

AUGUST 1980

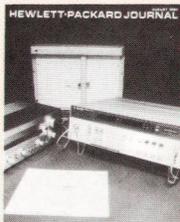
HEWLETT-PACKARD JOURNAL



Contents:

- 3 A Complete Self-Contained Audio Measurement System, by James D. Foote** *This new audio analyzer has everything needed for audio measurements—source, filters, detectors, voltmeter, and counter—all under microprocessor control.*
- 6 Audio Analyzer Applications** *The major areas are general audio testing, transceiver testing, and automatic systems.*
- 8 Making the Most of a Microprocessor-Based Instrument Controller, by Corydon J. Boyan** *In an audio analyzer, microprocessor control means automatic operation, “guaranteed” accurate measurements, and extra features.*
- 10 Design for a Low-Distortion, Fast-Settling Source, by George D. Pontis** *It's based on a state-variable filter with refinements.*
- 12 Floating a Source Output, by George D. Pontis** *The floating output lets the user eliminate ground-loop errors, sum signals, and add dc offsets.*
- 14 A Digitally Tuned Notch Filter, by Chung Y. Lau** *It eliminates the fundamental frequency component of the incoming signal for distortion and noise measurements.*
- 18 A Custom LSI Approach to a Personal Computer, by Todd R. Lynch** *Nine HP-produced large-scale integrated circuits make the HP-85 possible.*
- 23 Handheld Calculator Evaluates Integrals, by William M. Kahan** *Now you can carry in your pocket a powerful numerical integrator like those available on large computers.*

In this Issue:



Audio frequencies are frequencies within the range of human hearing, roughly 20 Hz to 20 kHz. Frequencies on either side of this range are often loosely classified as “audio,” too. Thus, the frequency range of the instrument shown on the cover, Model 8903A Audio Analyzer (page 3), is 20 Hz to 100 kHz. The 8903A is used for testing—among other things—many of the electronic devices that speak to us or play music, such as CB radios and high-fidelity systems. In our cover photo it's shown plotting the frequency response of the stereo amplifier on the left at different signal levels.

There are, of course, other ways of making the measurements in the 8903A's repertoire. What makes the 8903A better? First, it's a complete system that includes a low-distortion signal source, a counter, a voltmeter, and various filters and detectors. Second, all of this is under microprocessor control, automatically stepping through complicated sequences of measurements and computations. Third, it's extremely accurate; for example, it can measure total harmonic distortion (THD) down to 0.003% under normal conditions. Fourth, the 8903A has recorder outputs that make plotting results about as easy as it can be.

If you're not so fortunate as to have studied integral calculus in school, the integral of a function probably isn't the familiar and highly useful concept that it is to scientists and engineers. One way to think of an integral is this: draw the graph of the function as a meandering line on a piece of graph paper. Then the integral of the function is the area bounded by 1) the graph of the function, 2) the horizontal axis of the graph paper, and 3) two vertical lines called the upper and lower limits of integration. Sometimes the integral of the function can be expressed neatly in mathematical terms, but more often than not it can't. So various methods have been devised for estimating integrals using computers. Because most of these numerical integration programs run on very, very large computers, it seems like a small miracle that you can now carry a numerical integrator—a very good one—around in your pocket. Beginning on page 23, its designer tells us about its capabilities and limitations.

On page 18 is an article about the nine special integrated circuits that make the HP-85 Computer possible. This set of custom integrated circuit chips minimizes the cost of the electronics, eases the problem of cooling the computer, makes the small package possible, and provides features that couldn't have been included otherwise.

-R. P. Dolan

A Complete Self-Contained Audio Measurement System

This automatic, autoranging audio analyzer has the signal source, distortion analyzer, and counter to make the measurements most often needed in audio-frequency testing.

by James D. Foote

HEWLETT-PACKARD'S NEW MODEL 8903A Audio Analyzer (Fig. 1) is a complete audio measurement system for quick and accurate characterization of systems and signals in the frequency range 20 Hz to 100 kHz. The starting point for the 8903A is the classical distortion analyzer. Added to this are microprocessor control, a reciprocal frequency counter, rms detectors, and a programmable audio source. These provide accurate measurement of ac level, distortion, SINAD,* signal-to-noise ratio, and dc level. The audio source and the measurement circuits can work independently or together. The source is programmable in frequency and level and has very low distortion. The measurement circuits can monitor this internal source or any other independent input waveform. Together the source and measurement input can be used for swept response measurements.

*SINAD = ratio of signal plus noise plus distortion to noise plus distortion in a receiver's output.

All measurements are available at the push of a button. No knob adjustment or operator interaction is necessary. One simply applies the signal and selects the measurement mode. All control and processing are handled by the internal microprocessor. The microprocessor monitors the input signal and makes internal gain and frequency adjustments as required.

In automatic measurement systems, the 8903A is capable of rapid and straightforward remote control. Analyzer operations can be controlled and all measurements can be transferred via the Hewlett-Packard Interface Bus (HP-IB), Hewlett-Packard's implementation of IEEE Standard 488-1978. On the bench, the 8903A allows rapid and accurate circuit characterization when many repetitive measurements are necessary.

Major application areas for the 8903A Audio Analyzer are general audio testing, transceiver testing, and automatic systems. In general audio testing, the 8903A measures the



Fig. 1. Model 8903A Audio Analyzer makes the accurate measurements needed to characterize systems and signals in its frequency range of 20 Hz to 100 kHz. It has applications in general audio testing, transceiver testing, and automatic systems. Microprocessor control makes it automatic and easy to use.

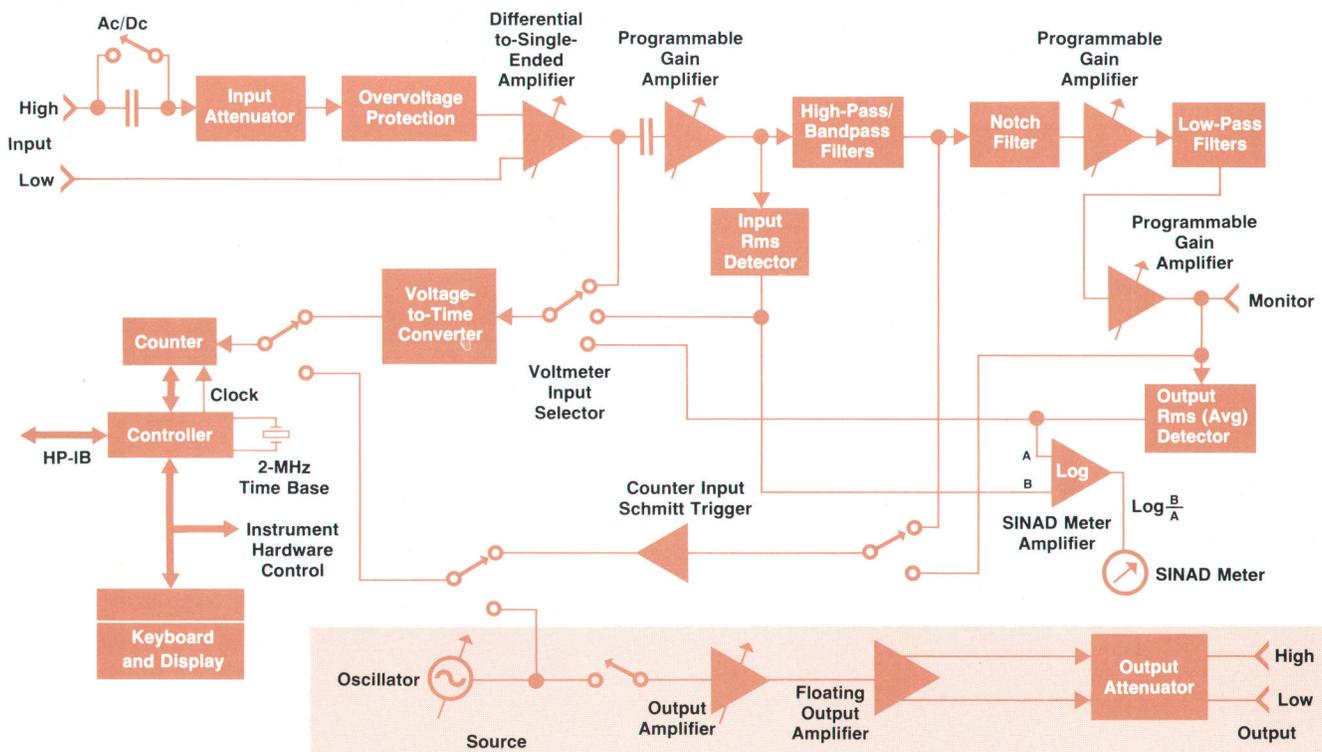


Fig. 2. The 8903A Audio Analyzer is basically a distortion analyzer, with a tunable notch filter to remove the fundamental frequency component of the signal and a detector to measure what remains, which consists of noise and distortion. Added to this are a microprocessor-based controller, a reciprocal frequency counter, a programmable audio source, and other resources, forming a complete audio measurement system.

frequency response and distortion characteristics of filters, high-quality amplifiers, audio integrated circuits, and other devices. The frequency of the internal source can be swept while making measurements in all modes. The analyzer provides recorder outputs and scaling for easy generation of plots using an X-Y recorder.

For transceiver applications the most common receiver measurements are SINAD for FM receivers and signal-to-noise ratio for AM receivers. A psophometric filter is included for making measurements to CEPT standards. Common transmitter measurements such as audio distortion and squelch tones are made using the 8903A with its companion instrument, the 8901A Modulation Analyzer.¹ In automatic systems, the 8903A provides many frequently needed audio functions, doing the work of an audio synthesizer, digital multimeter, frequency counter, and tunable notch filter. More details on specific applications are presented on page 6.

Control Philosophy

Front-panel control of the audio analyzer is simple, yet powerful. Most functions can be used and understood with very little training. The casual user can select amplitude, frequency, measurement mode, and filtering simply by reading the labels on the controls. More details are available on the instrument's pull-out card.

A great deal of measurement sophistication is built into the 8903A software. Measurement routines are structured to optimize measurement speed and accuracy. As a rule,

measurements triggered from the bus or initiated from the keyboard are accurate from the first reading. The operator needn't wait for successive measurements to verify that the reading has stabilized. The software algorithm monitors key voltages in the audio chain and waits until they stabilize before taking data. Not only does the software perform these functions much more rapidly than the operator, but it can also ensure the optimal convergence of the measurement with a repeatable, well defined technique. Distortion, SINAD, and signal-to-noise ratio, in particular, are examples of measurements that in the past required a significant amount of settling time and operator interaction. A classical distortion analyzer requires repeated adjustments to achieve an accurate distortion reading. More recent analyzers have offered semiautomatic tuning and leveling, but response time is often long, and operator interaction is required if the frequency, amplitude or relative distortion of the signal changes significantly. With the 8903A, some delay still exists, but the delay involved is minimized by careful circuit design and microprocessor control.

Special functions extend user control of the instrument beyond that normally available from the front panel. These functions are intended for the user who knows the instrument and the service technician who needs arbitrary control of the instrument functions. Automatic tuning and ranging, overvoltage protection, and error messages protect the user from invalid measurements during normal operation. When special functions are used, some of these

safeguards are removed, depending on the special function selected, and thus there is a degree of risk that the measurement may be invalid. However, there is no risk of damage to the instrument.

To enter a special function, the user enters the special function code (usually a prefix, decimal, and suffix) then presses the SPCL key. The special function code appears on the display as it is being entered. If a mistake is made during entry of the special function code, the user can press the CLEAR key and start over. When a special function is entered, the light in the SPCL key goes on if it is not already on. The readout on the display depends on the special function entered. It may be a measured quantity, an instrument setting, or a special code. In some cases the display is unaltered. Special functions can be entered from the HP-IB by issuing the special function code followed by the code SP.

Floating Input and Output

To eliminate troublesome ground loops, both the source output and the measurement input of the 8903A are floating. This is helpful in low-distortion or low-level ac measurements when it is necessary to reject potential differences between the chassis of the 8903A and the device under test. The 8903A also has EMI (electromagnetic interference) protection built into the source output and measurement input lines so that it can work in the presence of high EMI. All of the analog circuitry is shielded by an internal EMI-tight box. The output and input lines extending from this box to the front panel are shielded and terminated in BNC connectors. For user convenience, the BNC connectors are spaced so that BNC-to-banana adapters can be attached. Thus a banana or twisted-wire connection can be made to the instrument when EMI shielding is not critical.

Analyzer Architecture

The 8903A Audio Analyzer combines three instruments into one: a low-distortion audio source, a general-purpose voltmeter with a tunable notch filter at the input, and a frequency counter. Measurements are managed by the microprocessor-based controller. This combination can make most common measurements on audio circuits automatically. To add to its versatility, the analyzer also has selectable input filters, logarithmic frequency sweep, X and Y outputs for plotting measurement results versus frequency, and HP-IB programmability. Fig. 2 is a simplified block diagram.

The amplitude measurement path flows from the INPUT jacks (HIGH and LOW) to the MONITOR output on the rear panel, and includes the input and output rms detectors, the dc voltmeter (the voltage-to-time converter and counter), and the SINAD meter circuitry. Measurements are made on the difference between the signals at the HIGH jack and the LOW jack. Differential levels can be as high as 300V. Signals that are common to both the HIGH and LOW jacks are balanced out. Signals applied to the LOW jack must not exceed 4V.

The input signal is ac coupled for all measurement modes except dc level. The signal is scaled by the input attenuator to a level of 3V or less. To protect the active circuits, the overvoltage protection circuit quickly disconnects the input amplifier if its input exceeds 15V.

The differential signal is converted to a single-ended signal (referenced to ground) and amplified. The signal is further amplified by a programmable gain amplifier, which is ac coupled. The gain of this amplifier and the differential-to-single-ended amplifier are set to keep the signal level at the input rms detector between 1.7 and 3V rms to optimize its effectiveness and accuracy.

The output from the first programmable gain amplifier is converted to dc by the input rms detector and measured by the dc voltmeter. The output of the detector is used to set the gain of the input circuits and becomes the numerator of the SINAD measurement and the denominator of the distortion measurement. The gain of the input path is determined by measuring the dc level. The input rms detector also monitors the ac component (if there is one) and lowers the gain of the input path if the ac signal will overload the input amplifier. At this point either the 400-Hz high-pass filter or the psophometric filter can be inserted into the signal path. The 400-Hz high-pass filter is often used to suppress line hum or the low-frequency squelch tone used in some mobile receivers. The psophometric filter has a bandpass frequency response that simulates the "average" response of human hearing. It is often used to condition a receiver audio output when determining the receiver's input sensitivity.

During SINAD, distortion, or distortion level measurements, the fundamental of the signal is removed by the notch filter. The output from the filter is the distortion and noise of the signal. In the ac level and signal-to-noise modes the notch filter is bypassed. After amplifying and low-pass filtering, the output from the notch filter is converted to dc by the output rms detector and measured by the dc voltmeter.

During distortion or distortion level measurements, the notch filter is tuned to the frequency counted at its input. Coarse tuning is done by the controller, and internal analog circuitry fine tunes and balances the notch filter. During SINAD measurements, the controller coarse tunes the notch to the source frequency. Thus a SINAD measurement is normally made with the internal source as the stimulus; this permits measurements in the presence of large amounts of noise (where the controller would be unable to determine the input frequency). If an external source is used in the SINAD measurement mode, the source frequency must be within 5% of the frequency of the internal source.

The two programmable gain amplifiers following the notch filter amplify the low-level noise and distortion signals from the notch filter. The overall gain of the two amplifiers is normally set to maintain a signal level of 0.25 to 3V at the output detector and monitor. The 30-kHz and 80-kHz low-pass filters are selected from the keyboard. With no low-pass filtering, the bandwidth of the measurement system is 750 kHz. The filters are most often used to remove the high-frequency noise components in low-frequency distortion and signal-to-noise measurements. The output from the second programmable gain amplifier drives the rear-panel MONITOR output jack. Taking advantage of the increased amplification available at this point, the counter monitors this output in ac level and signal-to-noise modes.

The output rms detector is read by the dc voltmeter in the ac level, SINAD (the denominator), distortion (the numerator), distortion level, and signal-to-noise measure-

Audio Analyzer Applications

The 8903A Audio Analyzer's measurement capabilities reach far beyond conventional distortion analyzers. Much of this performance results from microprocessor control and HP-IB programmability. Numerous hardware features such as a fast counter, both analog meter and digital display, and switchable detector filtering allow the user to make unusual or special measurements with convenience and little auxiliary apparatus.

Consider the 8903A used at a test and calibration station in the manufacture of audio power amplifiers. A typical sequence of events might include an output offset null, frequency response check, distortion test, and noise measurement. The 8903A can perform all these measurements quickly with a single test setup. If an X-Y plotter is connected to the rear-panel outputs, the results of swept frequency measurements can be recorded on standard log/log or log/lin graph paper.

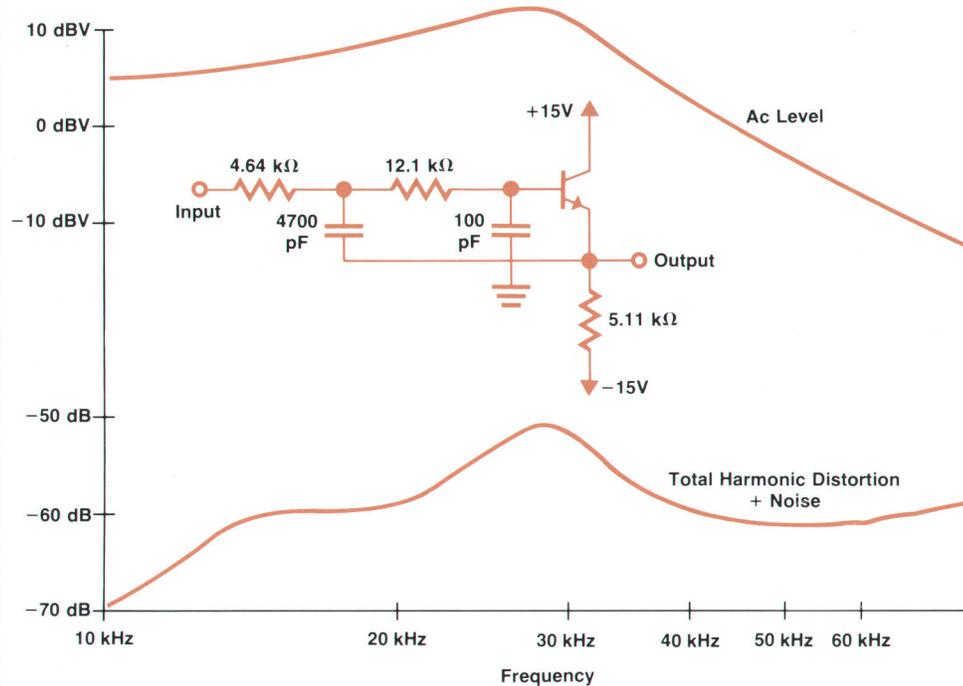


Fig. 1. Swept distortion and frequency response of a two-pole active filter, measured by the 8903A Audio Analyzer.

ment modes. It is also used to set the gain of the two programmable gain amplifiers. The detector can be configured internally to respond to the average absolute value of the signal instead of the true rms value. This option is provided because some measurement specifications for detection of distortion and noise specify the use of an average responding detector. Average responding detectors do not give an accurate indication of signal power unless the signal waveform is known. (If the waveform is Gaussian noise the reading will be approximately 1 dB less than the true rms value.)

In the SINAD mode the outputs from the input and output rms detectors are converted to logarithms, subtracted, and converted to a current by the SINAD meter amplifier to drive the SINAD panel meter. Since SINAD measurements are often made under very noisy conditions, the panel meter makes it easier to average the reading and to discern trends.

The voltage-to-time converter converts the dc inputs into a time interval, which is measured by the counter.

The 8903A uses a reciprocal counter. To measure frequency, it counts the period of one or more cycles of the signal at its input. Then the controller divides the number of cycles by the accumulated count. The reference for the counter is the 2-MHz time base, which also is the clock for the

controller. The counter has four inputs and three modes of operation:

1. Voltage measurement. The time interval from the voltage-to-time converter is counted. The accumulated count is proportional to the dc voltage. For direct measurements (ac level and distortion level), the count is processed directly by the controller and the result is displayed. For ratio measurements (SINAD, distortion, and signal-to-noise), the counts of two successive measurements are processed and displayed. For SINAD and distortion, the controller computes the ratio of the outputs of the input and output rms detectors. For signal-to-noise measurements, the output of the output rms detector is measured with the oscillator on and off and the ratio of the two measurements is computed.
2. Input frequency measurement. The signal from the last programmable gain amplifier or the high-pass/bandpass filters is conditioned by the counter input Schmitt trigger to make it compatible with the counter's input. The period of the signal is then counted, the count is processed by the controller, and the frequency is displayed.
3. Source frequency measurement. The counter measures the frequency of the oscillator during tuning and when

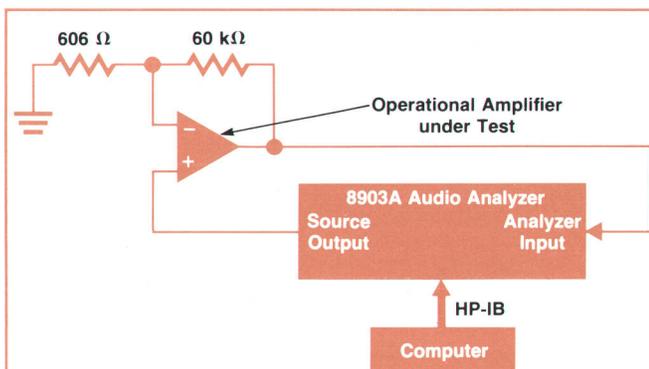


Fig. 2. Test setup for screening operational amplifiers.

