REPORT NO. DOT-TSC-OST-72-23

FIVE YEAR COMPUTER TECHNOLOGY FORECAST

Andres Zellweger
Transportation Systems Center
Kendall Square
Cambridge, MA. 02142

DECEMBER 1972

FINAL REPORT

DOCUMENT IS AVAILABLE TO THE PUBLIC
THROUGH THE NATIONAL TECHNICAL
INFORMATION SERVICE, SPRINGFIELD,
VIRGINIA 22151.

Prepared for:
DEPARTMENT OF TRANSPORTATION
OFFICE OF THE SECRETARY
Office of Research & Development Policy
Washington, D.C. 20590

NOTICE

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof.

1. Report No.
DOT-TSC-OST-72-23

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
FIVE-YEAR COMPUTER TECHNOLOGY FORECAST

5. Report Date
December 1972

6. Performing Organization Code

7. Author(s)
Andres Zellweger

8. Performing Organization Report No.
DOT-TSC-OST-72-23

9. Performing Organization Name and Address
Department of Transportation
Transportation Systems Center
Kendall Square
Cambridge, MA. 02142

10. Work Unit No.
R3543

11. Contract or Grant No.
OS323

12. Sponsoring Agency Name and Address
Department of Transportation
Office of the Secretary
Office of R&D Policy
Washington, D.C. 20590

13. Type of Report and Period Covered
Final Report

14. Sponsoring Agency Code

15. Supplementary Notes
This report was prepared for the TASS Program Office, TSC/PA.

16. Abstract
This report delineates the various computer system components and extrapolates past trends in light of industry goals and physical limitations to predict what individual components and entire systems will look like in the second half of this decade. The report will emphasize the nature of components (e.g. CPUs, primary memories, secondary memories, ultra large storage devices, etc.) and the system architectures that will be commercially available as "off-the-shelf" items rather than one-of-a-kind systems that might exist in five years.

17. Key Words
Computers, Memories, Processors, Networks, Forecast

18. Distribution Statement
DOCUMENT IS AVAILABLE TO THE PUBLIC THROUGH THE NATIONAL TECHNICAL INFORMATION SERVICE, SPRINGFIELD, VIRGINIA 22151.

19. Security Classif. (of this report)
UNCLASSIFIED

20. Security Classif. (of this page)
UNCLASSIFIED

21. No. of Pages
102

22. Price

PREFACE

For the past two years a study has been in progress at the Transportation Systems Center (TSC) under the auspices of the Office of the Secretary of Transportation to investigate the requirements for a design of a Transportation Analysis and Simulation System (TASS). In FY72 the primary objectives of this study were to determine TSC's long range (FY77 era) computer system needs and to determine options to meet these needs.[1] This report, a five-year computer technology forecast,has been produced to assist in the determination of options in the TASS planning effort.

The author gratefully acknowledges the careful review of drafts of this report and the subsequent helpful criticisms and comments provided by the TASS Program Office at TSC and the R&D Policy Analysis Division (TST-12) of the Office of the Secretary of Transportation.

[1] A more detailed discussion of project history and objectives can be found in "Transportation Analysis and Simulation Systems Requirements", H. G. Miller, and D. Hiatt, DOT-TSC-OST-72-26

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT.)

LIST OF ILLUSTRATIONS

LIST OF TABLES

1.0 INTRODUCTION

This report is a five-year computer technology forecast. It will project the state-of-the-art of technology applicable to future computer systems, and the impact of this technology on options for future computer system planners. The report will delineate the various system components and extrapolate past trends in the light of industry goals and physical limitations to predict what the individual components and the entire system will look like in the second half of this decade. All forecasts of this nature must be viewed with a certain degree of caution for they talk in terms not of what will happen, but of what is likely to happen.

The report will emphasize the nature of the system components (central processing units, primary memories, secondary memories, ultra-large storage devices, etc.) and the system architectures that will be commercially available as off-the-shelf items. It will not be concerned with one-of-a-kind prototype systems that might exist in five years. The reason for this is fairly obvious: the forecast is to be used in the planning of a service facility, not a facility for computer science research. Finally, computer software and input/output devices (e.g. printers, card readers, terminals, graphic displays, etc.) are beyond the scope of this report. In the long run, the direction taken by computer architectures is affected by application areas and user needs, marketability and profit potential, and by technological availability. Needs force the technology in the sense that needs determine the allocation of R&D funds. In the short run technological availability is more critical because a lead time of several years is required to get from the stages of invention and prototype to full-scale production. The implication for a short-term forecast such as this one is that to be commercially available in five years, a system must exist in prototype today. For this reason, to predict a computer architecture of 1977, we look at a number of computers whose prototypes are currently under construction or already in existence.

For the benefit of the casual reader, the report has been organized to contain a forecast summary (Section 2) before the body of the report. This summary is complete in itself, but the interested reader is urged to look at the details in the body of the report (Sections 3 thru 6). The report proper assumes no specific computer expertise (definitions and descriptions of concepts are provided where necessary), but a degree of technical sophistication is assumed.

2.0 FORECAST SUMMARY

2.1 INTRODUCTION

In the mid to late 1950's it was projected that one IBM 7090 class computer could accommodate all of the U.S. Government's computing needs and that approximately 10 such machines would satisfy the requirements of the entire United States. This prediction has been far surpassed; installed computers today number in the tens of thousands with the largest having over a hundred times the processing power of a 7090. It is interesting to note that the largest computers today cost less than ten times the three million dollars charged for a typical IBM 7090 system in the late 1950's. The growth in numbers and size of the 50's and 60's is expected to continue in the 70's. For example, the Geographical Fluid Dynamics Laboratory of the U.S. Department of Commerce, which has ordered one of the very large supercomputers capable of 66 million instructions per second, feels that it could use a computer at least twenty times as powerful for its simulations of the oceans and atmosphere.

The discrepancy between prediction and actuality is of course due to the once undreamed of, but now ever increasing range of computer applications. Today, with our increased understanding of the kinds of things that computers can do for us, few people are willing or, for that matter, feel it necessary to speculate on the limits of practical computer uses or on the absolute computing power required to satisfy man's needs. The current interest is in the types of applications that will emerge in the future, the demands that will be placed on computers by existing and new applications, and in the kinds of computers that will be built to meet these needs.

This report and, in particular, the summary in this chapter address the third area of concern - the computers of the future. In forecasting what the commercially available computers of the 1977 era will look like, we will consider the specific components of computing systems: processors, primary memories, secondary and bulk memories, and, to lesser extent, peripherals. Trends in components are examined in terms of technological feasibility, range of applicability, and commercial interest. This forecast of computer system building blocks combined with the design criteria set forth by the increasing demands for easier to use computers leads to a projection of 1977 commercially available computer systems.

2.2 GENERIC DESCRIPTION

A general purpose computer system consists of four types of elements: (1) a set of processors which operate on data; (2) a hierarchy of memories in which data is stored temporarily or permanently; (3) a group of devices used for input and output of data (including control information) from the memories; and (4) finally a set of interfaces to facilitate the flow of information between the rest of the system elements (see Figure 2-1). The nature of a particular application determines the degree to which the components in the various classes are used. While memories and processors will undergo some operational changes in the next five years, functional characteristics will remain fairly fixed. Major changes will occur in the I/O area as more powerful man-machine capabilities are developed.

Early stored program computers contained only a primary random access memory, a combination control-arithmetic unit and, for input and output, some simple device like a teleprinter which could type, punch paper tape, and read paper tape. With time secondary memories (at first magnetic tape units and later disk drives) and more sophisticated I/O devices (card readers and punches, printers, etc.) were added to computers. These early systems worked on one program from start to completion and normally performed only one operation (i.e. input, computation, or output) at a time. As systems became larger it became necessary, for reasons of time and cost efficiency, to overlap operations. To print a line, for example, information was first placed in a buffer and the central processing unit (CPU) was then used to continue computation and, as the need arose, to transfer further information to the printer output buffer. This process improved performance, but when a program's requirements for the various system resources were not balanced many parts of the system remained idle. Today multiprocessing and multiprogramming

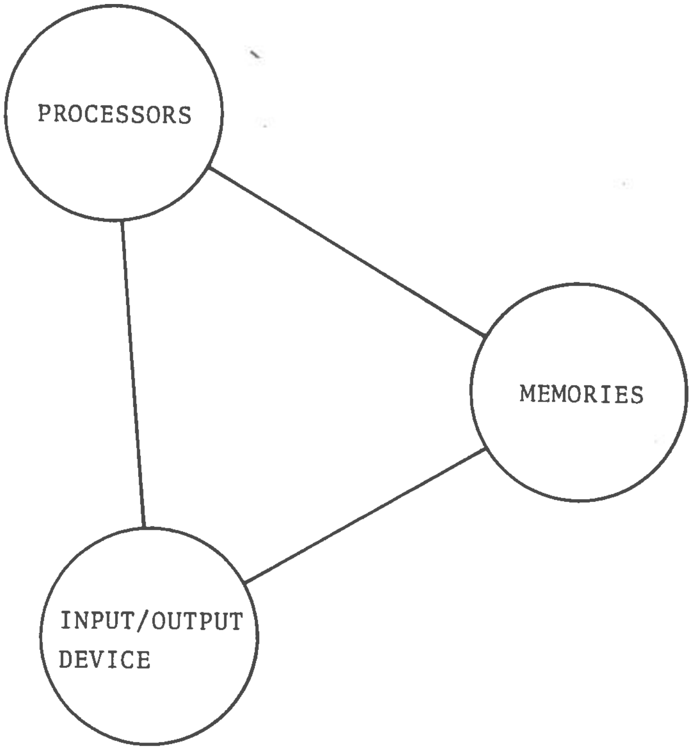


Figure 2-1. Basic Components of a Computer System

systems are designed, in terms of both hardware and software, to permit many simultaneous activities to share the system resources and thereby, hopefully, to keep all components busy as much of the time as possible.[1] This is done because the various resources in a computer system are expensive and could not be justified unless the usage rate is fairly high.

It is becoming more and more obvious that even a multiprogrammed or time-shared[2] system does not operate most effectively in a stand-alone mode. Systems cannot afford to contain very expensive pieces of hardware that, although needed, are only used occasionally.

[1] While one processor can ordinarily only execute one instruction at a time, it is possible to run several programs together and to let each use a resource (processor, memory, I/O device) as it is needed. Thus while one job is computing, another may be printing, etc.

[2] A time-shared computer system is also a system where many users are being simultaneously serviced by the computer. It differs from a multiprogramming system in that a time-sharing user works on-line via a terminal in the teletype class in an interactive manner.

Large software packages developed on one system often cannot be transferred for use at another site because of system incompatibilities. A larger data base might be needed by users at a number of different installations, but the cost of keeping multiple copies and of assuring accurate updating of each is prohibitive. To overcome these and other problems it is possible to connect a number of computers via telecommunication links to form a computer network. Such an arrangement enables a user of one system to run a program at another installation where the necessary hardware and software exist and to have this program interact with a program on his own system. Users of many systems can, via a network, access data at one location. Networks of computers are being established and their role is expected to expand in the 70's.

In conclusion, the simple computer consisting of a processor, memory, and I/O system has evolved to a computer network consisting of individual machines that may contain several processors that often operate in multiprogramming or time-sharing modes. New memory and CPU architectures have evolved and new functional components, especially for data communications, are becoming standard parts of computer systems. Subsequent sections in this chapter will indicate the direction that this evolution is expected to take in the 1970's.

2.3 IMPACT OF LSI

From a technical viewpoint, the prime mover that has made possible the multi-user computers and the networks discussed in the last section and that will shape the computer components of the 1970's is the semiconductor technology of the 1960's and the subsequent large scale integration (LSI) of components. The ability to make small silicon chips (.05 inches squared) each of which can contain the circuitry for complex logical functions or memories of up to 4096 bits[1] has had a marked effect on the relative costs of com-

[1] This current maximum production chip density is expected to double every year for at least the next five years.

puter components and thus on computer architectural design criteria[1]. This section will examine the specific impact that LSI will have on memories, minicomputers, and processors of the 1970's.

The per bit cost of memories, which used to decrease as memory size increased, is becoming less dependent on size and thus smaller memories are becoming relatively cheaper. This means that it is now economical to use a number of small but functionally separate memories in computer peripherals and terminals. The main memory could be organized into a number of smaller parallel modules or into a hierarchical structure. Associative memories[2] can be built at a reasonable cost because logic and memory functions can be combined on one chip. Very fast but small associative memories are already finding use in the hardware implementation of improved addressing schemes. Larger slower versions are expected to become computer system options for user applications in the near future.

The impact of LSI is most apparent in the minicomputer industry where a drastic downward trend in cost has been experienced in the last five years. This trend is expected to continue with a price drop by a factor of four predicted in another 5-10 years. This will be accompanied by a ten-fold speed increase (from one microsecond add times to 100 nanosecond add times). The availability of low-cost minicomputers that take up very little room is leading to a functional distribution of many of the program execution support tasks from the CPU to the peripherals and terminals. By the

[1] It is interesting to note that LSI technology is also affecting the actual computer design process. In particular, a large part of the cost of any integrated circuit is in the design of a specific chip type. This is leading to a standardization of chips and of packages of chips that will have an effect on computer design quite similar to the effect of higher level languages on application software design. Instead of designing computers at the level of logical functions (e.g. 'and' and 'or' gates) they will be designed in terms of more complex units (e.g. adders, shift registers, and scratch pad memories).

[2] An associative memory is a memory whose words are addressed by part or all of the content of the word rather than by the location of the word in the memory.

late 1970's "smart" disks, for example, are expected to have minicomputer controllers that, in addition to performing ordinary disk control functions, will be capable of receiving requests from diverse sources, queueing these requests for optimal disk utilization, and returning the desired data to the correct places. This type of functional distribution is called distributed computation.

In the 1970's we will see an increasing tendency towards computers that are parallel in the sense of having a number of similar processing elements and memories in one machine. The fact that a large part of IC chip cost is in the initial design is partially responsible for this trend because it makes it much more economical to use a small number of chip types repeatedly than to use many diffirent chip types.

2.4 TRENDS IN PROCESSORS

The trend in processors of the 70's is toward parallel operations, distributed computation, and microprogramming.[1] In this section we will examine some reasons for this trend and some of its implications.

Two primary reasons can be given for the architectural trends toward parallelism and distributed computation. First, computer speeds, which showed tremendous increases in the past twenty years, are approaching practical limits. Parallel schemes in the form of SIMD architectures or of Multiprocessors (i.e. computers with several CPUs), and distributed computers imply that several computations are being performed concurrently and thus single processor speed limitations are avoided. The second reason is that the advances in electronic technology discussed in the previous section make these architectures cost effective.

[1] A machine language computer instruction is composed of a number of logical operations carried out by switching circuits. Traditionally, these sequences were hardwired into a computer to define its instruction set. More recently these sequences of operations have been controlled by a program residing in a very fast memory (control store). It is possible to change such a microprogram to alter the computer's instruction set.

Examples of parallel architectures are associative and array processors. Both are called SIMD (single instruction multiple data) machines because a single instruction issued by a central control unit is executed simultaneously on a separate data element by each of several processors. The array processor has a memory attached to each of the processing elements while the associative processor has only one word (64-512 bits) tied to each processor.[1] The associative processor derives its name from its ability to execute the instruction on only those words whose content matches a pattern issued by the control unit. Both schemes are of interest, the array processor because it is suited for operating on many parallel sets of data and the associative processor because it can perform operations on subsets of data defined by some common property.

Distributed computing is advantageous not only because it allows simultaneous activities but also because it functionally distributes the computing to those processing elements best suited to performing the various tasks. Minicomputers in peripheral controllers, front-end processors (i.e. telecommunications processors and message concentrators), remote batch terminals, satellite graphics terminals, and intelligent terminals[2] relieve the CPU of the burdens of I/O operations, file maintenance, and communication tasks and thereby leave it free for performing the main program computations. In time-shared systems cost savings can be achieved by using minicomputers that interface the main computer to the terminals for handling editing and simple on-line calculations.

A major impact on computers of the 70's comes from the increasing availability of microprogrammed processors. By changing a com-

[1] For this reason an associative processor is often described as a memory in which each word has its own processor.

[2] An "intelligent" terminal is a terminal designed to do local processing in coordination with a large remote computer. It is usually made up of a minicomputer, a CRT, a keyboard, a small line printer, and a secondary storage medium such as a cassette tape.

puter's microprogram it is possible to make it emulate other computers and thus to avoid many software compatibility problems. It is also expected that reloadable control stores will extend the life of a computer. The IBM System 360, for example, had a six-year life span (1964-1970) but the life of the microprogrammable System 370 should be at least 10 years because, instead of announcing a whole new series om machines, IBM will be able to make advances in the 370 through the simple process of changing the microprograms.

Manufacturers of medium and large-scale computers do not encourage user microprogramming, but a number of minicomputers designed to be microprogrammed by the user are appearing. Users will be able to mold these machines for more efficient operation in specific application areas by adding one or two instructions or by redefining the entire instruction set. Microprogrammed minicomputers are expected to be used both as special purpose processing elements of larger systems and as stand-alone computers. It is expected that many of the microprogram control stores will be dynamically writable thereby permitting users to specify microprogram changes as part of their ordinary programs. Read-only microprogram memories pre-programmed for special applications (comparable to today's software packages) will become commercially available in the 1970's.

