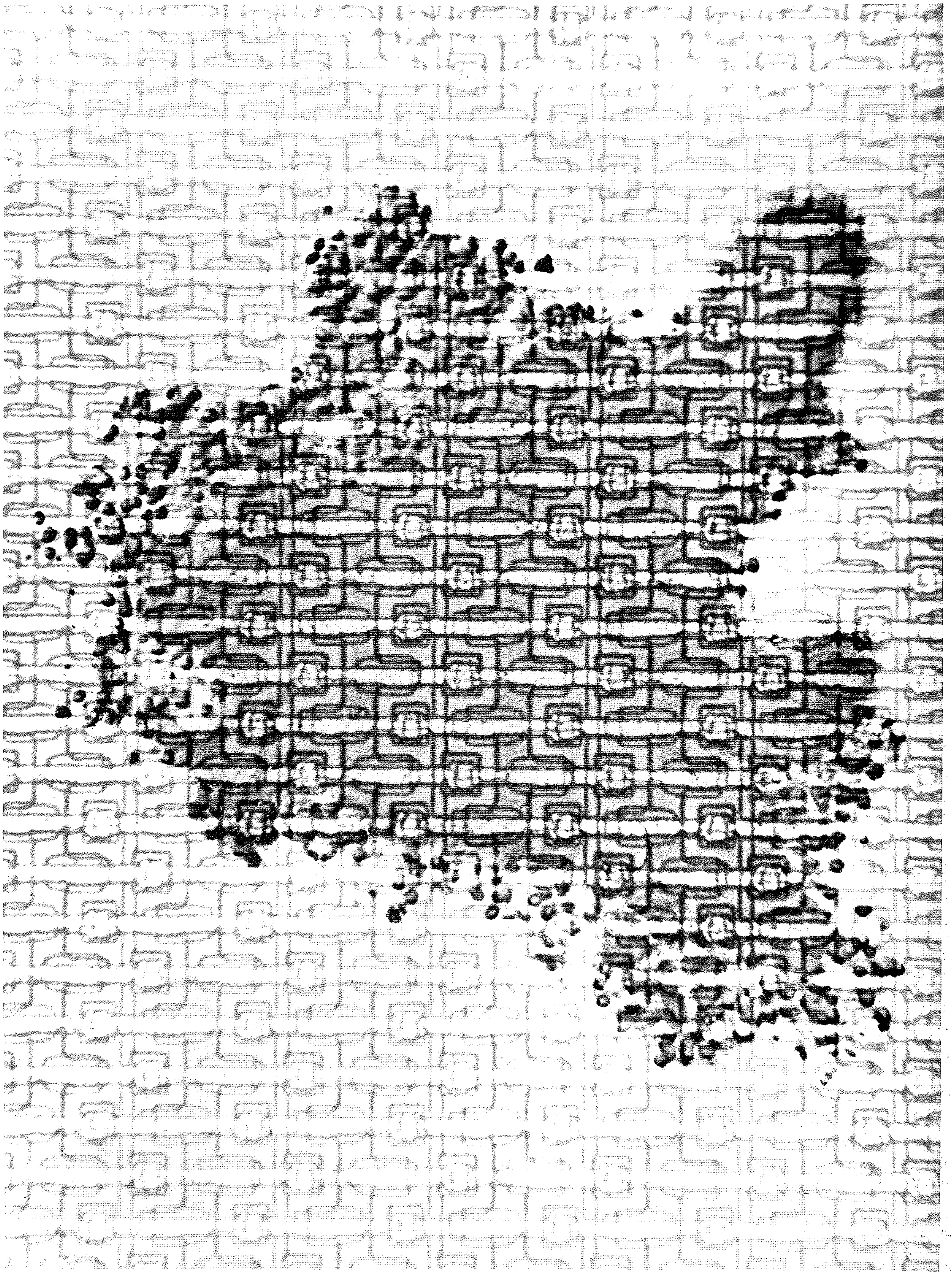
Stuart E. Madnick
Sloan School of Management
M.I.T.

The Future of Computers

Reprinted from Technology Review
Volume 75, Number 8, July/August, 1973

Cambridge, Massachusetts 02139



The blob is an amoeba, superimposed on a memory chip announced by I.B.M. in February, and now in test production. The amoeba covers 40 "transistor cells," each of which, .00184 inches square, can store a binary 0 or 1: a "bit." The entire memory chip, a quarter of an inch square, can store 8,192 bits. Photo courtesy I.B.M.

Stuart E. Madnick
Sloan School of Management
M.I.T.

The Future of Computers

A survey of a rapidly changing field. Briefly, computers will be more powerful, cheaper, and easier to use.

During the 1960s, after a decade of gestation, the computer captured the public's imagination and attention. It was in this period that many of today's engineers and managers first became exposed to, and knowledgable about, computer systems. Advances have continued into the '70s, but the scope of the industry has become so broad that it is difficult for anyone not deeply involved in the field to have followed the more recent changes. This paper will highlight many of these developments and forecast their future impacts. Due to the scope of the subject and the brevity of this article, it will be possible only to touch lightly on each matter, but it is hoped that this article will help to put developments into perspective and serve as an initial reference point for readers interested in pursuing a particular topic.

We will divide our subject into three categories: technological cost/performance breakthroughs in computer manufacturing; the evolution of computer system architecture, for both hardware (the computing apparatus) and software (computer programming); and major steps toward meeting the requirements and capabilities of the user.

Technological Cost/Performance Breakthroughs

The hardware in a conventional computer system consists of an ensemble of components, including the *processor*, which performs computation; *main storage*, which holds the data and instructions used by the processor; and *input/output devices*, which include input-only devices such as punched-card readers, output-only devices such as printers, and input-and-output devices (also called *secondary storage devices*) such as magnetic tape units.

Many technological developments in computer system hardware will have significant effects on cost, performance, size, and reliability. Since there is extensive activity in computer technology, we will single out only major trends. One warning is necessary: *Costs* in the computer industry are highly volatile, difficult to determine, and strongly influenced by production volume. *Prices*, however, are usually determined by market conditions and often have little technical significance.

☐ **Microprocessors** can be defined as small computer processors that consist of one or a few semiconductor integrated circuits, manufactured by techniques of large-scale integration (L.S.I.). Integrated circuits are based upon conventional semiconductor transistor principles, but by using special materials and production (for example, ion implantation and photographic masking), it is possible to manufacture and interconnect hundreds or thousands of transistors on a single silicon chip. Such a processor's size is measured in inches. A simple form of this technology has sparked the recent growth of hand-held electronic calculators.

There are many differences among the various microprocessors. One of the earliest and simplest, the Intel MCS-4, is a four-bit parallel processor, where a bit (short for *b*inary dig*it*) is a unit of information: a choice between a 0 or a 1 in the binary number system. A four-bit parallel processor performs operations on four bits simultaneously, rather than one at a time. The MCS-4 can execute 45 different instructions with a 10.8 microsecond instruction cycle—that is, it can execute almost 100,000 instructions per second. By comparison, one conventional computer, IBM's 370/145, has a 16-bit parallel processor, can execute over 100 different instructions, and has an instruction cycle of a microsecond or below (about 1,000,000 instructions per second).

The microprocessor does have important advantages in cost and size. The heart of the MCS-4, the 4004 Control and Arithmetic Unit, sells for $30 each in 100-unit lots. The entire 4-chip MCS-4 system sells for $48 each in 100-unit lots.

There are no serious technical impediments to the development of microprocessors, since they are typically based upon standard metal-oxide-silicon large-scale-integration (M.O.S./L.S.I.) technology. The present problems are largely customer education and acceptance. The manufacturing cost is very sensitive to production quantity; it costs only slightly more to manufacture 10,000 microprocessors than it would to produce 1,000. The joke that the only raw material required is a few shovels of beach sand, though not technically true, indicates the situation. (The table on the next page illustrates the sensitivity of L.S.I. cost to production quantity for processor and memory circuits.) The marketing problem becomes quite apparent when one recalls that after 25 years there are still less than 200,000 con-

---

Stuart E. Madnick studied electrical engineering and management at M.I.T., receiving four degrees, including a Ph.D. in 1972. In that year, he was appointed Assistant Professor of Management Sciences at M.I.T.'s Sloan School. Professor Madnick's research concerns the application and technology of advanced information processing systems—most recently, storage hierarchies and information system protection.

