Effect of Technology on Near Term Computer Structures

Given certain components, hardware and software techniques, and user demands an accurate picture of computer development in the near future can be plotted.

by C. Gordon Bell, Robert Chen and Satish Rege

The development of computers has been influenced by three factors: the technology (i.e., the components from which we build); the hardware and software techniques we have learned to use; and the user (market). The improvements in technology seem to dominate in determining the possible resulting structures. Specifically, we can observe the evolution of four classes of computers:

1. The conventional medium and large-scale, general purpose computer (circa 1950). The price has remained relatively constant and the performance has increased, thereby increasing the effectiveness.
2. The minicomputer (circa 1965). The performance has been relatively constant, with only a factor of 10 increase from ~1960 to ~1970, and the price has decreased.
3. Very low cost, specialized digital systems,e.g., desk calculators (circa 1968). The basic technology cost has decreased to a price which makes mass production feasible.
4. New, very large structures based on a high degree of parallelism (circa 1971+). The packing density and the reliability of the technology has increased, thereby making large, parallel computer fabrication feasible. These highly specialized structures offer significant increase in the performance/cost ratio for certain, usually large problems.

The following sections will briefly discuss the evolution of computing structures in terms of the technology, and general techniques. Conventional computers and minicomputers will then be discussed as they represent two of the common computer structures. The next section will briefly present desk calculators and other mass production digital systems, and the final section will outline several computers which utilize some form of parallel computation.

Historical Background

The first generation vacuum tube technology (circa 1945 ~ 1960) computers were built to perform long, tedious arithmetic calculations. Because of their relatively poor cost/performance and high cost they were used mainly for calculations which would otherwise be impossible (e.g., in ballistic calculations). During this early period the standard of comparisons was desk calculator man years.

By the second generation, with transistor and better random access memory technology (circa 1960), the cost/performance had significantly improved. This made current computer applications (e.g., business and university computing) more feasible. The development of FORTRAN and other higher level languages also broadened the user base and provided demand for more computing power. User demands began to reach and overtake technology, and new techniques had to be adopted to raise performance levels beyond what the device technology provided. This led to concurrent use of input/output with program execution, which in turn led to more general multiprogramming.

This is a working paper and may not be quoted or reproduced without written permission.

This work was supported by the Advanced Research Projects Agency of the Office of the Secretary of Defense (F44620-70-C-0107) and is monitored by the Air Force Office of Scientific Research.

Although integrated circuit logic technology (circa 1965) marked the third generation, better access to computers, e.g., via typewriter-like terminals, marked the generation for the user (circa 1968). Remote access (e.g., general purpose time-sharing) came into being, with further demands for computing power, not only due to the continuously expanding user base but also due to the capabilities and ease of use required by the users.

The continual improvement of cost/performance also had an effect of broadening the user base further, so that process controllers were now built in larger quantities, and various other digital equipment evolved, replacing manual and analog controllers, and doing the task of laboratory and testing instruments. Today we see the continuation of these trends. Business and scientific centers are pressing for more and better access to computing power. Technological progress is still widening the user base. The introduction of the small, minimal computer (i.e., the minicomputer) has also widened this base, demonstrating that as prices decrease there is an ever increasing market. Especially significant in this trend is the fact that the price of these computers is small and decreasing.

General Effects of Technology On Computer Design and Computer Structures

We have so far introduced technological progress as a monolithic entity, uniform over all computer systems components. Actually, of course, this is not so. The progression from vacuum tubes to transistors to integrated circuits has had its greatest impact on processor cost and performance. In comparison to this, memory technology and peripherals (i.e., input-output, and secondary memory) has improved less, and as a result, the cost-sensitive component in the computer system has been gradually shifting from the processor. Peripherals are beginning to take on the significant portion of the system cost. In contrast to technology, system design costs have risen; this shift is demonstrated by, for instance, the decreased emphasis on minimization in logic design, but on the other hand, reliability, mass producibility and maintainability are now the important design criteria.

One of the consequences of the shift in costs is a pressure to utilize the processor cost reduction to better advantage by increasing the processor/memory and processor/peripherals ratios in computer systems. A manifestation of this is the research into intelligent terminals (e.g., Bell et al, 1971), microprogramming (Husson, 1970), and other ways to convert processor performance into overall system performance.

In the following sections we will look at four areas of computer design which have been influenced by technology. The first two, modularity, microprogramming, deal with the design time and the flexibility of a resulting structure. With these techniques, more hardware (gates) is usually required. The latter two, the cache memory and reliable computers, can be applied to affect the computer structure performance.

Modularity, an organization to reduce system design time. One interesting consequence of technology is a pressure to reduce design and overhead costs by standardizing components at a higher level than the traditional gate or chip level, and then to design systems using these larger, standard components. The present medium to large scale integrated circuits are organized around very general functions. Standardization at the register transfer level has been used in the design and construction of special purpose systems, e.g., the Macromodules system (Clark, 1966) and the DEC PDP-16 (Bell and Grason, 1971). Microprogramming also offers certain design flexibility at this level. Standardization of even larger computer components is occurring, e.g., the Burroughs microprogrammable multiprocessor computer (Davis and Zuckers, 1971), and the highly modular, one-buss based systems (e.g., PDP-11). All in all, with the exceptions of microprogramming and improved automated design, computers are designed in essentially the same fashion as always. For example, we observe that little has been tried in the way of real, experimental machines[1] because of the still significant difficulty of fabrication.

Microprogramming, an organization to "regularize" design. Microprogramming is a technique which involves hard-wiring a lower level interpreter, and then providing access so that the user level language drives an interpreter written in terms of the lower level microinstruction set. Thus, the definition of the (higher) user level instruction set is a program (or microprogram). With a continued emphasis of the technology to find "regular" ways of constructing machines, microprogramming is very important. Almost all but the smallest and largest processors are currently microprogrammed. The importance of microprogramming lies in the fact that it utilizes current memory technology and permits somewhat flexible and complex structures to be built from smaller primitives (the microinstruction set processor). In current implementations, the microprogram memories operate at a factor of four to ten times the speed of the primary (program) memory. The interest in "user" or "dynamic" microprogramming stems from both its ability to provide variation in the instruction-set and a general desire for higher performance by using a faster memory. Therefore, it has been proposed for and used in

1.We know of only one machine, the SYMBOL – circa 1963 ~ 1970, see Rice et al, 1971.

solving problems ranging from built-in diagnostics to operating systems to higher level language interpreters.

