow, Common Lisp is used as the programming language, but this is not essential to FPD.

Program specifications are abstract machines that implicitly specify the changes of program state to be performed by imperative program operations. They can be specified using pre-post conditions.[2] The precondition is a predicate over the initial state and indicates the conditions for which the operation is defined to have an effect. The postcondition is a predicate that defines correct results by characterizing the relation between initial and desired final states.

For example, a Common Lisp form that performs set union and updates its first parameter can be specified by:

```
FORM (UPDATING-UNION a b )
IN b: set
UPDATE a: set
PRE True
POST a = 'a U 'b
```

where IN and UPDATE indicate read-only and input-ouput parameters, U is set union, and 'a,'b are the initial values of a and b.

Program specifications define the interface behavior of program operations, so they permit many implementations to share the same specification.

Program nodes contain the code for program operations. They implement the behavior defined by the program specification they satisfy.

The value space for both programs and program specifications is defined by *data type* nodes. In general there will be a spectrum from the most abstract types for use in specifications to the most concrete used by optimized programs. In this paper we use Common Lisp types at the concrete level and algebraic data types for specifications.[3]

For example, a specification of the data type set of natural numbers can be built on top of natural numbers as follows:

```
type set
ops
  phi: → set
  { _ } : natural → set
  _ U _ : set set → set
for all
  S,S1,S2 : set
axioms
  S U (S1 U S2) = (S U S1) U S2
  S U S1 = S1 U S
  S U S = S
  S U phi = S
```

where U is the infix union operator, and other operations such as intersection and set membership have been omitted.

FPD does not have a rigid distinction between specification and implementation. For example, program code may use abstract data types. The ability to mix levels of abstraction freely is especially useful as a means of describing programs in a stage of mid-development with only some types fully concrete.

To achieve real efficiency in the development of large-scale systems requires the use of generic modules, which can be tailored to fit different contexts by the use of parameters.[4]

Property specifications are used to place semantic restrictions on the formal parameters of generic modules. They define the properties that must be satisfied by any argument to be used in an instantiation. Currently modules can only be parameterized by data types. Property specifications are abstract data types that are intended to have a variety of interpretations, like meta-sort theories in Clear.[5]

A trivial requirement would be the existence of a single type element about which nothing is asserted:

```
property spec triv
type element
```

The triv property could be used to specify a generic list data type:

```
type list[ x : triv]
ops
  nil: → list
  cons: x.element list → list
  member: x.element list → Boolean
for all
  l: list e,e':x.element
axioms
  member(e, nil) = F
  member(e, cons(e',l)) = (e = = e') or member(e,l)
```

where == is equality.

A more substantial example of a property specification would be a requirement that a data type satisfies a partial order. This would be achieved by adding the axioms for a transitive, reflexive, and antisymmetric operator.

Design Steps

The arrows between nodes are design steps, which must satisfy proof obligations to ensure the correctness of the design decision.

The creative design step is an *extension* which permits a new node to be added to the design graph to increase functionality. The proof obligation is that the new behavior can be meaningfully constructed without affecting the behavior of the design graph. In the examples above, the program specification and the types have extended the types on which they depend.

At the heart of software development is stepwise refinement of data and control in which problems are solved by successively inventing and concretizing abstractions.[6] The process ceases when an efficient solution is reached.

Data refinements are recorded by *realizations*, which record the implementation of a data type in terms of a more concrete data type. The mapping from the concrete to the abstract type is defined by means of an *abstraction function* and an *invariant*. The invariant is a predicate that characterizes the concrete values that represent valid abstract values. The abstraction function is the possibly partial mapping from concrete to abstract values. The requirement for a realization is that every abstract value can be represented by at least one concrete value and that distinct abstract values do not have overlapping representations.

For example, sets can be implemented by lists. The necessary invariant is that only lists without duplicates represent valid set values.

```
INVARIANT PREDICATE Set_rep : list
forall l:list n: natural
Set_rep(l) iff member(n,l) implies once(n,l)
```

where once returns true if and only if n occurs exactly once in l.

```
ABSTRACTION FUNCTION Set_abs : list → set
forall l:list n: natural
Set_abs(nil) = phi
Set_abs(cons(n,l)) = { n } U Set_abs(l)
```

A control structure refinement is represented by an *encoding* from a program specification to a program node. The requirement is that the program code must satisfy the program specification.

A generic node is *instantiated* by supplying a data type for each formal parameter. The result is a new node in the graph structure. The proof obligation is that the actual parameters satisfy the axioms of the property specifications.

For example, a data type List_of_Nat can be constructed by instantiating the generic list. The only proof obligation in this case, that natural is a type, is trivial.

An Example Development

Fig. 1 draws together the examples developed above and shows how code for a list-based implementation of set of naturals can be developed.

The program specification Nat_Set_as_List realizes Set_of_Nat using the abstraction function and an invariant given earlier. Its pre-post conditions are built on top of List_of_Nat and its specification of the UPDATING-UNION operator is at the concrete level.

```
FORM (UPDATING-UNION a b )
IN b: list
UPDATE a: list
PRE Set_rep(a) and Set_rep(b)
POST forall x: natural
       member(x,a) iff member(x,'a) or member(x,'b)
       and Set_rep(a) and Set_rep(b)
```

The concrete version is correct if the final value of the list a contains just those natural numbers belonging to the initial values of a and b. Additionally, UPDATING-UNION, like all operators at this level, has to preserve the representation invariant.