		Manufacturing cost per chip in production quantities of:		
Circuit Type	*Complexity*	*10*	*100*	*10,000*
Logic circuits (processor)	2,000 (equivalent transistors)	$260.00	$30.00	$ 5.20
Random-access circuits (memory)	16,000 (bits per chip)	$422.40	$57.60	$20.48

The sensitivity to production volume of processor and memory circuits manufactured by large-scale integration. The need to develop a sizeable market to realize substantial savings in manufacturing costs is obvious. Source: G. A. Saxton & Co., Inc., "Future Computer Systems," by L. Duane Kirkpatrick, *Industry Note No. 33,* March 26, 1973.

ventional computers in the world. For economic production, it is desirable to sell that many processors *per year*.

In spite of these problems, the list of participants in the microprocessor sweepstakes reads like a Who's Who of the semiconductor industry. Companies enter and leave and re-enter this list continually.

The immediate need is to develop uses for microprocessors.

"Non-Computer" applications, by which we mean integrating the microprocessor into a larger system, have been achieved for years with minicomputers (computers selling for $10,000 or less), but will be accelerated by the low-cost microprocessor. The possibilities are limitless. Frequently mentioned applications include machine-tool control, telephone switching, medical electronics, automotive controls, and digital watches. In addition, there are many applications for which a low-performance, inexpensive microprocessor is quite adequate. Compared with human skills, a "slow" 100,000-arithmetic-operations-per-second "personalized" computer is quite impressive. For example, the Hewlett-Packard HP-35 Pocket Calculator (battery powered) can perform logarithmic and trigonometric operations with about 10-digit accuracy in less than a second. The newer HP-80 Business Calculator can perform interest rate, averaging, and other calculations at similar speeds. Such calculators are faster and easier, and are usually more accurate, than slide rules or table-interpolation techniques. Microprocessors will also have their impact upon sophisticated remote computer terminals. In fact, much of the current microprocessor activity is jointly sponsored by terminal manufacturers and the semiconductor companies. Finally, the larger and more powerful microprocessors could pose a threat to the current minicomputer market. These microprocessors could evolve into small stand-alone business computers (accounts receivable, payroll, etc.)

Medium-scale computers, such as the IBM 370/135, 370/145, and 370/155, represent the major segment of the current computer market. A single microprocessor and even its slightly more expensive and powerful cousin, the minicomputer, does not have the processing power or input-output flexibility to compete with the medium-scale computer. On the other hand, medium- and large-scale computers are becoming more and more decentralized. Separate processors are being used to serve different functions. In addition to the central processors that execute instructions, there are input/output processors (often called channels) and special input/output-device processors (often called control units). As the cost of microprocessors drops and their performance improves, one becomes tempted to use these general-purpose units in place of the many different specialized processors in a medium-scale computer. This trend is already apparent in the recent introduction of the Burroughs BI700 system, based upon the Burroughs "D" machine multiprocessor military computer, and the IBM System/370 Models 115 and 125, which use several separate microprocessors internally.

There are many factors likely to accelerate this trend toward multiprocessor configurations. One intriguing argument is based upon the impact of the rapidly changing state of the art on the development cycle—the time from product conception to full production. Typically, the development cycle is about five years for medium-scale computers, but the simpler minicomputer often breezes through in two years, occasionally in less than 12 months. Many important decisions must be made early in the design cycle, involving such matters as system performance, circuit types, memory modules, and packaging approaches. The designer of the medium-scale computer must make these decisions four to five years before production. If he chooses to use existing technology, the final product will be several years behind the state of the art. If he extrapolates and projects the availability of future technology, he is likely to make some bad guesses that will result in last-minute redesign and inefficiencies. By building medium-scale computers out of multiple general-purpose microprocessors, the final design may be slightly "less optimal" than one using specialized processors, but that is more than offset by the increased ability to use more current manufacturing techniques, as well as the advantages of much larger quantity mass production, which dramatically reduce the cost of such a system. The miniaturized size of the processors will allow considerable reductions in cabinetry, power supplies, etc. Modular construction will have its impact on other costly areas, such as servicing. Processors might even become disposable.

A long-range forecast for microprocessors is difficult. The present sales prices may be somewhat artificial, since few of the manufacturers are in full-scale production. There are improvements possible in all directions. Eight-bit and sixteen-bit parallel microprocessors have been developed. There has been at least one prediction of microprocessors costing as little as $1 within 25 years that will have a processing rate up to 10,000,000 instructions per second. But such a long-range estimate is strictly conjecture.

The dramatic reduction in microprocessor costs must be carefully considered. The current manufacturing costs of a computer system represents about 25 per cent of the sales price (marketing, software development, research, and profit account for the rest). The processor may represent only a third of the system, the other two-thirds being memory and input/output devices. Furthermore, the processor's electronics, excluding power supplies, etc., may represent less than a third of the processor's total cost. Thus, the advances in microprocessors affect less than three per cent of a medium-sized computer system's sales price! Thus, while we can expect tremendous advances in processor technology, the impact upon the user will be minimal unless there are also dramatic changes in other areas.

	In 1970		In the late 1970s	
System Component	*Cost*	*Percentage of total*	*Cost*	*Percentage of total*
Main memory (1,000,000 bytes)	$120,000	35	$ 2,400	4
Special memories	40,000	11	2,000	3
Processor logic circuits	50,000	15	2,600	4
Special circuits	20,000	6	8,000	12
Processor subtotal		32		19
Packaging	30,000	9	10,000	15
Power and cooling	30,000	9	15,000	23
Other	50,000	15	25,000	39
Miscellaneous subtotal		33		77
	$340,000		$65,000	

The costs—recent and projected for the end of this decade—of the component parts of a large computer's mainframe (its processor and its main memory). Source: "Future Computer Systems," G. A. Saxton & Co., Inc.

☐ **Main Storage** (also called main memory or, historically, "core memory") typically represents about a third of a computer system's cost. The breakthroughs in this area—both attained and predicted—have received considerable attention.

Semiconductor technology can produce circuitry capable of "storing" binary 0's and 1's. Due to the very high volumes, simplicity of structure, and modularity of these memory devices, all of the semiconductor technology benefits to processors also apply to memory. Most of the computers currently being manufactured use semiconductor main storage instead of the traditional magnetic ferrite core memory.

The semiconductor memory market is extremely competitive in both price and technology. The Intel 1103, a 1,024-bit M.O.S. semiconductor memory circuit, has become an industry standard and versions are manufactured by several companies. Due to economies of scale and a very steep "learning curve," the cost of such circuits has dropped by a factor of ten in a little over one year. With the eventual commercial maturity of larger semiconductor memory chips (4,096 bits and above), the cost per bit of memory is likely to drop by another factor of ten in the next few years. (The photograph on page 34 shows an 8,192-bit memory chip now in test production.)

We can expect future computers that have larger capacities, smaller sizes, and less costly main storages. As an indication of this trend, the minimum size IBM 370/135 (at 98,304 bytes of main storage; a byte is eight bits) is larger than the maximum size IBM 360/30 (at 65,536 bytes), its predecessor.

Main storage used to be the most expensive component of a computing system. Based on the estimates of the table on this page, it will soon be possible to obtain ten times the storage capacity at one-fifth the cost. Many of the current ploys aimed at conserving main storages (program overlays, input/output buffering and handling, for example) will be unnecessary. This will result in easier program development and simpler, yet more powerful, operating systems.

☐ **Intermediate and Secondary Storage.** The use of high-performance, direct-access secondary storage (such as magnetic disk units) was accelerated by the IBM System/360, introduced in 1964. By now, it is an area of considerable competitive pressure. The sales volume of the basic IBM 3330 Disk Storage Unit alone is expected to reach over 1 billion dollars. A single 3330 module, storing 100,000,000 bytes, has more than ten times the capacity and double the access speed of its ancestor, the IBM 2311, introduced less than ten years ago.

Current secondary storage devices are largely based upon rotating magnetic media (for example, magnetic disks, magnetic drums, magnetic tape strips, and so on). This technology can probably be pushed another factor of ten in capacity and a factor of two in speed in the next five years. Beyond that point, there are at least three limitations to the electromechanical approach. We are rapidly approaching physical limitations in magnetic media recording capacity and speed. Due to the need for costly mechanical motors, the cost per bit of storage may be decreased by increasing capacity, but the unit cost continues to increase. Finally, the mechanical approach has inherent speed and reliability limitations.