With all integrated circuit memories, the speed differential between primary memory and microprogram memory can disappear; hence programs and microprograms operate at nearly the same speed -- hence the advantages of a microprogrammed structure disappear. Although microprogramming will probably not continue to exist in its current form, highly regular logic will be used for implementing processors, control units, and more complex terminals.

The cache memory structure, to decrease memory access time. The cache memory (e.g., Conti, 1969 and Bell and Casasent, 1971) has evolved as a significant technique to reduce memory access time. With the cache structure, a small fast memory (which is organized to behave as a content addressable memory) is placed in the processor. For every memory reference, the processor first accesses the cache, and if data is present, the processor uses the data. If the data is not present, it is requested from a slower primary memory. There are several observations (see also Bell and Newell, 1971) regarding the cache structure:

1. It is predicated on a certain ratio of slow primary memory access time to fast processor logic. With faster primary memory will the cache still be worthwhile?
2. In a multiprocessor computer, cache memories reduce the loss in access time due to switching and multiple processor interference from a large, shared memory.
3. In cache-based computers (e.g., the 360 Model 85), the advantage of the microprogrammed structure disappears because simple conventional instructions and microinstructions are essentially identical and execute at the same rate.
4. The cache(s) can also be used to hold user microprograms.

Reliable structures to provide fault-tolerant computing. Another technological effect which can be observed is the increased emphasis on reliability. This emphasis is due, in part, to the fact that software is more reliable. Some pressure for very reliable computing has come from the aerospace industry which uses redundancy in both on-board and ground support computers. Various Electronic Switching Systems for Telephony have also used these structures. In fact, the *IEEE Transactions on Computers,* November 1971, has been devoted to fault-tolerant computing.

Reliability has not been neglected in standard computers which have: added memory bits which will correct single-errors, residue checking in the arithmetic registers, control sequences to retry instructions on failures, and built-in exercisors for checking various components. While many of the above checking features were present in early vacuum tube computers (e.g., Univac I), there is now renewed interest. The multiprocessor computers discussed in the last section provide another approach to achieving high reliability by redundancy using larger (processor) modules.

While the above section has examined certain techniques, the following sections will look at various structures.

Two Approaches to Better Cost-Effectiveness: The Minicomputer and the Conventional General Purpose Computer

With technological progress in cost/performance, two trends have become evident. The dividing line between the trends seems to be between the minicomputer and the medium-to-large scale general purpose computer.

Since their inception, minicomputers have more than retained their performance, and with improving technology, prices of minicomputers have consistently decreased. A paper on the trend of minicomputers (House and Henzel, 1971) shows the declining cost relationship – Figure 1 is taken from their paper.

The conventional general purpose computers have shown a contrasting trend, where machines have taken advantage of improved cost/performance by advancing to higher and higher performance instead of lower prices. Monthly rental for various equipment can be seen in Figure 2. Some coarse price categories (in \$K/month) are: 1~3.5, 3.5~10, 10~35, 35~100, >100. For instance, the transition within the IBM line for a nearly constant cost customer is: 650→704→7090→7094→360/65~75→370/165 and 1620→1130 and 1440→360/20. Such trends are demonstrated by observing that the equipment at various computing centers essentially remains at a constant or slightly increasing cost.

This interesting contrast is primarily due to the difference in application areas of these two classes of machines. The minicomputers have often been used in industrial process control applications. These uses have been generally looked on as limited applications to which the machine has to be fit; in short, the field is application-limited. Hence, little performance improvement has been desired, and the cost/performance improvement has been translated into cost reduction (and a larger market). On the other hand, for the conventional computer applications, there has generally been a desire for more computing power which cannot be

completely satisfied due to a lack of funds. Consequently, this field has been cost-limited, and any cost/performance improvement has been translated into performance improvement. Also, in these applications, an increase in performance has allowed users to attack more complex problems and use more powerful languages, each of which requires more performance.

It is interesting to speculate how long these contrasting trends will continue their separate ways, and whether new fields of applications will be opened up by the technology, and what the demands exerted by these new fields will be. Already, extended features in minicomputers (e.g., floating point arithmetic, caches, and segmented memories) tend to blur the distinction.

Although desk calculators will not be discussed in regard to future structures, a significant number are being produced and their designs are predicated on the same logic technology that is used for general purpose computers. An observation should be noted: the number of functions are increasing and desk calculators are beginning to compete with the minicomputers. In terms of

