VOLUME 11 □ NUMBER 2 IEEE spectrum FEBRUARY 1974

articles

departments

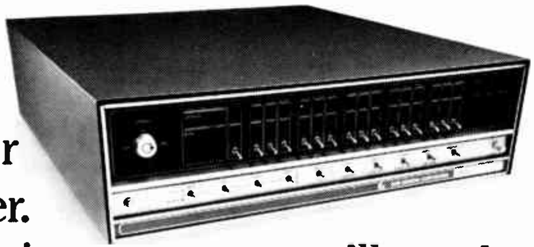
When we were designing our first computer, we decided to put as big a subsystem as we could on each PC board.

Within three years, the accepted way to build small computers was the Data General way.

And when we were working on our second computer, we made it completely compatible with our first. So all our software and peripherals could run on both computers. And the interfacing would be just as easy.

And we held on to that idea. The nine computers we've made since then are fully compatible, too. For the same reasons.

In other words, we invented what's now known as a "family" of small computers.



We're at it again with our Nova 2. Only this time we're helping you get your products out into the marketplace faster.

We've given the Nova 2 the operating systems you'll need to develop your software and run your hardware. So you won't have to come up with your own.

We're offering low-cost 16K memory so your programmers can work in high-level languages. And write software faster.

We have the general purpose interfaces you'll need to make that job go faster.

And we have the support organization you can call anytime you need help.

Making computers easier for you to work with isn't exactly standard in the industry.

Someday it may be.

At Data General, that day is now.

DataGeneral

The computer company you can understand.

Data General Corporation Southboro, Massachusetts 01772, (617) 485-9100.
Datagen of Canada Ltd., Hull, Quebec (819) 770-2030 Data General Europe, Paris, France 504-23-44.

Circle No. 1 on Reader Service Card

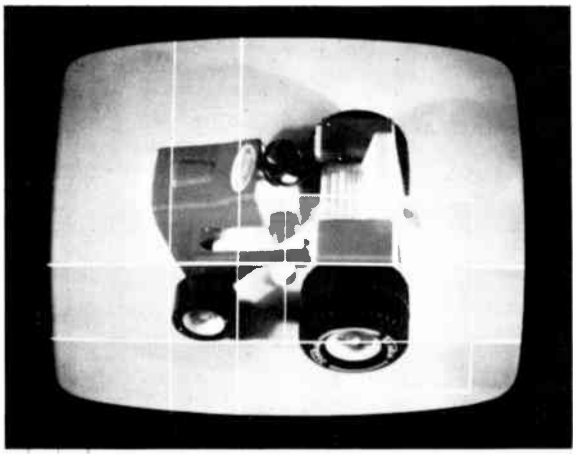
the cover

The assemblage that keynotes Spectrum's 48-page computer report hangs in the office of the Director of Communications, General Telephone and Electronics Information Systems. GTE commissioned artist Vin Giuliani to construct it.

spectrum

EDITORIAL STAFF *Editor:* Donald Christiansen

Managing Editor: Ronald K. Jurgen; *Senior Staff Writer:* Gordon D. Friedlander; *Staff Writer:* Marce Eleccion; *Associate Editors:* Howard Falk, Gerald Lapidus, Don Mennie; *News Editor:* Evelyn Tucker; *Contributing Editors:* Alexander A. McKenzie, Michael F. Wolff; *Copy Editor:* Ellis Rubinstein; *Editorial Assistants:* Ann L. Battiste, Stella Grazda; *Production Manager:* Ruth M. Edmiston; *Art Director:* Herbert Taylor; *Assistant Art Director:* Janet Mannheimer; *Technical Graphic Artist:* Morris Khan; *Art Assistant:* T. Dale Goodwin; *Business Manager:* Carl Maier



IEEE Publications Operations

IEEE PUBLICATIONS BOARD

F. S. Barnes, *Chairman*

		V. P. Kodali
J. J. Baruch	J. M. Early	R. W. Lucky
W. R. Beam	R. C. Hansen	J. E. Rowe
Donald Christiansen	A. L. Hopkins	A. C. Schell
Charles Concordia	S. J. Kahne	W. J. Spencer

STAFF

Elwood K. Gannett, *Director, Publishing Services*
Patricia Penick, *Administrative Assistant to the Director*

H. James Carter, *Manager, Editorial Services*
Lawrence Liebman, *Manager, Printing Services*
William R. Saunders, *Advertising Director*
Henry Prins, *Research Manager*
Carl Maier, *Advertising Production Manager*
Carol D'Avanzo, *Assistant Advertising Production Manager*

IEEE SPECTRUM is published monthly by The Institute of Electrical and Electronics Engineers, Inc. Headquarters address: 345 East 47 Street, New York, N.Y. 10017. Cable address: ITRIPLEE. Telephone: 212—752-6800. Published at 20th and Northampton Sts., Easton, Pa. 18042. **Change of address** must be received by the first of a month to be effective for the following month's issue. Please use the change of address form below. **Annual subscription:** IEEE members, first subscription $3.00 included in dues. Single copies $3.00. Nonmember subscriptions and additional member subscriptions available in either microfiche or printed form. Prices obtainable on request. **Editorial correspondence** should be addressed to IEEE SPECTRUM at IEEE Headquarters. **Advertising correspondence** should be addressed to IEEE Advertising Department, at IEEE Headquarters. Telephone: 212—752-6800.

Responsibility for the contents of papers published rests upon the authors, and not the IEEE or its members. All republication rights, including translations, are reserved by the IEEE. Abstracting is permitted with mention of source.

Second-class postage paid at New York, N.Y., and at additional mailing offices. Printed in U.S.A. Copyright© 1974 by The Institute of Electrical and Electronics Engineers, Inc. **IEEE spectrum** is a registered trademark owned by The Institute of Electrical and Electronics Engineers, Inc.

BPA

OTHER IEEE PUBLICATIONS: IEEE also publishes the PROCEEDINGS OF THE IEEE and more than 30 Transactions for IEEE Societies and Groups with specialized interests within the electrical and electronics field. Manuscripts for any IEEE publication should be sent to the editor of that publication whose name and address are shown on pp. 101-102 of the January 1974 issue of *Spectrum*. When in doubt, send the manuscript to E. K. Gannett, Publishing Services, at IEEE Headquarters, for forwarding to the correct party.

IEEE also copublishes, with the Institution of Electrical Engineers of London, England, *Electrical & Electronics Abstracts* and *Computer & Control Abstracts*, as well as their companion titles journals, *Current Papers in Electrical & Electronics Engineering* and *Current Papers on Computers and Control.*

MOVING? PLEASE NOTIFY US 4 WEEKS IN ADVANCE

Name (Please Print) ______________________

Address ______________________

City ______________ State/Country ________ Zip ________

MAIL TO: IEEE CODING DEPARTMENT
345 East 47 Street, New York, N.Y. 10017

ATTACH LABEL HERE

- This notice of address change will apply to all publications to which you subscribe.
- List new address above.
- If you have a question about your subscription place label here and clip this form to your letter.

spectrum FEBRUARY VOLUME 11 □ NUMBER 2

Spectral lines

Energy crisis: seeking the real alternatives, II

In January we addressed the allied problems of (1) determining the options available in the current and continuing energy situation, and (2) educating the general public to those options and their probable consequences.

Underlying this discussion was the premise that what the public thinks and how it reacts can significantly affect the overall situation. As an easily understood example, prediction of long lines at gasoline stations becomes, in part, self-fulfilling simply because drivers are gassing up three or four times more frequently than they normally would. And shortages at filling stations result in part because lots more gas is in auto tanks and owners' gas cans, not in station storage tanks.

On a grander scale, the reactions of thought leaders and pressure groups among the citizenry can influence the Government to assume certain postures with respect to global politics—which in turn may aid or hinder the flow of oil to the U.S. And the interactions of various interest groups can spawn new, unanticipated problems.

While there may be little reason to question the goal of educating the public on its options, a serious question does arise concerning whether the technological community can, indeed, define the options and attach to them possible consequences with any degree of certainty. Do we not have enough difficulty in determining what will be the probable effects of the energy crisis, at its current level of complexity, on our own electrical/electronics industries?

Already, the city of Los Angeles has enacted emergency legislation that would cut back consumption of power supplied by its municipal utility company. West Coast semiconductor manufacturers are worried that arbitrary cutbacks could do irreparable harm, as, for example, in the case of diffusion tubes which cannot be switched off and on at will. And customers of computer service bureaus are wondering just how likely are the possibilities of power blackouts damaging transistors, rotating memories, and read/write heads, as has been predicted by the service bureaus themselves in hearings before the California Senate Public Utilities and Corporations Committee.

Engineers concerned with possibilities such as those just mentioned may feel their problem unique—that new regulations and legislation touch them but not the power utilities themselves. Such is not the case, as underscored in a statement filed with the Federal Energy Office by the Edison Electric Institute on December 20. It concludes, concerning the allocation of middle distillates for use by the electric utility industry, that proposed mandatory allocation regulations do not provide priority status to the electric utility industry as intended by the Emergency Petroleum Allocation Act of 1973. Electric utilities have been classified in the lowest categories for middle distillate and residual fuel oils, the report asserts, noting that the utilities are now using about 100 million barrels of middle distillate fuel oil each year, most of it for combustion turbines and internal combustion engines. Such fuel oil is also used as ignition fuel for coal-fired generating units and for flame stabilization under low-load conditions.

The availability of residual fuels is just one of the many complexities facing the power industry that place it in the position of brother-in-arms with the electronics industries.

On the other hand, what bodes ill for some industries may prove just the opposite for others, like the communications industry. For now the incentives to communicate instead of travel are enhanced. Telephone utilities and video and data transmission services may come into even greater demand. And what about the semiconductor industry? Since solid state is an energy-saving technology, is it not possible to provide incentives to manufacturer and customer alike to "go solid state" faster, and in ways not previously considered? In this regard, one recalls that the Japanese Government subsidized all-solid-state television for domestic consumption because, compared to hybrid sets, it saved energy.

Today, microprocessors can be exploited to control energy-producing or energy-consuming processes at their sources to increase efficiency of energy usage. And, compared to cores, semiconductor memories are conservative of power. These examples suggest that electronics technology, if not itself stifled by materials or energy shortages, may operate in a positive mode insofar as the overall energy picture is concerned.

Such optimistic possibilities notwithstanding, in view of the many energy scenarios that can be constructed based on known options, it is clear that the energy crisis cannot be readily defined. If it could be, it would doubtless be representable by some multiloop nonlinear feedback system—which, unfortunately, is itself time-dependent.

In the meantime, we technologists shall have to address the situation piecewise, the best way we can—in pieces small enough to handle, but large enough to have lasting results. *Donald Christiansen, Editor*

Computer report I

Hard-soft tradeoffs

Large-scale integrated gates and read-only memories are displacing software from many of its traditional roles

From the system cost viewpoint, hardware and software have exchanged roles during the past two decades. In the early '60s, hardware used to account for well over 50 percent of system costs. Today, the hardware portion is substantially below 50 percent and that portion is declining more rapidly today than in the past.

There has been much attention given to the idea that large-scale integrated (LSI) circuits can economically provide computer functions that would otherwise be performed by software. Experience had demonstrated that this is indeed the case. In fact, a wide variety of such functions have been and are being implemented in hardware form.

Two all-hardware computer systems—SYMBOL and the HP-35 calculator—are discussed in separate sections of this article, as key examples of the extent to which hardware has been finding new roles in computing. But the role of hardware is also increasing in many more-conventional computer systems.

Hardware is used in many roles

Simple arithmetic functions—such as binary addition—were traditionally hardware implemented. Today, efficient computations for such complex functions as floating point arithmetic are being widely implemented in hardware form, and this is economically feasible because of the low cost of LSI hardware. For example, a floating-point hardware unit manufactured for use with a popular minicomputer—the Nova 800—reduces the execution time of a floating-point divide from about 900 μs to about 40 μs.

Howard Falk Associate Editor

And hardware substitutes, for functions traditionally performed by software, are finding applications far beyond arithmetic computations. Fast Fourier transform analysis, for instance, can be done using software and a general-purpose computer, but it is much more efficiently performed by hardware specially designed for FFT calculation.

Memory operations. Memory management seems to be one of the most important newer hardware application areas. The SYMBOL computer system, discussed in the box labeled "Probing the limits of hardware," was a pioneering effort that helped to demonstrate the feasibility of hardware memory management.

Allocating, reclaiming, and managing complex memory resources in systems using combinations of core, semiconductor, and disk memories is a very complex task. Many existing software-operated memory systems are the end-product of a long design and

(Text continued on page 38)

Computer system software features found suitable for hardware implementation

Memory allocation
Memory reclamation
Virtual memory management
Paging
Segmentation: absent segment interrupt
Memory and data protection
Stack operations
Address generation
Indexing
Indirect addressing
Storage protection

Multiple precision arithmetic
Decimal multiply and divide
Floating point arithmetic
Sorting
Data manipulation algorithms

Program linking and binding
Program relocation
Data relocation
Data structure
Format checking
Character string manipulation
Data-type conversion

Symbolic addressing
Variable field lengths
Variable data structures
Alphanumeric field manipulation
Context switching
Emulation
Queues
Links
Compilation
Task dispatching
Next software instruction fetch

Interrupts: interrupt checking
Trap catchers
Peripheral data transfer
Time-sharing supervision
Text editing
Control command instructions
Format checking
Peripheral data transfer

Parity checks
Error-control coding
Automatic retry
Automatic diagnosis

Probing the limits of hardware

All-hardware memory management was a key aim of the SYMBOL project from its inception in the early 1960s. Rex Rice and his colleagues at the Fairchild R&D Laboratory were convinced then, that the potential computing power of hardware had not yet been fully tapped, so they decided to build an experimental working system to explore the limits of computer hardware functional capabilities.

After six years of development, construction, testing, and operation at Fairchild and, later, at Iowa State University, SYMBOL had demonstrated the feasibility of constructing a high-level language, virtual-memory time-sharing system that can operate entirely without benefit of system software. To see just how this has been accomplished, let's take a look at the operation of the SYMBOL system.

How SYMBOL works

Users write programs for SYMBOL in a language called SPL—a high-level language somewhat similar to Fortran, Algol, or PL-1. Compared to these more conventional languages, SPL is more oriented towards users needs and less oriented towards controlling machine computation processes. For example, SPL has none of the word-length limitations imposed on the user by the other languages, and SPL makes no distinctions between data types such as real, integer, or Boolean.

For most computers, the machine and the language are designed quite separately and then brought together late in the design process. In the SYMBOL system, the starting point for the design was, in fact, the SPL language.

Unlike general-purpose computers that can execute many languages, the SYMBOL system can execute only SPL—but this single language is handled very efficiently. One of the main objectives of the SYMBOL system designers was to find out just how efficiently a powerful language could be executed.

Several distinct hardwired processors work together in the SYMBOL mainframe.

The *Central Processor* unit was designed specifically to execute SPL programs. It carries out arithmetic operations and manipulates character strings as well as performing such functions as array-referencing and procedure-activation.

For greater efficiency, the Central Processor executes preprocessed versions of programs. These are produced by the *Translator Processor*, which accepts the user's programs in character-string form (source code) and produces object code, ready for the Central Processor. Symbolic references in source code are replaced by memory addresses in object code. Also, in object code, expressions are rearranged into "Polish postfix" order that lends itself to efficient execution using a last-in first-out stack.

A similar process takes place in almost all conventional computers. However, in SYMBOL, the translation is performed separately—freeing the Central Processor for program execution. Furthermore, in SYMBOL, the object code is very similar to the source code, and the translation process is relatively simple to implement in hardware form.

Together, the *Input-Output Processor* and the *Channel Controller* handle the loading and editing of programs and the input and output of character strings as directed by user programs.

The *System Supervisor* coordinates the other SYMBOL processors and multiplexes tasks among them from up to 31 users. The Supervisor manages the transition of each task from one state to the next.

At the heart of the system is the *Memory Controller*. In addition to the conventional functions of referencing and replacement of words in memory, this unit also provides the automatic allocation, link tracing, and memory reclamation functions that are usually performed by software in other systems.

Virtual memory for the system is derived from a small core memory and a larger drum memory, managed by a drum controller and a page table that maps virtual memory addresses into core memory addresses. A section of the System Supervisor directs the drum controller to bring pages in and out as needed. All these functions are performed by hardware.

Machines for specialized problems

One major objection to the SYMBOL approach is that use of hardware implies a lack of flexibility since it is harder to modify than software. However, if flexibility is needed to correct serious design errors—such as those in basic data structure organization—system programs can be as difficult to change as system hardware.

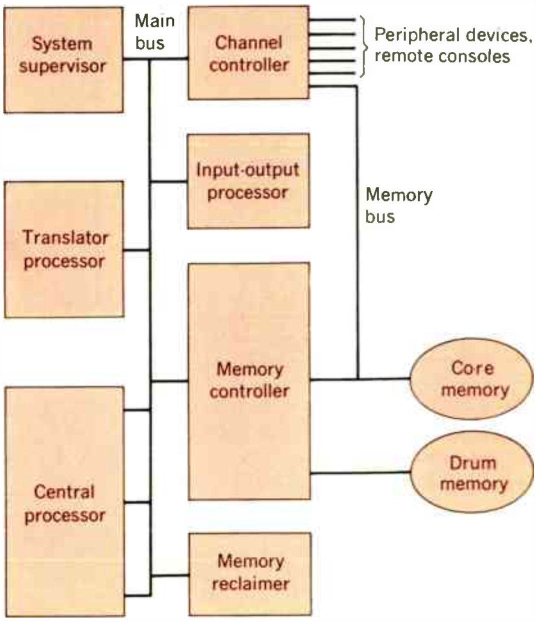
Computation tasks that were not foreseen when a system was designed often cannot be handled rapidly and inexpensively. But the solution to this problem need not be new software for a general-purpose computer. It may be more economical to use an entirely new machine designed for the task at hand.

When hardware was very expensive, it was necessary to make each computer so it could handle a great variety of user problems. Today, hardware is very cheap and there are many different machines to choose from. Through computer communications networks a user in Illinois can even do his computing on a machine in San Diego.

With many specialized machines available there is no need for overlapping general-purpose capability.

Hamilton Richards, Jr.
Iowa State University

SYMBOL system. Processors are shown as squares, memory components as circles.



Tradeoffs in electronic calculator design

One of the basic objectives, in the design of the Hewlett-Packard 35 and 45 electronic calculators, was to develop an architecture suitable for a family of machines, each of which would be directed to a specific group of users. Thus, the initial HP-35/45 designs were "slide rule" calculators for engineers and scientists, while the HP-80—released in 1973—was a business-financial calculator. Operating speed, fabrication cost, and design-production turnaround time were driving factors that influenced many of the design choices.

The architecture finally chosen—illustrated by the block-diagram of the HP-35—was implemented entirely in hardware, using two custom-wired gate-logic chips and three read-only memory (ROM) chips.

Software was too slow

It makes no difference how I start thinking about the question, "Why did you implement the HP-35/45 all in hardware instead of using software?" The answer always comes out the same: speed is the primary reason. We used hardware because software is too slow.

P-channel MOS integrated circuits, like those used in the HP-35 and 45, can be operated at clock frequencies of 50–200 kHz. Even at 200 kHz, the add time of these machines is 60 ms and the multiply time is 400 ms. These times are adequate for manual arithmetic, but are completely unacceptable for software routines to evaluate elementary functions like sin *x* or cos *x*, or for square roots.

For example, consider the time needed to evaluate sin *x* using a continued-product version of a MacLaurin series. To obtain accuracy of 10 significant digits from $-\pi/2$ to $\pi/2$, requires 9 multiplications, 8 divisions, and 8 subtractions. More than seven seconds would be needed for this computation, without even considering the time needed to convert from degrees to radians, resolve large arguments, and correctly assign the sign.

With a somewhat more sophisticated program, the time to compute sin *x* might be reduced to five seconds but, using ROM-microprogrammed routines, the computation takes less than one second.

The technique we actually used, called a cordic algorithm, relies on the ability of the microprogram to manipulate the internal registers of the calculator. While the relationship between registers is maintained by a simple stored algorithm, one of these registers is driven towards the input argument *x*. Because of the fixed algorithmic relationship another register is automatically forced towards sin *x*. Almost all cordic routines can be executed in 2 or 3 divide times.

For the electronic calculator designer, microprogramming allows great flexibility compared to keyboard-function programming. The designer is able to select any of the individual bits within a number with the same ease he can perform an addition on an entire register. Similarly, he can extract the integer or fractional parts of a number, or multiply a number by 10 simply by changing its exponent.

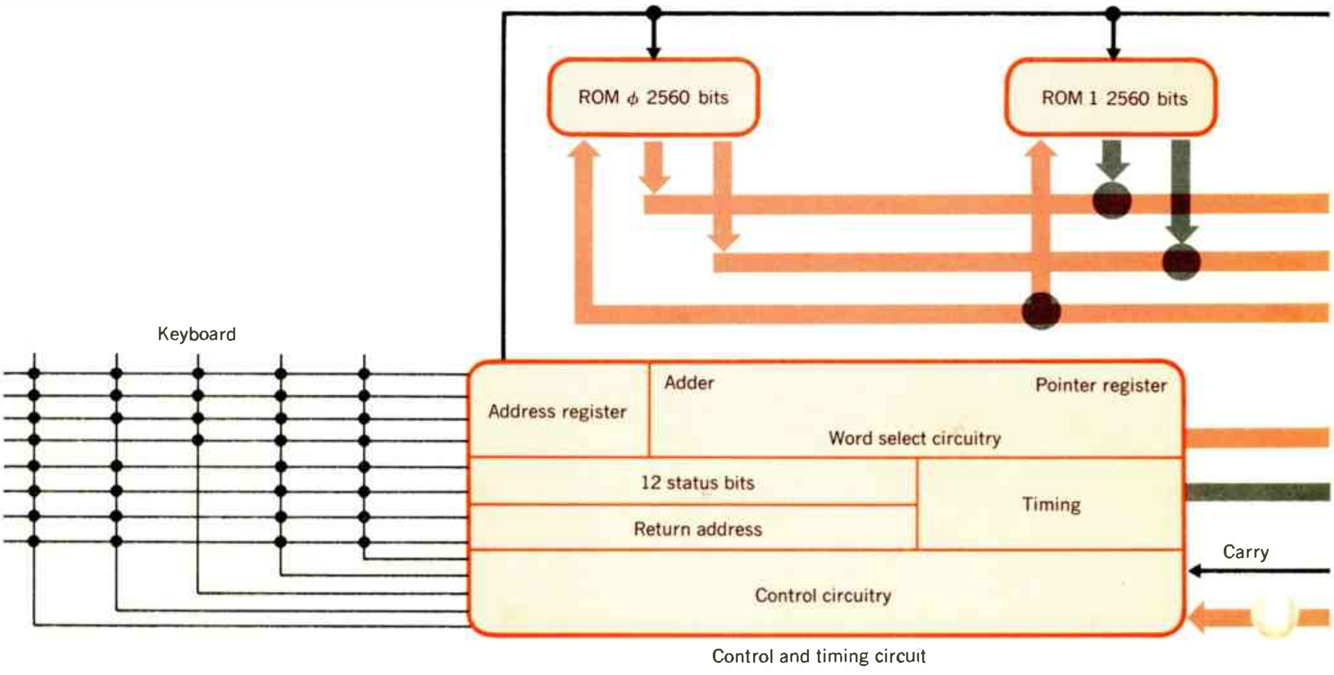
ROM-gate tradeoffs

Large-scale integrated circuit (LSI) chips with custom-wired gates operate at speeds about twice as fast as those of LSI-ROM circuits. As we have seen, speed can be a key design consideration—especially where time-consuming algorithms are used. Even the difference between a half second and a full second is quite noticeable to users.

If speed were the only consideration, custom-wired gates would have been used. However, the economics of integrated circuits dictate the use of as many copies as possible of each individual circuit chip. Using all custom-wired gate chips, each of the Hewlett-Packard calculators would have been a little faster and the chips would have been somewhat smaller, but the overall investment would have been significantly higher for those lower-volume unique chips in each calculator.

What we did was to separate out the logic functions we knew would be common to all the calculators. These were implemented in the form of two custom-wired LSI chips: the control and timing circuit, and the arithmetic and register circuit shown in the HP-35 block diagram.

The remaining logic, including all functions that were to be unique to each individual calculator design, was implemented in LSI-ROM chips. On these chips, the logic bit-pattern is determined by the final processing mask. The



unfinished ROM chips, ready for that final processing step, are standard high-volume units.

By using this mix of custom-wired and ROM chips, we achieved an architecture that gave reasonable processing speed along with flexibility and—equally as important—a shorter design and production turnaround time.

Turnaround time: a competitive necessity

If historians ask, 100 years from now, why a certain calculator design using random wired chips was unsuccessful, the answer will probably be that it just wasn't ready for production in time. There is a fantastic amount of market pressure in calculator design. Furthermore, the privilege of being first brings with it the right to set the ground rules.

If you make a design error in a chip custom-wired with gates, the turnaround time needed to change the chip is substantial. This time is at least five times greater for a custom-wired gate chip than it is to change the bit-pattern on a ROM chip, and even ROM turnaround time can sometimes be a problem. Depending on the production situation and the backlog of work semiconductor vendors have on hand, ROM turnaround time may be two to three weeks, or it may stretch to a couple of months.

In general, if large quantities of a particular logic chip are wanted—given sufficient design and development time—custom-wired gate chips can usually be designed for lower cost than the equivalent ROM versions. But the extra time needed to design and debug a gate-chip can be disastrous in a competitive market like the one for electronic calculators.

Add to this the fact that prices of standard ROM chips are continuing to drop—much more rapidly than the prices of custom-wired logic gates—and the pressures for using ROM implementations are overwhelming.

A healthy discipline for the designer

System designers tend to worry a lot when a computational routine is about to be locked up in hardware form. One bug in the system can mean recall of the product because there is no such thing as a software fix. In the case of products like the HP-35 or -45 electronic calculators, there is the nightmare vision of thousands of units being shipped back to the factory.

With such possibilities before him, the designer becomes extremely scrupulous about verifying his algorithms and the circuit implementation of each function. The end result is higher quality routines and circuitry. But, the design process is often a laborious one.

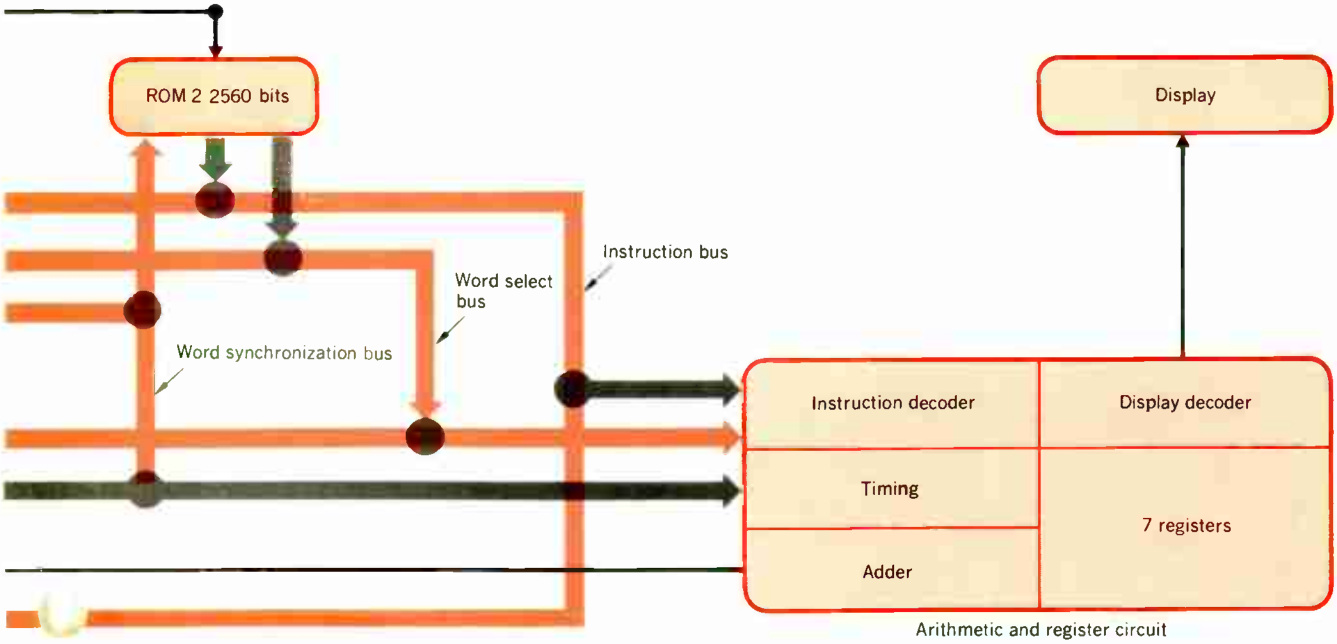
The initial euphoria a hardware designer feels when he is first given the freedom to use microinstructions can rapidly fade as he realizes he can now make mistakes in innumerable new ways. With the flexibility of microprogramming comes an ability to assemble a whole wheelbarrow full of new parts in seemingly endless combinations, but many of these can create subtle errors.

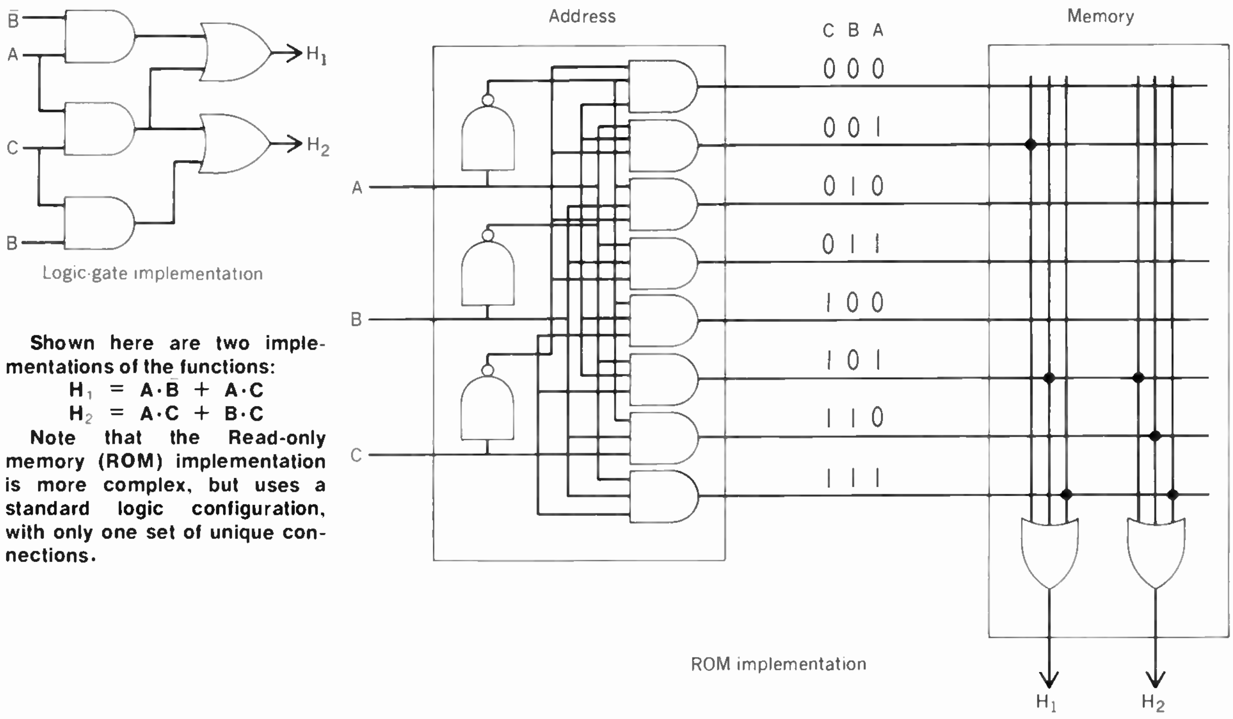
There is one microprogramming question I have used to check out new hardware designers. My question to them sounds simple: "Assume that numbers are expressed in scientific notation. Can you describe an efficient algorithm for adding two numbers?"

Those on the right track realize quickly that if the exponents differ by more than *n* digits—where *n* is the length of the numbers used in the computation—then it is a waste of time to perform the addition, because the sum will be the number with the more positive exponent. Most designers progress this far, but few inexperienced microprogrammers escape the trap that occurs when the number with the more positive exponent equals zero. I have seen more than one machine that gives the sum of 10^{-30} plus zero, as zero. An experienced designer will handle the problem by including a check to see if the most significant digit of either number is zero. If it is, then the sum is the other number.

Tom Osborne
Hewlett-Packard
Palo Alto, Calif.

Basic architecture of Hewlett-Packard calculators includes two custom-wired gate chips for control and arithmetic, as well as three read-only memory (ROM) chips.





Logic-gate implementation

ROM implementation

Shown here are two implementations of the functions:

$$H_1 = A \cdot \bar{B} + A \cdot C$$

$$H_2 = A \cdot C + B \cdot C$$

Note that the Read-only memory (ROM) implementation is more complex, but uses a standard logic configuration, with only one set of unique connections.

redesign process. Because of the complexity of the programming tasks and the constant need to maintain compatibility with previous systems, such software tends to become a relatively inefficient patchwork of old and new pieces.

Hardware memory management allows the system designer to incorporate the best features of existing software systems with a technique that uses no memory space for stored supervisory programs, employs the most efficient available methods, and operates at hardware speeds.

On a smaller scale, hardware techniques are increasingly used to format data for disk memories and control disk operations. In stack and queue operations—where part of the computer memory provides special controls for efficient processing—hardware is likewise replacing software.

Any computer operation performed frequently and repetitively is a possible candidate for hardware implementation. For example, error control functions, like cyclic redundancy character-checking for magnetic tapes, are now performed by bipolar LSI chips. At Iowa State University, an algorithm for recognizing hand-printed characters—programmed in the PL-1 language on an IBM 360/65—took 7 seconds to run. A hard-wired processor ran the same algorithm in 7 ms—a gain of 1000 to 1 in processing speed.

In computer peripheral devices, hardware functions are helping much more than in the past to lighten the load on central processor software. Communications controllers use modular multiplexer hardware to scan groups of lines in parallel, and in some systems, input-output bus signals are directly interpreted by hardware to activate such peripherals as phototypesetting machines.

Taking a more general approach, computer programs can be analyzed as they run, to determine which groups of instructions are used most often and consume the most program time. These groups can then be implemented by special-purpose hardware, to be called into operation whenever the main program needs them. For example, if the programming language used were Fortran, the commonly used argument-transfer routines might best be put in hardware form because these routines are so time consuming. Similarly, subscript calculations might be put in hardware, particularly if there was a good deal of array manipulation to be done. At Burroughs Corp., when fine-tuning techniques were implemented, programs that typically ran for 24 minutes were reduced to about 6-minute run times.

Reasons for using hardware

Sheer computational speed is often the basic reason for substituting hardware for software. For complex computations, involving a large number of operations, the speed improvement of hardware implementations can be considerable. This is true of specialized operations such as sorting and fast Fourier transforms; it is also true for general operations like floating-point arithmetic.

Enhanced system reliability is another frequent goal of hardware use. Hardware is less subject to inadvertent modification, and hardware monitoring of system and software operation offers practical debugging assistance.

According to Hamilton Richards of Iowa State University, experience with the all-hardware SYMBOL system suggests that there are significant payoffs in starting computer system designs with thorough exploration of all possibilities for hardware implementations. A hardware-first orientation focuses attention, at an early stage, on crucial system design questions that otherwise tend to be postponed—until late in the design process—with painful consequences. In SYMBOL, problems of memory allocation and garbage-collection, for instance, received early attention and this led to very satisfactory solutions.

The choice: custom-wiring or ROMs

As the integrated circuit art has matured, and larger arrays of circuits have become available in economical chip form, two basic hardware methods for implementing traditional software functions have evolved: custom-wired logic gates and read-only memories (ROMs).