2.5 TRENDS IN MEMORIES

Present generation computers are built around one large main memory because with core memories, the prevalent technology of the 50's and 60's, low costs are obtained by driving large magnetic arrays with a small amount of electronics. Although LSI has changed this size cost relationship, the single memory is a precept of today's operating systems and of many computational algorithms, so that one can expect a lifetime of at least another 5-10 years for this memory architecture. LSI will lead to a distribution of memories to more locally autonomous functional processing units, but the computations of a program will rely on the single central memory.

There is a tendency towards implementing what looks like one large memory to the user by a hierarchical physical structure. Frequently used instructions and data are copied from the main memory into a high speed cache memory to reduce access time. To relieve the user of the burden of coping with a limited memory size a scheme called a virtual memory system is implemented. The user is permitted to address a large amount of data as though it were all in the main memory; the computer stores the users data in a secondary (backup) memory and only brings it into the main memory when it is referenced. Normally a virtual memory scheme divides the user's address space into blocks of words and brings an entire block into the main memory whenever a word in the block is required. The scheme works because programs exhibit a property, called locality, whereby a reference to a word is more likely to be followed by a reference to a nearby word than by a reference to a remote word. It has been found that cache memory schemes are not as effective when used with virtual memory systems; thus it is expected that as more computers add virtual memory capabilities in the 1970's cache memories will disappear.

In today's technologically advanced computers the memory hierarchy contains microprogram memories, caches, main memories, magnetic disks (and possibly drums), and finally magnetic tapes. There are fairly clear tradeoffs between storage capacity, access time, and cost; but two gaps, between main memories and disk, and between disks and magnetic tapes are evident. The first gap is due to the need for a faster backup memory for virtual memory systems; the second gap is due to the storage requirements of the increasingly numerous large on-line data bases. Both gaps are expected to be filled in the next five years.

Primary memories provide access times that are on the order of a thousand times faster than disk access times but at a considerably greater cost. Two candidates that would fill this gap because they are potentially cheaper than primary memories but faster than electromechanical devices disks are an LSI-MOS semiconductor memory[1] and

[1] MOS (metal-oxide-semiconductor) is one of two prevalent semincoductor technologies. It is characteristically the slower but cheaper of the two semiconductors.

a magnetic bubble memory. Cheap LSI-MOS is expected to be commercially available in the mid 1970's. Considerable research is underway in the bubble area but commercial systems are not expected before the late 1970's or 1980. Bubble memories and perhaps some LSI-MOS memories will be Block Oriented Random Access Memories (BORAMs), where, instead of addressing single words, one addresses blocks of words. This obviously is no restriction if the BORAM is used as a virtual memory backup store.

The second gap in the memory hierarchy is being filled with the ultra large memory systems that are just now becoming commercially available. These memories have on-line capacities of a trillion or more bits at a cost of .0001-.0002 cents/bit. Access times are in the 1 to 10 second range. The most promising system is based on a discrete bit laser memory that can be written once and then read as often as desired. Recording is on replaceable rhodium coated plastic strips. Another system more suited for archival storage because of slower access time is based on helical scan recording on two-inch magnetic tapes. Techniques based on holographic recording are not expected to become important until the 1980's because as yet there is no practical way of holographic recording in a marketable device.

The secondary memory market is expected to be dominated throughout the 70's by moving head disk systems despite the projected rise of ultra large memories and the faster bubble and LSI-MOS memories. A life of at least another two years is predicted for IBM 2314 class systems, the mainstay of the 1960's. The newer IBM 3330 class system will be prevalent until the late 1970's. Some improvements in capacity are expected, but, more significantly, the trend towards smart disks will result in greater efficiency.

2.6 COST TRENDS

The trend towards cheaper and cheaper semiconductor devices and the evolutionary architectural changes are expected to have an impact on both the absolute cost of computers and the relative costs of components. This section provides a brief summary of these trends with particular reference to processors, memory, and minicomputers.

It is expected that the combined cost of a large processor and memory will drop by a factor of six by the late 1970's. This drop is due primarily to the decreasing cost of electronic components (memory and logic)[1]. Non-electronic components costs will also drop, but at a slower rate. Dennis and Smith (Ref. 7), for example, predict that for large processors packaging cost, currently at nine percent of the entire processor-memory cost, will rise to fifteen percent and that the relative cost of power supplies and cooling will rise from eleven percent to thirty percent.

In five years relatively more will be spent on a computer's memory than on the processor or I/O control logic because memory prices are decreasing more slowly than logic prices and because memory size is expected to increase. Dennis and Smith (Ref. 7) make the optimistic prediction that a memory costing over $100K today would cost under $3K by the late 1970's. Secondary memory (disk) cost will drop from .01 cents/bit in 1970 to .001 cents/bit in 1975. Bubble and optical memories with equivalent or better performance than disks are expected to bring secondary memories costs to .0001 cents/bit by the late 1970's.

As indicated in Section 2.2, the cost of minicomputer CPUs, memories and I/O control units is expected to decrease by a factor of four by the late 1970's. This accompanied by the increased availability of sophisticated software and the recent trend towards low cost peripherals specifically designed for minicomputers, will make general purpose minicomputers extremely cost effective.

In summary, while computer performance has been increasing at three to five times every five years, cost to users has been decreasing by a factor of three or five each five years. This trend is expected to continue throughout the 70's. The expected easier-to-use computers will have larger system overheads, but these will be absorbed by the lower hardware costs. Despite easier-to-use systems, increases, in people cost and the lower hardware cost are

[1] A spokesman for a printer manufacturer was recently quoted as saying that circuits priced at one dollar in late 1971 cost only thirty cents in mid 1972 (Datamation, June 1972, p. 108).

expected to change the ratio of software to hardware cost from the current 2-1 to 3-1 by 1975.

2.7 COMPUTER SYSTEMS OF THE 1977 ERA

Having looked at the future of some of the individual components of a computing system, it now becomes possible to project how a typical computer of 1977 might look. This section describes such a computer in terms of its components and its interface to the user. Finally, the role of this typical computer as part of a larger, perhaps national computer network is examined.

An average computer (see Figure 2-2) will contain a number of processors including CPUs, file controllers, I/O processors, and communications processors. These processors represent a step in the evolution towards a fully distributed computer. They will all have access to a large main memory divided into a number of modules. The overall memory structure will be hierarchical but the user will see a single virtual memory. Specialized components such as associative memories or processors, array processors, and microprogrammable special-purpose minicomputers might well be an integral part

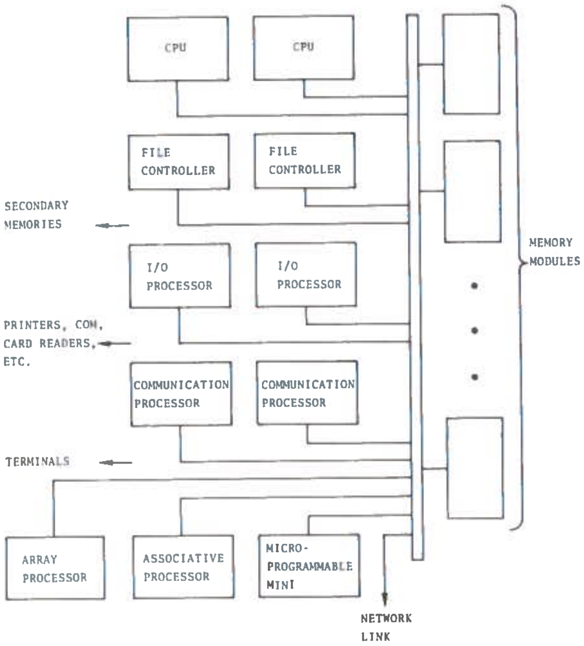


Figure 2-2. Computer Of The 1977 Era

of such a computer. It is expected that this average system of 1977 will be larger than the current average system.[1]

To the average user the computer of 1977 will look quite similar to the large time-shared or multiprogrammed computer of today. It is expected that the computer will be easier to use with simple, yet powerful machine independent procedure-oriented languages, data management languages, and command languages. Most scheduling, memory management, input/output control, file management, and communication management will be automatic. This will increase system overhead but the decrease in hardware cost and the projected savings in software development cost will make these inefficiencies tolerable if not justifiable.

The phenomenon of being easier to use will also be reflected in the hardware portion of the man-machine interface. Interactive computers will be accessible through well engineered CRT consoles that **have** some inherent computing capabilities. Data entry, which today represents 25-50% of EDP operating cost, will undergo perhaps the most revolutionary change of all the aspects of computing. The traditional keypunch will be replaced by keyboard-to-tape systems, optical character readers, on-line terminals, and, by the late 1970's, voice input systems. All of these will facilitate the entry of data at its source. Output from the computer, today done primarily via printers, will be done increasingly by computer output microfilm (COM) equipment. COM is reproducible, but, more importantly, is faster (by a factor of 20) than printers and thus will alleviate one of the bigger bottlenecks in computer throughput.

It is expected that many of the large computers of the late 1970's will be part of national computer networks. Networks make it possible for a number of computer systems to share workloads, data, programs, and specialized hardware. A large data base at one installation can be shared by the network community. A user can

[1] This prediction is based on the trend that shows minicomputers and large computers providing more "bang for the buck" than medium scale machines. It is expected that medium scale machines will eventually disappear.

send his data over the network to be processed by a set of application programs being maintained at another location. All network users can have access to such special equipment as the trillion bit laser memory or the ILLIAC IV, an experimental array processor, on the ARPANET.[1]

Just as distributed computing represents a decentralization of the traditional computer architecture, so computer networks represent a higher level of decentralized hierarchical computer organization. Computation is done at the user terminal, at a buffer computer, at a large local computer, and at remote network sites. This hierarchical structure, expected to still require the explicit attention of the 1977 user, will eventually become totally transparant and computers will be a utility used much like the telephone is today.

In conclusion, we predict that the user pressure towards easier-to-use computers and the fact that today hardware is technologically more advanced than software, will make the 1970's a period of assimilation. We will see an evolution and enhancement of computing equipment rather than a revolution characterized by totally new concepts. It is expected that although our understanding of software will increase, the cost of software will dominate the 1970's.

[1] The ARPANET is an experimental computer network being sponsored primarily for network research by the Advanced Research Projects Agency (ARPA) of DOD.

3.0 PROCESSORS

3.1 INTRODUCTION

The traditional computer organization, called the Von Newmann organization after the famous mathematician, John Von Newmann, is based on four functional units: the control unit (CU), the arithmetic and logic unit (ALU), the input/output unit (I/O), and the memory (see Figure 3-1). The combination of a CU and ALU is generally called the central processing unit (CPU). The CU fetches instructions from the main memory, interprets the instruction, and fetches the relevant data while the ALU performs the indicated operation on the data (the ALU may also contain register storage). The I/O unit, under direction of the CU, is responsible for the transfer of information between the memory and the external world. Some conversion between coding schemes may also be done by the I/O unit. The operation of a Von Newmann computer is basically serial since the CU, which can only perform one operation at a time, drives the other three components.

Today a single computer can contain a number of different processors to perform a variety of functions with considerable independence. For example, some computers have one or more I/O pro-

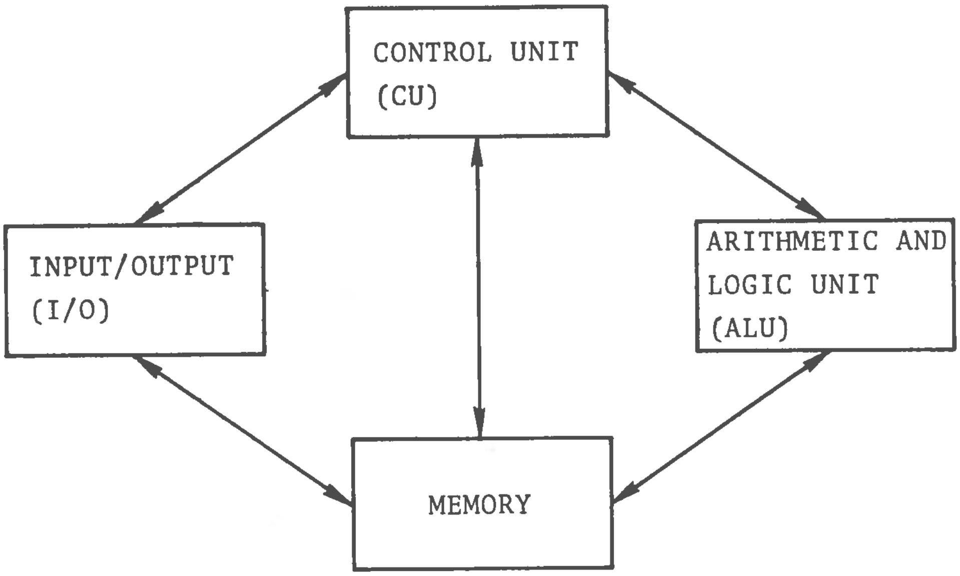


Figure 3-1. Von Newmann Organization of a Computer

cessors that can control the I/O activities of the computer without having to interfere in the activity of the CPU. For example, the Control Data 6600, introduced in 1964, contains ten peripheral processors that are able to direct, monitor, and time-share the CPU. The IBM 3330 disk system found in the 370 line of computers contains what is essentially a processor whose sole purpose is to control the disk system and its interface to the main memory. When two or more of the processors in the computer are CPUs, the system is called a multiprocessor. Originally only the larger computers contained a number of independent processors but this situation is changing in light of the decreasing cost of minicomputers. In the early part of 1972, Memorex announced two new small scale systems, the MRX 40 and 50, in the IBM 360/20 class that are made up of eight individual processors, four for I/O and four for actual processing.

Clearly a number of functionally different processors are emerging. While there is some interest in the role of specialized processors with respect to the architecture of an entire computer system, this report will only look at the specific structure of what must still be considered the heart of the computer, namely the CPU. Attention will be paid to the concept of micro-programming both as it applies to central processors and to peripheral processors.

3.2 PROJECTION OF INSTRUCTION TYPES AND SPEEDS

Two major concerns in the construction of a CPU are the types of instructions it can execute (the instruction set) and the speed with which it can execute the instructions. The second of these leads to the measure MIPS (million instructions per second) that is frequently used to evaluate the power of the computing system.[1] General purpose computers have more or less standard instruction re-

[1] The importance of this measure is the evaluation of modern computing systems that support many concurrent but varied operations is overated. It does not reflect how well a computer performs any of the I/O operations; it does not reflect the adequacy of the instruction set; and it does not reflect the overhead (often 50 per cent or higher) imposed by the operating system. Moreover, with some of the new CPU architectures that will be discussed below the measure is highly dependent on the actual instruction stream being executed.

pertoires that are augmented with instructions for special features that have been built into the machine. For instance computers with hardware pushdown stacks normally have PUSH and POP instructions. The intended application area (business or scientific) may have some impact on the actual choice of instructions. Business oriented machines, for example, have more byte (as opposed to word) oriented instructions. It is unlikely that instruction sets will change in the next five years except through the possible addition of new instructions to take advantage of some of the new CPU architectures that will be discussed below. Micro-programming will have some effect on instruction sets since it will enable users to define new instructions for their own specific applications.

Execution speeds have seen dramatic increases in the 25-year history of computers (Table 3-1). The earliest computer, the Harvard Mark I (1944), was built out of electromechanical relays and had an add-time[1] of 300 msec. The introduction seven years later (1951) of the UNIVAC I, with an add-time of 300 μsec, marked a thousand-fold increase in computer speed. Another thousand-fold increase was achieved in 1964 with the CDC 6600 and its 300 nsec execution rate.[2] Today's fastest machines are approaching 100 million operations per second (i.e. thirty times the speed of a CDC 6600). These high ratios of speed increase have been due primarily to advances in electronic technology. Relays gave way to vacuum tubes; these were replaced by discrete solid state devices; and most recently, the technology has gone to integrated microcircuits.

The question one can then pose is whether or not this fantastic speed increase can continue in the future. If we restrict ourselves to single CPU machines of the classical Von Newmann architecture that operate in a serial manner the answer is no. The absolute limitation in this case seems to be the speed of light although prac-

[1] The add-time is the time required by the machine to add two numbers.

[2] In all fairness it must be pointed out that the CDC 6600 did make use of multiple arithmetic and logical units to achieve this e execution rate.

TABLE 3-1. COMPUTER ADD-TIMES

Computer	Year of Introduction	Add-Time	Speed Improvement Factor
Mark I	1944	300 μsec	x 1000
UNIVAC I	1951	300 μsec	x 1000
CDC 6600	1964	300 μsec	x 1000
?	19??	300 picaseconds	

tical limits are imposed by heat considerations. An electrical pulse can, in 300 picaseconds (i.e. .3nsec.), travel through only about four inches of wire. It can be argued that it is impossible to build a general purpose computer so small that two words can be brought to an adder and processed by the adder anywhere near this four inch constraint.[2] A more sophisticated argument that does not depend on the ability to miniaturize is based on the thermal properties of electronic components. It turns out that one cannot pack components as close as one would like to overcome speed of light limitations because the components generate more heat than can be carried away by fluids in a cooling system.[1] The thermal limit on switching speed is computed to be about 10^{-11} seconds. One might run computers at low (or even cryogenic) temperatures where the thermal problem appears in smaller dimensions and at higher speeds. This does not seem practical in light of the problems that are already being experienced in cooling the computer rooms for the IBM 370 line to normal temperatures. Experts predict that speeds can increase by a factor of only 10 to 1000 beyond the fastest

[1] Today's transitors have an internal power density of thousands of watt/cm^2 at the p-n junction while the maximum heat transfer to fluids at room temperature is only around one hundred watts/cm^2.