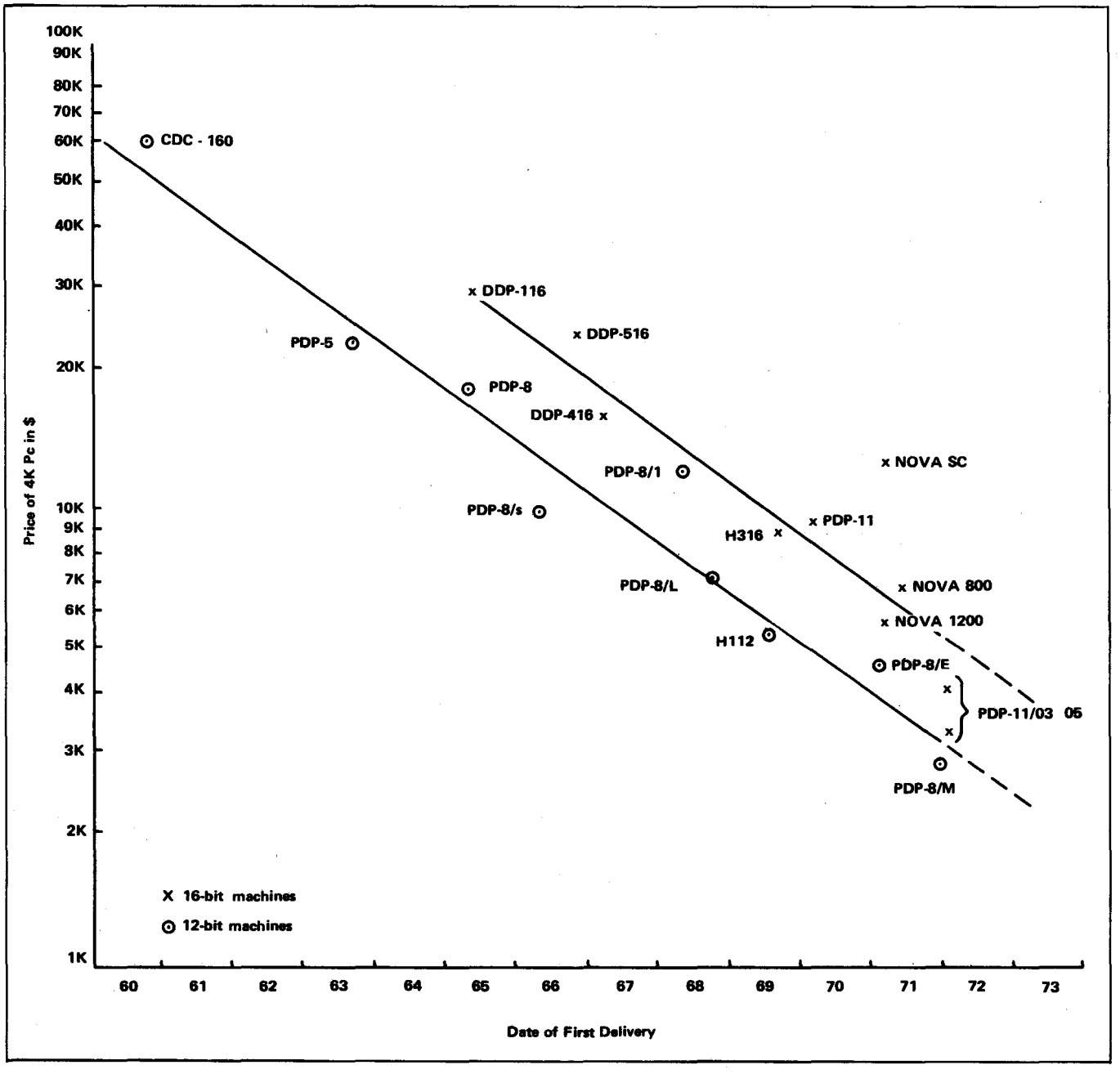


Fig. 1. Plot of cost of 12 and 16 bit machines, based on 4K Pc's, beginning in 1960 and extended through 1970. (Data taken from House and Henzel, 1971, courtesy of Computer Design Magazine.)

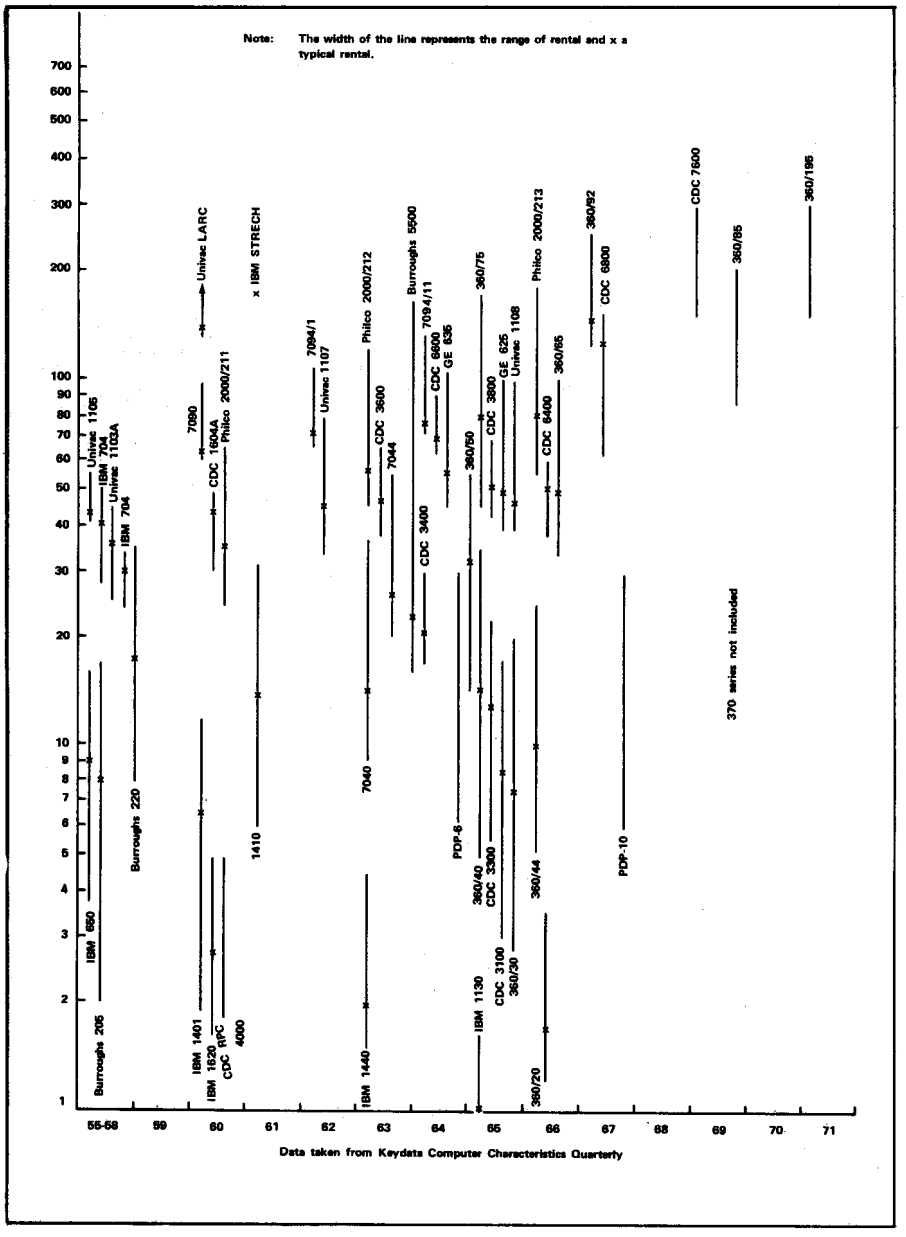


Fig. 2. Monthly rental vs. first delivery for various computers.

the minicomputer, high volume production is yet to come, but several "computer-on-a-chip" integrated circuit computers are being marketed (e.g., Intel); a 256 word, 8-bit memory and a 4-bit wide data path processor costs about $100. The design, like the desk calculator, is predicated on a very large market.