The swept frequency measurement with plot capability finds many applications in the laboratory. Fig. 1 shows the swept distortion and frequency response of a two-pole active filter. The upper curve shows the magnitude response, while the lower shows distortion. Notice that the analyzer input magnitude covers a 30-dB dynamic range. During the sweep, the 8903A is automatically setting input gain before performing the distortion measurement.

If an HP-IB controller is available, there are many more applications for the 8903A. Fig. 2 shows a simple test setup for screening operational amplifiers. With no other instruments in the system, a computer-controlled 8903A can rapidly and accurately measure input offset voltage, input noise voltage, and distortion. It can also be used to measure the gain-bandwidth product of the op-amp, provided it is not greater than 30 MHz. The controller test program can be written to provide either a go/no-go output or a listing of measurement results.

Many of the special measurement modes are available through the use of special functions. For example, the 8903A can be used as a test amplifier with gain settable from -24 dB to $+94$ dB. Signal filtering can be added by selecting the appropriate front-panel controls, or the special functions can be used to put the instrument into a notch or bandpass filter mode. Of course, signal frequency and amplitude will be measured and displayed if the operating loads are chosen correctly. This mode of operation makes it possible to count

verifying that the oscillator frequency is within tolerance. This frequency is normally not displayed.

The source covers the frequency range of 20 Hz to 100 kHz. It is tuned by the controller to the frequency entered from the keyboard, using a tune-and-count routine. (Note that the frequency is not obtained by frequency synthesis.) The switch following the oscillator is closed except in the signal-to-noise measurement mode or when an amplitude of 0V is entered from the keyboard.

The source output amplifier and output attenuators provide 77.5 dB of attenuation in 2.5-dB steps, and 2.5 dB of attenuation in 256 steps. This gives an open-circuit output from 0.6 mV to 6V. The floating output amplifier converts the ground-referenced input to a floating output. Either output, HIGH or LOW, can be floated up to ten volts peak.

The entire operation of the instrument is under control of the microprocessor-based controller, which sets up the instrument at turn-on, interprets keyboard entries, executes changes in the mode of operation, continually monitors instrument operation, sends measurement results and errors to the front-panel displays, and interfaces with the HP-IB. Its computing capability is also used to simplify circuit operation. For example, it forms the last stage of the

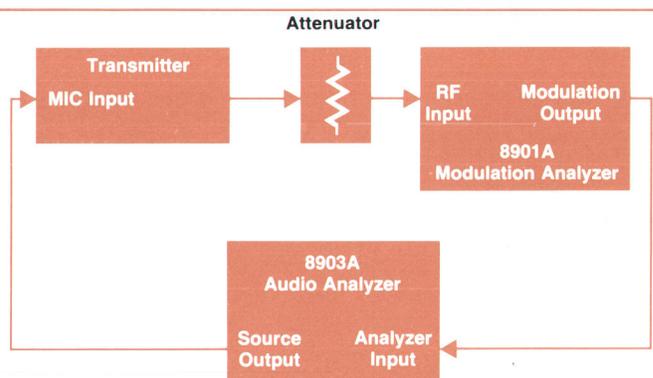


Fig. 3. Transmitter test setup using the 8901A Modulation Analyzer and the 8903A Audio Analyzer.

the frequency of a signal whose amplitude is in the low tens of microvolts.

Transceiver Testing

Much effort has gone into the design of the 8903A to facilitate the audio measurements required for automatic, programmed, transceiver testing. For example, the worst-case source frequency error of 0.3% allows the 8903A source to replace a synthesizer for squelch-tone generation. In addition, a binary programming mode is available through the HP-IB that causes the 8903A to generate a tone burst sequence that can be used to unlock a coded receiver. A related function allows the 8903A to measure burst tones generated by an external source, such as a transmitter under test. The packed, four-byte output allows the 8903A to output frequency measurements as often as every eight milliseconds.

When the 8903A is used in conjunction with the 8901A Modulation Analyzer, almost all transmitter tests can be automated. Fig. 3 shows the block diagram. With the source turned off and the transmitter keyed, the squelch-tone frequency can be counted. Then, the 400-Hz high-pass filter can be switched in to eliminate the squelch tone from the remaining measurements. With the source output turned on, the 8903A can easily be programmed to make the necessary measurements to determine distortion and microphone sensitivity.

counter, converts measurement results into ratios (in % or dB), and so on. It also executes routines for servicing the rest of the instrument as well as itself.

Input Circuits

Numerous design constraints were imposed on the input attenuator/protection/amplifier circuitry. An input impedance of 100 k Ω is necessary to prevent the input circuitry from unnecessarily loading the device under test. On the other hand, maintaining a good signal-to-noise and distortion ratio, good frequency response to 100 kHz, automatic operation and reliable performance with input signals as large as 300V is very challenging. Consider the 300V constraint. It is possible for a 300V signal to appear suddenly at the input while the instrument is measuring a 50-mV input level. Not only must no damage occur, but also the overload recovery must be quick, and the input protection circuit must not be allowed to degrade the input noise floor when measuring 50 mV. The 300V signal may also have spikes or transients that far exceed 300V, or the user may inadvertently apply a larger signal. In all these cases the circuit must recover without causing a safety hazard to the user, destroying internal components, or even blowing a fuse.

Making the Most of a Microprocessor-Based Instrument Controller

by Corydon J. Boyan

In the 8903A Audio Analyzer, most of the tasks that lead to the display of measurements are coordinated by a microprocessor. The processor (an F8 with 18K bytes of ROM and 192 bytes of RAM) counts and tunes the internal audio source, sets the input amplifier's gain, tunes the notch filter, sets the output amplifier's gain, and measures the voltages that will be used to generate each reading. This means that the 8903A can, among other things, automatically take distortion readings without calling upon the user to turn several knobs when seeking a null. This ability alone is ample justification for basing the instrument controller on a microprocessor, but the 8903A goes far beyond this in applying the power of its controller.

"Guaranteed" Accurate Measurements

The 8903A is HP-IB programmable, and this brought some important factors into consideration during the design. The performance of the internal source when settling from one frequency or level to another, for example, is a key factor in assuring the validity of the first measurement taken after changing the frequency or level. Similarly, the settling performance of the input and output amplifiers and the notch filter under various adverse signal conditions becomes very important when the instrument is trying to deliver an accurate first measurement after a change in operating parameters (e.g., after tuning to a new frequency). Add to this the desire to make measurements as rapidly as possible and you have a very interesting problem for a microprocessor to solve.

For example, every time the internal source is tuned, the processor spends about 170 milliseconds tuning to and verifying the frequency. This translates to over 86,000 operations. Of equal complexity is the job of setting the correct gain for the output amplifier and allowing the circuits to settle before the output amplifier voltage is read, with the object of never giving the user an invalid reading while at the same time delivering the reading in as short a time as possible. To accomplish this, the routine that controls the output amplifier makes use of such techniques as slope checking and frequency dependent delays to ensure rapid, valid readings. These techniques make possible accurate readings in half the time it might otherwise take. Referring to the flow chart, Fig. 1, note that the key to this routine is the technique of measuring the rate of change of the signal on the output amplifier (the slope) before attempting to check the signal level to determine if the output amplifier gain is properly set. This is done because it is quite common, after the signal level has suddenly changed (for example, when the notch filter is suddenly switched in

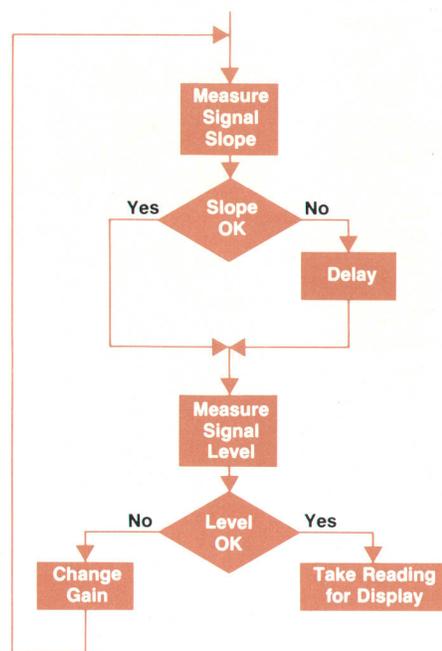


Fig. 1. 8903A leveling algorithm. The microprocessor checks the rate of change of the input signal as well as its level to make certain that the signal has settled before the output amplifier gain is set.

when going from ac level to distortion mode), to have a rapidly falling output amplifier voltage pass through the acceptable region, and thus have the gain appear to be properly set, when in fact the signal is on its way to a level that will require more gain for the proper reading. To keep the controller from being fooled by this phenomenon, we take two voltage readings in rapid succession, and from them calculate the rate of change of the signal. The time over which we measure this slope is longer at lower frequencies, and is in fact looked up in a table in ROM, based on the frequency at which we're operating. If the rate of change is too fast during this period, we delay before checking the level (see Fig. 2). This delay is also frequency dependent and table-driven. Note that if we simply delayed each cycle to keep from risking

Dc inputs and low-level differential amplification for common-mode rejection are necessary features for an instrument like the 8903A. One consequence of automatic ranging is that low-capacitance mechanical switching techniques cannot be used effectively. Needed are high-voltage reed relays, which affect high-frequency performance and require the use of compensation capacitors.

For dc operation the first part of the input signal path is dc coupled with the input blocking capacitor bypassed. The output of the differential-to-single-ended amplifier can then be monitored to obtain an accurately scaled representation of the input dc level.

Input voltages larger than 3V are attenuated by the input attenuator, a network of resistors that divide down the input signal. The appropriate tap point is selected by a reed relay network. If an overload occurs, the maximum attenuation setting is enabled.

To protect the sensitive input amplifier following the attenuator from short-term transients, an overvoltage protection network is used. For low-level signals the transfer impedance is low and signals applied to the input are coupled to the differential-input amplifier. However, for input signals large enough to damage the amplifier the output of the protection network is limited to a safe value.

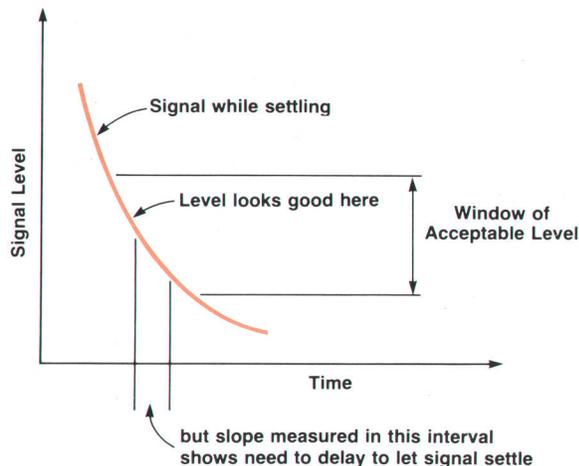


Fig. 2. An example of slope checking by the 8903A microprocessor.

displaying invalid readings, every measurement would take longer, perhaps by several hundred milliseconds. The slope checking gives us a rapid check on signal quality which eliminates the need to delay every time.

After the slope check (and possible delay) we assume the output amplifier is well settled and ready for leveling. We then measure the voltage at its output, and if too high or low, adjust the gain accordingly. After adjusting the gain, we make another pass through the leveling algorithm (unless we are now in the highest-gain range). To keep from getting caught in infinite loops, we require that after the first pass the gain never be reduced, and assume that if it is, we have an unstable signal, which causes us to start the entire measurement cycle (tune source, level input, tune notch, level output) over again. The final protection from infinite loops comes from counting the number of times we restart without displaying (each time we put a "----" pattern on the display), and after 128 times we display Error 31, which also goes out to the HP-IB.

Sweep

Because the 8903A has the ability to generate accurate and comparatively rapid measurements automatically, with no need for the user to insert delays in HP-IB routines or wait for the display to settle before taking a reading, it is possible to have the instrument sweep itself over a range of frequencies without user or computer intervention. Here the calculating power of the microprocessor is brought to bear on the problem of determining the frequency increment for each new point in the logarithmic sweep, based on the sweep range and the number of points per sweep (which can be set by the user). All the user need do to get a series of measurements spanning a frequency range is set the start and stop sweep frequencies and the number of

points per sweep, and press the SWEEP button. This makes it easy to measure the frequency response of an amplifier.

X-Y Recorder Output

The calculating power of the microprocessor-based controller is also apparent in the operation of the 8903A's X-Y recorder outputs. These outputs are driven by digital-to-analog converters controlled by the microprocessor, rather than directly from internal detectors. As a result, the user does not have to worry about the recorder output voltage abruptly changing when the analyzer autoranges. The microprocessor scales the recorder output according to the displayed reading and the plot limits entered via the keyboard. The recorder outputs are always between zero and ten volts so the recorder's zero and vernier controls need be adjusted only once. Thus, a properly scaled plot is easily generated by using the sweep and the X-Y outputs, without any need for an external controller.

Special Functions

Hidden behind the basic measurements are nearly forty special functions, which provide extended measurement capability and many service aids. For example, the analyzer can be given a load resistance in ohms and commanded to display ac level in watts. Another special function changes the number of points per decade in a sweep, and several special functions modify display operation. Normally the left and right displays indicate the frequency and level (or distortion, etc.) of the signal applied to the analyzer input. Sometimes, when using the 8903A just as a source, the user may want the analyzer to display the frequency and level of the source. Special function 10 provides this display.

Service aids provide front-panel display of many internal voltages and settings. Without microprocessor control each special function would require one or more switches on the front panel instead of one SPECIAL key, and would therefore probably not be included. Thus, the processor allows implementation of useful features the user would not otherwise get.

Corydon J. Boyan



Cory Boyan received his BSEE and MSEE degrees from Stanford University in 1974 and 1976. With HP since 1974, he's contributed to the design of the 436A Power Meter, the 8662A Synthesized Signal Generator, the 8901A Modulation Analyzer, and the 8903A Audio Analyzer. He's taught microprocessor design at Foothill College and served as chief engineer of Stanford FM station KZSU. Born in Boston, Massachusetts, Cory now lives in Mountain View, California. His interests include FM radio broadcasting, photography, backpacking, and the art of comedy—he's an avid fan of radio and television comedy groups.

The network consists of two back-to-back diodes which open up under large-signal conditions.

The differential amplifier consists of three high-performance amplifiers. These have the necessary noise, dc offset and frequency performance so that they do not degrade the signal quality. The total input amplifier chain acts as a 4-dB/step amplifier with leveling hysteresis, which maintains the post-amplifier signal level within 6 dB (3 to 1.5V rms). Should the output level change with time and deviate from this range, the gains of the attenuator, the differential-to-single-ended amplifier, and the programmable gain amplifier are adjusted to compensate.

To summarize, the gain of the input amplifiers is modified in three ways. First, the microprocessor monitors the input detector. If the detector voltage is too high or too low, the microprocessor varies the gain of the attenuator/amplifier chain to bring the level within bounds. Second, the overvoltage protection network limits if the input signal exceeds $\pm 15V$. Third, any sustained input overload trips the input overload detector. This detector monitors the input rms detector and the differential amplifier, and if either exceeds a certain absolute voltage limit, the overload detector trips, resetting the gain of the entire amplifier chain to its minimum value (maximum attenuation). This

Design for a Low-Distortion, Fast-Settling Source

by George D. Pontis

To fulfill the requirements of the 8903A Audio Analyzer, the built-in source must have good performance in certain key areas. First of all, for swept measurements, it must be readily programmable and fast settling. It must also have low distortion and noise, and it must have very good amplitude accuracy over the entire frequency range of 20 Hz to 100 kHz.

The combination of these conflicting requirements suggested the use of a high-performance RC oscillator instead of a synthesizer. The synthesizer is easy to program and settles quickly, but it is difficult to build a synthesizer with noise and distortion more than 80 dB below the fundamental. Synthesized designs also do not have sufficient absolute level accuracy or flatness without leveling, and a leveling loop that does not unduly degrade the distortion and settling time would be very difficult to design.

Unfortunately, none of the common RC audio oscillator designs looked suitable either. Usually the amount of distortion is inversely proportional to the settling time. Also, the tuning elements usually float, making the circuit difficult to interface with programming lines.

For these reasons a state-variable oscillator was chosen, similar to that proposed by Smith and Vannerson in 1975.¹ Since this oscillator is built around a state-variable filter structure, inexpensive JFET switches can be used easily to switch the tuning elements. More important is that the ALC design permits very rapid settling without trading off good distortion performance. Although the programmed frequency does not have the accuracy of a synthesizer, sufficient resolution is available to permit firmware tuning to within $\pm 0.3\%$ of the programmed value.

This oscillator design can be described as a state-variable filter in

which the resistor that determines the Q is replaced by an analog multiplier. The control signal for the multiplier is provided by an automatic level control (ALC) circuit. The ALC circuit compares the oscillator amplitude with a stable dc reference obtained from a temperature-compensated reference diode. The resulting error signal is processed through two paths. One path carries the cycle-by-cycle proportional error to the controlling multiplier. This provides very fast settling. The other path includes an integrator in the loop to eliminate nearly all of the steady-state error. This design is theoretically capable of settling the output amplitude within two cycles after small-signal amplitude disturbances.

There are two important refinements in the 8903A oscillator. The first is the use of a special two-stage peak detector. This consists of track/hold and sample/hold amplifiers to eliminate any distortion-causing ripple on the detected peak output. The second refinement is the addition of an ALC loop gain control to compensate the leveling loop, cycle by cycle, for changes in oscillator amplitude. This greatly decreases the large-signal settling time of the oscillator, which is important when switching from one range to another.

Fig. 1 shows the oscillator integrators. The gain constant of these integrators is changed in three-octave steps by selecting the feedback capacitor. This gives us range switching. Coarse tuning within each range is done by switching the input resistors. The eight binary-weighted resistors provide 255 usable steps. Placing the switching devices at the virtual ground point permits the use of JFETs with low drain-to-source on resistance (R_{DS-on}). An individual transistor switch conducts when its gate is allowed to rise to ground potential, and turns off when its gate is pulled to the negative supply, $-15V$.

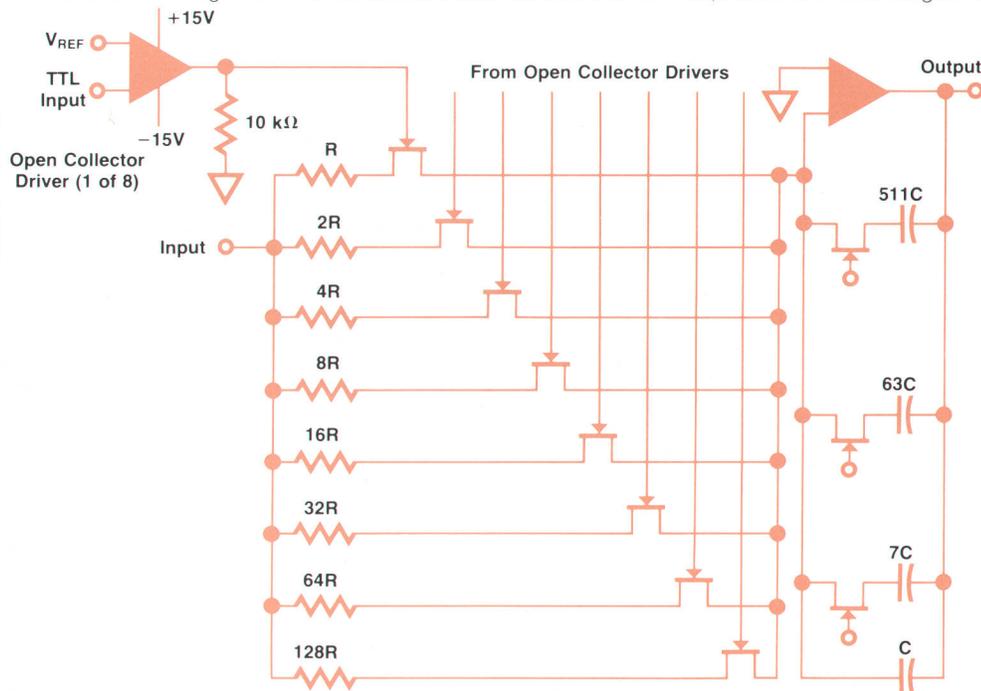


Fig. 1. The gain-switched integrator used in the 8903A's internal oscillator. Capacitors are switched to change ranges. Resistor switching provides coarse tuning within each range.

reset occurs within ten milliseconds if the overload is severe. This protects the input from burning out or blowing a fuse, and allows for rapid overload recovery.