[2] This particular argument was brought to the author's attention in a lecture by Commander Grace Hopper at Harvard University in the Spring of 1972.

contemporary circuits before the heat limitations will be reached. (Ref 19, 32).

Winograd of IBM has established the absolute minimum times required for arithmetic operations.[1] He found that that today's computers can add on the order of 60 to 80 percent of the Winograd limit and multiply at 30 percent of the limit. Thus one cannot expect much speed increase through improved logical implementation of adders and multipliers. CPU efficiency can be increased by another factor of 2 or 3[2], thus the classical Von Newmann single stream processor is today within a factor of 3 to 5 of maximum CPU efficiency for any given technology. The foregoing discussion clearly indicates that while improvements in electronic technology will make some speed increases of traditionally organized CPUs possible, a physical limit is rapidly being approached. Computers will be built to keep up with the faster and cheaper memories that are being developed, but the speed increases will be obtained through microprogrammed processors, new CPU architectures, and new computer organizations.

3.3 MICROPROGRAMMED PROCESSORS

While electronic technology and the instruction logic cannot yield much improvement, the power of the single stream CPU will be increased through the ability to microprogram[3] the CPU. Higher pro-

[1] It turns out that a compromise must be made between addition and multiplication speed because the number representations required for fastest addition and fastest multiplication are different.

[2] These figures are derived from observations of how much time the CPU of typical machines is idle.

[3] A machine language instruction is composed of a number of logical operations carried out by switching circuits. In early computers sequences of such operations were hardwired to define the instruction set of computers. More recently these sequences of operations have been controlled by a program residing in a very fast, often ready-only, memory. It is possible to change such a microprogram to alter the computer's instruction set.

gram execution rates will result from the microprogramming of complex instructions for specific applications (e.g. Fast Fourrier Transforms or matrix multiplications). In effect, one microprogrammed instruction will perform an operation that would require a number of standard machine instructions. A small increase in control (microprogram) memory size and a corresponding addition to the microprogram often allows a large reduction of main memory and leads to improved performance.

Microprogrammed machines are being viewed very favorable by computer manufactures because they present the means for emulating one computer with another.[1] Theoretically this makes it possible to run programs designed for one computer on an entirely different machine by merely changing the computer's microprogram. The IBM 370/135 and 370/145 for instance use a console read-only disk from which the microprogram control memory can be loaded, chiefly for emulation purposes. It is expected that reloadable control stores will extend the life of a computer. The system 360, for example, had a six year life span (1964-1970), but the system 370 life could be at least 10 years. Instead of announcing a whole new series of machines, IBM will be able to make advances in the 370 through the simple process of changing the control store routines.

If the microprogram memory is writable, it becomes possible to dynamically (i.e. under program control) change the microprogram and thus alter the nature of the computer. A program in a stream of programs could, before execution, dynamically load an emulator for another machine. Dynamic microprogramming also permits modification of the instruction set during the execution of a program, thus making it possible to define new instructions as they are needed. If, for example, a program finds that it could make use of some higher level operation (perhaps an APL vector primitive) it could write the microprogram memory to make the desired instruction available.

It has been suggested that dynamic microprogramming can be used

[1] Essentially this means that a microprogram is written to duplicate the instruction set of some other computer with the effect of allowing a duplication of that machine's behavior.

to upgrade the operation of the single stream CPU by providing a means for dynamic allocation of the CPU resources[1] (Ref. 10). In a 360/91, for example, the total resources are capable of over 70 MIPS (million instructions per second) but the 91 normally runs at only 6 MIPS. To take full advantage of available resources one could control the flow of data and the communication between the resources within a single micro-instruction. Such a scheme presents a number of tradeoffs between microprogram storage efficiency, performance, and flexibility. Considerable research is required to find the best solution to these tradeoff questions. While dynamically microprogrammable machines are becoming available, it is unlikely that the sophisticated microprograms required for CPU resource allocation schemes will be off-the-shelf items in the near future.[2]

3.4 INCREASING SPEED THROUGH PARALLELISM

3.4.1 SISD and SIMD

The traditional computer organization has come to be known as Single Instruction Stream - Single Data Stream (SISD) because the CPU is executing a single sequential data stream which operates on one data element at a time. Increased performance in SISD processors can be achieved through parallelism or concurrency of operations. In the previous section a scheme was described for achieving concurrency (i.e. simultaneous use of a number of resources) by means of microprogramming. Concurrency in SISD computers has also been achieved by using cache memories and instruction overlap. While dynamic microprogramming normally requires user awareness, the latter two techniques provide implicit parallelism. That is, the

[1] The "resources" of the CPU are adders, multipliers, storage resources, addressing resources, etc.

[2] Further enhancements could be achieved by sharing the resources among a number of processors. Each processor would fetch an operand, prepare the instruction and then request a common execution unit to schedule and execute the instruction. During execution the next instruction could be prepared by the processor. Such a scheme would depend on careful synchronization to make sure that the processors are in different phases of an instruction.

programmer need not concern himself with the fact that a particular scheme is used to speed up execution of his code.[1]

A cache memory is a very high speed buffer memory that contains a portion of main memory. Since most computer programs exhibit a considerable degree of addressing locality (i.e. given that word x of the memory is accessed, there is a high probability that the next word accessed is near x) some access time to slower memory can be eliminated by putting recently used blocks into the cache.[2] The objective of the cache is to make the computer behave as though its entire memory had the speed of the cache. Studies at IBM indicate that the 360/85 achieves 81% of the performance of an ideal machine whose entire memory operates at cache speed (Ref. 21).

Instructions normally involve a number of phases.[3] With instruction overlap execution of an instruction begins not after the previous instruction is completed but rather after the previous instruction has completed one or more phases. This effectively increases the bandwidth of the computer. When a machine uses instruction overlap care must be taken to properly coordinate two consecutive instructions whenever the second depends on the results of the first.[4] The ultimate limitation is that only one instruction can be

[1] The term transparancy has been applied to features of a computer system, both in hardware and software, that are implicit. Here one would say that the parallelism is transparent to the user.

[2] The IBM 360/85 uses such a cache memory. The cache has an 80 nanosecond access time (compared to 1 microsecond for the main memory). The 360/85 cache (16, 24K, or 32K bytes) holds sectors of 1K bytes of contiguous main memory, but these are divided into 64 byte blocks. The blocks are transferred into the cache on a demand basis.

[3] The typical phases are instruction address generation, instruction, fetch, instruction decode, operand address generation, operand fetch, instruction execution.

[4] This limitation becomes especially poignant in the execution of branch instructions. Stone(Ref. 29) has described an interesting organization that uses dynamic register allocation via pushdown stacks to eliminate register conflicts and delays due to conditional branching.

decoded at a time. The scheme of instruction overlap has been used, among others, in the IBM STRETCH, CDC 6600 series, and the 360/90 series.

A more explicit form of parallelism occurs when a single instruction stream is used to process a number of different data streams. This architecture is known as Single Instruction Stream, Multiple Data Stream (SIMD). Basically one instruction is issued for concurrent execution on several sets of operands. Increased performance normally results from the use of more hardware for instruction execution. For the past few years three types of SIMD architectures have been developed and, although some are now in production use, knowledge is still lacking to make full and efficient use of these machines. The three types of SIMD computers are the pipeline processors, the array processor, and the associative processor.

3.4.2 Pipeline Processors

The pipeline processor is in one sense an extension of the SISD overlapped instruction architecture. Here, the actual execution resources (which correspond to the phases that were overlapped) are broken down into a number of stages, say n. To add m pairs of numbers one begins the first addition at time t_o, the second at $t_o + \frac{t_a}{n}$, the third at $t_o + \frac{2t_a}{n}$, etc. where t_a is the time required for one addition. Evidently, at time $t_o + \frac{(n-1)\ t_a}{n}$ there are n additions underway (see Figure 3-2). At that point the pipeline (or adder in this case) is running at full capacity. It is possible to stage most of the processor resources (including the data fetch and store) to achieve such a pipeline effect (see Figure 3-3).

Under ideal circumstances the pipeline computer is able to process instruction at n times the individual instruction execution rate. The problem lies in the fact that to operate in this efficient manner, one must perform a large number of similar operations simultaneously because it is not possible to stage different in-

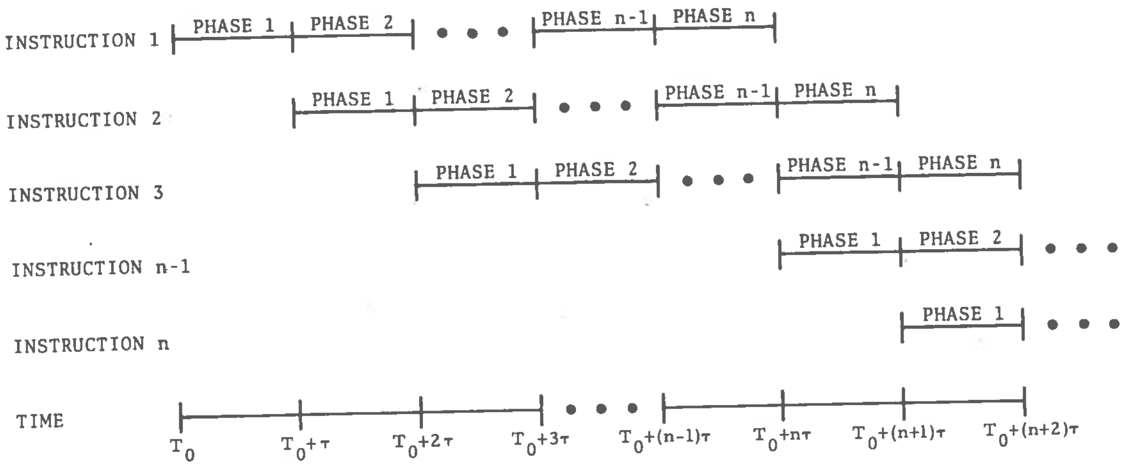


Figure 3-2. Pipeline Execution of an Instruction Divided into n Stage. (All n instruction must be of the same type.)

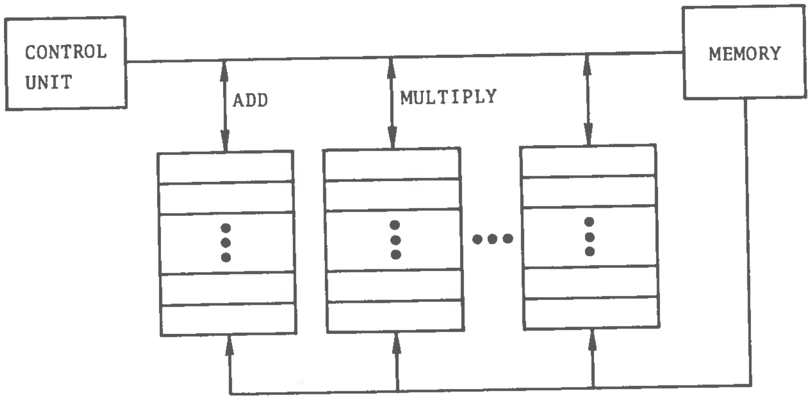


Figure 3-3. Schematic of a Pipeline Processor with Staged Dedicated Resources

struction types (e.g. multiplication cannot be performed in the pipeline for addition).[1] Another limitation is that operands and results for one stream must be stored in consecutive memory locations. To be economical one must keep the pipeline operating at fairly high throughput rates because the hardware required to make the arithmetic and memory units into pipelines is expensive.

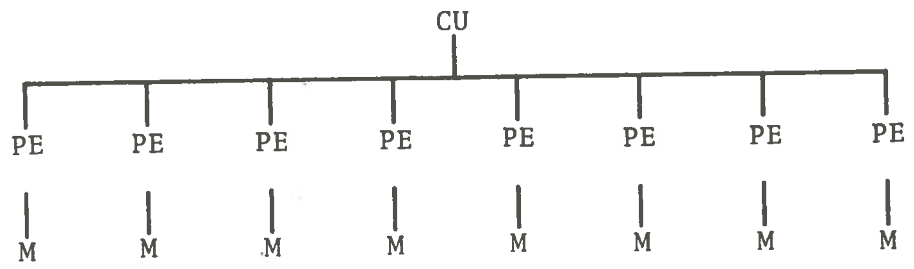
Today, a true pipeline processor, Control Data Corporation's STAR computer is operational and commercially available.[2] More pipeline processors are expected to appear in the 70's, but they will be big machines in the STAR class because pipelines are expensive to build and thus not economically suited for smaller computers. Pipeline processors are not expected to cut into the general purpose computer market because their applicability is limited to very large, usually scientific computations that can be expressed in SIMD terms.

3.4.3 Array Processors

In an array processor parallelism is achieved by replicating the execution resources many times. One control unit controls an entire array of processing elements, each of which has its own memory (see Figure 3-4). An instruction is issued by the control unit (CU) to all the processing elements (PEs). These obtain the data from their memory and execute the actual instruction. It is possible for a PE to mask out and thus not execute the instruction and to apply its own index to an address issued by the CU. Since there is only one CU, the PE's must operate in lockstep, this is, they must all perform the same operation. An array processor is designed for fast computation rather than I/O or control functions. Early indications are that one of the major problems in the use of array pro-

[1] Clearly a pipeline processor could be run in a SISD by feeding it an ordinary mix of instructions, but to achieve a pipeline effect (and thus efficiency) the computer must be given a single instruction to be executed on many pairs of operands (SIMD).

[2] The STAR is one of three or four super computers (in the $25 million class) competing for large scale scientific applications.



CU - CONTROL UNIT
PE - PROCESSING ELEMENT
M - MEMORY

Figure 3-4. Array Processor

cessors will be the task of making data available at rates comparable to the high processing speed.[1] One approach is to use a special I/O processor to provide a data path from high speed disks to the PE memories. Data paths are also provided between the PE's themselves. Operating systems, as well as assemblers and compilers, reside in a separate control computer (not to be confused with the CU responsible for directing the operation of the array processor. In the ILLIAC IV two PDP 10s (model 1077) serve as control computers. It is possible to attach an array processor to any very large computer. Thus, in a sense, an array processor must be regarded as a special rather than a general purpose computer.

Array processors data back to the SOLMON computer built by Westinghouse for the University of Illinois in the early 1960's. This computer had little impact on the computing community, but its

[1] It is estimated that the ILLIAC IV will perform 150 million 64-bit, rounded, normalized, floating point additions per second.

successor, the ILLIAC IV, has aroused considerable interest and has led to the commercial availability of array processors.[1] The ILLIAC IV, a joint venture of Burroughs Corporation and the University of Illinois, will become operational at the NASA Ames Research Center in 1972. The ILLIAC IV and an associated trillion bit laser memory[2] will be widely available via the ARPANET, a national computer network.

One normally thinks of array processors as operating on vectors rather than scalars. Thus the concept of the DO loop in FORTRAN is replaced by vector operation concepts as found in APL. Since computing has been done in a serial manner until array processors became available, most algorithms and techniques developed by numerical analysts and computer scientists have been oriented towards serial representations and problem solutions. Work has been in progress, particularly at the University of Illinois, to develop new solution approaches and new higher level languages to take advantage of array architectures, but these research efforts are still in their infancy. It is expected that this research, combined with the availability of ILLIAC IV to the expanding ARPANET community, will yield a body of knowledge that will make array processors into useful augments to general purpose computers.

3.4.4 Associative Processors

An associative processor, like an array processor, has a number of processors that receive an instruction from a central control unit. Each processing element is tied to one large word of memory (64-512 bits) rather than a memory consisting of many words (see

[1] Burroughs Corporation is actively marketing array processors and predicts 5-10 orders in the next five years.

[2] See Section 5.3 for a discussion of this UNICON memory.

Figure 3-5).[1] A processor executes the issued instruction only if the content of its memory matches a pattern issued with the instruction by the control unit. For this reason the memory of an associative processor is called content addressable. Memories have been built that are content addressable but do not have a processor attached to each word. Small associative memories (8-64 words of 32 or 64 bits) have been successfully used for memory management in paged computers or computers with cache memories.[2] Larger associative memories (256-4095 words) can be attached to general purpose computers for applications that involve operations on data subsets characterized by a particular data property.

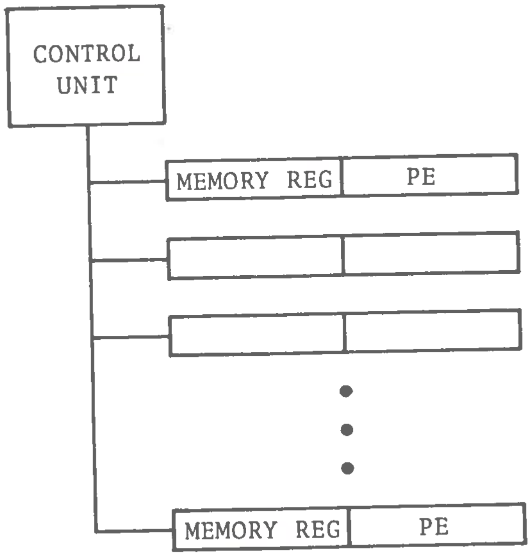


Figure 3-5. Associative Processor

Associative data structures have been used in computer graphics for a number of years, but the required associative memory has generally been simulated by ordinary random access memories (RAMs).