Many software functions can be implemented by circuit gates that perform elementary Boolean logic, and these gates can be wired together with connection patterns determined by the particular function that is to be performed. When the gates are produced in integrated circuit form, the result is a custom-wired chip capable of performing only those specific functions for which it was designed.

Integrated circuit chips can also be produced as standard arrays of read-only memory (ROM) elements. Desired connections between these elements are made in the final processing step of the chips. The result is a stored pattern of zeros and ones which can function in the same way as a stored program in a traditional computer memory. The sequence of small logical events that are set into motion by this stored pattern is called a microprogram.

Custom-wired gates can perform computation functions faster than ROMs, and both of these hardware techniques are considerably faster than software. On the other hand, considerable design-time is necessary to lay out the connections for a complex custom-wired function. ROM design is a bit easier, but still requires greater manpower investment than an equivalent software routine would. This is in part due to microprogram coding difficulty. To effect a simple operation, such as adding two numbers, for instance, every step, including such details as operand fetch and data transfer, must be specified in the microprogram. Generally three to ten microinstructions and, in some cases, up to 50 microinstructions must be executed to perform an operation that would otherwise be specified by a single software instruction.

Since a capital investment of thousands of dollars is needed to begin fabrication of a single custom-wired chip, these chips can usually be considered only for applications where they will get large-volume use, and where design turnaround times of several months can be tolerated. ROM chips come close to being standard semiconductor-house production items because only one custom-designed interconnection layer is needed. They can economically be used in small quantities, and can often be obtained with design turnaround times of a few weeks.

Microprogramming for speed with flexibility

An interesting example of the great flexibility of microprogramming is provided by the Burroughs B1700 computer. This protean machine was designed to act as if it were almost any computer the user wants, and to run a wide variety of programs and languages efficiently.

The general concept of a machine like the B1700 is a simple one. It contains a microprogrammable memory that controls its effective internal architecture so it can play the role of another (target) computer—one for which the user, presumably, has software already on hand.

In practice, hardware details can cause considerable performance problems, particularly if the microprogrammed machine and the target computer are quite dissimilar at the hardware level. Typical problems that have arisen in the past are those involving system synchronization, time delays, word length, and techniques for handling data overflow or discard. Many of these problems have been solved in the Burroughs design.

At present, microprogramming is an elite activity, performed effectively only by a small number of expert practitioners. The work is detailed, precise, time-consuming, and considerably more expensive than present-day software programming. But, computer manufacturers have found they can get dramatic improvements in system performance by converting software into microprogrammed form.

For example, at Hewlett-Packard part of the Fortran-8 package for HP-2100 minicomputers was converted to firmware, including such items as extended precision, call sequences, address pointers, and address compilation. With these items performed by firmware, benchmark measurements have indicated throughput increases of from 5 to 20 times. Similar improvements in system performance have been obtained by many computer manufacturers using firmware to implement their computer designs.

There seems to be general agreement in the industry that microprogramming will continue to find increasing use in computer system design. More controversial is the worth of microprogramming performed by computer users. One manufacturer feels that user-microprogramming is ineffective and capable of producing little, if any, improvement in performance. Companies like IBM and Digital Equipment Corp. tend to discourage those customers who want to do their own microprogramming.

Nevertheless, users continue to have strong reasons for using microprogramming. It is true that the computer manufacturer is likely to know the best ways to microprogram his system to make it most generally useful to a very broad spectrum of possible users. But that does not necessarily mean that the needs of every user will be best met by that system.

For example, control of a petroleum cracking operation may require checking a complex equation many times each second—a function that would be better performed by microprogramming than by software. It is unlikely that standard firmware supplied by a computer manufacturer could meet this user's needs as effectively as they could be met by custom-designed microprogramming.

With microprogramming, the user can also fine tune his programs. It has been said that many, perhaps most, user programs spend a large amount of their time running and rerunning a very small amount of their code. By microprogramming the often-used sections of such programs, significant improvements in performance have been obtained. But such results depend on intimate knowledge of the particular program and its application, and they cannot be realized unless the user gets customized microprogramming.

Information for this article was received from many sources. Special thanks are due to Michael Sporer (Prime Computer, Inc.) and Ming T. Miu (Honeywell Information Systems).

Computer report II

More power by networking

Whether computers communicate as equals or in a superior–subordinate mode, the outcome is usually positive

Computer networks have evolved for reasons of economics and efficiency to permit information resident in one computer to be shared with others, to enable large numbers of users to have remote access to a computing center, and to control the flow of information from one point to another. Not surprisingly, computer networks have evolved in much the same way as other types of networks in our society such as those in communications, power, and transportation. The evolutionary pattern has been to begin with pairs, which are expanded to chains, which in turn lead to series-parallel arrangements such as trees and loops. The networks are interconnected in a variety of ways—they may be based on superior–subordinate relationships, such as commonly found in task-oriented organizations (e.g., industrial or government networks), or else they may be predicated on a more democratic relationship because of more general goals (e.g., networks for research and education).

In all networks, communication is initiated by one computer. However, the ensuing transactions with respect to the initiator can be as a superior demanding a task be done by a subordinate or simply as a request for help.

C. Gordon Bell Digital Equipment Corp.

I. Overview of Network Structures

	Direct connection	Independent store-and-forward network
Cost	Cost small for simple networks	Additional machines needed
Limitations	Not applicable to large nets or long distances. Applicable to limited traffic matrix (e.g., STAR)	
Ease of design	Simple	Functionally independent of host and its operating system, but the net is costly and complex
Reliability	Based on redundancy	Can be made arbitrarily large and therefore reliable
Performance	Fixed performance which can be modified using dial-up to assist in overloads	A network can increase performance with more links
Applicability	For small minis	Large interconnection nets where communication costs are small fraction of overall cost

Some networks consist of a number of small computers used as an alternative to a single large computer. This is done to ensure that the computer structure is operational at all times. It also can simplify system development if functions are partitioned carefully. Such a network permits greater economy by selecting machines that are the most cost-effective for each part of the problem being solved, and then distributing the processes according to costs and capabilities. Such a network may be under control of either a large or small computer.

A key element in all networks is intercomputer and computer-to-terminal communications. A well-established trend is the use of minicomputers connected to larger machines to handle the communications line control functions, commonly called front-end processors. (The minis replace hardwired, inflexible communications control units.) In addition, the minicomputer may perform a variety of simple data processing tasks on a more cost-effective basis than a large machine. The tasks include editing, calculations, and interpretation of languages such as BASIC. Although such a simple pairwise intercommunication could hardly be called a network, there is a trend to use interconnected computers whose functions are specialized. Another similar trend is to use remote job entry computers as intelligent terminals, connected in the form of a one-level tree (or star).

Some of the tasks performed by minicomputers in controlling communications include transforming and controlling the communications lines and modems, controlling a wide variety of terminals (e.g., performing character code conversion, echoing, error checking, and speed selection), and multiplexing characters from multiple incoming lines into a single outgoing line to a host computer.

Network configurations

There are three basic types of network structures (Fig. 1): directly connected networks, store-and-forward networks, and hybrid networks. The direct-connected structure provides for a direct communications link between the transmitter and receiver. Most of today's networks operate in this manner because it offers the simplicity of a fixed communications pattern. The structure may be a one-level hierarchy (called a one-level tree or star), most often in the form of a central site connected to a number of satellites. Strictly speaking, a tree is not a network, since there are no closed loops for alternative routes among the links. The characteristics of directly connected networks are listed in Table I.

Store-and-forward networks contain separate switching computers, which provide buffering to assure that the networks operate smoothly regardless of the speeds or data formats of the machines involved. This type of network also permits any machine to communicate with any other in the network. The ARPA network (covered later in this article) is a good example of this type of structure and, in the author's opinion, large-scale networks will inevitably follow this approach.

A hybrid configuration is a direct-connected network, but with the intermediate computer nodes providing store-and-forward capabilities. This approach has seen little use.

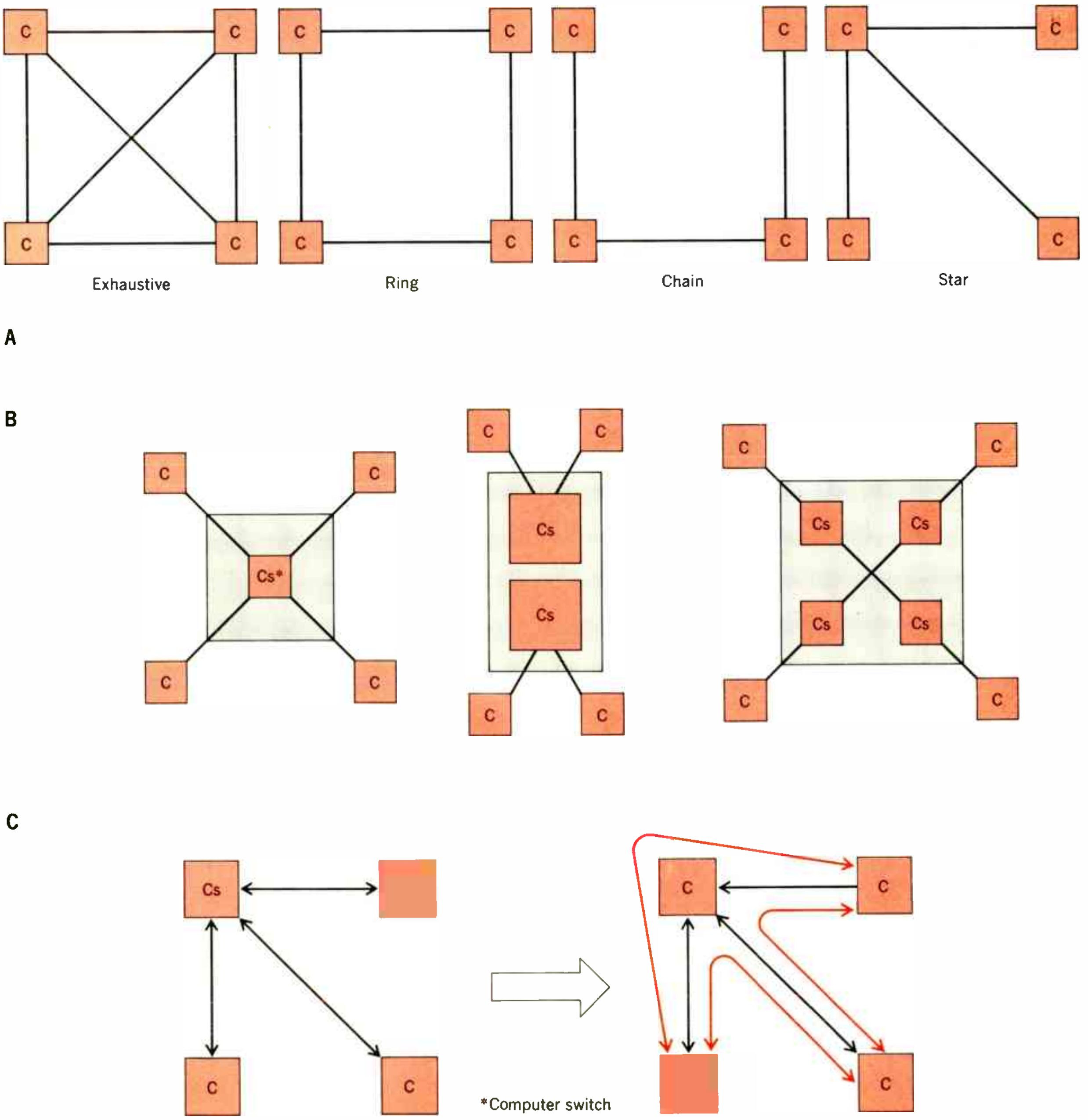
Considerations in network design

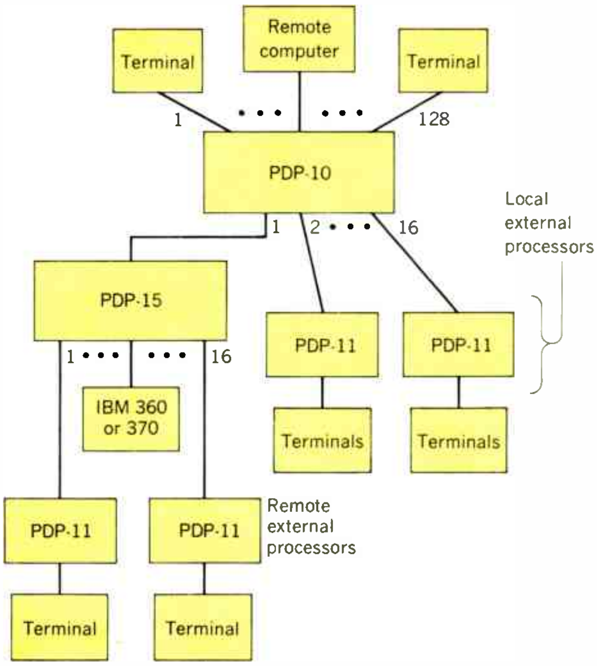
An important element to successful network design is a thorough knowledge of computer components, together with a proper definition of the information processing problem over the life of the network. The most important component parameters are link capacities, error rates, costs, and reliability. Processing capacity also is an important factor and it must be measured, not on the basic hardware, but in terms of the operating system and languages specifying the information processing problem.

The information processing problem should be defined both in terms of data processing needs and the location of hardware. The processing needs should be based on the number of statements, file sizes, and the delay time that can be tolerated for the processing services.

The physical location requirements are based on the economics of whether the processing equipment

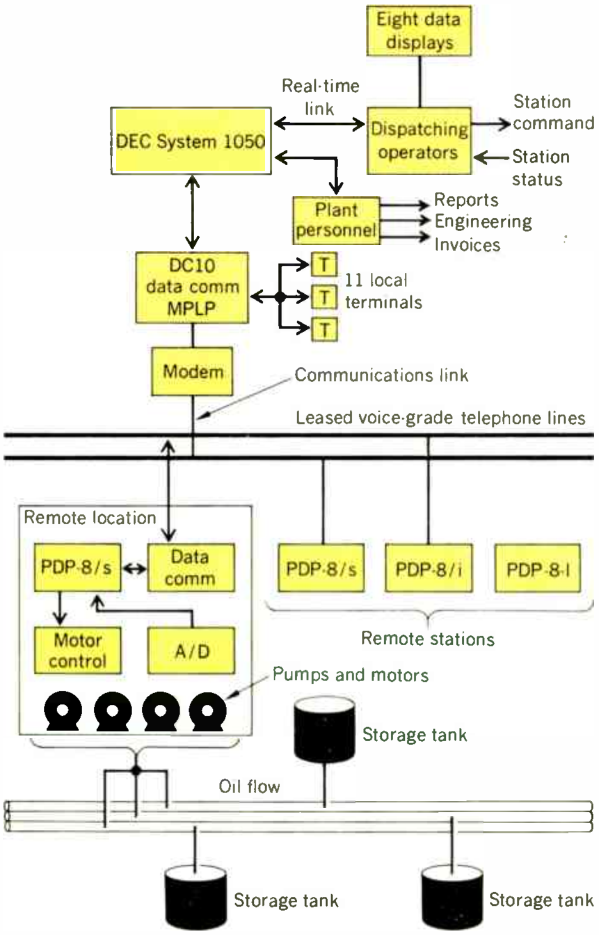
[1] Network structures. Direct-connected networks (A) may be configured in a number of different ways, but most often they take the form of a star (a central computer connected to satellites). Store-and-forward networks (B) contain independent switching computers to optimize the data flow and work assignments. In a hybrid configuration, (C), the store-and-forward function is performed by a single, dedicated channel in a directly connected network.





[2] Laboratory interconnection programming system is designed for control of diverse computers. At the heart of the system is a PDP-10, which functions as the central computer. Its purpose is to file, to provide central support of satellite computers, and to switch information.

[3] Canadian International pipeline network communicates with minicomputers over voice-grade telephone lines. The minicomputers are located at remote, unattended pumping and oil storage stations along the 1900-mile pipeline.



should be local or remote. For example, suppose it is desired to display the output of a flow-rate meter and there are switches with which an operator can enter the flow-rate limits. Processing of this problem can be handled entirely locally or at a remote computer—or possibly even distributed among several machines (if the information is needed elsewhere).

In designing a network, the criteria that usually apply to components—i.e. cost, performance, and reliability—also apply to networks. Several additional guidelines are:

- Minimizing the number of computers and computer types tends to decrease programming costs.
- Designing a network, where possible, with identical components allows a variety of functions for backup.
- Duplicating functionally specialized computers offers reliability.
- Unless the node is replicated or made into a duplexed tree, tree-structured (star) networks are not satisfactory for most networks that require high reliability.
- Communication links and the protocols that establish error-free transmission are probably the least understood by computer engineers.
- Never build a structure before competing, alternative structures have been considered.

Typical networks in use today

Several networks in use today illustrate the various network approaches. Those covered below are the DEC Laboratory Interconnection Programming System (LIPS), a simple tree structure; the Lawrence Livermore Laboratory system, a directly connected network with some switching (hybrid) capability; and the ARPA network, designed for computer network research.

The DEC LIPS System is a general purpose, PDP-10 computer for controlling a network of diverse computers. Strictly speaking, it is a one-level tree, which permits a number of independent satellite computers to be connected to a large central computer, as shown in Fig. 2. Its overall function is to file, to provide central support of satellite computers, and to switch information among the computers. The network has been implemented at Oak Ridge National Laboratory (IBM 360/91, 360/75, 2780, three Systems Engineering Labs 817/840s, seven PDP-4/7/15s), the Abbott Laboratory, Knolls Atomic Power Laboratory (two Control Data Corp. 6600s, one 7600), and Rolls Royce Engine Testing Laboratory (two English Electric KDF-9s, one IBM 360, eight PDP-8/11s).

The LIPS network has appeared in various laboratories that use a number of large, scientific computers. The functions being performed by the relatively large, general-purpose time-sharing system are:

- Scheduling and loading of the large satellite computers. In several cases (e.g., Oak Ridge and Knolls), the PDP-10 appears to other machines as a remote job entry terminal. In the case of Rolls Royce, where the PDP-10 schedules two KDF-9s, utilization of the latter was increased from 25 percent to 75 percent and the number of supporting operators was reduced from 22 to 10.
- Switching. Since there are several large machines available, a given job can be routed to the machine that can best handle load.

- Editing. Since the files reside in the PDP-10, they can be edited locally, avoiding costly movement of the files and trivial processing of them.
- Interactive processing of small jobs. These tasks are carried out in a manner akin to editing, using either interactive interpreters (e.g., APL) or load-and-go compilers (e.g., BASIC).
- Pooled specialized central facilities. Printing, punching, display, film reading, and specialized plotting are provided by the central facility.

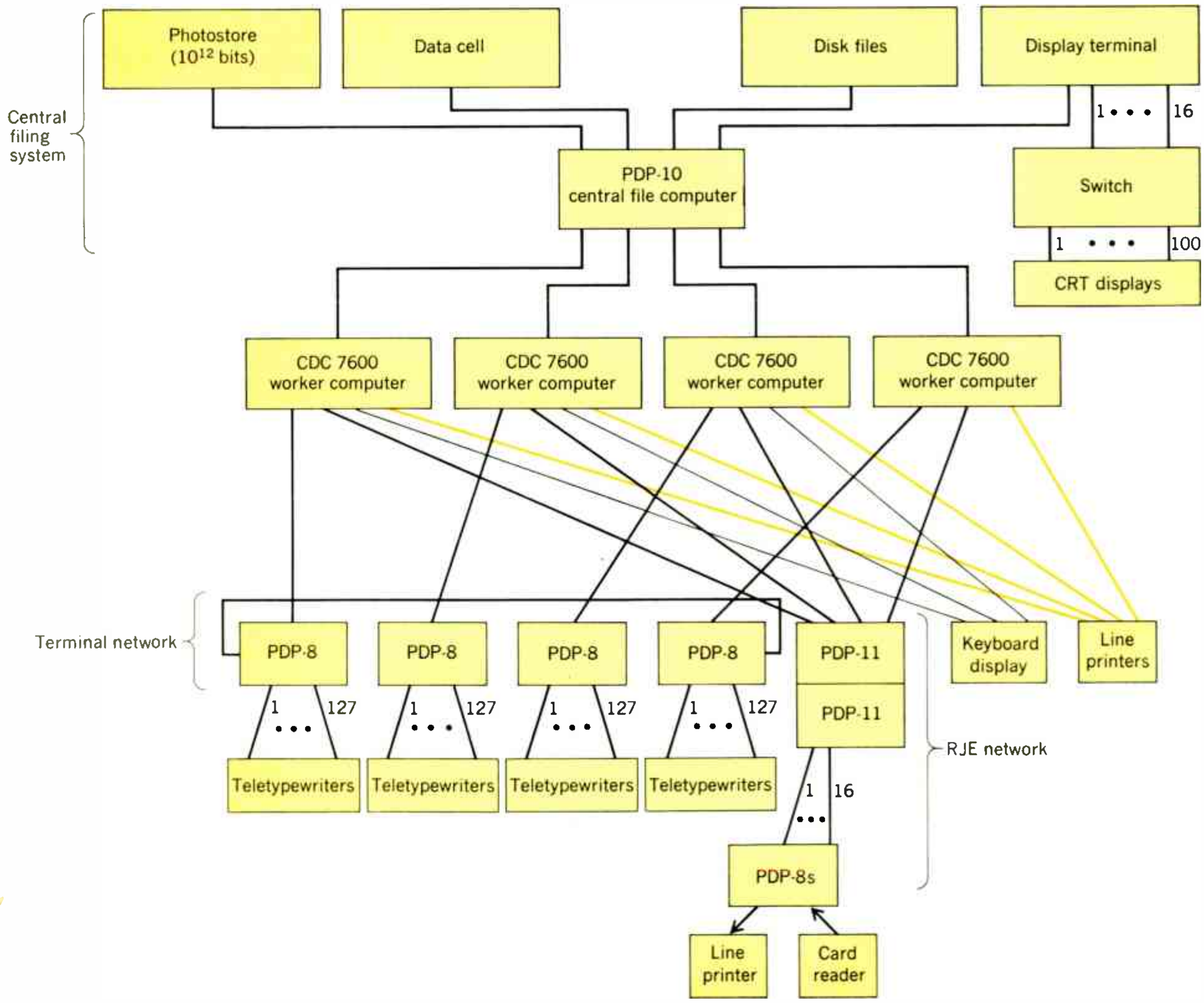
The LIPS approach has also been used in minicomputer networks. It has seen service in laboratories where minicomputer nodes are employed to meet real time and preprocessing demands. These structures are similar to the industrial control systems in which minicomputers control individual, real time processes. The need for such a network is predicated on the poor adaptability of a single, large computer to meet real time response demands; poor reliability of a single site; the high cost of simple, short wordlength calculations required in real time preprocessing; and the physically isolated inputs and outputs associated with a process.

The functions usually performed at a LIPS central computer site include:

- Management of a central data base. Usually a significant economy results from removing file devices from the local computers and placing them in a single site. Overall, there is less hardware and software for operating systems in the individual machines.
- Centralized editing and language translation (program preparation). Large machines, which contain large primary memories, usually provide the best facilities for minicomputer program preparation.
- Pooling of specialized facilities (e.g., printers, punches, tapes, displays, plotters).
- Significant computations that cannot be handled at the minicomputer nodes. This permits the use of more extensive languages such as COBOL and PL/1, which usually are not supported by minicomputers.

The Canadian Interprovincial Pipeline (IPL) network (Fig. 3) is a one-level tree in which 44 PDP-8 minicomputers are connected to a central PDP-10 computer via several polled, synchronous, communication links. The minicomputers are located at remote, unattended pumping and oil storage stations along the 1900 miles of the pipeline. Each minicomputer controls several large pumps, and monitors

[4] Lawrence Livermore Laboratory's Octopus network gives several hundred subscribers access to four worker computers and a central file computer. The terminal network accommodates 500 teletypewriters, 16 remote job entry consoles, keyboard-displays, and printers.



pressures, flow rates, gravity, viscosity, power, and other station parameters. The central site also contains a minicomputer to back up the central central computer, but with fewer monitoring functions. The central computer monitors and records the behavior of the pipeline and controls the individual sections. One advantage of such centralized control is that the amount of electrical energy supplied to move the oil can be optimized. Thus, the network has both local and global control.

An alternative approach would be to multiplex all inputs and outputs, and then transmit them to a central site for control. This would require that communications lines always be operational, whereas the existing structure carries out local control at each site without frequent intervention from the central control computers.

The computer utility

To date, computer networks have been built by special classes of users. However, the next logical step is an information utility for whole "communities," such as businesses, homes, and government departments, which would provide services such as credit card transactions, printed message delivery, news distribution, and library information retrieval. Such a utility network might take advantage of the economy of scale by assigning information processing tasks to machines best able to handle a given processing tásk or else they could distribute the work load among many machines for optimum efficiency. Computer networks already take advantage of geographic time zones to assign jobs to machines during off-hours.

In the nearer future, the existence of specialized data bases will begin to make program and file sharing desirable because of the excessive time and costs involved in shipping data in physical form and in the costs of maintaining and updating redundant files. For example, we might expect to see specialized networks for libraries as a first step in retrieval networks. Indeed, computer networks may eventually provide the only economical solution to the problem of maintaining and disseminating the information in libraries.

The Lawrence Livermore Laboratory's Octopus network, started in 1964, appears to be the earliest, and most general, of the general-purpose computer networks, because it provides a wide range of functions with different computers. Several hundred simultaneous on-line users are given access to four worker computers.

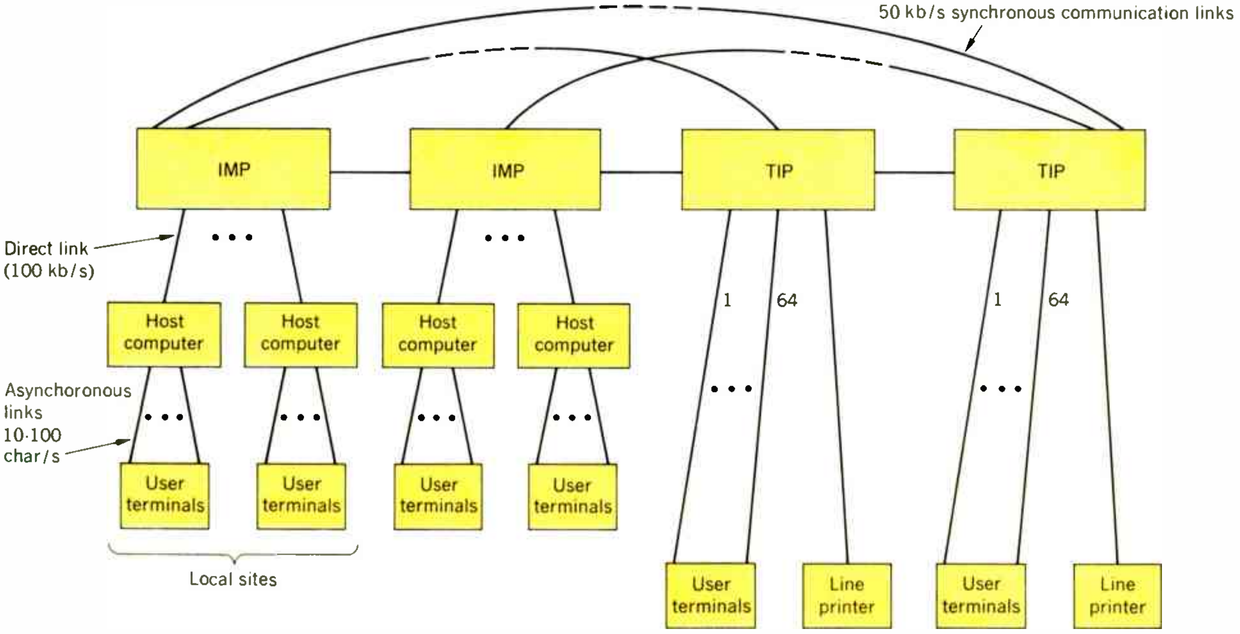
At the heart of the network (Fig. 4) are four large worker computers (three CDC 7600s and one CDC 6600) and a central file computer (a DEC dual processor PDP-10) with access to a data cell, disk files, and a 10^{12}-bit IBM photostore (controlled by an IBM 1800 computer).

Hardwired 10-Mb/s links connect the large computers to the file computer. The terminal computers and the large file are connected to the switching computer via high data rate links. The terminal network connects over 500 teletypewriters to the four worker computers via four PDP-8s, and the remote job entry (RJE) network connects up to 16 remote job entry consoles. In addition, a set of keyboard displays and line printers are connected to all the large computers in the system.

The main purpose of the network is to permit any user to have access to any of the large computers so that the network will operate even with a number of machines down.

The Advanced Research Projects Agency (ARPA) network was conceived in 1968 and placed in operation at four sites in 1969. As of January, 1973, 34 centers had been connected.

[5] The ARPA network structure consists of large-scale central computers (hosts) and interface computers (IMPs and TIPs). The TIPs were added after the network was initially designed.



For more on networks

Bell, C. G., Habermann, A. N., McCredie, J., Rutledge, R., and Wulf, W., "Computer networks," *Computer*. Oct./Sept. 1971, pp. 14–23.

Cady, G. M., and Luther, G., "Trade-off studies in computer networks," *IEEE Compcon*. 1973, pp. 147–150.

Frank, H., and Frisch, I. I., *Communication, Transmission, and Transportation Networks*. Reading, Mass.: Addison-Wesley, 1971.

Ornstein, S. M., Heart, F. E., Crowther, W. R., Rising, H. K., Russell, S. B., and Michael, A., "The terminal IMP for the ARPA computer network," *Proc. AFIPS SJCC*. 1972, vol. 40, pp. 271–280.

Roberts, L. G., and Wessler, B. D., "Computer network development to achieve resource sharing," *Proc. AFIPS SJCC*. 1970, vol. 36, pp. 543–567.

Roberts, L. G., "Network rationale: a 5-year re-evaluation," *IEEE Compcon*. 1973, pp. 3–5 and pp. 39–44.

Speers, G. S., "Monitoring control by distributed computing," *Datamation*. July 1973, pp. 47–49.

Wecker, S., "A design for a multiple processor operating environment," *IEEE Compcon*. 1973, pp. 143–146.

The purpose of this system is to investigate broadly the use of a computer network; to explore an alternative method of message switching; to provide a wide range of computing facilities to a community of computer and physical scientists, for program and file sharing; and to permit the users to communicate with each other in a variety of ways.

The ARPA network (Fig. 5) is a packet switching, store-and-forward network, whose nodes consist of interface processor computers (IMPs). The computers are linked together by up to four 50-kb/s synchronous communication lines. Messages of up to 8000 bits are transmitted among central (host) computers on a packet basis (up to 1000 bits/packet), with a packet delay of approximately 0.1 second when several IMPs are involved.

A TIP (terminal interface processor) is an IMP without a host computer, but with the capability for up to 64 terminal users to access other hosts. TIPs were added after the initial network was designed, and are used at about one third of the sites.

The hosts are either standard, large-scale computers (e.g., PDP-10s or 360/370s using the same operating systems) or else specialized computers. The latter type includes the ILLIAC-IV at NASA Ames, a large 360, a CDC 7600, a Burroughs B6700, the MIT Multics System, and one or two large 10^{12}-bit files. In addition, hosts monitor and report the network activity, provide information about documentation and programs of interest, and hold user "mailboxes" through which individuals communicate with each other.

The network has been operated in the following ways:

• Remote use of computers either from a termination on a host or via a TIP, or else on a batch or interactive basis. Since computers can be used over long distances, the network functions as a broker for computing facilities. In this way, a particular site can choose not to own and manage a facility, but instead to buy file storage and computation at a variety of other sites. Since ARPA controls the operation of the individual sites, when a site becomes overloaded, the agency can expand the configuration or move some of the users to another site.

• File movement and printing. A user may retain text or programs at one or more sites and then transfer files to particular sites for program execution. Some of the sites have elaborate printing devices for both arbitrary character sets and graphics.

• Personal messages. Users can communicate with each other in various ways. A message for another user may be placed in a "mailbox" located at a host. For example, a message was placed in a mailbox of another user at 11 p.m., requesting latest information on the ARPA network use. By 10 a.m. the next morning, the information was available. This form of communication is faster than a letter or telegram; it is low-cost; and it doesn't require the simultaneous availability of the communicants. Still another advantage of the mailbox is that it permits batch processing of messages by each user.

On the other hand, users can be directly linked via terminals, for both direct two-way and conference real time communication. In the conference mode, a message is broadcast to all attendees.

Finally, terminals may be cross-connected to a single program. In this mode, multiple terminals can have their inputs and outputs connected in parallel. Thus, one user can demonstrate a program by taking several users through a "scenario."

• Machine-to-machine subroutines. In the simplest case, a user has access to a program in a given machine. If the user program requires a program on *another* machine, the first program calls it in a manner akin to subroutines stored in a single computer.

• Access to a large common data base. For example, a 10^{11}-bit meteorological data base is being prepared to be shared among six research sites.

The ARPA Network is important because it provides multiple links per node, so that nodes and links can fail, but the network still functions. Although it could have been built earlier, it requires high-reliability, high-performance minicomputers; time-shared system modes; and 50-kHz synchronous communication links that became available in the past few years. The network design was carried out in an exemplary fashion with extensive analysis, internal documentation, and technical papers. As it became operational, measurement experiments were conducted to verify the analytical tools it developed. It is clearly the archetype for future large-scale networks.

C. Gordon Bell (F) is vice president of engineering for Digital Equipment Corp., Maynard, Mass., on leave as professor of electrical engineering and computer science at Carnegie-Mellon University, Pittsburgh, Pa. He received the B.S. and M.S. degrees in electrical engineering from the Massachusetts Institute of Technology in 1956 and 1957, respectively. Mr. Bell is the holder of several patents in digital systems, the designer of several computers, and he has published books and papers on computer architecture.

Computer report III

Data by the packet

Because computing costs are now so low, an unusual new concept of data communications is feasible

The most dominant force over the past 20 years in both computer and communications architecture has been the continuous and rapid decrease in the cost of computer hardware. Not only has this electronics revolution affected the capability and design of computer systems, but it has also made possible a radically new concept of data communications. Called *packet switching*, this concept is strongly dependent on the cost of computing since it uses computers to correct transmission errors, to provide high reliability through alternate routing, and to allocate communication bandwidth dynamically on a demand basis rather than as a preassigned bandwidth.

The impact of widespread packet-switched communications networks on the computer field should be momentous. With data communications priced on a distance-independent basis, it should become economically feasible for terminals and computers throughout the country to access efficiently—on demand—a wide variety of computer services. The economics of resource sharing will undoubtedly eliminate the need for many medium-scale data processing centers and permit a considerably higher degree of specialization for the larger service centers. Moreover, packet switching will make possible far more economic realization of a comprehensive electronic funds-transfer system, on-line credit authorization, integrated corporate data networks, as well as nationwide access to all forms of data banks and retrieval services.

All these developments hinge on the availability of low-cost computer communications. Although packet switching economizes on transmission costs by maintaining high line utilization, until recently the cost of computer hardware had continued to be too high to permit practical packet-switching networks. The price of computing has been decreasing far more rapidly than the price of communications, however, and one could have anticipated a crossover point where the cost of using computers to allocate bandwidth became cheaper than the cost of the communications themselves. According to my estimate, this crossover occurred during 1969. As a result, packet switching has now become quite an economic and viable technology, permitting the establishment of large, cost-effective networks nationwide. This development in data communications will in turn have substantial effect on the course of computer technology as preliminarily indicated by results from the United States ARPA (Advanced Research Projects Agency) network research activity.