Recently the "smart terminal" has been discussed (e.g., Bell et al, 1971). This terminal, which includes a stored program computer, has the ability to be adapted to a range of applications from remote entry of display information to key punching. As yet, the "computer-in-the-home" has not been tried even on an experimental basis. Here, not only the technology is lacking, but also the techniques which would provide the "home" user an instrument that would carry out functions beyond that of an interesting oddity and status symbol.

Highly Parallel[2] *Computers*

This section will consider examples of computers that have been under consideration for a long time but have not achieved production status. The simplest explanation is that they could not have been built with prior technology, because of component density, cost, performance and reliability. Hence more will be built in the future. They all utilize some form of parallelism. The first such structure to be considered is the multiprocessor computer — a computer which has a number of processors sharing a common primary memory. Although two-processor systems have been in production, no significant number of multiprocessor systems exist with three or more processors. Although physically quite different, the computer network can be considered a similar structure. With the network, a number of computers are usually all operating on different tasks. The emphasis is to share facilities, programs and data.

Next, three computer structures, ILLIAC IV, CDC STAR, and Goodyear STARAN, are considered since they are designed to operate on elements of a string or an array in parallel. An operation is specified by a single instruction. Whereas ILLIAC IV is designed for numeric processing, STAR is designed also for string processing. Whereas STAR utilizes pipelines to simultaneously operate on multiple operands, ILLIAC IV operates in parallel on sixty-four operands specified by a single instruction, while STARAN is designed around a small, content-addressable memory (hence processing is almost within the memory).

---

2 Although the reader should recognize a distinction in the methods used to obtain parallelism, we make no such distinction in terminology. One possible meaning of parallelism is that it includes only structures which compute in a lockstep fashion on a number of elements of an array (ILLIAC IV, STAR and STARAN). Concurrency, or concurrent processing, would define computers which operate on many tasks in parallel; hence would include multiple processor computers and computer networks which break up and operate on one or more jobs in a less fixed fashion

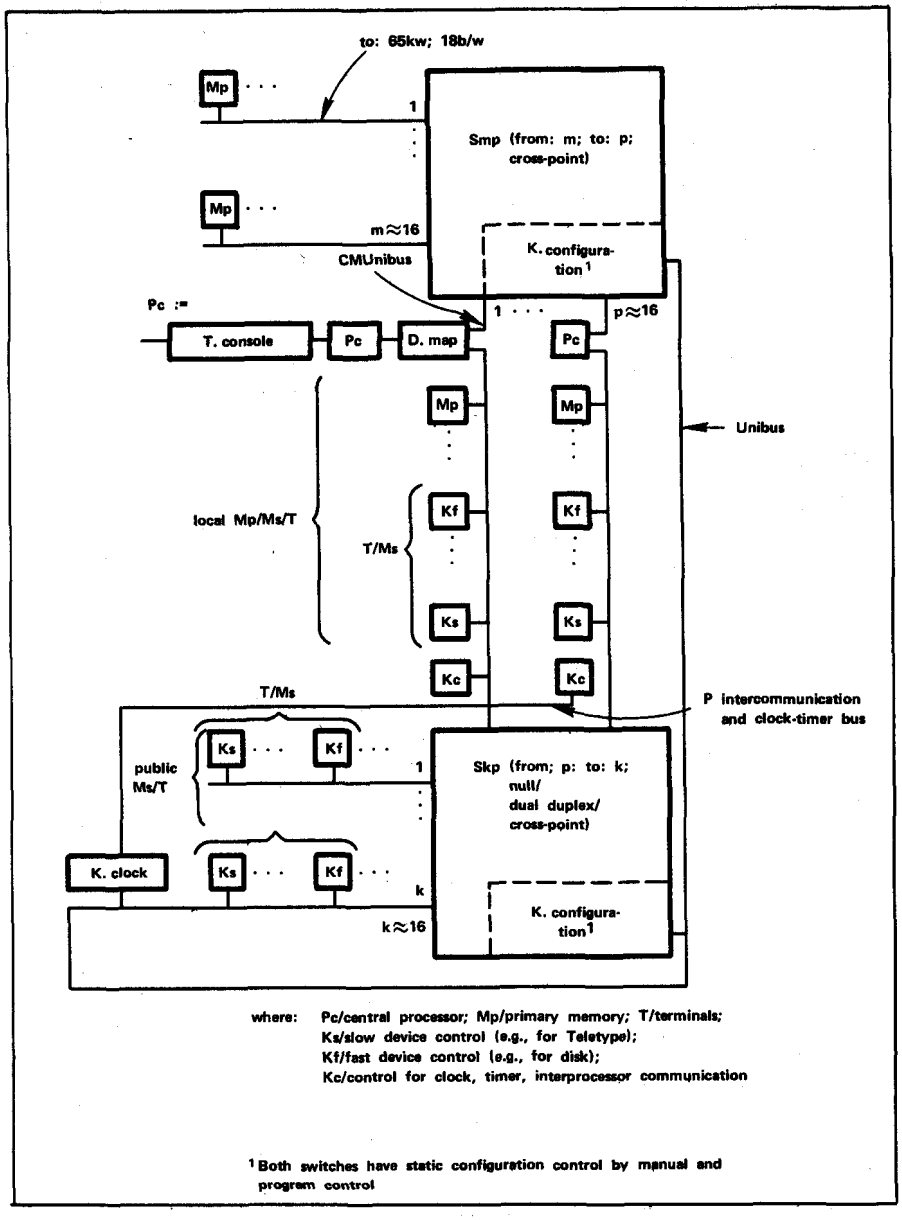


Fig. 3. Proposed CMU multiminiprocessor computer/C.mmp.