[1] Associative processors are often described as memories with a processor attached to each word.

[2] See the discussion on virtual memories in Section 4.3.

More recently interest in associative memories and associative processors has increased because the emergence of LSI technology[1] and the corresponding price decrease of component circuitry has made possible the construction of associative processors and memories at a reasonable cost. Goodyear Aerospace is now marketing the STARAN, a full fledged associative processor with up to 32 arrays of 256 words, (256 bits/word). As in the case of array processors, research is underway to develop algorithms and techniques for the efficient solution of appropriate problems with associative processors. Although STARAN is a stand-alone (special purpose) computer, it is expected that associative processors and associative memories will be used as subsystems of general purpose computers that can provide the I/O operating system, and assembler and compiler capabilities for which the associative processor is not particularly well suited.

3.4.5 Other Arrangements

Most SIMD computers do not strictly qualify for one of the three classes (i.e. pipeline, array, and associative) that have been discussed in this section. Instead there is really a continuum of possible architectures with ideas borrowed from each architectural type. The STARAN, for instance, while advertised as an associative processor is made up of a number of parallel arrays and thus has many of the properties of an array processor. Another computer, the Advanced Scientific Computer, that has a SIMD architecture but doesn't fall into any of the three classes, will be discussed in this section. The All Application Digital Computer (AADC) being designed for the US Navy also exhibits some SIMD properties, but, because of its novel system architectural approach, it will be described in Section 6.2.1.

The ASC is a super computer being built by Texas Instruments (TI) to compete with the CDC STAR. This machine, rated at 16 times the power of a 360/91 can operate at speeds up to 66 MIPS. An ASC is scheduled

[1] LSI stands for large scale integration. This refers to the miniaturized integrated circuits that are starting to be used for computer construction (see Section 4.2).

for customer delivery in July of 1973, but TI already has a working version procession seismic data in its Austin Texas Laboratory. The ASC has a 160 nsec semiconductor memory of one million 32 bit words. The CPU contains four arithmetic units capable of both scalar and vector operations. The vector operations give it the appearance of an array processor, but the machine also has a pipeline for streaming data and instructions. While there is only a single instruction stream consisting of a mixture of scalar and vector operations, up to twelve instructions can be processed at once. The ASC has, in addition to the CPU, a channel processor and a peripheral processor.

3.5 MINICOMPUTERS

A minicomputer is, as the name implies, a small general purpose stored program computer. It is characterized by (a) a short word length (19 bits or less, but normally 16 bits), (b) limited instruction set (no floating point operations, no memory to memory move instructions), (c) fewer high speed registers, (d) main memory in small modules (4K to 8K words), (e) a limited addressing capability (due to the short word size), and (f) simpler I/O capabilities. A minicomputer is normally used in a hands-on manner by one person at a teletype.

The past few years have seen an immense rise in the popularity of the minicomputer. The market for minis is fast moving with over ten firms in active competition. This rise in popularity, which is expected to continue, is due to a marked downward price trend[1] coupled with the realization that a mini can effectively handle a large class of applications. Until now the available software for minis has been limited - most users have had to develop their own extensions of rather primitive operating systems. Unfortunately,

[1] Minimal configuration (i.e. 4K words of core and no peripherals) minis that cost in the $15-20K range a few years ago are now available for $3-4K. This downward trend in prices will continue well into the 1970's.

this has led to a proliferation of incompatible software[1] which has negative implications with respect to program transferability.

Competition among manufacturers is such that in order to get a solid customer base more emphasis is being placed on standardization and good software with upward extendibility. People with large system experience are becoming involved with minis and are applying their knowledge that has been gained there in the past 15 years.

The minicomputer boom is predicted to continue as long as prices keep decreasing and as new applications are found. Prices of the mainframe and of memories[2] will continue to decrease (drop by a factor of four is expected in another 5-10 years), but perhaps more importantly, the cost of minicomputer peripherals is just beginning on a sharp downward trend. The latter means that the power of low cost mini systems will drastically increase in the next few years because it will become feasible to add disks, tape drives, and printers to a mini without getting into the $50-$100K price range. Another factor that will contribute to the mini boom is the increasing availability of microprogrammable minis achieved with very fast read only memories (ROMs) or with dynamically writable memories. While cost is decreasing, advances in large scale integrated (LSI) circuitry[3] are also leading to faster minicomputers. While average add times today are on the order of one microsecond and the fastest minis are at 300 nsec, it is predicted that by 1977 the average add times will be in the 100 nsec range. A survey by Auerbach (Ref. 1) reports the following breakdown of 1970 dollar minicomputer sales:

[1] For example, least a dozen different assemblers exist for the DEC PDP-8 computer.

[2] See the discussion of the changing memory cost-size curve in Section 4.1.

[3] See Section 4.2.

45%	industrial and process control
3%	peripheral control (COM, OCR, line printers, etc.)
20%	communications control (concentrators, switching)
10%	computations
22%	data acquisition and reduction

It is expected that the second, third, and fourth categories will see a particular rise in the next few years. The lower cost of very small systems with cheap memories and microprogramming capability will lead to a much greater use of minis for peripheral control. Functions formerly requiring costly use of CPU or central I/O processors can be performed in the peripheral itself.[1] Minis that control peripherals permit running of diagnostics without burdening the central computer.

Communications is playing an ever increasing role in large computing systems (see Section 6.3); it has been found that minicomputers are ideally suited for data concentrator and message switching roles. The added intelligence given to these system elements through use of a mini can result in considerable overall system efficiency gains. The ARPANET[2] has been able to make very reliable and efficient use of communication lines by using a minicomputer-based store and forward message network.

Finally, the expected standardization, the availability of better systems and applications software, the lower system (i.e. CPU + memory + perpherals) cost, and the increased efficiency resulting from dynamic microprogramming will lead to increased use of minicomputers for general purpose computation. Minicomputers are beginning to become much more modular than in the past. It is possible to go to a vendor with a list of features (interrupt structures, arithmetic packages, I/O capabilities, etc.) and to have him assemble the desired mini by inserting the appropriate plug-in cards

[1] See the discussion of intelligent disks in Section 5.1

[2] A network of computer systems (see Section 6.3).

into a mainframe.[1] The mini owner will be able to buy plug-in packages that provide specialized software in the form of a read-only-memory or perhaps a microprogram. The implications of these trends in minicomputer architecture are that the computers can be more easily tailored to specific applications, and, because of increased efficiency in solving a particular problem, additional cost gains will be realizable. Waks and Kronenberg (Ref. 33) predict that these changes will be accompanied by the rise of a "minicomputer extension" specialist, versed in both software and hardware, who will be able to put together such a minicomputer hardware-software system for a customer.

The minicomputer, when used in communications and peripheral control, is essentially a building block for a larger computer system. Other building blocks that will increasingly be incorporating minicomputers are front-ends for time sharing systems and intelligent terminals, graphics processors, and special purpose processors in large systems.[2] Minis are currently being used as stand-alone time-sharing systems, but, while well suited to functions of editing and on-line calculations, these systems are inadequate to handle jobs requiring extensive computations. On the other hand, heavily interactive jobs make inefficient use of large time-shared systems where core storage is at a premium and page swapping is expensive. Thus, it is reasonable to put a minicomputer between the user terminal and the large computer to handle those jobs for which it is well suited. Currently one must run one's entire job on one computer or the other, but research in canonical representations of computations promises to provide the ability to switch a job in progress from one machine to the other whenever the job characteristics warrant such a transfer.

[1] The PDP-16 being marketed by Digital Equipment Corporation is configurable from a set of about 20 asynchronous modules called register transfer module (RTM). For a discussion of this concept see References 4 and 5.

[2] See Section 6.2.3 for a discussion of minis as special purpose processing elements.

Similar arguments about the relative cost effectiveness of minis and large computers for particular types of processing have led to the incorporation of minicomputers into on-line terminals ("intelligent" terminals) and into dynamic graphics consoles. A number of years ago it became fairly clear that the graphics processing portion of an interactive graphics application in a multiprogramming time-shared environment was best performed by a satellite minicomputer, but the cost of such a configuration was prohibitive before the massive reduction in mini prices.

Intelligent terminals are usually made up of a minicomputer, a CRT, a keyboard, a small line printer, and a casette tape. They can be used both as stand-alone computers (calculator, data storage device, etc.) or as arms to a host computer. Some intelligent terminals are also designed to act as a host for other keyboard-CRT or teletype terminals. Currently intelligent terminals are used primarily for data entry and for printing the numerous forms required in the manufacturing and distribution industries. Their use is expected to increase and their role will expand as the concept of a computer utility gains a foothold.[1]

A final use of minis is in the construction of a minicomputer multiprocessor or network as a replacement for current medium-to-large scale computing systems.[2] The low cost of minis, the ease with which they can be microprogrammed, and the availability of crossbar switches or multiport memories (to allow accesses by any mini CPU to any memory module) has aroused interest in this architectural concept for future systems. Two specific designs have

[1] "Computer utility" refers to the idea that a computer system is made up of a hierarchical network of computers, with each level of the hierarchy performing the processing for which it is best suited. Users access the utility via terminal systems (ranging from simple teletypes, possibly connected to a time-shared minicomputer, to intelligent terminals) that feed a local computer which in turn is part of a network of computers (see Section 6.4).

[2] See Section 6.2.2 for a discussion of the architectural aspects of minicomputers multiprocessors.

been proposed by researchers at Carnegie Mellon University and at Newcastle-on-Type (Ref. 3). The chief benefits claimed for such systems of cooperating minis are lower operating system overhead, dynamic reconfigurability, and lower system cost. The MRX40 and MRX50 (see Section 3.1), although not reconfigurable minicomputer networks, are microprogrammed multiprocessors made up of four CPUs and four I/O processors. One interesting aspect of these two machines is their ability to emulate (by microprogram) a 360/20 in one of two multiprogram partitions and to run in normal mode in the other.

3.6 CONCLUSIONS

In this section we have tried to show that the fantastic computer speed increases of the past twenty years are approaching practical limits. To overcome these limits parallelism will have to be incorporated into the processors of the future. One already popular approach is to use multiprocessor configurations where several independent processors share storage at some level.[1] A number of approaches have been proposed to obtain parallelism within a single CPU. Instruction overlap and microprogramming for dynamic processor level resource allocation are ways of increasing the efficiency of single instruction stream, single data stream (SISD) machines. Three approaches are being taken to obtain additional throughput via the more explicit parallelism of single instruction stream, multiple data stream (SIMD) CPU architectures. All three- pipeline processors, array processors, and associative processors - only operate at high efficiency for specialized types of applications. Pipelines and array processors operate most effectively when one operation is replicated many times (thus they are well suited to array or vector type operations). Associative processors are best suited to operate on content-defined subsets of a

[1] Multiprocessors are discussed in more detail in Section 6.2.

data array.[1] In the next 5-10 years SIMD architectures will be finding a place in computing systems and expertise in their efficient use will increase. It is expected that they will be used as special purpose processors and not as CPU's for general purpose computing systems.

The CPUs of general purpose commercial computers will continue to be of the SISD type, but they will be designed for higher throughput rates and better overall systems performance. This will be achieved through more effective software, through the use of virtual addressing schemes,[2] and through a greater distribution of peripheral processing functions. The larger CPUs will use instruction overlap and cache memories. Most CPUs will be microprogrammable but not so much to increase efficiency as to facilitate emulation of predecessors and of competetitor's computers. Manufacturers of computers above the mini class will not encourage user modification of the microprograms and, as a result, it is unlikely that the larger commercial CPUs will be dynamically microprogrammable.

In the next five years the rapid downward trend of minicomputer prices will continue. This coupled with microprogrammability and decreasing prices of small memory modules will lead to increased use of minis over a larger range of applications. Of particular interest to this forecast will be the use of minis as building blocks in larger computer systems. Minis will be found in terminals, peripheral controllers, data concentrators, message switching units, and special purpose front end data processors. General purpose minicomputer systems are predicted to become much more popular in the

[1] In transportation-related applications array processors are particularly well suited to simulations where many items (e.g. vehicles, radar signals, etc.) must be identically processed in each time quantum. Associative processors could be used to advantage in Air Traffic Control (ATC) conflict detection where a subset of all planes must be selected on the basis of proximity to some location. Associative processors or memories are also of use in graphics applications where many of the data structures have associative character.

[2] See Section 4.3.

next five years because the increased availability of sophisticated software and the emergence of low price minicomputer peripherals will make this mode of computing extremely cost effective.

4.0 PRIMARY MEMORIES

4.1 INTRODUCTION

Memories, just like processors, have seen dramatic changes since the completion of the Harvard Mark I in 1944.[1] The Mark I's memory consisted of a set of mechanical wheels whose position represented a digit between 0 and 9. ENIAC (1946), the first stored program computer, had a memory made of storage tubes (triodes and pentodes). The EDSAC (1949), with a cycle time on the order of one millisecond, used mercury delay lines[2] to store data and instructions. Later versions used cathode ray tubes[3] as memories and thus were able to reduce access time to the order of microseconds. An MIT project, begun in 1947, produced the Whirlwind I with the first electromagnetic core memory. Although the highly successfully UNIVAC I of the early 1950's still used magnetic delay lines, by 1953 it became clear that electromagnetic cores would be the dominant memory technology. This type of memory was used throughout the 50's and 60's and is still used in most commercial computers made today - indeed, the words core and primary memory have become synonymous.

The advent of integrated circuits (IC) and the subsequent miniaturization that led to large scale integration (LSI) has produced a technology that is currently competing successfully with magnetic cores and that will eventually replace cores. Whereas the cost per bit of core memories decreases with size, that of LSI memories is

[1] The Mark I was an electromechanical, but not stored program, computer built by Howard Aiken of Harvard in cooperation with IBM.

[2] These are tubes filled with mercury. Electronic pulses coming into the tube on a wire are transformed into mechanical vibrations by a crystal. The vibrations, stored in the tube, define a bit pattern.

[3] Electrostatic bit patterns were stored on the face of the tube.

more or less linear. The economic attractiveness of smaller memories resulting from this relationship will have a profound impact on the architecture of commercially available computers of the 70's. Rather than building a computer around one large memory, it now becomes possible to have several smaller memories without an increase in cost. Some are for specific applications (e.g. to give a memory to terminals, to store information for peripheral control, as buffer memories for peripherals, to store microprograms, to perform address translation for virtual memories by associative techniques, etc.) while others provide the main or primary memory function. It becomes feasible to investigate the use of a number of parallel memories or of a hierarachical memory structure (such as that of the 360/85) based on very fast cache memories (40 nsec.) backed by a larger (10^5-10^7 byte) high speed memory (on the order of 600 nsec).

This section will begin with the technical and economic details of semiconductor technology in the late 60's and 70's as it applies to primary memory construction. The following sections deal with the memory concepts that have been tried in prototypes and a few advanced commercial systems which are expected to become dominant in the 1970's. A related discussion of trends in system prices (CPU + memory + peripherals) can be found in Section 2.

4.2 CORE VS. SEMICONDUCTOR MEMORIES

Semiconducting devices exhibit a number of characteristics that make them attractive as memory elements. Chief among these are high speed, high density, high level output, and low power consumption. The devices are interesting from the architectural viewpoint because they permit the combination of logic and storage functions on the same chip[1] and thus address decoding circuits can live together with the memory cells. This characteristic has made it possible to readily implement associative memories. Information

[1] The circuitry in IC (integrated circuit) technology is integrated into small chips. The more advanced techniques for miniaturization permit LSI (Large Scale Integration) chips that are .05" square and contain over 50 gates.

can be read out of semiconductor memories without destruction of the information whereas in core memories (and any memory that is based on magnetic phenomena) reading destroys information and thus every memory access requires a subsequent information refresh cycle. Not only is reading of a word in an LSI memory nondestructive but, once a word has been addressed, it remains available for reading, writing, or both as though it were a register in the CPU. LSI memories can be constructed with an inherent bit mask, directly loadable from a stored work, to allow selective reading and writing of bits in a memory word. Thus a masking operation can become part of memory read and write rather than a separate, time-consuming operation. Advances in the construction of integrated circuits over the past ten years promise batch fabrication techniques that will lead to very low cost. Finally, the drive, sense, control, and interface logic of all of today's memory devices is made from semiconductors, and thus the same expertise can be applied to both the design of the memory and the various aspects of its interface.

Two major integrated circuit technologies, one based on the bipolar junction transistor and the other based on metal-oxide-semi conductor (MOS) field effect transistors, are in use today. Bipolar ICs are capable of very fast switching speeds and thus are used for very high speed memories where the larger device area and higher power consumption can be tolerated. MOS ICs are easier to make (and thus cheaper), smaller (by a factor of 4 to 5), and require less voltage, but they have slower switching speeds. In the long range it is expected that bipolar technology will not be used for memories above 10^6 bits while MOS memories may reach 10^8 bits.