One of the main reasons for the great interest in computer networks today is the considerable economies that can be achieved through resource sharing. With a computer network that is reliable and responsive enough to permit the full resources of a remote computer to be employed locally, it becomes possible for organizations to consider the elimination or reduction of local computing facilities and the utilization of remote service centers instead. This not only can lead to considerable cost savings, as we shall see, but also has many additional advantages, including increased reliability of a multicomputer service center, improved service from a variety of competing suppliers, and increased flexibility for expansion. The main incentive for an organization changing over, however, is the cost saving. To get a feel for the magnitude of this effect, it is useful to examine the initial experience in an environment—the ARPA Computer Network (ARPANET)—where remote use of computing has been fostered by providing adequate communications facilities.[1-3]

Lawrence G. Roberts
Telenet Communications Corporation

Cost-effective remote-access computing

Although initiated in 1969, the ARPANET was not sufficiently developed to permit useful resource sharing activity until mid-1971. As network reliability and effectiveness proved out, ARPA and its contractors found many cases where newly required computer capacity could effectively be obtained through the network rather than by adding local computing resources. By March 1973, several contractors were making substantial use of the network for a majority or all of their computing resources. Several of the computer centers on the network had grown to become substantial suppliers of computer service, providing not only time-sharing service but also remote batch service. At that time, an accounting was made of the total computer usage obtained through the network, and an estimate was made for each user of the cost of purchasing comparable time on outside computers or leasing the necessary in-house computer facilities to do the same job apparently being done through the network. Table I tabulates this information by user organization, identifying both the cost of computing with and without the network. A total of $2 million a year was being spent on computing resources accessed through the network—resources that would have cost $6 million per year if the network had not existed! This savings of $4 million per year more than offset the $3.5 million annual cost of the network.*

At this stage in its development, the network was only using 20 percent of its capacity; thus, at the

* This cost includes amortization of the system's IMP (interface message processor) minicomputer units.

same communication cost, the ARPANET could permit five times as much remote computing to be performed. At full traffic load, the network cost would be 35 percent of the actual computing cost, or 11 percent of the original computing cost. Clearly, for a moderate investment in communications, a computing cost reduction factor of three can be achieved through resource sharing.

Examining the ARPA data, it is useful to identify a few of the different categories of network usage and explore the sources of the cost savings. The largest individual user organization was the University of Illinois, which before the network came into full operation leased its own Burroughs 6700 computer to develop programs for the ILLIAC IV computer. After the University of California at San Diego put its large B6700 complex on the network, Illinois was able to terminate its computer lease and obtain the computing support far more economically from the larger installation through the network. Moreover, Illinois required access to the ILLIAC IV computer that it obtained from the NASA Ames Research Center through the network, remote batch service on the IBM 360/91 at U.C.L.A., and time-sharing services from U.S.C. and M.I.T. To obtain this collection of services locally, Illinois would have had to continue leasing the Burroughs computer, purchase time on a local IBM 360, and either purchase a time-sharing machine or obtain service from a commercial vendor. Therefore, not only would it cost the University three times as much to obtain equivalent computing power locally, but it would not have had a convenient way of accessing these machines and transferring data and programs between them.

This case is an excellent example of a user requiring many different types of service, each being most efficiently provided by a large specialized service center. For example, program development and debugging is extremely cost-effective on a time-sharing computer, but the execution of large numerical applications programs can often be done a factor of ten less expensively on a large batch processing machine.

Another type of saving is exemplified by Massachusetts Computer Associates, a small computer software firm. This group would use the equivalent of one third of the capacity of a PDP-10 time-sharing system, if they operated locally on a dedicated basis. The required facilities, however, were provided through the network from the University of Southern California, taking advantage of the fact that peak usage hours on the east and west coasts of the U.S. do not coincide due to the time zone difference. A large number of such users of various sizes can be accommodated by a single service center with considerably greater economy of operation than with a number of independent centers. The consolidation of maintenance, software, and operational personnel in one location considerably reduces costs, as most of the time-sharing companies have found, and nationwide operation increases the machine utilization considerably due to the time zone spread.

Although there are many other combinations of resource sharing represented within the ARPA network community, these two examples illustrate the main sources of cost savings resulting from hardware sharing. Additionally, large but less easily quantifiable

I. Computer resource usage within ARPANET*

User Organization	Activity	Remote Usage, in thousands	Projected Cost for Local Replacement, in thousands
University of Illinois	Parallel processing research	$ 360	$1100
NASA Ames Research Center	Air foil design and ILLIAC	328	570
Rand Corporation	Numerical climate modeling	210	650
Mass. Computer Associates	ILLIAC IV compiler development	151	470
Lawrence Livermore Laboratory	TENSOR code on ILLIAC	94	370
Stanford University	Artificial intelligence research	91	180
Rome Air Development Center	Text manipulation and resource evaluation	81	450
ARPA	On-line management	77	370
Seismic Array Analysis Center	Seismic data processing	76	300
Mitre Corporation	Distributed file network research	60	240
National Bureau of Standards	Network research	58	200
Bolt Beranek & Newman Inc.	TENEX system support	55	80
Xerox Palo Alto Research Center	Computer science research	47	100
University of Southern California Image Processing Lab	Picture processing research	35	70
University of California, Los Angeles	Network measurement	28	90
Systems Control, Inc.	Signal processing research	23	70
University of California, Santa Barbara	Network research	22	70
U.S. Air Force Range Measurements Lab	ARPANET management	17	60
Institute for the Future	Teleconferencing research	13	40
Miscellaneous	Computer research	192	580
	Total	$2018	$6060

*Annual remote computer usage cost is based on March 1973 data.

cost savings are anticipated to be forthcoming in the future from the sharing of data base and software resources.

Computer–communication interdependence

The large savings possible from resource sharing are dependent upon having available an economic means of switched communications to access these resources. This communication facility must be reliable, sufficiently responsive to meet all of the interactive demands of time-sharing users, have sufficient data transmission speed for high-speed printer and display output, be distributed to scattered users throughout the country, and most importantly be sufficiently economic so as not to destroy the cost savings achieved through resource sharing. In the early 1960s, this kind of communications service was not available. It is only with the extremely rapid development of computers themselves that it has been possible to reduce drastically the cost of providing such a communications service to computer data users.

Since computer input/output typically is extremely bursty in nature, requiring peak data transfer rates ten to one hundred times as great as the average data rate, it is necessary to share statistically among many users a communication channel of fixed data transmission capacity in order to achieve reasonable economy. Without doing this, the cost of data communications would be 10 to 100 times as expensive as the raw communication bandwidth, thereby making resource sharing cost-ineffective over even moderate distances.

Packet-switching technology as is used in the ARPANET has been developed to permit the statistical sharing of communication lines by many diverse users. By dividing the data traffic into small addressed packets (1000 bits or less in length), extremely efficient sharing of communication resources can be achieved even at the burst rates required by interactive computer traffic. Since resource-sharing computer usage has only become economically viable with the development of packet switching, it is instructive to look at the cost trends that made this technology possible. To do this, it is necessary to have a model of the average resources utilized in moving data through a packet-switching communications network.

The basic design of a packet-switching network, as exemplified by the ARPANET, consists of a collection of geographically dispersed minicomputers called *interface message processors* (IMPs) interconnected by many 50-kilobit/second (kb/s) leased lines. An IMP accepts traffic from a computer attached to it called a *host*, formats it into packets, and routes it toward its destination over one of the 50-kb lines tied to that IMP. Each IMP in the network receiving a packet examines the header and, making a new routing decision, passes it on towards its destination, possibly through several intermediate IMPs. Thus, a packet proceeds from IMP to IMP in making its way to its destination. The destination IMP collects the packets, reformats them into messages in the proper sequence, and submits them to the destination host computer. Throughout the process, each IMP checks the correctness of each packet by means of both hardware- and software-based error-control techniques. If the packet is received incorrectly due to a transmission error on the line, the IMP does not acknowledge receipt and the preceding IMP must retransmit the packet, perhaps over a different path. Because the network uses high-speed transmission lines and short packets, and all data is stored in high-speed primary memory in the IMPs (as opposed to disk drives and other secondary storage devices) average end-to-end transit delay for a packet is 0.1 second.

This general design of a packet-switching network has proven so successful in the ARPANET that a new component of the communications industry has been formed—"value-added" packet carriers, offering the public packet communications service on a nationwide regulated basis. Two of the four value-added carriers that have applied to the FCC—Telenet Communications Corporation and Packet Communications, Inc.—intend to offer a service patterned directly on the ARPANET. The FCC has responded extremely positively to this new development in the communications field and, indicating its intention to adopt a policy of "liberal entry" in this field, has already approved the first of the value-added carrier applicants.

In a nationwide packet-switching network, there is practically no dependence of cost on distance, and for this reason both Telenet and PCI intend to charge only for the *number* of packets moved, independent of the distance. If we assume that the traffic itself is evenly distributed geographically, the average distance traversed by a packet would be about half the cross-country mileage, or 1200 miles in a network spanning the U.S. The number of *hops* (intermediate leased lines traversed) required on the average to deliver a packet is a logarithmic function of the number of *nodes* in the network and therefore increases very slowly once the network exceeds about 40 nodes. For triply connected networks of 40 to 100 nodes, the average number of hops required is three or four. Since

II. Data communications costs since 1960

Date Introduced	Data Rate	Cost per Million Bits
1960	2.4 kb	$1.00
1963	40.8 kb	$0.42
1964	50.0 kb (Telpak A)	$0.34
1967	50.0 kb (Series 8000)	$0.33
1974–76	56.0 kb (DDS)	$0.11 (est.)

III. CPU time and cost estimates for the ARPA network

Date	Machine	Cost	Total Processing Time per Kilopacket	Cost per Kilopacket
1970	Honeywell 516 IMP	$100k	26.2 seconds	$0.168
1971	Honeywell 316 IMP	$ 45k	35.4 seconds	$0.102
1974	Lockheed SUE IMP	$ 10k*	41.2 seconds	$0.026

*Estimated per processor based on multiprocessor configuration.

commercial packet-switching networks, such as that proposed by Telenet Communications Corporation, will employ "central office" facilities serving many users in a city, it is not likely that the number of central office nodes will exceed 100 for many years to come. Therefore, taking four hops as a reasonable estimate, the average packet would traverse five IMPs and four lines on its course between two locations 1200 miles apart. The average length of these leased lines would be 300 miles, a number that clearly fits the model and also matches the ARPANET experience.

Therefore, two cost factors are involved in moving the average packet: (1) the communications cost for four 300-mile leased lines, and (2) the computation required in five IMPs.

Communications cost trends

Over the past decade, there has been very little change in the actual per-mile price of leased communications channels. However, in the more recent past there have been substantial changes in modem technology, permitting higher-speed data transmission over these lines. and in the near future the American Telephone and Telegraph Co. expects to introduce Digital Data Service (DDS). a totally digital technology at a substantially lower price.

Considering the period 1960–1980, one detects a downward trend in data communications cost. To quantify this, let us examine the cost per million bits of data moved for reasonably high-bandwidth leased lines 300 miles in length (including termination and modem costs). The actual costs tabulated in Table II have been multiplied by four so that they are truly representative of the incremental cost for moving one million bits (or a thousand packets) over a packet-switched network for a total distance of 1200 miles. The calculations are based on the lines being fully utilized in full-duplex mode for eight hours each working day (173 h/month—the same level of utilization assumed in calculating subsequent computer costs). It should be noted that a single leased line 1200 miles long would in general be 20–50 percent less expensive, but such a dedicated line would not permit the multipoint access and usage-based charging provided by a packet network.

Based on this data, if one takes the cost of service for each year between 1960 and 1980 and computes the least-mean-square exponential fit, one finds that the cost of service has been decreasing by 11 percent per year over that period, resulting in a factor-of-10 decrease in communications cost each 22 years. A similar analysis was made of the costs of dial telephone service over a distance of 1200 miles, using the highest data rate modem feasible over dial telephone lines. For the dial case, a comparable rate of decrease in service costs was found, largely due to improvements in modem performance. but with an average cost per megabit eight times greater than the wideband leased line case. Thus, at least over the last decade or so, the cost of data communications service has been decreasing fairly consistently and it is safe to assume that continued advances in digital transmission technology will continue this trend into the future. There is, however, no indication that a remarkable breakthrough is about to occur. drastically decreasing the cost rate of terrestrial transmission service from common carriers.

Computer cost trends

The cost per unit capability of electronics and computer equipment has often been studied, and in all cases the trend was found to be remarkably consistent at about a factor-of-ten decrease in cost every five years. In a 1969 study made by the author[4] analyzing the cost performance of complete computer systems vs. their date of first delivery, it was found that the cost performance was increasing exponentially at a rate of 1.56 per year. This study was based on the least-mean-square best fit for all machines introduced between 1955 and 1969 having at least 32-bit word length. In a later study by Newport and Ryzlak[5] examining the cost performance of communication processors, it was found that the cost performance was increasing at a rate of 1.61 per year. In this study, minicomputers introduced between 1960 and 1972 were also examined.

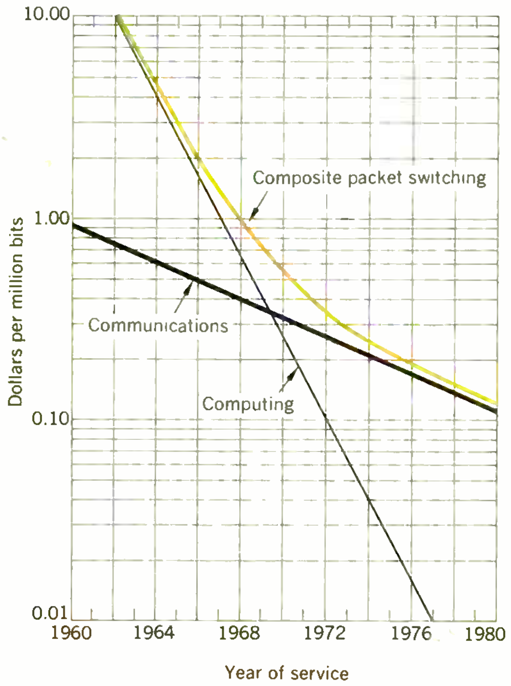
Given the remarkably consistent rate of increase found by these and other studies it is quite clear that the rate of growth of computer cost performance is well established. In order to fix such a curve absolutely, however, it is necessary to examine the actual CPU time consumed by an IMP to process 1000 packets of information. Very accurate studies have been made by Bolt Beranek and Newman Inc. (BBN)—ARPA's contractor for building and operating the ARPANET—of the CPU time required for the IMPs used within the ARPANET. The time consumed depends somewhat on the length of the packet, but for consistency (equating 1000 packets to one megabit of data) full packets of a thousand bits will be considered. In Table III, the three computers developed for ARPA by BBN to provide IMP service are considered. The date of introduction, the cost, and the total CPU time consumed to move a thousand full packets through five IMPs in the network are listed. The additional processing required to accept and deliver these packets to the host computer is also included. The cost per kilopacket of traffic was found by depreciating this equipment over 50 months and assuming operations and maintenance expense to equal equipment cost; operation was set at eight hours each working day for consistency with our earlier communications cost calculations.

These costs are quite consistent with the previously described growth rate of cost performance with time and form the basis of an actual trend curve for the computational cost of packet transmission. To be consistent with the treatment of communications trends, the dates that should be utilized in considering the IMP processing costs should be delayed a year from the date of introduction to correspond more accurately with their average usage date.

Composite costs for packet switching

The graph shown in Fig. 1 shows the trend curve for the data communications and computational cost associated with moving a million bits or a thousand packets of information over an average distance of 1200 miles within a nationwide network. As can be seen, even with an 11-percent decrease per year, the cost of communications remains relatively flat,

[1] A comparison of cost performance trends for computing, data communications, and packet-switching systems over a nationwide network. The impact of satellite communications on packet switching is particularly emphasized in the graph on the right, which also demonstrates the cost effectiveness of reducing the regional size of the national communications network via satellites.



whereas the cost of computing has been decreasing with extreme rapidness; these curves intersect in 1969. Before that year, the cost of the computer power required to provide dynamic allocation of communication resources would have made packet switching extremely expensive to implement and, in fact, the first packet-switched network—the ARPANET—was not initiated until 1969. The colored curve shows the overall cost of communications plus computation associated with packet switching. Since only the incremental cost of moving a kilopacket through the net have been considered here, these results should not be interpreted as indicating the entire cost of operating a communications service, but only the incremental costs associated with moving large quantities of data. Also, since the communications trend in particular has been smoothed considerably, whereas in fact the present cost of 50-kb/s communication lines is still 33 cents/Mb, the resultant cost for any particular year should not be considered as an accurate indicator of precisely what the price ought to be. For example, in Telenet Communications Corporation's publicly filed prospective tariff, incremental packet traffic is priced at 48 cents/kilopacket. This is very close to the total cost of communication (33 cents) and IMPs (10 cents) today but corresponds to 1970 on the graph.

A more general interpretation can be made of Fig. 1. Except for the numerical absolute scale numbers, it is entirely representative of the costs associated with any system that utilizes both communications and computing components in a fixed ratio! If the communications segment is only a small fraction of the overall system, the crossover will be much later in time, but the identical shape will be preserved. Thus, for any data processing system providing remote computing capability, the curve indicates that as time passes the costs will become more and more dominated by communications. This means that it is all the more important for new communications techniques such as packet switching to be introduced in order to reduce this component of the total system cost.

IV. Cost estimates for INTELSAT communication satellites

INTELSAT	Usage Year	Number of Circuits	Lifetime, years	Total Cost	Cost per Circuit per Year
I	1965–67	240	1.5	$ 8.2M	$22 800
II	1967–68	240	3	$ 8.1M	$11 300
III	1968–71	1 200	5	$10.5M	$ 1 800
IV	1971–78	6 000	7	$26.0M	$ 600
V*	1978–85	100 000	10	$28.5M	$ 30

*Estimated.

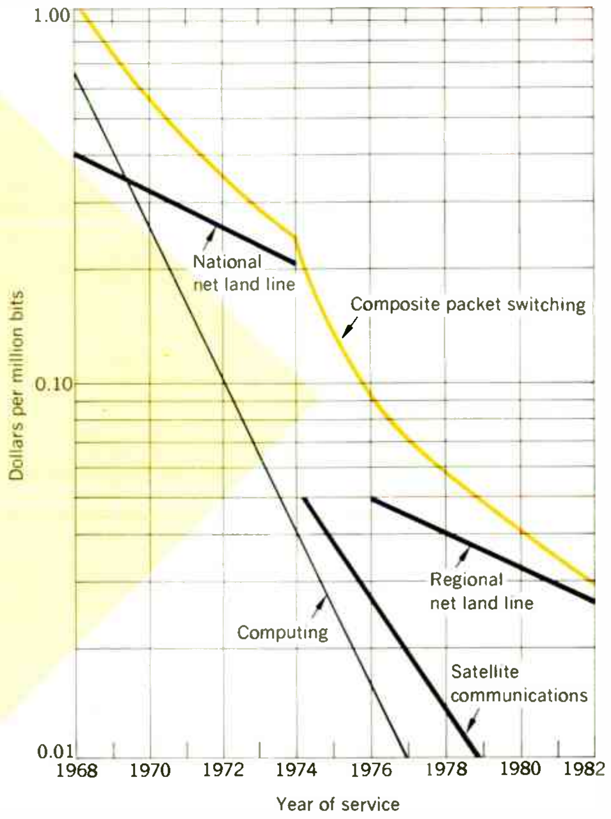
The future—satellite communications

Although terrestrial communications cost appears to limit the future price of computer-communication service, including packet-switching networks, the situation is rapidly changing with the introduction of domestic satellites. The cost trends for satellite technology are not yet sufficiently well documented for great confidence; however, a preliminary estimate can be made from the international satellite progress over the last decade. Table IV presents the space-segment cost per circuit for the INTELSAT series of satellites.

Applying the least-mean-square exponential fit to this data, the rate of technological improvement in the cost performance for satellites is found to be 40.7 percent per year, or a factor of ten every 6.7 years. This can only be treated as a crude estimate of the cost trend for satellite communication, but since it is quite in keeping with the general cost trend for electronics, it is a quite credible growth rate.

Preliminary estimates of the actual cost of wide bandwidth digital communications service on domestic satellites are in the vicinity of 5 cents/Mb for service in early 1974. As depicted in the right-hand chart of Fig. 1, if the cost trend for domestic satellites follows the initial international trend, the projection of satellite communication cost performance is very close to that of computers.

Satellites can be used extremely efficiently within a packet network when used on a broadcast basis between a large number of ground stations. A detailed study of the techniques for broadcast satellite use and



their superior performance over the use of satellites as point-to-point channels has already been described in a previous paper.[6] The cost of long-distance communications between ground stations can be reduced significantly as indicated by the satellite trend in Fig. 1. However, terrestrial leased lines are still required to interconnect the central offices of the packet network within each ground station region. By 1976–78, there should be sufficient domestic satellite stations in operation so that 12 regions can be established within the U.S., thereby reducing the size of the regional packet network to a 200-mile radius. In this case, instead of using four 300-mile leased lines, each packet would traverse four 50-mile leased lines (two in each ground station region) plus the satellite hop.

Thus, the overall cost of packet switching would be the sum of three components: computing, the satellite link, and the 50-mile terrestrial links. The trends of these three areas are plotted in Fig. 1, and the overall cost of packet switching shown to break in 1974 with the introduction of satellite service. Satellites will play an important role in reducing the future cost of packet-switching service and, for this reason, broadcast satellite service has been included as an integral part of the proposed offering by Telenet for its nationwide packet-switching system.

In the more distant future, additional technological improvements are expected to continue to occur in the packet-switching field. When they are introduced, high-frequency (15–30-GHz) satellites will most likely permit direct satellite interconnection of the central offices in a packet-switching network. This will produce an additional break in the cost trend and leave only the local distribution problem. The utilization of cable television systems and packet radio broadcast techniques[7] should permit substantial improvements in the future local distribution scene. Hence, it is entirely possible that the cost trend for packet switching will continue to decrease at almost the same rate as the cost of computation if these new packet communication techniques are vigorously pursued.

Where we stand

Experience with ARPANET has demonstrated that computing service can be obtained remotely through a computer network at one third the cost of a local dedicated system. To obtain this factor-of-three cost reduction, however, one must have available a highly responsive and reliable communications system capable of handling the peak data rate of the burst traffic normally associated with computer usage. In order to provide such an enhanced communications service, it is necessary to embed significant computational and logical capabilities within the network itself.

Until recently, the cost of computer hardware, as compared to that for the raw communication facilities, was too high to permit widespread use of computer intelligence within a data communications system. However, computer hardware costs have been falling so rapidly relative to communication facilities costs that "intelligent" and enhanced data networks have suddenly become a viable technology.

REFERENCES

1. Roberts, L. G., and Wessler, B., "Computer network development to achieve resource sharing," *Proc. AFIPS Spring Joint Computer Conf.*, pp. 543–549, 1970.

2. Heart, F., *et al.*, "The interface message processor for the ARPA computer network," *Conf. AFIPS Spring Joint Computer Conf.*, pp. 551–567, 1970.

3. Kahn, R. E., "Resource-sharing computer communications networks," *Proc. IEEE*, pp. 1397–1407, Nov. 1972.

4. Roberts, L. G., "Data processing technology forecast," internal U.S. Government paper, Apr. 23, 1969.

5. Newport, C. B., and Ryzlak, J., "Communication processors," *Proc. IEEE*, vol. 60, pp. 1321–1332, Nov. 1972.

6. Roberts, L. G., "Dynamic allocation of satellite capacity through packet reservation," *Proc. National Computer Conf.*, pp. 711–716, 1973.

7. Roberts, L. G., "Network rationale: a 5-year reevaluation," *Proc. COMPCON 73*, Feb. 1973.

Lawrence G. Roberts (M) obtained the Ph.D. degree in electrical engineering from M.I.T. in 1963. He is presently president of Telenet Communications Corporation, Washington, D.C., a firm that plans to establish and operate a nationwide packet-switching service as a common carrier under FCC regulation. Formerly director for information processing techniques with the Advanced Research Projects Agency of DOD, Dr. Roberts was principally concerned with new information processing techniques in the areas of computer systems, graphics, artificial intelligence, signal processing, weather modeling, and man–machine interaction. In addition, he was responsible for developing the ARPA computer network. Prior to his government experience, he led the computer software group at M.I.T.'s Lincoln Laboratory and was instrumental in developing a three-dimensional input device (the Lincoln Wand), the APEX time-sharing system, the CORAL list structure system, and several graphics systems. Dr. Roberts has written many professional articles and has lectured throughout the world. He was selected the 1965–66 national lecturer for ACM, of which he is a member; he is also a member of Sigma Xi.

Computer report IV

A checkup: minicomputer software

Great expectations for new applications hinge on the usefulness of software packages

Back in 1971, Charles Jackson—a software-oriented EE—visited a minicomputer manufacturer's demonstration facility to try its machine. A salesman had assured him that the manufacturer's Fortran was working, so Mr. Jackson tried to run a well-debugged benchmark program. It compiled, began to run, although giving strange results, and finally halted abnormally. No one could explain this mysterious action.

This is the kind of experience—fortunately an increasingly infrequent one—that has led to many irritated statements about the poor quality of minicomputer software. Perhaps the most telling general complaint is that programs written for one minicomputer can rarely be used on another. Like preindustrial wooden machines, minicomputer software is said to be largely composed of noninterchangeable components, handcrafted at great expense, often unreliable, and difficult to maintain.

A related complaint is that applications software is so poorly developed that minicomputers can only be applied by users who are exceptionally competent.

This article will review examples of minicomputer software in a number of important application areas, hoping to help the reader make his own judgments about quality and diversity.

A brief look at the history of minicomputer software development, and at some current trends in the field may help to put these application examples into focus.

Putting logic modules together

Minicomputers were born when circuit manufacturers—particularly Digital Equipment Corporation—realized they could put together computers, using the logic circuit modules they ordinarily sold to experimenters and instrumentation designers. Applications were few, because only a few capable and sophisticated people—primarily in industrial and educational organizations—were able to use these machines.

The manufacturers soon realized that, to widen their markets, they would have to supply at least a small amount of software. Basic tools such as an assembler (see Fig. 1) and editing programs to help with debugging were about the minimum they could afford the user, and that is what was offered. The reason the early minicomputers could be sold at comparatively low prices was that they were hardware components, not complete systems with a full complement of software support.

As applications gradually broadened, especially in process control, in communications systems, and in scientific and engineering problem-solving, higher-level languages like Fortran and Basic were implemented for minicomputers, but it was not until the past few years that business applications programs and languages have become available to minicomputer users.

Howard Falk Associate Editor

How big can a minicomputer grow?

Between minicomputers and large machines, the boundaries seem uncertain. Many computer people feel that it is word length that makes the difference. By this measure, a machine that can handle computer words longer than about 16 bits is more or less large, while machines that handle 16 bits or less are likely to be minicomputers. The main point is that longer word length brings with it enhanced ability to process large amounts of data.

Certainly, in the past, minicomputers have been used mainly in applications that require only short word lengths—for example, in logic and control applications rather than for the massive number-crunching required by many scientific computations. But it is by no means clear that this will continue to be the case in the future.

Availability of memory resources—particularly of large-scale memory—has been another means to draw the line between minicomputers and larger machines, but the distinction also seems to be losing its significance.

Recently there has been tremendous growth in the computing power that a hardware dollar can buy—mainly due to the maturing of large-scale integration technology. As a result, the gap has been closing between minicomputer hardware capability and big computer hardware capability, and the distinction between these two classes of machines rests more heavily than ever on software differences.

What direction, then, has minicomputer software been taking?

One major trend has been towards more sophisticated systems software—the kind of software that manages computer system resources so that user programs can operate efficiently. For example, tremendous progress is being made in memory management, with current minicomputer features often superior to those offered by large computers only a few years ago. The days are gone when a user had to load his program in absolutely fixed, defined minicomputer memory areas. Today, minicomputer systems provide the user with automatic management of drums and discs, as well as main memory, so he no longer has to keep track of physical locations.

A key achievement of minicomputer systems software has been the use of input-output (I/O) handlers—standard software packages that connect standard peripheral equipments into the system.

For example, when the user's program says "get" and "put" data, the I/O handler for the punched-card reader signals the reader to move a card, then it converts the card information into the format that the computer needs, and finally it transfers the information into the computer's memory.

A second trend is towards increased compatibility. Minicomputer manufacturers are becoming more conscious of the need to make their current software compatible with the software they introduced last year and plan to make available next year.

Hardware compatibility is already available for several "lines" of minicomputers—that is, there are families of machines that can use the same peripherals, have the same standard interfaces, memory controllers, etc. And, compatible software for these machines—with the flexibility of use that is available in conventional, large computer systems—is becoming available.

Software as well as hardware compatibility is very important to the system designer who uses minicomputers, because it allows him to start with a minimum system, test a process he wants to use, and then apply directly what he has learned in a full-scale system, without having to reprogram what he has already done.

A third trend is towards greater accuracy for minicomputer software. Computational accuracy depends both on hardware and on software..For example, accurate floating point arithmetic can be implemented in software on an inexpensive minicomputer, but from the minicomputer manufacturer's viewpoint, writing these routines may require many man-months of programming effort, and the routines may be slow-running. Using a more expensive minicomputer, capable of handling larger numbers during each memory cycle, the same accuracy might be obtained with less programming effort and faster run-times. With floating-point as a built-in hardware feature of the machine, the run time would be still faster, while the programming effort would be minimal. Of course, such hardware adds to the cost of the computer system.

The word-sizes of minicomputers have been increasing—many machines now handle 16-bit words, and 32-bit machines are becoming available. Built-in hardware capabilities like floating-point arithmetic are, likewise, becoming more widely available, and these trends promise continued improvement in economically available computational accuracy.

As minicomputer systems software becomes more sophisticated, and these systems begin to look more and more like large computers, price advantages are moving minicomputers into traditional areas of large computer applications.

Minicomputer business applications software

Virtually every U.S. company with 5000 or more employees now has computers installed, but these make up a very small part of all U.S. companies. Minicomputers are expected to open up huge new business applications markets, reaching out to the remaining 99 percent. In 1973, an estimated 8000 minicomputers were shipped to smaller businesses; by 1977, more than half of all the operating general-purpose business computers are expected to be minicomputers.

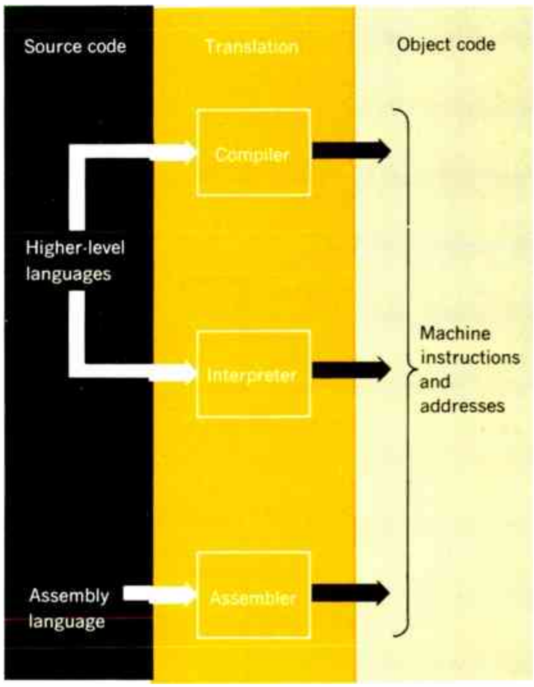
This market is in its early stages of growth because

Some software terminology

Machine instructions and addresses (object code) are in the form of binary zeros and ones, fine for the computer, but difficult for human beings to recognize and manipulate.

Languages (source code) used by programmers may be *higher-level*—like Fortran or Basic—with statements that express the users intention fairly directly. These can be *translated* by a software *compiler* which produces ready-to-run programs containing machine instructions and addresses. Alternatively, an *interpreter*—either hardware or software—may be used to directly translate the higher level language into machine instructions and addresses at the time the source code is being read by the computer.

Assembly language statements are more like machine code than like human instruction, but they are phrased in letters and numerals. An *assembler* program translates these symbols into binary-coded machine instructions and addresses.



[1] **Software terminology relationships are indicated in the diagram.**

business applications for minicomputers are a relatively recent development. Early attempts at such applications were frustrated by programming difficulties. Business programmers were accustomed to such conveniences as the use of a single COBOL instruction to add two decimal dollar figures. The prospect of writing 200 minicomputer instructions to achieve the same result did not seem very attractive.

The first higher-level minicomputer business language appears to have been a COBOL-like language called SAIBOL, developed in 1967 by Wilbur Highleyman for use on PDP-8 machines. In 1973, a COBOL compiler was announced for the Raytheon Data Systems RDS-500 machine.

Minicomputer business software is now finding its way into many specialized applications. For example, one software package now being sold to automobile dealers uses a parameter- or table-driven approach. The software design caters to the fact that each individual automobile dealer wants different kinds of computer-generated reports and charts of accounts. The computer installer uses fill-in tables and control parameters to set up each dealer's files as desired, using a relatively simple keyboard-input procedure. In this way, the same software package can be used by many different dealers without need for additional programming.

Software progress in the more traditional minicomputer applications areas of technology and industry is illustrated by the following selected examples of current software packages.

A telecommunications access package

The General Automation Telecommunications Access Method (GATAM-16) is a modular set of programs that provide for synchronous transmission of binary-coded data. GATAM-16 can transmit either Extended Binary Coded Decimal Interchange (EBCDIC) data, or United States of America Standard Code for Information Interchange (USASCII) data, or a transparent mode using Binary Synchronous Communications (BSC) techniques.

GATAM-16 provides for: point-to-point communication with another GATAM-16 station, an IBM 2780 Data Transmission Terminal, or an IBM System/360 Model 20; point-to-point or multipoint communication with an IBM System/360 or 370 computer; and off-line card reader-to-printer and card reader-to-punch processing.

Included in the facilities provided by GATAM-16 are such automatic functions as:

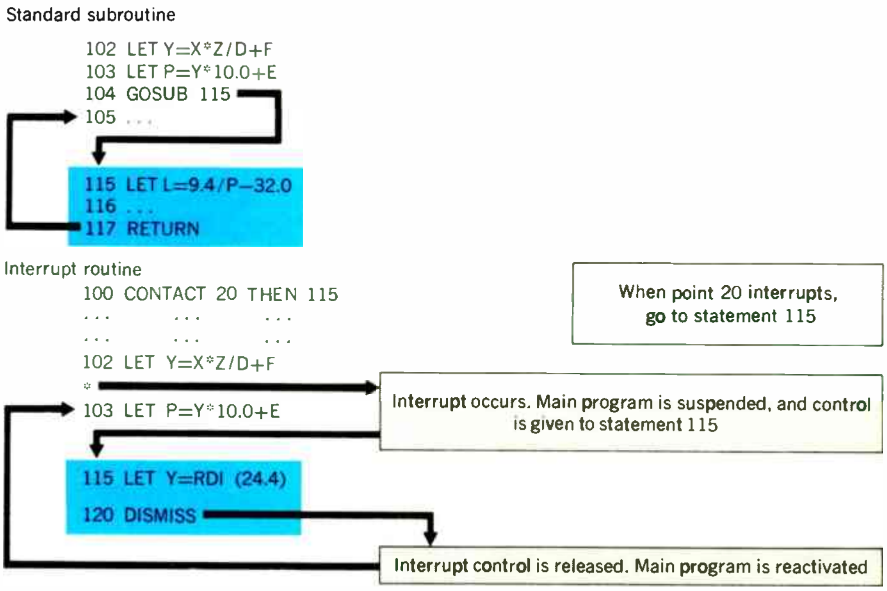
- Bidding for a line to resolve contention
- Polling and selection in a multipoint environment
- Error checking
- Message block format
- Time-out control
- Interpretation of data link control characters
- Synchronization
- Temporary delays in message transmission
- Sequential checking of message blocks, and
- End of message transmission

The GATAM-16 package runs on a General Automation SPC-16C Computer under either the RTX-16 (with bulk storage) or the RTOS-16 operating systems. Documentation includes a reference manual and source-language listings. The package is included, at no extra cost, with appropriate General Automation equipment configurations for a $2000 license fee.

An automated test system package

The Texas Instruments ATS-960 software is designed to test analog, digital, and hybrid electronic assemblies at high speeds. It is capable of performing static functional, and dynamic testing as well as analog and digital fault isolation.