There are numerous technologies being pursued that lead to storage devices falling between the traditional high-cost, high-performance main storage and the lower-cost, low-performance electromechanical secondary storage. All of these intermediate storage approaches have been successfully demonstrated in the laboratory and, in some cases, in limited production. The most successful should be in full production within five years. These technologies include:

M.O.S./L.S.I. Shift Registers. The technology is similar to that of M.O.S./L.S.I. main storage modules, but rather than direct "referencing" of each storage location, the data bits are circulated (shifted) along a register. Data is usually read or written only at the ends of the shift register, analogously to the use of read/write heads of a conventional rotating electromechanical storage device. It is possible to produce M.O.S./L.S.I. shift registers with higher capacities and lower prices than those of comparable direct-access M.O.S./L.S.I. main storage.

Charge-Transfer Devices, specifically Charge-Coupled Devices (C.C.D.'s) and Bucket-Brigade Devices. C.C.D.'s are logically similar to the M.O.S./L.S.I. shift registers but are based upon the controlled movement of electrical charge rather than on transistor-like circuits. They consist of three layers: the semiconductor material, an oxide layer and metal electrodes. C.C.D.'s are potentially more compact, simpler, and lower in cost than conventional integrated circuits. Bucket-Brigade Devices differ from C.C.D.'s in that they can be constructed from discrete components.

Magnetic Bubbles. Again, the logical concept is the same as that of shift registers. The physical phenomenon is based upon the development of "magnetic bubbles" in certain materials that can be made to move along paths on the surface. Although storage and retrieval of data from magnetic bubbles is slower than from other technologies, the bubbles offer the possibility of very low cost through very high densities (1 billion bits per inch has been forecast).

Optical (Laser Beam) and Electron Beam Storage. There are several beam approaches being developed. One strategy is based upon focusing the beam on the surface of a material and, depending upon the intensity, either reading or writing a bit at that spot. Alternatively, using laser and interference pattern (holographic) concepts, an array of information can be accessed at a time. Some devices can both read and write, while others can only read (write once, non-erasable permanent storage). Current components could produce a holographic read-only memory of 10 to 100 million bits with an access time of about five microseconds.

	Cost, in cents per byte	*Typical unit capacity, in millions of bytes*	*Random access time, in microseconds*
Main storage (now)	10	.1—1	.6
Secondary storage (now)			
moving head	.01	100	40,000
fixed head	2	10	5,000
Main storage (1975)	1	.1—2	.2
Intermediate storage (1975)	.01—.1	1—20	1—1,000
Secondary storage (1975)	.002	200	20,000

The costs, capacities, and access times of storage now, and projected (by the author) for 1975. "Intermediate storage" signifies a new class of devices now being created that will provide their data faster than secondary storage, but will not cost as much as main storage.

The current prices of storage and the prices anticipated for 1975 are shown in the table on this page. It is important to note that intermediate storage devices are both cheaper and faster than "fixed head" secondary devices such as magnetic drums, in which there is a separate read/write head for every recording position and the appropriate head is electronically selected. But "fixed head" storage represents a small fraction of the secondary storage market. The new intermediate storage devices will coexist with faster, though more costly, main storage and slower, though less costly, "moving head" secondary storage, in which there are fewer read-write heads (which are expensive) than there are recording positions and the heads are mechanically moved from one position to another. The future of these storage levels will depend upon changes in computer system architecture and applications to take maximal advantage of each level's unique characteristics.

☐ **Archival Storage.** By standards of a decade ago, today's secondary storage devices have enormous capacities. An eight-module IBM 3330 Disk Storage Unit has a capacity of 800 million bytes or "characters." (One byte is not required for a character, but it is generally the industry standard at this time. A byte's eight bits allow 256 different permutations of 0's and 1's—more than enough to encode upper and lower case letters (26 times 2), numbers (10), and punctuation and "control" characters—"carriage return" and others.) If we assume that there are about 4,000 characters on a single-spaced 8½" by 11" sheet of paper, 800 million bytes is comparable to 200,000 sheets of paper. Yet potential information storage requirements greatly exceed this capacity. For example, as part of an anti-trust defense, IBM had submitted over 27 million documents as of January, 1973.

To satisfy these needs, there has been considerable work on the development of "archival storage devices" capable of storing enormous amounts of information economically. These devices are often called terabit memories since they are designed to hold over 1 trillion bits of storage (1 trillion bits is about 120 billion characters, or roughly 30 million 8½" by 11" sheets of paper). There are several archival storages already on the market, including Grumman's MASSTAPE, Ampex's Terabit Memory (TBM), Precision Instruments' UNICOM, and IBM's 1360 Photo-Digital Storage (PDS). They provide direct access to over a trillion bits of storage with a maximum delay of a few minutes. Typical cost per byte is around .001¢. In some devices, such as MASSTAPE and TBM, the information is erasable and rewritable, as in conventional computer storage devices; in others, such as UNICOM and PDS, the information is permanently written and is not erasable (like using ink pens).

Even today, an archival storage unit can hold the capacity of 10,000 conventional magnetic tape reels online—that is, directly retrievable by the computer system. The recording medium is usually removable and can be stored offline, requiring a fraction of the space of a magnetic-tape library. The cost of the recording medium itself drops to around .00005¢/byte. The installation of at least one archival storage device was justified by its elimination of the cost of purchasing, and the space requirement for storing, thousands of reels of magnetic tape. More importantly, by having all of the storage online, slow and error-prone manual handling of magnetic tapes can be eliminated.

An 8½″ by 11″ document, again assuming 4,000 characters of text, could be stored in computerized form for 4¢ online or .2¢ offline. Thus, it may be cheaper to store information in a computer than on paper! The implications for effective use of archival storage units are not yet fully understood. Experiments in the future uses of archival storages are just beginning, such as the DATACOMPUTER and TABLON projects. This is an area that can have a tremendous impact upon society.

The capacities, speeds, and prices cited are already available on the market. In the next five years, we can expect significant advances, especially in laser and electron beam approaches similar to the UNICOM and PDS units.

☐ **Other Trends.** System reliability will increase due to extensive use of electronic circuits, error-checking and -correcting techniques, and economical redundancy—the use of duplicate elements either to serve as "spares" or to check the correctness of each other. The marriage between computers (and their offspring—terminals) and communications will intensify. Computers are already being used to control communications in A.T.&T.'s Electronic Switching System (E.S.S.). Digital communication, as contrasted with voice communication, is increasing rapidly. This area of tremendous potential is complicated by many factors, including technology, Federal Communications Commission regulations, relatively inexperienced and rapidly growing competition, and A.T.&T.

Computer System Architecture

The computers of 25 years ago were high-speed calculators used to generate ballistic trajectories. Systems are now being used for purposes undreamed of then, yet the basic computer structures have not changed much. As a recent conference speaker stated, "After 25 years of growth, the computer industry has reached its infancy." Research during the past decade is about to pay off in new and more effective approaches to computer architecture.

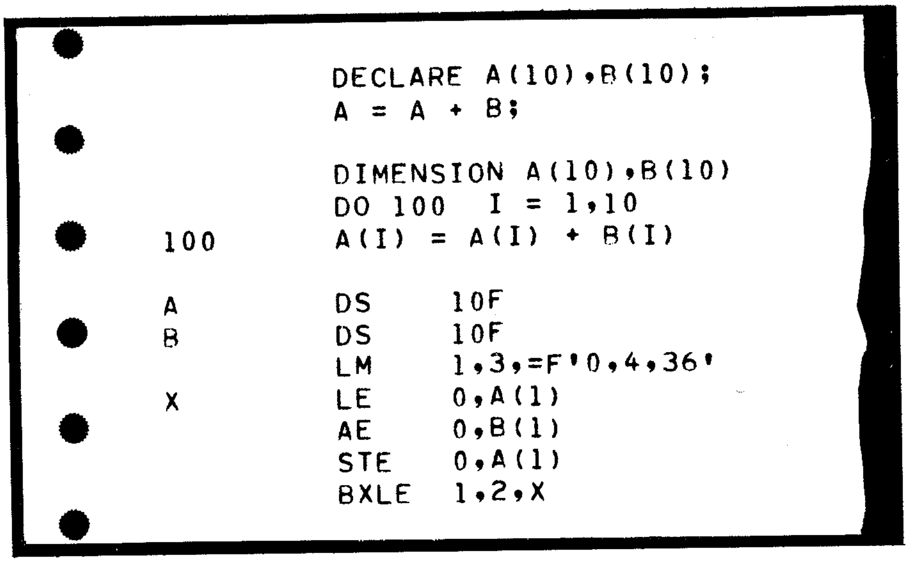
☐ **Multiprogramming,** the interleaved execution of two or more programs, is standard on most medium and large-scale systems. But the procedures presently needed for multiprogramming are often awkward; they require a large, sophisticated operating system, and frequently introduce considerable performance overhead. By analyzing the fundamental requirements for multiprogram operation and incorporating these features into the computer hardware, operating systems can be made much simpler and more efficient. Rudimentary attempts to accomplish this can be seen in the old Honeywell 800 series and the recent Singer Ten System.

