COMPUTER SYSTEMS: WHAT THE FUTURE HOLDS*

Harold S. Stone
University of Massachusetts

ABSTRACT

Continuing advances in device technology will result in substantially higher speed devices at rapidly diminishing costs. These changes will in turn have a significant impact on computer architecture in the next decade, and on the wide-scale proliferation of computer systems into new applications.

The microprocessor of today will eventually evolve to a processor with the power of a minicomputer or perhaps a medium-scale computer of today. Non-mechanical auxiliary memories are likely to be available as well. The computational power and low cost of these computer systems will see them used in the home, office and industry for a wide variety of new applications.

Medium-scale systems will tend to be total systems that are service oriented rather than hardware oriented. A major service will be that of the information utility to provide data to a widely distributed pool of on-site computers.

Large-scale computer systems have the potential to achieve two to three orders of magnitude speed improvement over the next decade. A large portion of this may come from the faster devices. Another significant portion will come from higher parallelism. For large numerical computations, the vector processor of today may evolve to a hybrid vector processor-multiprocessor to provide efficient operation on both scalar and vector types of computations.

I. INTRODUCTION

The past two decades have seen truly phenomenal advances in computers, but the potential of computers has barely been realized. The advances in computer technology anticipated in the next decade will be so widespread that computers will directly affect the living habits and quality of life of almost every person in the United States.

Since computer architecture is largely driven by device technology and software interfaces, Section II of this paper is devoted to an analysis of the devices that may be available in the 1980s, and to the smaller end of the computer scale. Here's where growth in the next decade will be most rapid. Medium-scale computers are treated in Section III, where we project that medium-scale computers will tend to be better oriented to the specific needs of the

---

*This paper is an abbreviated version of the article that appears in Computer Science and Scientific Computing, Academic Press, New York, 1976, edited by J. M. Ortega.

user than their predecessors of today. Finally, for large-scale computers, Section IV indicates that rather few new ideas in high-speed computer architecture are likely to appear in the next decade, but there is room to attain about two to three orders of magnitude increase in speed by perfecting present ideas.

II. ADVANCES IN DEVICE TECHNOLOGIES--THE COMPUTER ON A CHIP

Semiconductor and integrated circuit technologies have consistently achieved advances in density, speed, and power consumption over the history of solid state devices. Figure 1 illustrates some of these trends [Turn2]. Densities double roughly every two years at the present rate. Assuming that this continues and the 16K bit chip is a standard in 1976, then the megabit memory chip may appear late in the 1980s. To obtain densities leading to megabit chips, it will be necessary to achieve new breakthroughs in the resolution of the etching process by moving from visible light to electron-beam scanning techniques or beyond.

Apart from achieving greater resolution, there are other gains to be made from new processes. In the past decade, processes based on MOS (metal-oxide semiconductor) techniques have been characterized by high density, low power consumption, but low speed. The competing technology is bipolar, with high speed, but roughly one fourth the density and additional complexity in its fabrication. TTL (transistor-transistor logic) has been the favored type of bipolar technology for implementation of reasonably fast logic, and ECL (emitter-coupled logic) is another bipolar technology that attains the fastest logic speed. Unfortunately, the power consumption of ECL is very high, and its density is low, thereby leaving the designer no clearly best choice for a logic family.

Recent changes in technology seem to have pointed bipolar and MOS processes in the same direction. MOS circuits diffused onto a sapphire substrate instead of the traditional silicon substrate attain notably higher speeds than standard MOS circuits, but this technology has not yet overcome some obstacles that have impaired its development. In the bipolar technology, a new offshoot known as I^2L (integrated-injection logic) greatly simplifies the masks for active gates, thus increasing circuit density while retaining speed. I^2L logic has a speed more nearly that of ECL rather than that of the slower T^2L logic. If either I^2L or silicon-on-sapphire technologies succeed in attaining their respective goals, then one may have high speed, high density, and low cost all in one family.