Notch Filter

The notch filter design challenge was twofold. First, it was necessary to design a low-distortion, low-noise filter that was also programmable. Second, to minimize overall

The switching scheme is economical because multiple resistor packages and quad comparators can be used to interface to TTL levels.

Fig. 2 is a block diagram of the oscillator. Integrators U1 and U2 and inverter U4 form the state-variable filter structure. Fine tuning is done by U4. This stage uses a switched resistor network similar to that used for coarse tuning in the integrators. As the resistors are switched the gain of U4 changes, effectively altering the amount of signal transmitted from the output of U2. The gain is proportional to $\sqrt{A-B/R}$, where R is the parallel combination of the selected input resistors and A and B are constants that provide a $\pm 5\%$ fine tuning range. This gives the oscillator enough resolution to tune within $\pm 0.2\%$ of any frequency within the range of the instrument.

It is generally true for sinusoidal oscillators that purity and settling time are chiefly functions of the ALC circuits or mechanisms used. The two-stage peak-detector circuits are the key to the performance of the 8903A oscillator. Oscillator amplitude data is obtained by the track/hold and sample/hold amplifiers in the following manner. Switch S1 closes during the time the output is at its negative peak. Capacitor C1 rapidly charges, following the sine-wave amplitude up to its positive peak. At this time S1 opens, holding the peak voltage on C1. Switch S2 then quickly closes and opens again, thus updating the sampled peak level held by capacitor C2.

The two-stage scheme has several advantages. For one, the first stage may be optimized for fast data acquisition, while the second stage is optimized for long hold time. Fast acquisition is essential for

good amplitude accuracy at high frequencies. Low droop is important to maintain low distortion at low frequencies. Also, this scheme has no steady-state ripple in the sampled output which would cause distortion to appear on the multiplier output and in turn on the main oscillator output.

In practice, the two-stage scheme is also easier to implement than a very fast single-stage sample/hold circuit. The required circuit functions are accomplished with simple JFET switches and unity-gain buffers as shown in Fig. 2.

For sampled data systems in general, settling time is strongly dependent upon loop gain. For this circuit the ideal integrator gain is linearly proportional to frequency. This effect is achieved by switching the integrator resistor on once per cycle for a duration of $35 \mu\text{s}$. This process increases the integral error signal with frequency. The duty cycle increases with frequency until the integrator switch S3 is left closed continuously for frequencies greater than 25 kHz. Above 25 kHz, the oscillator easily settles in less than one millisecond.

8903A oscillator performance is largely limited by the quality of commercially available, reasonably priced analog multipliers. The multiplier used is decoupled slightly (from optimum) to reduce its contribution to THD and noise. This extends the oscillator settling time to a period of four to five cycles.

Reference

1. E. Vannerson and K. Smith, "A Low-Distortion Oscillator with Fast Amplitude Stabilization," *International Journal of Electronics*, Vol. 39, No. 4, 1975, pp. 465-472.

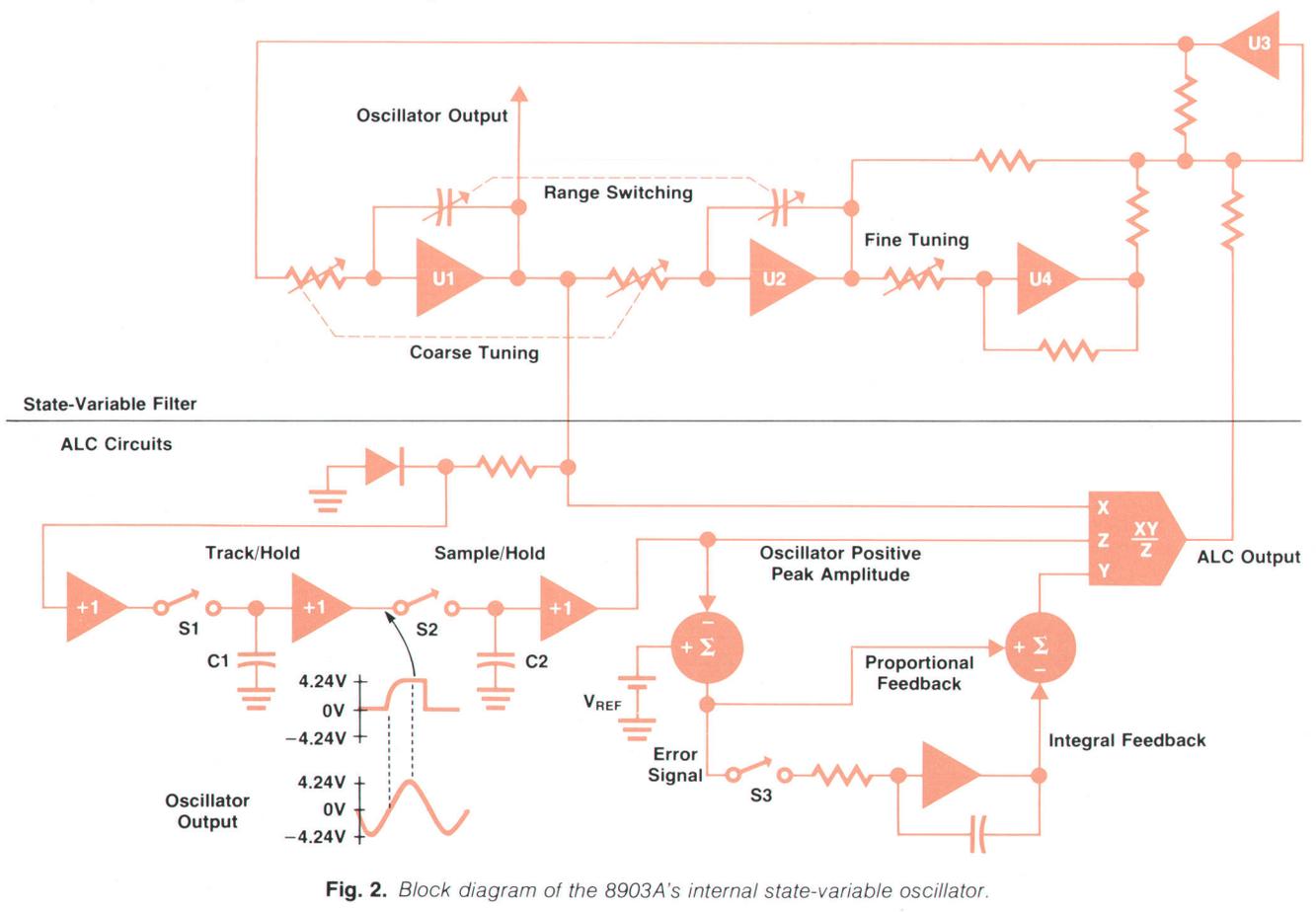


Fig. 2. Block diagram of the 8903A's internal state-variable oscillator.

measurement time, it was necessary to develop an accurate, rapid fine-tune mechanism with quick recovery from overloads and mistuning. To achieve good Q over the frequency range, an active RC filter is necessary. To tune this device, a

variable resistance or conductance device is needed, since achieving the tuning range with variable capacitors or inductors is impractical. Many resistively tunable active configurations are feasible. However, determining the op-

Floating a Source Output

by George D. Pontis

To provide the greatest versatility in both benchtop and systems applications, the 8903A Audio Analyzer's built-in source is floating. This lets the user eliminate ground-loop errors, sum signals, and add dc offsets to the source output.

Previous designs used a separate, isolated power supply for the source circuits. This method is straightforward and offers very good low-frequency common-mode rejection. However, there are several reasons why this arrangement is not used in the 8903A.

The biggest problem with the floating power supply approach is interfacing with the digital programming lines. Since only three of the thirteen attenuator lines have relay isolation and none of the nineteen oscillator lines can be floated, over thirty lines must be coupled in some manner. One solution is to float only the final output stage. This eliminates the need for couplers, but requires a high-performance differential-input amplifier to reject the common-mode signal that appears at the input of the floating stage. Since a floating power supply is still required, the cost of this approach is relatively high.

The 8903A solves this problem with a single-ended-to-differential output converter. This circuit, shown in Fig. 1, operates on the instrument's ground-referenced $\pm 15V$ supplies and requires only two operational amplifiers. A precise combination of negative feedback, positive feedback, and cross-coupling yields a symmetrical differential output with infinite common-mode rejection and a well-defined output impedance.

An analysis of this circuit is generally a tedious procedure because of the number of components involved. However, the high degree of symmetry in the circuit can be exploited to great advantage by using the relations $R2/R1 = R12/R11$, $R3 = R7$, $R6 = R10$, and $R4 = R5 = R8 = R9$. From these relationships, one can derive the expression $R2/R1 = (2R6 + R4)/2R3$, which is a necessary condition for achieving an infinite common-mode output impedance. Then it is easy to calculate the differential output impedance and the open-circuit voltage gain. The resulting equations can be manipulated to find suitable values. For the resistor values used in the 8903A, the associated gain is 1.125 and the output impedance is 480 ohms. The output is further padded with a 120-ohm resistor to yield the desired 600-ohm output impedance.

It would have been possible to use resistors that gave an output impedance of exactly 600 ohms instead of 480 ohms, but this would have required setting up and stocking a supply of several extra odd resistor values. As it is, the circuit is realized using 0.1%, 25-ppm resistors that are also used elsewhere in the instrument.

timum match between the filter configuration and the variable resistive element is not straightforward.

Let's go through the alternatives and the tradeoffs. Switchable resistor networks have good distortion and noise characteristics but do not provide continuous tuning coverage and require extensive switching circuitry. Photoresistors can be driven over a large resistance range and provide continuous tuning. However, the noise and distortion they add to a signal are greater than the required level of 90 dB below the signal level. They can be used as fine-tuning elements if coupled only partially into the circuit. These devices can also be slow and are awkward to control rapidly, reducing the tuning speed. Finally, they tend to vary significantly from device to device and with time and temperature, making compensation difficult.

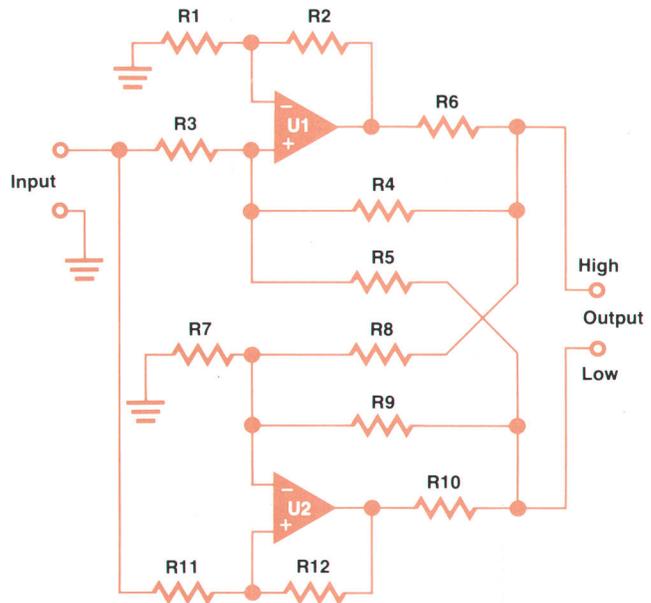


Fig. 1. Single-ended-to-differential output converter provides a floating output for the 8903A's internal oscillator.

The easiest way to see how this circuit works is to eliminate either the inverting (low) or noninverting (high) half of the circuit by shorting the respective output to ground. Fig. 2 shows the reduced circuit when the low half is grounded. If R4 is disconnected, the circuit will have a forward gain of about one, and an output impedance of 276 ohms. R4 works in conjunction with R6 to provide voltage and current feedback that causes the gain and the output impedance to rise.

To demonstrate that the output is truly floating, we ground the input and apply a test source to both outputs. Ideally, the current flow from the test source should be zero. Fig. 3 shows a block diagram and the reduced circuit for this test. Here it can be quickly calculated that the output of U1 will rise just enough over that of the test source to make the current through R6 cancel the current through R4 and R8. Note that the current flowing through sources V1 and V2 is supplied by the other half of the circuit, which is not shown.

Four-quadrant analog multipliers also do not have the 90-dB performance necessary, but they too can be lightly coupled into the circuit for fine tuning. These devices are fast, inexpensive, and easy to drive. There are many variations on this type of circuit, some of which can be obtained in integrated form. Those most suitable use a differential pair of bipolar transistors as a variable gain element by varying the common-mode current.

Light bulbs as variable resistive devices are relatively linear but are slow and have a limited dynamic range. Thermistors, diodes and other nonlinear devices would all be useful only for fine-tune applications. The drive and compensation circuitry for all of these alternatives would be complex and the overall performance marginal.

The tuning elements selected were switchable resistor

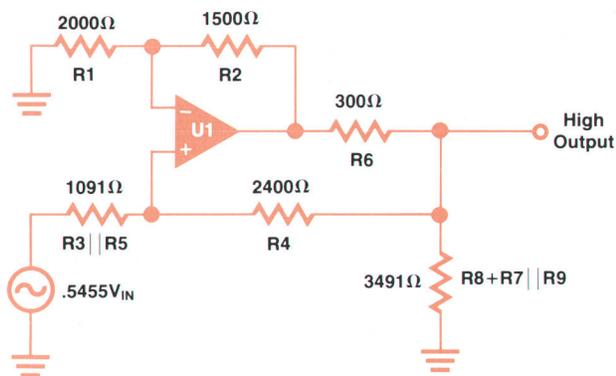
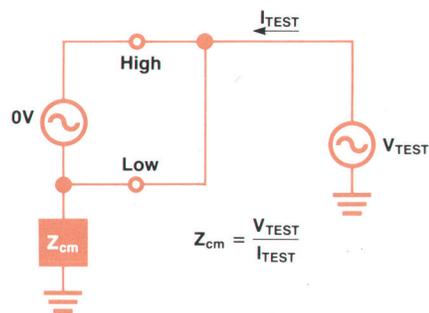


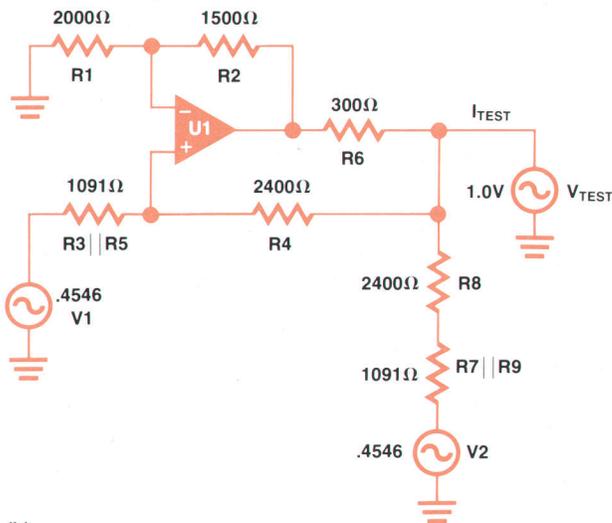
Fig. 2. Eliminating the inverting side of the circuit of Fig. 1 by shorting the low output to ground results in this reduced circuit.

In practice it was found that parasitic effects and the electromagnetic-interference (EMI) filters degrade the circuit balance when the frequency approaches 100 kHz. However, each board is tested for a minimum of 50 dB common-mode rejection at 1 kHz. A typical unit has greater than 40 dB rejection at 100 kHz. Also, a 10-k Ω resistor internally ties the low output to the chassis ground. This provides a reference for the output when no external load is connected. Without this resistor, the common-mode output voltage is indeterminate.

One initial concern about the circuit was difficulty of troubleshooting problems, such as one of the resistors drifting out of tolerance, causing poor common-mode rejection. This problem was solved by implementing the following test procedure. First, the input to the floating amplifier is set to exactly 1.00V rms using the special functions built into the 8903A. Then, the technician shorts the low side to



(a)



(b)

Fig. 3. (a) To demonstrate that the output of the circuit is floating, the input is grounded and a voltage source is applied to both outputs. (b) Reduced circuit for this test (only half of the circuit is shown). Current through R6 cancels the current through R4 and R8. Current through V1 and V2 is supplied by the other half of the circuit (not shown). Thus no current is drawn from the test source.

ground and measures the potentials at several circuit nodes. The measured values can be compared to the calculated values published in a table in the 8903A manual. A second set of measurements can be made, if necessary, with the high side grounded. By observing the deviations between the measured and calculated values, it is easy to locate the faulty component.

George D. Pontis

Born in Los Angeles, California, George Pontis attended the University of California at Santa Barbara, receiving his BSEE degree in 1975. He joined HP in 1978 and contributed to the design of the 8903A Audio Analyzer. Formerly a contributing editor of *Audio Magazine*, he's a member of the Audio Engineering Society. He lives in Palo Alto, California and enjoys outdoor sports, especially tennis, skiing, and swimming.



networks for low-distortion coarse tuning, and a four-quadrant multiplier for fine tuning between the discrete steps of the switchable resistors. Since the four-quadrant multiplier is coupled into the circuit only enough for $\pm 7\%$ tuning, it does not contribute significantly to the overall noise and distortion. A resistor switching network is best implemented if one end of the network is dynamically kept at ground potential; this relieves many constraints on the switching network. To this end, a state-variable* notch configuration is used (see page 14). With this design, low-distortion tuning over a 10:1 frequency range is achieved.

*The name "state-variable filter" is a consequence of the fact that the equations that describe the system can be written in a form that fits the classical state-space description of a linear system, i.e., $\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$, where \mathbf{x} is the response (or state variable) of the system to an input \mathbf{u} .

Capacitors are switched into the network to change frequency in three-octave bands and provide complete coverage of the frequency range of the analyzer.

To complete the fine-tuning path, a synchronous detector mixes the filter output waveform with the fundamental waveform. If any fundamental exists in the notch output, a dc current is generated to fine tune the notch. The critical parameters here are 100-dB dynamic range and rapid operation. A FET double-balanced mixer was selected. The input mixing signal that drives the FET is a square wave, rich in odd harmonics, so the circuit responds to odd-order harmonics as well as to the fundamental. This is the classical solution. A complete null may not be achieved if third, fifth, or higher-order odd harmonics are present. Total error can

A Digitally Tuned Notch Filter

by Chung Y. Lau

The notch filter in the 8903A Audio Analyzer rejects the fundamental frequency component of the incoming signal. This filter consists of a state-variable active filter and fine-tune and fine-balance control circuits. A simplified schematic of the filter minus the control circuitry is shown in Fig. 1.

Conventional notch filters (bridged-T, etc.) used in distortion analyzers are not well suited for digital control because they require expensive relays or analog switches. The state-variable filter approach has the following advantages.

- The Q of the filter is fixed and independent of frequency.
- Tuning is accomplished by switching resistors and capacitors in the two integrators. Inexpensive JFETs are used as switching elements and the switch drivers are simple because both the gate and the channel of the JFET are virtually at ground potential when the switch is on.
- Three filter outputs (low-pass, bandpass, and high-pass) are available simultaneously. The low-pass and high-pass outputs are phase-shifted from the input by 90° when the filter is tuned to the input frequency. These two outputs are used in the control circuits, so extra phase-shifting circuits are not needed.
- Distortion generated in the fine-tune circuitry is filtered by two integrators before appearing at the notch output.

In distortion mode, the filter is tuned in the following manner. The microprocessor counts the input frequency and tunes the notch filter to the same frequency by switching in the proper capacitors and

resistors. When the filter is tuned to the fundamental frequency of the input signal, the bandpass output (V_{BP} in Fig. 1) inverts the fundamental component of the input signal. U5 sums V_{IN} and V_{BP} to cancel the fundamental component of V_{IN} exactly. However, because of the phase and amplitude characteristics of V_{BP} , the harmonics present in V_{IN} are relatively unattenuated.

Analytically, we have

$$\frac{V_{BP}}{V_{IN}}(s) = \frac{-s\omega_0/Q}{s^2 + s\omega_0/Q + \omega_0^2}$$

and

$$\frac{V_{BP} + V_{IN}}{V_{IN}}(s) = \frac{s^2 + \omega_0^2}{s^2 + s\omega_0/Q + \omega_0^2}$$

where ω_0 is the center frequency in rad/s. Therefore, there is zero transmission at $s = \pm j\omega_0$.