Semiconductor memory devices are made more attractive by a high degree of miniaturization (large scale integration). LSI technology permits the placing of many thousands of individual devices on a single silicon chip. Efforts are just now starting to provide some degree of standardization in chips and in combinations of chips[1]. This would permit, at the expense of some excess in

[1] Raytheon, for example, is building up a library of one inch square Raypaks containing 6 chips apiece. A module of 6 Raypak contains 1800 gates but work is underway to get a 6000 gate packing density. Raytheon builds the AADC (All Application Digital Computer), a sophisticated general purpose computer, from 10 chip types and 5 Raypak types.

logic, a more or less off-the-shelf construction of computing elements without the need for tedious chip design. This puts the basic element in computer design at a much higher level and thus will lead to lower design costs. The parallel can be drawn with software design using higher level programming languages. The use of a limited number of standard chips will also lead to lower chip costs since the high chip development cost can be spread over a much larger volume for any given chip. Another advantage of constructing computers from LSI chips is that fault tolerance can be built into the chips by using proper coding schemes. Although LSIs major commercial impact so far has been on memories, microprogrammable minicomputers are appearing on the market with MSI (medium scale integration) logic and with LSI microprogram read only memories. Chip and package standardization will strengthen this trend towards miniaturized logic.[1]

In 1970 commercially available bipolar memory chips used 256-bit chips with speeds in the 150-200 nsec range at a cost of 4¢/bit. Now (1972) bipolar chips contain 1024 bits and speeds are in the 100 nsec range. Similarly, MOS chip densities increased from 1024 bits/chip in 1970 to 4096 bits/chip in 1972. Read-only-memory (ROM) chips are now available with 8192 bits.[2] (Cores, by comparison, have a density of 1000-3000 bits per square inch.) This trend of doubling chip density every year is expected to continue for at least another five years.

To see how this impacts cost one must consider not only the cost of manufacturing a chip, but also the cost of chip packing and

[1] Intel Corp. has already announced an entire computer (MCS-4) on only two chips. The MCS-4 has a 4-bit parallel CPU with a slow 10 μsec cycle time, 45 instructions, a 256 word (8 bit) control memory, and a 4 bit I/O port. The chips cost $63 apiece. (See Datamation, January 1972, p. 5).

[2] LSI memory chips also contain decoding logic to circumvent the problem of an excessive number of pins and the associated costly interconnections.

and the part number of the chip.[1] Dennis and Smith (Ref. 7) give the total chip cost as:

$$\text{chip cost} + \text{chip package} + \frac{\text{part number cost}}{N}$$

where the chip cost and package cost are both in the $2.50 range. As chip development techniques improve further, the ratio of package cost to chip cost will probably increase. Since the number of memory chips of any one kind produced is quite high and since part number cost will eventually decrease from tens of thousands of dollars to the order of $1000, this implies that the chip density and cost per bit will be almost inversely proportional. It should be pointed out that the $2.50 per chip refers to maximum yield density configurations and that higher performance cost considerably more.[2]

The development of a particular LSI chip is both costly and lengthy, thus it has been proposed that ROM chips be used as a replacement for logic design. One application is to do arithmetic by table lookup. Thurber and Berg (Ref. 30) have described a number of ways of constructing logic modules for both specific and general functions from ROMs. This interesting concept will continue to be investigated, but it is not likely to have any impact on computer technology in the 1970's. There is a greater likelihood that ROMs will be used to replace gating functions in microprogrammed machines and thereby reduce the cost of the control portion of the computer.[3] It is clear from the above discussion that the cost of IC memory will decrease by close to a factor of two per year for several more years. This will be accompanied by a more gradual increase in memory speeds. Despite this, cores today capable of 500 nsec speeds

[1] Part number cost is the cost of generating and handling a particular chip type. It is the cost, after logic design, attributable to the uniqueness of the part.

[2] To get some idea of the relation of speed to cost, the follow 1971 full system level memory costs per bit can be cited: <60 nsec = $.50; <200 nsec = $.10; <500 nsec = $.05; <1 μsec = $.03; >1μsec = $.02.

[3] See (Ref. 6) for a discussion of this concept.

at costs of 1¢/bit and predicted to provide 300 nsec speeds in the near future, are expected to be competitive well into the 1970's. AMPEX recently (5/72) announced the development of a new temperature independent material (TIM) for computer core memories which is capable of operating from -25°C to 100°C. This will lower costs by at least 25 per cent since the temperature compensating electronics of today's cores can be eliminated.[1] Another factor keeping cores alive is their non-volatile nature. That is, while loss of power destroys the content of a semiconductor memory, cores retain their magnetization (and thus their information) without power.

Two other magnetic memory technologies, thin film and plated wire, were considered for development at one point, but both are being replaced by ICs or improved cores. The 360/95 used a thin film memory of one million bytes with a 120 nsec speed (effectively 200 nsec due to the large size) and the UNIVAC 9300 had a similarly sized plated wire memory. Plated wires were expected to provide sub 100 nsec memories of 100,000 bytes at less than 1¢/bit by the mid 70's, but in light of recent LSI developments this performance will not be competitive. While the potential of semiconductor memories has been evident for a number of years, until recently, it was thought that semiconductor memories would play a limited role in the primary memory market until the late 1970's. They were projected to be used only in the very high speed memories for cache, microprogram stores, and associative memories. This outlook was considerably modified when IBM announced that 370/135 and 370/145 computers would have IC memories.[2] The 370/155 and 165 are still produced with core storage, but independent memory manufactures are

[1] Although AMPEX plans to use the material for all the cores it produces, this material will be particularly useful in minicomputers used in uncontrolled environments such as factories and warehouses.

[2] Most IC memories today that require very high speeds do not use LSI. The 370 main memories (200 nsec) for example, use 128 bits/chip with 4 chips to a package (called MSI for medium scale integration).

offering monolithic[1] semiconductor memory replacements. Advantages cited are that they generate less heat than cores, draw less power than cores, and are smaller than cores (a 2 megabit semiconductor 370/155 memory uses 1/4 the room of the original equipment core memory). There is some debate as to the relative reliabilities of the two types of memory. One manufacturer, ITEL, feels that the IC technology is more reliable, while another, AMPEX, is sticking with core replacements for original equipment on the 370/155 and 165 for reliability reasons. In any case, the 1972 state-of-the-art makes the two technologies competitive.[2] This state of affairs will change in the next few years because, although some core improvements are still possible (as evidenced by AMPEX's TIM), semiconductor technology offers considerably more room for future progress.

4.3 MEMORY ORGANIZATIONS

Present generation computers are built around one large main memory because with magnetic core, thin film, or plate wire memories, low costs are obtained by driving large magnetic arrays with a small amount of electronics. As indicated earlier, this size cost relationship has changed with the introduction of semiconductor memories but the single memory is a precept of today's operating systems and of many computational algorithms, so that one can expect a lifetime of at least another 5-10 years for this memory architecture. While LSI will lead to a distribution of memories to more locally autonomous functional processing units, the computations of program will rely on the single central memory. Some commercial computers may multiplex the main memory to achieve increased speed but today's knowledge of the efficiency of alternate memory architectures and the associated computational algorithms is insufficient to predict whether the single memory or some multiplexed

[1] Monolithic refers to the fact that several active circuits (flip flops, logic gates, etc.) are contained on one silicon chip.

[2] To give the reader a feeling for costs one might point of that AMPEX sells a 512K byte module for $225K (about 5 1/2¢/bit).

memory will turn out to be the most effective organization in the long run.

Designers of computers and computer software strive for two basic qualities in the traditional main memories, namely size and speed. Cost considerations require compromises in hardware design, thus a number of schemes to make memories look both fast and very large to a user have been devised. Fast memory operation can be obtained by using a two level memory consisting of a small amount of fast bipolar memory and a larger backing store of slower memory (see Figure 4-1). In current systems, the fast memory is in the 50-100 nanosecond range while the large memory is in the 500-1000 nanosecond range. The high speed memory, called a cache (or buffer) memory, contains a copy of the most recently used portions of the main memory. The cache memory depends on the notion of locality of program execution for its effectiveness (see Section 3.4). To the user, this cache memory is entirely transparent. That is, he sees only one memory that is operating at speeds much closer to that of the cache than that of the actual main memory.

An ideal cache memory is essentially an associative memory, each word of which contains both the address and the contents of a word in the main memory.[1] Data in the cache is found by using the main memory address for an associative lookup. A truly associative cache memory of this sort is still quite expensive and does not offer any "look ahead" capability.[2] Thus, cache memories that have been implemented[3] use a cache that is organized into a number of small blocks. The cache itself is no longer an associative memory, but is accessed through an associative memory.

[1] See the discussion of associative processors and associative memories in Section 3.4.3.

[2] The strictly associative scheme is called a "look behind" scheme since only words already used are placed in the cache. The transfer of a block is termed "look ahead" since one expects words following the one currently addressed to be used soon although they have not been used yet.

[3] IBM uses caches in its 360/85, 360/195, 370/155, 370/165, and 370/195.

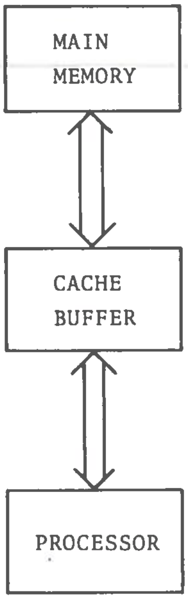


Figure 4-1. The Cache Concept - a small, high speed buffer between the processor and a slower main memory makes the main memory seem to operate at speeds near that of the cache.

Access to the block oriented cache requires not only an associative table lookup, but also some form of hardware address translation. The efficiency of a cache memory depends on the block size and on the speed and size ratios between the cache and main memory. There is of course a dependence on the actual computer application, but designers can base their judgements on average programs.[1,2] It has been found that the efficiency of a cache decreases sharply as the number of concurrent users increases. Typically a cache operates at 96% of its rated speed for one user, at 75% for five users and as low as 25% for large numbers of time share users.

Normally, caches are used only in larger computers and, because

[1] An average program is a vague notion that has received considerable attention. It will not be discussed here because it is beyond the scope of this forecast.

[2] A cache can become more effective if it favors the more frequently used supervisory programs.

of design tradeoffs made for certain performance objectives, even some larger computers (e.g., IBM's 370/135 and 370/145) opt for faster, single-level main memories. Others, like the Control Data 7600, do not have a cache but use a fast, small internal buffer memory through which information passes under software control. A cache can only be justified when the combined cost of the cache, main memory, and extra hardware is less than the cost of a single memory that provides equivalent throughput, thus the absence of caches in smaller machines with small memories. As LSI memories in the 100-200 nanosecond range become cheaper and as more general purpose computing systems offer foreground time-sharing (and thereby increase the number of concurrent users), it is likely that caches will disappear altogether.[1]

The control design of a machine also has some effect on the cost/performance analysis of a cache. While in the past, cost considerations led to the use of cheaper read only memories for microprogram control stores, the current trend is toward writable microprogram memories. It is now possible to use a high speed main memory for both its normal application and the dynamic microprogram. Thus, a larger investment in main memory speed and size can be justified. This argument ~~is~~ more applicable to small to medium scale machines where the execution rate requirements do not demand microprogram stores in the sub 100 nanosecond range. The current rule of thumb is to use caches in machines that have cycle times of 100 nanoseconds or less and memory of one million bytes or more.

To make a computer's main memory appear larger than it really is, one uses a scheme whereby the address space of the user is independent of the actual addresses in the physical memory. The objective is to create a "virtual" memory, only bound in size by a computer's addressing scheme, parts of which reside in the main memory and others on a secondary storage medium (normally a disk). In effect, the

[1] It has also been argued that multi-level caches will survive because speeds of bipolar (cache) and MOS (main) memories are improving at the same rate.

main memory serves as a buffer for the large address space of the secondary memory just like the cache is a buffer for the main memory. The main and secondary memories consist of a number of fixed size "pages" which are in the 1000 byte range.[1] An address issued by the user consists of two parts, a page address and a displacement within the page. A small, but very fast (50 nanosecond) associative memory contains the absolute address of the pages (i.e. that issued by the user) together with the main memory address of the page. To access data it is first established whether the desired page is in main memory.[2] Once the page is in the memory, a hardware address translation takes place to compute the main memory address from the page displacement address.

A second, less popular scheme for virtual memory implementation is based on variable size segments where a segment is normally a complete content or function-related block. This breakdown into logically distinct units each of which can be given a unique name has advantages from a programmer's viewpoint. Segments make it easy to deal with programs as modular units and with variable size data structures and they facilitate protection and sharing of program and data modules. The direct physical implementation of a segment-oriented virtual memory leads to memory management problems that are avoided by a fixed size paging scheme. Some virtual memory systems have been constructed to have both logical segmentation and physical paging, and thus the advantages of both, at the expense of a more complicated address translation mechanism. It is possible to use both a virual memory and a cache memory together. With such a scheme, rather than going through an address transfor-

[1] The main memory/secondary memory access time ratio is on the order of 1:1000 (rather than 1:10 as in the case of cache/main memories) and thus the design criteria are quite different. The optimal cache block, for example, is in the 32 byte range.

[2] Usually, a virtual memory system is based on demand paging. That is, a page is brought into the memory whenever it is addressed and one of a variety of algorithms is used to decide which page currently in the main memory is to be replaced.

mation for each memory access, one can limit the transformation requirement to cases where the desired word is not in the cache since the cache can contain the page-displacement address. The user of both types of virtual memory systems need only think in terms of virtual (i.e. page-displacement) address - the whole memory mapping process is transparent to him.

Associative memories can be particularly useful both for cache and for virtual memory applications in multiprocessor systems (i.e. systems with one main memory or several main memory modules and a number of processors that share the common memory). When a number of processors share a single memory, interconnection delays tend to slow the system. Such delays can be significantly reduced if an associative cache buffer memory is put between each processor and the main memory (see Figure 4-2).[1] An associative memory for paging control can be constructed to contain, besides page identity and main memory address, some bookkeeping information to be used in replacement algorithms. Page priority information can be quickly compared in this memory. Another associative memory might be used to manage the various tasks that are being executed by the multiprocessor (see Figure 4-3). Essentially, this memory is a convenient place to keep frequently used information that must be accessed on the basis of data content (e.g. task priority.). These two types of associative memories make up a "hardware executive".

Another type of memory that is getting increasing attention, and one that will probably find considerable use in the virtual memory systems of the future, is the block oriented random access memory (BORAM). In a BORAM savings are achieved by building logic that only permits access to blocks of words rather than each individual word. Since virtual memories normally deal only with complete pages a BORAM is ideally suited as a large backup memory for either a cache or a fast main memory in a virtual memory system. The concept of a BORAM is not new but has not caught on, probably because its applicability was rather limited. More attention will be placed on

[1] Meade(22) suggest that a multiprocessor of cache buffered minicomputers offers an attractive cost/performance ratio.

BORAMs as more and more computers are built with virtual memories. Whether or not the BORAM will find continued use will depend on the cost effectiveness of hierarchical main memories for large computing systems. This will be determined largely by the future of a cheap and reliable MOS technology and on trends in the overall computer architecture. A cheap but fast MOS chip[1] would favor the single random access memory (RAM) since hierarchical memories derive their cost effectiveness from the speed and price differential between the various levels of memory. The BORAM is most likely to be useful as a large common memory backing individual processor memories in a multiprocessor system. In any case, IC BORAMs are not expected to be available in the commercial systems of the next five years.[2]

4.4 CONCLUSIONS

Magnetic cores became established as the primary memory technology in the mid 1950's and retained this dominance throughout the 1960's. Today the place of cores is being challenged by semiconductor technology and the promise of faster and cheaper LSI memories. Core memories will remain competitive into the second half of the 1970's, but will not survive in the long run because of the low cost of semiconductor memory.

LSI memories will be commonplace by 1973 and MOS-LSI will be making its impact in the late 70's. Today core and semiconductor costs are quite competitive, but by 1975 semiconductors will be producible for one fourth the cost of equivalent cores. Despite this, there are indications that for a while the prices of semiconductor memories will stay close to core levels for profit. Production costs will eventually reach a point of diminishing returns since, as cost and selling prices decrease, the break-even quantity goes

[1] Currently MOS-LSI technology suffers from reliability problems and it is not yet clear whether or not improvements are forthcoming. MOS ICs as a whole now have over 25% of the total digital market and are expected to have over 40% of that market by 1975.

[2] See Section 5.3 for a discussion of a smaller, slower speed BORAM based on the domain tip (DOT) technology.