Parallelism by multiple processor computers – Carnegie-Mellon University C.mmp. One way to obtain more computing power is to simply couple a large number of processors to a shared primary memory and use them to execute one or more programs. This structure is perhaps the oldest, most straightforward method to achieve parallelism. Burroughs designed and implemented two processor systems using this concept in the B5000 (Lonergan, 1961), and in the larger D825 (Anderson, et al, 1962). There has been a resurgence of interest in these structures for several reasons: reduced processor costs compared with other systems costs — hence the processor cost effectiveness can be improved and maintained over a wide range of performance; higher density logic permits the necessary switching structures to be built; the structure has the potential for very high reliability since there are a large number of identical parts, none of which are critical; and the techniques for constructing multiprogrammed operating systems for uni-processors have evolved to the point where these systems can be considered.

There are several research projects for constructing systems of this form at the University of California (at both Berkeley and Irvine), at the University of Illinois, and a related project which is concerned with extreme reliability at the University of Newcastle-Upon-Tyne. The Carnegie-Mellon University (CMU) project, C.mmp (for "multi-mini-processor computer"), is based on PDP-11 processors, and its structure is shown in Figure 3. The performance[3] of the system versus the number of processors is shown in Figure 4; and the performance/cost ratio versus the number of processors is shown in Figure 5.

Carnegie-Mellon's C.mmp research project is based both on the need of computing power for the real time processing of speech input and on the need for a machine for research into multiprocessing. A structure of this type permits several types of usage (arranged in decreasing order of perceived difficulty) to be studied:

1. Parallel processing – A single job is broken into a number of independent tasks which can be processed in parallel. Multiple processors are assigned to these tasks. A significant problem exists on how to break a job into tasks. Some processing tasks may be organized this way. Also, this structure seems quite appropriate for discrete and continuous simulation.
2. Pipeline processing – A single job is broken into a number of independent tasks which are processed in parallel using a co-routine processing structure. Processors are assigned to the various co-routines; and each co-routine feeds results to successive processor routines. Real time processing and pattern recognition tasks are typical

3.Based on a calculation which assumes random references to the memory modules by all processors.

applications, e.g., speech and EKG signal processing. Also, compilation may take advantage of this type of processing.

3. Network processing – Specialized functions are assigned to the various subcomputers which may have several processors. Jobs are passed among the computers and computers may also be dedicated to filing and input-output. This organization is similar to that used in the large CDC computers.
4. Specialized hardware processors – as the above processing applications are extended, it may be desirable to add hardware for interpreting special languages (e.g., LISP) or for signal processing (e.g., FFT).
5. Conventional multiprogramming with multiprocessors – Multiple, independent programs are assigned to independent processors on a one-at-a-time basis.
6. Independent computing – The multiprocessor structure is partitioned, by hardware, to form independent uni- or multi-processors. The computers formed in this way are from a common, shared set of components.

It is perhaps most safe to predict that multiprocessor computers will come into widespread use in the next decade. The motivation for them is not only economic, based on desirable cost/performance ratios and performance functions over a range of cost[4] but also because of potentially high reliability.

Parallelism by multiple computer networks – the ARPA network. Another form of parallelism can exist by interconnecting a number of computers together as in the Advanced Research Projects Agency (ARPA) Network (Roberts, et al, 1970). This structure currently connects about 25 physically remotely separated computers. Minicomputers are used for store-and-forward message switching and for interface to the larger computers. The structure permits many of the styles of processing as discussed in the multiprocessor computer, C.mmp, but is not constrained to a single physical location. On the other hand, certain applications are not appropriate to this structure because of the delay times and limited bandwidth encountered in transmitting data among the computers. We believe a system of this form is probably the forerunner of many future computing and communications systems, simply because of the need to communicate data among machines.

With the computer population explosion, especially within a single organization, there is usually pressure to interconnect the computers for transmitting and sharing data for reliability and for better utilization of resources (e.g., files, hard copy input-output, and communications

4.The performance can be increased simply by adding processors, without any overall change in configuration.

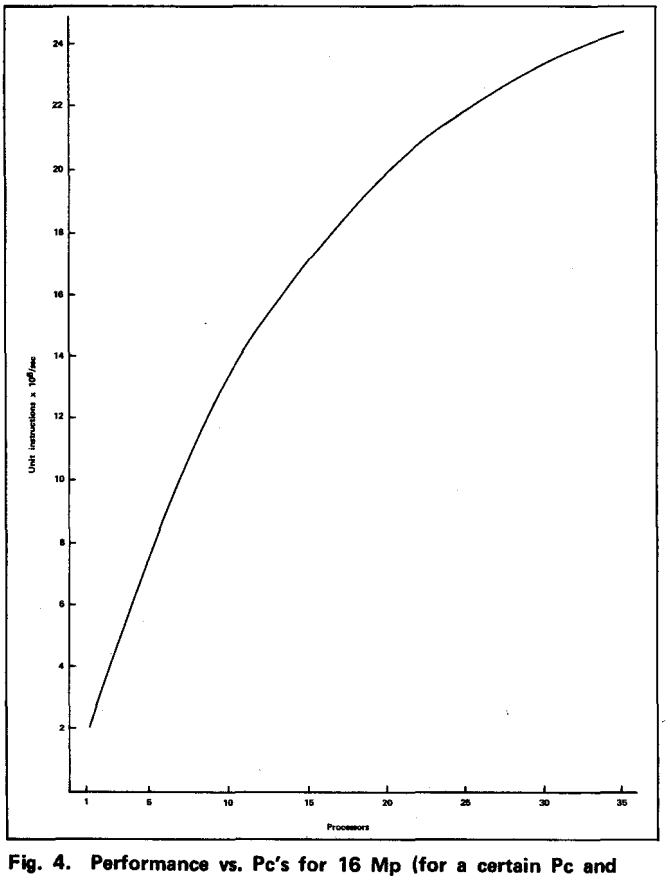


Fig. 4. Performance vs. Pc's for 16 Mp (for a certain Pc and Mp speed) for C.mmp.