Up to four test stations may be operated simultaneously, with each station performing a different type of test on components or circuit boards. For example, one station can perform fault isolation on digital circuits while a second performs go/no-go tests on analog assemblies. At the same time, a third station can be used for program debugging or editing while a fourth is performing incoming inspection of integrated



[2] The Industrial Basic contact statement sets up a control routine that goes into action as soon as an external event—the closing of switch points—occurs. Standard Basic offers no facility for responding to real-time external actions: subroutines can proceed only under the control of the main program.

Standard subroutine

```
102 LET Y=X*Z/D+F
103 LET P=Y*10.0+E
104 GOSUB 115
105 . . .
115 LET L=9.4/P−32.0
116 . . .
117 RETURN
```

Interrupt routine

```
100 CONTACT 20 THEN 115
. . .       . . .       . . .
. . .       . . .       . . .
102 LET Y=X*Z/D+F
*
103 LET P=Y*10.0+E
115 LET Y=RDI (24.4)
120 DISMISS
```

When point 20 interrupts, go to statement 115

Interrupt occurs. Main program is suspended, and control is given to statement 115

Interrupt control is released. Main program is reactivated

A look at minicomputer Fortrans

These descriptions and comments are based on personal experience with Fortran programming over the past four or five years.

Data General Fortran

The larger versions of this Fortran are as powerful as the Fortran IV used on IBM 360 machines. Even the simpler versions have many extra features such as double precision complex variables, array index lower bounds other than one, unformatted I/O. as many as 128 array dimensions. and text variables. With larger versions—using 12K words of core memory—the user also gets such features as: data initialization using the data statements. external statements. mixed mode arithmetic. and equivalence statements.

A two-pass compilation procedure is used to produce a binary tape from Fortran source statements. The tape contents are relocatable—written so they can be located and executed in various different memory locations. Maximum program size is 300 lines of coding with the 8K configuration and in excess of 400 lines of coding with the 12K configuration (comments are not counted). The maximum program size also depends on array storage requirements and on the library subroutines used by a particular program.

This software runs on a Data General Nova 1200 computer. A Fortran IV user's manual supplied by Data General (approximately 60 pages) describes language syntax, operating and compilation procedures. and diagnostic messages. The software and documentation are supplied with the computer equipment at no added cost.

Digital Equipment Corp. 4K Fortran

In sharp contrast to the powerful Fortran just discussed, this version is very limited. It handles only single-dimension arrays. and provides only single precision calculation. With 6 or 7 significant digits. Subroutine calls are not allowed, and output is restricted to standard formats.

A one-pass compiler is used. Maximum program size is about 100 lines of coding—excluding comments—with 200 floating-point storage locations allowed.

This Fortran runs on a PDP-8, -8I, or -8E computer with 4K words of storage. A programmer's reference manual (approximately 30 pages) supplied by DEC. describes language syntax, as well as operating and computation procedures, and diagnostic messages.

The software is included in initial equipment charges. Binary-coded tape copies are available at $15 each.

DEC 8K Fortran

This language is basically similar to the standard ASA Fortran II. However, it provides only single-precision arithmetic, and permits no mixed-mode arithmetic. No more than two array-dimensions are allowed.

A two-pass compilation procedure is used to produce a relocatable binary tape from Fortran source statements. Maximum program size is between 200 and 300 lines of coding (excluding comments) depending on the size of storage arrays and the number of library routines used. Software used for compilation, loading. and running Fortran programs include an 8K Fortran compiler. as well as a SABR assembler. linking loader, and a two-part Fortran library.

This software runs on a PDP-8, -8I, or -8E computer with 8K words of storage. Disk facilities may be used to speed compilation, loading, and data handling. Programmer's reference manuals for DEC 8K Fortran and the SABR assembler (each approximately 30 pages in length) describe language syntax, operating and compilation procedures. and diagnostic messages.

Cost of the software is included in initial equipment charges. Binary-coded tape copies are available at $30 per set.

Hewlett-Packard 4-K Fortran

This language is similar to the standard ASA Fortran IV, but is restricted to handling two-dimension arrays and single-precision arithmetic. Also, it allows no logical "if" statements, and no alphanumeric input, and provides no variable-comparison capability.

A four-pass compilation procedure is used to produce a relocatable binary tape from the Fortran source statements. Maximum program size is about 100 lines of Fortran code, excluding comments. Exact allowable size depends on the requirements for array storage and the library subroutines used by any particular program.

This Fortran can run on the HP 2114A or B, the HP 2115A or the 2116A, using 4K words of memory. A programmer's reference manual for H-P 4K Fortran and for H-P 8K Fortran as well (approximately 50 pages in length). describes language syntax, compilation and operating procedures, and diagnostic messages. Software and documentation are supplied with the computer equipment at no added cost.

H-P 8K Fortran

Although similar to ASA standard Fortran IV, this language provides no double precison arithmetic. arrays are limited to two dimensions, and no logical "if" statements are allowed.

A two-pass compilation procedure for Fortran source statements, produces a relocatable binary tape. Maximum program size is 200 to 300 lines of coding depending on array storage requirements and the library subroutines that are used by a particular program.

This Fortran can run on the same machines as the H-P 4K Fortran, but requires 8K words of memory capacity. The programmer's reference manual is the same one supplied for 4K Fortran. Software and documentation are included in the original equipment cost.

George W. Yaeger. Jr.
Computer Applications Team
Fort Monmouth. N.J.

General Automation Fortran

Two versions of this SPC-16 Fortran IV are available for systems with 12k or 16k memories. Both provide several useful Fortran extensions. Extensive optimization of object code is performed. The compiler is one-pass, occupies 8k of core, and uses no work files. Global flow analysis and related optimization, characteristic of large-machine Fortrans is thus impossible, but run-time performance is still surprisingly good.

Noel Stanton
Ohio State University

circuits. At each station, the desired operational mode—normal test sequence under program control, recycle one or more tests, manual advance one test at a time, etc.—can be selected by the operator.

All test programs for the ATS-960 system are written in ATLAS, an industry-standard language. New test programs can be written and tested on the ATS-960 system while it is processing production circuits. The programmer can change, add, or delete instructions or test sequences by means of the test station keyboard.

Any test instrument having remote control capability can be incorporated into an ATS 960 system. Among those already in use are: Sorenson power supplies, E-H Research Labs 1501 pulse generators, Data Pulse 153/154 pulse generators, Wavetek 154 waveform generators, Hewlett-Packard 5326A/B timer-counters—DVMs, Dana 8100 counters, and Tektronix R568/230 pulse measuring units.

The ATS-960 package runs on a Texas Instruments 960A computer. Documentation is available only to customers, and the software package is not for sale separate from the hardware (complete system prices range from $35 000 to $250 000).

An industrial, higher level language

Industrial Basic, used at Digital Equipment Corp., is an extended version of the widely used standard Basic language. Special commands and features have been added to allow a direct control of input and output devices, scheduling parameters, and interrupt conditions.

Standard Basic may be thought of as a synchronous language—that is, each command produces a result at the time it is executed, and the only commands that can be given are those that have immediate results.

Industrial Basic extends the standard, synchronous functions of Basic to read and drive industrial-type devices. In addition—and this is its key capability—Industrial Basic can service *asynchronous* external events such as time delays, pushbutton commands, and alarm signal responses.

Industrial Basic services these asynchronous events, using process interrupt routines that allow the computer operating system to perform the needed functions automatically as they are required (Fig. 2).

Specific new functions of the language include:

- Analog input—allows sampling on A/D converter
- Analog output—allows voltage current output from a D/A converter
- Read digital input—allows the sampling of contacts, switches (mechanical or solid-state), thumbwheels; single-point or multipoint input
- Read digital output—provides a convenient form of status read-back.

Industrial Basic runs on any PDP8-E, -F or -M computer. A user's manual is available. This software is sold as part of the 8/C Industrial System, and it is also separately available for $1200.

A string-processing language

Just as Fortran is designed for numerical calculations, Interdata's version of SNOBOL is designed for string processing. Strings are simple lines of data having arbitrary length and content. SNOBOL relieves the programmer of concern for any string's actual location and length, while it maintains a dynamic pool of storage for all strings. SNOBOL's basic functions are to locate strings (by pattern-matching) and to manipulate them as needed. Additional functions include program flow control, radix conversions, and string input/output.

Engineering applications of SNOBOL include:

- Wire run-list processing. Such items as signal-names and pin-names are strings. Manipulations like run-list editing and formatting can easily be written and executed in SNOBOL.
- Test equipment control. Commands and responses encountered when communicating with test equipment often lend themselves to string-structuring and thus to SNOBOL programming.
- Telecommunications. SNOBOL is ideal for utility background tasks such as billing, and editing in a communications environment. However, because of a rather large amount of time overhead, SNOBOL is not suitable for most heavy load communications applications.
- CRT data base inquiry. Just about all communications with a CRT are string-structured and are thus ideal SNOBOL applications.
- Data acquisition and process control. When this type of data is string-structured and very fast data rates are not needed, SNOBOL would be applicable.

Interdata SNOBOL runs on any configuration of Interdata models 5, 50, 70, 74, or 80 machines. A user's manual is available. Price quotations for this software are available on request.

An extensive Basic library

A wide variety of minicomputer programs, written in the Basic language are available from Hewlett-Packard Corp.

In electrical engineering areas of interest, for instance, the list includes programs for:

- A-c circuit analysis
- Low-pass filter design
- Microwave parameter conversion
- Mixer spurious-response calculation
- Active filter design
- Time function evaluation
- Analysis of a balanced polyphase induction motor
- Network simultaneous-equation solution
- Electric field strength calculation
- Linear programming model
- Fast Fourier transforms
- Bessel functions of the first kind
- Simultaneous-function graphs

These programs were submitted by users of HP-2100 computers. At the H-P Software Center, the programs are test run, checked and are then made available for distribution.

Program library handbooks are available. These contain complete documentation—including listings—for most of the programs. Individual programs are available on punched-tape—with documentation—for $10 each. Magnetic tape and disc versions are also available.

Material for this article came from many sources, but inputs from Wilbur Highleyman (Minidata Services) and Bernard Lacroute (Digital Equipment Corp.) were especially helpful.

Computer report V

Power supplies: how to trim the budget

Orphaned in the world of decreasing cost-per-bit, power conditioning equipment cries for cost reduction

In these energy-short days of the mid-1970s, consumers of electric power face the limits of quantity as well as quality in meeting current needs. Computer systems don't require great amounts of power, but they do have very particular appetites that must be met if meaningful operation is to be maintained. With only a brief interruption or fluctuation in diet, these complex machines can quickly forget all their clever tricks.

Yet curiously enough, the design of computer power supplies remains an area ripe for further economy and cost reduction. *Spectrum*'s survey of supply vendors, users, and computer manufacturers doing their own power engineering, disclosed strong opinions on overpriced hardware and underutilized low-cost designs. These factors are significant because power supply investments often total 25 percent of a computer system's cost while the engineering effort expended to achieve improved design can be comparatively small.

Costs, cuts, and components

There are three basic component areas in which most everyone making computer power supplies feels cost reductions are due—power semiconductors, magnetics, and electrolytic capacitors. Sola Electric Division, Sola Basic Industries, Elk Grove Village, Ill., picks the semiconductor area as already exhibiting a trend in this direction because power switching transistor prices have dropped 20 to 30 percent per year over the past three years. For example, a typical silicon power switching transistor in use today—with $V_{ceo} = 300$ (sustaining), 15-ampere collector current, and a current gain-bandwidth product of greater than 5 MHz—now sells for around $2. Reliable devices of this type should be available for about 50 cents within four or five years.

Fast-recovery power diodes were identified by ACDC Electronics (Oceanside, Calif.), Sperry Univac (Blue Bell, Pa.), Electro Module, Inc. (Pomona, Calif.), Honeywell (Billerica, Mass.), and Prime Computer (Natic, Mass.), as another solid-state component group where significant savings could and should be achieved. And ACDC Electronics indicates that desirable performance specifications for high-speed power rectifiers are: forward drop (V_F), less than 1 volt at 50 amperes; recovery, less than 200 ns; reverse leakage at 30 volts, 100°C case temperature, less than 10 mA; and price, less than $5.

Finally, both Sola Electric and ERA Transpac Corp. of Moonachie, N.J., included SCRs—particularly those used for protective circuitry in high current supplies—under the heading of high-cost components. Discounting inflation or unforeseen technical breakthroughs, Sola expects a 50-percent reduction in SCR prices over the next five years as applications expand for the less highly regulated type of supply in powering computers, office copying machines, and heating equipment.

Magnetics, the second category of perhaps overly expensive components, includes transformers and chokes which make up the unwieldy bulk of many large supplies operating at 50 or 60 Hz. Interdata (Oceanport, N.J.) has found that the ferroresonant-type power transformer for 50–60-Hz operation is particularly expensive in the large sizes required to handle greater than 500 watts. Reduction of size and expense could lie in high-frequency (1–20-kHz) operation to eliminate the need for large inductors. Interdata's approach is to design them out by going to direct off-line regulation and running a high-frequency inverter for isolation and step-down.

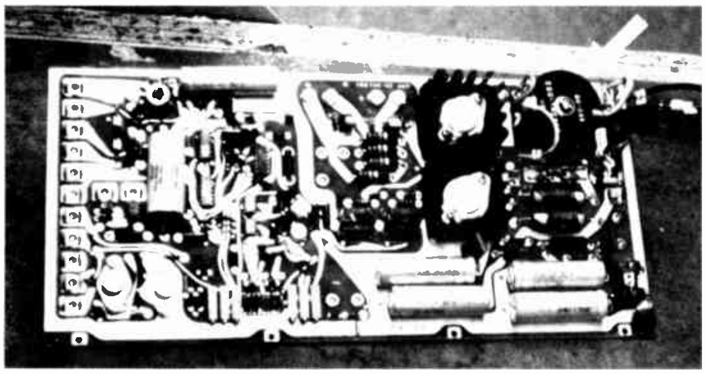
Unlike solid-state devices, the value added during the manufacture of magnetics is small, with costs breaking down primarily into materials (copper, steel, aluminum) and labor procurement, says RO Associates, Menlo Park, Calif. Experience and economics teaches that such items only become more dear with time, and some manufacturers of supplies using dissipative-type regulators at 50 or 60 Hz have tried to compensate by using off-shore magnetics. Reliability of low-priced "bottom line" units has suffered as a result. RO Associates feels that, once again, switching-type designs operating above 1 kHz offer the best opportunity for increased efficiency and size reduction.

Aluminum electrolytic capacitors were named by Sola Electric, Electro Module, General Automation, Anaheim, Calif., and Tel-Dynamics, Fort Washington, Pa., as the third prime target for component cost reduction. But the shortages of etched aluminum foil affecting capacitor manufacturers are well known, and at least some capacitor customers are resigned to the fact that prices may go still

Don Mennie Associate Editor

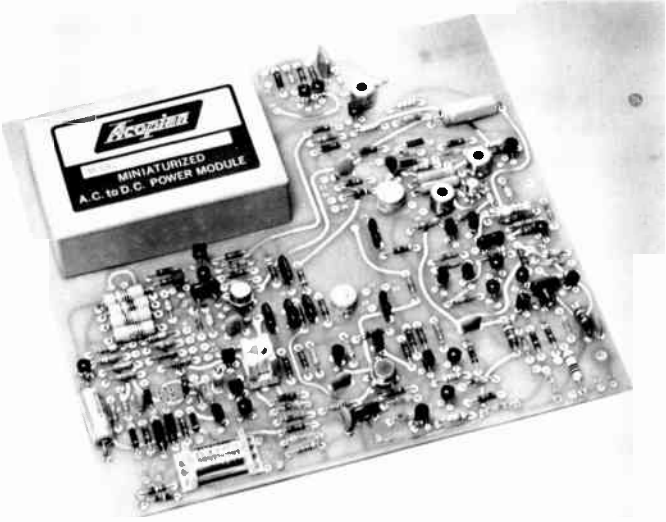
higher. Though timeworn, about the most practical suggestion on cutting capacitor costs comes from Cornell Dublier, New Bedford, Mass. They advise using EIA-listed standard capacitors wherever possible. These are widely second-sourced, far easier to obtain, and cheaper than the vainly sought-after specials.

For capacitor applications not involving electrolytics, Cornell Dublier recommends two handy references. EIA publication RS 401 "Paper, Paper Film and Film Dielectric capacitors for power semiconductor applications" contains data and specifications aimed at engineers designing high frequency switching supplies. Of similar origin, RS 392 "Fixed paper dielectric capacitors for ac applications" should be useful to those building supplies around ferroresonant transformers. Copies of these standards may be obtained from the Electronic Industries Association Engineering Department, Standards Orders, 2001 Eye Street, N.W., Washington, D.C. 20006.



A look inside this switching supply from Interdata illustrates the kind of weight reductions made possible by high-frequency operation. The 10.4-kg supply uses a transformer that tips the scales at a scant 227 grams. An arrow points out these magnetics among circuitboard components (upper righthand corner).

Miniature power modules from Acopian Corp. mount directly on printed circuit assemblies, eliminating outboard supplies plus associated wiring, connectors, and hardware. Model 5E150 shown here delivers 5 volts at 1.5 amperes, but ±15-volt tracking power modules are also available. Prices range from $49 to $105 per module.



Season of the switch

After careful selection of standard parts (if you're a manufacturer) or avoiding the pitfalls of overspecification (if you're a customer), much of the remaining opportunity for savings comes from system designs made cost effective by switching circuits, spot regulators, or simply by relaxed specs. Many loads, such as servo amplifiers in peripheral equipment, can often operate satisfactorily from unregulated supply voltages.

Hearty endorsement for the switching approach comes from Electro Module commenting that "the only realistic way to reduce cost is through the application of switching techniques in power supply design." Although switching circuits are more complex and the components more expensive when compared with standard supplies, the decrease in size, weight, and thermal stress (reduced heat sinking), and the resultant increase in reliability, is said to produce the optimum system.

Sola Electric also favors the high-speed switching approach and claims a substantial size/weight reduction (typically more than 50 percent) for one of its new high-frequency (20-kHz) designs featuring switching transistors rather than the low-speed silicon power transistors now in common use. Presently, the cost of a 20-kHz unit is comparable with the bulky series-regulator-type supply. The benefits of reduced magnetics and filtering (at 20 kHz) are offset by the greater cost and lower reliability of high-frequency switching transistors. Though purchase price savings on switching supplies will not be fully realized until improved, low-cost switching transistors are in volume production, the power supply customer now has the available option of a small, light, cool-running supply which allows immediate cost reductions in the related areas of end item equipment design and manufacture.

Local regulation?

Monolithic voltage regulators used on individual circuit boards (spot regulation) offer another way to boost performance of unregulated supplies while significantly reducing cost and packaging problems, according to Optical Electronics, Inc. (Tucson, Ariz.) and Sola Electric. General Automation and ERA Transpac Corp., also aware of mechanical layout problems, say supply packaging has long been a secondary consideration to computer packaging. Such neglect can often lead to requirements for premium parts and difficult assembly techniques.

Microdata Corp., Irvine, Calif. favored the three-terminal regulator for simplifying protection and sequencing circuits because thermal shut down and current fold back are built in. Specifically mentioned was National Semiconductor's LM 309, which also provides overvoltage protection.

But Standard Power and ERA Transpac indicate that discrete regulator assemblies—installed as needed throughout the computer circuits—offer the best means of cost cutting. With power supply components tucked away at convenient places throughout the configuration, heavy cabling is eliminated and single transformers with multiple outputs may efficiently serve a variety of circuits. In actual usage, however, external capacitances, control networks, rectifiers, heat sinking, and mounting facilities were felt to water down the advantages offered by integrated circuit spot regulation.

Not for everyone!

But spot regulation is no all-purpose panacea, and the experiences of several other survey respondents show this method at a disadvantage where an optimum dollar watt configuration is important. Citing consumption of premium pc board space, a need for complex sequencing circuitry (for orderly computer shut-down), difficult trouble shooting, and low efficiency, General Automation summarizes the shortcomings of distributed power systems. Honeywell also concludes that on-the-chip distributed power supply circuits do not appear too attractive for use on [Honeywell] logic boards because each board draws about 2.5 amperes, implying several regulator chips would be needed.

Critics of regulator chips for pc board use mentioned that these IC's can be helpful in regulating control voltages for support circuitry within central supplies themselves.

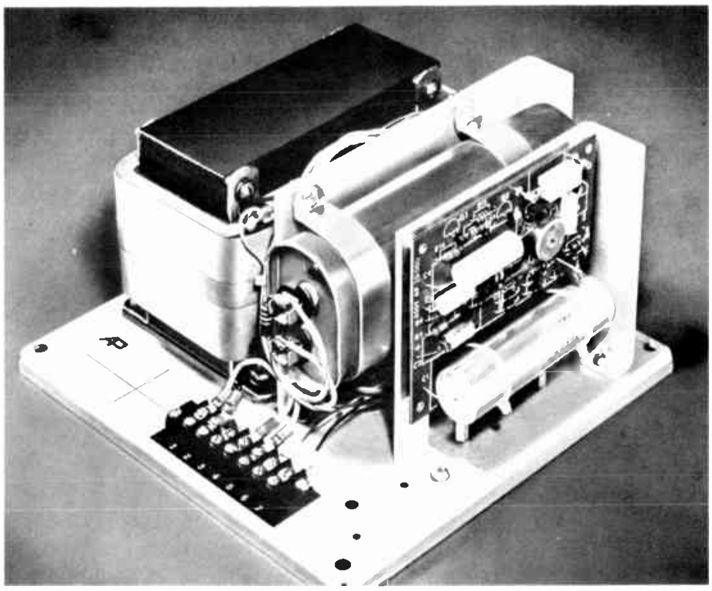
Wants and needs

The fewer frills and extras a power supply has, the more inexpensive it becomes. Among the higher power devices, such support circuitry for adjustment, protection, and sequencing is a minor cost consideration because it isn't closely related to total output capability. But support circuitry can be a significant expense in low- and medium-power units. Concluding there is little any *single* manufacturer can do, Sola Electric believes that industry-wide standardization of support circuitry could reduce costs significantly. In truly standard, modular, off-the-shelf form, Sola expects supporting circuitry could be available at half its present cost.

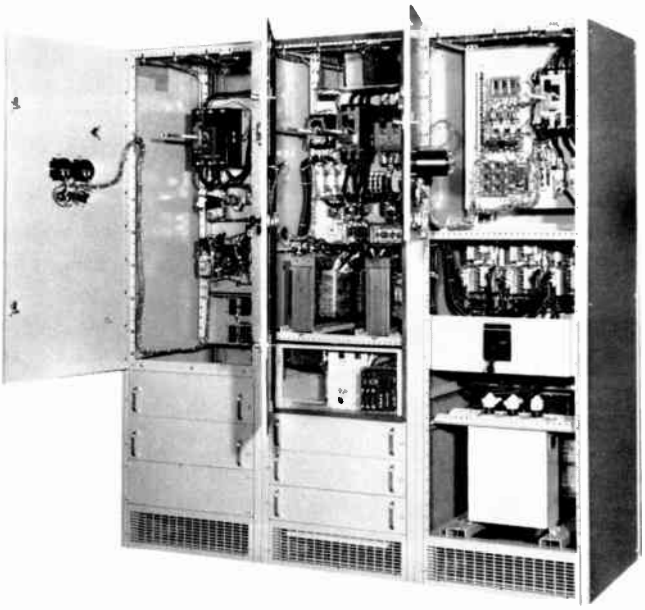
Progress in this direction is evidenced in the computer field—where industry followers do all they can to make their products compatible with the leaders—and in the recent NEMA Standards Publication PY1-1972 "Stabilized power supplies direct-current output," which summarizes industry-wide terms and definitions.

When pruning the power supply budget, Acopian (Easton, Pa.) asks consideration for the narrow adjust or "slot-type" supply. It's less expensive and more efficient than one which provides the equivalent current through a broad voltage range. In the same vein, Standard Power (Santa Ana, Calif.) has reintroduced the unregulated supply for such noncritical tasks as driving relays. One of their $14 units offers a volt-ampere output that would cost $60 to $75 in a highly regulated version. Where better performance is needed, Standard Power maintains that 0.1–0.2-percent line and load regulation is sufficient for most computer and digital system requirements. But it's not unusual to find customers requesting 0.01-percent regulation to drive TTL circuits when in fact 1.0 percent is adequate.

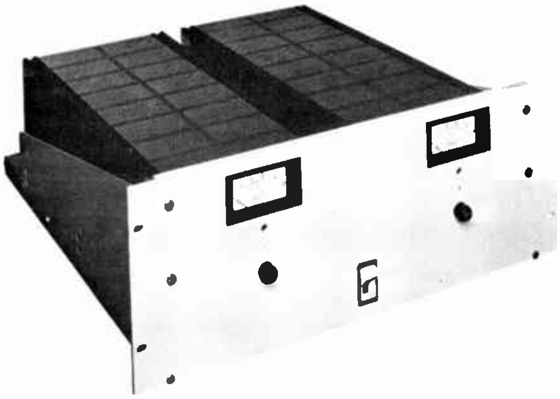
Building a supply rather than buying one allows savings in the selection of supporting circuit components. On low power supplies, Interdata uses plastic SCRs for crowbars following a current-limited output with a series fuse to protect the load and SCR should the limiter fail. Their high current supplies have a magnetic sensor which activates over-current protection without dissipating much power, unlike similar protection using a current sense resistor. Electro Module and Honeywell favor integrated circuits of increasing complexity—such as quad comparators—to reduce support circuit costs. CMOS circuits, exhibiting high noise immunity and consuming few watts, are an excellent choice for this application. Constructing



Central to Advanced Power's Controfluxer™ line is a new ac voltage regulator, available alone or with full dc output circuitry. Basically, this ac power conditioner is an oil-filled capacitor in series with a specially designed transformer. Output power (117 volts ac or 234 volts ac) is derived only from the capacitor's charge, which is alternately drained through the magnetics and then refreshed by the raw ac input (60–130 volts ac or 120–260 volts ac). DC regulators driven from this stable ac need not be designed for input variations, and the voltage drop across the pass transistors can be reduced accordingly —boosting efficiency. A feedback control accommodates 45- to 66-Hz line frequencies for truly international flexibility. Versions operating at 20 kHz and competitive with conventional switching regulators will be introduced in six months.



In critical computer applications, where any nonscheduled outage will cause severe economic loss, an uninterruptible power supply (UPS) is a necessity. The Westinghouse UPS shown here provides rectified line or direct battery power to a common dc bus. This dc drives an inverter which supplies continuous current to critical ac loads. For a three-phase rack-mounted UPS using a 5-minute battery, costs range from $500 to $1000 per kVA.



Hardwired to a common output, either of the two power modules in this redundant system from Acopian Corp. may continue to provide rated voltage and current should one section fail. This scheme provides extremely high reliability where two independent sources of commercial ac power are available. Standard models priced $695 to $1095 provide 3.6- to 28-volt, 5- to 32-ampere dc output.

computer memory and logic with CMOS is also considered a good way to reduce power supply costs. All that's needed to power a CMOS-built computer is a linear transformer, a rectifier, and a filter according to the Digital Equipment Corp., Maynard, Mass. The CMOS-powered circuits can take a 40-percent variation in voltage, and supplies with that kind of regulation are very inexpensive.

Power in parallel

When a single solid-state device can't deliver the required load power, the use of devices in parallel—sometimes as many as ten transistors at a time—is one workable technique for obtaining more current and simplifying heatsink problems. RO Associates is cautious about paralleling power transistors and reminds designers that besides the use of emitter resistors for load splitting, some kind of matching is usually required for optimum operation of the semiconductors. While the need for emitter resistors for current sharing was recognized unanimously by manufacturers surveyed, Interdata explained that using unsorted transistors that have *not* been selected for matched forward base-emitter voltages (V_{be}) does have some advantages. Although the resulting design is less efficient, this method avoids field replacement problems and the cost of selected transistors sets.

ACDC Electronics claims extensive use of parallel power transistors in series-pass regulator power supplies outputting up to 100 amperes. They also mention paralleling transistors in switching circuits as another possibility, with the principal requirement still being emitter resistors. Sperry Univac reports a significantly reduced MTBF with parallel transistor operation.

SCR applications

When they have a choice, power supply designers prefer power transistors over SCRs because of the latters' commutation problems and propensity for generating electrical noise. Sola Electric says only considerable on-site testing can determine the exact nature of such troubles, which are then cured with a custom retrofit. But where very high output power is desired, SCRs can provide the most efficient and practical approach.

Interdata recommends SCRs as switching elements in supplies operating up to 1 kHz and delivering over 2 kW. At very high frequencies, SCRs are not considered practical because of slow recovery time. ACDC Electronics finds SCRs useful for very large phase control circuits at power frequencies of 50–400 Hz. They are also recommended for support circuitry applications as crowbars or slow-start rectifiers.

One point up for debate among the companies surveyed is the practical high-frequency limit for SCR switching operation. ACDC Electronics picks the upper cutoff as 5 kHz, while Honeywell indicates very favorable experiences operating between 10 and 20 kHz. Interdata is the most conservative with their previously mentioned maximum of 1 kHz. Apparently, as RO Associates reflects, there is still a lot of "art" in the practical application of SCRs.

No interruptions

Dips, transients, brownouts, or interruptions in power—even for only a few milliseconds—can cause long-term shutdowns of many complex computers and data handling systems. Spilled tapes must be rewound and programs reloaded into scrambled memories (see *Spectrum*, Nov. 1973, pp. 76–81). Solving this problem successfully means coping with the bulk and expense of an uninterruptible power supply (UPS).

Sola Electric produces such a system and their UPS uses a line-power dc supply feeding an inverter that is always "on." When the power supply senses line voltage shifts beyond a preset threshold, it switches to battery power. There is enough energy stored in the supply's filtering to hold up the inverter output through the sensing and transfer period, so the computer never sees any power interruption. But this lone advantage can cost many extra kilowatts.

Sola's UPS system, most efficient under full load, will still suffer about a 20-percent efficiency loss because some power must be dissipated maintaining the inverter "on." This percentage loss will be even higher in systems operating at less than full load, where the UPS is designed to allow for future expansion. For applications which can tolerate up to 20-ms outage, Sola makes a momentarily interruptible power supply which is far more efficient than their true UPS.

Where interruptions are not tolerable, Honeywell is in basic agreement with Sola. They state that the only present viable approach for uninterruptible power sources involves the extensive use of continuously charged batteries either to support the dc input to a static inverter, which in turn supplies ac to the system, or to support the dc output of large supplies feeding local dc regulators.

The basic inverter-type UPS scheme is also favored by ERA-Transpac and Interdata. But Electro Module, RO Associates, and ACDC Electronics endorse a variation based on high-frequency (20-kHz) dc to dc conversion where the outputs are the required computer system voltages. This eliminates the expensive and inefficient magnetics needed for 50–60-Hz, 115-volt ac inverter operation. Other benefits include no audible noise (singing) and no discontinuity on switchover, since the long drop-out time of the regulated converter allows transfer to battery operation without interruption. Motor-generator sets can take over from battery systems if commercial power cannot be restored quickly.

Acopian suggests that redundancy is an overlooked factor in insuring that computer system power is not interrupted by line failures or transients. When two independent ac sources are available, each power supply of a redundant pair may operate from a different source, so problems on just one line will not affect system operation. Satellite tracking stations, data communications networks, and critical process control systems are often powered by such an arrangement.

Customers byte vendors

Computer power supply designers were asked by *Spectrum* to assess their working relationships with component manufacturers. And the parts vendors got generally unfavorable reviews regarding their willingness to make changes or improvements which would bring down the cost of computer power conditioning equipment.

About the most generous comments came from Sola Electric, crediting TRW, Motorola, General Electric, and Elrad Manufacturing as the handful of industry leaders doing research and aggressively seeking improvements in size, weight, performance, and cost. General Automation termed component manufacturers "fairly responsive to user needs" but underlined that improved price performance is required in high ripple current aluminum electrolytics, high-voltage, high-current switching transistors, and high-current Schottky diodes.

Electro Module, however, says component manufacturers do not understand the needs of computer systems and described a newly announced low-impedance electrolytic as a prime illustration. This capacitor has a ripple current rated at less than a quarter of that appropriate for its size and capacitance when load current and permissible ripple voltage are the selection criteria. Listing delivery and reliability problems, IBM claims capacitors for high-frequency (20-kHz) operation are not well designed or properly tested by manufacturers.

Semiconductor houses were also cited for a lack of practical understanding, with the conclusion that their specification sheets and application notes are virtually useless. Electro Module insists maintaining an intensive knowledge of internal transistor structure is necessary in order to design reliable computer power supplies.

Sperry Univac complained that component manufacturers were not responsive, especially in the area of hybrid power devices. After an order is placed, experience has shown that many such advertised items are available in limited numbers only—with no commitment to build more!

ACDC Electronics homed in on the economics of the situation with the all-too-common observation that the component manufacturer's present ability to produce is not up to the demand. This results in poor motivation for cost improvements.

Tele Dynamics/Wanlass was also pessimistic, confirming that component manufacturers are not responsive to new or special designs. Costs are higher where deviations from catalog items occur—or techniques change.

But are these complaints simply a result of poor motivation or apathy on the part of component vendors? Cornell Dublier doesn't think so, at least where capacitors are concerned, and says that good applications data or technical consultation is available from several of the largest suppliers—including themselves, naturally! As they see it, anxious customers often pursue parts distributors or company sales representatives who are usually not equipped or qualified to help with technical problems. Unfortunately, under such circumstances ignorance is not usually admitted, and attempts are made at some sort of field fix. Burdened with misinformation, the customer becomes confused and angry.

The answer? Go directly to the capacitor manufacturer's applications engineers with any important questions. Insist on talking with someone capable of making practical suggestions.

spectrol
TRW
POTENTIOMETER BUYER'S GUIDE AND ENGINEER'S HANDBOOK
BOURNS
TRIMPOT® PRODUCTS
ADJUSTMENT POTENTIOMETERS
PRECISION POTENTIOMETERS
VARIABLE RESISTORS
CTS
electronic components
CTS
DALE
DALE ELECTRONICS, INC
AB
ALLEN-BRADLEY
Helipot Components
DAYSTROM POTENTIOMETERS
BOURNS

Computer report VI

Minicomputer interfaces: know more, save more

When a standard interface won't do the job, you can pay for a custom design . . . or do it yourself

Attractive price/performance ratios for minicomputers lose some of their appeal once a potential user learns that individual peripherals required to complete a useful system—input/output (I/O) devices, mass storage units, etc.—may cost as much or more than the minicomputer itself. If a potential buyer is not dissuaded by that realization, he may be when he also learns that the peripherals may be useless without special interface devices; devices that he may have to design and build himself.

To understand the interface problem, it is important to note the clear distinction between interfaces and controllers. Referring to the typical minicomputer system configuration shown in Fig. 1, note that the controller is that portion of the electronics which is dedicated to controlling the peripheral device itself. For example, a line-printer controller includes electronics for print-head selection, paper motion, print timing, etc., all of which apply to the line printer itself once it has received data and a command from the interface logic.

Interfaces, as a rule, contain address decoding logic since many interfaces may attach to the same I/O bus. Since processors usually run at speeds much greater than peripherals, the interface is required to act as a synchronizing unit during data transfer. Level conversion, command decoding, multiplexing, and data request logic are all interface functions.

Data transfers between processors and peripherals are usually handled by interfaces in one of three modes or methods: under program control, interrupt control, or direct memory access control.

Program-controlled data transfer

Transferring data under program control requires the least amount of logic in the interface unit but requires constant monitoring by the processor. Once a data transfer is initiated, the processor is required to monitor the ready status to see when a transfer can again be made. It is up to the processor to keep close tabs on everything that is taking place in the interface unit.