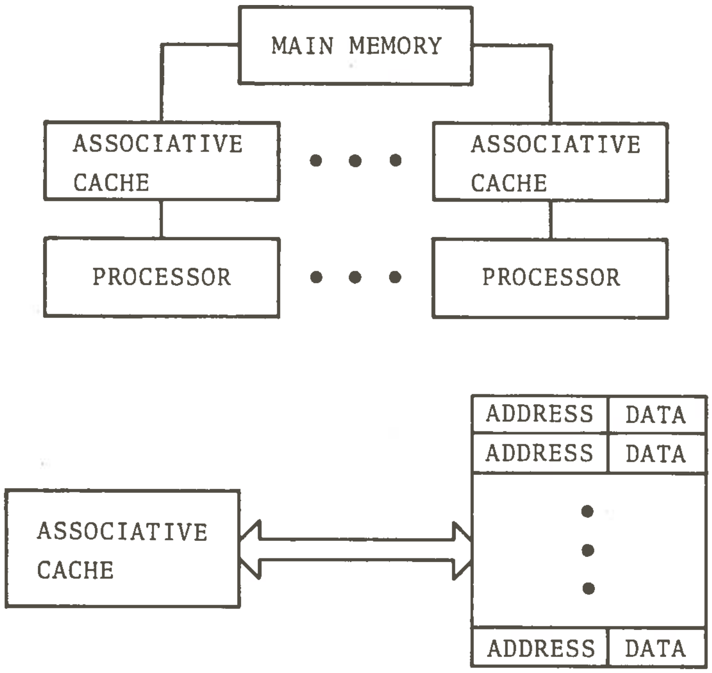


Figure 4-2. Use of Associative Cache Memories in a Multiprocessor Systems

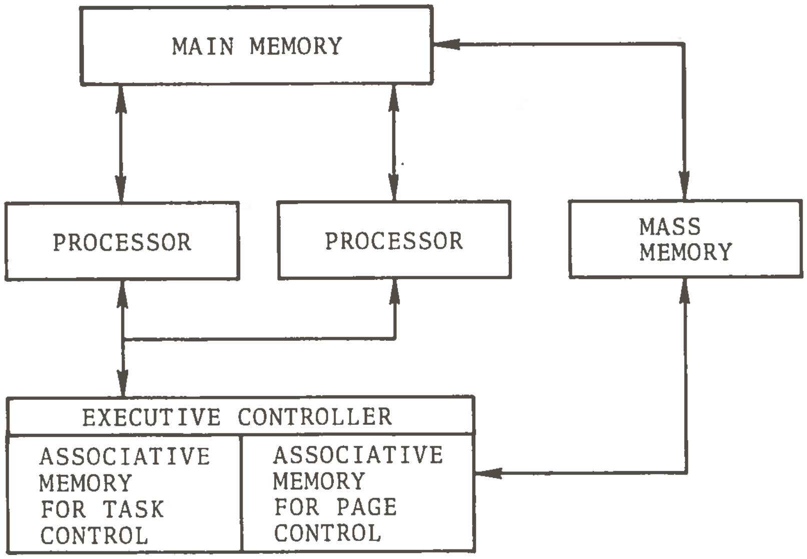

MAIN MEMORY
PROCESSOR
PROCESSOR
MASS MEMORY
EXECUTIVE CONTROLLER
ASSOCIATIVE MEMORY FOR TASK CONTROL
ASSOCIATIVE MEMORY FOR PAGE CONTROL

Figure 4-3. Associative Memories used for Executive Control in a Multiprocessor System

up sharply. Before this occurs, however, there will be a significant drop in memory prices. For example, a memory that today costs $120K will, with equivalent speed and size, cost less the $5K in the late 1970's. International Data Corporation predicts[1] that by the end of 1976 the main memory market will grow significantly, with a 259% increase in total megabits installed. Due to price drops the total value of this equipment will grow by only 92% from $8.4 billion at the end of 1971 to $16.2 billion at the end of 1976.

Improvements in LSI memory cost will be achieved primarily through increased miniaturization. Today the faster bipolar chips contain 1024 bits and the slower but cheaper MOS chips contain 4096 bits and, by 1975-76, densities of 64,000 bits/chip are predicted. Higher densities lead to lower per bit cost since chip costs will remain fairly constant. Larger capacity chips will also lead to cheaper component cost since fewer interconnections will be required per bit. In a sense the whole concept of computer design will change from minimization of gate counts to minimization of interconnections. Reliability per chip will remain constant or will increase, thus leading to a per bit reliability improvement by a factor of 20 to 50 by 1975-76.

Semiconductor memories are not only cheaper but they have a more linear cost/size curve than magnetic memories and thus make small random access memories economically feasible. The combination of logic and memory on a single chip makes associative memories (AM) economically attractive (2 to 5 times the cost of RAMs by the mid 1970's. While the concept of a single large main memory is still ingrained in computer designers and users, these two outgrowths of semiconductor technology will have a profound effect on computer memory structures and architecture in the 1970's.

Associative memories form the basis of both caches (50 nsec AM) and of virtual memory systems (300 nsec AM). Larger associative memories will become a part of future computing systems for specialized user functions (see Section 3.4.3). Small random access

[1] See p. 39 of the May 10, 1972 issue of Computerworld.

memories will be used to hold microprograms, both for CPU or ALU control and for the control of peripherals. Small RAMs will be used as low cost buffers in peripherals and in terminals. The incentive for such distributed control is the possibility of reduced data flow in the system and thus fewer interconnections and, in remote access systems, Iower communication costs. Large single memory computer systems will still be predominant in the mid 1970's, but their memories will be faster, more reliable and considerably cheaper. It has been shown that the locality of programs makes possible immense gains in performance through the use of cache concepts. The projected progress in the cheaper MOS memories already available with speeds of 175 nsec and cycle time of 300 nsec in sizes of several million bits, will probably lead to the replacement of cache systems in all but the highest performance systems. Recently R.F. Elphant of IBM's Component Division predicted that large systems would use a hierarchical memory system consisting of a central ALU surrounded by control stores (i.e. microprogram memories) surrounded by local stores and buffers, surrounded by main memory, surrounded by backing stores. Technologies in the hierarchy will include cores, semiconductors, bubble and domain tip (DOT) memories.[1]

While the importance of cache memories may diminish, virtual memories, intended to give the user the impression that the main memory is very large (and thereby eliminating much of his worry about secondary memory management), will become much more important. The paging mechanism on which they are based will be particularly beneficial to multiprogram and time-share users. In August of 1972 IBM gave official industry sanction to the virtual memory concept with an announcement of hardware virtual addressing options for its line of 370 computers.

[1] See Section 5.2 for a discussion of the bubble and DOT technologies. The issue of memory hierarchies is also discussed in Section 5.4.

5.0 MASS MEMORY

5.1 STANDARD MAGNETIC RECORDING DEVICES

Mass or secondary memory is the memory that holds files of information and serves as a backup for main memory during a computation. Secondary memory has traditionally consisted of some form of magnetic recording on an iron oxide surface. In the 1950's and early 1960's the predominant medium was the magnetic tape; since then disk systems have been the major mass memory device. Disks were an important advance since they provided random access to data and were easier to use and set up than tapes. Tapes, because of their mobility, ease of storage, and low cost (.001 cent/bit versus .02 cent/bit for disks) are still used, but primarily to store large and infrequently used data files and to hold backup copies of disk files.

Drums and fixed head (or head per track) disks have been used where access time to mass memory is critical,[1] but because these disks and drums are not removable from the drive, the impact of these devices is minimal in relation to regular disk systems for ordinary data storage. The role of drums and fixed head disks could expand in the 1970's because the devices' characteristics make them ideally suited for page storage in virtual memory systems.[2] An inhibiting factor is the high cost of the extra heads needed. If access time of movable-head devices decreases, they will replace fixed head devices for virtual memory system.

[1] The access time in head-per-track devices is just the time taken for the medium to rotate to the position containing the data (1/2 the revolution time on the average - in the 10 millisecond range). Ordinary disks require additional time for moving an arm containing the heads to the proper cyclinder. Of course, efforts are made to keep related information on the same track to minimize arm movement.

[2] The prime candidate is the IBM 2305 fixed head disk, designed for the top of the IBM line. The 2305 has a capacity of 5.4 million bytes with a 2.5 millisecond access time. The capacity is expected to increase.

While in primary memories access time is the chief performance criterion; in secondary memories on-line capacity[1] is more critical. That is not to say that fast access is not a desirable feature in secondary or mass storage systems. Capacity can be equated chiefly to recording density since tape length and the number of recording surfaces on a disk are standardized. Early tapes had 256 bpi (bits per inch) and today's most advance disk systems[2] have 3000-4000 bpi.

Until now, increased recording densities have been fairly easy to achieve, but we are rapidly approaching a technological, although not physical, limit. Of the three basic aspects - the writing process, the demagnetization process, and the reading process - the first, or writing process, is imposing current limitations. To get increased recording density one must go to thinner and thinner media (early tapes had a 500 micro inch coating while that of the IBM 3330 is 50 micro inches); but, as the medium gets thinner the signal gets weaker. To increase signal strength one must increase magnetization but that spreads the magnetic field. This means that when one tries to write one bit, the neighboring bit is still close enough to be affected. To increase recording density one must therefore change the balance in this tradeoff between resolution and signal strength.

Recording density can be increased both by increasing the bit (i.e. linear) density and by increasing the track density.[3] With the current technology one could probably increase bit density by another factor of two and track density by another factor of two,

[1] The on-line capacity is the amount of data available without human intervention. In a disk system this is the product of the number of disk drives and the capacity of one disk pack.

[2] The standard of advanced disk systems is the IMB 3330.

[3] On Both tapes and disks bits are recorded serially on a number of parallel tracks. On tapes one reads several tracks at the same time while on disks one normally reads only one track at a time.

thereby yielding a fourfold increase in capacity.[1] By changing the technology from today's discrete transducers to integrated transducers, track density could be increased by another factor of four. Work is underway to develop such integrated transducers, but commercial availability is not expected in the next five years.

To project what sort of disk systems will be used in 1977, let us consider today's reference systems, the IBM 2314 and the IBM 3330. Table 5-1 gives the basic system characteristics. The 2314, which was the mainstay of disk systems until the introduction of the 3330 in the early 70's, will have a life of at least another two years. In efforts to extend this life a double density version (.5 billion byte capacity) has been introduced.[2] The 3300 class of systems is expected to dominate the disk market until the late 1970's (see Table 5-2). Some improvements are foreseen, both in the area of capacity[3] and in the efficiency of disk systems.

TABLE 5-1. CHARACTERISTICS OF STANDARD 1972 DISK SYSTEMS

	IBM 2314	IBM 3330
PACK CAPACITY	29×10^6 bytes	100×10^6 bytes
DRIVES/SYSTEM	8	8 (+1 spare)
TOTAL CAPACITY	$.23 \times 10^9$ bytes	$.8 \times 10^9$ bytes
AVERAGE ACCESS TIME	60 msec	30 msec
TRANSFER RATE	312 kbyte/sec	806 kbyte/sec

[1] Some of this increase may result from using a plated disk in place of an oxide coated disk. In laboratories densities of 12-13000 bpi have been achieved with plated disks.

[2] This leads to transfer rates that, like those of the IBM 3330, are too high for machines below the 360/65 class. Adaptors for smaller machines are available, but they increase the disk system cost.

[3] Univac recently announced a system with a capacity of over 2 billion bytes.

TABLE 5-2. DISK MEMORY MARKET

	1971	1976
TOTAL MARKET UNITS VALUE	135,000 $3.6 billion	165,000 $8.3 billion
2314-TYPE SHARE	66,000 $1.9 billion	60,000 $1.7 billion
3330-TYPE SHARE	5000 .2 billion	57,000 $2.9 billion

The expected increase in efficiency of disk systems is primarily an outgrowth of availability of low cost semiconductor logic and memory, in this case incorporated into the controller of the disk system. The 3330 class already has such features as overflow multitrack operation, command retry in case of malfunction, rotational position sensing, storage control diagnostics (i.e. no CPU intervention is required for disk diagnostics), concurrent operations, error log, and statistical usage log.[2] The statistical usage log records things like arm movement in the hope that these measures will be useful in optimizing device use at some future date. Rotational position sensing identifies a number of sectors (100 or more) with respect to head position and thus permits optimizing access on a single track. Concurrent operation refers to the fact that multiple requests, possibly from different sources, can be handled.

[1] Figures due to International Data Corp. (See Computerworld, May, 10,1972.

[2] No one system has all of these features, but every feature listed does exist in some commercial system of the 3330 class.

The 1970's will see considerable progress in the area of such "smart" disks. Device controllers will become full fledged minicomputers capable of receiving requests from diverse sources, queueing these requests for optimal disk utilization (i.e. minimum delay due to head positioning and disk latency), and returning the desired data to the correct place. By the 1980's the concept of distributed computing will possibly give rise to separate memory controllers that control all levels of memory, allocate storage, transfer data between memories, optimize memory operations for all users, and perform processes such as counts and compares in data searches.

5.2 BUBBLE MEMORIES

In the past few years considerable attention has been focused on research with a new technology that promises to fill the three to four orders of magnitude speed gap between electronic bulk memories and electro-mechanical disks. These new devices, called bubble memories, make use of a physical property of thin magnetic garnet films (mounted on non-magnetic single crystal garnets) to support small magnetic domains (bubbles) and to allow these domains to be moved in the material through application of external magnetic forces. Bits are "on" or "off" depending on the presence or absence of a domain at a given point. These memories have no moving parts since a generator and detector are fixed and the information (i.e. domains) is moved about in two dimensions.

From these characteristics one can see that bubble technology is not particularly well suited for construction of random access memories. Bubble memories are adaptible to long shift registers (the longer the register the lower the generator and decoder requirement for a given size memory). It is possible to construct a number of parallel shift registers and to read a number of bits (say 8 or 32) in parallel every time the domains move. A set of these parallel shift registers can then be used as one block in a block oriented random access memory (BORAM).[1]

In current prototype bubble memories, the bubbles are about 1/4 mil in diameter and spaced 1 mil apart, thus a 20x20 mil piece can hold 200 bits. A 2"x4"x5" memory of 20 million bits is predicted in the near future and it is expected that in five years densities of 4 million bits/square inch will be achievable. Now bubbles move at a rate of 100 kilobits/second but this will soon increase to about 1 megabit/second. Advantages cited for bubble memories are small size, absence of moving parts, non-volatility, low power requirement (a 20 watt coil can handle 20 million bits), and the ability to build fixed logic functions into the memory. Cost is estimated at about $75 per wafer (3"x3") or .025 cents/bit.[2] That would put the memory into the IBM 3330 price range and in a slightly faster speed range.

Bubble memory work is progressing in at least twenty labs around the country with the leader, Bell Labs, having 100 professionals committed. No bubble memories are yet for sale but Bell is building a megabit prototype and it is expected that actual production of megabit devices will begin in 1975. Bubble memories are seen as direct competitors of head per track disks in the mid to late 70's. Despite the efforts expended in the development of this technology, it may turn out that bubbles will be short-lived because of the expected low cost of LSI memory in the late 1970's.

In May 1972, Cambridge Memories Inc. of Newton, Mass. announced an off-the-shelf available memory that is related to bubble memory. This Domain Tip (DOT) memory stores data in the form of magnetic domains that move through channels etched on an aluminum film. These compact BORAM's have a one microsecond access per block (2K of 16 bit words) and transfer rates up to 1 MHz. They are available in

[1] See section 4.3.

[2] These are rough estimates since no bubble memories have been produced in any quantity. The estimates are due to Dr. Alan Smith of the Sperry Rand Research Center.

sizes up to 16 million bits and are priced at about $2300 per megabit (in quantities of 200). A smaller, slower, and cheaper version (4K x 16 bits for $490) is also available. The smaller units could, with some loss in speed, replace cores in calculators and terminals at 1/10 core cost while the larger units could compete with small disk system (IBM 2311 class) at greatly increased performance for about the same cost/bit.

5.3 ULTRA LARGE MEMORIES

As the power of computers, the amount of processing done, and the size of computerized data files has grown, more and more secondary memory devices (tape and disk drives) have been added to computer systems. Libraries of tapes and disks have become unwieldly and expensive. The task of mounting tapes and disks has lead to user delay and inconvenience. This indicates that a slower but online ultra large memory has a definite place at the bottom end of the memory hierarchy. A typical ultra large memory might have a capacity of one trillion bits. This translates to 100,000 reels of 800 bpi magnetic tape or, in terms of on-line storage, about 150 eight spindle 3330 disk system ($6.4x10^9$ bits). Gross (Ref. 13) predicts that in ten years ultra large storage systems will range in size from a minimum of three to ten times of the largest disk system to a maximum of 10^{14} bits. Access times will be in the 1-10 second range.[1] By 1976 a 10^{12} bit memory is expected to cost in the vicinity of one million dollars. A system of 10^{13} bits should cost three million dollars. Several ultra large memory systems are on the market and, although few are in use, it is expected that sales will increase as more computer networks come into existence since these will enable a number of installations to share one ultra large memory.

[1] This is somewhat deceptive since the mode of operation will be to locate a large subset of a file (1-10 seconds) and then to transfer this (or portions thereof) at a high rate to a disk for actual data access. The access characteristics of several systems will be discussed in this section.

The technological basis for ultra large memories comes primarily from schemes employing optical (laser) and electron beams for reading and writing, primarily on silver halide films.[1] These devices permit recording up to 10^8 bits per square inch[2] whereas magnetic devices are limited to 10^6 bits per square inch. A moving beam can only cover a part of the entire recording surface of a large memory so that the recording medium must be moved mechanically to provide beam access to the entire storage area (hence the slow access times). Once the medium has been positioned data can be transferred rapidly since no more mechanical movement is required.