Far more significant approaches can be found in the VENUS Project at MITRE Corporation. In that system, many of the basic multiprogramming primatives, such as those required for synchronization—WAIT and SIGNAL—are provided in the hardware. This enables the VENUS hardware to know which computations are currently executable, and VENUS can automatically select which to run next. Similar experiments exist in the advanced development laboratories of most major computer manufacturers.

Many of these multiprogramming facilities will be compatible with existing software. But to attain even greater effectiveness, especially in the multiple microprocessor system discussed in a preceding section, it will be necessary to develop new programming styles. PL/I (Programming Language/One) provides some of the necessary features for parallelism and synchronization—EVENT variables and the WAIT statement—but other programming languages are needed and are being developed.

☐ **Microprogramming and Control Hierarchies.** The early computers, though voluminous, were relatively simple. They performed additions, subtractions, and comparisons. As users developed requirements for advanced mathematical processing (vector and matrix operations, for example), extensive non-numerical processing (such as character code conversion or information retrieval), and intricate problem-solving, far more sophisticated computers were desired. The microprogramming manufacturing technique makes it feasible and economical to produce such systems. There are many low-level hardware functions (for example, setting a bit, starting up the main memory, or moving information between internal registers) required to accomplish one "instruction" such as *ADD X* to *Y*. In a microprogrammed computer, these low-level functions are controlled by a "microprogram" that indicates the steps required to accomplish each instruction. These microprograms are much easier to change and design than it is to use conventional direct electronics to generate the control signals for the low-level functions. In the VENUS Project, many of the operating system functions (for example, synchronization, interlocks, memory mapping, and allocation) have been incorporated into the basic computer microprogram hardware (called "firmware"). This approach has also been used to greatly simplify and speed up the operation of high-level programming languages (we will discuss programming languages later in this article), such as COBOL, FORTRAN and PL/I on the Burroughs 1700 system, and APL on an experimental IBM system. This trend will continue on future systems, providing far more powerful and efficient programming facilities.

☐ **Virtual Storage and Storage Hierarchies.** IBM has recently popularized the concept of "virtual storage," the automatic management and movement of information between main storage and secondary storage. Similar approaches have been used in many earlier systems by other manufacturers, such as Burroughs and RCA (now UNIVAC). Virtual storage greatly simplifies the tasks of the programmer—the major cost in application development as well as in improving system performance. It does so by eliminating explicit constraints on program size caused by limitations of main storage capacity and by eliminating much of the explicit concern for "efficient" use of main storage. If information is needed, it is auto-



```
            DECLARE A(10),B(10);
            A = A + B;

            DIMENSION A(10),B(10)
            DO 100  I = 1,10
100         A(I) = A(I) + B(I)

A           DS     10F
B           DS     10F
            LM     1,3,=F'0,4,36'
X           LE     0,A(1)
            AE     0,B(1)
            STE    0,A(1)
            BXLE   1,2,X
```

A short lesson in computer programming to demonstrate the differences between high-level and low-level programming languages. Shown above are segments from three computer programs, each of which produce the same result: the addition of two vectors A and B, where each vector is a collection of ten quantities. The first segment is composed of two statements in the high-level language PL/1; the second consists of three statements in the high-level language FORTRAN; the last consists of seven statements in IBM 360 Assembler, a low-level language. (In all three segments, we disregard the matter of introducing the data into storage.) The DECLARE statement in PL/1, the DIMENSION statement in FORTRAN, and the DS (Designate Storage) statements in 360 Assembler all serve the same function: they reserve space in storage for two collections of ten quantities each, and name the beginnings of the storage locations A and B. In PL/1, only one additional statement is necessary to add A to B and store the result where A was stored. In FORTRAN, two statements are necessary; the computer is instructed to execute statement 100 ten times, each time incrementing the "counter" I by one, until all ten elements of A are added to the corresponding elements of B. In 360 Assembler, the programmer's work is considerably more complicated. The programmer must explicitly arrange for the movement of the elements of the vectors in turn from storage into the processor, must write a conditional branching instruction to perform a comparison between a counter and a constant (which he must provide to the processor) to check if all ten elements have been added, and have the computer "branch" back to the beginning if the computation is not complete. Computers can execute only instructions even simpler than those of Assembler language; high-level languages are translated into these basic statements by computer programs called compilers.

matically moved into main storage. If main storage becomes overcrowded, information is automatically moved to the slower and cheaper secondary storage devices. In addition to easing application development, these techniques often lead to improved overall system performance.

The effective use of the intermediate storage technologies we described earlier requires an automatically controlled storage hierarchy that provides a virtual storage encompassing main, intermediate, and secondary storage. Research in this approach is going on in development laboratories as well as universities, including M.I.T. The current problems will probably be resolved in time to allow the use of intermediate storage devices in storage hierarchies for the next generation of computer systems. The combined effect should further reduce programming costs while increasing system efficiency.

☐ **Remote Computing.** One of the most significant impacts of information processing systems will be in this area, which includes time-sharing, centralized data bases, and computer networks.

Time-sharing, the simultaneous access to, and shared use of, a centralized computer by users at remote terminals, has not met the lofty projections of its advocates during the mid 1960's, but it has, and will continue to have, tremendous effects. Most of the earlier technical problems have been long overcome and the reduced system cost makes time-sharing systems very attractive. Many people, "burnt" during the expensive time-sharing fever of the 1960s, are surprised to find that powerful time-sharing systems are commercially available for less than $20,000 a month (such as the VM/370 operating system on the IBM System/370 Models 135 or 145). More limited time-sharing systems, such as those utilizing minicomputers and restricted to the BASIC programming language, are available for a fraction of IBM's price. At the other extreme, powerful "computer utilities" capable of handling hundreds of simultaneous users have finally reached the marketplace. These include the new Honeywell H6180 Multics (Multiplexed Information and Computing System) which was developed in conjunction with M.I.T.'s Project MAC.

The full impact of time-sharing systems has been stalled by the lack of entrepreneurial efforts in application areas. We are now beginning to see the emergence of companies that use the time-sharing concept to provide useful and convenient facilities directly to the end-user—the user of an application-oriented program, who is usually not a programmer. Typical application areas include online advertising media analysis, engineering/manufacturing/production control systems, and sophisticated lens-design programs. Many of the larger time-sharing services companies have already found that most of their revenue is derived from the proprietary application program services rather than from their traditional "raw" computer services.

As the complexity of modern-day business increases, it is necessary to place increased reliance upon computer-assisted controls. New information-handling concepts coupled with the economics of secondary and archival storage devices make centralized data accessed through remote terminals feasible. Many of the earlier disasters of "online real-time total management information systems" can be attributed to the naivety of the users and implementors rather than interpreted as an indictment of the basic concepts.