The state-variable filter alone can provide only about 15 dB of fundamental rejection because of the coarseness of the frequency tuning (use of discrete resistor values, resistor mismatches, etc.). Fine-tune and fine-balance control circuits are necessary to achieve a notch depth greater than 90 dB. The fine-tune circuit insures that the filter is exactly tuned to the input frequency (no phase error) and the

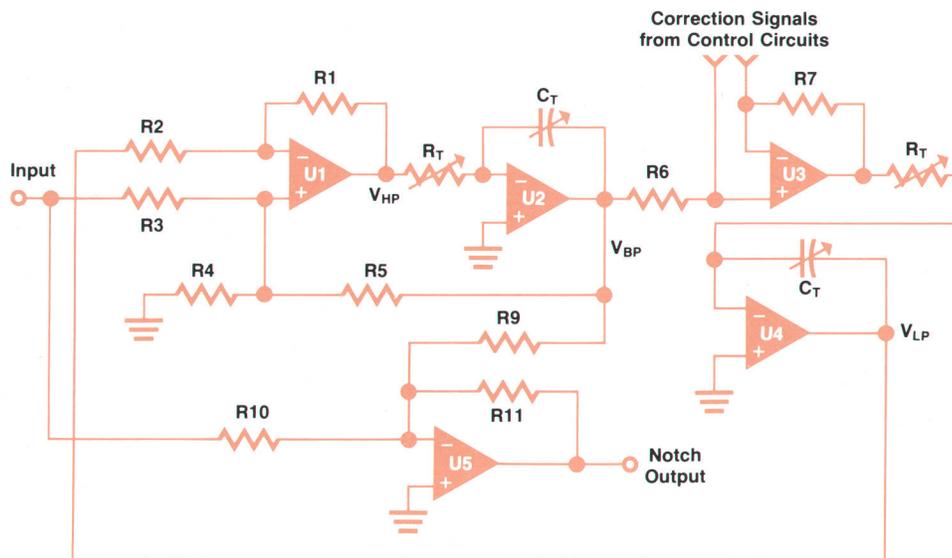


Fig. 1. 8903A notch filter. Correction signals from the control circuits of Fig. 2 provide fine tuning and fine balance.

be as much as 0.46 dB, but this is considered reasonable.

Rapid response is achieved in two ways. First, when an overload occurs, the circuit gain is quickly reduced by the output amplifier circuit, minimizing the transient and thus minimizing the recovery time of the synchronous detector from the overload. Second, the response time of each frequency band has been optimized. Theoretically, it is impossible to fine tune a filter as rapidly at 20 Hz as at 1 kHz. It simply takes a much longer time to detect a null at 20 Hz, but even more important is that the time response of a 20-Hz notch filter is much greater. In fact, a fundamental burst

tone applied at the input of the notch will not immediately be nulled at the output. The entire input will first appear at the output and then decay away in proportion to $f(\text{notch}) \cdot Q$ where Q is the Q of the notch. Thus, as the notch frequency is increased, so is the speed of the fine-tuning circuitry. The bottom frequency band (20-200 Hz) is relatively slow but nulls optimally at 20 Hz. For higher frequencies the instrument response time improves.

Output Amplifier

For the amplifier following the notch filter, gain accuracy

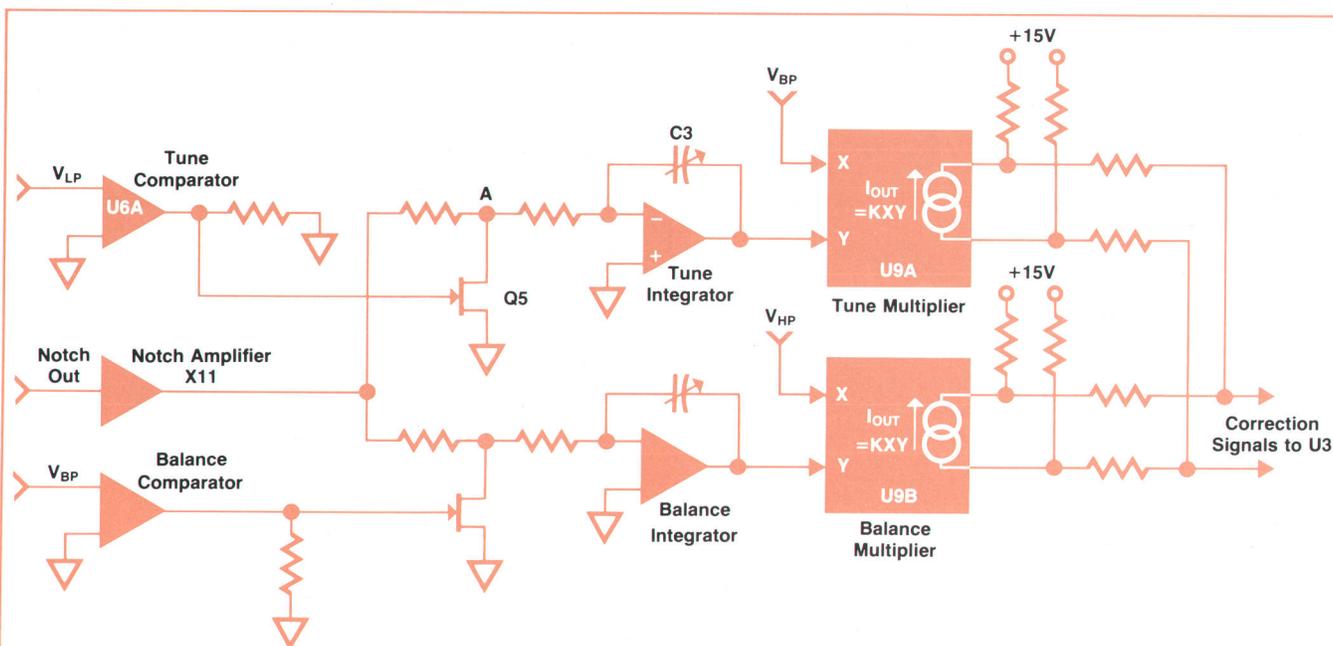


Fig. 2. 8903A notch filter fine tuning and balance circuits.

fine-balance circuit insures that $|V_{BP}/V_{IN}| = 1$ at the input frequency (no amplitude error). The fine-tune circuit is described in some detail here; the fine-balance circuit operates in a similar manner. Fig. 2 is a simplified schematic of both circuits.

The fine-tune circuit operates as follows (see Fig. 2). The low-pass filter voltage V_{LP} , which is phase-shifted 90° from the input, drives the tune comparator U6A, which turns JFET switch Q5 on and off. When Q5 is on, point A is essentially grounded and no current flows into the integrator capacitor C3. When Q5 is off, the notch amplifier feeds the tune integrator. Because of this chopping action, the dc current that flows into the integrator is caused by the notch output component that is synchronous with V_{LP} .*

The tune integrator output is a dc voltage that can be changed only by the dc current flowing into the integrator. This voltage feeds one input of the multiplier U9A. The other input to the multiplier is V_{BP} . The product of the two inputs is summed into the state-variable filter via U3 (see Fig. 1). The net result is that the tune integrator voltage can change the effective resistance (R_T in Fig. 1) and hence the center frequency of the notch filter.

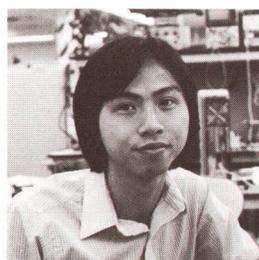
The direction of change is such that any notch output components in synchronism with V_{LP} are reduced. At steady state, the

tune voltage is stable, although not necessarily zero, and no dc current flows into the tune integrator. Therefore, there is no notch output component synchronous with V_{LP} .

The fine-balance circuit insures that there are no components synchronous with V_{BP} in the notch output. V_{BP} and V_{LP} are both at the fundamental input frequency and are in quadrature with each other. Thus, at steady state, there is no fundamental frequency in the notch output.

*Actually, third harmonics in the notch output can also cause dc current to flow in the integrator. This error causes a small amount of the fundamental frequency component to pass through the notch filter. However, the maximum error contribution to distortion measurements from this effect is only about 0.46 dB.

Chung Y. Lau



Chung Lau is a native of Hong Kong. He received his BSEE degree in 1975 and his MSEE degree in 1976 from the University of California at Berkeley. With HP since 1976, he's worked on the 8901A Modulation Analyzer and contributed to the design of the 8903A Audio Analyzer. Chung lives in Cupertino, California, and enjoys bridge and photography.

and good frequency response are important. In ac level measurements the signal travels through the output amplifier chain and is detected by the output rms detector. Any error degrades the 8903A's performance. The output amplifier allows the low-level distortion products leaving the notch to be accurately detected by limiting the required dynamic range of the rms-to-dc converter. The rms detector is accurate over only a 30-dB range, and the output amplifier boosts these signals into the range of the detector. As with the input circuitry, rapid recovery from overload conditions is crucial. If the notch becomes mistuned because of a disturbance at the input, the output suddenly

increases dramatically, sending the amplifier into overload. This in turn generates dc offsets in the amplifier chain that can take seconds to decay, even if the amplifier may start operating again much sooner. Thus a large low-frequency impulse appears at the amplifier output along with the signal being amplified. The composite signal is transferred to the output rms detector, which responds to a time-weighted average of the total. If the impulse is a significant fraction of the signal, the rms detector will not give a true indication of the signal amplitude. This can cause problems when the instrument ranges automatically. In effect, it forces the leveling algorithm to wait much longer to con-

firm that the amplifier output is within the detector's range. This in turn slows down the rate at which the instrument can determine the proper measurement range and display a reading. Even if the impulse is 20 dB less than the signal, the detector error can be as much as 0.5%. This will also increase the amount of time required by the instrument to make an accurate reading from the detector once the proper range has been obtained.

To alleviate these problems, the size of the transient is minimized in three ways. First, operational amplifiers and circuit configurations are used that have better than average overload immunity. Second, an overload detector is placed at the output of the rms detector. If the signal level becomes too large, the overload detector trips and the amplifier gain is reset to 0 dB. Third, a 13-Hz high-pass filter is placed before the output detector. This significantly reduces the duration and amplitude of any transient and hence keeps the transient from significantly increasing the total measurement time. The only delay factors that remain are the controllable and predictable settling times of the notch circuit and the rms detector.

The response time of the output rms detector is a compromise between rapid settling and low-frequency accuracy. A configuration was selected that settles to within 1% of a 10:1 step in 350 milliseconds, and has a steady-state error of 0.2% at 20 Hz. This includes filtering in the detector and additional filtering following the detector to reduce excess ripple. For leveling purposes the ripple is not significant, so the microprocessor uses the detector output when leveling and avoids the extra delay contributed by the additional filter. The output detector and filters could have been designed with switchable time constants to respond more rapidly for higher-frequency signals. However, the penalties would have been additional circuit complexity and the ambiguity of not knowing when to invoke the longer time constant. A 20-kHz signal, for example, might still have a significant low-frequency component, which would cause excessive error with a more rapid time constant.

Oscillator

Many of the design considerations for the notch filter apply equally to the oscillator. In both cases tuning considerations were the same, with switchable resistor networks used as the decade tuning elements and four-quadrant multipliers used for amplitude control. In many ways the oscillator and notch circuits can be seen as duals. The notch generates a pair of zeros on the $j\omega$ axis that reject the fundamental component, while the oscillator generates a pair of poles on the $j\omega$ axis that generate sustained oscillations. The trick in the oscillator is to keep the poles exactly on the axis to maintain constant output amplitude. This must be done continuously by the automatic leveling circuit. If the frequency of the circuit deviates from the desired frequency, the circuit can be fine tuned by the microprocessor, which monitors the output frequency on a sampled basis. The major performance goals of the oscillator were low noise and distortion, rapid amplitude and frequency settling, and digitally programmable frequency control. Again the state-variable filter configuration along with a special leveling circuit offered the flexibility and performance required. The oscillator design is described on page 10.

It was determined during development that the oscillator would have to run at a constant output level to maintain reasonable settling and noise performance. It was also desired to have a floating output. The attenuator and output amplifier circuit (see page 12) takes the oscillator output level and translates it to the selected floating output amplitude. To minimize cost and achieve overall output accuracy goals the attenuation is done in two stages. Coarse amplitude steps are implemented with a 2.5-dB/step attenuator network. Smaller steps are provided by a resistive ladder network that adjusts the amplitude linearly in small discrete steps. The combination can adjust the amplitude within a nominal $\pm 0.15\%$ worst case. Computation of the proper switch settings is an easy job for the computational skills of the microprocessor.

EMI Design

Meeting the required electromagnetic interference (EMI) and susceptibility goals was a bit more challenging than initially expected. Large-amplitude RF fields tend to generate voltages on exposed cabling and circuits. These voltages overdrive many of the active circuits, causing nonlinear operation and distortion. To avoid direct exposure to these fields, the analog circuits are housed in an internal EMI-tight box. The box has an aluminum frame around the sides. The bottom cover is the ground plane of a printed circuit board and the top cover is a removable EMI-tight lid. Removal of the lid, which is held in place by only two screws, makes all the circuits available for service. The microprocessor boards are sufficiently shielded by the instrument cabinet and do not require the extra shielding. To keep the RF fields from developing voltages on the cabling feeding the circuits, special precautions were taken. First, from the inner box to the front panel, shielded cable is used. Second, BNC connectors are provided on the front panel. The BNC connectors allow the attachment of shielded cables directly to the instrument if desired, thus preventing EMI pickup. The instrument's digital circuitry also generates EMI related to harmonics of its 2-MHz clock. This problem was minimized by means of RF gaskets on some of the cabinet seams and by installing an EMI suppressing filter on the power line input. As a result, the instrument will not disturb sensitive receivers operating nearby, and yet will perform well near a powerful transmitter.

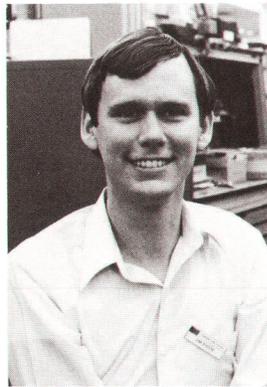
Frequency Measurement

A key feature of the 8903A is its ability to measure frequency automatically, even when the input waveform may have a significant amount of noise and distortion and the amplitude may vary from 6 mV to 300V. Part of this problem is solved because the instrument is autoranging and keeps the leveled waveform within 6 dB over most of the input amplitude range. But before the signal can be accurately counted it must first be converted into a binary signal having the same period as the major frequency component in the waveform, and herein lies a problem. If a zero-crossing circuit is used, noise may cause multiple crossings and a false indication of the frequency. Hysteresis in the detector will help, but if the hysteresis is too large, smaller-amplitude waveforms may not trigger the detector at all while large-amplitude waveforms will have relatively little

hysteresis protection when large noise components are present. To alleviate this problem, the 8903A employs variable hysteresis. As the peak amplitude of the signal varies, so does the hysteresis level, which is maintained at approximately one-half the positive peak for the positive portion of the waveform and approximately one-half the negative peak for the negative portion of the waveform. Hence noise immunity remains constant regardless of the incoming waveform. Hysteresis is implemented with a bipolar peak detector followed by a dual comparator. The waveform is transferred to a reciprocal counter, which measures the period of the signal, and the microprocessor inverts this period to get frequency.

Acknowledgments

Many people throughout HP contributed to the successful introduction of the 8903A. It must be stressed that the instrument's success is the result of the total contribution of many people from early investigation through production. So first and foremost may I thank all of those who contributed their time, enthusiasm, and support. On the R&D team I would like to thank Allen Edwards for contributing to the original project concept and leading the project through early development, Chung Lau for his overall technical support and especially for his efforts in developing the notch and input circuits and verifying overall instrument performance, Cory Boyan for the initial oscillator circuit investigation, software development and coordination, and digital circuit development, Bob DeVries for product design, Derrick Kikuchi for overall software development and latch board development, George Pontis for developing the oscillator, attenuator, output amplifier and power supply circuits as well as various special test fixtures to verify



James D. Foote

A native of Madison, Wisconsin, Jim Foote earned his BSEE degree at the University of Wisconsin in 1972 and joined HP in 1973. In 1975 he obtained his MSEE at Stanford University. He has served as a design engineer on both the 8901A Modulation Analyzer and the 8903A Audio Analyzer, and as project manager for the 8903A. Jim has just joined HP's Disc Memory Division and has moved to Boise, Idaho with his wife and daughter. Among his interests are reading, walking, racquetball, skiing, chess, and doing odd jobs around the house.

instrument performance, Peter Lee and Jim Stewart for industrial design, and Bruce Creedy for initial product design. Special thanks also to Ray Shannon and Jim Stinehelfer who were instrumental in the early product definition. Other key contributors include Rick Pinger and Jim Harmon in providing service and operating documentation, Bob Stern and Bob Rands in product marketing support, Bob Cirner and Ken McFarland in parts scheduling and procurement, Greg Hoberg, Bob Shatara, Phillis Nakano, Dana Kreitter, and Rich Mills for production support, and Charlie Sallberg and Chuck Clavell for reliability engineering test and support.

Reference

1. A.P. Edwards, "Precise, Convenient Analysis of Modulated Signals," Hewlett-Packard Journal, November 1979.

SPECIFICATIONS

HP Model 8903A Audio Analyzer

SOURCE

FREQUENCY RANGE: 20 Hz to 100 kHz.
 FREQUENCY RESOLUTION: 0.3%.
 FREQUENCY ACCURACY: 0.3% of setting.
 OUTPUT LEVEL RANGE: 0.6 mV to 6V open circuit.
 OUTPUT LEVEL RESOLUTION: 0.3% or better.
 OUTPUT LEVEL ACCURACY (open circuit): 2% of setting.
 60 mV to 6 V, 20 Hz to 50 kHz; 3% of setting, 6 mV to 6 V, 20 Hz to 100 kHz; 5% of setting, 0.6 mV to 6 mV, 20 Hz to 100 kHz.
 FLATNESS (1 kHz reference): ±0.7%, 20 Hz to 20 kHz; ±2.5%, 20 Hz to 100 kHz.
 DISTORTION & NOISE: The higher of: -80 dB or 30 μV, 20 Hz to 20 kHz, 80 kHz BW, -70 dB or 95 μV, 20 kHz to 50 kHz, 500 kHz BW, -65 dB or 169 μV, 50 kHz to 100 kHz, 500 kHz BW.
 IMPEDANCE: 600Ω ± 1%.
 SWEEP MODE: Logarithmic sweep with up to 500 points/decade or 255 points between entered start and stop frequencies, whichever is smaller.

AC LEVEL

FULL RANGE DISPLAY: 300.0V, 30.00V, 3.000V, 300.0V, 30.00mV, 3.000mV, 300.0mV, 30.00mV, 3.000mV, 300.0mV, 30.00mV, 3.000mV.
 OVERRANGE: 33% except on 300V range.
 ACCURACY: ±2% of reading 30 V to 300V, 20 Hz to 1 kHz; ±2% of reading, 50 mV to 30V, 20 Hz to 20 kHz; ±4% of reading, 0.3 mV to 30V, 20 Hz to 100 kHz.
 AC CONVERTER: True-rms responding for signals with crest factor <3 and harmonics up to 80 kHz typical. 3 dB measurement BW: >500 kHz typical.

DC LEVEL

FULL RANGE DISPLAY: 300.0V, 48.00V, 16.00V, 4.000V.
 OVERRANGE: 33% except on 300V range.
 ACCURACY: ±0.75% of reading (400 mV to 300 V); ±3 mV (<400 mV).

SINAD¹

FUNDAMENTAL FREQUENCY RANGE: 20 Hz to 100 kHz.
 DISPLAY RANGE: 0 dB to 99.99 dB.²
 ACCURACY: ±1 dB, 20 Hz to 20 kHz; ±2 dB, 20 Hz to 100 kHz.
 INPUT VOLTAGE RANGE: 50 mV to 300 V.
 DETECTION: True rms (average detection selectable by internal switch).
 RESOLUTION: 0.01 dB for SINAD ratios >25. For ratios <25 the display is rounded to the nearest half dB to reduce digit flickering of noise signals. (Full resolution is available by defeating this feature using special function 16.1).

ANALOG METER: Active in SINAD only and for SINAD ratios <18 dB (or 24 dB using special function 7.1).

TUNING: notch filter is tuned to analyzer source frequency.

DISTORTION

FUNDAMENTAL FREQUENCY RANGE: 20 Hz to 100 kHz.
 DISPLAY RANGE: 0.001% to 100% (-99.99 dB to 0 dB)
 ACCURACY: ±1 dB, 20 Hz to 20 kHz; ±2 dB, 20 kHz to 100 kHz.
 INPUT VOLTAGE RANGE: 50 mV to 300 V.
 RESIDUAL NOISE AND DISTORTION: The higher of 0.01% (-80 dB) or 30 μV, 20 Hz to 20 kHz, 80 kHz BW, 0.03% (-70 dB) or 95 μV, 20 kHz to 50 kHz, 500 kHz BW, 0.056% (-65 dB) or 169 μV, 50 kHz to 100 kHz, 500 kHz BW.
 DISPLAYED RESOLUTION: 0.0001%, <0.1% distortion, 0.001%, 0.1% to 3% distortion, 0.01%, 3% to 30% distortion, 0.1%, >30% distortion.
 DETECTION: True rms.