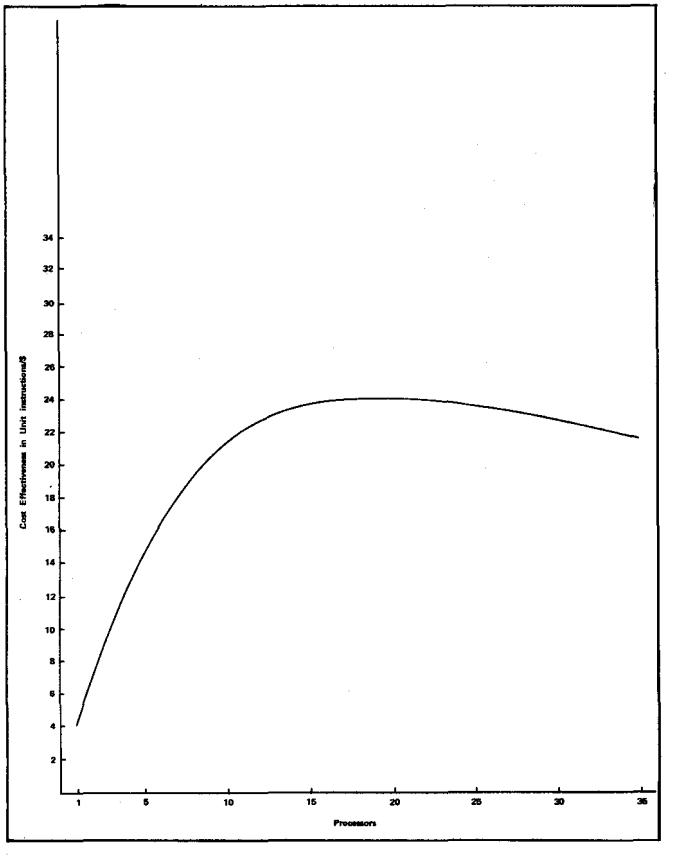


Fig. 5. Cost effectiveness (unit instructions/$) vs. Pc's for C. mmp.

equipment). The problem is especially acute with minicomputers, since the basic computer cost is low and the cost of files and hard copy input-output equipment is relatively high. Also certain tasks on the minicomputer are especially costly when operators have to manipulate paper tape. Although eventually a general structure like the ARPA network will probably evolve, a simple short term solution will help "balance" the minicomputer. The support facilities most minicomputers need include filing, printing and plotting, perhaps language translation, and sometimes computational assistance, particularly if either large problems or problems with "real" arithmetic are run. A larger timesharing computer, or single minicomputer, could provide the above services to perhaps 10 other minicomputers using medium speed (1200 ~ 9600 bit/sec) communication lines.

Parallelism by replication of multiple processing elements – ILLIAC IV. The most obvious way to achieve parallelism is with the University of Illinois' ILLIAC IV, shown in Figure 6. The structure was actually proposed and described as the Solomon Computer in 1962 (Slotnick, 1962)[5]. The current one-quadrant machine is one-fourth the proposed size. It has 64 Processing Elements (PE's) which operate on 64 elements of data simultaneously. All of the PE's are controlled by the central, common, Control Element; hence they all carry out the same operation at a given time. The utility of a structure of this type is predicated on array data such as encountered in linear programming, sets of partial differential equations, and phased array signal processing. For example, based on the specifications in the Barnes paper (1966), one quadrant of the machine can execute over 240 million add instructions per second. Fixed head disks read and write data in the individual PE memories. Programs are set up by a medium scale computer, the B6500, which accesses the PE's.

At this stage in ILLIAC IV's development it is hard to predict what its effect on future computer structures will be. Already there have been proposals for smaller scale, similar, specialized structures. If ILLIAC IV can be made to operate complete with a secondary memory system, a multiprogramming operating system, and programming languages, it might achieve the desired economics of scale; hence it could be directly replicated or repackaged to use large scale integration.

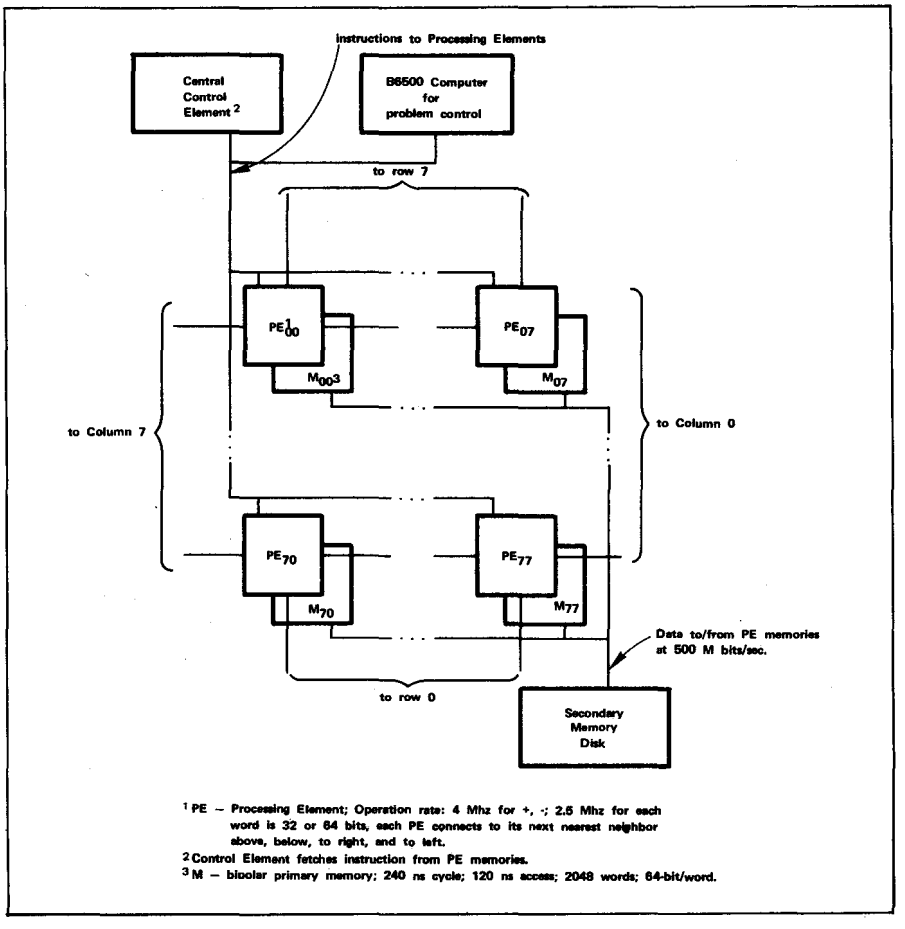
To illustrate the effects of technology on the structure, the designers (Barnes, et al, 1966) state, "It is only by virtue of high density integration (50- to 100-gate package)[6] that the design of a three-million-gate system can be contemplated."