Projecting these developments into architecture has a very interesting impact on the innovation known as the *microprocessor*. A microprocessor is essentially a complete processor compact enough to be constructed on a single chip. Actually, one often finds several chips used to make up a full-fledged computer with one chip consisting of the arithmetic logic and processor registers, another chip holding control memory, and yet another chip used for random-access memory. Input/output interfaces may be on yet other chips. As density of fabrication increases, the chip boundaries will grow larger and the number of different chips will be reduced.

We have three data points on the power of microprocessors. The 4-bit microprocessor was introduced in quantity in 1971, the 8-bit in 1974 and the 16-bit is being shipped in quantity in 1976. This is consistent with the claim that density increases by a factor of two about every two years. The chips themselves are increasing in size, too. Again projecting this forward by several years, we find that the complexity of the arithmetic unit of a microprocessor may attain that of sophisticated medium-scale machines of today by the 1980s. Figure 2 illustrates a speculation on where the trend may lead.

Although microprocessors will have the power of today's minicomputers, or more, in the 1980s, there is a major obstacle that must be crossed before microprocessor based systems can lead to substantial cost reductions in conventional minicomputer systems. The problem is mechanical auxiliary memory.

Fortunately, there are several possible nonmechanical replacements for auxiliary memory in various stages of development. Magnetic bubble memories are nonvolatile magnetic shift-register memories in which storage densities comparable to MOS memories have been achieved. Random-access time may be as low as 20 microseconds, more likely somewhat higher, but still some 100 times faster than access to rotating mechanical devices.

Another attractive storage medium is also shift-register oriented, and known as *charge-coupled device* (CCD) technology. CCD memories are volatile shift registers made up of capacitors. Charge in capacitors must be kept in circulation, unlike bubbles in magnetic bubble memories, but otherwise CCD performance characteristics closely approximate magnetic bubble memory characteristics. The first CCD memory chips for computers announced commercially appeared in 1975 and had 16k bits per chip. This puts CCD technology slightly ahead of magnetic bubbles, since bubbles had not reached the market place by 1975.

One other technology today is a candidate for replacing mechanical auxiliary memory, namely, electron-beam addressable memory (EBAM). This technology uses electron-beam techniques to deposit charges in a small region of a surface, and to read them out at a later time. EBAM is several years behind the development of CCD and bubble memories, but, once perfected, could be a strong contender since access to memory is by random-beam addressing rather than by serial access to shift registers.

III. MEDIUM-SCALE COMPUTERS

Computer manufacturers have to face the 1980s with a mixture of joy and grief. The joy stems from potential unit sales of 100 to 1000 times the present number of systems sold as computers move into every imaginable application. The grief is due to the decreasing cost of the hardware itself so that total sales volume of the hardware may drop precipitously even while unit sales are growing enormously. All the while this is happening, the end-user finds that a paltry sum buys him hardware of incredible potential, but to make it do his job he has to pour many thousands of dollars into software and program development.

So how will these trends affect medium-scale machines? Medium-scale computers will be designed to use inexpensive additional logic wherever possible to facilitate flexibility, and enhance the range of services that can be done effectively on the machine.

Among the several trends for medium-scale computers that are perceptable are the following:

1. A "rich" instruction set is included that permits many higher level operations to be done efficiently.
2. The use of microprogramming with a writeable control store will be prevalent, so that new instructions can be implemented by the user after physical delivery of the machine. New instructions might be included for each compiler target language to increase efficiency of execution of object code, and emulation of one architecture by another will be commonplace.
3. Large memories, both real and virtual, will simplify problems of writing programs of large size.
4. Executive and control functions will be done by special purpose hardware insofar as is possible to simplify the operating system and control program.
5. Virtual machine architecture will be widely used to aid the writing and debugging of the control software that cannot be implemented in hardware.

Projecting present trends forward to the late 1980s, we see that a device comparable in cost and size to the electric typewriter could be as powerful as a medium-scale computer of 1976. This will have a great effect on decentralizing the computer center as we know it today. What will be the function of shared-resource medium-scale computers then?