Two types of recording schemes are used - discrete bit recording and holographic recording. In the first scheme a beam moves to record (or read) each individual bit while in the second an entire array of bits, usually several thousand, is stored as one hologram. The entire hologram must be retrieved since no particular location in the hologram corresponds to a particular bit. Currently no suitable material exists for fast and cheap holographic recording, thus today's prototype holographic memories are ROMs.[3] Until some form of writing becomes possible at a reasonable cost, holography methods will not be suitable for ultra large memory systems. French scientists at Thomson-CSF claim to have developed a recording technique based on electrical fixing and erasing that takes between a milli-

[1] One ultra large memory system currently being marketed uses transverse recording on magnetic tape. This system, called the Terabit Memory (TBM), will be discussed later in this section.

[2] These densities are not yet achieved in production systems, but they have been exhibited in the laboratory. Commercial systems will achieve densities of 2 x 10^7 bits/square inch by 1977 and level off at 5 x 10^7 bits/square inch in the 1980's.

[3] Hitachi is reported to have a holographic (Read-Only-Memory) system with a 200 million bit capacity and 2μsec access time. The memory consists of a 5x5 cm plate that can hold 10,000 holograms. Optical Data Systems, Mountain View, California, is field testing a 12 megabit system with 1-1.5 sec access time and a cost of .01 cents/bit. Film cartridges containing the holograms are sent to ODS for information updating.

second and a second per hologram, but it is not expected that holographic technology will produce a commercial read-write ultra large memory system within the next five years. Discrete bit systems will be available in 1977 in writeable but non-erasable versions. With these information can be written once (for example, holes are burned into the medium by a laser) and records can be updated by copying onto unused portions of the medium.

Perhaps the first ultra large memory was the IBM Photodigital Store at the Lawrence Radiation Laboratory in Livermore, California. The storage medium is photographic film in the form of small chips (half the size of a playing card), each containing 4.7 million bits. The Photostore has 6750 cells of 32 chips each and thus a trillion bit capacity. Chips are accessed in pneumatic tubes and stored in trays. Chips in the trays can be replaced manually to permit off-line storage. The system writes with an electron beam in a vaccuum on a film that is automatically developed. Reading is done with a flying spot scanner. The Photodigital Store (IBM 1360) was originally announced in 1966 and five systems were delivered. It was not attractive economically and is no longer being produced, but there is no reason why the electron beam technology should not be pursued in the future.

Precision Instruments Inc. is marketing a discrete bit laser memory called Unicon. One of these trillion bit memories has been sold to NASA for attachment to the Illiac IV and the ARPANET. The Unicon stores its information on 5" by 31" rhodium coated plastic strips. The entire system is made up of 400 such strips, each of which holds 2.5×10^9 bits (a little less than 3×10^7 bits per square inch). The Unicon writes on the strips by burning holes in the film and reads by illuminating these holes. To perform read and write operations, a strip is mechanically placed on a rotating drum and the optical system is moved for track access. Strip changes take 5-10 seconds and, once a strip is mounted, access to 128 word (32 bit) blocks takes about 150 miliseconds. The transfer rate for reading is well into the megabit range. System cost of the Unicon comes to about .0001-.0002 cents/bit and new strips can be purchased for about $40 (per bit cost equivalent to magnetic tapes).

Low strip cost is important since one can only write on a strip once.

An alternative to laser and electron beam recording devices that is being marketed is the AMPEX Terabit Memory (TBM). The first of these systems are now being delivered. The TBM is a 2 inch magnetic tape system that uses helical scan recording, the same technique that is used for video tape recording. Unlike the Unicon, TBM is a fully read-write system. Its size varies from .1 to $3x10^{12}$ bits and is expected to increase to $7x10^{12}$ bits by 1976 and $15x10^{12}$ by 1981. Packing densities are 1.4 x 10^6 bits/sq.in. (1971), $3x10^6$ bits/sq. in. (1976) and $7.5x10^6$ bits/sq. in. (1981). Data rate is 6 megabits/sec and will be 2-1/2 times that by 1976. Average access time is quoted as 15 seconds. Cost per bit is between .0001 and .0005 cents/bit (depending on size) and is expected to decrease by a factor of two by 1980. Although tape search speed is 1000 inches per second, the TBM cannot compete with the Unicon system in terms of overall access speed since it is basically a serial (as opposed to random) access device. The TBM will be used chiefly in batch applications as a tape replacement system where a 500-1 recording medium reduction is possible.

The future of ultra large memories seems assured with two system starting deliveries, several others announced, and the three computer giants, IBM, Honeywell, and Control Data working on their own trillion bit memories. The cost of these system is high and the market still small, thus vigorous competition is not expected and, as a result, the current systems will not be outdated by 1977. Capacity is seen to increase by a factor of 10 (to 10^{13} bits), access times will improved by at most a factor of two for equivalent size memories, and per bit costs will decrease by 25 to 50% in the next five years. The long range prediction is that erasable holographic systems have the greatest potential for the 1980's.

5.4 CONCLUSION

In the next five years secondary storage media will see some evolutionary progress leading to lower cost and greater reliability, but no revolution of new devices is foreseen. Disk access times will remain in the 10's of milliseconds and maximum capacities may reach 5 billion bytes. The standard of the late 60's, namely the IBM 2314 class system with its .23 billion byte capacity, will remain useful for another two years. This lifetime could possibly extend into the second half of the decade for the double density version of the 2314. The current standard, the IBM 3330 class system with a capacity of .8 billion bytes, will dominate the disk market until the late 1970's although the cost per bit, currently in the .02 cent range, will be cut in half. The current controllers of 3330 class systems have some built-in "intelligence" and it is expected that in the next five years progress will be made toward the development of a "smart" disk.

While moving head disks will be the mainstay of the secondary memory technology of the 1970's, the trend towards hierarchical structures observed in the area of electronic memories is also becoming apparent at the level of secondary memories (see Figure 5-1). The large gap in access speed between traditional primary (electronic) memories on one hand and disks (electromechanical) on the other needs to be filled to improve performance of the virtual memory systems projected for the mid-1970's. The other level of the memory hierarchy that is predicted to become firmly established in the next five years is the ultra large memory with on-line capacities in the trillion bit range and access times of a few seconds.

Several possibilities exist for producing fast paging memories to fill the first of the above gaps. They include fast fixed-head drums or disks, slow but cheap MOS LSI memory, and BORAMs based on a bubble technology. The fixed head rotating devices have an edge today because that technology is established. IBM, for example, is marketing the 2305 fixed head disk with a 2.5 millisecond access time and a 5 million byte capacity. These devices are expensive relative to their performance and could easily become obsolete if moving-head disk performance improves sufficiently. Bubbles and MOS LSI are potentially cheaper than today's main memories and much faster than electromechanical memories. The prospect of a cheap MOS LSI memory is good, but, due to reliability problems, this type of memory is not expected to fill the existing gap for several years.

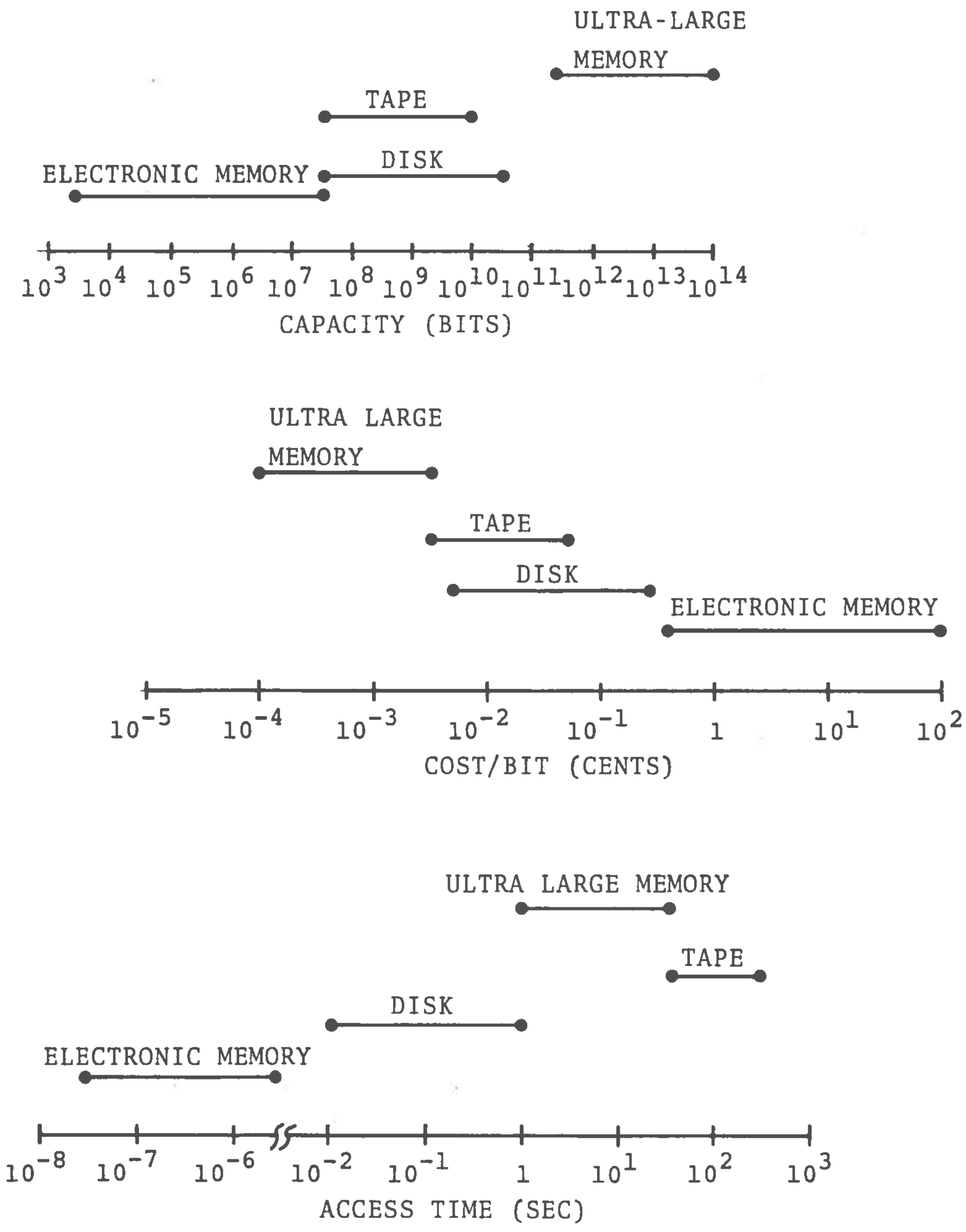


Figure 5-1. Memory Hierarchy in 1972

Bubble memories show promise, but they are still in a developmental state and thus will not become commercially available before the mid to late 1970's. The domain tip memory, a relative of the bubble memory, has appeared on the market, and if it proves itself will very likely replace the fixed head devices before 1975.

Ultra large memories offer the dual benefit of a very high on-line capacity and cheap, compact off-line storage. This makes the systems well-suited to large data base and archival storage. The first magnetic tape system using transverse recording, the Terabit Memory (TBM), was installed this year. It offers a 500-1 reduction in off-line storage requirements over magnetic tapes and thus will be used for archival storage, but even though the medium is capable of erasure and recording, the TMB's slow access time (15 seconds) will not make it competitive for heavily used on-line storage.

Ultra large memories systems based on laser or electron beams can provide both the recording density and random access required for fast access. Of the two recording schemes, discrete bit and holographic, the former will have the greatest impact in the 1970's. Unicon, a trillion bit laser system that allows reading and writing but not erasing, will become an operational part of the ARPANET this year. Holographic ultra large memory schemes are not expected to become competitive until the 1980's because of problems with the recording process. Experience with both TBM and Unicon in the next few years should provide a great deal of insight into the benefits and pitfalls of ultra large storage systems, but it is predicted that such systems will find a place in the memory hierarchies of the future.

6.0 SYSTEM ARCHITECTURES

6.1 INTRODUCTION

Traditionally the label "system architecture" has been applied to the study of how the components of a single computer system (e.g main memory, buffer memory, processor, printers, plotters, CRTs, secondary storage, terminals, etc.) interconnect and interact. This definition has come to be somewhat too narrow in the past few years because the single computer is no longer a strictly autonomous entity, but often a part of a larger computer network. A number of networks have been constructed, primarily for the sharing of resources, and they are predicted to proliferate in the 1970's. Thus the scope of a discussion on system architecture must be expanded to include the study of computer networks and the means for moving data in the network.

This section is divided into two smaller sections to reflect this new aspect of system architecture. Section 6.2 deals with computer systems in the more narrow sense. Attention will be paid to the concepts of multiprocessing and of distributed computing but in the context of a single system. Section 6.3 looks at a range of computer networks, from the single interconnection of two or three computers to the emerging computer utility that is based on a hierarchical arrangement of functionally diverse machines. This utility, interestingly enough, is again a manifestation of distributed computing, only this time on a much larger scale.

6.2 SINGLE COMPUTER SYSTEM

6.2.1 Multiprocessors

In Section 3 we discussed a number of promising approaches to increasing the throughput of a single processor. The approach that has generally been taken in the past to upgrade the performance of a computer has been to use more than one processor where additional processors are used for peripheral operations, for program execution, or for both. To avoid some of the terminology confusion of the literature on this subject, we shall restrict the use of the label multiprocessor to those systems that have more than one processor capable of user program execution.

Today's standard multiprocessor consists of a number of processors, a large single main memory and a mass memory (see Figure 6-1). The processors are connected to the memory by means of a bus, a cross bar switch, or, if a multiport memory is used, directly through a port. Several concurrent jobs may share a processor (multiprogramming) but a single job will not use more than one processor. A multiprocessor configuration adds to the reliability of the computer since a system is normally reconfigurable if one or more of the processors fails. Multiprocessors do require additional coordination and control and, after a certain number of processors are added to the computer, this overhead tends to defeat the objective of increased throughput.

The advent of semiconductor memories and the resulting availability of associative memories have made it possible to increase the efficiency of the traditional multiprocessor. Two particular schemes that have been proposed are the addition of a cache memory between each processor and the main memory and the addition of a hardware executive.[1] The latter is an associative memory for paging control and for task or job control.

The availability of cheap smaller memories and the advent of MOS-LSI memories or cheap BORAMs (block oriented random access memories) give rise to another multiprocessor approach. Instead of using a single main memory, each processor has a small or medium sized memory attached to it and, at the same time, has access to a BORAM for both program and data storage. A common memory provides a means for passing data between the processors (see Figure 6-2). In such a system much of the overhead required in traditional systems to avoid memory interference is absent because access to the BORAM is restricted to page faults.[2] Transfer of pages from the

[1] See Section 4.3 and Figures 4-2 and 4-3.

[2] A page fault occurs whenever a word in a page not currently in the main memory is addressed. To satisfy this memory reference, a page must be transferred from the BORAM to the main memory. See also the discussion of virtual memories in Section 4.3.

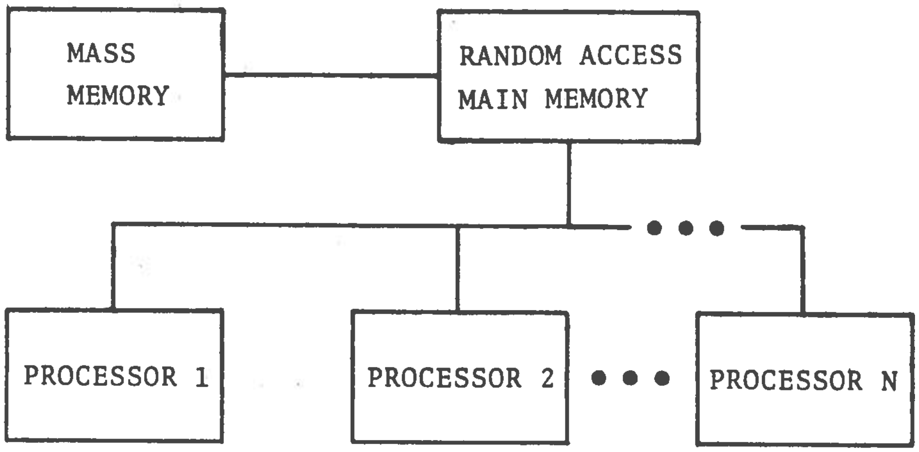


Figure 6-1. Traditional Multiprocessor

BORAM to the processor memory is fast because of microsecond page access and high data transfer rates (in the 10 MHz range). Hardware cost savings over the traditional scheme accrue from a cost savings in main memory and from simplified switching between processors and memory.

A multiprocessor along these lines is being constructed for the US Navy with a prototype scheduled for completion by the end of 1972 and first delivery to the Navy expected by the end of 1973. It is called the All Application Digital Computer (AADC) and is projected to replace a majority of the Navy's current computers. The AADC processor, built wholly from a small set of LSI components,[1] is only 4"x5"x5" in size but is capable of 2-3 million instructions per second.[2] An entire AADC computer will be arbitrarily configurable from these processing elements and their local 4096 32-bit

[1] The computer is constructed from modules consisting of 6 Raypaks (a Raytheon trademark) which in turn consist of 6 LSI chips. A processor is built from 3 module types, 5 Raypak types, and 10 chip types.