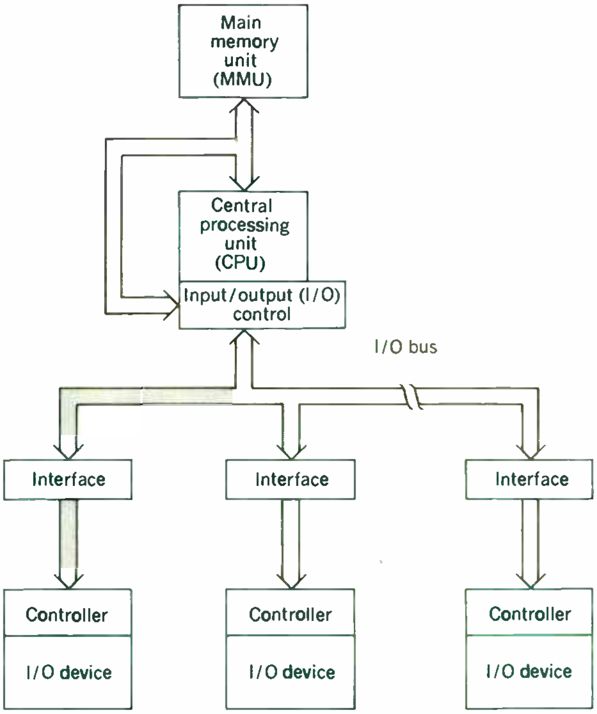
The simplified block diagram of a typical interface unit shown in Fig. 2 and the typical timing signals on the I/O bus shown in Fig. 3 are helpful in understanding program-controlled data transfer. (Fig. 2 is also used in describing the other two data transfer modes.)

In general, there are four types of I/O instruction sequences involved in programmed data transfers: control, sense, data out, and data in. Control sequences—read, write, and stop, for example—are sometimes referred to as order-out or command-out sequences. (Control and command will be used interchangeably throughout the following discussion.)

Sense instructions are used for testing various status conditions in the interface, controller, and peripheral. Before a transfer is made, for example, the program usually tests the peripheral to see that power is on and that the selected peripheral is connected online to the processor. The types of instruction associated with various tests are: jump on response (skip

A. A. J. Hoffman Texas Christian University
Robert L. French Avcon, Inc.
Guy M. Lang Dallas, Texas

[1] Typical minicomputer system configuration for I/O functions. Not shown, but often included, are mass storage units such as magnetic-tape or magnetic-disc devices. The CPU performs the logic, control, timing, and arithmetic functions of the computer. The MMU stores both instructions (the program) and working data for ready and immediate access by the CPU. The I/O devices serve as a means for transmitting information (data and instructions) to and from the computer.



Who has the broadest line of Potentiometers?

It stands to reason that we wouldn't ask the headline question unless we were certain of the answer.

And we are. Unquestionably, the answer is BOURNS

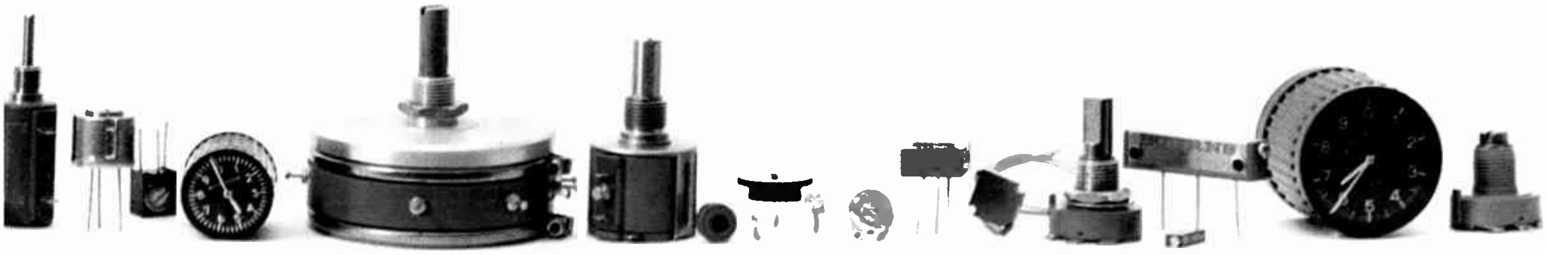
Compare catalogs if you like. You'll find over 12,000 model/terminal/resistance combinations offered in our ***TRIMMER POTENTIOMETER*** section. Single-turn and multi-turn . . . round, square and rectangular — BOURNS has them all, in wirewound or cermet models.

No need to look elsewhere for a complete line of ***PRECISION POTENTIOMETERS***. Check the precision section of the Bourns catalog. Over 20 different standard bushing mount models are displayed, including the unique KNOBPOT® integral dial/potentiometer family. If you have servo-mount requirements, or need "something special" . . . our fully staffed custom design and manufacturing organization is at your service. Promptly.

More? You bet.

Bourns popular "SPACE SAVER" line of cermet ***CONTROLS*** are finding wide acceptance in cost-conscious consumer and industrial applications. You can expect to see exciting developments in this line.

Our cermet and composition open frame ***VARIABLE RESISTORS*** are super for budget PC board applications. They're priced as low as 10¢ in production quantities. They're from Bourns, so you can count on cost-effective quality and service . . . even at such low prices.



When you think of variable resistive components . . .
look no farther than your Bourns Master Catalog. It's your
supermarket of quality, cost-effective resistive components.

BOURNS INC

TRIMPOT PRODUCTS DIVISION • 1200 COLUMBIA AVE., RIVERSIDE, CALIF. 92507

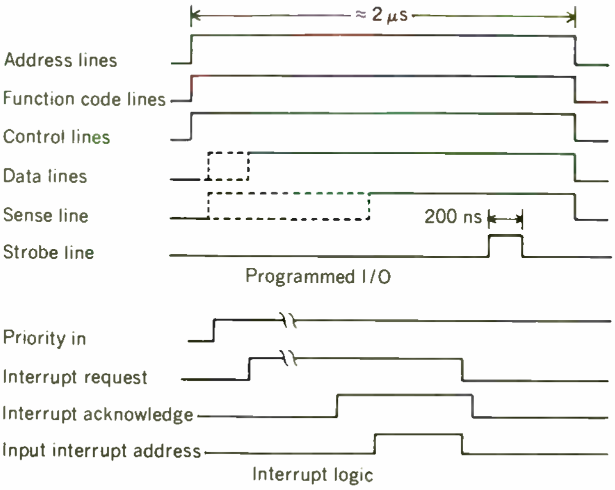
I. Typical sense instructions that are applicable to a magnetic tape unit

Function Code Lines			
FC1	FC2	FC3	Type of Test
0	0	0	Device or controller ready
0	0	1	Record gap
0	1	0	Tape position at load point
0	1	1	Tape positioned at end of tape
1	0	0	Parity error
1	0	1	Data available (in or out)
1	1	0	Command rejected
1	1	1	Write protected

the next instruction; instead jump to the specified instruction) and jump on no response. Figure 3 is helpful in understanding the sequence of events during these tests. The I/O control unit in the processor places a particular device address on a set of address lines. There are typically five or six address lines that give the processor the capability of addressing as many as 64 different devices. Each interface attached to the I/O bus contains an address decoder which monitors continuously the address lines. When the interface detects its own particular code combination, it enables the various other functional blocks within the interface. At about the same time that the address is made available on the address lines, the I/O control generates function and control signals. Generally, there are three function code lines and four control lines. The control lines pass information to the interface unit telling it whether the present instruction is a command, a sense instruction, a data-in transfer or a data-out transfer.

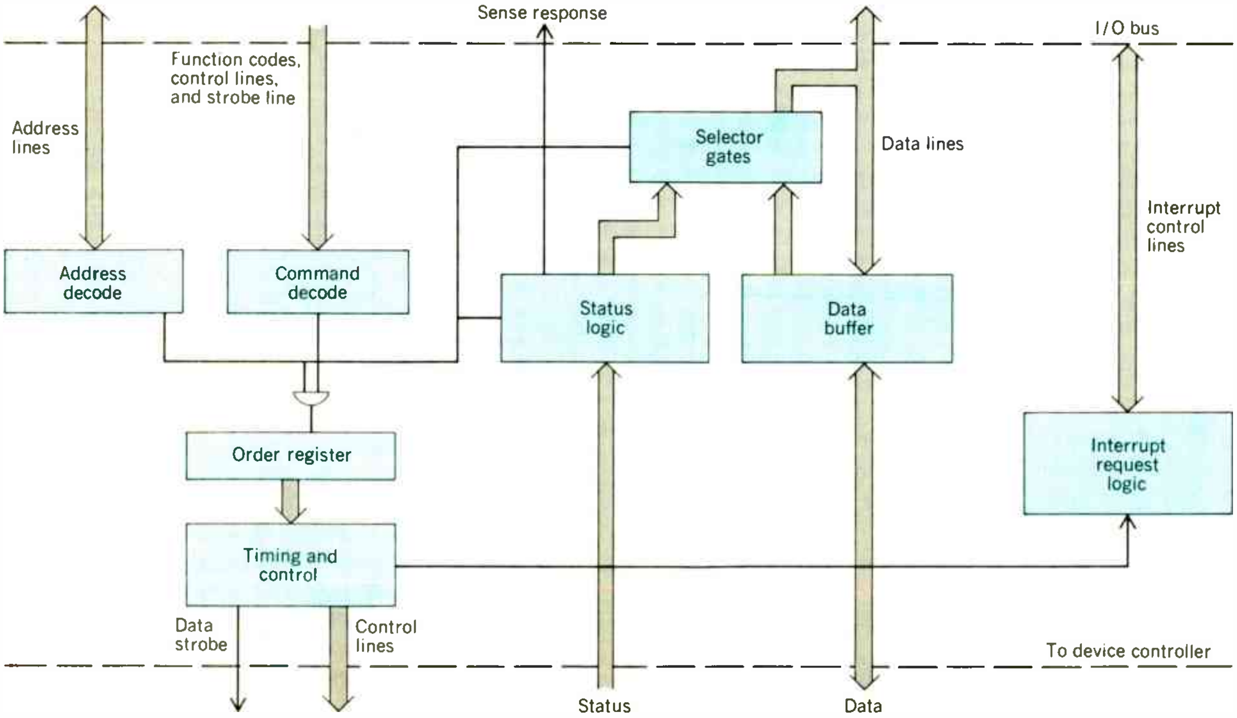
During a sense instruction, the function code lines tell the device what status condition to test. As an example, Table I illustrates certain tests applicable to a magnetic tape unit.

A function decoder in the interface recognizes which status condition is being tested, selects that condition, and gates a sense response line back onto the I/O bus if the condition exists. The I/O control unit monitors this sense line and causes the program to skip or not to skip the next instruction in the program. The total time involved in an instruction sequence of this type varies from 0.8 to 8 μs, depending upon the particular processor. The only timing consideration is that the interface is usually required to respond by the 50-percent time in order to be recognized. This instruction is terminated by lowering the



[3] Usual, but not all inclusive, timing signals on the I/O bus for a minicomputer.

[2] Interface functions for a minicomputer.



address, command, and function code lines. The interface responds by removing the sense line signal.

During a command-out instruction, the I/O control unit again raises the appropriate address lines, function code lines, and command line. Once again the interface decodes its address and enables other logic to come into action. The particular command issued again depends upon the condition of the function code lines. Table II gives a typical command structure for a magnetic tape.

II. Command instructions for a magnetic tape unit

Function Code Lines			
FC1	FC2	FC3	Type of Command
0	0	0	Write a record
0	0	1	Read a record forward
0	1	0	Go to load point
0	1	1	Backspace one record
1	0	0	Write end of file
1	0	1	Read a record in reverse
1	1	0	Erase forward
1	1	1	Stop write

At some point during the instruction cycle, the I/O bus also generates a strobe pulse. It usually occurs when all the other lines are assumed to be stable and free of transition edges to insure that the propagation delays within the logic are accounted for and the correct command will be strobed into the order register in the interface. The strobe pulse itself is used to strobe the order register. The I/O control unit terminates the cycle, as before, by removing the various address, command, and function code signals. Some processors condition the strobe pulse generation or a sense response from the interface unit.

Once a particular command is accepted by the interface, it has the responsibility to decode the command and issue a similar set of commands to the peripheral controller. With a magnetic tape unit and a write command, for example, the interface is required to issue to the tape controller a forward drive signal, a write enable signal, and an erase enable signal. These signals start the tape unit in motion and cause a data block separator—usually an erased gap between data blocks—to be written on the tape.

In a data-out transfer, which is similar to a command-out instruction, the data lines are strobed into a data output buffer or holding register. A typical sequence can be described by again considering the magnetic tape unit during a writing operation. If the tape transport is an 800 b/in, 25 in/s unit, it requires a data transfer rate of 20 000 (800 × 25) bytes/s. Most processors, however, can transfer data at a much higher rate. In order to write the data in a form that can be recovered at a later time, the tolerance of the 20 000 bytes/s must be maintained within close limits.

Once the magnetic tape has been started, and the gap generation is in process, the interface generates a signal called data buffer ready or data not available. Under program control operation, the processor is constantly monitoring this status by means of a sense instruction. When it sees a data buffer ready response on the sense line, the processor issues a data out instruction. The interface responds to the address lines and, when the strobe appears, clocks the data lines into an output buffer or holding register. At the same time, the interface lowers the data buffer ready signal mentioned earlier. The data are held by the interface until a data block separator has been written on the tape. The interface then transfers this byte to the tape controller.

The program, meanwhile, has incremented a word counter and tested the count for completeness. If the count is not complete, the program jumps back to the sensing data buffer ready signal. Once the data have been transferred from interface to controller, the interface again raises the data buffer ready signal. The process is repeated again and again until all bytes have been transferred. Then the program causes a write stop command to be output. The interface is then responsible for writing an additional erased gap, delaying while the gap is being generated, removing the forward drive signal from the tape controller, delaying again while the tape motion ceases, and then raising a ready signal.

A data-in instruction is just the opposite of a data-out instruction. In this case, the interface accepts data from the controller, places the data in a holding buffer, and raises a line called data available. When the processor sees this line come true by means of a sense instruction sequence, a data-in instruction is issued. The address and control lines are used to gate data from the interface onto the I/O bus data lines. The strobe on the I/O bus is used to indicate to the interface that the data have been accepted. The interface then lowers the data available signal and the process is repeated until the record gap is detected, at which time the operation is terminated.

Some of the additional interface functions are parity generation, parity error detection, unusual end conditions, and timing rate errors.

Interrupt-controlled data transfer

Transferring data under interrupt control is more complex for the interface. In this case, it is the responsibility of the processor to initiate the process and, then, it is the responsibility of the interface to inform the processor when to make the data transfer. In a typical data transfer process implemented under interrupt control, the processor determines from its operating program when it is time to output a block of data to some peripheral. The processor initiates the transfer by enabling interrupts from the selected peripheral and by setting up any required flags and counters at a specified subroutine location in core. A start I/O command is then issued to the interface and the processor proceeds to its next task. The interface, meanwhile, is monitoring the peripheral. When the interface determines that the peripheral is ready to accept a data word or byte, the interface generates an interrupt request to the processor. Upon seeing an interrupt request, the processor halts the task it is doing, jumps to a subroutine for processing the request (in this example, the processing consists of outputting a data character), and then returns to the task that it was performing.

During the interrupt process, the interface is usually required to provide the address of a jump to subroutine instruction, called a vector address. The interface is also required to provide the necessary logic for generating the interrupt request and recognizing that it is being serviced. This method of transfer frees the processor for performing other tasks between transfers. The software is not required to monitor the interface continually and allows several tasks or peripherals to be controlled simultaneously.

Here, in some detail, is how a typical interrupting sequence is accomplished. Assume that a command has been issued to some device to cause a series of data outputs (e.g., write a record on tape). The processor then proceeds to another task. Once the interface issues the appropriate signals to the tape controller, it proceeds to issue an interrupt request to the processor (it is assumed that, at some prior time, the processor has enabled the interrupt control logic in the interface). Since many devices in a system are capable of generating interrupt requests, a priority technique is usually required. Figure 3 shows a signal line on the I/O bus called priority in. This signal first passes through the interface with the highest priority. If it does not require an interrupt, it passes the priority onto the second interface in the chain. The priority is passed along until some interface issues an interrupt request. The chain is then broken.

Assume that the device in question has a requirement for additional data. When this requirement is true and the priority-in signal is true (meaning no higher priority devices are requesting service), the interface causes the interrupt request line to become true. Most processors are capable of responding to interrupts with a signal called interrupt acknowledge after every instruction cycle or at least within a few microseconds. When an interface receives the interrupt acknowledge signal from the processor, the interface responds by gating an interrupt subroutine address (vector address) onto the I/O bus data lines. The processor accepts this address and jumps to the program at that location and begins execution. Once this happens, the interface no longer needs service and, consequently, drops its request. The subroutine usually contains sense instructions used to determine why the interface generated the interrupt request in the first place. For instance, if it sees that a data transfer is ready, it will check a word counter and either issue another data out command or a write stop command.

Figure 2 shows a block called interrupt request logic required for sequencing an interrupt request.

Direct-memory-access control

In direct-memory-access control, as the term implies, the interface transfers data directly to the memory unit. The processor initiates this type of transfer by supplying the interface with a starting memory address and a word count and then proceeds to other tasks. When a transfer is to be made, the interface requests a memory cycle time and, when granted, causes data to be transferred directly to memory. The processor merely halts for one memory cycle time and transfers are made completely transparent to the program. This method is by far the fastest and most efficient from a processor utilization standpoint but, as far as the interface is concerned, is more expensive. Circuitry is required for supplying the memory address, keeping track of how many transfers have been made, and when the operation is complete.

Direct memory access refers primarily to the method of transfer. The logic that performs this transfer is sometimes referred to as a channel rather than an interface. The channel must contain a memory address register to control the location in memory to or from which data are transferred. A word counter, or other similar device, must be provided to keep track of the number of transfers performed. Logic must be provided for gaining access to the memory and for providing all of the necessary timing and control signals associated with reading and writing from memory.

Since the direct memory access channel must contain additional logic, it requires the most complex hardware of the three interfacing methods discussed. But it is probably the easiest to implement with software. Transfers are made without interfering with the operating program since the channel essentially steals a memory cycle from the processor. The channel, however, must first request and then be granted access to the memory. The memory address, as well as data, must be supplied in addition to timing and control signals. After the transfer, the memory address and word count are incremented and a test is made to see if the desired transfers have been made.

Direct-memory-access transfers are usually associated with devices that require extremely high data rates such as disc drives or processor-to-processor transfers. They are also employed when a processor is required to be servicing a large number of peripherals or is required to spend as little time as possible doing I/O processing.

Software interface drivers

Software routines for controlling peripheral devices and the transfer of data between the processor and peripherals are called drivers. Figure 4 shows a simplified flow diagram of a typical driver for transferring data from a processor to a peripheral under program control. When called, the routine first checks the device status to determine when it is ready to accept data. Once ready, data is transferred byte-by-byte (or other discrete unit) until the peripheral buffer is filled. In most cases, a control command is then given to execute the device function (print or write, for example). The process is repeated until completed. Error checking or other useful steps often accompany the simplified steps shown.

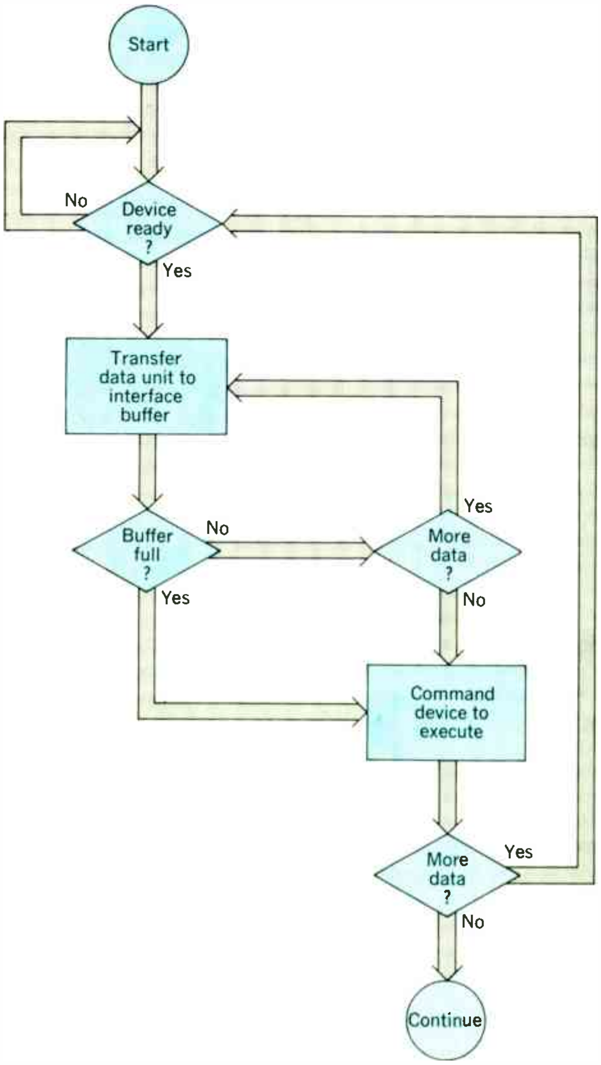
A driver for an interrupt-controlled, data-transfer interface would be structured in a similar way except that the processor would be returned to other tasks while awaiting an interrupt. Drivers may also provide buffering for input or output. This function is particularly useful in interrupt-controlled interfaces. The output from an applications program is stored in a buffer by the routine until the particular peripheral device is available. The applications program can then continue processing before the output is complete.

Driver routines may be incorporated directly into the applications programs in the case of a fully dedicated application. Normally, however, they are incorporated in the operating system. Most operating systems are supplied with a variety of drivers to support the particular line of peripherals offered by the processor manufacturer. As a result, one can often simplify the overall task of interfacing a "foreign" peripheral by designing the interface hardware to emulate a similar device already supported by the software. This technique eliminates the need to develop new drivers and to implement these drivers inside other software packages such as the editor, debugger, and assembler.

Do-it-yourself interfacing

In practically all cases, minicomputer manufacturers offer a standard line of peripherals with interfaces. The peripherals offered are usually supplied to them by reputable manufacturers and are warranted by the mainframe supplier. If, for some reason, the standard peripheral offered by the computer manufacturer does not meet the specifications required by the user or is unacceptable from a cost standpoint, there are three possibilities open to the user. After selecting the appropriate peripheral, the user can obtain a custom interface from the computer manufacturer, from a third-party source such as a systems house, or he may select to design the interface himself.

[4] Simplified driver routine for program-controlled data transfer in a minicomputer system.



The number of units required may influence a user to build his own interface. Peripherals purchased from a computer manufacturer often have a stiff handling charge or margin added to the price. The same peripheral can sometimes be obtained directly from the original manufacturer at a substantially reduced price. If a large number of these units are to be used, the cost savings may exceed the cost of the one-time engineering charge associated with the design.

If the user decides to design his own interface, what specific information is required? Usually the minicomputer manufacturer and the peripheral manufacturer both offer interface reference manuals free of charge. These manuals are quite complete with timing diagrams, signal descriptions, and may offer sample illustrations for typical interfaces. After studying the manuals, the next step is to study the I/O section of the programmer's reference manual. This second manual applies primarily when the user is interfacing a somewhat standard peripheral device where the emulation approach (the user makes his own device look—to the computer—like the similar manufacturer's device) may be used to avoid software changes or additions. In some cases, the minicomputer manufacturer does not offer drivers for particular devices (e.g., plotters). In this case, reference to the programming manual may or may not help.

The next step in the design process should be a specification of the complete instruction set applicable to the device, giving address assignment, all status conditions, commands, interrupt levels, and vector address. Once these are approved by the programming group, the task remaining is one of instruction implementation—essentially a logical formulation of gates and memory elements arranged in an order to sequence through various levels or events.

A. A. J. Hoffman (SM) is director of the Computer Science Program and professor of mathematics and physics at Texas Christian University. He joined the staff of the university in 1962 and was director of the Computer Center from then until 1973. Dr. Hoffman received the B.S. degree, the M.S. degree in mathematics, and the Ph.D. degree in physics from the University of Texas. He serves on *IEEE Spectrum*'s editorial board and, from 1965 to 1972, was editor of the *IEEE Transactions on Geoscience Electronics.*

Guy M. Lang is presently an independent consulting engineer. From 1970 to 1973, he was manager of engineering and manufacturing for Command Systems Corporation (now Avcon, Inc.). Mr. Lang's other industrial experience was with Docutel Corporation and Sylvania Electronic Defense Laboratory. He received the B.S.E.E. degree from San Jose State College and the M.S.E.E. degree from Stanford University.

Robert L. French (SM) is vice president in charge of R&D and engineering at Avcon, Inc., and its subsidiary, Hotel Computers, Inc. He has also worked for General Dynamics, helped organize Radiation Research Associates, Inc., and helped organize and became president of Command Systems Corporation. He received the A.A. degree in engineering from Santa Ana College, the B.S. degree in physics and mathematics from Murray State College, and the M.S. degree in physics from Vanderbilt University.

The effervescent years: a retrospective

Behind the Computer Revolution lay the vision and perseverance of a handful of pioneers committed to open information exchange

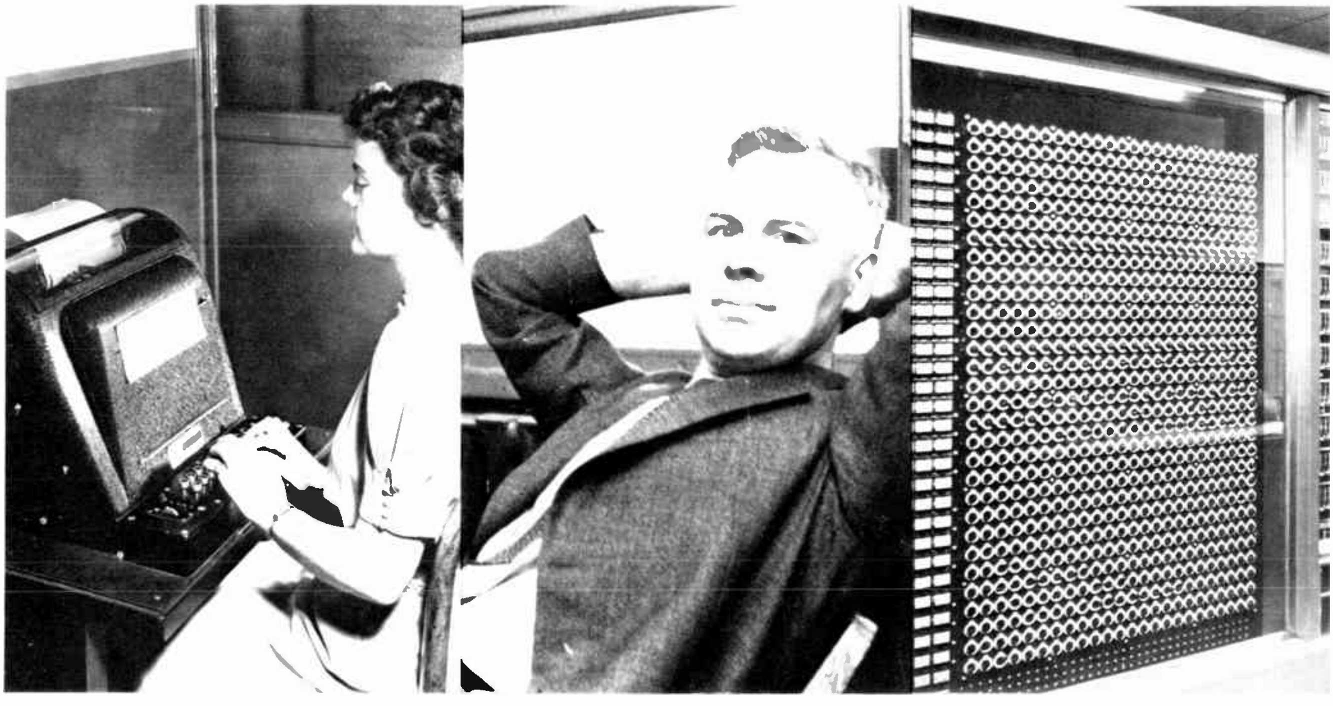
The following alarm was sounded in 1950 by a prominent mathematician: "There is currently such a shortage of trained mathematicians required to operate the modern computing machine that these machines are not working full time. With the number of machines being built or projected it is probable that within ten years, two thousand persons will be required in this work. . . . this is a substantial fraction of all professional mathematicians in the country."[1] Eight years later, a RAND Corporation scientist told a computer conference that "Southern California is today generally regarded as a center of activity in high-speed computing, having perhaps the highest density of machines and active prominent people as any area in the world." In support of this claim, he displayed a map of digital computers installed and operating in Greater Los Angeles. It revealed a grand total of 32 machines!

These appraisals provide a good indication of how small the computer world looked during the 1950s. And yet only a few years were to pass before computers found their way into virtually every large factory and office in the United States, Europe, and Japan. The reasons for this surprise explosion are many and complex, involving a web of technological, architectural, conceptual, economic, and marketing decisions too complex to be tracked in an article such as this. One strand can be unravelled, however; it ties a small group of technical pioneers into the environment for computation that existed between the mid-1930s and the early 1950s, a period which has so far received relatively little attention from historians of science and technology.

Henry Tropp Humboldt State University

The seeds of the computer revolution were sown between 1935 and 1950, by men of great vision whose primary interest lay not in accumulating patents and "shares of the market," but in disseminating their newfound knowledge to all who would listen. As I examine the computational environment between 1935 and 1950, I will be sketching the broad outlines of this process, necessarily omitting much that occurred in an effort to bring out what I consider to be the highlights (see Box, p. 78).

The 1930s and 1940s were decades when the word "computer" generally meant a person rather than a



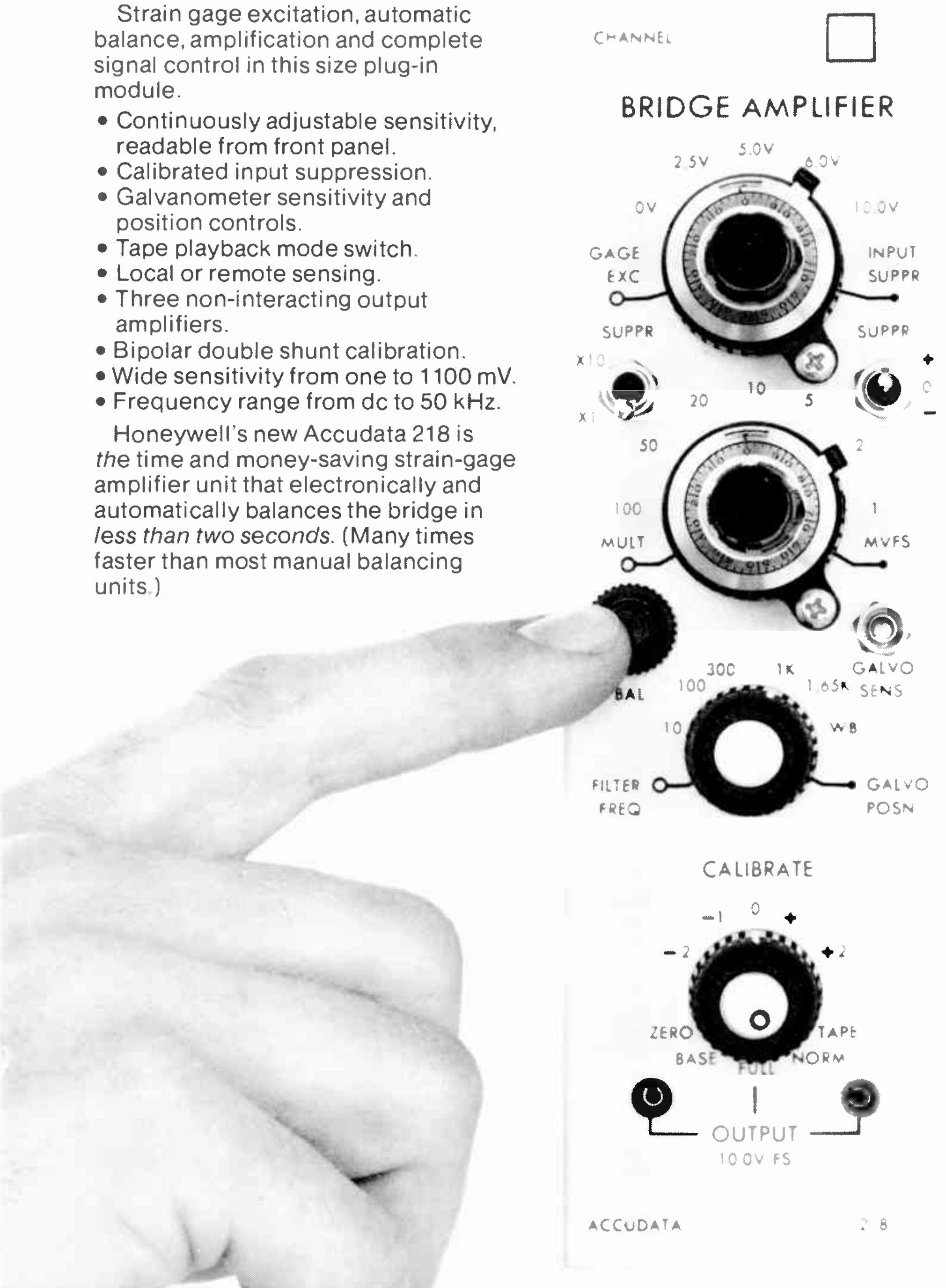
Tic-Push-Toc-It's balanced!

New Honeywell signal conditioner with automatic 2 second bridge balance - saves time - saves money.

Strain gage excitation, automatic balance, amplification and complete signal control in this size plug-in module.

- Continuously adjustable sensitivity, readable from front panel.
- Calibrated input suppression.
- Galvanometer sensitivity and position controls.
- Tape playback mode switch.
- Local or remote sensing.
- Three non-interacting output amplifiers.
- Bipolar double shunt calibration.
- Wide sensitivity from one to 1100 mV.
- Frequency range from dc to 50 kHz.

Honeywell's new Accudata 218 is *the* time and money-saving strain-gage amplifier unit that electronically and automatically balances the bridge in *less than two seconds*. (Many times faster than most manual balancing units.)



Accudata 218—Actual Size

Other time-saving features: direct reading sensitivity for rapid calibration—reduced set up time. Tape playback selector switch eliminates need to change signal input and recalibrate system sensitivity when switching from record to playback.

AND...full scale output switch mode that eliminates external recorder calibration techniques. Internal, calibrated input suppression that eliminates the need for external power supply and hook-up.

PLUS...three non-interacting outputs that provide insurance against loss of data in the event of shorts or overloads.

Below: Two more of Honeywell's fine family of Accudata signal conditioning units: The Accudata 122/123—solid state, wide band, wide gain range differential input dc amplifiers. And the Accudata 117—a direct coupled, single ended floati, wide band amplifier of moderate gain for use as a general purpose signal conditioning instrument.

For a FREE short form catalog and specifications on the Accudata 218, write or call: Lloyd Moyer, Honeywell Test Instruments Division, P.O. Box 5227, Denver, Colorado 80217 (303) 771-4700.

Honeywell
The Automation Company

On Reader Service Card, Circle No. 28 for Information, No. 29 for Demonstration

machine. Three hundred years of well-documented work by mathematical and inventive giants like Pascal, Leibniz, and Babbage had led to a great array of ingenious mechanical devices for tabulation and record-keeping. Furthermore, there was little interest in large scale, systematic computation in industry, even among those who might have been expected to be the most interested—the mathematicians. Such interest as there was existed among a handful of men working quite independently of one another on specific and complex research problems in such disparate fields as astronomy, weather forecasting, space charge physics, and network analysis. It was these men—principally George Stibitz, Howard Aiken, and John Mauchly—who built the first automatic digital computers in order to do their own particular research.

The Stibitz and Bell Labs relay machines

George Stibitz was a young mathematician who had been intrigued with electrical gadgetry since his boyhood in Dayton, Ohio. (He remembers nearly setting his home on fire at the age of eight when he overloaded the living-room circuits running an electric motor his father had brought him.) Stibitz attended what for the World War I era was a very experimental and advanced high school—one that had been established in Dayton by Charles Kettering and some associates from Delco. The school was quite informal and the science program heavily oriented toward individual student projects.

When Stibitz went to college he concentrated on experimental physics and mathematics. He received his M.S. from Union College in 1927, and spent the following year making radio propagation measurements for the General Electric Co. This work was performed in an isolated farmhouse, and he and his partner rigged a voice-actuated electrical communication link that allowed them to operate their equipment by remote control. They also used the link on winter mornings to open the damper on the fire in the coal stove before leaving their home.