In the 1980s there will still be need for central computers for computer users to access. Access will be less for computational power than for information from central data files. The data will be a resource and a commodity of trade by that time if it is not already now. The user will almost certainly use the central data base for numerical data, catalogs, bibliographies, mail, and text, quite apart from uses he makes of programs stored centrally. Since information is created in real time, a computer user must tap that information through access to one or more centralized data bases even when he is able to satisfy his computational needs for that data through the purchase of inexpensive hardware.

IV. LARGE-SCALE SYSTEMS

By early 1976 a number of very high-speed computing systems had been installed and were in operation. Some of the systems use a standard serial instruction set, and use a number of clever design techniques to achieve high speed. For example, the CDC 7600 system uses multiple functional units that can operate simultaneously, and uses an intricate instruction scheduling mechanism to keep these units busy as much as possible, even executing the instructions out of order if that results in a net increase in speed.

One trend that has emerged in recent years is that of using a computer with a vector instruction set. Each vector instruction in such a machine operates on entire vectors instead of single elements. When a vector instruction is issued on a vector computer, that one instruction manipulates all of the elements of the vector operands, and achieves a great deal of parallelism of operation with a large gain in speed.

Two distinct types of computers with vector instructions have been delivered. One type is the *array computer* of the ILLIAC IV class in which each element of the vector is treated by an independent processor. Figure 3 shows a control unit linked to 64 processors in an array by a broadcast bus. Each instruction issued results in 64 responses, each on a different element of a vector of length 64. The other type, the *pipeline computer*, as exemplified by the CDC STAR, has the computational unit partitioned into successive stages, each of which can be busy simultaneously. A vector operation is initiated by placing the first operand pair into the first stage of the computation; as they pass on to the second stage, the next pair is passed into the empty first stage. Thus if there are N stages in the pipeline, N different operations may be in operation simultaneously, each in a different stage. Figure 4 illustrates the structure of a typical pipeline computer. Floating-point operations can be conveniently divided into about eight successive stages, and the pipelines themselves can be replicated to give additional parallelism.

To give some idea of the parallelism achievable on the present machines, ILLIAC IV has 64 processors, but each processor can do two single precision operations simultaneously, so that 128 different computations can be executed at once. The CDC STAR has an effective parallelism of about 32. The parallelism achievable is impressive, but is representative of designs in progress well over five years ago. The ILLIAC IV uses an integrated circuit memory, but no large-scale integration. The CDC STAR uses neither integrated circuit memory nor large-scale integration. It is obvious that technological changes available today can be included in the next generation of these computers to gain a potential speed improvement of approximately another factor of 10 at no increase in cost. If we take into account the advances that are certain to appear in the next five years in integrated circuit technology, then this could contribute a total factor of 50 improvement in speed over machines in operation today.

Unfortunately, a factor of 50 is not enough for the very large-scale problems for which these computer systems are built. Most notable of the massive calculations are fluid dynamics problems and weather analysis. We will still be a factor of 10^5 too slow to solve these problems in their full detail.

The obvious answer to attain higher speed is to increase the degree of parallelism where possible. When logic costs drop very low, the number of identical units that can be put into a design of marketable cost can increase from 10^2 in 1976 to perhaps 10^3 or 10^4 in the late 1980s. Unfortunately, the speed increases attainable fall short of being equal to the replication factor.

A number of lessons have been learned from experience with vector computers like STAR and ILLIAC. A few of the principal ones are given below:

1. When algorithms can be cast in vector form there are significant advantages due to elimination of unnecessary overhead for individual elements.

2. It is possible to incur substantial overhead in vector algorithms in communicating information among elements of a vector when operations on one element are influenced by the value of another element.

3. There are numerous tricks for casting serial algorithms into vector form. A programmer may have to experiment with various alternatives to obtain the best alternative. The best vector algorithms for particular problems may be quite unconventional and, in fact, may not be very efficient when performed in equivalent serial form.

4. Major bottlenecks occur when sequential scalar operations have to be done in between vector operations. This reduces the effective speed of a highly parallel machine drastically and the effect becomes more pronounced in machines as the parallelism increases.

By all appearances the vector machine is not the final answer, although the range of problems for which vector machines are well-suited has proved to be much larger than anticipated because of innovations in parallel algorithm and architectural features.