SIGNAL TO NOISE

FREQUENCY RANGE: 20 Hz to 100 kHz.
 DISPLAY RANGE: 0 dB to 99.99 dB.
 ACCURACY: ±1 dB.
 INPUT VOLTAGE RANGE: 50 mV to 300V.
 RESIDUAL NOISE: The higher of -80 dB or 30 μV, 80 kHz BW, -70 dB or 95 μV, 500 kHz BW.
 RESOLUTION: Same as SINAD.
 OPERATION: The analyzer displays the ratio of the input voltages as the internal source is automatically switched on and off.

FREQUENCY COUNTER

RANGE: 20 Hz to 150 kHz.³
 RESOLUTION: 5 digits.⁴
 ACCURACY: 0.004% ± 1 digit.
 INPUT SENSITIVITY: 50 mV in distortion and SINAD modes, 5.0 mV in ac level and signoise modes.
 COUNTING TECHNIQUE: Reciprocal with 2 MHz timebase.

AUDIO FILTERS

400-Hz HIGH-PASS FILTER: 3 dB cutoff frequency 400 Hz ± 40 Hz; Roll-off: 140 dB/decade
 PSOPHOMETRIC FILTER: CCITT Recommendation P53. Deviation from ideal response: ±0.2 dB at 800 Hz; ±1 dB, 300 Hz to 3 kHz; ±2 dB, 50 Hz to 3.5 kHz; ±3 dB, 3.5 kHz to 5 kHz.
 30-Hz LOW-PASS FILTER: 3 dB cutoff frequency, 30 kHz ± 2 kHz; Roll-off: 60 dB/decade.
 80 kHz LOW-PASS FILTER: 3 dB cutoff frequency, 80 kHz ± 4 kHz; Roll-off: 60 dB/decade.

REAR PANEL INPUTS/OUTPUTS

RECORDER OUTPUT: X: 0-10 Vdc typical corresponding to log of oscillator frequency. Y: 0-10 Vdc typical corresponding to displayed value and entered plot limits.
 RECORDER OUTPUT RESISTANCE: 1 kΩ nominal.
 MONITOR OUTPUT: In ac level mode provides scaled output of measured input signal. In SINAD, distortion, and distortion level modes provides scaled output of input signal with the fundamental removed.

General

INPUT IMPEDANCE: 100kΩ ± 1% shunted by <300 pF with low terminal grounded.^{5,6}
 CMRR: @ 60 Hz: 60 dB for differential input <2V, 36 dB for differential input <48V, 30 dB for differential input <48V.
 REMOTE OPERATION: HP-IB, all functions except the line switch, low terminal ground switches, and the ×10-10 increment keys.
 TEMPERATURE: Operation 0° to 55°C; storage -55°C to 75°C.
 POWER REQUIREMENTS: 100, 120, 220, or 240 volts ± 5%, -10%; 48-66 Hz, 100 or 120V ± 5%, -10%; 48-440 Hz, 100 VA max.
 WEIGHT: Net 12.3 kg (27 lb).
 SIZE: 146 H × 425 W × 440 D mm (5.75 × 16.8 × 17.3 in)
 EMI: Conducted and radiated interference is within the requirements of methods GE03 and RE02 of MIL STD 461A, VDE 0871, and CISPR publication 11. Conducted and radiated susceptibility meet the requirements of methods CS01, CS02, and RS03 (1 volt/meter) of MIL STD 461A dated 1968.
 PRICES IN U.S.A.:
 8903A Audio Analyzer \$5800.
 Option 001: Rear panel connections instead of front panel for source output and analyzer input, \$50.
 MANUFACTURING DIVISION: STANFORD PARK DIVISION
 1501 Page Mill Road
 Palo Alto, California 94304 U.S.A.

1. SINAD is a sensitivity measurement computed from the ratio of signal plus noise and distortion to noise and distortion.
 2. Residual noise and distortion same as for distortion.
 3. 20 Hz to 100 kHz in SINAD and distortion.
 4. Resolution is limited to 0.01 Hz for input frequencies <100 Hz.
 5. In ac level mode input resistance is 101kΩ ± 1%.
 6. Input capacitance is 330 pF for Option 001.

A Custom LSI Approach to a Personal Computer

by Todd R. Lynch

THE NEW HP-85 PERSONAL COMPUTER, which was featured in last month's issue, is a system totally integrated into a single package. Included in this system are a CRT, printer, tape drive, and keyboard. To control the I/O (input/output) devices and to interface to ROM (read-only memory) and RAM (random-access memory), the design uses LSI (large-scale integrated) circuits designed and fabricated by Hewlett-Packard.

The design of custom chips minimizes the cost of the electronics. Also, the power dissipation is reduced to a level that permits the use of air-convection cooling, eliminating the need for a fan. The LSI designs save large amounts of printed-circuit-board space, making a small system package possible. By designing LSI circuits dedicated to each mechanical subassembly, features can be added to the overall system that would be nearly impossible to incorporate with discrete logic designs.

HP-85 LSI Chip Set

There are nine custom LSI circuits in the HP-85. These circuits are interconnected as shown in Fig. 1. Eight of the LSI circuit designs use NMOS technology. The ninth circuit uses a bipolar technology with two layers of metallization.

The heavy emphasis on LSI design makes the HP-85 very compact. The electronics in the machine consists of these circuits plus a power supply, clock generator, CRT raster scan circuitry, and various motor and printhead drive circuits.

The system is partitioned as follows.

- CPU (central processing unit). The CPU commands the bus control lines and interfaces directly to the system

operating commands (firmware) stored in the four ROMs.

- ROMs. Each ROM chip has the same basic custom design, differing only in the bit pattern permanently stored on each chip.
- Read/Write Memory. The user memory consists of eight commercially-available dynamic $16K \times 1$ RAMs.
- RAM Controller. This chip is designed to interface between the CPU and the RAMs.
- Buffer. This chip is designed to provide the capability for system expansion by adding plug-in units in the rear of the machine.
- Keyboard Controller.
- Printer Controller.
- CRT Controller.
- Display Memory. Four commercial dynamic $16K \times 1$ RAMs are used to store data for the display. Approximately one-fourth of this memory is used for the alphanumeric data and the remainder is used for graphic data.
- Cartridge Controller.
- Sense Amplifier. This bipolar chip interfaces the tape cartridge controller to the magnetic tape head.

Table I is a summary of the custom LSI chip sizes, number of pins (wiring connection points), and power dissipation.

The machine's sixteen-bit address allows direct access to 65,536 bytes of information. The memory map in Fig. 2 shows how this space is allocated. Note that 32K bytes of the address space are devoted to ROM while 16K bytes are for RAM. Additional RAM and ROM capacity may be added to the system. The I/O address space occupies the upper 256 bytes of memory. Each I/O device controller has from two to four dedicated addresses assigned to it.

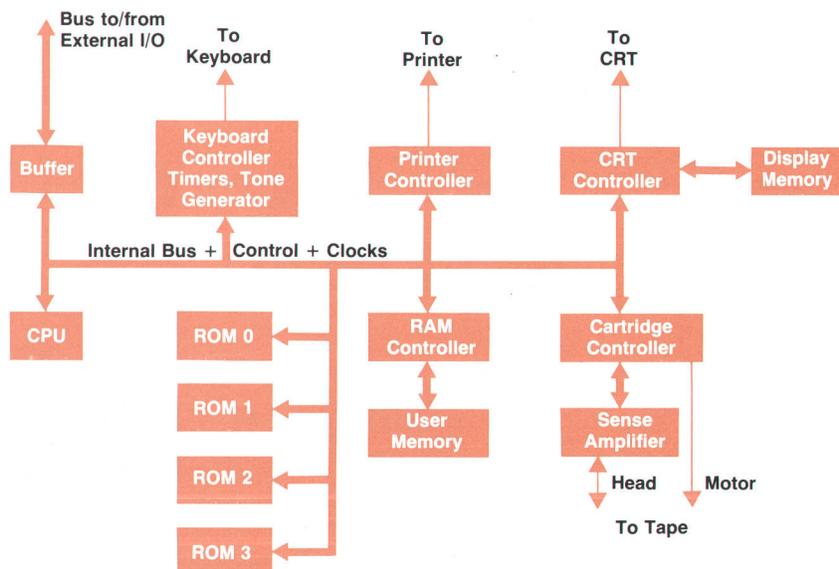


Fig. 1. The HP-85 system block diagram contains nine custom LSI circuit designs. The user and display memories are the only parts in the diagram that use commercially available circuits. The development of custom circuits enabled the system to be contained in a compact package at low cost and without the need for a cooling fan.

Table I

Custom LSI circuit characteristics. All of the circuits are fabricated with NMOS silicon gate technology except for the sense amplifier which is made with dual-layer-metallization bipolar technology.

Circuit:	Size (mm)	#Pins	Power (mW)
CPU	4.93×4.01	28	330
ROM	4.75×5.41	28	200
RAM Controller	2.67×3.56	40	220
Buffer	2.54×3.35	40	340
Keyboard Controller	3.78×4.39	42	200
Printer Controller	4.78×5.44	40	300
CRT Controller	4.14×5.51	40	200
Cartridge Controller	3.63×3.86	28	55
Sense Amplifier	1.50×1.55	16	150

CPU Design

Several commercially available CPUs were considered at the beginning of the project, but none could provide all the features needed to efficiently implement a powerful scientific BASIC language machine. BASIC requires high-precision arithmetic, which is best accomplished with decimal rather than binary numbers. Many different stacks are needed to parse (separate a statement into executable steps) and execute the language. The ability to handle variable-length data is required for variable-length tokens (bit sequences from one to several bytes in length), and multilevel vectored interrupts are needed to handle the I/O devices in real time. The design of the custom NMOS CPU incorporates all of these requirements plus many more.

Fig. 3 shows a block diagram of the HP-85 CPU. Major blocks are the 7000-bit PLA (programmable logic array), 64-byte register bank, and eight-bit ALU (arithmetic logic unit) and shifter. Each CPU instruction is decoded by a microprogram in the PLA that directs the rest of the chip to perform the desired function.

A very powerful feature of the CPU, which is incorporated into the PLA microprogram, is the ability to handle

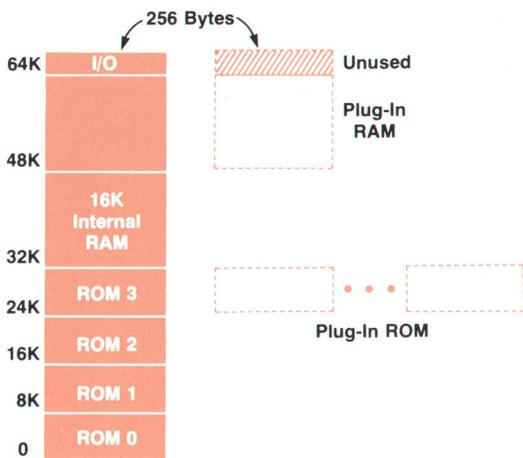


Fig. 2. The HP-85 memory map allocates four 8K-byte system ROMs and one 16K-byte internal RAM. The RAM is used for user memory and 256 dedicated I/O locations. By using bank selection, up to six more 8K-byte ROMs may be added. The RAM may be expanded by 16K bytes if a plug-in module is added to the system.

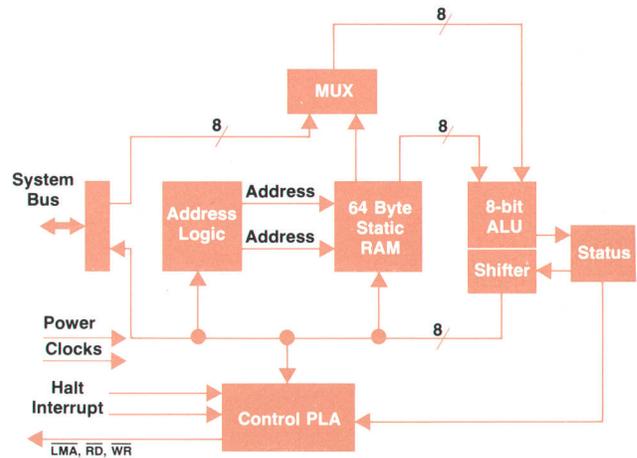


Fig. 3. The CPU in the HP-85 is a custom NMOS circuit design. A 7K-bit programmable logic array (PLA) controls the interactions between the register bank, the ALU, and the rest of the HP-85 system.

data ranging from one to eight bytes in length. This multi-byte feature lets the programmer, for example, add two eight-byte mantissas, increment a sixteen-bit address, or load into the CPU a three-byte token from memory, each with a single instruction. With an off-the-shelf CPU, this would require setup and iteration within a software loop.

The eight-bit ALU is capable of shifting and can do both decimal and binary arithmetic. The programmer sets a mode bit to specify the type of shifting or arithmetic desired. This, combined with the multi-byte feature, lets the programmer easily work with signed, floating-point mantissas up to sixteen digits in length or two's-complement binary integers up to sixty-four bits in length.

Of the 64 eight-bit registers contained in the CPU, one pair is dedicated to the program counter, one pair to the stack pointer, and one pair to internal index calculations. The rest are general-purpose registers. One advantage of having a register as the program counter is that it can easily serve as one of the operands for any CPU instruction. If it is modified by an instruction, then an immediate jump to the new location occurs. The stack pointer points to the subroutine return address stack stored in memory. Additional stacks can be created with any other consecutive pair of registers in the CPU. Thus, the programmer can maintain many different stacks at any one time. Sixteen instructions are dedicated to manipulating data on the currently designated stack. Any pair of consecutive registers can also be used as a sixteen-bit index register. This lets the programmer index into several data arrays at the same time.

To handle real-time I/O devices, the CPU has multilevel vectored interrupt handling ability. Up to 127 interrupt vectors can be accommodated by the architecture. The CPU can also be halted by an external device. This lets that device control the system bus, so it can have direct memory access at high speeds.

No accumulator is present in the CPU. This is made possible by the design of the 64-byte register bank. The registers are constructed as a two-read/one-write memory. This means that, at a given time, any two bytes can be read

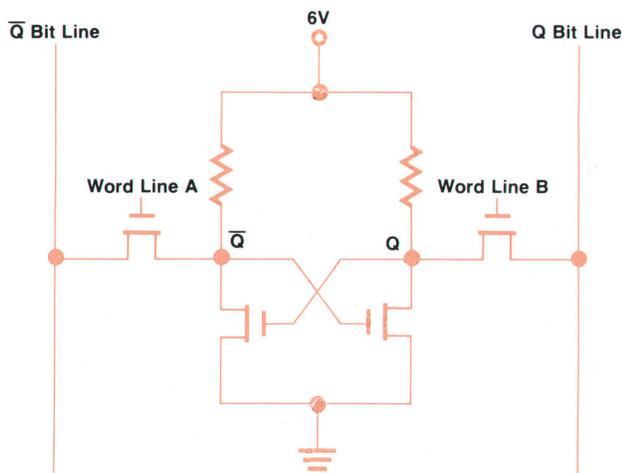


Fig. 4. Independent control of each word line is used to achieve a cell that can be read out onto either bit line. The RAM made up with these cells can be read two words at a time.

from the memory and operated on in the ALU. The result is returned to one of the accessed byte locations. This sequence takes one processor cycle (1.6 μ s).

To design a two-read/one-write memory, a standard static RAM cell with a special variation is used (see Fig. 4). In a normal static RAM cell, a common word line enables both transmission gates between the bit lines and the cell. In a two-read cell, there must be two different word lines. Word line A enables complement information from one cell to be read out onto the \bar{Q} -bit line while word line B enables true information from another or the same cell to be read out on the Q-bit line.

The circuit design of such a cell must be done carefully. Since the bit lines are precharged before reading, it is possible that by enabling a single word line, a cell could be made to flip (change state) rather than be read. Inadvertent flipping is prevented by using a low voltage for the word line and a proper size ratio between the cell's pull-down transistor and the transmission gate.

The static RAM cell's pull-up device also received close scrutiny during the design phase. A cell with a low-power depletion-load pull-up requires an area larger than 6400 square micrometres (10 square mils) and a quiescent power of 120 microwatts. By using a polysilicon pull-up resistor, the RAM cell area is reduced by 40% and the power is reduced by 80%. The savings in area amounts to a reduction in size of more than 0.25 mm on each side of the die. The reduction in total power consumption is fifty milliwatts.

To obtain these reductions, a polysilicon resistor process was developed for the CPU. An additional masking step is required to define the regions on the chip where the polysilicon layer is lightly doped, hence creating the polysilicon resistors. The doping level was chosen for a sheet resistance of $10^7 \Omega/\square$. An undoped layer could not be used because the sheet resistance was so high that the junction leakage currents at elevated temperatures resulted in unwelcome voltage drops across some resistors. It was also discovered that if the contact mask overlapped the polysilicon resistor area, the aluminum metallization could spread into the lightly doped resistor and lower its resistance.

System Control and Timing

Eight bus and three control lines leave the CPU. The eight-bit bus is used to time-multiplex a sixteen-bit address, instructions, and multibyte data quantities. The three control lines— \overline{LMA} (load memory address), \overline{RD} (read) and \overline{WR} (write)—indicate to the system circuits what type of information is on the bus.

When \overline{LMA} is low, all chips in the system know that one byte of a two-byte address will be placed on the bus. The \overline{LMA} signals always come in pairs since all addresses are sixteen bits. Each chip in the system reads this address and compares it to the address range to which the chip is supposed to respond. Setting \overline{RD} low indicates that the CPU wants to read the contents of the address most recently sent out. When \overline{WR} goes low, the CPU wants to write into the location most recently addressed.

The multibyte feature is accomplished by sending out an address with two \overline{LMA} signals followed by one to eight \overline{RD} or \overline{WR} signals. The circuit being read from, or written to, is expected to increment its memory address register every time it sees a \overline{RD} or \overline{WR} . The CPU sometimes can fetch consecutive instructions from memory by simply sending out additional \overline{RD} commands. This speeds up the instruction rate of the machine since sending out an address for every instruction and piece of data is not necessary.

Four nonoverlapping phases (Fig. 5) with 200 ns width and 200 ns spacing clock the circuits in the HP-85. In one cycle the system can access a preaddressed memory location, read it into the CPU, add it to a CPU register and store the result in the CPU register. An eight-byte add requires eight cycles plus the time to fetch the command, which is usually three cycles.

Before any chip writes on the bus, the bus gets precharged to a logic one. Therefore, the circuit desiring to put a one on the bus must merely continue to hold the bus high. If it wants a zero, it must discharge the bus. In NMOS technology it is easier to discharge a bus than to bring it high. Consequently, the circuits were designed with large devices to discharge the bus and relatively small devices to maintain a high level on the bus. This minimizes the chip area required for each circuit's driver logic.

The system bus has fewer spurious transitions using the

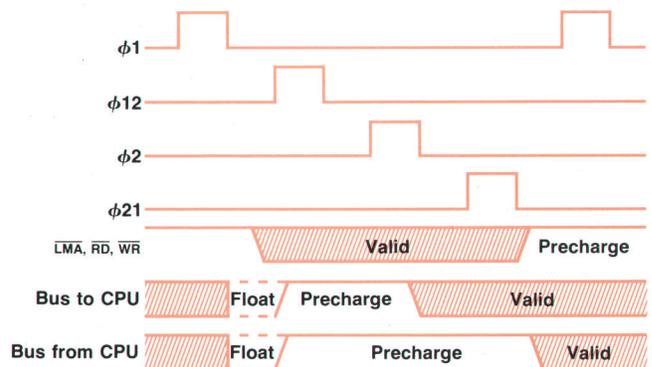


Fig. 5. HP-85 system timing and control. The control lines are valid by the time the ϕ_{12} clock pulse occurs. If a system chip is to be read, it must respond during the ϕ_2 clock pulse. Information then enters the CPU, goes through the ALU, and is stored during the next ϕ_1 clock pulse. When the CPU sends out addresses or data, they are valid by the time the ϕ_1 clock pulse occurs.

precharged scheme. Timing is such that the driving circuit is not enabled until its data is valid and ready to go on the bus. It is not possible to discharge the bus inadvertently when the chip wants a logic one to be output.

ROM

The NMOS system ROM is organized as an 8K by eight-bit array. Because this circuit must respond to multibyte transfers, its memory address register is designed to increment as \overline{RD} commands are given. The circuit can be enabled or disabled by a bank select command. When plug-in ROMs are used, it becomes necessary to selectively enable or disable ROMs in the address space from 24K to 32K (Fig. 2). Each ROM in this address space has a unique eight-bit bank number, and only one bank can be active at any given time.

The core of the ROM consists of 64-input, minimum geometry, NAND gate arrays. The principle of operation is to precharge the NAND stack from both ends. This requires 600 ns. At the beginning of phase ϕ_{12} , one 64-device stack per bit of output is selected. If the stack discharges, the bus bit will be a zero, otherwise it will be a one. The transistors

in the stack are preprogrammed during fabrication with the depletion mask. A depletion transistor is a logic zero, an enhancement transistor is a logic one. A high voltage on either transistor causes conduction, while a low voltage causes conduction only through the depletion device.

When trying to discharge a 64-device stack, 63 inputs are high and the selected bit is low. If a depletion device is at that location, the stack discharges, producing a zero.