After the year with GE, Stibitz went to Cornell for his Ph.D. in mathematical physics, which he received in 1930. That fall, he joined Bell Telephone Laboratories as a research mathematician and became involved with investigations into circuit theory that eventually led him into relay circuit problems. One evening, in 1937, he brought home a couple of relays, some flashlight batteries and bulbs, and wired up a simple binary adder that is referred to in the literature as the Model K (for Kitchen) Computer. This was a significant event because although Stibitz had been introduced to the notion of binary arithmetic in his high school math book, this was the first time we know of that anyone actually successfully adapted the concept to mechanical computation.

While Stibitz's colleagues were amused that relays could be used in this way, his computer didn't seem to arouse any serious interest. Stibitz's main assignment at this time at Bell Labs involved working on switching network problems that required dealing with complex numbers. These computations were so laborious that they had to be done by a group of about ten women using pencil and paper along with the crude mechanical calculators of the day. One day

[1] At the far left is one of the three operator stations for the first Bell Laboratories relay computer, which went into operation in January, 1940. The computer evolved out of the pioneering work of research mathematician George Stibitz, shown in the accompanying photo, which was taken in the early 1940s at Bell Labs.

[2] The first thoroughly automatic computing machine was the Automatic Sequence Controlled Calculator, or Mark I (below), which was built at Harvard between 1939 and 1944 by IBM engineers under the leadership of Howard Aiken, who is shown in the accompanying photo from that period. (Harvard Univ. Computation Lab photos).

a fellow employee suggested to Stibitz that the work might go faster if two or three calculators could somehow be hooked together. Stibitz knew nothing of earlier attempts to do automatic computation (he had never even heard of Babbage), but the suggestion interested him and he proceeded to design a circuit for an electromechanical calculator that could multiply and divide complex numbers. Stibitz's Complex Number Calculator was built by a team of Western Electric engineers headed by Sam Williams, and when it was completed in 1939 (with modifications so it could also add and subtract), it could perform these arithmetical operations three times faster than had heretofore been possible.

But the Complex Number Calculator had some unusual features that made it far more than just another special purpose calculator. First, it had an extremely high degree of reliability—one that would not be exceeded in the electronic automatic computers for many years. Second, because Stibitz felt so strongly about the need to prevent the machine from making arithmetic errors, he built in a checking code so that if a relay failed, it would be impossible to get a wrong answer. This Excess-3 code, as it is now known, also allowed a defective relay to be readily identified.

Another notable feature of Stibitz's machine was that it could be operated from a remote teletypewriter terminal. The demonstration of this to a meeting at Dartmouth College in the summer of 1940 marked the first public display of remote data processing.

It is quite conceivable that Stibitz's contribution to the future of computers could have ended with the construction of the Complex Number Calculator. Shortly after it was finished, he approached the management at Bell Labs with a request for approximately $50 000 to take the next major step and build a large-scale computer, but he was turned down. Management evidently didn't share his feeling for the importance of automatic computation, at least to a telephone company. However, World War II changed all this, because now the need for computation became crucial, particularly in ballistics where research had been underway at the Aberdeen Proving Grounds since the early 1930s. As a result of his work, Stibitz and others at Bell Labs were asked to aid the Army, and a series of five relay computers were eventually built for the prime purpose of doing ballistics and other calculations important to the defense effort.

The Bell program was highly significant, for it demonstrated that relay calculators could perform essentially error-free calculations around the clock, seven days a week, with minimal down time due to malfunction. Thus, although it became apparent that the increase in speed would probably level off to roughly a factor of ten over manual computation, the reliability and ease of maintainability of these computers made people extremely reluctant to move into electronic computers until the 1950s, when new confidence in this technology began to emerge. Moreover, the Bell relay computers had special features which, though designed to meet specific wartime needs, found their way into the mainstream of the postwar computer art. For instance, the 1942 Model II relay computer had a tape program input and also utilized biquinary error detection*; while floating, rather than fixed, decimal computation—which Stibitz conceived in 1940—made its appearance in the Model V computer.

The achievements of George Stibitz have probably not received the public recognition they deserve. This is at least partly due to the fact that he is a rather shy, retiring man who to this day is reluctant to do much public talking about his achievements, with the result that one learns about them mostly from others.

In some respects Stibitz presents a marked contrast to Howard Aiken, another great technical visionary of the preelectronic computer period. While Aiken was not a publicity seeker, neither was he retiring. He was an extremely forceful personality whose perseverance and determination not only led to remarkable technical achievements, but made an indelible impact on all who encountered him.

Aiken and the Mark I computer

Howard Aiken grew up in Indianapolis, Ind., and was forced to go to work after the eighth grade. His first job was as a switchboard operator twelve hours a night, 30 nights a month, with the Indianapolis Light and Heat Company. During the day Aiken attended Arsenal Technical High School, which was another first-class secondary school of its time. The superintendent of schools in Indianapolis took a personal interest in Aiken and arranged a series of examinations so that the boy could graduate early and thereby ease his work-load somewhat. This same superintendent then encouraged Aiken to attend the University of Wisconsin by assisting him to get a similar job with the gas company in Madison. In 1923, Aiken received his B.S. degree and, overnight, a promotion to chief engineer of the gas company.

Aiken had been a power engineer for more than ten years when he decided in 1935 to go back to school and enroll first at the University of Chicago and then Harvard as a graduate student in physics. His doctoral research on the laws of space charge involved him in laborious hand calculations of nonlinear differential equations which, in turn, led him to investigate the possibility of high-speed, automatic calculation.

Harvard in those days was an extremely unlikely place in which to find support for this kind of research project. It was a center of "pure" research, and Aiken was forced to persevere in the face of considerable skepticism and, in fact, very strong opposition. For instance, some people claimed that a machine to do what Aiken proposed would require so many parts that, based upon elementary probability theory, some parts would just never work. (A similar argument was later advanced in opposition to Eckert's and Mauchly's ENIAC project.) Others held that even if Aiken were lucky enough to get his machine working, it would only be a short time before it would have performed all the work it could possibly be required to do, thereby ending up as a museum piece, which was something a university should not produce! Thus, although Aiken received important encouragement from the astronomer Harlow Shapley and Business School Professor Ted Brown, his was essentially a lonely battle, won only by his great determination to reach a very specific and carefully thought out goal.

* A scheme wherein each decimal place is represented by seven relays arranged in groups of two and five.

This goal was to build a thoroughly automatic computing machine controlled by a coded sequence of instructions and capable of producing a punched card or typewritten output. Aiken wrote his proposal for such a machine in 1937 and this document was published in *Spectrum*, August 1964. The reader of the document will be impressed by two things. First, of course, is the tremendous technical vision it reveals. But even more striking is its similarity to the actual machine that was dedicated at Harvard seven years later, in 1944—the Mark I. It is quite clear that what Aiken did was first to analyze carefully the procedure for solving mathematical problems by hand and then to specify a procedure for performing the same operations mechanically.

One of Aiken's first attempts to build his machine was made with the Monroe Calculating Machine Company. He went to Monroe's chief engineer, G. C. Chase, who has described how "Aiken outlined to me the components of a machine which would solve his problems. His plans provided automatic computation and the four rules of arithmetic, preestablished sequence controls, storage and memory of installed values, sequence control which could automatically respond to computer results or symbols, together with a printed record of all that transpires within the machine, and a recording of all the computed results.[3] Chase was convinced Aiken's machine would be important to Monroe's future business and did his best to get his management to agree. But they decided it was too impractical and turned him down. Chase then urged Aiken to approach IBM and suggested that Professor Brown of the Harvard Business School would be a good initial contact. It was through Brown that Aiken met Thomas Watson, Sr., and in 1939 a contract was signed whereby IBM, with financial support from the U.S. Navy, would build the Mark I.

The Mark I is sometimes called a Babbage-type machine, but although Aiken readily admitted to "learning more about computation from Babbage than any other individual," it would not be fair simply to infer, as some have, that Aiken built Babbage's machine. Without trying to detract from the fantastic accomplishments of Charles Babbage, I would point out that he and Aiken differed in a very important way. Babbage evolved a series of technical concepts, first for building a difference engine and later for building the analytical engine, which he conceived in 1833 (this was the first design for a universal automatic calculator). However, in every case, before any significant portion of his machine was physically completed, Babbage would go off into new and better conceptions. One has the image of wandering with this quirky British genius through an infinite series of rooms, each representing some new and yet unfinished level of machine development.

With Aiken, however, his Mark I was conceived at some point in the mid-1930s, articulated very carefully and precisely, and built almost to specifications with, moreover, materials already on hand. There is no technology in the Harvard Mark I that was not available in the 1930s.

The development of the Mark II, III, and IV at Harvard is characterized by much the same methods, with each model designed to take advantage of a specific level of technology. It is clear from my interviews with Dr. Aiken and his colleagues that at some point he would say, "at this stage, new ideas stop and we build. Any new ideas that come afterwards will be saved for the next machine." His first electronic machine, the Mark III, was only built at a time (the late 1940s) when he felt the technology was sufficiently reliable. Many people have interpreted the late arrival of Mark III as a reluctance on Aiken's part to admit that electronics "was the way to go." This was not true. He merely waited until the reliability of the components reached his exacting standards.

Computers in Iowa

The unfriendly environment in which Aiken worked contrasts with the supportive one that existed at Iowa State College in Ames. While all but a few of the most enlightened universities were ignoring computation during the 1930s, Iowa had become an important center for mathematical statistics as a result of the interest of Henry Wallace. Wallace (later President Roosevelt's Secretary of Agriculture as well as Vice President) introduced IBM tabulating equipment at Iowa during the 1920s when he demonstrated its use for statistical analysis in agriculture and weather forecasting.

The following letter, which Wallace wrote to Cuthbert C. Hurd, Feb. 21, 1965, provides a nice feeling for what it was like in those days:

> *In the late winter [February and March] of 1923 I taught a course in machine calculation of correlation coefficients at Iowa State, using a cheap key driven machine. In the concluding session of ten lessons I decided to demonstrate how the IBM machine could be adapted to correlation work. For this purpose I used a truck from the farm to enable me to haul an IBM machine from Des Moines to Ames. I never used an IBM machine as an aid to breeding work but I did try to use it for some time in trying to predict weather. While this work was a flop. I think that Larry Page, the man who conducted it for me had something to do with starting [Jerome] Nemias of the Weather Bureau on his long-range weather forecasting.*
>
> *For purposes of assembling yield test data in form to study, the Pioneer Hi-Bred Corn Co. of Des Moines has long used punched cards and the IBM machine so as to assemble the facts so they may be studied. The company now owns a machine and keeps it busy assembling data on yield test with both corn and chickens.*

An extensive program in computation evolved from this start and, by the mid-1930s, a computer project was begun by John V. Atanasoff, a professor of physics and mathematics. In a letter to the college research council in March 1939, Prof. Atanasoff explained that as early as 1933 he had begun thinking about mechanizing the otherwise "extremely arduous" solution of systems of linear simultaneous equations having many unknowns. Continued Atanasoff:

> *About two years ago I came to a realization that computing machines can be much simplified by changing from the use of numbers to the base 10 to the use of numbers to the base 2. Further study has reinforced this point of view and it now seems possible to build into a small machine of perhaps the size and intricacy of a Monroe a computational ca-*

pacity over twice that of the 8 bank punched card tabulation. This would be, as far as I am aware, the most powerful computing machine in existence and would furnish a direct and satisfactory method of solving simultaneous linear equations.

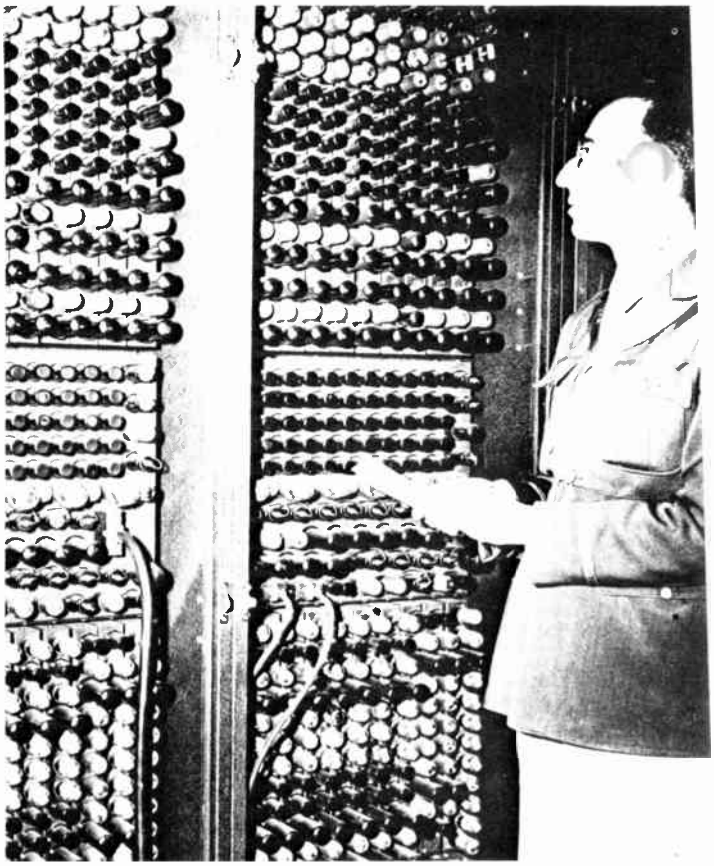
While the basic idea of this machine is original, I have made careful calculations and believe it will work. Therefore, I wish to request a grant of $450 for a research fellow in physics to work on the plan and $200 for materials and mechanical assistance.

Atanasoff's computer was begun but never completed, and during World War II he left Iowa to do research for the Navy. But the project may have had a greater impact than we know, for during the summer of 1941, John Mauchly visited Atanasoff at Ames. Mauchly had met Atanasoff the previous winter at an AAAS meeting in Philadelphia, and the two men discovered their common interest in computation. For several years, Mauchly had been interested in applying statistical methods to the analysis of meteorological data, and, realizing that weather forecasting could only be improved with faster and cheaper methods of computation, had turned his attention to how this might be accomplished.

Mauchly, Eckert, and ENIAC

As a boy, John Mauchly had tinkered with mechanical adding machines and simple electronic circuits, and in 1925 he received a scholarship to The Johns Hopkins University engineering school. He didn't care for engineering though, and after two years switched to physics as a result of interest aroused by his father's work at the Carnegie Institution Department of Terrestrial Magnetism. Under a plan that made it possible to work for a Ph.D. without stopping for a bachelor's or master's degree, Mauchly began immediately on a doctoral program in physics. He was able to get a scholarship and also earned part of his tuition as a junior instructor in the physics department. A brilliant student, he had the distinction of being elected to both Phi Beta Kappa and Sigma Xi in the same year.

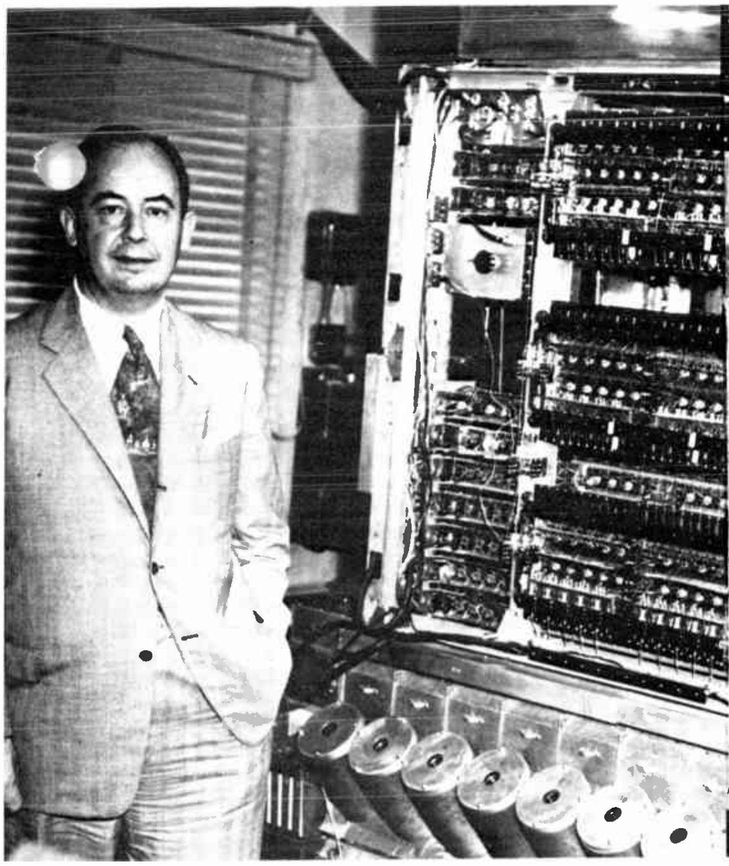
Mauchly received his Ph.D. in 1932, writing his thesis on an analysis of the carbon monoxide molecule. This work required a great deal of computation with an old Marchant desk calculator (called the lima bean computer by people who used it because they had to depress a lima bean shaped handle to move the shift carriage.) Jobs were scarce in 1932, and so Mauchly took the post of research assistant to his thesis advisory, Joseph Eachus, for 50 cents an hour. Their work included calculating the energy levels in the formaldehyde spectrum, and Mauchly suggested that he study the calculation procedures to see whether there were any tricks that could speed the process. Prof. Eachus responded that, while this might be beneficial, he should just go ahead and do it the way he was told. Mauchly decided that, although this might be the best advice from Eachus's point of view, he would try anyhow and he didn't think Eachus should dictate *how* the answers were to be obtained. He decided it would be fair to spend some of his own time, without charging Eachus, to see whether or not he could speed the calculations. Then, if he were successful, he'd recoup that time by adjusting his charges proportionally. The upshot was that

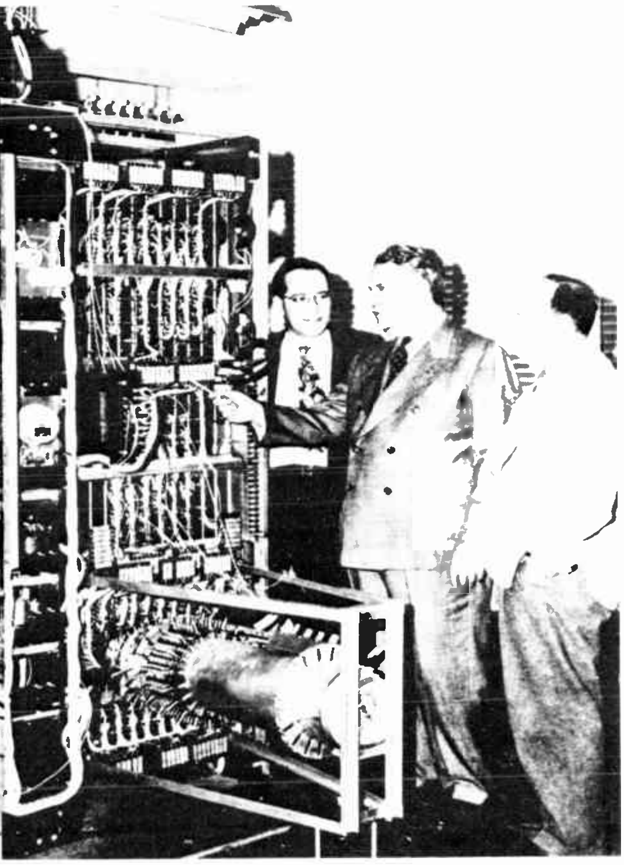


Mauchly discovered he could so simplify the process that the actual labor was cut in half.

In 1933 Mauchly obtained a position as professor of physics and sole member of that department at Ursinus College. While at Ursinus, he came across the publications on using punched cards for computation that had been written by Wallace J. Eckert of Columbia University's computational laboratory. Prof. Eckert was an astronomer whose interest in celestial mechanics led him to do pioneering work on scientific computation by punched card methods (his best known publication on the subject is *Punched Card Methods in Scientific Computation*[4]). When Mauchly read Eckert's papers, he realized how little he understood about statistics and began to study the subject. In 1936 he got a summer job in his father's section at the Carnegie Institution and started applying what he had learned about statistics to the weather data that Carnegie had been gathering. He continued this kind of work with some of his students at Ursinus and in 1940 he presented a paper to the American Physical Society on his search for statistically verifiable patterns in precipitation.

A year later Mauchly attended a summer program in electronics that was set up in Philadelphia by the University of Pennsylvania's Moore School of Electrical Engineering. This was one of a number of such programs sponsored by the U.S. Government to stimulate training in electronics in view of the impending world crisis. While there, Mauchly received an invitation to join the Moore School faculty for the coming year as an instructor. Even though this was a step down in academic rank, Mauchly considered it a step





[3] At the left, Lt. Herman Goldstine is shown looking at some of the 18 000 vacuum tubes that went into the world's first electronic digital computer, ENIAC.

[4] In 1946, the year that ENIAC was completed, Lt. Goldstine went to Princeton to assist John von Neumann in building an important forerunner machine, the Institute for Advanced Studies computer seen in the center photo with von Neumann next to it.

[5] In 1946, ENIAC's creators—John Mauchly and J. Presper Eckert—went on to form their own company and build the UNIVAC I computer which is shown above with, from left to right, Mauchly, Leslie Groves, and Eckert. (Smithsonian Institution photos)

up in academic direction and he promptly accepted the appointment. He also realized he would have better facilities, such as a Bush differential analyzer, with which to pursue his research interests.

Since the mid-1930s the Moore School had held a contract to work on ballistics problems for the U.S. Army at Aberdeen Proving Grounds. This work naturally accelerated when the possibility of U.S. involvement in the European conflict became strong, and Mauchly became involved in it soon after his arrival in Philadelphia. One of his assignments was the supervision of the construction of ballistics tables, and just as Stibitz did at Bell Laboratories, Mauchly laid out a "program" for a group of computers, the computers being women who sat there and performed these operations according to the schedule he laid out. Mauchly soon conceived the idea of reducing this procedure to one that could be done sequentially by other than human means. In this case, the means were electronic, and in 1942 he wrote a proposal for such a device. Mauchly's proposal was rejected, however, apparently because it was the general consensus that the war was going to be short, and that all efforts should be concentrated on making it shorter still. Although projects such as Mauchly's were interesting and exciting, they were not viewed as likely to have any immediate impact on the war and therefore it was generally felt that they should be postponed until the postwar period.

However, in 1943 the environment changed. The pressure for ballistics tables was building at Aberdeen and the computational needs far exceeded the available means of meeting them. A young lieutenant and mathematician, Herman Goldstine, was assigned to find ways of increasing the output of needed tables. When he heard of Mauchly's proposal, his immediate reaction was that it should be resubmitted. However, a thorough search of departmental files failed to produce it. The proposal had apparently vanished. Fortunately, the secretary involved in preparing the original draft had kept her shorthand notes, and from these the original was basically reconstructed. (A few years ago, a copy of the original draft proposal was uncovered and, amazingly, only trivial differences appear when comparing the original with the reconstructed version.)

It was during this period that Mauchly met J. Presper Eckert and one of the famous partnerships of modern technology began. Mauchly was a great technical visionary who happened to know a little bit about engineering, but Eckert was completely the engineer—and a highly gifted and imaginative one at that. He had started teaching at the Moore School soon after receiving his B.S. in 1941. Eckert was an extremely versatile man and he worked on a wide variety of projects. For example, he designed and built a device for measuring the concentration of naphthalene vapor, worked on instruments to measure metal fatigue, and built a device for measuring the strength of very small magnetic fields which was employed in antisubmarine warfare experiments. In addition, he worked on radar problems and on various devices for measuring targets with great accuracy.[5] Eckert was the type of engineer who, when presented with problems, could come up with a variety of solutions. He didn't give a single solution, but conceived a number of possibilities, which were then examined to determine the best one for the particular need. It was this kind of brilliance and flexibility that made Eckert and John Mauchly such an ideal pair to design the ENIAC and supervise its construction.

In 1943, while studying for his master's degree in electrical engineering, Eckert started working with Mauchly on reconstructing the original proposal. On April 9, 1943, while the two men were in another room writing an even more detailed proposal, Goldstine made a formal presentation for funding to officials at Aberdeen, including the mathematician Oswald Veblen and Col. Leslie Simon, director of the Ballistic Research Laboratory. The proposal was accepted and the result was the successful completion in 1946 of ENIAC (Electronic Numerical Integrator and Computer). Eckert was chief engineer and Mauchly consultant on the $400 000 project.

With some 18 000 vacuum tubes, ENIAC could perform 5000 additions or subtractions per second, thereby solving problems 1000 times faster than Aiken's Mark I at Harvard. This meant, as visitors would be proudly told, that a projectile's trajectory could now be computed before the projectile landed!

Despite such achievements, however, it seems clear that without certain other developments, progress would quite likely have halted, for some time anyway, with the completion of the ENIAC, Mark I, and Bell Labs machines. Stibitz, Aiken, Eckert, Mauchly, and a few others possessed great vision, but even in this group there were only a few who really foresaw the potential for high-speed computing in business and industry. Left alone, the computing fraternity would probably have gone on to build a few larger machines strictly for the solution of scientific problems.

Little interest in the executive suite

If one examines the environment at major U.S. corporations during the 1930s and early 1940s, there is a striking lack of interest in any aspect of automatic computing. I have yet to see a publication from that period that reveals an interest in putting research and development funds into finding better ways of doing automatic computation. At Bell Laboratories, for instance, Stibitz's Model I Calculator might well have been the last in the series if it hadn't been for the pressure of wartime needs and the accompanying government support.

IBM was similarly uninterested in advancing the computer art. In those days the company didn't even have a formal engineering or research department. IBM engineers and inventors were attached to the patent department and for special projects, they reported directly to Thomas Watson, Sr. Although many IBM people did significant work, particularly on the Harvard Mark I, the company seems to have been primarily interested in improving existing products and building a strong patent position for possible future applications. I have been able to discern no evidence of a vision at IBM—or at any other corporation—in those days that was comparable to that held by Aiken, Mauchly, Stibitz, Atanasoff, or J. P. Eckert or Wallace Eckert.

Why then did IBM support Aiken? I believe the answer to that lies in the great personal interest Thomas Watson, Sr., had for supporting educational institutions. In the early 1930s, for instance, Ben Wood, a Columbia University professor, was doing statistical analysis of test results and Watson supplied him with the necessary IBM equipment. This work, which later led to the machine scoring of examinations, served to introduce Wallace Eckert to the possibility of using IBM equipment for his astronomical calculations. By 1934 Eckert, with Watson's help, had established one of the few university computation laboratories in the U.S., later the Watson Laboratory. Eckert used this equipment to produce lunar ephemeri that have remained valid enough to be used for Apollo.

I am convinced that similar motives lay behind Watson's support of Aiken; it was Watson, the altruist, wishing to help Harvard, rather than Watson, the head of IBM, seeing a way to increase corporate profits. After all the Mark I was never supposed to do work for IBM, nor did it as far as I know. IBM had no work set up for it and certainly never saw it as a first step by the company into the computer business. That did not come until a decade later.

The National Cash Register Company offers a particularly intriguing industrial "might-have-been." NCR actually had an electronic computing device constructed during the late 1930s. It was a high-speed arithmetic machine which could add, subtract, and multiply electronically, and presumably, this machine could have become the first commercial electronic computer had the company wished to pioneer in this field. However, NCR management was not interested in automatic computing per se, but only in improving its existing line of office equipment. As a result, it directed its research efforts to such activities as designing and building a new line of small thyratron tubes. We have no way of judging just how influential this work was for computer technology generally, for the bulk of it was carried on during World War II for an application in cryptanalysis and still remains classified.

There were, of course, a few instances of business interest in computers during the late 1940s. A British company, the Lyons Catering Company built the Leo computer to schedule the flow of products among its various shops.[6] In 1946. Stibitz himself designed a desk-size electronic computer for the Barber-Coleman Company, but it proved to be 15 years ahead of its time and was never produced commercially. All in all, however, there was relatively little commercial activity or interest in the computer during the 1940s.

The push from new problems and technology

Three factors changed all this and proved a catalytic bridge between the great conceptual breakthroughs of Aiken, Stibitz, Mauchly, and Eckert, and the computer explosion of the 1960s. The need to solve a new class of scientific problems arose, along with the emergence of the necessary new technologies. In addition, there was a relatively swift and effective dissemination of information about these new technologies and computing in general.

During the 1920s and 1930s scientists and applied mathematicians were concerned largely with solving ordinary and partial differential equations. Important progress was made in solving ordinary differential equations through the development of differential analyzers by Vannevar Bush and his colleagues at M.I.T. As for partial differential equations, satisfactory solutions could be found with the classical pencil-and-paper techniques of the 19th century until the development of quantum mechanics and, especially, the atomic bomb project, yielded problems in turbulence, implosion theory, particles at high speeds and so on. These problems could only be solved with digital computers, and computers, moreover, that were much faster and had much greater storage capacity than a machine like ENIAC or the Harvard Mark I. The pressure to solve these new kinds of problems was the impetus to embark on a new generation of machines. And in contrast to the first machines of the pioneers, these machines could not be constructed out of essentially off-the-shelf components. They would require entirely new technologies, particularly those that arose from wartime advances in solid-state physics, communications, and control systems.

John von Neumann, the Hungarian-born mathematician, was a man whose work in statistics, shock waves, ballistics, detonation problems, hydrodynamics, aerodynamics, meteorology, and the applicability of mathematics to game theory and to computer design made him one of the major intellectual figures of the twentieth century.[7] He had worked at the Institute for Advanced Study in Princeton during the late 1930s with Alonzo Church, the famed logician, and Alan Turing. Turing's 1936 paper, "On Computable Numbers . . .,"[8] is one of the milestones in computer development because it showed that a machine was capable of solving any problem that a human being following explicit instructions could solve. Turing's conception involved the idea that one could take any function and break it down into simple repetitive operations. He gave a specific definition of the notion of an algorithm and its meaning. He then proposed a machine with a finite number of states or conditions (essentially its alphabet), which has the ability to look at an infinitely long tape on which these symbols would appear in consecutive cells. The machine would then have the ability to read a symbol, change it to any other symbol that it had in its finite list, and enter a new state or condition. Turing then conjectured that any problem which one could think of solving could be placed into this format.

Although we don't know the precise nature of the discussions between von Neumann and Turing, we do know that von Neumann was greatly influenced by Turing and his paper. In later years, when people came to von Neumann for advice and information, he advised them that the Turing paper contained the key concepts and anyone interested in computers should start with it.

During World War II, von Neumann was a consultant to the atomic bomb project at Los Alamos as well as to the Aberdeen Proving Grounds. As a result, he recognized the need for machine computation of shock-wave problems and he also became familiar with the work then underway on ENIAC. In 1946, he, Herman Goldstine, and Arthur W. Burks wrote a landmark report, "Preliminary discussion of the logical design of an electronic computing instrument," which paved the way for the first computers in which the program could be stored and modified electronically, without the hours of manual rewiring required in the ENIAC. The first internally stored program computer to run was designed and built at Cambridge University by computer scientist Maurice Wilkes. The EDVAC, which Eckert and Mauchly initiated with von Neumann in 1946, became operational in 1950.

The Aberdeen group split up in 1946, with Eckert and Mauchly forming their own company to produce the BINAC (for Northrop Aircraft Company) and the UNIVAC commercial computers, and von Neumann, with Goldstine as assistant project director, returning to Princeton to supervise the construction of an experimental computer known as the IAS (Institute for Advanced Study) Computer. Princeton was as unlikely a place as Harvard in which to find support for building machines, yet von Neumann's reputation was so great that the prevailing attitude was "If Johnny wants it, let him do it." The IAS computer incorporated such advances as parallel processing, and possessed what was for that time the extremely large memory of 1024 words. (Von Neumann is reported to have felt that for all practical purposes, this gave him an infinite memory.) The IAS was the forerunner, both philosophically and architecturally, of a whole series of machines, including ILLIAC, at the University of Illinois; AVIDAC and ORDVAC, constructed at the Argonne National Laboratories; JOHNNIAC, at the RAND Corporation; and MANIAC, at Los Alamos. Today's generation of computers is essentially of the von Neumann type.

Meanwhile, other important technological advances were being made between 1946 and 1951. Eckert was turning to radar technology and applying the techniques of mercury delay line storage to the EDVAC and UNIVAC I machines; this technology was successfully demonstrated by Wilkes in the Cambridge University EDSAC; the UNIVAC group was also pioneering the successful use of magnetic tape; Aiken was using magnetic drum storage and building his electronic Mark III and Mark IV machines; at the University of Manchester, in England, F. C. Williams was building a machine with magnetic drum storage and making an important breakthrough by developing an electrostatic storage tube; and at the British National Physical Laboratory, Turing was designing the logic for a machine that was too ambitious. Under the instigation of Harry Huskey, the project was scaled down and when completed, became the computer now known as Pilot Ace.

One of the most significant developments of this period occurred at M.I.T. with the construction of the Whirlwind computer. This was the first machine developed for real-time applications (air defense, in this case) and it drew heavily on communications and control system technology. Among its many innovations was the coincident-current magnetic core memory. Developed by Jay Forrester, at M.I.T.'s Digital Computer Laboratory, the coincident core memory turned what had been the least reliable element of the computer into one of the most reliable elements.

In this manner each of several new machines came to embody some particular technical advance which, taken together, proved that the electronic computer could be more than a temperamental mechanical genius that only ran on occasion. With the delivery of UNIVAC I in 1951 to the Bureau of Census, computers ceased being one-of-a-kind machines. Of course, size, reliability, and cost would remain handicaps for some time. Nobody ever wanted to have to build a machine with 18 000 tubes again! But as the coincident core memory was followed first by the transistor and then by the technology of integrated circuits, these gradually ceased to be obstacles. Further, the technology evolved to the point where computers were cheap enough and reliable enough that business and industry clamored for them.

Free and open exchange of ideas

Just how did this acceptance come about? What led to the surprising explosion in installations that occurred during the 1960s? While there were obviously many factors, I am convinced that the considerable effort made by the computer pioneers and their students to disseminate information was critical. This effort began with a 1946 summer conference at the Moore School. Approximately two dozen people were invited to hear pioneers like Eckert, Mauchly, von Neumann, and Goldstine explain the first electronic computer as well as the conception for the next stage in computer evolution—the EDVAC stored program computer. The Cambridge University machine, EDSAC, was a direct outgrowth of this conference, for Maurice Wilkes was one of the attendees and he decided EDVAC was a bit too ambitious and that he would build a scaled-down version.

Subsequently, Howard Aiken arranged two major conferences—one in 1947 and one in 1949—at Harvard, in which he invited everybody he could get his hands on who had contributed anything to computation to present a paper on a specified subject. A similar conference was held at Cambridge, England, in 1949, while between 1948 and 1951 IBM sponsored five symposia on computing. By collecting the small core of people who were involved with computers and exposing them to an equally small core of people who were interested, these conferences played a vital role in advancing the state of the art.

Formation of the Eastern Computing Society (now the Association for Computing Machinery) in 1947 was another important event. The Society had no intention of publishing journals; rather its aim was to bring key people together and spread the gospel by personal contact. Its meetings became forums for discussion of the latest machine technology, problem-solving methods, and programming concepts.

On the West Coast, there was the Digital Computer Association, a most informal and occasionally riotous group which used to meet monthly in Los Angeles. This group—a forerunner of an organization now known as SHARE—was made up of IBM equipment users who were interested in cooperating with one another in an attempt to stop duplication of efforts and to share new discoveries.

In this way, as people traveled around the country exchanging information and learning especially about the implications of the transistor technology that was then emerging, an awareness grew that computers could be used in a practical way for all sorts of applications besides those of scientific research.

Indeed IBM's decision to launch its 701 stemmed from this awareness. The 701 was the first production model scientific computer and its acceptance resulted in the production of a computer for commercial uses, the 702. The 701 design began in 1950, with the outbreak of the Korean War, when the Government asked IBM to build a scientific computer. The company discovered, however, that it could pick up a number of industrial orders (18!) for computers that would aid in such projects as designing jet engines. Thomas Watson, Jr., (then IBM's president) recalls being convinced "that we were in the electronics business and that we'd better move pretty fast."[9] IBM's

Source of the tale

The material constituting this article on digital computer history is based on research done during my tenure as principal investigator for the Computer History Project jointly sponsored by the Smithsonian Institution and the American Federation of Information Processing Societies. This research is ongoing, and the essential data base is not yet complete.