T. C. Chen (ref. 1) among others observed the performance deficiences from intermixing parallel and serial processes. Figure 5 illustrates a typical duty cycle for an array processor in which one processor is kept busy initializing a vector process, then all N processors are ganged together performing the vector operation. Chen observed that a pipeline computer duty cycle figure has the form of staircase in figure 6, to show how each successive state initiates activity slightly later than its predecessor stage. The shaded region in dark boundaries is exactly equal to the unshaded region in dark boundaries, so that the shaded area of the pipeline computer duty cycle is exactly equal to the shaded area of an array processor computation as shown in the previous figure. With this observation it is clear that there is a potential performance decrease in a pipeline computer due to a phenomenon very much like the serial overhead prior to a vector computation in an array computer.

The ILLIAC IV is designed to perform the computation shown in figure 5 as shown in figure 7, where the serial computation is done in a single control unit, and is done while the previous vector operation is in progress in the arithmetic processor array. This vastly reduces time lost due to interspersing serial and parallel operations. The equivalent processing duty cycle for the pipeline computer is shown in figure 8, which simply shows one vector operation initiated before the termination of the prior one. The CDC STAR pipeline computer presently does not have the facility to execute in this manner. Thus, the STAR duty cycle is more like that shown in figure 9.

To achieve better total performance than is predicted by Chen's pessimistic analysis, it is clear that the architecture of the 1980s will have a mix of processors, some of which are dedicated to serial types of tasks, and some dedicated to highly parallel or iterative types of tasks. Execution overlap among processing units will have to be significant to attain the speed potential of having many arithmetic units.

With microprocessors so inexpensive, there is an obvious motivation to construct vector or multiprocessor computers from arrays of microprocessors. While the individual speed of any one microprocessor may be moderate, the ability to gather 10^3 or 10^4 processors together in a single computer can lead to a very high-speed computer with tremendous computing power for reasonable cost. Hardware advances have unfortunately, outstripped architectural and algorithmic advances, to the extent that it is now possible to construct arrays with incredible computational power, except that it is not clear what form the arrays should take and how calculations should proceed in them.

To summarize the current trends for high-speed machines, a factor of 50-speed improvement is possible by the end of the 1980s from technological advances in devices, but the demands of very large problems will stimulate evolution of the architecture itself. Vector machines look more promising than multiprocessors for large-scale problems for the long-term future, but some mix of the two may emerge and prove to be the best solution.(See ref. 2.)

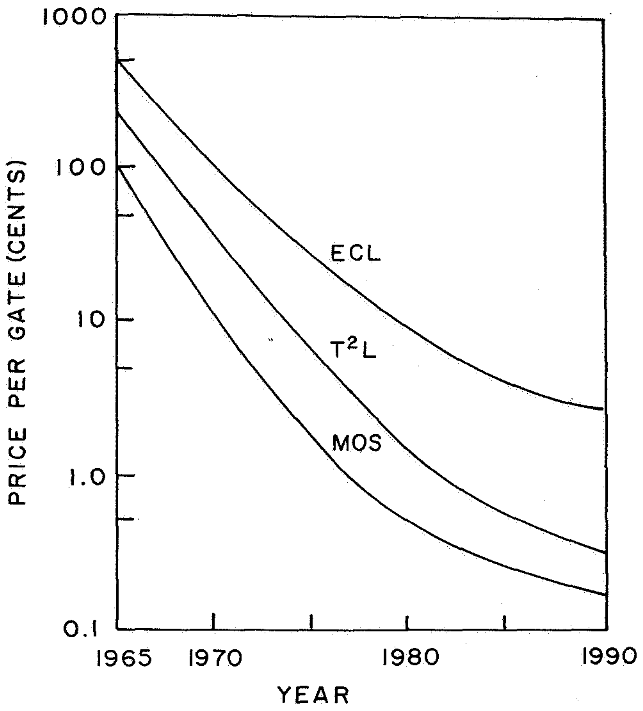
V. CONCLUSIONS

With technological advances leading the way as we move into and through the next decade, computer architecture will evolve to enhance the proliferation of the microprocessor, the utility of the medium-scale computer, and the sheer computational power of the large-scale machine. The most dramatic changes will be in new applications brought about because of ever lowering costs, smaller sizes, and faster switching times. There is no evidence at this time that the rate of advance in computer technology will slow significantly in the 1980s. We are truly undergoing a Computer Revolution of the scale of the Industrial Revolution.