The program code module Nat_Set_as_List provides Common Lisp for the operations, for example:

```
(defmacro updating-union (a b)
  (setf ,a (set-union ,a ,b))
)
```

where set union is a function that nondestructively appends two lists and removes duplicates.

The proof obligation is that the body of the macro satisfies the pre-post condition.

The function set union uses car, cdr, etc., which are program operations on lists. Consequently Nat_Set_as_List extends the program specification List_of_Nat.

Machine Support

The initial motivation for FPD was as a means of organizing program development. However, it also provides a suitable basis for developing software engineering tools. Two kinds of development are underway at the Bristol Research Centre: library support for design graphs and support for formal specification.

Currently there is a central project module library that provides system version control and source control features for design graph modules. An extension allows the computation of the transitive closure of module dependencies to control recompilation and the checking of proof obligations in the face of updates. In the longer term it is intended to explore the possibility of making deductions about module relationships.

It is vital to provide the machine processing of specifications. It helps eliminate errors in understanding. Machines are meticulous in looking for errors. They help stop the fatal flaw of handwaving around difficult issues. The larger the specification the more vital is the need for machine support.

Machine processing and the expressive power of logic pull in opposite directions. The more powerful a mathematical system, the more difficult it is to provide effective support beyond syntax and type-checking. Real tasks require powerful notations and powerful theorem provers, and for this the necessary technology does not yet exist.

Unfortunately, verification only becomes viable at the level of propositional logic, which is too weak for software specification. FPD compromises by using executable equational data types, which allow specifications to be tested. The test cases are saved with modules for use later in the life cycle as a standard for testing code.

Impact on Life Cycle

The strength of FPD is that it makes software designs explicit. They can therefore be desk checked and subject to review by structured walkthrough. Additionally, the data type specifications that underlie designs can be executed. Executing specifications allows design bugs to be found at design time.[7] The combination of checking in the large (structured walkthroughs) with testing in the small (executable specifications and code) means that very high-quality error-free software can be produced. This has been confirmed in practice.

FPD permits substantial code and design reuse, since every interface has a specification, which may be used for matching a module against a proposed context.

Code optimizations and upgrades take place in the context of a documented design with associated test cases. Maintenance effort is therefore substantially reduced, since it is not necessary to reinvent abstractions to do the work.

Acknowledgments

The authors wish to thank P. Arnold, Dr. C. Dollin, A. Moore, and T. W. Rush of HP Laboratories for their contribution to the development of FPD.

References

1. W.L. Scherlis and D.S. Scott, "First Steps Towards Inferential Programming," *Proceedings of the IFIP Congress*, 1983, pp. 199-212, North-Holland, 1983.
2. C.B Jones, *Software Development - A Rigorous Approach*, Prentice-Hall, 1979.
3. H. Ehrig and B. Mahr, *Fundamentals of Algebraic Specification 1*, EATCS Monographs on Theoretical Computer Science, Vol. 6, Springer-Verlag, 1986.
4. J.A. Goguen, "Reusing and Interconnecting Software Components," *IEEE Computer*, Vol. 19, no. 2, 1986, pp. 16-28.
5. R.M. Burstall and J.A. Goguen, "The Semantics of Clear, a Specification Language," *Proceedings of the 1979 Copenhagen Winter School on Abstract Software Specification*, LNCS Vol. 86, Springer-Verlag, 1980, pp. 292-332.
6. N. Wirth, "Program Development by Stepwise Refinement," *Communications of the ACM*, Vol. 14, pp. 221-7, 1971.
7. C.P. Gerard, D. Coleman, and R.M. Gallimore, "Formal Specification and Design Time Testing," submitted for publication in *IEEE Transactions on Software Engineering*, available as HP Laboratories Report HPL-BRC-TM-86-020.

Hewlett-Packard Company, 3200 Hillview Avenue, Palo Alto, California 94304

Bulk Rate
U.S. Postage
Paid
Hewlett-Packard
Company

HEWLETT-PACKARD JOURNAL

October 1987 Volume 38 • Number 10

Technical Information from the Laboratories of Hewlett-Packard Company
Hewlett-Packard Company, 3200 Hillview Avenue
Palo Alto, California 94304 U.S.A.
Hewlett-Packard Central Mailing Department
P.O. Box 529, Startbaan 16
1180 AM Amstelveen, The Netherlands
Yokogawa-Hewlett-Packard Ltd., Suginami-Ku Tokyo 168 Japan
Hewlett-Packard (Canada) Ltd.
6877 Goreway Drive, Mississauga, Ontario L4V 1M8 Canada

00055601 HPJ 10/87
C BLACKBURN
JOHN HOPKINS UNIV
APPLIED PHYSICS LAB
JOHNS HOPKINS RD
LAUREL, MD 20707

CHANGE OF ADDRESS: To subscribe, change your address, or delete your name from our mailing list, send your request to Hewlett-Packard Journal, 3200 Hillview Avenue, Palo Alto, CA 94304 U.S.A. Include your old address label, if any. Allow 60 days.

5953-8565