A large portion of the account presented here is based on personal interviews with the individuals discussed and their close associates. Clearly, many people have been omitted who might have been included—the file of the oral history portion of the project contains interviews with approximately 250 individuals!—but such omissions were made only in the interest of limiting the discussion.

Thus, for example, Vannevar Bush and his work in analog computation couldn't be included in an article limited to a discussion of digital developments. Similarly, the work of Comrie and others in England and Europe, the developments of the West Coast computer industry, and the contributions at the National Bureau of Standards (in particular, its association with the depression-spawned Mathematical Tables Project in the area of computation) are other important milestones deemed beyond the article's scope though worthy of mention. Lastly, it is important to identify the role of Harry Huskey and Sam Alexander in developing the National Bureau of Standards' SEAC and SWAC computers and to cite the work in electronics of people like Halsey Dickinson and Ralph Palmer (IBM) and Joseph Desch (NCR). But, again, space limitations prevented their inclusion.

All these people, and many more, however, are represented in the extensive collection of tapes and related documentation available to interested researchers at: Computer History Project, Smithsonian National Museum of History and Technology—Room 4601, Wash., D.C. 20560—*H.T.*

willingness to gamble on what was then volume production of computers played a major role in making people aware there was a technological revolution on hand.

Equally important was the growth of university programs that by the 1960s emerged as the academic discipline of computer science. One of the seminal courses was Aiken's Organization of Large-Scale Computing Machinery, which he introduced in the school year 1946–47 when he had just returned to Harvard from his Navy duty. This eventually evolved into a course on switching theory and produced quite a few important doctoral dissertations.

During the ensuing decade there was a growing interest in computation, coupled with the need to train people in the field, at universities like Michigan, Stanford, Purdue, Illinois, U.C.L.A., M.I.T., Cambridge, and Manchester.

It seems fair to say that the absolutely free and open exchange of ideas played a critical role in the explosive growth of computers. Aiken, for instance, was not particularly interested in what others were doing except insofar as he could draw upon their technology for his machines. But he was incredibly interested in disseminating everything he knew. Men like Aiken and Mauchly were anxious to talk with anyone who would listen, and their projects were major training grounds for many people who went on to contribute significantly to this new industry. These included people like Frederick Brooks, Kenneth Iverson, and Grace Hopper. Hopper was involved in the Mark I project and then played a key role in the UNIVAC project where she was personally responsible for the early work on compilers. After she joined the Eckert-Mauchly Corp., she frequently accompanied Mauchly in the presentations he was making to the Bureau of Standards, the Census Bureau, Lockheed Aircraft, and other interested organizations.

The visitor's log book at Harvard during the Mark I era reads like a *who's who* in computation and technology. Stibitz visited Harvard as did Mauchly, Jay Forrester, Louis Ridenour, and John von Neumann. Maurice Wilkes and L. J. Comrie visited from England, Couffignal came from France, van der Pohl came from Holland, and Swoboda came from Czechoslovakia. There were even members of a Russian delegation who visited before the Cold War era.

Aiken traveled a great deal during this period as well. He made many trips to Europe, visiting and giving symposia throughout the continent. I doubt that there was a nascent computer project in all of Europe that he didn't influence in some way.

A good feeling for the environment at Harvard is provided by the following reminiscence of Kenneth Iverson at last spring's memorial service for Aiken:

> *More important than Dr. Aiken's formal courses were the community he established in his Computation Laboratory and the welcome he extended to graduate students—each was given a desk or space of some kind and soon became part of a lively and entertaining crowd of mathematicians, designers, and technicians. It was a community in which pride of work was pervasive, extending even beyond the technical corps, and one in which everyone seemed glad to take time to discuss his work. It was also a community in which each student was treated as an individual, and our views of Dr. Aiken's tutelage may therefore differ widely, but I believe that we all agree on the significance of his influence on our professional careers.*

As I have indicated, technical societies on both coasts provided a forum where actual problems were freely and openly discussed. Everybody wanted to show everybody else some new thing he had been able to do with his particular machine. People couldn't wait to share a new program, a new way of solving a problem, or ingenious procedures and short-cuts.

I believe it is serious food for thought that such openness is no longer the mark of the computer fraternity. Howard Aiken never worried about patentability, but with the growth of commercial involvement this became a serious matter. I remember the days when someone would lecture at a university on a computer program he had just written and when you asked him about it afterwards, he would take your address and mail you a copy. But today the programs are as proprietary as the hardware patents. There is no question that we have moved into a more closed environment than existed when the seeds for the computer revolution were sown. As to what this bodes for the prospects of the future evolution in computers, I leave to *Spectrum's* readers to ponder.

REFERENCES

1. Clippinger, R., "Mathematical requirements for the personnel of a computing laboratory," *American Mathematical Society*, vol. 57, p. 439, 1950.

2. Gruenberger, F. J., "A short history of digital computing in Southern California," *Computing News*, pp. 145-22—145-30, 1958.

3. Chase, G. C., "History of mechanical computing machinery," *Proc. of ACM*, Pittsburgh, Pa., pp. 1-28, May 1952.

4. Eckert, W. J., *Punched Card Methods in Scientific Computation*. New York: Thomas J. Watson Astronomical Computing Bureau, Columbia University, 1940.

5. Rosenberg, J., *The Computer Prophets*. New York: MacMillan, 1969.

6. Simmons, J. R. M., *LEO and the Managers*. London: MacDonald, 1962.

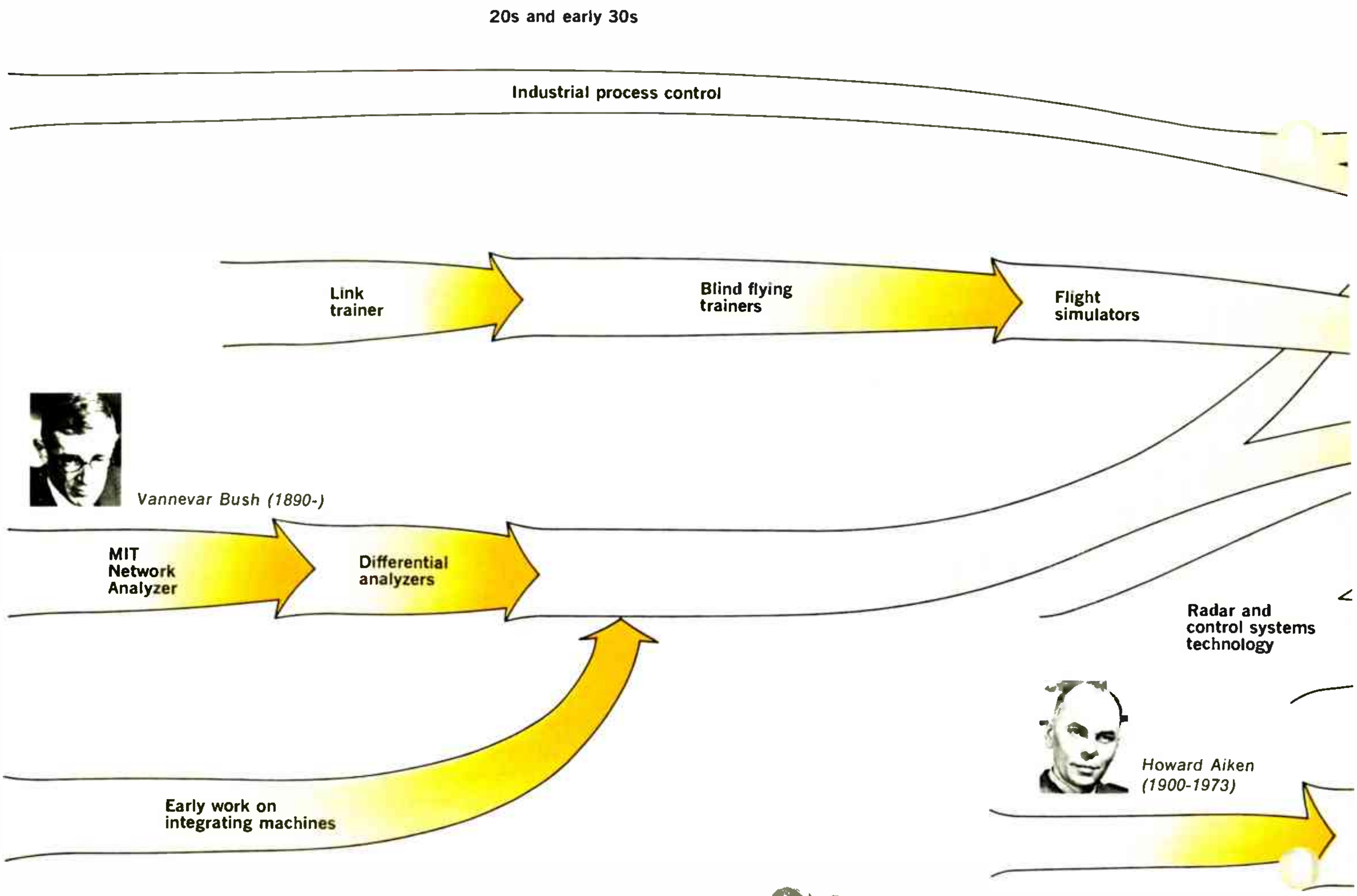
7. Goldstine, H. H., *The Computer from Pascal to von Neumann*. Princeton, N.J.: Princeton University Press, 1972. This is an important work which gives heavy emphasis to the role von Neumann played directly in the logical design and programming of the early digital computers, as well as to a variety of areas of applications including automata theory, hydrodynamics, numerical meteorology and numerical analysis. Also, *Collected Works of John von Neumann*, Abraham Taub, ed.; Halmos, Paul, "The Legend of John von Neumann," *American Mathematical Monthly*, p. 382, April 1973.

8. Turing, A. M., "On computable numbers, with an application to the Entscheidungsproblem," *Proc Lond Math Soc* (2), 42 230.

9. *IBM THINK*, April 1973.

Henry Tropp is a mathematics professor with a strong interest in the history of mathematics and of computers. He teaches at Humboldt State University, Arcata, Calif., which he originally joined in 1957. From 1971 to 1973, Mr. Tropp was the principal investigator of the Computer History Project at the Smithsonian Institution's National Museum of History and Technology. This research concentrated on documenting the evolution of the electronic computer from the decade prior to the dedication of ENIAC through the first generation of commercial computers.

Mr. Tropp has his B.S. degree in mathematics from Purdue University and his M.S. from Indiana University. He has also done graduate work at the Universities of Michigan, Washington, and, most recently, at the University of Toronto where he is completing his Ph.D. in mathematics.



Early computer developments

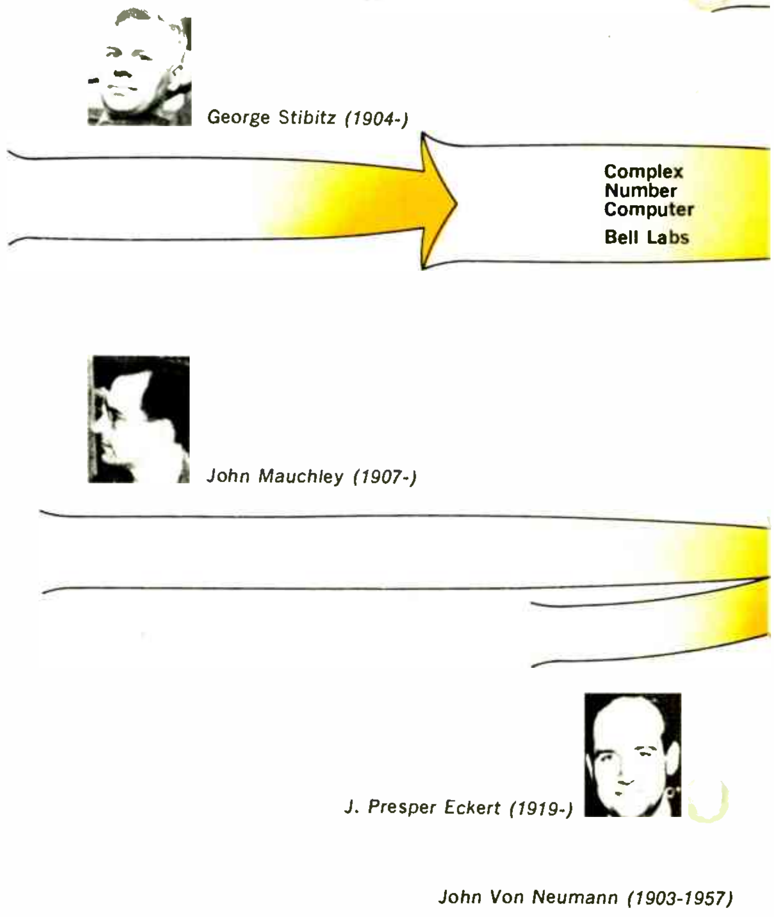
Events on this chart depict the story of early computer developments—in the United States—following closely the story told in the article, "The emergence of the digital computer" by Henry Tropp, in this issue.

Names of many of the significant contributors and builders of these machines are shown, and some place-names are used to further identify the developments.

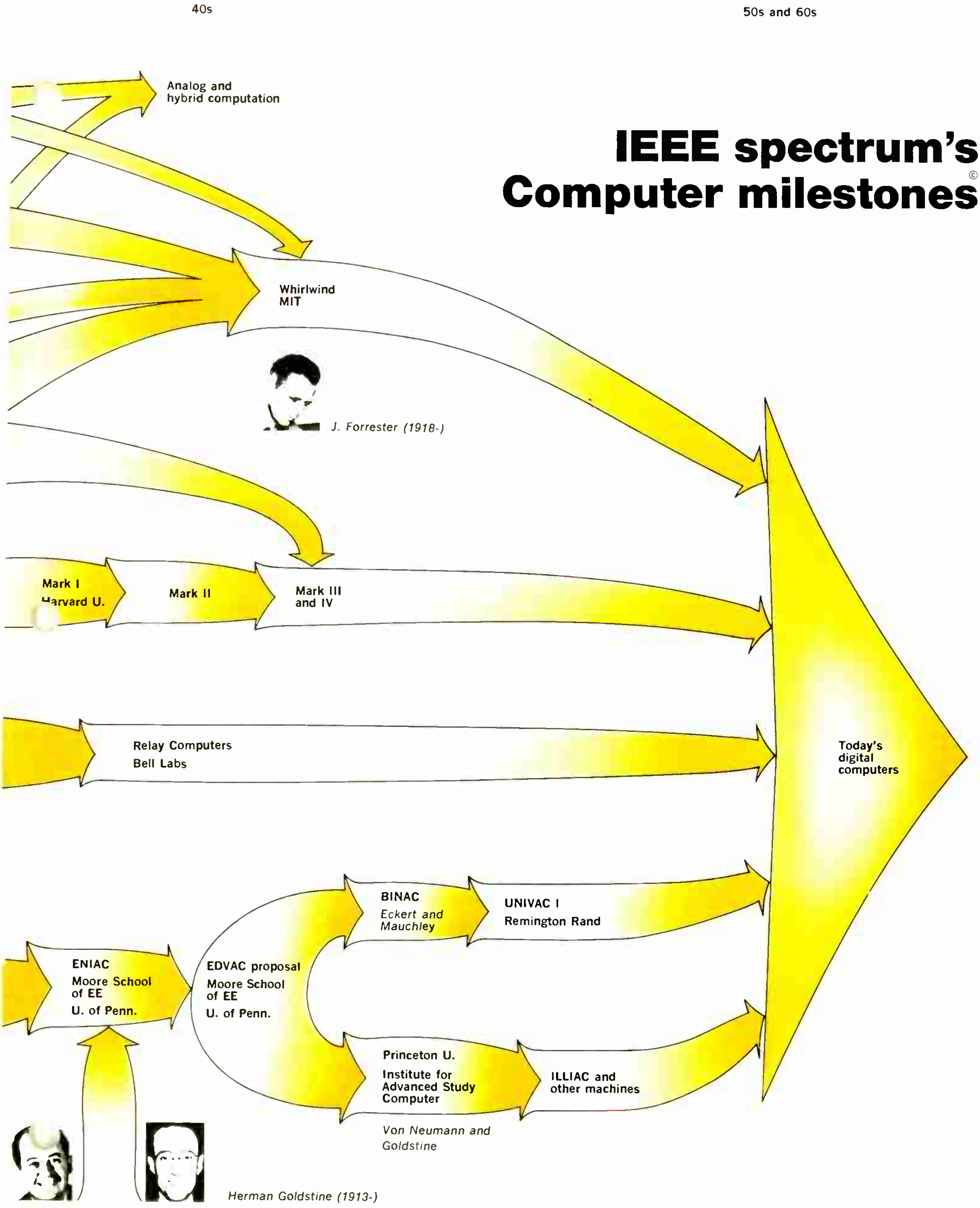
The chart is by no means intended to give a complete picture of early developments—many machines of the period are not shown at all. In fact, a complete presentation would be too complex for publication here.

A number of computer people who participated in the developments of the period were asked to comment on the chart. Some pointed out that the chart ignores important work outside the U.S., such as the fact that Maurice Wilkes had the EDSAC machine operating in England before any stored-program machine in the U.S. Names not mentioned on the chart were cited as very important to early computer development. These included: Harry Huskey, who worked on ENIAC, EDVAC, SWAK, and—with Sam Alexander—on the SEAC machine; Louis Fein and the RAYDAC machine; Julian Bigelow, who worked with Von Neumann at Princeton; and a number of early machines at IBM like the Electronic Multiplier, the SSEC, and the IBM 603.

Prepared by the *IEEE Spectrum* staff



IEEE spectrum's Computer milestones ©



Computer report VIII

An $88 book shelf

Eleven volumes, selected from hundreds available, form this digital engineers' reference and study library

The digital engineer today faces a number of new design challenges. In addition to the combinational and sequential logic circuits he may have been designing in past years, he must now deal with the design of small special-purpose computers based on large-scale integrated circuits. He will need to be familiar with programming techniques to some degree, since many of his circuits will use stored programs, possibly in read-only semiconductor memory storage. More than ever before, he needs to know the functional and operational differences between various forms of logic and memory circuits (TTL, MOS, CMOS, beam-lead, etc.), and how they can be powered and interfaced. He needs access to techniques for coding and error detection methods, as well as to other techniques for reliability and data security. He needs to know the "tricks" used in organization of large computers, for ultimately they seem to become important even in very small digital systems.

W. R. Beam Consultant

After weighing many considerations (see Box, p. 84) and working up several different checklists, an $88 reference "library" has been selected for the digital engineer. Kohonen's *Digital Circuits and Devices* heads the list, because it offers the most basic coverage. In Part 1, it treats the design of both combinational and sequential circuits, including the more important optimization procedures. It then moves to arithmetic circuits and more complex sequencing or control circuits. Karnaugh maps and state diagrams are used liberally and Kohonen's examples are applicable to computers or to other types of digital circuits. In Part 2, the discussion shifts to circuit considerations: how the various common semiconductor circuit types operate, what factors limit their speed and operating voltages, and how briefly the devices are monolithically formed in various silicon fabrications. A variety of timing circuits is described, in-

small computer handbook 1973
SEMICONDUCTOR MEMORIES
Digital Circuits and Devices
The Design of Digital Systems
John B. Peatman
VOLUME 1
BREUER
Theory and Design of Digital Computers
Lewin
MCS-8 MICRO COMPUTER SET • 8008 USERS MANUAL
intel
IEEE PRESS
A Practical Guide to Minicomputer Applications
Fred F. Coury
Introduction to data communication
Introduction to programming
Microprogramming Handbook

cluding linear time-base generators. The treatment of memory and storage is brief, but highly informative. The final chapter deals with transmission of digital signals, a circuit topic little appreciated by many digital engineers.

Peatman's *The Design of Digital Systems* was selected because of its wealth of attention to *noncomputer* applications of digital circuits. Although he also includes a treatment of combinational and sequential logic circuits, Peatman gives special attention to system timing problems, as well as circuits (such as shift registers) commonly used in noncomputer digital systems. Chapter 6, dealing with the organization of digital systems, explains methods of deriving circuitry when only system functions are specified. Chapter 7 deals with input and output on an actuator–transducer basis, with examples typical of process control or data acquisition systems. Chapter 8 gives examples of "algorithmic processes" particularly pertinent to instrumentation uses of digital systems. An excellent set of appendixes considers a mix of special concerns ranging from circuit-family compatibility, through packaging, to fault detection. Throughout the book, plentiful problems aid self-study because each is clearly keyed to the topic that it is meant to illustrate.

Lewin's *Theory and Design of Digital Computers* is the only true "computer book" among the hard-cover books listed. It was selected because it deals with a greater number of the important computer concepts, in more concise and engineer-oriented form, than any other single book examined. Even such topics as microprogramming and virtual storage are ably and concisely handled. It's also the closest to a pure reference book of any in the group selected—topics are boldface where defined in the text material, the many numbered references are clearly tied to specific points in the text, and (wonder of wonders!) a large fraction of the exemplary problems are provided with detailed solutions in the rear of the book. Only one slight caution to the reader: a minor part of the nomenclature is distinctively British ("highway" instead of "bus," and "bistables" for "flip-flops"). But most nomenclature is universal, as are the illustration symbols. I won't attempt to list the topics, but will simply state that the book is unusually complete in its coverage of computer hardware topics.

As for the more specialized selections, one would expect that the variety of choice here would be immense. However, the law of supply and demand rules, and choices are really quite limited. In the hard-cover area, we have selected only two topics which, although in some sense specialized, should be high on the modern digital engineer's list of concerns. The first is treated in *Semiconductor Memories*, edited by Jerry Eimbinder. With memories playing an ever more important role as hard-wired logic is replaced by memory-controlled programmable logic, the choice of memory technology and operating modes is a critical part of systems design. But the memory field is fast-moving. We chose this particular book from a surprisingly large field of three because it best describes the "guts" and operating characteristics of most of the now-known types of semiconductor memory technologies. Editor Eimbinder has solicited knowledgeable people in various technologies, and they have included extensive applications data, in addition to the basics. Further, the book is a collection, uniformly edited with many illustrations, of written-to-order essays, each complete within itself and no more than 15 pages long. Happily, commercialism, often a problem in a field with many proprietary developments, is largely absent.

The next book chosen was selected because every digital designer needs to know *something* about design automation, even if he's only going to build a single system. The nature of semiconductor packaging and the need for partitioning circuits on to cards, boards, frames, etc., demand an orderliness that cannot be accidental. Large firms spend millions on DA software, which most firms cannot afford. In editor Breuer's *Design Automation of Digital Systems*, one can find both the fundamentals and some deeper analysis. Even the less mathematically inclined will find valuable guidance in organizing circuit design, layout, and testing. Ralph Preiss' introductory chapter provides a design-process checklist for digital systems. Chapters 2 and 3 deal with logic synthesis methods that are more advanced, and applicable to larger systems, than those in the other books selected. Chapter 4 deals with selecting larger (e.g., card-level) module functions, Chapter 5 discusses placement and interconnection of modules, and Chapter 6 considers interconnection routings. The final chapter treats fault-testing, a universal problem of the builders of complex digital systems. While many of the techniques are most useful in relatively large systems

An $88 reference and study bookshelf for the digital engineer:

Kohonen, T., *Digital Circuits and Devices.* Englewood Cliffs, N.J.: Prentice-Hall, 1972, 466 pp., $16.50.

Peatman, J. B., *The Design of Digital Systems.* New York: McGraw-Hill, 1972, 457 pp., $15.50.

Lewin, D., *Theory and Design of Digital Computers.* New York: Wiley, 1972, 383 pp., $15.75.

Eimbinder, J., *Semiconductor Memories.* New York: Wiley, 1971, 214 pp., $12.75.

Breuer, M. A., *Design Automation of Digital Systems*, vol. 1, "Theory and Techniques." Englewood Cliffs, N.J.: Prentice-Hall, 1972, 420 pp., $15.00.

Coury, F. F., *A Practical Guide to Minicomputer Applications.* New York: IEEE Press, 1972, 211 pp., IEEE Member Edition (paperbound) $5.00. Clothbound Edition, $9.95 ($7.50 to members).

Introduction to Programming. Maynard, Mass.: Digital Equipment Corp., 1973, (approx.) 400 pp., $2.00.

Small Computer Handbook. Maynard, Mass.: Digital Equipment Corp., 1973, 404 pp., $1.00.

Murphy, D. E., and Kallis, S. A., *Introduction to Data Communication.* Maynard, Mass.: Digital Equipment Corp., 1971, 78 pp., $1.00.

Microprogramming Handbook (second edition). Santa Ana, Calif.: Microdata Corp., 1972, 433 pp., $2.50.

MCS-8 Micro Computer Set User Manual. Santa Ana, Calif.: Intel Corp., 1973, 126 pp., $1.00.

How the books were selected

This is *not* a book review.

From more than 40 recently published books, 11 have been selected to constitute a small reference and study library for the digital engineer. What characteristics did we look for? We wanted a set of books covering the fundamentals with only a small overlap in content. We considered each book's *reference* value. And we monitored the cost-effectiveness of each selection. No doubt, your personal reference shelf will contain high-cost specialized items representing your particular areas of interest, but we leave such additions in your own hands.

such as computers, the principles apply very broadly.

Coury's *Practical Guide to Minicomputer Applications* is a book chosen with the following rationale in mind: Much of the work of a present-day digital system engineer may involve applying minicomputers to data manipulation, process control, instrumentation, etc. The best approach to learning minicomputer application is through example, and Coury's book includes a wide range of examples. Equally valuable are the sections on peripheral and software consideration, and on minicomputer selection. And because one of the principal differences between building hardware and using a minicomputer is that the digital engineer must produce or procure *programming*, several papers in this book deal with software systems used in different types of applications. (This, by the way, is one of IEEE's low-cost large-format reprint books, with carefully selected reprints pulled from a wide range of IEEE publications as well as from non-IEEE journals.)

Our remaining book selections are products of manufacturers. Happily, some of the topics which can't be (or simply are not) treated adequately in publishers' products are available from the makers of minicomputers or semiconductor components. Some of this literature may be freely distributed at conferences, and the cover prices of a few dollars, at most, make some of these publications real bargains. The items that we have included in this reference collection are somewhat unusual in having both a substantial tutorial value as well as reference usefulness. Our choice does not imply a product recommendation, in any case.

From Digital Equipment Corporation, which may have been the originator of minicomputer paperbacks, we've selected three books of strong tutorial value. *Introduction to Programming*, though dealing with programming of the 12-bit PDP/8 machines, illustrates a great many of the techniques needed in operating a minicomputer. It illustrates the sort of manufacturer-supplied and user-written portions of software typical of minicomputer usage in engineering applications. (The PDP/8's language is so simple that it doesn't interfere with the tutorial process, whereas more representative and more modern minis have languages and instruction sets which tend to dominate manufacturers' programming manuals.)

Small Computer Handbook is a complementary guide to the PDP/8 hardware, written by engineers for engineers. It is representative, in its description of peripherals and input-output interfacing, of many minicomputers. The third DEC book, *Introduction to Data Communications*, is really a primer on the topic, and as much background as the majority of digital engineers may need. Full-size books in this area tend to be either too mathematical or too expensive. All that this one will do is give you the jargon and principles of remote data transmission. If you want to know about modems, or organization of data networks, you'll need more specialized books.

Our next selection illustrates, through rather specific examples, the use of microprogramming (in this case, in a minicomputer) to achieve special hardware results. Microdata Corporation's *Microprogramming Handbook* is three items in one: an excellent computer glossary, a detailed lesson in the use of microprogramming to construct logic of your choice, and the description of a general-purpose minicomputer developed through such microprogramming. Although some of the large commercial computers have more sophisticated microprogram facilities, they are very specific and do not convey the concepts as well as in a simpler machine.

Our final selection represents a field that is so new, so exciting, and so important that only a semiconductor manufacturer can write a book on the subject: microprocessors. Intel's *Users Manual* for the 8008 microcomputer (". . . on a chip") contains everything from electrical data to descriptions of the programming language by which (using a larger computer!) one can write programs for the 8008. With microprocessors, the digital engineer reaches the ultimate in complexity-at-low-cost. Such manuals as this one are the only practical way to find out much about these highly specialized but *very* important components. Although, eventually, there may perhaps be a modicum of standardization in microprocessor instruction sets, pin connections, and the like, today each manufacturer builds what *he* thinks is best.

Your complete library will no doubt contain much additional manufacturer's data, mainly of the catalog variety, as well as your personal favorite articles from IEEE and trade publications—and, of course, your own textbooks, if you went through school after digital circuits became important. I hope that the collection described here, though certainly not exhaustive, will prove highly useful as a base on which to build acquisitions in special-interest areas of your own.

Walter Beam (F), a graduate of the University of Maryland where he also received the Ph.D. degree, has for the past four years been an independent consultant to computer manufacturers and users. Prior to this, Dr. Beam was variously affiliated with RCA, IBM, and Rensselaer Polytechnic Institute. Within the IEEE, he served as editor of IEEE's tape cassette series, "Soundings," and is currently Editor of IEEE Press.

Reprints of this special 48-page computer report (pp. 34–84 of this issue), (No. X74-021) are available at $3.00 for the first copy and $1.00 for each additional copy. Please send remittance and request, stating report number, to IEEE, 345 E. 47 St., New York, N.Y. 10017, Att: SPSU. (Reprints are available up to 12 months from date of publication.)

The fast-breeder reactor: when, where, why, and how?

An LMFBR plant will be built and the HTGR is under study; the factors involved are efficiency and safety

Caught in the vise of a tightening fuel and energy crunch—especially in fossil fuels—the U.S. has turned considerable attention toward a crash program for nuclear power generation. But the nation could be in a bind, as far as fissile materials are concerned, within the next 20 to 30 years. Although experts differ widely on this matter, there is a body of authoritative opinion that contends that reserves of natural uranium ores will be greatly depleted in the 1985–95 time frame, unless fast-breeder reactors, which produce more fuel than they consume, become operational. They refer to President Nixon's statement made back in April 1971: "Our best hope for meeting the nation's demand for clean energy lies with the fast breeder reactor." At that time, Mr. Nixon gave the nod to a program to build a commercially viable FBR by 1980.

In the fast-breeder development, there are essentially two ways to approach the objective: a liquid metal fast-breeder reactor (LMFBR), in which the coolant is metallic sodium, and a gas-cooled fast reactor (GCFR)—or its variations—in which hot helium would boil water into steam at more than 810°K—hundreds of degrees higher than the temperatures reached by the boiling water and pressurized water used in most conventional nuclear power plants.

Nuclear energy economy

As Donald Burnham, chairman of Westinghouse Electric Corporation has stated, "Our society must put more of its energy into the form of electricity and reserve oil and gas for such vital tasks as only these fuels can handle." This increased supply of energy will be generated from a nuclear power base. The resources of uranium and the availability of enrichment are limited; thus, we must provide for a self-supporting nuclear generation capability. Many people in the power industry feel that the breeder reactor is the answer to this requirement.

Operating experience

The LMFBR has been conditionally accepted by all of the major industrial countries (United States, U.S.S.R., Japan, Germany, France, Great Britain, and Italy) as the only viable breeder reactor system. Sodium is an excellent heat transfer medium, and the present designs utilize low-pressure systems (not much above ambient atmospheric pressure). In this country, the experimental breeder reactor (EBR-II) in Idaho Falls,* which is the test bed for the fuel for a demonstration plant (to be described later in this article), has operated for 10 years without serious mishap. Also, more than 10 years' operating experience has been gained in British and French fast-breeder reactor facilities.

Gordon D. Friedlander Senior Staff Writer

Anatomy of the LMFBR[1]

Advocates of the FBR concept believe the LMFBR offers a solution to both the problems of nuclear fuel resources and the enrichment of uranium.† Essentially, the LMFBR will utilize about 70 percent of the potential energy in uranium ore. By contrast, only 1 to 2 percent of the potential energy of uranium is utilized in fission reactors; the remainder consists of waste tailings. For example, it is estimated that the tailings produced by 1980 from light-water reactors alone will be more than 228 000 tonnes of depleted uranium. However, a 1000 MWe LMFBR would consume less than a tonne of uranium annually; therefore, 228 000 tonnes of this element would supply total U.S. power requirements—at today's capacity level of 250 000 MW—for about 1000 years. Thus, the proponents of the LMFBR contend that, when this type of breeder is established as an operational entity in the power industry, the sharp increase in uranium requirements we are presently witnessing will fall off sharply.

A quantum jump in overall efficiency. Added to the enormous increase in uranium ore utilization, the LMFBR would obtain more usable energy from uranium. For instance, a gram of plutonium fissioned in a FBR will furnish about 1½ times more heat energy than would the same gram of plutonium fissioned in a conventional thermal neutron reactor. Also, the thermal efficiency of conversion of heat energy to electric power in LMFBR is about 40 percent compared with the typical 33 percent for a light-water reactor. In fact, the breeder system utilizes up to 70 times more of the extractable energy from the uranium ore and provides a notable increase in thermal efficiency.

How the LMFBR works.[2] Figure 1 is a cutaway diagram showing the essential components of the LMFBR Demonstration Plant to be built in Tennessee. The 59 elements of the complex are listed.

In substance, the plant consists of a reactor and re-

*The first electricity generated from the atom in the U.S. was produced in a sodium-cooled reactor (EBR-I) in Idaho Falls. Further, sodium-cooled systems were considered in the early planning for the U.S. Navy's nuclear-powered submarines.

†Conventional fission reactors require that mined uranium must be processed through gaseous diffusion plants to increase the fissile fraction of the isotope U-235 from its natural level of 0.7 percent to levels of 1–5 percent.

actor-containment structure, fuel-service building, plant-service building, turbine wing, and a water-cooling tower complex. The reactor-containment building will have a normal atmospheric environment above the operating floor (which is at approximately grade elevation). This will permit personnel to have access to all above-grade areas of the structure during operation. However, inert gas controlled environments will be provided for the reactor and the primary heat-conveyance system in the vaults below the operating floor. Adequate cladding should ensure the containment of hazardous and radioactive materials.