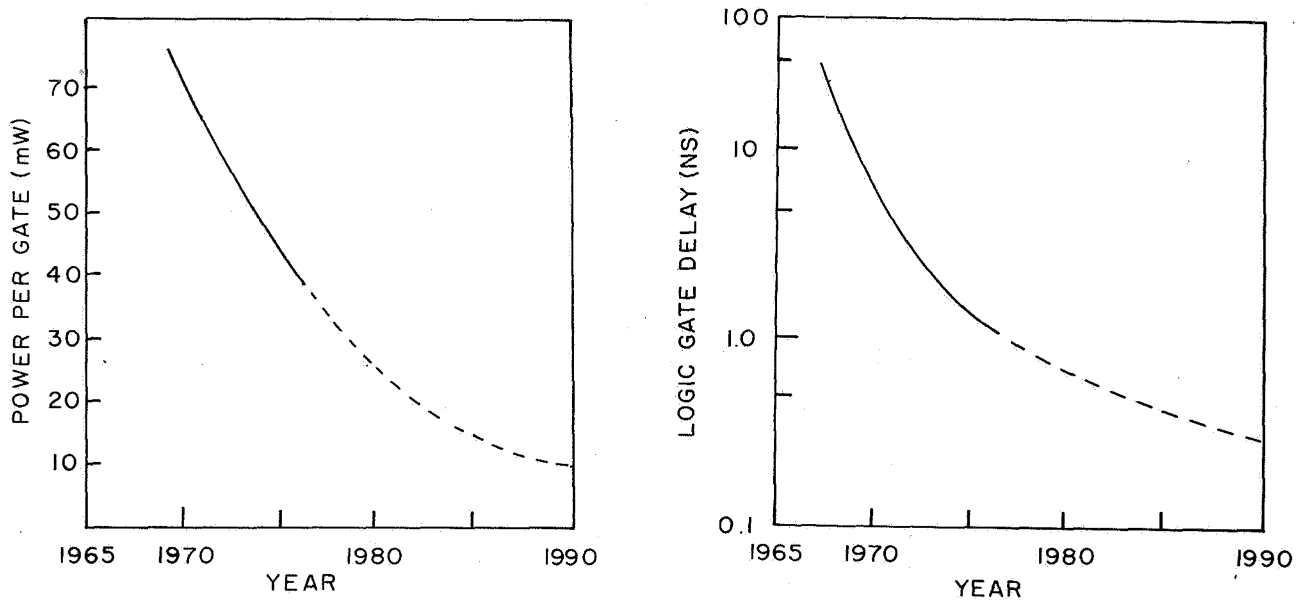
REFERENCES

1. Chen, T. C., "Parallelism, pipelining, and computer efficiency," *Computer Design*, pp. 69-74, January 1971.

2. Turn, Rein, *Computers in the 1980s*, Columbia U. Press, New York, 1974.



(a) Price trends.



(b) Power consumption trends.

(c) Logic speed trends.

Figure 1.- Trends in device technology.