Parallelism by pipeline processing – the CDC STAR. The Control Data STAR derives its name from the STring and ARray data it is designed to process (Holland and Purcell, 1971). In fact, the machine was directly influenced by APL (A Programming Language). The computer is actually a computer network consisting of nine computers which execute the operating system, handle the files and deal with the input-output equipment; and the very large central computer which handles the processing on the string and array (actually vector) data. The data is organized into vectors (as in ILLIAC IV) and each processing unit operates on the elements of the vector sequentially. The memory has a data rate of 12.8 gigabits/sec. The data is fed to the pipeline processing unit at up to 100 million 32-bit operands per

5.The current machine, ILLIAC IV, described by Barnes, et al (1966), has been under design since 1966 at the University of Illinois and Burroughs Corporation; the fabrication is being done by Burroughs; and the machine was scheduled for installation in early 1970.

6.Emitter Coupled Logic.

Fig. 6. One quadrant of ILLIAC IV Computer.



second. This would imply an internal clock rate of 100 megahertz, but since the clock rate is 25 megahertz 40 ns per clock), there are two, two-pair pipelines for floating point data which operate on alternate data. A simplified diagram of the structure of the central computer of STAR is given in Figure 7.

STAR derives its performance from an organization in which there are 4 x n data items being operated on in parallel in the four n-stage pipelines.[7] Comparing the STAR structure with that of ILLIAC IV: the STAR data types and operations are extensive; STAR has better utilization of data operation hardware (pipelines); and the control for the STAR pipelines is undoubtedly more complex (in terms of control steps) because of the higher execution rate per processing element. Assuming roughly the same performance for the two machines, ILLIAC IV requires 16 elements to perform the same operations that one-half of a STAR pipeline performs.

7.n varies with the operation, but there are about thirty, 40-ns stages for floating operations.

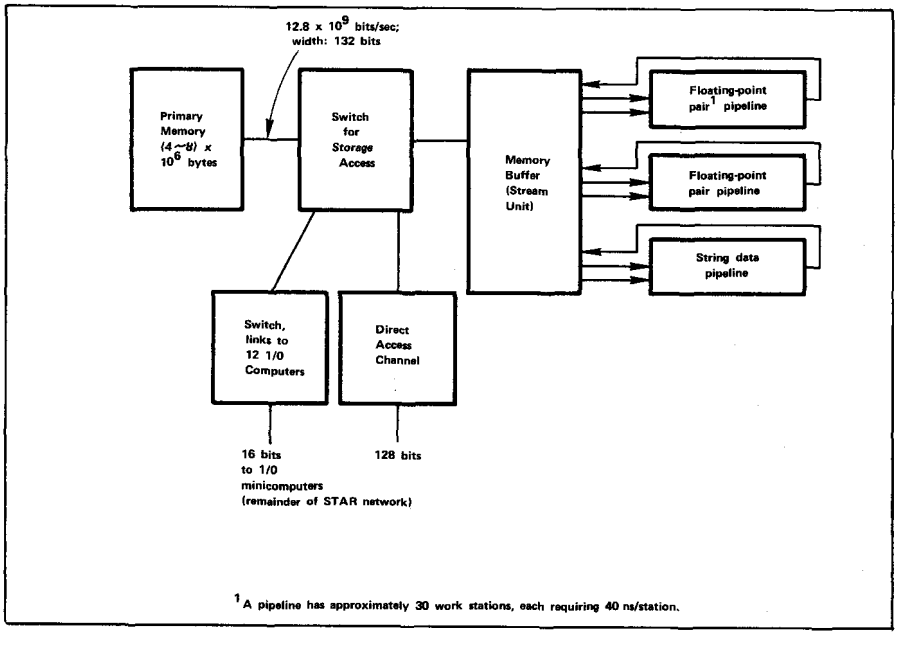


Fig. 7. CDC STAR Central Computer

STAR promises to be almost revolutionary in its abilities. It was influenced by a revolutionary language (APL); it is being implemented by a group with experience and success in building large scale computers; and its approach to a distributed operating system, which is currently operating, is also significant. All in all, STAR appears to be a very important future computer structure.

Parallelism by local processing within the memory – The Goodyear Aerospace STARAN. STARAN (Goodyear, 1970) is a computer based on a conventional, fast primary memory to hold a program, together with an associative memory. The computer is designed to operate as a special purpose computer connected to a host computer. For the last decade, associative memories have been discussed in the literature. A recent bibliography (Minker, 1971) lists several hundred articles. They have an amount of parallelism equal to the number of words in the memory.

STARAN programs are written to carry out operations on the content addressable memory. Operations include both integer comparison and integer operations on fields (e.g., add to all memory cells). The content addressable memory module is 256 words by 32 bits, and there can be up to 32 modules. The content addressable memory is a conventional word addressed memory organized (accessed) as both 32-bit words and 256-bit words. Comparing a register with all 256 words is carried out sequentially on each of the 32 bits at a rate of one bit per 0.15 μsec. Thus it takes 4.8 μsec to search a 256-word table. The effective processing rate is $\frac{256}{4.8}$ megahertz or 53.3 million[8] word search operations per second for a single memory module.

Here, again, the operation is similar to, but at a lower logic level than the ILLIAC IV computer. Parallelism is achieved by simultaneously operating on 256 operands.[8] For cost reasons, the operations are carried out sequentially, within a module, on operands.