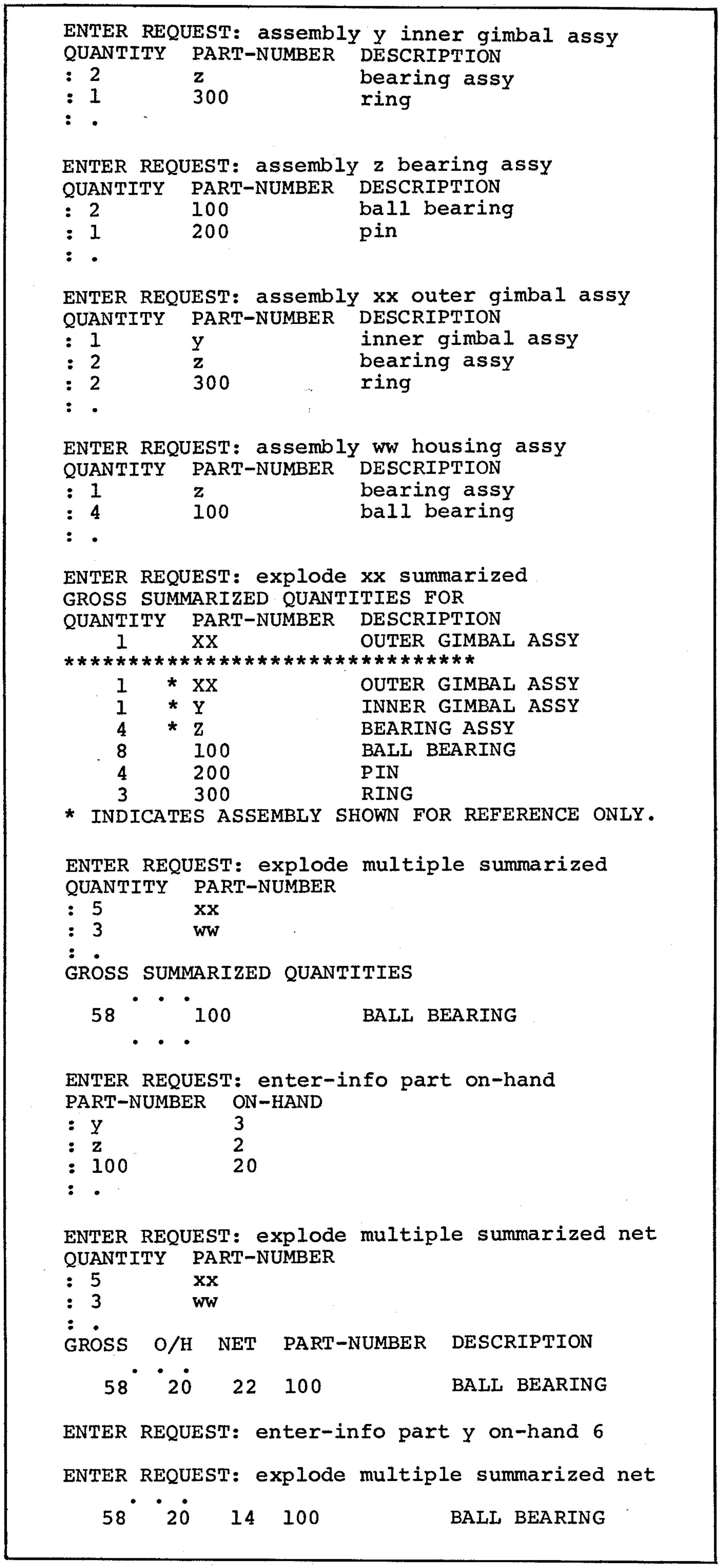
The developing need for more global optimization of large systems, used by such concerns as a decentralized manufacturing company or perhaps the Federal Reserve System, makes it necessary for local computer systems to communicate and exchange information. Advances in this area are being made by projects such as the government-funded ARPANET and Michigan's MERIT system. Commercial versions of these systems are already appearing on the market (for example Bolt, Beranek and Newman's Interface Message Processor systems originally developed for ARPANET).

☐ **Protection.** The topics of information system protection and security have received considerable attention in the press, but their full implications are probably not yet apparent to most observers. It is unlikely that anyone would make a serious attempt to steal your company's payroll program or even the customer list—although it is possible. On the other hand, consider a multi-million dollar software development company whose major assets—proprietary application programs—may be represented by a single reel of magnetic tape. The lack of effective technical and legal safeguards has been a major obstacle to the growth of the application-oriented time-sharing services market. Fortunately, many, though not all, of the problems have been solved.

It is useful to divide the problem of protecting the information in a computer system into three parts: *validation* (How is bad information kept out of the system?); *integrity* (How is destruction or loss of information prevented?); and *security* (How is unauthorized access to the information prevented?).

It makes little sense to lock your information system in a lead vault guarded by Marines if the information is meaningless or incorrect. In a

A simple example (that manages to become fairly involved) of an information system to be used by a manufacturer at a remote computer terminal. Lower-case typing is by the user; upper-case typing is by the computer; "ENTER REQUEST:" initiates each exchange of information. The user has begun by listing the component parts of an inner gimbal assembly, a bearing assembly, an outer gimbal assembly, and a housing assembly. This information is stored by the computer. The user then requests a summary of the parts required for an outer gimbal assembly ("explode xx summarized"), and the number of ball bearings required for a production run of five outer gimbal assemblies and three housing assemblies ("explode multiple summarized"). Finally, the user enters the quantities of assemblies now in stock ("enter-info part on-hand"), and wishes to know, first, how many ball bearings are required for a given production run ("explode multiple summarized net"), and then how many are required if it were six rather than three inner gimbal assemblies (part y) in stock. (Source: MITROL, Inc.)



```
ENTER REQUEST: assembly y inner gimbal assy
QUANTITY  PART-NUMBER  DESCRIPTION
: 2       z            bearing assy
: 1       300          ring
: .

ENTER REQUEST: assembly z bearing assy
QUANTITY  PART-NUMBER  DESCRIPTION
: 2       100          ball bearing
: 1       200          pin
: .

ENTER REQUEST: assembly xx outer gimbal assy
QUANTITY  PART-NUMBER  DESCRIPTION
: 1       y            inner gimbal assy
: 2       z            bearing assy
: 2       300          ring
: .

ENTER REQUEST: assembly ww housing assy
QUANTITY  PART-NUMBER  DESCRIPTION
: 1       z            bearing assy
: 4       100          ball bearing
: .

ENTER REQUEST: explode xx summarized
GROSS SUMMARIZED QUANTITIES FOR
QUANTITY  PART-NUMBER  DESCRIPTION
    1     XX           OUTER GIMBAL ASSY
********************************
    1   * XX           OUTER GIMBAL ASSY
    1   * Y            INNER GIMBAL ASSY
    4   * Z            BEARING ASSY
    8     100          BALL BEARING
    4     200          PIN
    3     300          RING
* INDICATES ASSEMBLY SHOWN FOR REFERENCE ONLY.

ENTER REQUEST: explode multiple summarized
QUANTITY  PART-NUMBER
: 5       xx
: 3       ww
: .
GROSS SUMMARIZED QUANTITIES
     . . .
  58      100          BALL BEARING
     . . .

ENTER REQUEST: enter-info part on-hand
PART-NUMBER  ON-HAND
: y          3
: z          2
: 100        20
: .

ENTER REQUEST: explode multiple summarized net
QUANTITY  PART-NUMBER
: 5       xx
: 3       ww
: .
GROSS  O/H  NET  PART-NUMBER  DESCRIPTION
      . . .
   58   20   22  100          BALL BEARING

ENTER REQUEST: enter-info part y on-hand 6

ENTER REQUEST: explode multiple summarized net
      . . .
   58   20   14  100          BALL BEARING
```

New Jersey town preparing the data for computing the tax rate, a value of $10,000,000 for one resident's house was incorrectly placed in computer storage. (A decimal point was misplaced or a keypunch operator was sleepy.) After the tax rate was computed, the mayor was pleased to announce that the town had one of the smallest rate increases in the state. Months later, an irate resident complained of a $10,000,000 property assessment on his 8-room house. At last report, the mayor was looking at a sizable budget deficit.

Techniques have been developed to prevent the loss of information stored in a system. These include "backup" copies that are automatically generated (for example, by the Multics backup system) and duplicate copies that are automatically maintained.