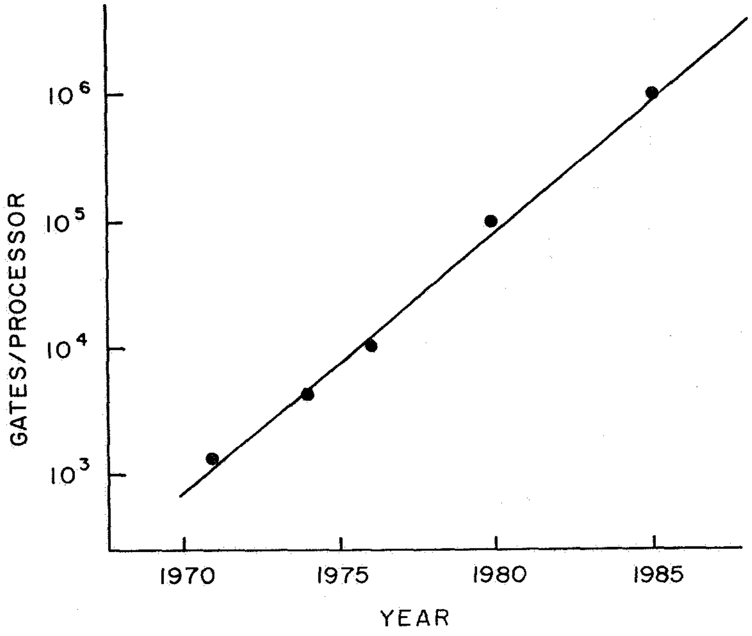


Figure 2.- Microprocessor complexity.

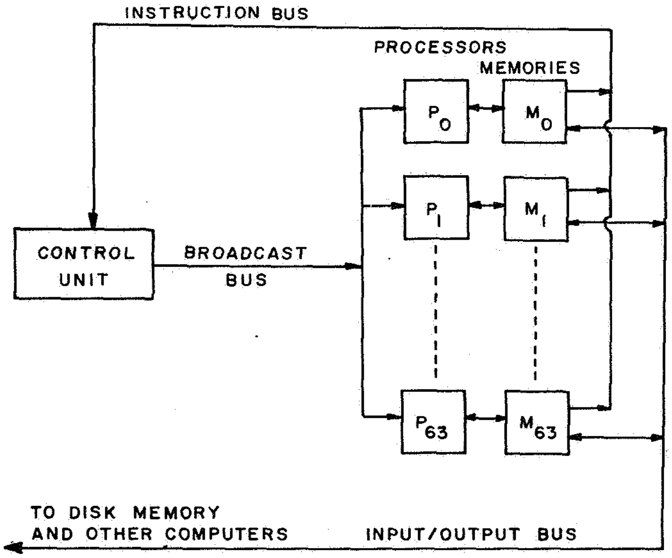


Figure 3.- An array computer (ILLIAC IV).

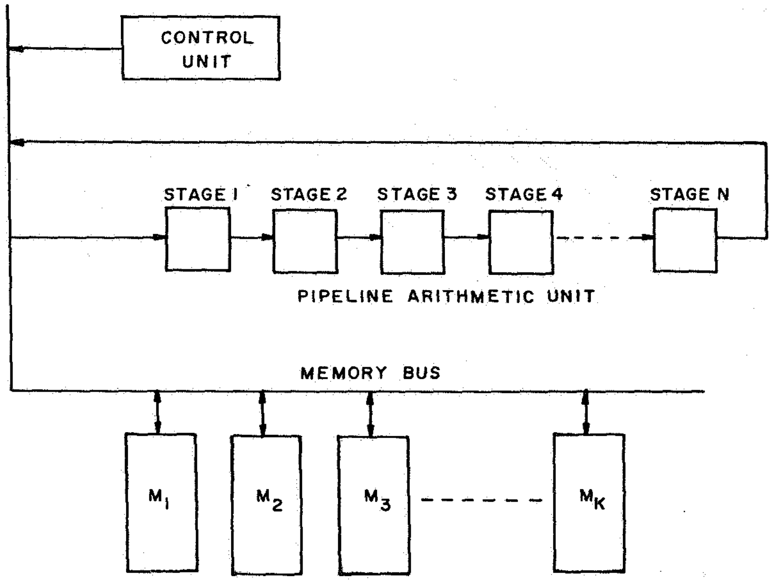


Figure 4.- A pipeline computer.