RAM Controller

The NMOS RAM controller chip interfaces the CPU to eight 16K dynamic RAMs. Because it is a memory controller chip it has an incrementing memory address register. Also, the address space to which it responds is mask programmable. Therefore, a RAM controller chip different from the one in the mainframe of the HP-85 is used for the 16K RAM plug-in. This chip knows not to respond to dedicated I/O addresses in the upper 256 bytes of memory.

An important function of the RAM controller chip is refreshing the dynamic memory. An internal timer on the chip tells it when the next sequential location is due to be

The HP-85 Software Development System

by Nelson A. Mills

The HP-85 is based upon a new eight-bit custom processor. One of the first steps in the development of the machine was to provide a set of software development tools. The HP-85 software development system consists of an assembler, a hardware interface for the HP-85 simulator, and software debug system (see Fig. 1).

The assembler for the HP-85 processor was designed to run on an HP 1000 Computer. The assembler supports the full range of the HP-85 processor's instruction set and provides several useful pseudo-operations. Programs may be either absolute or relocatable and the program origin may be reset at any time. Several instructions are provided to facilitate data definition and may exist locally within

the program, or may be retrieved from a file of global data definitions.

The hardware interface designed for the HP-85 software development system provides the ability to use an HP 1000 to control execution of the HP-85 simulator. The interface contains an on-board RAM which can be downloaded from the HP 1000 and is used to simulate the HP-85 ROM. Included are two breakpoint registers which can be used to halt execution of the HP-85 processor at either of two specified addresses. The processor can be made to execute steps in either a continuous-run mode, or in a single-step mode where execution halts after each instruction. The interface also provides the ability to halt the HP-85 processor at any time, and to read data from or write data to the system RAM or CPU registers.

The software debug system, provided as part of the development system, runs in the HP 1000 and performs four important functions. It includes a relocatable loader that is used to download the RAM on the interface board with the software under development. It controls execution of the HP-85 simulator via the interface board. It displays the current status of the breadboard, including CPU status, program counter, and the current contents of all CPU registers and any specified memory locations. Finally, it provides the means for system developers to modify the contents of memory, registers, status, or program counter, and to clear and set breakpoints.

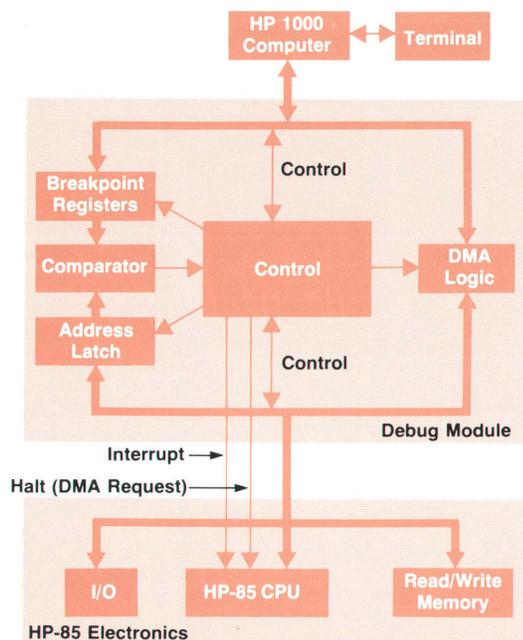


Fig. 1. HP-85 software development system.

Nelson A. Mills



Joining HP in 1976, Nelson Mills worked on the HP-85 operating system and interpreter. He is now the project manager for high-end firmware at the Corvallis Division. After receiving a BS in mathematics from Albion College, Michigan in 1961, Nelson spent five years in the U.S. Navy and then worked for ten years as a systems programmer. He and his family—wife and two children—live in Corvallis, Oregon. Outside of working hours Nelson enjoys photography, hiking, camping, and coaching basketball and AYSO soccer.

refreshed. The chip then waits for a cycle when memory is not being accessed to do the refresh. There is a two-element queue in case a vacant memory cycle does not come before the following location must be refreshed.

Keyboard Controller

The circuit needed just to interface to the keyboard would have been quite small. Consequently, four programmable timers were added to this NMOS chip. The timers can count up to 27 hours with one-millisecond resolution. Each has a maskable interrupt. When the timer gets to its preset count, it interrupts, resets and counts again.

The keyboard portion of the chip can scan an 8×10-key keyboard plus three dedicated keys—shift, control, and caps lock. The key debounce time is mask programmable from 1.67 to 11.69 ms. After the key has been debounced, the chip generates an interrupt. An internal ROM is used to convert the key position to its ASCII* equivalent.

Another feature of the chip is an output to a speaker for audio tones. Via firmware, a 1.2-kHz output tone can be obtained, or the frequency can be varied by periodically setting and clearing an internal flip-flop. This output can be accessed by the programmer using a command that can specify the frequency and duration of the tone.

CRT Controller

The NMOS CRT controller¹ interfaces the CPU to a 127-mm diagonal CRT and its dedicated display memory. The memory is four 16K×1 dynamic RAMs. This is enough storage to hold a full display of graphics information (256×192 dots) plus four displays of alphanumeric information (4×32 characters/line × 16 lines/display). The chip must make sure that this memory is properly refreshed during the vertical retrace. The interface to the video drive circuitry consists of a vertical sync pulse (60 Hz), a horizontal sync pulse (15.7 kHz), and a video line (4.9 MHz). An internal ROM provides a dot pattern translation of the stored ASCII characters.

Printer Controller

The NMOS printer controller² is the interface between the CPU and the thermal moving-head printer. To perform this function, the circuit must control a printhead drive motor, a paper advance motor and an eight-dot printhead. An internal 32-byte RAM allows firmware to buffer one print line of alphanumeric data at a time. An internal ROM translates an ASCII character into its appropriate dot pattern.

Tape Controller and Read/Write Amplifier

The NMOS tape controller and the bipolar read/write amplifier³ control the tape unit in the HP-85. The read/write amplifier reads and writes data on the tape through a two-track magnetic head. The tape is formatted using a 1:1.75 delta distance code, in which 8-kHz flux reversals represent zeros and 4.6-kHz flux reversals represent ones.

The tape controller encodes digital information into this format and sends it to the read/write amplifier. When reading the tape, the signals from the read/write amplifier are decoded by the controller. The tape drive motor direction and speed are also controlled by the chip.

*American Standard Code for Information Interchange

Buffer

The NMOS buffer is the ninth chip in the system. With its help, the system bus can be expanded to include all of the plug-in I/O slots. It is capable of driving a 150-pF load. The delay between NMOS-level input and NMOS-level output is 50 ns. The signals that the buffer passes are eight bus lines, three control lines, and the interrupt and halt lines. The bus and control lines are designed to be bidirectional.

The clock lines are not buffered since this would cause unwanted skewing. The clock generator is capable of driving all internal as well as all external loads.

Acknowledgments

A great deal of credit must be given to the team that made this LSI chip set possible. Tim Williams designed the RAM controller and buffer. Donn Wahl designed the ROM. The keyboard controller was an effort from Jim Hutchins and Jerry Erickson. Jerry also designed the CRT controller. The printer controller was the work of Clement Lo. Implementation of the NMOS CPU was done by Jim Axtell. The cartridge controller design is that of Doug Collins. The sense amplifier is the result of the combined efforts of Mike Moore, Doug Collins and Mike Barbour. Special thanks must be given to Don Hale and Howard Shishido, who did the majority of the chip mask designs. Howard Honig contributed to the project by writing most of the test programs for the LSI circuits. Mike Pan, Rock Davidson, and Tom Kraemer also contributed to the chip set design effort. Bob Tillman and Steve Larsen helped get the NMOS process and polysilicon resistor development up to production standards. Payne Freret contributed some good ideas to the CPU design. Rick Bell did the design of the clock oscillator and generator.

Reference

1. J.F. Bausch, "A High-Quality CRT Display for a Portable Computer," Hewlett-Packard Journal, July 1980.
2. C.C. Lo and R.W. Keil, "A Compact Thermal Printer Designed for Integration into a Personal Computer," Hewlett-Packard Journal, July 1980.
3. D.J. Collins and B.G. Spreadbury, "A Compact Tape Transport Subassembly Designed for Reliability and Low Cost," Hewlett-Packard Journal, July 1980.

Todd R. Lynch



Todd Lynch is a native of Rochester, New York. He joined HP Laboratories in 1972 after working for a year on process control logic designs. Since coming to HP he has worked on computer architectures, and since transferring to what is now HP's Corvallis Division, Todd designed the architecture for the HP-85 CPU. He was the project manager for the integrated circuits in the HP-85 and is now project manager for high-end mainframes. He received a BS degree from Grove City College, Pennsylvania in 1970 and an MSEE from the University of Wisconsin in

1971. Todd is co-chairman for the Willamette Valley Junior Tennis Tournament and in addition to tennis enjoys woodworking, softball, and skiing. He lives in Albany, Oregon with his wife and new son.

Handheld Calculator Evaluates Integrals

The HP-34C is the first handheld calculator to have a key that performs numerical integration almost automatically. It may change your attitude towards what used to be regarded as a dreary tedious task.

by William M. Kahan

NUMERICAL INTEGRATION has been the subject of about two thousand books and learned papers, with a dozen or so "new" methods published every year. And yet the task in question has a simple geometrical interpretation seen in Fig. 1: given an expression $f(u)$ and lower and upper limits y and x respectively, the value

$$I = \int_y^x f(u) du$$

represents the area under the graph of $f(u)$ for u between y and x . Why so much fuss?

As I write this an electrical engineering colleague, Professor J. R. Woodyard, enters my office and asks to have

$$I_1 = \int_0^1 \left(\frac{\sqrt{u}}{u-1} - \frac{1}{\ln u} \right) du$$

evaluated on my HP-34C Calculator (Fig. 2). Let's do it.

Step 1. Key into the calculator under, say, label **A** a program that accepts a value u in the display (X register) and displays instead the computed value of the integrand

$$\sqrt{u}/(u-1) - 1/(\ln u)$$

Fig. 3 shows an HP-34C program to do this.

Step 2. Restore the calculator to **RUN** mode and set the display to, say, **FIX 5** to display five decimal digits after the point, which are as many digits of the integrand as my client says he cares to see. (More about this later.)

Step 3. Key in the lower and upper limits of integration thus, **0 ENTER** **1**, thereby putting 0 into the Y register and 1 into X.

Step 4. Press \int_y^x **A**, wait 25 seconds until the display shows 0.03662, then press $\times \leq y$ to see 0.00001. We have just calculated

$$I_1 = 0.03662 \pm 0.00001.$$

That was easy—too easy. Woodyard smiles as if he knew something I don't know. Could the calculator be wrong? How does the calculator know the error lies within ± 0.00001 ?

Many other questions come to mind:

- Why is numerical integration impossible in general?
- Why do we persist in trying to do it anyway?
- How do we do it? How well do we do it?
- How does the \int_y^x key compare with other integration schemes?

- What can go wrong and how do we avoid it?
- What else have we learned?

These questions and others are addressed in the following pages.

Tolerance and Uncertainty

Integrals can almost never be calculated precisely. How much error has to be tolerated? The \int_y^x key answers this question in a surprisingly convenient way. Rather than be told how accurately $I = \int_y^x f(u)du$ should be calculated, the HP-34C asks to be told how many figures of $f(u)$ matter. In effect, the user is asked to specify the width of a ribbon drawn around the graph of $f(u)$, and to accept in place of I an estimate of the area under some unspecified graph lying entirely within that ribbon. Of course, this estimate could vary by as much as the area of the ribbon, so the calculator estimates the area of the ribbon too. Then the user may conclude from Fig. 4 that

$$I = (\text{area under a graph drawn in the ribbon}) \pm (\frac{1}{2} \text{ area of the ribbon})$$

The calculator puts the first area estimate in its X register and the second, the uncertainty, in the Y register.

For example, $f(u)$ might represent a physical effect whose magnitude can be determined only to within, say, ± 0.005 . Then the value calculated as $f(u)$ is really $f(u) \pm \Delta f(u)$ with an uncertainty $\Delta f(u) = 0.005$. Consequently **FIX 2**, which tells the calculator to display no more than two decimal digits after the point, is used to tell the calculator that decimal digits beyond the second cannot matter. Therefore the calculator need not waste time estimating $I \pm \Delta I = \int_y^x (f(u) \pm \Delta f(u))du$ more accurately than to within an uncertainty $\Delta I = |\int_y^x \Delta f(u)du|$. This uncertainty is estimated together with $I \pm \Delta I$, thereby giving the calculator's user a fair idea of the range of values within which I must lie.

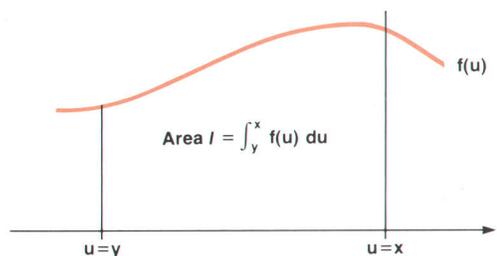


Fig. 1. An integral interpreted as an area.

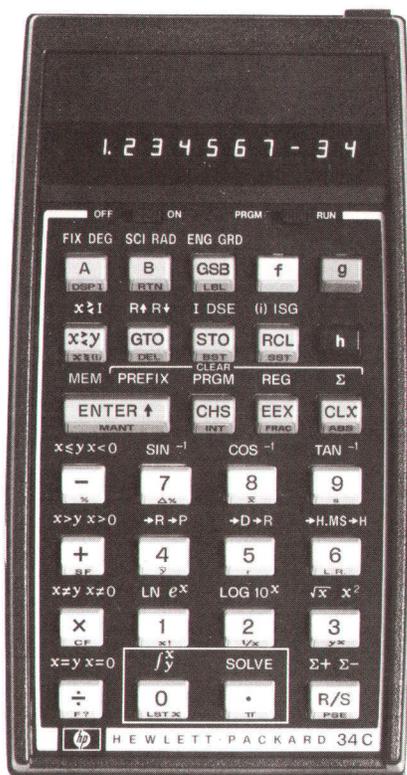


Fig. 2. HP-34C Calculator has keys to solve any equation and to compute integrals.

The uncertainty $\Delta f(u)$ is specified by the user via the display setting. For instance, **SCI 5** displays six significant decimal digits, implying that the seventh doesn't matter. The HP-34C allows the user's *f*-program to change the display setting, thereby providing for uncertainties $\Delta f(u)$ that vary with *u* in diverse ways. But users usually leave the display set to **SCI 4** or **FIX 4** without much further thought.

By asking the user to specify $\Delta f(u)$ instead of ΔI the HP-34C helps avoid a common mistake—wishful thinking. Other integration procedures, which conventionally expect the user to specify how tiny ΔI should be, blithely produce estimates of *I* purporting to be as accurate as the user wishes even when the error $\Delta f(u)$ is far too big to justify such claims to accuracy. The HP-34C does not prevent us from declaring that *f*(*u*) is far more accurate than it really is, but our attention is directed to the right question and not distracted by questions we cannot answer. Whether we specify Δf after a careful error analysis or just offer a guess, we get estimates $I \pm \Delta I$ that we can interpret more intelligently than if we got only *I* with no idea of its accuracy or inaccuracy.

A Survey of Integration Schemes

Students are taught the fundamental theorem of calculus:

$$I = \int_y^x f(u) du = F(x) - F(y) \quad \text{provided} \quad \frac{d}{du} F(u) = f(u)$$

This means that one could calculate *I* if one could discover somehow an expression *F*(*u*) whose derivative is the given expression *f*(*u*). Students are taught integration

as a process, applied to expressions, that starts with *f* and ends with *F*. But in professional practice that process hardly ever succeeds. A compact expression *F*(*u*) is almost always difficult or impossible to construct from any given *f*(*u*). For instance, neither

$$\int_{-\infty}^x \exp(-u^2/2) du / \sqrt{2\pi} \quad \text{nor} \quad \int_0^{\infty} \exp(-u + x \ln u) du$$

possesses a closed form, that is, an expression involving only finitely many elementary operations (+, −, ×, ÷, ln, exp, tan, arctan, ...) upon the variable *x*. Nevertheless, both integrals can be approximated arbitrarily accurately by aptly chosen formulas. So often do statisticians and engineers need values of those integrals that formulas for them, accurate to ten significant decimal digits, can now be calculated in a few seconds by pressing a key on certain handheld calculators. (Press **Q** on the HP-32E to get the first integral, the cumulative normal distribution; press **x!** on the HP-34C to get the second integral, the gamma function $\Gamma(1+x)$, whether *x* be an integer or not.)

Almost every rare integrand *f*(*u*) whose indefinite integral $F(x) = \int^x f(u) du + c$ is expressible in a compact or closed form can be recognized by a computer program that accepts the string of characters that defines *f* and spews out another string that represents *F*. (Such a program is part of the MACSYMA system, developed at MIT, that runs on a few large computers—two million bytes of memory—at several universities and research labs.) Perhaps the terms “compact” and “closed form” should not be attached to the expression *F*(*x*), since usually, except for problems assigned to students by considerate teachers, the integral

LBL A	Begin with <i>u</i> in the X register
ENTER ↑	Save <i>u</i> in the stack
\sqrt{x}	... \sqrt{u}
LST X	Recall <i>u</i>
1	
−	
÷	... $\sqrt{u}(u-1)$
x ≤ y	Recall <i>u</i> again
LN	
1/x	... $1/\ln(u)$
−	
RTN	Display $\sqrt{u}(u-1) - 1/\ln u$

Fig. 3. This program makes the HP-34C calculate the integrand $\sqrt{u}(u-1) - 1/\ln u$ when the argument *u* is in the X register and key **A** is pressed. Labels **B**, **0**, **1**, **2**, or **3** would have served as well as **A**.

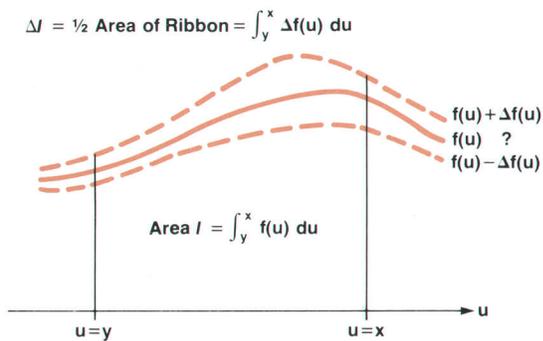


Fig. 4. The graph of an uncertain integrand $f(u) \pm \Delta f(u)$ can run anywhere in the ribbon bounded by the dashed lines. The area under such a graph, $I \pm \Delta I$, is uncertain by $\pm \Delta I$, which is one-half the area of the ribbon. The HP-34C displays its estimate of $I \pm \Delta I$ in its X register and holds an estimate of ΔI in its Y register.

$F(x)$ far exceeds the integrand $f(u)$ in length and complexity. Shown in Fig. 5 are two compact forms and one closed form for $F(x)$ when $f(u) \equiv 1/(1+u^{64})$. The extent to which $F(x)$ is here more complicated than $f(u)$ is atypically modest out of consideration for the typesetter. The formulas in Fig. 5 will remind many readers of hours spent on calculus problems, but they do not provide economical ways to calculate $F(x)$ for any but very big or very tiny values of x . When I use the HP-34C's \int_y^x key to calculate $F(1) = \int_0^1 du/(1+u^{64}) = 0.989367 \pm 0.000004$ the answer appears in 200 seconds including 20 seconds taken to enter the f-program plus 180 seconds for a result (in **SCI 5**). Calculating $F(1)$ from any formula in Fig. 5 takes at least about ten times longer, not including the time taken to deduce the formula. Engineers and scientists have long been aware of the shortcomings of integration in closed form and have turned to other methods.

Perhaps the crudest way to evaluate $\int_y^x f(u)du$ is to plot the graph of $f(u)$, like Fig. 1, on uniformly squared paper and then count the squares that lie inside the desired area. This method gives numerical integration its other name: numerical quadrature. Another way, suitable for chemists, is to plot the graph on paper of uniform density, cut out the area in question, and weigh it. Engineers used to measure plotted areas by means of integrating engines called planimeters. These range from inexpensive hatchet planimeters of low accuracy to Swiss-made museum pieces costing hundreds of dollars and capable of three significant decimals. (For more details see reference 1). Nowadays we reckon that the computer will drive the graph plotter so it might as well integrate too.

Today's numerical integration techniques are best explained in terms of averages like

$$A = I/(x-y) = \int_y^x f(u)du/(x-y)$$

which is called "the uniformly weighted average of $f(u)$ over the interval between x and y ." Another kind of average,

$$A = \sum_{j=1}^n w_j f(u_j) \quad \text{where } w_j > 0 \text{ and } \sum_{j=1}^n w_j = 1,$$

is a finite weighted average of n samples $f(u_1), f(u_2), \dots, f(u_n)$.