Security, and the related issues of privacy, have received considerable attention. Modern computer systems provide facilities, such as "supervisor" and "user" modes of operation, to help enforce security. Due to the complex interdependencies of information and users, it has been difficult to develop security techniques that are exactly sufficient for their intended purpose and yet do not impose unnecessary constraints on system use. M.I.T.'s Sloan School Information Systems Security Project is investigating these problems as part of IBM's 5-year $40 million joint research effort in conjunction with university, industry, and government investigators.

```
Pick up a big red block.
     OK.

Find a block which is taller than the one
you are holding and put it into the box.
     BY "IT", I ASSUME YOU MEAN THE BLOCK
     WHICH IS TALLER THAN THE ONE I AM
     HOLDING.
     OK.

What does the box contain?
     THE BLUE PYRAMID AND THE BLUE BLOCK.

Will you please stack up both of the red blocks
and either a green cube or a pyramid?
     OK.

Which cube is sitting on the table?
     THE LARGE GREEN ONE WHICH SUPPORTS THE
     RED PYRAMID.

Is there a large block behind a pyramid?
     YES, THREE OF THEM: A LARGE RED ONE, A
     LARGE GREEN ONE AND THE BLUE ONE.

Put a small one onto the green cube which
supports a pyramid.
     OK.

Put the littlest pyramid on top of it.
     OK.

Does the shortest thing the tallest pyramid's
support supports support anything green?
     YES, THE GREEN PYRAMID.

Had you touched any pyramid before you put the
green one on the little cube?
     YES, THE GREEN ONE.

When did you pick it up?
     WHILE I WAS STACKING UP THE RED CUBE, THE
     LARGE RED BLOCK AND A LARGE GREEN CUBE.

Why?
     TO GET RID OF IT.

Why did you do that?
     TO CLEAN OFF THE RED CUBE.

Why did you clear off that cube?
     TO PUT IT ON A LARGE GREEN CUBE.

Why did you do that?
     TO STACK UP THE RED CUBE, A LARGE RED
     BLOCK AND A LARGE GREEN CUBE.

Why did you do that?
     BECAUSE YOU ASKED ME TO.
```

Systems that are Easier to Use

We have thus far presented advances that are driven primarily by technological innovation in hardware manufacturing and system architecture. It is reasonable to ask, "What about the user?" In addition to the unselfish desire to make life more pleasant for users, there are many important dollars-and-cents reasons that are accelerating manufacturers' and researchers' efforts. If the prices of hardware drop as anticipated, it will be possible to do next year's processing at less cost than this year's. One estimator claims that the current world-wide inventory of processors and memories could be replaced at a cost of less than $1 billion by 1975. Thus, if there is not continual and massive growth of the market, the industry will stagnate—and its dollar revenues will shrink tremendously.

All these gloomy statements have potentially applied over the past 25 years. Fortunately, the market has always grown much faster than prices could drop, and a demand has developed for larger and more powerful systems. When the early ENIAC computer was built, reliable experts predicted that 100 such machines would satisfy the country's computational needs for the rest of the century. Needless to say, the market was somewhat larger.

We will take the position that there is enormous potential market growth for computerized processing and information systems. But we can identify three major bottlenecks to growth:

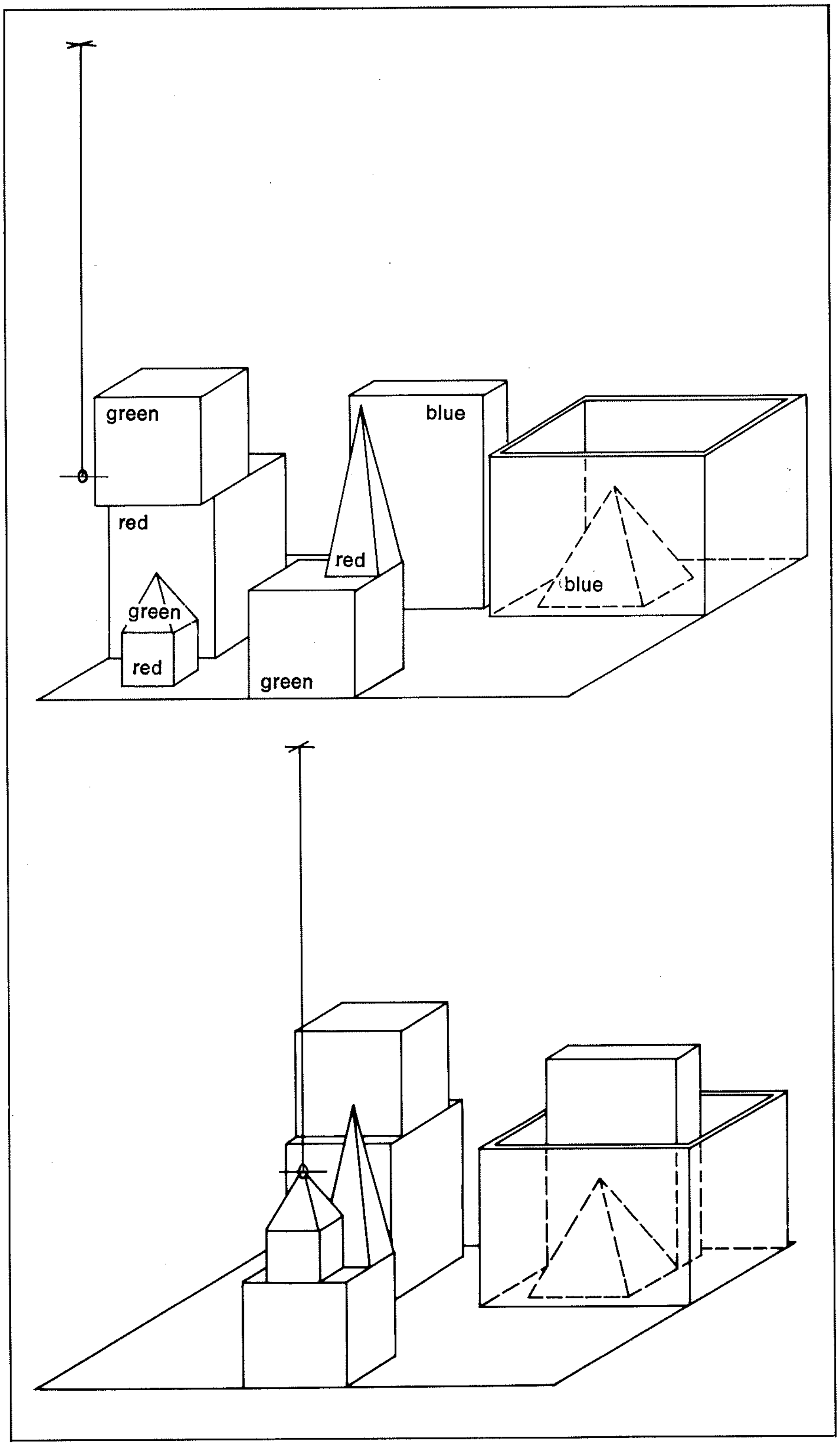
Salaries vs. Hardware. It is estimated that close to 70 per cent of the development and operation costs of new application areas is tied to people (including their salaries, office space, fringe benefits, etc.) and only about 30 per cent is tied to computer hardware. If the hardware costs were to drop to zero, there would still be relatively little increased incentive to develop applications at a faster pace.

Maintenance vs. Development. In most mature data processing installations, about 80 to 90 per cent of the personnel and costs are devoted to the operation and maintenance of existing applications. This leaves only about 10 to 20 per cent of the budget for the development of new application areas.

User Sophistication and Education. The first two bottlenecks relate primarily to the market of current, relatively mature users. An even larger market is found in present nonusers, frequently representing small and unsophisticated concerns. The third bottleneck is tied to making systems usable by the uneducated non-user.

In this section we will discuss approaches and techniques, in existence or being studied, that attack these bottlenecks.

☐ **High-Level Languages and Problem-Oriented Languages.** A "machine language"—the language in while each statement represents a single instruction to a computer—is awkward and tedious for human use in expressing a problem. Instead, the user expresses his problem in an "English-like" high-level language (H.L.L.), such as FORTRAN (**FOR**mula **TRAN**slator) or COBOL



At left, some sample dialog from the Blocks World System, created by Professor Terry Winograd at M.I.T.'s Artificial Intelligence Laboratory. The system includes a computer programmed to understand English-like language. Human commands and questions concerning a scene containing a table, a "hand," an open box, and an assortment of blocks and pyramids are in lower-case typing; the computer's responses are in upper-case typing. The system responds with an "OK" after executing an order. Examples of its processing: For the second request, the computer must deduce which block it is holding and determine the referents of the words "one" and "it." In the command to "put a small one onto the green cube," the system interprets "one" in light of the previous request to mean a block. Later, the meaning of "littlest," a word not yet in the computer's vocabulary, is surmised from the computer's knowledge of the word "little." The drawing above shows the initial and final states of the scene.