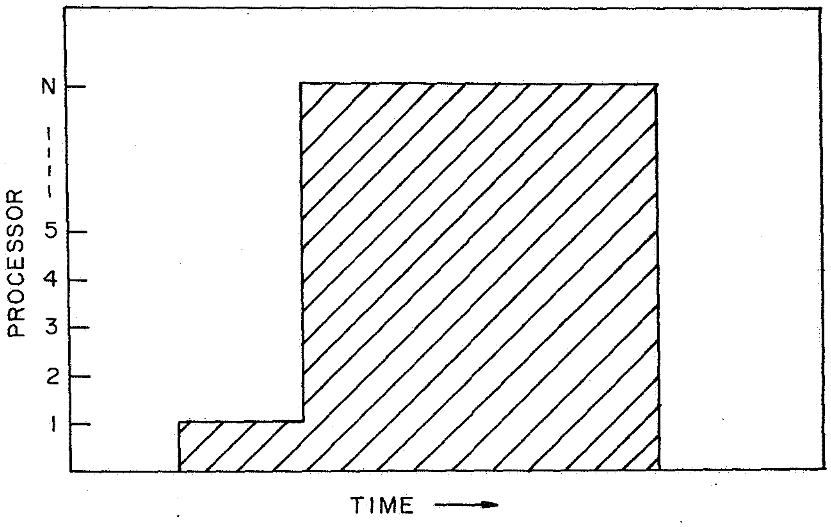


Figure 5. Duty cycle for an array computer.

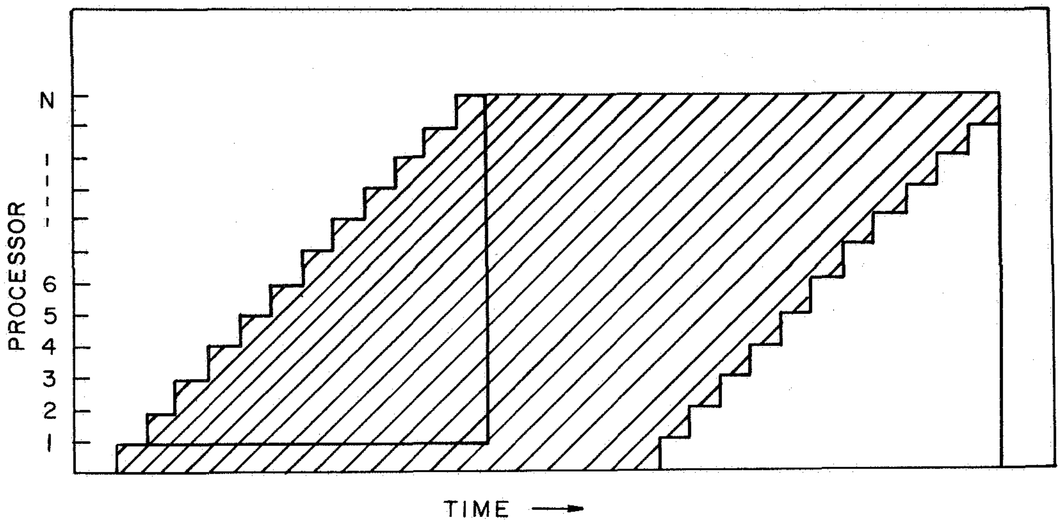


Figure 6.- Duty cycle for a pipeline computer.