The applications for STARAN (see Goodyear, 1971) include: radar signal processing to aid in aircraft tracking, pattern recognition, weather forecast processing, and data management.

STARAN can effectively utilize the technology which is emerging. It is a simple structure. What is as yet to be determined is: Are there enough potential users of a small, specialized memory structure, or will the structure have to wait for larger memories or lower cost memories? Can data be read in and out of the memory fast enough? What applications will now emerge because a memory like this is available?

Conclusions

We have tried to show how technology, together with techniques and users, has influenced computer structures. We believe the technology today is more exciting than in the past because it opens up more opportunities for the design and use of

8.There can be up to 32 times this parallelism by a configuration with multiple modules.

computers. We have shown some of the applications of this technology in stimulating low cost and highly parallel computing structures. There are other possibilities for providing both highly variable and highly reliable computers. There will also be more computers, and this will create a need for better overall organization (networks).

References

Anderson, J. P., S. A. Hoffman, J. Shifman and R. J. Williams, "D825-A Multiple Computer for Command and Control," *AFIPS Proc. FJCC,* Vol. 22, pp. 86–96, 1962.

Barnes, J. H., R. M. Brown, M. Kato, D. J. Kuck, D. L. Slotnick, and R. A. Stokes, "The ILLIAC IV Computer," *IEEE Transactions on Computers,* C-17, Vol. 8, pp. 746-757, August 1968.

Bell, C. G. and D. Casasent, "Implementation of a Buffer Memory in Minicomputers," *Computer Design,* pp. 83–89, November 1971.

Bell, C. G., and J. Grason, "The Register Transfer Design Concept," *Computer Design,* pp. 87–94, May 1971.

Bell, C. G., D. R. Reddy, C. Pierson, and B. Rosen, "A High Performance Programmed Remote Display Terminal," *Proc. IEEE Computer Conference,* September 1971.

Bell, C. G. and A. Newell, "Possibilities for Computer Structures 1971," *AFIPS Proc. FJCC,* Vol. 39, pp. 387–394, 1971.

Bell, C. G. and A. Newell, *Computer Structures: Readings and Examples,* McGraw-Hill Book Company, New York, 1971.

Clark, W. A., "Macromodular Computer Systems," *AFIPS Proc. SJCC,* pp. 337–401, May 1967.

Conti, J., "Concepts for Buffer Storage," *Computer Group News,* pp. 9–13, March 1969.

Davis, R. L. and S. Zucker, "Structure of a Multiprocessor Using Microprogrammable Building Blocks," National Aerospace Electronics Conferences, 1971.

Goodyear Aerospace, "STARAN – A New Way of Thinking," a Goodyear Aerospace brochure, Akron, Ohio, 1971.

Holland, S. A. and C. J. Purcell, "The CDC STAR-100: A Large Scale Network Oriented Computer System," *Proc. IEEE Computer Conference,* pp. 55–56, September 1971.

House, D. L. and R. A. Henzel, "The Effect of Low Cost Logic on Minicomputer Organization," *Computer Design,* Vol. 10, No. 1, pp. 97–101, January 1971.

Husson, S. S., *Microprogramming Principles and Practices,* Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1970.

Lonergan, W. and P. King, "Design of the B5000 System," *Datamation,* Vol. 7, No. 5, pp. 28–32, May 1961.

Minker, J., "An Overview of Associative or Content-Addressable Memory Systems and a KWIC Index to the Literature: 1956–1970," *Computing Reviews,* Vol. 12, No. 10, pp. 453–504, October 1971.

Rice, R., et. al, SYMBOL (four papers on a hardware-implemented high-level language) *AFIPS, Proc. SJCC,* Vol. 38, pp. 563–616, 1971.

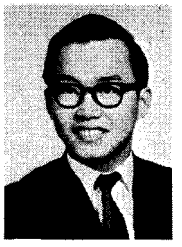
Roberts, L. G. and B. D. Wessler, "Computer Network Development to Achieve Resource Sharing," *AFIPS Proc. SJCC,* pp. 543–549, 1970. (Introduction to four other papers on ARPA network, pp. 551–597.)

Slotnick, D. L., W. C. Borck, and R. C. McReynolds, "The SOLOMON Computer," *AFIPS Proc. FJCC,* Vol. 22, pp. 97–107, 1962.

C. GORDON BELL received the B.S. degree in electrical engineering in 1956 and the M.S. degree in 1957, both from Massachusetts Institute of Technology, Cambridge, Massachusetts.

In 1959 he was with the Speech Communications Laboratory at the M.I.T. Division of Sponsored Research and from 1959 to 1960 was a Research Engineer with the Electronic Systems Laboratory at M.I.T. From 1960 to 1966 he was Manager in charge of computer design at the Digital Equipment Corporation (DEC), Maynard, Massachusetts, responsible for the PDP-4, 5 and 6 computers; he is also a consultant to DEC. He is presently a Professor of Electrical Engineering and Computer Science at Carnegie-Mellon University, Pittsburgh, Pa. He is a co-author of the book, *Computer Structures: Readings and Examples,* McGraw-Hill, 1971. Research interests center on the design of computer systems.

Bell is a member of the Association for Computing Machinery, Eta Kappa Nu, and a Senior Member of the IEEE.



ROBERT C. CHEN received the B.E.E. degree from Rensselaer Polytechnic Institute in 1966 and the S.M. from M. I. T. in 1968. From 1968 to 1969 he was engaged in design automation and simulation at Burroughs Corporation in Paoli, Pa. Since 1969 he has been working towards the Ph.D. at Carnegie-Mellon University.

He is a member of IEEE and ACM. He is also a member of Tau Beta Pi, Eta Kappa Nu and Pi Delta Epsilon, and an associate member of Sigma Xi.

SATISH L. REGE received the B. Tech. degree in Electrical Engineering from the Indian Institute of Technology, Bombay, India in 1966.

Between 1966 and 1968 he worked for IBM World Trade Corporation in Bombay, India and then came to the United States where he obtained his M.S. in Electrical Engineering from the University of Pittsburgh, Pittsburgh, Pa. in 1969. Presently he is working for a Ph.D. degree at Carnegie-Mellon University, Pittsburgh, Pa. His research interests are in the design of computer systems.