(COmmon Business Oriented Language), and the computer translates the problem into machine language, using a computer program called a compiler. The net effect is that it is easier and faster for the user to write computer programs.

The illustration on page 40 shows the amount of effort required to specify a vector addition using two H.L.L.s—PL/I and FORTRAN—and the "assembly language" (low-level language) for the IBM System/360 computer. The PL/I and FORTRAN forms are converted to the low-level form by the appropriate compilers.

A Problem-Oriented Language (P.O.L.) is a high-level language specialized for a particular application area. For example, the COGO (Co-ordinate Geometry) language is intended for use by civil engineers, and provides facilities for conveniently expressing arcs, angles, areas, and so on.

H.L.L.s and P.O.L.s have been in use for many years. But in the past, it has been expensive and difficult to implement compilers. Some H.L.L.s and P.O.L.s have not been much easier to use than machine language. Finally, the use of H.L.L.s and P.O.L.s often resulted in programs that were less efficient than programs manually translated into assembly language. These problems have been largely overcome, due to research resulting in techniques that produce economical, efficient, and more powerful compilers; the shift in cost from computer hardware to people, so that even if the compiler is inefficient, a manual translation would usually be much more expensive; and new computer architectures that provide for efficient operation of H.L.L.s and P.O.L.s—such as an experimental computer system, employing microprogramming that directly executes APL (A Programming Language) statements rather than assembly language instructions.

☐ **Generalized Application Packages.** While high-level languages make programming easier, they still require programmers. Yet every company seems to want a payroll program, an inventory control program, an accounts receivables program, and so on. The vast majority of current programmers are working on projects that have already been done at other companies.

The problem is usually not due to company secrecy. In fact, some companies have found that marketing their programs has brought in considerable extra revenue. The problem, in general, is that no two programs are exactly the same, even when they are written for similar purposes. The ubiquitous payroll program, for example, may differ due to handling of salaried vs. non-salaried employees, whether the company is intra-state or inter-state, pension plans, etc. Thus, one company's program may be worthless, or at best of minimal use, to another company.

Yet there is usually a manageable number of mutually exclusive options whose combinations and permutations result in the tremendous diversity within a given applications area, like payrolls. A generalized application program attacks this problem by providing the capability to handle all of the possible options. The user merely specifies which options he needs and what particular values must be used. This approach has been pushed quite far by the Applications Customizer facility used on IBM's small-scale System/3 computers and the Business Management System on the Burroughs B1700, B700 and L series.

Generalized application packages have made the computer usable by, and economical for, a large market—the small users. But developing a generalized application program is usually much more costly than developing any single non-generalized application program. Thus, only large companies with many customers can justify the initial cost. Furthermore, as computers move into areas with ever more options (such as production scheduling and market analysis) and totally new areas (such as medical applications), the cost of developing the generalized program increases and the market size decreases. For now, the growth of entrepreneurial software companies, in addition to the present computer manufacturers, will result in considerable activity in the generalized application package area.

☐ **Information Handling and Documentation.** The actual writing of computer programs is only part of the cost of developing new systems. The use of high-level languages has made some programmers so effective that it takes more people to explain how to use the software than it took to write it. The "documentation" of programs has become a serious bottleneck. While the use of "English-like" high-level languages that are easy to understand reduces the amount of additional documentation, some is still required. The truly self-explanatory program does not yet exist.

Just as compilers were developed to help in the programming and translation process, various tools have been developed to aid in the documentation process. Online manuscript processing systems, such as IBM's SCRIPT/370, make it convenient for the programmer to create and update system documentation without the additional cost and delays of requiring separate secretarial services. These tools, although originally developed by computer people for computer people, are finding increasing usefulness in any type of project that requires substantial amounts of documentation which is revised and updated over a long period of time.

☐ **Intelligent Information Management Systems.** The computational uses of computers are continually dropping in importance compared with information storage and processing. Decreasing costs coupled with the emergence of archival storage devices are accelerating this trend. Many companies made attempts during the 1960s to develop "online real-time total integrated management information systems." In many cases the entire concept and the planning were poorly handled. In other cases, the existing technology did not meet the requirements.

The illustration on page 41 shows the use of an online information processing system for manufacturing companies. Everything to the right of a colon and in lower case was entered by a user at a remote terminal; all other printing (in upper case) represents responses typed by the computer. The user can describe, for permanent storage, how his manufacturing assemblies are put together. Once the information has been entered, questions can be asked—fairly straight-forward questions, but ones that are deceptively difficult to unravel in companies with thousands of parts and hundreds of products. By exploiting modern information management technology, good "human engineering" (ease of use) and the time-sharing concept, powerful systems are now available to manufacturing companies that couldn't afford or didn't want in-house computer facilities.

Consider an example of a company's computerized personnel files. If the names of each employee's parents are provided, it should be possible to inquire how many father/son pairs are currently employed by the company. In most conventional systems, such a query could not be made unless the system were modified to handle it. This example can be extended further by considering the question of how many grandfather/grandson pairs are currently employed. Note that the grandfather-and-grandson information may not be explicity stored in the data base. But by using the information on parents and children of each employee, the grandfather/grandson pairs can be identified. Systems that are automatically able to make these discoveries are often termed "intelligent" data base systems. There is considerable activity in this area in industry and universities, including M.I.T.'s Sloan School of Management. These intelligent data bases will greatly enlarge the present information systems market for both large and small users.

☐ **Artificial Intelligence and Automatic Programming.** The term "artificial intelligence" often brings to mind robots, chess-playing computers, and other far-out sounding concepts. In recent years, the field of artificial intelligence has developed many concrete results. Two significant advances have been goal-directed programming and natural (English) language capabilities.

Goal-directed systems, such as M.I.T.'s PLANNER and CONNIVER, differ from conventional programming techniques. The user expresses, in reasonably precise terms, *what* he wants done rather than the *how*-he-wants-it-done required by conventional programming. This approach makes it easy and convenient to build larger and more complex systems and puts much of the mechanical problem-solving burden on the computer rather than on people.

The example presented on pages 42-43 illustrates the use of goal-directed techniques as well as recent results in natural language understanding. This work was performed at the M.I.T. Artificial Intelligence Laboratory. At the right is a collection of blocks as seen by a mechanical eye. These blocks can be manipulated by a mechanical hand. At the left is a dialogue between human (lower case typing) and computer (upper case typing). The "Blocks World" system allows the user to converse about children's blocks. Knowledge about concepts such as "on top of" and "supports" is incorporated into the system. The goal-directed approach can be seen at the end of the dialog as the system explains how and why it took certain actions. The final position of all the blocks is shown in at the lower right.

The Blocks World system, although representing tremendous progress in artificial intelligence and natural language understanding, is still largely a "toy" system. Researchers are exploring ways to extend these techniques toward solving many of the programming development bottlenecks. By combining the application-specific knowlege of generalized packages, the power of flexible information management systems, and the goal-directed approach, it may be possible to communicate with the computer in English and have it directly produce programs without the need for "programming" as it currently exists. The Advanced Research Projects Agency (A.R.P.A.) of the Department of Defense is funding research in such areas, under the heading of "automatic programming," at several universities, including M.I.T.'s Project MAC Automatic Programming Division. A similar effort exists at the University of Michigan.

Control by the Users

We have attempted to identify significant trends in computer system development that are likely to have substantial impact upon the ways that computers are used.

A point to consider is that realizing the full potential of these advances depends on making dramatic changes in information system usage and structure. In many cases, these changes are more under the control of information system users than manufacturers. An informed user community is an important requirement for future progress.

Suggested Readings

Microprocessors

Lapidus, Gerald, "MOS/LSI launches the low-cost processor," *IEEE Spectrum,* (Nov. 1972), 33-40.

Rhea, John, "Processor on a chip—Will it sell?, *Electronic News,* (Sept. 25, 1972).

Memory

G. A. Saxton & Co., Inc., "Computer Memory Technology," *Industry Note No. 12,* (March 22, 1971).

Ayling, J. K., " Monolithic Main Memory is Taking off," *1971 IEEE International Convention Digest,* (March 1971), 70-71.

Howard, Harry, "Memories: Modern Day 'Musical Chairs'," *EDN/EEE,* (August 15, 1971), 23-31.

Thompson, Steve, Jack Morton and Andrew Bobeck, "Memories: Future Storage Techniques," *The Electronic Engineer,* (August 1971), 33-39.