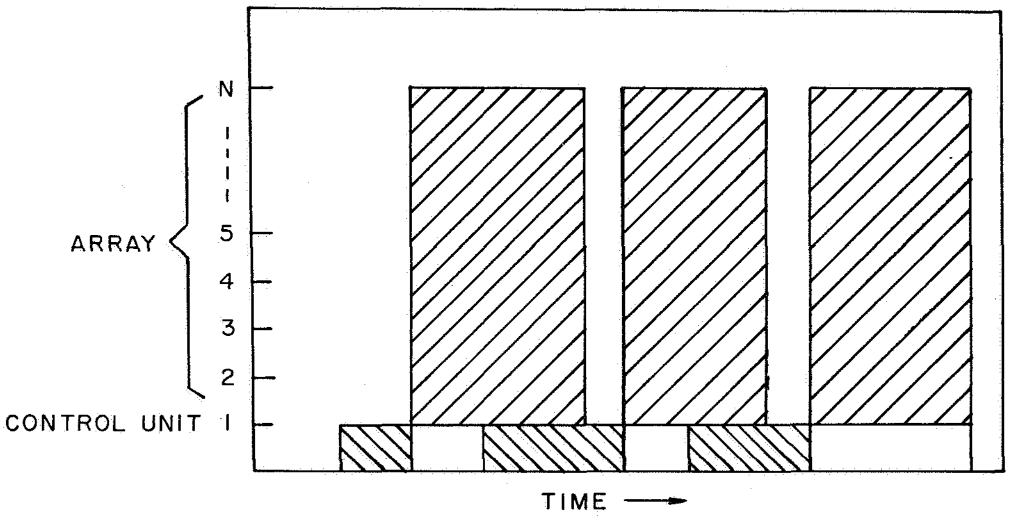


Figure 7.- ILLIAC IV duty cycle.

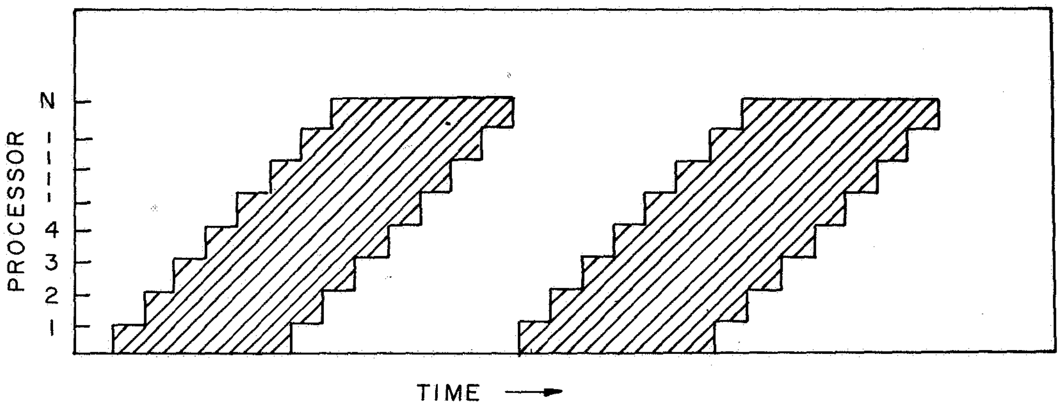


Figure 8.- Duty cycle for pipeline equivalent of ILLIAC IV.

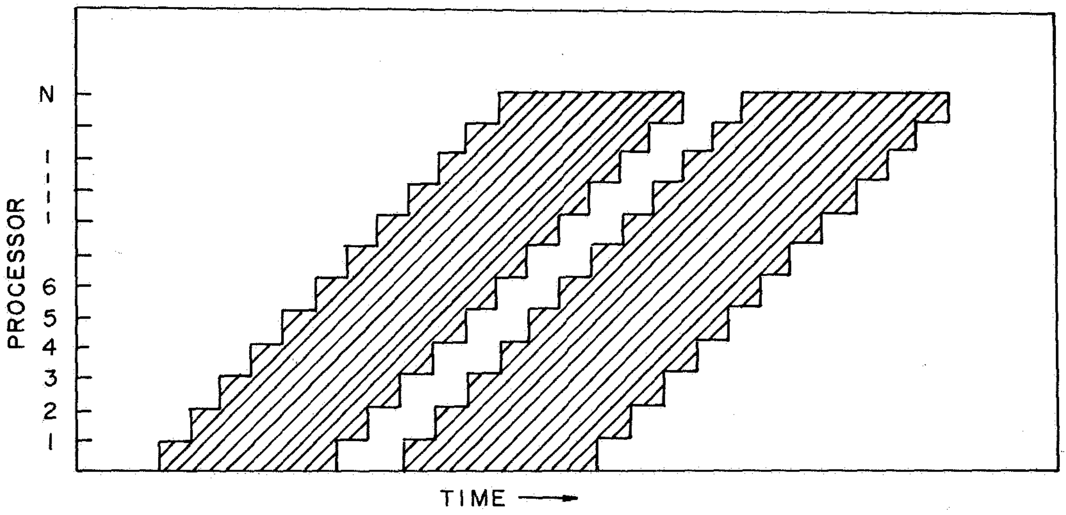


Figure 9.- Duty cycle for STAR.

NASA-Langley, 1976