Provided the sample arguments u_1, u_2, \dots, u_n , called nodes, all lie between x and y the sample average A will approximate, perhaps poorly, the desired average A , and hence provide $I = (x-y)A$ as an approximation to $I = \int_y^x f(u)du$. Statisticians might be tempted to sprinkle the nodes u_j randomly between x and y —that is what Monte Carlo methods do. But randomness is a poor substitute for skill because the error $A - A$ tends to diminish like $1/\sqrt{n}$ as the number n of random samples is increased, whereas uniformly spaced and weighted samples provide an error that diminishes like $1/n^2$. Other more artful methods do even better.

Different numerical integration methods differ principally in the ways they choose their weights w_j and nodes u_j , but almost all have the following characteristics in common. Each average A is associated with a partition of the range of integration into panels as shown in Fig. 6. Each panel contains one node u_j whose respective weight is

$$w_j = (\text{width of panel } j)/(\text{width of range of integration}).$$

The formula given above for A amounts to approximating the area in each panel under the graph of $f(u)$ by the area of a rectangle as wide as the panel and as high as the sample $f(u_j)$. The simplest method is the midpoint rule, whose nodes all lie in the middles of panels all of the same width. Other methods, like the trapezoidal rule and Simpson's rule, vary the panel widths (weights) and nodes in ways designed to exploit various presumed properties of the integrand $f(u)$ for higher accuracy. Which method is best? If this question had a simple answer there would not be so many methods nor would we need texts like "Methods of Numerical Integration" by P.J. Davis and P. Rabinowitz,² which contains 16 FORTRAN programs and three bibliographies with well over 1000 citations.

For example, consider Gaussian quadrature. This method is widely regarded as "best" in the sense that it very often requires fewer samples than most other methods to achieve an average A that approximates the desired A to within some preassigned tolerance. But the weights and nodes of Gaussian quadrature take quite a while to calculate. Programs to do so, and the resulting tables of weights and nodes for various sample counts n , have been published.³ Had we chosen Gaussian quadrature for the \int_y^x key we would

$$F(x) \equiv \int_0^x f(u) du \quad \text{where } f(u) \equiv 1/(1+u^{64}).$$

$$F(x) = x \sum_{k=0}^{\infty} (-x^{64})^k / (64k + 1) \quad \text{if } x^2 \leq 1$$

$$= \frac{\pi}{64} \csc\left(\frac{\pi}{64}\right) \text{sign}(x) + x \sum_{k=1}^{\infty} (-x^{-64})^k / (64k - 1) \quad \text{if } x^2 \geq 1$$

$$= \frac{1}{32} \sum_{k=1}^{16} \left(\sin \theta_k \cdot \arctan\left(\frac{2x \sin \theta_k}{1-x^2}\right) + \frac{1}{2} \cos \theta_k \cdot \ln\left(1 + \frac{2}{\frac{x+x^{-1}}{2 \cos \theta_k} - 1}\right) \right)$$

$$+ \left(\frac{\pi}{64} \csc\left(\frac{\pi}{64}\right) \text{sign}(x) \quad \text{if } x^2 > 1\right) \quad \text{where } \theta_k = (k-1/2)\pi/32$$

Fig. 5. Formal integration transforms many a simple expression $f(u)$ into messy formulas $F(x)$ of limited numerical utility.

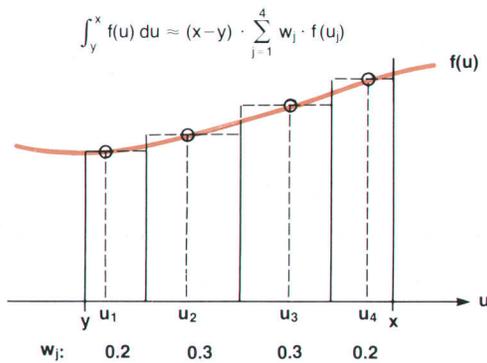


Fig. 6. The integral, regarded as an area, is here divided into four panels each of whose areas is approximated by the area of a rectangle as wide as the panel and as high as a sample.

have had to store at least as many nodes and weights as we could expect to need for difficult integrals, amounting to at least several hundred 13-digit numbers, in read-only memory. But that would have left no space in the HP-34C for anything else, so a different method had to be found.

The \int_y^x key could not use a method that generates just one average A because that gives no indication of how accurately it approximates A . Instead we looked only at methods that sample repeatedly and with increasing sample counts $n_1 < n_2 < n_3 < \dots$ to produce a sequence of increasingly accurate averages A_1, A_2, A_3, \dots . Provided that sequence converges to A so fast that each $|A_{k+1} - A|$ is considerably smaller than its predecessor, the error $|A_k - A|$ can be approximated accurately enough by $|A_k - A_{k+1}|$, and the last average A_{k+1} can be accepted in lieu of A as soon as $|A_k - A_{k+1}|$ is tolerably small.

How small is "tolerably small"? That depends upon the area of the ribbon discussed above under "Tolerance and Uncertainty." Since the integral $I = \int_y^x f(u)du$ inherits an uncertainty $\Delta I = \left| \int_y^x \Delta f(u)du \right|$ from the uncertainty $\Delta f(u)$ in the integrand, so does $A = I/(x-y)$ inherit an uncertainty $\Delta A = \Delta I/|x-y|$, which may be approximated by

$$\Delta A = \sum_{j=1}^n w_j \Delta f(u_j)$$

in the same way as A is approximated by A . Indeed, A and ΔA can be computed together since they use identical weights and nodes. And so the sequence A_1, A_2, A_3, \dots is accompanied by a sequence of respective uncertainty estimates $\Delta A_1, \Delta A_2, \Delta A_3, \dots$. Now "tolerably small" can be defined to mean "rather smaller than ΔA_{k+1} ."

The foregoing argument provides an excuse for accepting A_{k+1} in lieu of A whenever two consecutive estimates A_k and A_{k+1} agree to within ΔA_{k+1} , but it provides no defense against the possibility that convergence is not so fast, in which case A_k and A_{k+1} might agree by accident and yet be both quite different from A . The \int_y^x key waits for three consecutive estimates A_k, A_{k+1} , and A_{k+2} to agree within ΔA_{k+2} . Only the most conservative integration schemes wait that long. While this conservatism strongly attenuates the risk of accidental premature acceptance of an estimate, the risk that three consecutive estimates might agree within the tolerance and yet be quite wrong cannot be eliminated. Later, under "How to Deceive Every Nu-

merical Integration Procedure," some such risk will be proved unavoidable, but the risk now is so small that further attenuation is not worth its cost.

The combination of ignorance with conservatism is surprisingly costly. Had we known in advance that A_k would be accurate enough we would have calculated none of the other averages. Instead, waiting for three consecutive averages to agree could easily cost some methods almost 6.25 times as many samples as if only A_k had to be calculated, and more than that if the sample counts n_1, n_2, n_3, \dots are not chosen optimally. For the \int_y^x key we chose $n_k = 2^k - 1$ and we chose a method whose successive averages each share almost half of the previous average's samples, thereby preventing the cost of ignorance from much exceeding a factor of 4.

Memory limitations precluded the use of another family of methods known as adaptive quadrature. These methods attempt to distribute nodes more densely where the integrand $f(u)$ appears to fluctuate rapidly, less densely elsewhere where $f(u)$ appears to be nearly constant or relatively negligible. They succeed often enough that the best general-purpose integrators on large computers are adaptive programs like Carl de Boor's CADRE; this and others are described in reference 2. Alas, adaptive programs consume rather more memory for scratch space than the twenty registers available in the HP-34C.

What Method Underlies the \int_y^x Key?

The HP-34C uses a Romberg method; for details consult reference 2. Several refinements were found necessary. Instead of uniformly spaced nodes, which can induce a kind of resonance or aliasing that produces misleading results when the integrand is periodic, the \int_y^x key's nodes are spaced nonuniformly. Their spacing can be explained by substituting, say,

$$u = \frac{3}{2}v - \frac{1}{2}v^3$$

into

$$I = \int_{-1}^1 f(u)du = \int_{-1}^1 f\left(\frac{3}{2}v - \frac{1}{2}v^3\right) \cdot \frac{3}{2}(1-v^2)dv$$

and distributing nodes uniformly in the second integral. Besides suppressing resonance, the substitution confers two more benefits. One is that no sample need be drawn from either end of the interval of integration, except when the interval is so narrow that no other possibilities are available, and consequently an integral like

$$\int_0^3 \frac{\sin u}{u} du$$

won't hang up on division by zero at an endpoint. Second, $I = \int_y^x f(u)du$ can be calculated efficiently when $f(u) = g(u)\sqrt{|x-u|}$ or $g(u)\sqrt{(x-u)(u-y)}$ where $g(u)$ is everywhere a smooth function, without any of the expedients that would otherwise be required to cope with the infinite values taken by the derivative $f'(u)$ at $u = x$ or $u = y$. Such integrals are encountered often during calculations of areas enclosed by smooth closed curves. For example, the area of a circle of radius 1 is

$$\int_0^2 \sqrt{u(4-u)} du = 3.14159 \pm 8.8 \times 10^{-6}$$

which consumes only 60 seconds when evaluated in **SCI 5** and only 110 seconds to get $3.141592654 \pm 1.4 \times 10^{-9}$ in **SCI 9**.

Another refinement is the use of extended precision, 13 significant decimal digits, to accumulate the sums that define A_k , thereby allowing thousands of samples to be accumulated, if necessary, without losing to roundoff any more information than is lost within the user's own f-program. The last example's 10 significant decimal digits of π could not have been achieved without such a refinement.

How Does the \int_y^x Key Compare with Other Integrators?

What most distinguishes the HP-34C's \int_y^x key from all other schemes is its ease of use. No step-size parameters, no plethora of error tolerances, no warning indicators that "can usually be ignored." Only the minimum information needed to specify $\int_y^x (f(u) \pm \Delta f(u)) du$ has to be supplied. And because the \int_y^x key is effective over so wide a range of integrals it ranks among the most reliable procedures available anywhere. Usually it is far faster than simpler procedures like the trapezoidal rule or Simpson's rule commonly used previously on calculators. For integrands defined by programs that fit comfortably into a mid-sized handheld calculator that can hold at most 210 program steps, the \int_y^x key is comparable in speed (count the number of samples) with the integrators available on large computers. For much more complicated integrands the best adaptive integrators on large computers are appreciably faster.

One of the HP-34C's most important components is its

Owner's Handbook. It is for most owners the first guide to the foothills of an awesome range of new possibilities. Two chapters are devoted to \int_y^x . The first is introductory, and allows the user to evaluate simple integrals effortlessly and confidently. The second chapter is a longer explanation of the power and the pitfalls, concerned mainly with numerical integration generally rather than with the HP-34C in particular. This chapter had to be included because its explanations and practical advice are not yet to be found in any text likely to be consulted by an owner, nor are they supplied by the instructions that accompany other integrators on other computers or calculators. This second chapter is part of the educational burden that must be borne by innovators and pioneers. The Owner's Handbook provides no formulas for the nodes and weights used by the HP-34C because they are not needed to understand how the \int_y^x key works; instead they can be deduced from information in this article.

Every numerical integrator like \int_y^x , which executes a user-supplied program to get the integrand's value $f(u)$, imposes constraints upon that program. Some constraints, like requiring f to have a smooth graph on the interval of integration, are practically unavoidable. Others are nuisances like

- Begin the f-program with a special label, say **A'**.
- Do not use certain memory registers, say #0 - #5.
- Do not use certain operations, say = and **CLR**.

The \int_y^x key is encumbered with no such nuisances. The f-program may begin with any of several labels, so several different integrals can be calculated during one long computation. The f-program may use memory registers freely and may use any operation key except \int_y^x itself. One of

Evaluation in RUN Mode	Integrand in PRGM Mode	Equation in PRGM Mode
FIX 5	LBL A	LBL B
CLEAR REG	x ≥ I ... save u	I ... get u
0	SOLVE B ... for v	× ... uv
ENTER ↑	x ≥ I ... save v, get u	LST x
1	I	e^x ... e ^u
\int_y^x A	÷ ... u/v	×
Wait for answer / to be displayed	RTN	1
		+ ... 1+uve ^u
		ln
		I
		-
		+
		RTN ... v-u+ln(1+uve ^u)

Fig. 7. A program to evaluate $I = \int_0^1 u du/v(u)$ where $v = v(u)$ satisfies $v - u + \ln(1 + uve^u) = 0$.

those keys is the HP-34C's powerful **SOLVE** key.⁴ Consequently this calculator is currently the only one that can evaluate conveniently integrals of implicit functions.

For example, let $v = v(u)$ be the root of the equation

$$v - u + \ln(1 + uv^u) = 0.$$

Then

$$\int_0^1 u \, du/v(u) = 1.81300 \pm 0.000005$$

results from a program rather shorter than on any previous calculator; it is exhibited in Fig. 7.

Furthermore, \int_y^x may be invoked, like any other function, from within a program, thereby permitting the HP-34C to **SOLVE** equations involving integrals. For example, solving

$$\int_0^\pi \cos(x \sin \theta) d\theta = 0$$

for $x = 2.405\dots$ takes a short program contained in the Owner's Handbook, and exhibits the first zero of the Bessel function $J_0(x)$.

How to Deceive Every Numerical Integration Procedure

Such a procedure must be a computer program—call it P—that accepts as data two numerical values x and y and a program that calculates $f(u)$ for any given value u , and from that data P must estimate $I = \int_y^x f(u) \, du$. The integration procedure P is not allowed to read and understand the f -program but merely to execute it finitely often, as often as P likes, with any arguments u that P chooses. What follows is a scheme to deceive P.

First ask P to estimate I for any two different values x and y and for $f(u) \equiv 0$. Record the distinct arguments u_1, u_2, \dots, u_n at which P evaluates $f(u)$. Presumably when P finds that $f(u_1) = f(u_2) = \dots = f(u_n) = 0$ it will decide that $I = 0$ and say so. Next give P a new task with the same limits x and y as before but with a different integrand

$$f(u) \equiv ((u-u_1) \cdot (u-u_2) \cdot \dots \cdot (u-u_n))^2.$$

Once again P will calculate $f(u_1), f(u_2), \dots$, and finding no difference between the new f and the old, P will repeat exactly what it did before. But the new integral I is quite different from the old, so P must be deceived.

The HP-34C's \int_y^x key can be hoodwinked that way. Try to evaluate $\int_{-128}^{+128} f(u) \, du$ using first $f(u) \equiv 0$ programmed in a way that pauses (use the **PSE** key) to display its argument u . The calculator will display each sample argument it uses, namely 0, ± 88 , ± 47 and ± 117 . Next program

$$f(u) \equiv (u(u-88)(u+88)(u-47)(u+47)(u-117)(u+117))^2$$

and evaluate $\int_{-128}^{+128} f(u) \, du$ again. The calculator will say that both integrals are 0, but the second polynomial's integral is really 1.310269×10^{28} . That polynomial's graph, shown in Fig. 8, has the sharp spikes that characterize integrands troublesome for every numerical integration procedure. To calculate the integral correctly, reevaluate it as $2 \int_0^{128} f(u) \, du$, thereby doubling the spikes' width compared with the range of integration.

The threat of deceit impales the designer of a numerical integrator upon the horns of a dilemma. We all want our integrators to work fast, especially when the integrand $f(u)$ is very smooth and simple like $f(u) = 3u - 4$. But if the integrator is too fast it must be easy to deceive; fast integration means few samples $f(u_j)$, implying wide gaps between some samples, which leave room for deceitful misbehavior. Figs. 9a-9e illustrate the dilemma with two estimates of $\int_y^x f(u) \, du$. The first estimate is based upon the three samples drawn at the white dots, the second upon seven samples including those three white plus four more black dots. Fig. 9a shows why all sufficiently smooth graphs $f(u)$ that agree at all seven samples have nearly the same integrals, but Fig. 9b shows how two integrands could provide the same samples and yet very different integrals. The coincidence in Fig. 9b is unlikely; successive estimates based upon increasingly dense sampling normally would reveal the difference as in Fig. 9c. However, situations like those illustrated in Figs. 9d and 9e are very likely to deceive.

Textbooks tell us how to avoid being deceived: avoid integrands $f(u)$ among whose first several derivatives are some that take wildly different values at different places in the range of integration. Or avoid integrands $f(u)$ that take wildly different values when evaluated at complex arguments in some neighborhood of the range of integration. And if wild integrands cannot be avoided they must be tamed. We shall rejoin this train of thought later.

Improper and Nearly Improper Integrals

An improper integral is one that involves ∞ in at least one of the following ways:

- One or both limits of integration are $\pm\infty$, e.g.,

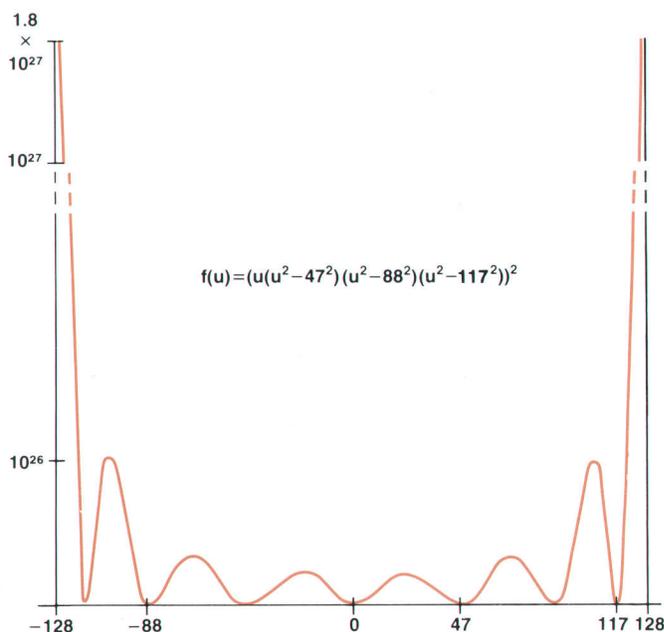


Fig. 8. The polynomial $f(u)$ was devised to deceive the HP-34C into miscalculating its integral as 0 instead of 1.31×10^{28} . This spiky graph is typical of integrands that can baffle any numerical integrator. 73% of the area under the graph lies under two spikes whose widths span less than 4% of the area of integration.

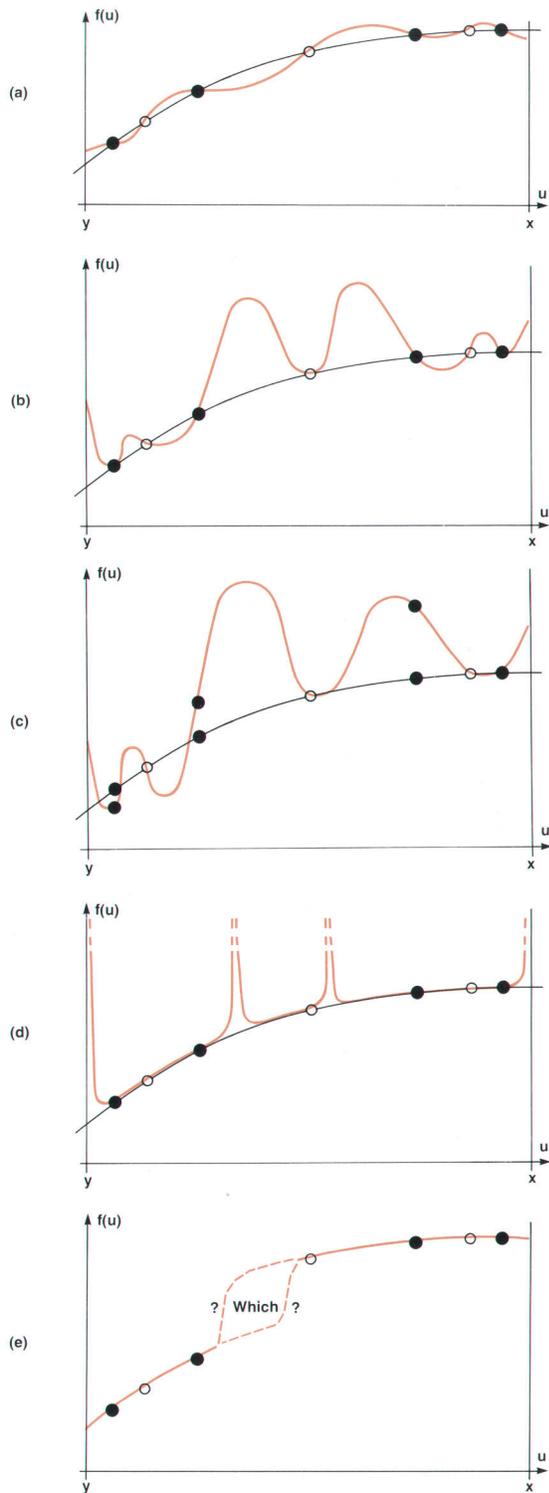


Fig. 9. Few samples (open circles) mean fast integration but a large possibility of error. More samples (solid dots plus open circles) usually mean more accuracy, but not always, as in (b), (d), and (e). (a) Which is the graph of $f(u)$? No matter; both have almost the same integral. (b) Which is the graph of $f(u)$? They have very different integrals. (c) Here two graphs that coincide on the first samples \circ are distinguished by a significantly different outcome after second samples \bullet are drawn. (d) If the graph of $f(u)$ has a few sharp and narrow spikes, they will probably be overlooked during the estimation of the integral based on finitely many samples. (e) If the graph of $f(u)$ has a step that was not made known during the estimation of the integral, then the estimate may be mistaken.

$$\int_{-\infty}^{\infty} \exp(-u^2) du = \sqrt{\pi}/2.$$

- The integrand tends to $\pm\infty$ someplace in the range of integration, e.g.,

$$\int_0^1 \ln(u) du = 1.$$

- The integrand oscillates infinitely rapidly somewhere in the range of integration, e.g., $\int_0^1 \cos(\ln u) du = 1/2$.