Bobeck, Andrew H., and H. E. D. Sconil, "Magnetic Bubbles," *Scientific American,* (June 1971), 78-90.

Fields, Stephen, "Silicon Disk Memories Beat Drums," *Electronics,* (May 24, 1971), 85-86.

Madnick, Stuart, "Storage Hierarchies," *M.I.T. Project MAC Technical Report,* (1973).

Gilder, Jules, "3 Years After Birth, CCDs Head for First Commercial Applications," *Electronic Design,* (January 4, 1973), 36-40.

Archival Storage

Dell, Harold R. "Design of a High Density Optical Mass Memory System," *Computer Design,* (August 1971), 49-53.

Gentile, Richard B., and Joseph R. Lucas, Jr., "The TABLON Mass Storage Network," *SJCC 38,* (1971), 345-356.

Haines, J. B., "The Evolution of the MASS-TAPE System," *1972 IEEE International Convention Digest,* (March 1972), 140-141.

Penny, Samuel J., Robert Fink, and Margaret Alston-Garnjost, "Design of a Very Large Storage System," *FJCC/37,* (1970), 45-51.

Microprogramming

Foster, Caxton, "The Next Three Generations," *Computer,* (March 1972), 39-42.

Remote Computing

Roberts, L. C., and B. D. Wessler, "Computer Network Development to Achieve Resource Sharing," *SJCC,* (1970), 543-549.

Multiprogramming

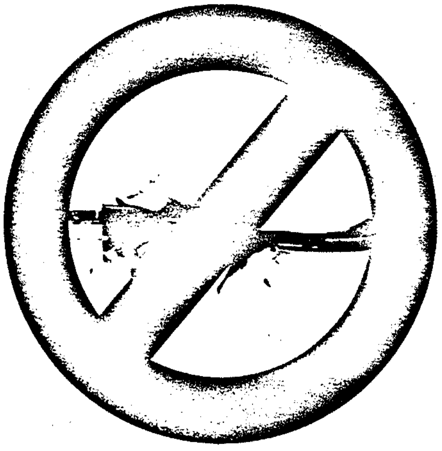
Liskov, B. H., "The VENUS Multiprogramming System," MITRE Technical Report 2004 (August, 1970).

Artificial Intelligence

Winograd, Terry, "Procedures as a Representation for Data in a Computer Program for Understanding Natural Language," Project MAC TR-84, (February, 1971).

Winograd, Terry, *Understanding Natural Languages,* Academic Press, 1972.

The First Line



Energy Crisis—Short and Long

The issue of energy supplies for an increasingly industrial and populous world has many dimensions, in which today's "energy crisis" has a very short time frame. In this issue of *Technology Review* we bring you reports on two new sources of conventional energy fuels whose time frames may be described as "intermediate"; their major contributions to U.S. energy supplies cannot come much before the 1980s. The point is important: few technological "fixes" are available on a time frame shorter than this.

Next month the *Review*'s focus will be on energy conservation—at once a short-term and a long-term issue. As our authors will reveal, some more or less superficial and surely modest contributions to resolving our energy shortage can be made very quickly through conservation. More significant energy savings will take longer; but, our authors will contend, they need not bring with them massive changes in our way and standard of life.

Science in Society

Belated, we record with pride that Victor Cohn of the *Washington Post*, whose regular contributions to the *Review* continue in the current volume (he is also an esteemed member of our Advisory Board), has won the first Science-in-Society Award for life science reporting from the National Association of Science Writers. Mr. Cohn's prize was for a series on sickle-cell anemia published in the *Post*; that was also Mr. Cohn's subject in *Technology Review* for January, 1973 (*pp.* 4-5).

Index to Volume 73

The long-awaited index to issues of *Technology Review* published in 1970-71 is now available. Copies are being mailed to all libraries subscribing to the *Review* and to individual subscribers who have recorded with us their wish for an index. Others should now make requests. There is no charge.

Indexes to Volumes 72, 74, and 75 are in preparation.

Letters

The Key Ecological Unknown

In his first contribution to Technology Review ("Ecology: Hard Push on a 'Soft' Science," October/November, pp. 16-17), Ian C. T. Nisbet skirts around but does not nail down a few political realities. Any assessment of environmental impact—to be meaningful—must necessarily be built upon assumptions as to public attitudes as well as on technical findings. Our benchmarks for clean air, clean water, or clean streets have less to do with a definable "degree above zero" than with what various members of the public consider acceptable or attainable. No matter how accurately the probability and magnitude of, say, certain fish kills are quantified, the *impact* of such kills is essentially subjective, loaded with emotional factors not amenable to cost-value analysis. It is the impact of a particular set of findings or predictions on the minds of men—not the impact of the pollutant on the environment *per se*—that is our key unknown.

Thus it would appear that those responsible for environmental impact studies really should begin with attitude measurement in the affected communities. This might well call for a scientific opinion profile of a representative sample of citizenry to determine their levels of concern for changes in the environment, good or bad, from the standpoints of health, economics, recreation, and esthetics.

Only with such a public opinion study at hand can the significant impact of probable environmental changes be forecast. A seemingly major problem of sewage disposal may be whitewashed, while a minor degradation which runs counter to a certain nostalgic chord may blow the lid off. Without such an analysis to set a baseline for an impact study, the findings of scientists and engineers will never satisfy the ecologists, the anti-ecologists, or the folks who pay the bills.

E. Scott Pattison
Dunedin, Fla.

Never Dissemble an Assembly

I was unable to verify the information system example given by Stuart E. Madnick (see "The Future of Computers") on p. 41 of the July/August issue. On the last line of the example I came with a net of 10 (when part y on-hand is changed to 6). Since part y contains 4 of part 100 it is easy to see that increasing y by 3 will result in a reduction of 12 for the net required of part 100.

The quantities given for a y of 3 and z of 2 check out. The last answer would also have checked out if z had been reduced to 0.

Could you please explain.

Ralph H. Evans
Alexandria, Va.

Professor Madnick explains:

The problem that Ralph Evans noted is very common. I have on several occasions seen professional auditors and inventory control experts somewhat confused or befuddled by this phenomenon. The key concept is, of course, that you never dissemble an assembly in order to get the parts out of it. This means, for example, that you would never take a part z bearing assembly, which is assembled and on the shelf, and tear it apart to extract the two part-100 ball bearings that are inside of it. This assumption, besides being fairly obvious, is, in fact, generally accepted in the manufacturing industry. With this thought in mind the problem that is noted by Ralph Evans becomes apparent. When we had three ys on hand, we needed 22 part-100s. When we increased the number of ys by three, that logically adds 12 more part-100s. Based upon the reasoning of Ralph Evans, and most other people, it would seem that the requirements for part 100 dropped to 10—that is, 22 minus 12 = 10. But this assumption is only true if we need to use all 3 ys or were willing to cannibalize the ys. The latter case of cannibalization is excluded. If one goes back and examines the original requirements needed to build the 5xxs and 3vys as stated, we don't need to use all of the ys. In fact, only five out of the six ys can be used for the production order without recourse to cannibalization. Thus, four of the part-100 ball bearings that are encompassed in a y cannot be used, and we must separately order or obtain four ball bearings. Thus the actual requirement for part-100 ball bearings becomes 14. It turns out that if one takes a step-by-step analysis of the actual requirements, this answer becomes obvious. The main problem occurs when people attempt to make shortcut, quickie solutions to the problem and do not consider the specific requirements that are needed at each level of the assembly process. This is one of the key things that we wanted to show in the example of an information system able to take these actual manufacturing requirements into consideration.

Nothing Disrespectful in the Occult

In "Status Comes to Occult Science" (*October/November, pp. 18-19*), Peter Gwynne states, "And in the eyes of the scientific community, it is only this type of experimentation that will give the study of the paranormal the respectability towards which it is now striving."

In my opinion, it all comes down to a matter of labels. Why call the field occult or paranormal? It is a study of normal functions which we have not as yet managed to understand. Mental telepathy is a perfectly normal function of the mind that is trained for it; there is nothing paranormal about it. The so-called aura is also just an emanation from living things; just because we do not quite understand it is no reason to find the study lacking in respectability.

If people are seriously trying to study an aspect of life that is not yet fully understood, why not give them all the encouragement and help possible? In other words, change the terminology, and let's push ahead into the most important territory of all.

Joan C. Westcott
Sarasota, Fla.