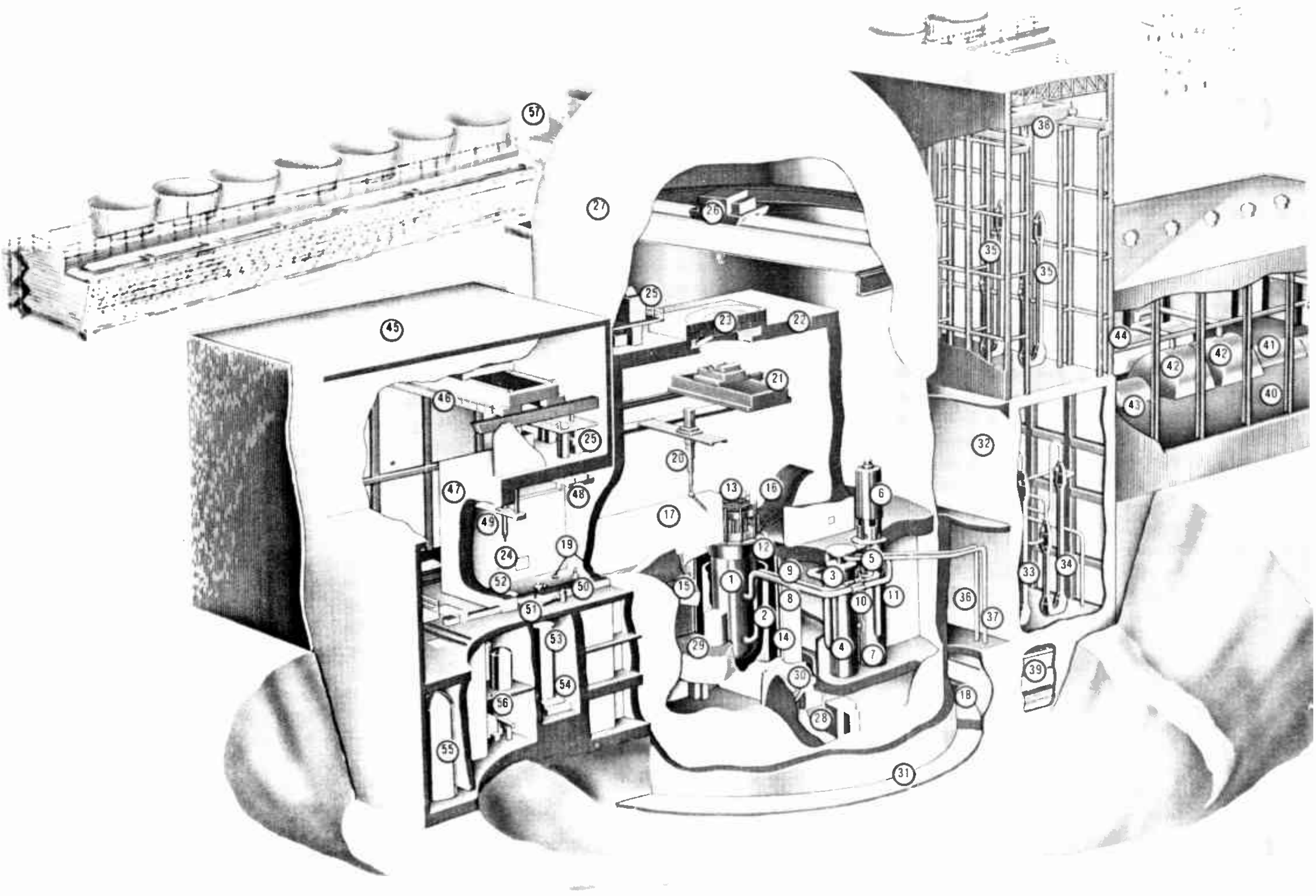
Liquid sodium (Na), at 660°K enters the reactor near the bottom of the vessel. In flowing upward, the metallic sodium passes around and through the core and blanket fuel assemblies, rises in temperature to 810°K, and exits near the top of the vessel.

A system of three parallel piped circuits conveys the approximately 1000 MWe, generated by the reactor, through primary and intermediate sodium loops to steam generators that produce steam at 755°K and 10 000 KN/m².

Pumps are located in each of the three primary and intermediate loops, and intermediate heat exchangers transfer heat from one to the other. (The pumps will be located in the "hot leg" as in the Fast Flux Test Facility (FFTF) facility presently under construction at Hanford, Wash.) Guard vessels around the principal components will ensure Na coolant containment and continued reactor core cooling in the event of inadvertent coolant leaks.

The core and blanket areas are composed of 348 hexagonal-shaped ducted fuel assemblies (Fig. 2), each about 4 meters in length. The active core, about 1 meter long, consists of 108 and 90 assemblies in the inner and outer core zones, respectively; each assembly contains 217 fuel rods.

The core will be initially fueled with homogeneously mixed plutonium and uranium oxides of between 19- and 27-percent plutonium for the oxide portions



of the inner and outer zones. The fissile plutonium content for the initial core fuel loading will be 1109 kg; the maximum design temperature of the stainless-steel cladding will be 930°K.

The core is designed for annual refueling, with one third of the assemblies replaced during each event. The initial fuel rods will be of the wire-wrap type (identical to those designed for the FFTF), with average peak burnups of 50 000 and 80 000 MWd/t,* respectively. Subsequently, recharges of an advanced-design fuel assembly are expected to attain average and peak burnups of 100 000 and 150 000 MWd/t.

The blanket areas surrounding the core consist of 150 hexagonal assemblies (similar to Fig. 2), each containing 61 wire-wrapped fuel rods charged with depleted uranium dioxide (UO_2) pellets. The expected breeding ratio is more than 1.2 to 1.

Primary and auxiliary equipment for normal and emergency conditions, plus safety provisions. The primary pumps will be single-stage centrifugal units, with axial suction and 125 000 liters/min capacity. Normal pump speed will be 900 r/min. Coolant flow under power failure conditions is provided by a natural circulation capability.

The steam-generator units illustrated are of the shell-and-tube type, with nonradioactive sodium flowing on the shell side and water/steam through the tubes. The shell is constructed of high-resistance stainless steel and the tubes are of Incoloy® to minimize corrosion/erosion problems.

The intermediate heat exchangers (IHX), located between the primary and secondary loops, are vertical, once-through counterflow shell-and-tube units, with the pressure on the secondary loop higher than that in the primary loop so that, in the case of any leakage, there would not be fission products in the secondary loop of the heat conveyance system. IHX dimensions will be about 13.7 meters long, and 2.3 meters in diameter.

The reactor will be provided with 19 boron carbide control rods, of which 15 will be used for normal control, and four will be reserved as an alternative shutdown system.

A safer approach to the FBR?

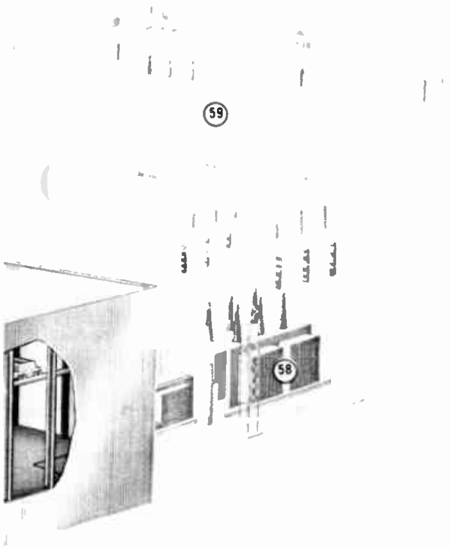
Engineers are presently working on a gas-cooled breeder nuclear power plant that could present strong competition to the LMFBR. This gas-cooled fast breeder is an extension of the HTGR† design (mentioned at the outset of this article) and has been under development over the past 15 years at Gulf General Atomic, formerly a branch of Gulf Oil's Energy and Environmental Systems Company.

As envisioned by its proponents, the gas-cooled breeder would function in conjunction with three HTGRs in close proximity. The breeder would regenerate its own fuel and, in addition, supply surplus fuel to the other reactors.

Prominent engineers in the power industry, plus

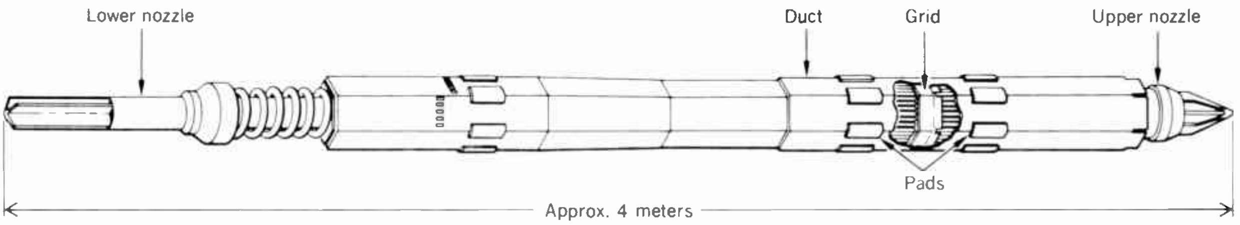
* This quantity indicates "megawatt-days per tonne," a unit employed for expressing the burnup of fuel in a nuclear reactor.

† This is the high-temperature gas-cooled reactor.



[1] Cutaway isometric view of the various components of the LMFBR demonstration plant to be built at Clinch River, Tenn. Scheduled for operation by 1980, the plant will serve as the "testing crucible" for future FBRs envisioned in the AEC's long-term program.

Legend:
1. Reactor vessel
2. Reactor vessel guard vessel
3. Intermediate heat exchanger (IHX)
4. IHX guard vessel
5. Primary sodium pump
6. Primary sodium pump drive motor
7. Primary sodium pump guard vessel
8. Primary sodium hot leg piping
9. Primary sodium cold leg piping
10. Isolation valve
11. Check valve
12. Reactor closure head
13. Control rod drive mechanisms
14. Shielding
15. Operating floor
16. Power and instrument cables
17. Ex-vessel transfer machine housing
18. Primary sodium storage tanks
19. Fuel storage tank port
20. Overhead manipulator
21. Refueling hot cell crane
22. Refueling hot cell (under-the-plug-concept)
23. Refueling hot cell roof port
24. Viewing window
25. Hot cell environmental control equipment
26. Containment building polar crane
27. Containment barrier
28. Compartment cooling and inerting equipment
29. Primary sodium purification equipment
30. Stairwell
31. Reactor building foundation
32. Steam generator building
33. Steam generator–evaporator
34. Steam generator–superheater
35. Steam generator—spare
36. Intermediate sodium cold leg piping
37. Intermediate sodium hot leg piping
38. Steam generator building crane
39. Sodium storage tank
40. Turbine-generator building
41. Electric generator
42. L.P. Turbine
43. H.P. Turbine
44. Moisture separator and reheater
45. Fuel service building
46. Fuel service building crane
47. Fuel service hot cell
48. Fuel service hot cell crane
49. Fuel service manipulator
50. Fuel service rotor drive
51. New fuel transfer valve
52. Shipping cask port
53. Irradiated fuel shipping cask and cart
54. Fuel canal
55. Emergency gaseous radwaste storage tank
56. Gaseous radwaste system equipment
57. Forced-draft cooling towers
58. Main stepup transformer substation
59. Switchyard



[2] A typical hexagonal-shaped ducted fuel assembly—one of the few hundred that will be initially installed in the core and blanket areas

members of the Joint Committee on Atomic Energy in the U.S. Congress, have repeatedly prodded the Atomic Energy Commission (AEC) to appropriate more money on gas-cooled reactors as a backup to the liquid-metal breeders that have been in small-scale operation for many years. For example, Representative Mike McCormack (D-Wash.) is a vocal advocate of the gas-cooled breeder. In a recent speech, before the Atomic Industrial Forum, McCormack reiterated this position. As possibilities foreseen in the role of the gas-cooled reactor, he cited the introduction of U-233 (instead of U-235, and plutonium) into the nuclear fuel resources, the linking of gas-cooled reactors directly to generating turbines, and ceramic containers that would permit future gas-cooled breeders to function at substantially higher temperatures and efficiences than those presently contemplated.

The fuel of an HTGR differs from that of the water-cooled reactors in that it contains large amounts of thorium which can be "fertilized" within the reactor to form U-233—a nuclear fuel with properties similar to those of U-235.

Coolant disadvantages. Professor Manson Benedict of M.I.T. feels that the coolant for each type of breeder has inherent disadvantages; but, these drawbacks can probably be mitigated. For instance, the sodium coolant of the LMFBR is opaque; thus, reactor operators cannot directly see what is occurring in

Opposing viewpoints on the FBR

Additional to the hazard of a radioactive sodium explosion, the LMFBR program has come under attack from many conservationists. David Brower, president of Friends of the Earth, says: "The core of a breeder reactor contains . . . about one tonne of Pu-239—a radiological poison so toxic that if properly reduced and dispersed, a tonne of it would far more than suffice to give lung cancer to everyone on earth."

With something as "hot" as plutonium (the principal ingredient of nuclear weapons), having a half life of 100 000 years, it is fairly obvious that extraordinary precautions will have to be taken in storing waste radioactive materials for hundreds of years before their radiation subsides to a safe level. The environmentalists emphasize that any competent nuclear physicist can readily convert breeder material into a king-sized nuclear weapon.

Countering these dire warnings and pessimistic reactions, Thomas Nemzek, the AEC's breeder project director, says: "The danger to the man in the street is negligible." He further argues that many computer safety analyses have indicated the very remote possibility of such a melodramatic scenario as that envisioned above. Finally, he virtually rules out the possibility of theft or hijacking of radioactive elements—especially plutonium—because of the extraordinary security measures that will be taken.

Nevertheless, critics stress that, even with the most elaborate safety and security systems imaginable, the chances of human and/or mechanical failure still remain—especially if scores of commercial FBRs are constructed in accordance with the AEC's present timetable.

In the matter of economics

The opponents of FBRs feel that their most effective argument, however, is on the question of overall economics. They note, for example, that this touchy subject is a paramount concern to the 350 utilities supporting the Clinch River demonstration project. As a safeguard against construction cost overruns, the utilities attempted to restrict their support commitment to the basic $250 million pledged, thereby letting the AEC pick up any extra tabs. But the Congressional Joint Committee on Atomic Energy would not buy this guarded approach last year, and a more equitable compromise formula was agreed upon.

Advocates of the FBR program contend that it could save the U.S. $2 billion per year in fuel costs by the 1990s. But the joker in the deck, insofar as that argument is concerned, is the probability that utility revenues will be more than $100 billion annually by the final decade of this century. Thus, the fuel savings would only represent about 2 percent of the industry's total revenues.

The critics allege that *capital*, not *fuel*, is the back-breaking expenditure in the operation of nuclear plants—especially the FBRs, which are certain to be far more expensive to build than light-water reactors. In view of this, they ask, why risk the problems, hazards, and unforeseen contingencies of an FBR-based power scheme? Therefore, the opponents urge a slowdown in the U.S. breeder-development program and the redirection of funds into crash R&D programs of economical coal gasification and nuclear fusion.

But despite all of the flak directed toward the FBRs, few opponents are demanding a complete halt to the projects. Instead, the majority of the critics urge an "agonizing reassessment" of the entire matter of breeder technology—including its viability and environmental hazards. They believe that a delay of several years' time is not that critical and it would be imprudent to "put all our *ergs* in one basket!"

the reactor. Furthermore, the sodium becomes highly radioactive and must be completely protected from any air or water. Such an occurrence could produce a violent explosion. However, the helium gas of a gas-cooled reactor is far less efficient than sodium in removing heat from the core of the reactor.

A giant leap forward?

But if the verdict is not yet in as to which of the two versions of the FBR is most viable, the LMFBR, at least in terms of practical application, holds a clear lead over the GCF. A major step was taken on November 15, 1973, toward President Nixon's 1971 objective alluded to previously when the Project Management Corporation (PMC) let a contract for about $90 million Westinghouse Electric Corporation for the nuclear portion of the United States' first large-scale demonstration LMFBR plant (to be called the Clinch River Breeder Reactor Plant) to be built at Oak Ridge, Tennessee. Through subcontracts now being negotiated, Westinghouse will be assisted by Rockwell International Corporation and the General Electric Company. Negotiations are also nearing completion with Burns & Roe, Inc., Oradell, N.J., for architect-engineering consulting services. The contract was signed on January 25, 1974.

The demonstration plant will have an output capacity of 350-400 MWe. The plan is to have the facility on the line by 1980, followed by a subsequent five-year demonstration period. The project is jointly funded by the U.S. Government, electric utilities throughout the country, and segments of the nuclear utility industry.

Organization and financing. PMC is a nonprofit corporation that was established to manage the project. Under terms of contracts signed with the Tennessee Valley Authority (TVA), Commonwealth Edison Company, and the AEC, PMC will provide overall management and coordination for the design, construction, and management of the plant. The plant will be operated by TVA as part of its power grid. The estimated construction cost of the project is $700 million.

The project has the broad support of both electric utilities and manufacturing industries in the U.S. As an indication of this, about 380 electric utilities have pledged more than $245 million. This represents the power industry's largest commitment to a single R&D effort. In addition, Westinghouse will contribute $7.3 million (plus environmental services valued at $300 000), and has offered to make some contributions in equipment.

The five objectives. The primary aim of the project is to demonstrate the value and environmental desirability of the FBR concept as a practical and economic option for the generation of electric power. Other principal objectives include

- Confirmation of the value of the LMFBR in conserving critical and nonrenewable natural resources.
- Development of important technological and economic data for the benefit of Government, industry, and the public.
- Provision of a broad base of experience and information vital to commercial and industrial applications.
- Verification of certain key characteristics and capabilities of breeders for utility system operation. (These encompass qualifications for licensing, reliability, safety, economic maintenance, and flexibility—plus electric power generated at reasonable cost.)

The ground covered, and what's ahead

The PMC consortium is convinced that the basic technology for the LMFBR is at hand. Most of the major design decisions have already been made, and much preliminary work has already been accomplished at the site. For example, maximum flood level studies, flora and fauna surveys, and meterological data acquisition have been completed and/or acquired. Geological and seismic studies will be completed this year.

The next major step is to translate to full-scale the nuclear steam supply system components and assemble them into a reliable power plant that will meet utility system operating requirements. The AEC's Fast Flux Test Facility (FFTF) will take the intermediate step in size extrapolation from earlier breeder developments and will serve as a point of departure for the Clinch River demonstration plant.

Safety, safeguards, and waste disposal

The advocates of the LMFBR have amassed a sizable stockpile of ammunition in defense of the concept. They cite, for example, that although Hans Bethe, a Nobel Laureate and physicist at Cornell University has been critical of the direction in which the LMFBR program has been going, he very strongly endorses the inherent safety of the system that includes

- A self-regulating mechanism in the event of overheating of the fuel.
- Its low probability of failure, and the sophisticated instrumentation subsystems that will be provided to monitor and preclude any potential failures.

The proponents concede the plutonium is an essential ingredient of a nuclear weapon; however, they contend that the plutonium generated in a commercial reactor is isotopically very different from that which has been produced for military applications. Reactor-grade plutonium has high Pu-240 and Pu-242 contents which greatly increase the plutonium requirements for weapon manufacture. In addition, there are the very complex chemical reactions required to reduce the plutonium to its correct metallic form (the LMFBR utilizes an oxide).

Furthermore—and most importantly—the trigger-mechanism for a weapon is a well-guarded secret; and, just making a "critical mass" is far from constructing a bomb.

Novel schemes are being investigated for the disposal of radioactive wastes by "burning" in reactors rather than by underground storage. For example, the long-life fission products can be transmuted in a reactor to short-lived isotopes that will decay to negligible radioactivity within decades instead of centuries.

REFERENCES

1. Taylor, J. J., "Conserving our energy resources with the liquid-metal-cooled fast-breeder reactor," internal paper, Westinghouse Electric Corp., 1973.
2. Jacobi, W. M., "Preliminary liquid metal fast-breeder demonstration plant design," *Nuclear Energy Digest #1* (Westinghouse Electric Corp.) pp. 3-4, 1973.

Special report

INTERCON '74: getting down to business

Conference program organizers are serious about new products, new markets, and solving society's problems, too

As the handiwork of electrical and electronics engineers becomes more entwined in the fabric of society, issues such as energy and materials shortages, international trade policies, pension reform, CATV, pollution control (among a host of other front-page stories) increasingly involve the electronics industry. These subjects, as well as numerous reports on the latest developments in all areas of electronic technology, make up the technical program of INTERCON '74. An added element this year is a series of special sessions organized by various IEEE Societies and Groups on timely subjects such as alternative energy sources, energy and social values, health care, and pollution, as well as design-oriented topics. Nine of these special sessions are listed in the table on p. 92. More are being organized at the time of this writing.

Gerald Lapidus Associate Editor

Another innovation this year is the condensation of last year's four days of technical sessions into three, so that attendees need not miss sessions on the last day because of travel arrangements.

Here are some of the highlights of the sessions—including, incidentally, two special evening programs on subjects of general interest. A schedule of the regular daytime program appears on the facing page.

Behind the headlines

Shortages of semiconductors are causing great concern throughout the industry. Superimposed on skyrocketing demand are both a shortage of silicon and the effects of power reductions on device manufacturers. Session 5, **"The Semiconductor Crunch,"**

Fiber optical communications may hold the key to future expansion of telephone systems. The state of the art will be reviewed in session 12.

ieee intercon74 Professional Program Schedule (Statler Hilton Hotel)

Date, Time	Grand Ballroom	Terrace Ballroom	Georgian Room	Penn Top South (18th floor)	Gold Ballroom	Sky Top Room (18th floor)	Penn Top South (18th floor)
Tuesday March 26 9:30 a.m.— 12 noon	Session 1 Electronic Calculators: Market, Design, Applications	Session 2 Imaging with Charge Transfer Devices	Session 3 The Foreign Thrust to Capture U.S. Markets	Session 4 The Changing Face of CATV	Session 5 The Semiconductor Crunch: When Will Deliveries Catch Up with Demand? What Will Happen to Prices?	Session 6 Radio Amateur Space Communication	Session 7 Testability—The Key to Automation in Circuit Testing
Tuesday March 26 2 p.m.— 4:30 p.m.	Session 8 Point-of-Sale Systems	Session 9 CCDs in Analog Signal Processing	Session 10 How Will Computer Costs Come Down?	Session 11 New International Markets	Session 12 Optical Communications	Session 13 Automatic Test Equipment—Lessons Learned and Deterrents to Wider Application	Session 14 Investment Plans for Engineers and Employers
Wednesday March 27 9:30 a.m.— 12 noon	Session 15 Technology Assessment and the Engineer	Session 16 Data Base Systems—A Technical Perspective for Engineers	Session 17 Microprocessor Architectures	Session 18 Packaging Concepts for Solid State	Session 19 Switching Systems Control by Minicomputer	Session 20 User Software for Automatic Test Equipment	Session 21 New Technology in Telephones
Wednesday March 27 2 p.m.— 4:30 p.m.	Session 22 Developing End-Use Markets for Electronics	Session 23 Recent Advances and Trends in Computer Storage	Session 24 Microprocessor Applications	Session 25 Electronic Displays in Action	Session 26 Control and Instrumentation Systems: For the Operator for a Change	Session 27 The Uncommon Carrier—Problems and Plans	
Thursday March 28 9:30 a.m.— 12 noon	Session 28 Using Memory Today	Session 29 Alternatives to Random Logic	Session 30 Solid-State Systems for Consumer and Industrial Products	Session 31 The Changing Marketing Interface—Who Wins, Who Loses?	Session 32 Growth Potential in New Microwave Semiconductors	Session 33 Required and Available Instrumentation Systems	Session 34 Earth and Ocean Physics Application Program
Thursday March 28 2 p.m.— 4:30 p.m.	Session 35 High-Density Semiconductor Memories	Session 36 Solid-State Systems for Automotive Electronics	Session 37 Venture Capital: Frontier of Finance	Session 38 Data Networks	Session 39 New Instruments, New Concepts, New Technology	Session 40 Go/No-Go Testing—A Thing of the Past?	

Group- and Society-organized sessions (Monday)

1. Affluence and effluence (evening)
2. Technology for health care delivery (one-half day)
3. Alternative energy sources (one-half day)
4. Engineering applications of superconductivity (one-half day)
5. Flexible composites in printed circuitry (full day)
6. Managing the engineer—art, science, or black magic (one-half day)
7. Open forum: energy and social values (full day)
8. Hybrid microelectronics clinic on semiconductor bonding (one-half day)
9. Plasma technology and its application to the production of power (one-half day)

(Tuesday morning) will focus on the various components of the problem including lead times and pricing, troublespots in discretes and ICs, the effects of shortages on procurement methods, and the prognosis for the future. The panel discussion will consist of presentations by two users, a distributor, and two manufacturers. A lengthy discussion period is also planned.

With **pension legislation** slowly winding through the legislative process, engineers will want to know the details of proposed programs. Session 14 (Tuesday afternoon) will be a panel discussion on the subject. The participants include Frank Cummings, of IEEE's council on Government affairs; Dick Backe, chairman of IEEE's Pension Committee; Bill Astrop, IEEE's investment counselor; and Tom Sternau, IEEE's insurance plan administrator. The panel will cover the IEEE-sponsored nonqualified plan, portable qualified plans, methods of investment for company-sponsored plans, and recent actions by the IEEE Pension Committee.

A **CATV system** used to be a one-way cable network to distribute television signals from a central point to standard TV receivers at subscribers' homes. The TV signals were mostly TV broadcasts received at a central antenna site, sometimes adding one or more locally originated programs or information.

While this description still fits most of the systems now in operation, new systems (some of which are already operating) present a completely new approach. The receiver may shortly be designed to utilize the full capabilities of cable transmission, in addition to TV reception. Other types of terminal devices will open the door for new services available for the home, such as computer access. Some of these services will be interactive, requiring bidirectional cable operation. Session 4 (Tuesday morning) will deal with the technology of these developing capabilities, will relate the experience of those who have used them, and will report on related work in FCC committees.

The growth of electronics into totally new areas requires continuing study of potential end-use markets. Among the basic principles of **market development** are learning the market needs and size, knowing who makes the buying decisions, knowing when and how these decisions are made, and structuring the marketing program to cope with these factors. Session 22 (Wednesday afternoon) will concentrate on marketing in three new areas: point-of-sale systems, municipal governments, and the textile industry. The session will lead off with a paper on selling in a climate of change.

CCD image sensors such as the first CCD TV camera will be covered in session 2.

The **automotive industry** represents about a $100 million market for the electronics industry today. By 1980, this figure is expected to be seven times what it is today. Session 36 (Thursday afternoon) will describe the systems that these dollars will be paying for and how these systems—currently developed on a piecemeal basis—may one day be under control of a central processor. Some of the topics covered will be charging systems, solid-state transducers, and electronic fuel injection. In addition, a paper in session 24 on microprocessors will describe their possible use as on-board vehicular computers.

The INTERCON nightcap

Although it is often difficult to face evening meetings after sitting through a number of daytime sessions, INTERCON attendees will find the two evening sessions well worth their time. For those with occult interests, the Tuesday evening session will take a look at **technology in the twilight zone**—phenomena for which present laws do not provide adequate explanations, but which have recently received the attention of serious investigators and have produced experimental results that are impressive. Some feats to be scrutinized include: communication with people in shielded environments such as submarines, measuring and photographing fields in and surrounding living objects, and use of the subconscious mind to solve complex technical problems. In addition, a paper will be presented on telepathic channel capacity.

"Affluence and Effluence" will be the subject for Monday evening when a panel will deal with the question of whether a rich society must also be a wasteful society. On docket are discussions of waste in product manufacture, economics of recycling, waste of human resources, and energy waste.

New trends in computer technology

Computing costs are being affected by reductions in circuit costs, counterbalanced by increases in software costs. **"How Will Computing Costs Come Down?"** (session 10, Tuesday afternoon) will examine these factors in terms of present systems and describe how they will affect the expected growth of distributed computing and personal computing. There will also be a paper on semiconductor memory costs.

Microprocessors are relatively new and therefore applications information presently is limited. Although a multitude of uses can be foreseen, there are relatively few applications now in the field and much of the information is preliminary or proprietary. Session 24 (Wednesday afternoon) will take up three **microprocessor applications,** including terminal control for a data communications network, point-of-sale terminals, and on-board automotive computers. The final paper of the session will relate microprocessors to minicomputers—past, present, and future. A related session (17, Wednesday morning) will be concerned with several new **microprocessor architectures.**

New memories have brought about new cost-effective ways of designing digital systems, by providing the alternative of building logic systems with read-only memories instead of logic gates. Session 28 (Thursday morning) will focus on new types of **read-only memories** such as MNOS and PROM devices and will describe design techniques for replacing random logic with memory.

Also in a logic design vein, a session on **alternatives to random logic** (29, Thursday morning) will be concerned with difficult LSI chip layout problems that arise from the highly irregular and complicated structure of random logic. Suggestions will be offered for reducing layout complexity by using conventional forms of logic or else changing to more powerful designs. These suggestions relate to using programmable IGFET custom logic, array logic, and threshold logic.

In the area of **mass storage,** although today's systems primarily use magnetic tapes and disks, new technologies may soon be vying for these applications. Among them are magnetic bubbles, charge-coupled devices, and optical beam addressable memories. These new technologies will be reviewed in session 23 (Wednesday afternoon), along with a discussion on the future of magnetic recording for computer storage.

An eye on communications

Optical communications promise unsurpassed message-handling capabilities. Session 12 (Tuesday afternoon) will review the state of this art, with papers on systems, concepts of wave propagation in optical fibers, fabrication of optical fibers, telephone systems based on fibers, CATV and subscriber networks, and military communications.

Integrated circuits offer the promise of providing better performance, new services, and greater convenience for telephone sets. Session 21 (Wednesday morning) will cover some new approaches toward **telephone set design,** including an electronic telephone, a loudspeaker telephone, the "Touch-A-Matic" telephone, a picture telephone, and coin telephone signaling.

The essential details

INTERCON '74 will be held in New York City in the week of March 25. The technical program will be held at the Statler Hilton Hotel and the exhibits at the Coliseum. (Free-of-charge shuttle bus service will be provided between both locations.) Exhibits will be run from Tuesday, March 26, through Friday, March 29. The technical program will commence on Monday with nine or more special Group and Society organized sessions. The main 40-session technical program will be held on Tuesday, Wednesday, and Thursday of the week.

Specialized common carriers and **domestic satellite carriers** are planning expansions of regional and national private line service networks. A panel discussion (session 27, Wednesday afternoon) will detail expansion plans and cover problems such as local loops, equipment difficulties, financial arrangements, and Government regulation.

Charge-coupled devices are quietly starting a revolution in analog signal processing because they can do jobs that could not previously be handled by analog techniques. In addition, CCDs can perform these functions while reducing cost and power requirements by factors ranging from 10 to 100. Session 9 (Tuesday afternoon) will cover the design and application of several types of transversal filters and other devices used to process analog signals.

Although the CCDs described above are not yet commercially available, CCD image sensors are already heading toward the marketplace. One CCD TV camera has already been announced and at least two others are now in the works, with emphasis being placed on increasing resolution and developing new transfer techniques. Scheduled for presentation in session 2 are papers on high-resolution sensors, a parallel transfer register, charge-coupled devices, a charge-coupled imager for 525-line television, a CID imager, and noise considerations in solid-state imagers.

Recent progress in **microwaves** will be covered in session 32 (Thursday morning), which will focus on developments having both commercial and military potential. Included are millimeter IMPATT diodes; a 29-dB, 600-MHz amplifier; GaAs FETs and their applications in hybrid microwave ICs; and DMOS for microwave linear and switching applications.

New technology is radically changing **instrument design,** both in terms of the ways instruments are built and in the addition of new functions required for taking measurements on new devices. Session 39 will provide examples of these two new directions. Included for discussion are papers on a 3½-digit autoranging multimeter built on a hybrid IC, a logic test instrument that stores and displays up to 16 channels of data, a microwave synthesizer, and recent advances in fast storage oscilloscopes.

A related session on Thursday morning (session 33) is concerned with problems and solutions for writing adequate **test specifications** for increasingly complex systems. Included for discussion are a review of test systems and trends, data-base driven testing, and aids for general purpose test systems.

Inside IEEE

IEEE forum studies energy-crisis impact

Concern over the effects of the energy shortage on IEEE members, and on the entire engineering community, was expressed in Washington, D.C., on January 11, when representatives of IEEE met with other interested persons from industry, the Government, and professional organizations to discuss the situation. IEEE President John J. Guarrera was moderator of the Institute-organized meeting.

One of the prime purposes of the Conference on Impact of the Energy Crisis on the Electrical and Electronics Industries was to lay plans for studies to forecast this impact. Such studies would have three major phases.

The first phase is essentially a gigantic technology forecasting and assessment task considering the economy as a whole, and the probable impact of the energy crisis on consumer life styles.

The second phase would translate the findings of the first phase, together with the data on energy and materials shortages in the electrical and electronics industry, into forecasts of profitable product lines that would sustain the industry.

The third phase will require assembly of extensive information concerning present employment of engineers in different segments of the electrical and electronics industry, studies of probable changes and shifts in employment, and overall effects on the industry.

William Morsch, IEEE Manager for Technology Forecasting and Assessment, discussed models to be used in predicting the effects of the crisis on engineering manpower. He pointed out that new R&D spending can change the total output and thus can have an impact on demand and on engineer employment. He recommended the "reengineering" of energy-using products. For example, the petrochemical industry needs electrical engineers to increase the energy efficiency of its processes.

IEEE Past President Harold Chestnut, Executive Director Donald Fink, and Robert Cotellessa, IEEE Vice President for Technical Activities, also participated in the meeting.

Others in attendance included Hort Ashley, AIAA president and president of the Coordinating Committee of Engineering Society Presidents; W. Kenneth Davis, Chairman, Task Force on Energy, National Academy of Engineering; and Robert L. Loftness, Electric Power Research Institute; as well as IEEE Congressional Fellow Ron Larson, and Ralph Clark, of IEEE's Washington Office.

IEEE names winners of '74 annual awards

The IEEE Board of Directors, acting upon the recommendation of the Awards Board, has announced the 1974 recipients of the Medal of Honor, the Major Annual Awards, and the Prize Paper Awards. All the awards will be presented during IEEE INTERCON '74 in New York, N.Y., March 26–29.

The Medal of Honor, IEEE's highest award and the Institute's equivalent of a Nobel Prize, goes to Rudolf Emil Kalman, director of the Center for Mathematical System Theory at the University of Florida, Gainesville.

The winners of the Major Annual Medals are as follows: Edison Medal—Jan A. Rajchman; Founders Medal—Lawrence A. Hyland; Lamme Medal—Seymour B. Cohn; Education Medal—John G. Truxal.

Prize Paper Award winners are: W. R. G. Baker Prize Award—David B. Large, Lawrence Ball, and Arnold J. Farstad; Browder J. Thompson Memorial Prize—Jorn Justesen.

The Medal of Honor and the Major Annual Medals will be presented at the Banquet on March 27 and the Prize Paper Awards at the Directors' Reception on March 26.

Rudolf Emil Kalman (F) is being awarded the Medal of Honor "for pioneering modern methods in system theory, including concepts of controllability, observability, filtering, and algebraic structures."

Born in Budapest, Hungary, May 9, 1930, he received the S.B. and S.M. degrees from the Massachusetts Institute of Technology and the D.S. degree from Columbia University, all in electrical engineering.

In 1953 he joined the staff of the M.I.T. Servomechanisms Laboratory as a research assistant, and after a year there transferred to the E. I. du Pont de Nemours Experimental Station in Wilmington, Dela., in the same capacity. From 1957 to 1958, he directed a research group at the IBM Research Laboratory, Yorktown Heights, N.Y., leaving to become a research mathematician at the Research Institute for Advanced Studies, Baltimore, Md., and subsequently associate director of research and head of the RIAS Center for Control Theory.

After serving as professor of mathematical system theory at Stamford University in 1964–71, Dr. Kalman joined the faculty at the University of Florida, where he is presently graduate research professor and director of the Center for Mathematical System Theory. He has held visiting appointments in France, England, and Switzerland.

Dr. Kalman is an editor of the *Journal of Computer and System Sciences, Mathematical System Theory, Journal of Statistical Physics, Kibernitis, Stochastics,* and *Folia Clinica Internacional.* In 1962 he was named Outstanding Young Scientist of the Year by the Maryland Academy of Sciences.

Jan A. Rajchman (F), staff vice president, information sciences, RCA Laboratories, Princeton, N.J., will receive the Edison Medal "for a creative career in the development of electronic devices and for pioneering work in computer memory systems."

Recipient of the diploma of electrical engineering in 1934 and the degree of doctor in technical sciences in 1938, both from the Swiss Institute of Technology in Zurich, Dr. Rajchman started with RCA in 1935, working on electron optics. He developed, the now widely used electron multiplier tubes, and during World War II, as one of the first to apply electronics to computers, he developed a resistive matrix switch that was the first read-only memory. He also conceived and developed the first purely digital random-access electronic memory—the selective electrostatic tube. In his present post, he provides technical guidance to RCA Laboratories in Tokyo and Zurich, directs an exploratory research group in Princeton, and assists in the overall management of RCA Laboratories.

In 1947 Dr. Rajchman was corecipient of the Levy Medal of the Franklin Institute for his work on the betatron, and in 1960 he received the Morris Liebmann Award for his many contributions to magnetic devices. He was elected to the National Academy of Engineering in 1966.

Lawrence A. Hyland (F), general manager and vice president, Hughes Aircraft Company, El Segundo, Calif., has been named recipient of the Founders Medal "for leadership and management in the field of electronics."

Mr. Hyland was born in Nova Scotia on August 26, 1897. He joined the U.S. Naval Research Laboratory in 1926 as an associate radio engineer, and while there was the first to demonstrate the reflection of radio waves from objects—an important step in the development of radar. Later, he founded and for five years headed the Radio Research Company of Washing-