Improper integrals are obviously troublesome. Equally troublesome, and therefore entitled to be called nearly improper, are integrals afflicted with the following malady:

- The integrand or its first derivative changes wildly within a relatively narrow subinterval of the range of integration, or oscillates frequently across that range.

This affliction can be diagnosed in many different ways. Sometimes a small change in an endpoint can render the integral improper, as in

$$\int_{0.0001}^1 \ln(u) du = -0.99898 \dots \rightarrow \int_0^1 \ln(u) du = 1.$$

Sometimes a small alteration of the integrand can render the integral improper, as in

$$\int_{-1}^1 dx/(x^2 + 10^{-10}) = 314157.2654 \dots \rightarrow \int_{-1}^1 dx/x^2 = \infty.$$

Sometimes the value of the integral is nearly independent of relatively huge variations in one or both of the endpoints, as is $\int_0^x \exp(-u^2) du \approx \sqrt{\pi}/2$ for all $x > 10$. Regardless of the cause or diagnosis, nearly improper integrals are the bane of numerical integration programs, as we have seen.

During the HP-34C's design a suspicion arose that most integrals encountered in practice might be improper or nearly so. Precautions were taken. Now that experience has confirmed the suspicion, we are grateful for those precautions. They were:

1. Avoid sampling the integrand at the ends of the range of integration.
2. By precept and example in the Owner's Handbook, warn users against wild integrands, suggest how to recognize them, and illustrate how to tame them.

The second precaution ignited controversy. Against it on one side stood fears that its warnings were excessive and might induce paranoia among potential customers. Who would buy a calculator that he thinks gets wrong answers? Actually wrong answers were very rare, thanks in part to the first precaution, and many attempts to vindicate dire predictions about mischievous improper and nearly improper integrals were thwarted by unexpectedly correct answers like

$$\int_0^1 \ln(u) du = 0.9998 \pm 0.00021$$

in 2 minutes at **SCI 3**. Or

$$\int_0^{30} \exp(-u^2) du = 0.886227 \pm 0.0000008$$

in 4 minutes at **SCI 5**. If the wages of sin be death, O Death, where is thy sting?

On the other side stood a number of embarrassing examples like

$$\int_0^{400} \exp(-u^2) du$$

miscalculated as 0.0 ± 0.0000000005 in 14 seconds. Another, had we known it then, would have been Woodyard's example at the beginning of this article; the correct answer

$$\int_0^1 \left(\frac{\sqrt{u}}{u-1} - \frac{1}{\ln u} \right) du = 0.03649 \pm 0.00000007$$

in 23 minutes at **FIX 7** differs from **FIX 5**'s wrong answer 0.03662 in the worst way; the error is too small to be obvious and too large to ignore. Adding to the confusion were examples like

$$A(x) = x^{-1} \int_0^x \sqrt{-2 \ln \cos(u^2)} du/u^2 = 1 + x^4/60 + x^8/480 + \dots$$

for which computation in **SCI 4** produced ridiculous values like $A(0.1) = 0.95742 \pm 0.00005$, $A(0.01) = 0.58401 \pm 0.00003$, and $A(0.001) = 0$, all impossibly smaller than 1. This example appears to condemn the \int_y^x key until the integrand $f(u) = \sqrt{-2 \ln \cos(u^2)}/u^2$ is watched for small arguments u and seen to lose most of its figures to round-off, losing all of them for $|u| \leq 0.003$, despite an absence of subtractions that could be blamed for cancellation. Then the example appears to condemn the whole calculator. Who wants responsibility for a calculator that gets wrong answers?

Don't panic! The answers are wrong but the calculator is right.

How to Tame a Wild Integral

Forewarned is forearmed. Every experienced calculator user expects to encounter pathological examples like some of those above, and expects to cope with them. The question is not "whether" but "when"? And that is when attention to detail by the calculator's designers is rewarded by the user's freedom from petty distractions that can only complicate a task already complicated enough. But like the dog that did not bark,* the absence of distracting details may fail to be appreciated. That is why the examples explained below have been chosen—to illustrate the advantages of liberated thought. Work them on your calculator as you read them; don't skim them like a novel. Then you may come to think of your calculator the way I think of mine, as a trusted friend who stays with me when I need help.

The integral $A(x)$ above contains an integrand $f(u) = \sqrt{-2 \ln \cos(u^2)}/u^2$ that loses its figures when u becomes tiny. The problem is caused by rounding $\cos(u^2)$ to 1, which loses sight of how small u^2 must have been. The solution compensates for roundoff by calculating $f(u)$ as follows:

$$\begin{aligned} \text{Let } y &= \cos u^2 \text{ rounded.} \\ \text{If } y = 1 & \text{ then let } f(u) = 1 \\ & \text{else let } f(u) = \sqrt{-2 \ln y} / \cos^{-1} y. \end{aligned}$$

The test for $y = 1$ adds four steps to the f -program and, provided \ln and \cos^{-1} are implemented as accurately as on all recent HP calculators, the problem goes away.

*See the last few paragraphs of the Sherlock Holmes story "Silver Blaze" by Conan Doyle.

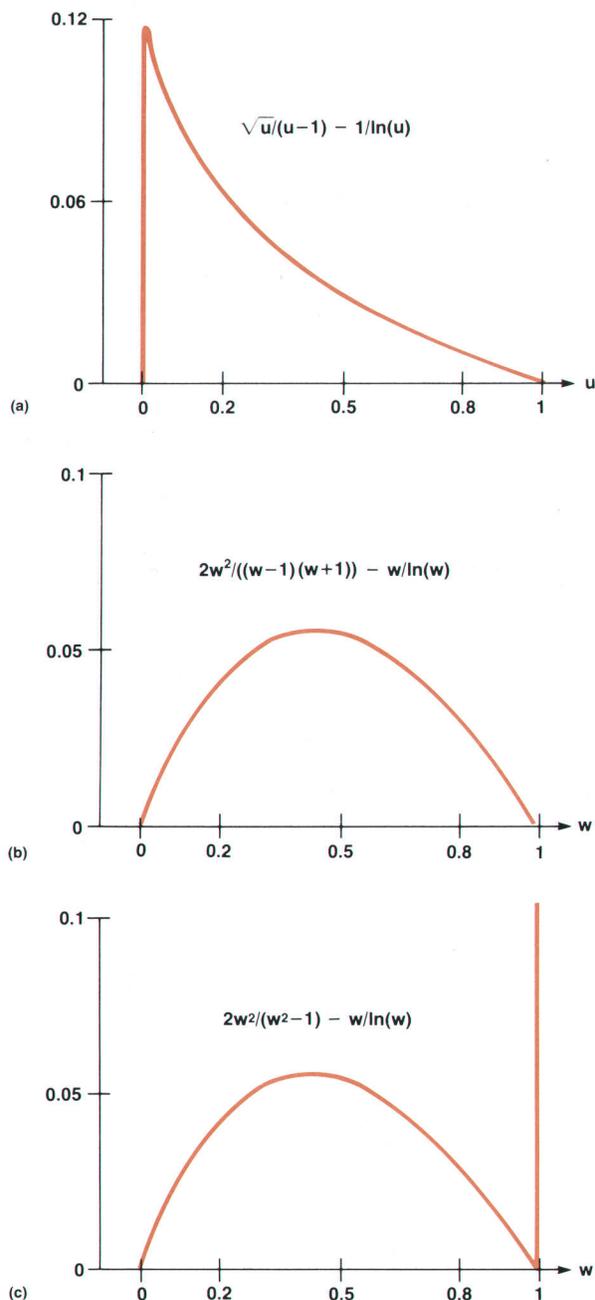


Fig. 10. Substituting w^2 for u turns the wild graph (a) into the easy one (b). But do not replace $((w-1)(w+1))$ by (w^2-1) because roundoff errors introduce a spike, as shown in (c).

Woodyard's example I_1 has an integrand $f(u)$ whose derivative $f'(u) \rightarrow \infty$ as $u \rightarrow 0$. The graph of $f(u)$ shown in Fig. 10a looks like a lovers' leap. Stretching the u -axis near $u=0$ by substituting $u = w^2$ turns the precipice into the hummock shown in Fig. 10b and transforms the integral into an easy calculation:

$$I_1 = \int_0^1 \left(\frac{2w^2}{(w-1)(w+1)} - \frac{w}{\ln w} \right) dw$$

The HP-34C computes this as 0.03649 ± 0.000005 in 100 seconds at **FIX 5** or 0.0364900 ± 0.00000008 in 200 seconds

at **FIG 7**. Do not replace $(w-1)(w+1)$ by (w^2-1) because the latter loses to roundoff half of its significant digits as $w \rightarrow 1$ and introduces a gratuitous spike into the integrand's graph shown in Fig. 10c, which was plotted on an HP-85. Do not worry about $w = 0$ or $w = 1$ because they don't happen, but do worry that as $w \rightarrow 1$ the integrand approaches the unreliable expression $\infty - \infty = 0$. This means that **FIG 7** displays about as many digits as could possibly be correct for all $w < 0.999$, beyond which the $\int_{\mathbf{y}}^{\mathbf{x}}$ key draws few if any samples because it converges so fast.

The graphs of $\exp(-u^2)$ over $0 \leq u \leq 300$ and of $1/(u^2 + 10^{-10})$ over $-1 \leq u \leq 1$ both resemble huddled mice with very long tails stretched out hundreds or thousands of times as long as their bodies. Plotting the graphs on a page of normal width is futile because the bodies get squashed into vertical whiskers.

Most people who integrate such functions numerically cut off the tails. Thin tails can be cut almost indiscriminately without much degrading the accuracy or the speed of integration. Such is the case for $\int_0^x \exp(-u^2) du$, which $\int_{\mathbf{y}}^{\mathbf{x}}$ evaluates in less than, say, 4 minutes at **SCI 5** provided that x , if bigger than 4 or 10, is cut back to something between 4 and 10. But $\int_{-x}^x du/(u^2 + 10^{-10})$ has too thick a tail to cut without losing accuracy or patience when x is large. That is why Procrustean methods are not recommended. Better to shrink the tail via an artful substitution like $u = \lambda + \mu \tan v$ where λ lies within the body of the mouse and μ is roughly that body's width. Doing so with $\lambda = 0$ and $\mu = 1$ changes $\int_0^x \exp(-u^2) du$ into

$$\int_0^{\arctan x} \exp(-\tan^2 v) (1 + \tan^2 v) dv$$

which $\int_{\mathbf{y}}^{\mathbf{x}}$ evaluates in three minutes at **SCI 5** even when x is as big as 10^{10} . Don't worry about $\tan \pi/2$ because it can't happen on a well-designed calculator.

$$\int_{-x}^x du/(u^2 + 10^{-10})$$

benefits miraculously from the foregoing substitution when $\lambda = 0$ and $\mu = 10^{-5}$, but values near those do almost as well.

Another technique might be called "subdivide and conquer." It subdivides the range of integration into subintervals upon each of which the integrand $f(u)$ is tame, although $f(u)$ may look wild on the range as a whole. For example, $f(u) \equiv \sqrt{u^2 + 10^{-10}}$ has a V-shaped graph practically the same as that of $|u|$. Evaluating $\int_{-3}^5 f(u) du$ accurately takes a long time if done with one press of $\int_{\mathbf{y}}^{\mathbf{x}}$, but subdividing the integral into

$$\int_{-3}^0 f(u) du + \int_0^5 f(u) du$$

takes two presses of $\int_{\mathbf{y}}^{\mathbf{x}}$ and one of $\Sigma+$ but much less time.

Subdivide and conquer works best when combined with apt substitutions. For example, if the formulas in Fig. 5 were unavailable how would $F(\infty) = \int_0^{\infty} du/(1 + u^{64})$ be calculated?

$$\begin{aligned} F(\infty) &= \int_0^1 du/(1 + u^{64}) + \int_1^{\infty} du/(1 + u^{64}) \quad \dots \text{subdivided} \\ &= \int_0^1 du/(1 + u^{64}) + \int_0^1 w^{62} dw/(w^{64} + 1) \quad \dots u=1/w \end{aligned}$$

$$\begin{aligned} &= \int_0^1 (1 + u^{62}) du/(1 + u^{64}) \quad \dots \text{merged via } w=u \\ &= 1 + \int_0^1 (u^{62} - u^{64}) du/(1 + u^{64}) \quad \dots \text{some algebra} \\ &= 1 + \frac{1}{8} \int_0^1 (1 - v^{3/4}) v^{55/8} dv/(1 + v^8) \quad \dots u=v^{3/4} \text{ to shrink a tail} \\ &= 1.000401708155 \pm 1.2 \times 10^{-12} \end{aligned}$$

in 10 minutes at **SCI 8**. Thus we have calculated $F(\infty) = (\pi/64)\text{csc}(\pi/64)$ to 13 significant decimals on a ten-significant-decimal calculator.

Oscillatory integrals like $\int_0^1 \cos(\ln u) du$ sometimes succumb to stretching substitutions like $u = v^2$ that damp the oscillations, but generally oscillatory integrals cannot be calculated accurately and quickly without sophisticated tricks beyond the scope of an article like this. A simple trick worth trying when the period of oscillation is known in advance is called *folding*, though it is really another instance of subdivide and conquer. Here is a didactic example.

$$\begin{aligned} I_3 &= \int_0^{600\pi} \frac{\sin^2 u}{\sqrt{u} + \sqrt{u+\pi}} du \quad \text{du = still running after over three hours at SCI 5} \\ &= \sum_{n=0}^{599} \int_{n\pi}^{n\pi+\pi} \frac{\sin^2 u}{\sqrt{u} + \sqrt{u+\pi}} du \\ &= \sum_{n=0}^{599} \int_0^{\pi} \frac{\sin^2 v}{\sqrt{v+n\pi} + \sqrt{v+n\pi+\pi}} dv \end{aligned}$$

after being subdivided and with $u = v + n\pi$. Exchanging \int and \sum produces

$$I_3 = \int_0^{\pi} \sin^2 v \cdot \sum_{n=0}^{599} \frac{1}{\sqrt{v+n\pi} + \sqrt{v+n\pi+\pi}} dv.$$

At this point a program should be written to calculate the sum, but because the example is didactic the sum collapses to yield

$$I_3 = \int_0^{\pi} \frac{600 \sin^2 v}{\sqrt{v} + \sqrt{v+600\pi}} dv = 21.10204 \pm 0.00007$$

in 5 minutes at **SCI 5**.

Now for a final example drawn from life:

$$V = \int_0^{\infty} \frac{du}{(a^2+u) \sqrt{(a^2+u)(b^2+u)(c^2+u)}} \quad \text{for } a=100, b=2, c=1.$$

This integral pertains to the electrostatic field about an ellipsoidal body with principal semiaxes a, b, c .⁵ The ellipsoid is needle-shaped like an antenna or a probe. The classical approach transforms V into a standard form called an elliptic integral of the second kind and interpolates on two variables in published tables to get a numerical value. The following approach takes less time.

First transform the improper integral (\int_0^{∞}) into a proper

one by substituting, say, $u = (a^2 - c^2)/(1 - v^2) - a^2$ to get

$$V = \lambda \int_{\mu}^1 \sqrt{(1-v^2)/(v^2 + \alpha)} dv$$

where

$$\lambda = 2/((a^2 - c^2) \sqrt{a^2 - b^2}) = 2.00060018 \times 10^{-6}$$

$$\mu = c/a = 0.01$$

$$\alpha = (b^2 - c^2)/(a^2 - b^2) = 3.001200480 \times 10^{-3}$$

Now, as always happens when $a \gg b > c$, the integral is nearly improper because α and μ are both so nearly 0. We suppress this near impropriety by finding an integral in closed form that sufficiently resembles the troublesome part of V . One candidate is

$$\begin{aligned} W &= \lambda \int_{\mu}^1 dv \sqrt{v^2 + \alpha} = \lambda \ln (v + \sqrt{v^2 + \alpha}) \Big|_{v=\mu}^1 \\ &= \lambda \ln ((1 + \sqrt{1 + \alpha})/(\mu + \sqrt{\mu^2 + \alpha})) \\ &= 8.40181880708 \times 10^{-6} \end{aligned}$$

Then

$$\begin{aligned} V &= W + \lambda \int_{\mu}^1 (\sqrt{(1-v^2)/(v^2 + \alpha)} - 1/\sqrt{v^2 + \alpha}) dv \\ &= \lambda \int_{\mu}^1 \left(\frac{W/\lambda}{1-\mu} - \frac{v^2}{(1 + \sqrt{1-v^2}) \sqrt{v^2 + \alpha}} \right) dv \\ &= 7.78867525 \times 10^{-6} \pm 1.3 \times 10^{-14} \end{aligned}$$

after seven minutes at **FIG 8**. Don't worry about $\sqrt{1-v^2}$ as $v \rightarrow 1$ because the figures lost to roundoff are not needed and its infinite derivative doesn't bother the HP-34C.

Conclusion

A powerful mathematical idea has been placed at the disposal of people who will invoke it with fair confidence by pressing a button marked \int_y^x without having to understand any more about its internal workings than most motorists understand about automatic transmissions. Integrals that might previously have challenged the numerical expert and a big computer now merely amuse the scientist or engineer, and tomorrow they will be routine. And now those engineering students who do attend classes in numerical analysis need no longer be expected to memorize the names nor the remainder terms of quadra-

ture formulas but may instead be taught to use integration wisely.

Acknowledgments

Stan Mintz was the first to request an \int_y^x key for the HP-34C. Dennis Harms helped to select the algorithm and microprogrammed it into the calculator. Robert Barkan wrote the Handbook's two chapters on \int_y^x . Then Dennis helped to shorten this article. Working with these people has been a pleasure.

References

1. F.A. Willers, "Practical Analysis," translated by R.T. Beyers, Dover, New York, 1948, Chapter 3.
2. P.J. Davis and P. Rabinowitz, "Methods of Numerical Integration," Academic Press, New York, 1975.
3. A.H. Stroud and D. Secrest, "Gaussian Quadrature Formulas," Prentice-Hall, New Jersey, 1966.
4. W.M. Kahan, "Personal Calculator Has Key to Solve Any Equation $f(x)=0$," Hewlett-Packard Journal, December 1979.
5. J.A. Stratton, "Electromagnetic Theory," McGraw-Hill, New York, 1941, pages 201-217.

William M. Kahan



William Kahan is professor of mathematics and computer science at the University of California at Berkeley. An HP consultant since 1974, he has helped develop increasingly accurate arithmetic and elementary functions for the HP-27, 67/97, 32E, and 34C Calculators and the HP-85 Computer, financial functions for the HP-92 and 38E/C, and other functions for the 32E and 34C, including \int and SOLVE for the 34C. A native of Toronto, Canada, he received his BA and PhD degrees in mathematics and computer science from the University of Toronto in 1954 and 1958, then taught those subjects at Toronto for ten years before moving to Berkeley. A member of the American Mathematical Society, the Association for Computing Machinery, and the Society for Industrial and Applied Mathematics, he has authored several papers and served as a consultant to several companies. He is also co-author of a proposal to standardize binary floating-point arithmetic that has been adopted by several microprocessor manufacturers and is soon to be promulgated by the IEEE. He is married and has two teenage sons.

Address Correction Requested
Hewlett-Packard Company, 1501 Page Mill
Road, Palo Alto, California 94304

HEWLETT-PACKARD JOURNAL

AUGUST 1980 Volume 31 • Number 8

Technical Information from the Laboratories of
Hewlett-Packard Company

Hewlett-Packard Company, 1501 Page Mill Road
Palo Alto, California 94304 U.S.A.
Hewlett-Packard Central Mailing Department
Van Heuven Goedhartlaan 121
1180 AM Amstelveen The Netherlands
Yokogawa-Hewlett-Packard Ltd., Suginami-Ku
Tokyo 168 Japan

0200032503 E&E HARR & JA00
MR JULIAN A HARRIS
CHAYO ELECTRONICS LTD
P O BOX 2807
PENSACOLA FL 32503

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

CHANGE OF ADDRESS . . . To change your address or delete your name from our mailing list please send us your old address label. Send changes to Hewlett-Packard Journal, 1501 Page Mill Road, Palo Alto, California 94304 U.S.A. Allow 60 days.

HP Archive

This vintage Hewlett-Packard document was
preserved and distributed by

www.hparchive.com

Please visit us on the web!

On-line curator: John Miles, KE5FX

jmiles@pop.net