[2] Many of these are vector and matrix operations, thus the power is actually greater than 3 MIPS. Al Deerfield, an optimistic Raytheon Spokesman, compares a processor to the 360/195 in terms of power.

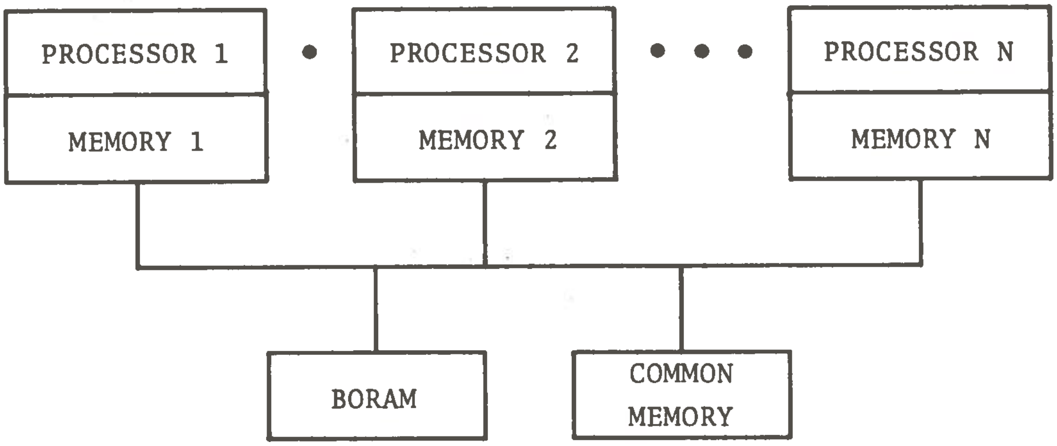


Figure 6-2. Future Multiprocessor Configuration

word memories (called task memories), a BORAM, and an I/O subsystem that provides access to a set of 250 nsec, 8-16K word random access memories (RAMs), to secondary memories, and to user interfaces (see Figure 6-3).[1] Original plans call for a BORAM with 2μsec access and 100-150 nsec word transfer rates for reading but with very slow writing rates.[2] When executing a program, the processors can directly address both the task memories (virtual addressing) and the main memory modules.

The AADC is also interesting because of the architecture of the processor itself. Instead of following the Von Newmann organization, the processor consists of a program management unit (PMU), an

[1] The Navy feels that 90% of its installations will need only one processor.

[2] This is called a "mostly read" memory.

arithmetic processor (AP), an arithmetic processor queue (APQ) and a "block" (see Figure 6-4). The PMU provides the memory interface and is the fetching mechanism. In a sense, the PMU gets the instructions ready for processing and then places them in the APQ. The AP is the actual execution mechanism. It is not an ordinary arithmetic processor because it can execute scalar, vector, and matrix operations and because it has an entire set of APL primitives[1] built into the logic. The "block" is a type of push-down stack used to handle parentheses and to defer operations so that APL statements can be executed at the hardware level without interpretation or compilation. Data types are associated with the data addresses, so that the processor is data insensitive. The APQ is important because it allows the relatively slower PM to work ahead of the AP and thus balances off the variety of instruction speeds.

Although the AADC is more of a scientific than business oriented computer, it is expected to be installed in large quantities. This has several advantages without which a novel architecture would be unlikely to survive. First of all, it is economically feasible to expend sufficient sums of money to develop good software. The Navy is developing its own language (CMS III) which will contain APL. There will also be an operating system and compilers for JOVIAL and FORTRAN. Secondly, with a large user population it is more likely that progress will be made in determining how the machine can be used most effectively and how it might be modified for increased performance.

6.2.2 Minicomputer Multiprocessors

None of the multiprocessor configurations of the previous section are meant to have interaction between tasks in different processors. Because the processors were fairly large, it was not necessary to use more than one for any given job and thus some difficult control problems were avoided. A proposed configuration where several processors might interact while processing one job is the minicomputer multiprocessor. In Section 3.5 it was suggested that the low cost of minicomputers, the ease with which they can be microprogrammed, and the availability of crossbar switches or

[1] APL is a powerful interactive programming language whose primitives (i.e., basic set of operations) include vector and matrix operators.

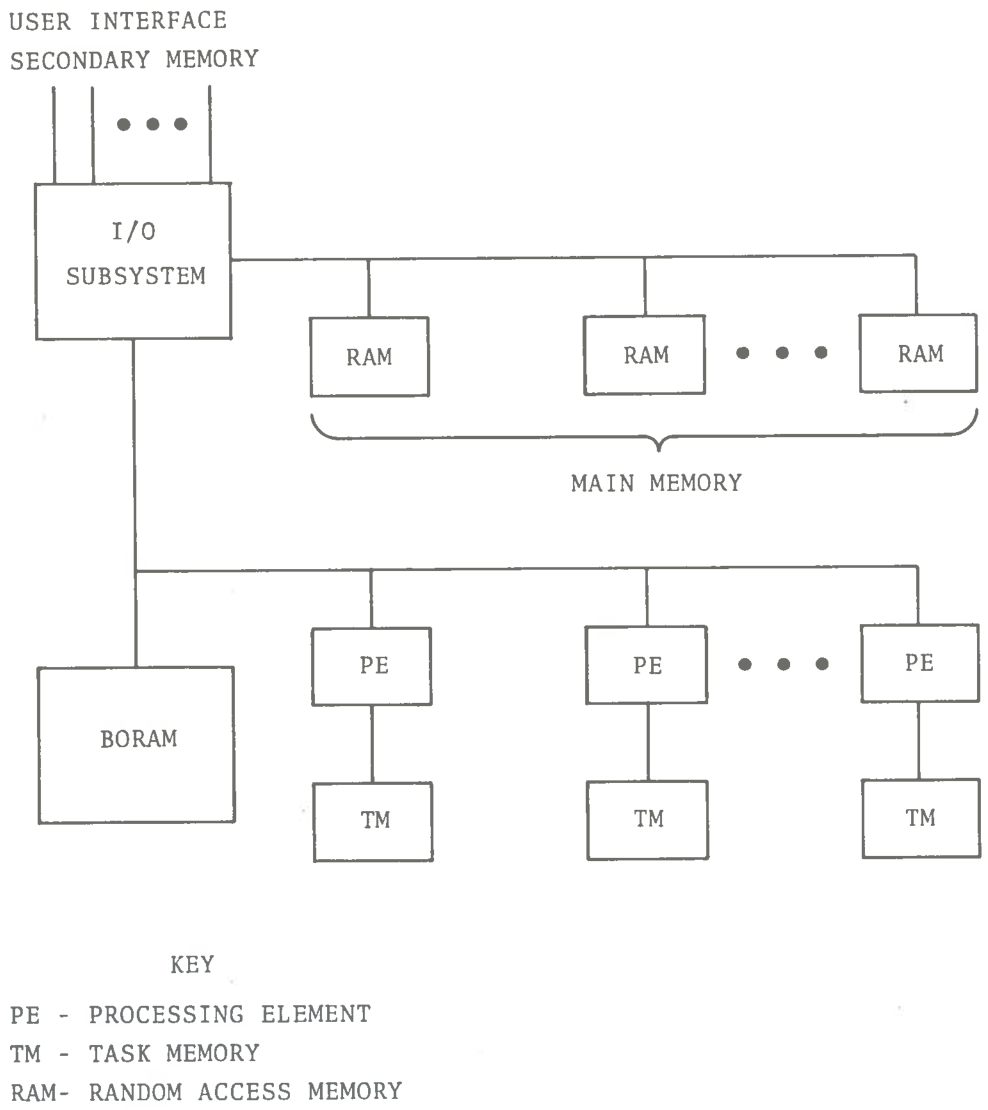


Figure 6-3. AADC Multiprocessor

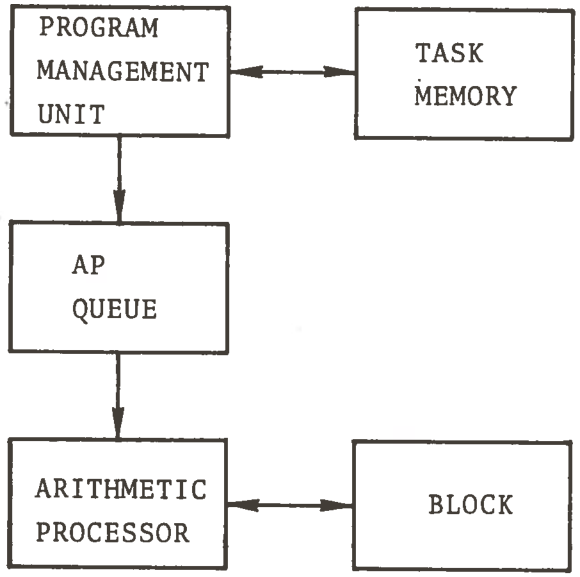


Figure 6-4. Processor (PE) of the AADC Computer

multiported memories make a minicomputer multiprocessor an interesting replacement for the traditional medium-to-large scale computer. Bell and Newell (Ref. 3), for instance, argue that two technologically inhibiting factors in the development of multiprocessors are the relatively high cost of processors and the cost and reliability of cross bar switches. Because miniprocessor costs are dropping rapidly and because 16-bit machines require a smaller crossbar band width, they conclude that minicomputers multiprocessors are more likely to proliferate in the next five years than large multiprocessors.

A minicomputer multiprocessor looks quite similar to the traditional multiprocessor described in the last section. A number of actual processors (with hardware for address translation) are connected, via a crossbar, to a number of primary memory modules. Terminals containing processing memory can be attached directly and a second switch connects the processors to buses which are attached to secondary memory and terminals (see Figure 6-5).

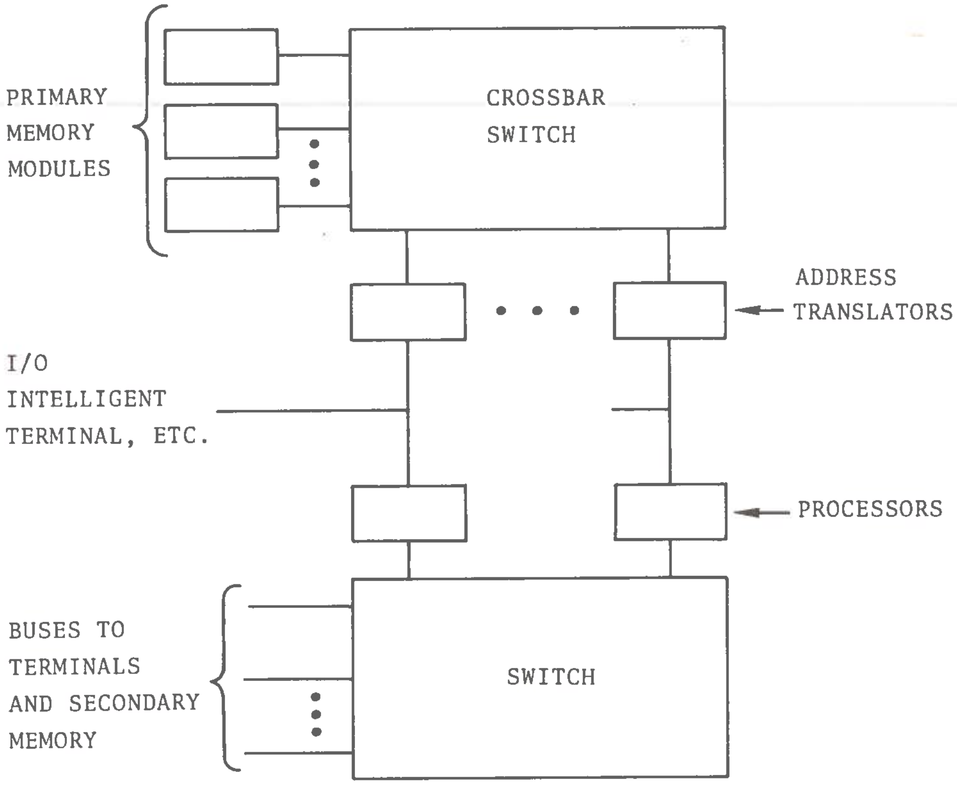


Figure 6-5. Minicomputer Multiprocessor

A number of arguments can be given for such a configuration. It is possible to use each of the minis for a natural functional subsystem of a large scale application. Although some extra care must be taken in the problem statement, the configuration does facilitate cooperation among the problem subtasks while reducing system overhead for both input-output and operating system programs. The system is fully reconfigurable in the sense that it is possible to add and delete processors and memory modules as well as a host of peripherals. Individual processors can be microprogrammed for special applications without affecting the rest of the computer. Although the crossbar switch degrades the execution rate of individual processors each processor has access to a much large memory space. The common access to the secondary memory and to the peripherals represents a considerable savings over an equivalent number of independent minicomputers.

6.2.3 Distributed Computing

The multiprocessors described in the last two sections have all had processors that were functionally interchangeable and independent. Another alternative that has been forwarded (and used as early as the mid 60's by the CDC 6600) is to organize the system into a main processor and a number of smaller support processors. The support processors are used primarily to relieve the main processor in a large computer of the burdens of I/O operations, file maintenance, and communications tasks.

The boom in LSI technology seems to favor a slightly altered version of this approach but one that has the same motive - to relieve the main processor of the burdens of peripheral functions and control. Instead of using support processors (which in turn require interfaces to peripheral device controllers) for these operations the controllers themselves are upgraded into minicomputer modules.[1] Front-end processors (i.e. telecommunication processors and message concentrators) remote batch terminals, graphics terminals, and "intelligent" terminals all contain minicomputers.[2] This concept is called "distributed computing" because the overhead required to make the system work (as opposed to the applications program) is distributed to the peripherals (or peripheral controllers) whose operations are being supervised. The roles of the traditional peripheral controller and of much of the operating system dealing with I/O are combined in this minicomputer. This has the added advantage of making controller changes easier since what formerly required a hardware or microprogram modification can now be done through a software change.

Distributed computing most often refers to peripheral and terminal control but there is no reason why main memory management should not be performed by a minicomputer module. Similarly, the

[1] The cost of mini CPU's and memories are becoming so low that one should think of them as modules of only a few LSI chips, each just like one thinks of a magnetic tape drive or disk drive as a module.

[2] See Sections 3.5 and the discussion of the "smart" disk at the end of Section 5.1.

separation of instruction preparation and instruction execution in the AADC is an example of functionally distributed computing.

6.2.4 Conclusions

In the last three sections we have discussed some of the current thoughts and experimental concepts in multiprocessor systems, both of large and small computers, and of distributed computing by a number of functionally oriented minicomputer modules around one or more central processors. In this section we will try to examine the effect of these ideas on commercial systems of the next five years. We will attempt to show what a computer system will look like in 1977.[1]

Although it was not particularly stressed in this chapter, the impetus for most of the recent architectural concepts has come from the LSI revolution. The rise of cheaper semiconductor memories will lead to faster minicomputers with fewer architectural features. The reason is twofold: (1) machines must be simpler to increase cycle time so that the machines can keep up with the faster memories and (2) the cost must be lower to balance the lower memory cost. These simpler, cheaper, and faster minis will be ideally suited to augment the complex main processor(s) of a functionally distributed computer system. At the other end of the spectrum, the large scale computer will become cheaper (through use of LSI) and more effective (through use of cache memories, associative hardware executives, and functional distribution of computing). Already today the small and large machines offer a lower cost per instruction than medium sized machines. These arguments lead to the prediction that there will be more mini and large scale computers while the middle range will lose ground in the 1970's.

The trend towards distributed computing, particularly in the areas of smarter peripheral controllers, communication front-ends,

[1] A similar discussion with more emphasis on what the system will look like to the user can be found in Section 2.7. See also Section 6.3.4.

and intelligent terminals, has been quite evident in the past two or three years and is expected to have a significant effect on computer system architectures in the next five years. Big processors will be used for number crunching (i.e., basic computations and arithmetic operations) without interference from bookkeeping operations, data transfers between levels of the memory hierarchy, etc. Functional decentralization permits cheap and possibly limited minicomputers to do that processing for which they are better suited than the CPU of the system. It is expected that many more intelligent terminals, many with remote batch capabilities, will exist in 1977. They will handle much of the interactive load that requires fast response and some data handling but little computation.

These large distributed machines will generally have one or two main processors. It is unlikely that really large scale (i.e. five or ten big central processors) multiprocessors will be seen in 1977 era. Figure 6-6 shows what a typical system might look like. There is a good chance that these machines will have cache memories and hardware executives, but it is less likely that individual processor memories and backup BORAMs (such as that of the AADC) will be prevalent in the commercial computers of 1977. It is not expected that systems with more than one main processor will be used as parallel processors, that is, a single job will not be using the two processors at the same time. There will be parallel processing in the sense that various parts of the job will be done simultaneously in different portions of the distributed machine.

Minicomputer multiprocessors as replacements for larger computers have been receiving increased attention lately because of the rapidly decreasing cost of miniprocessors and small memories. One commercial system (the Memorex MRX computer) has already been announced, but it has only four processors (two for computations and two for I/O functions) and thus still falls in the small computer category. To attain the power of a medium-to-large scale computer, at least ten and possibly as many as fifty mini CPUs would be required. It is not expected that such systems will become commercially available by 1977 although some prototypes will undoubtedly be constructed. Two major reasons can be given for this. First,

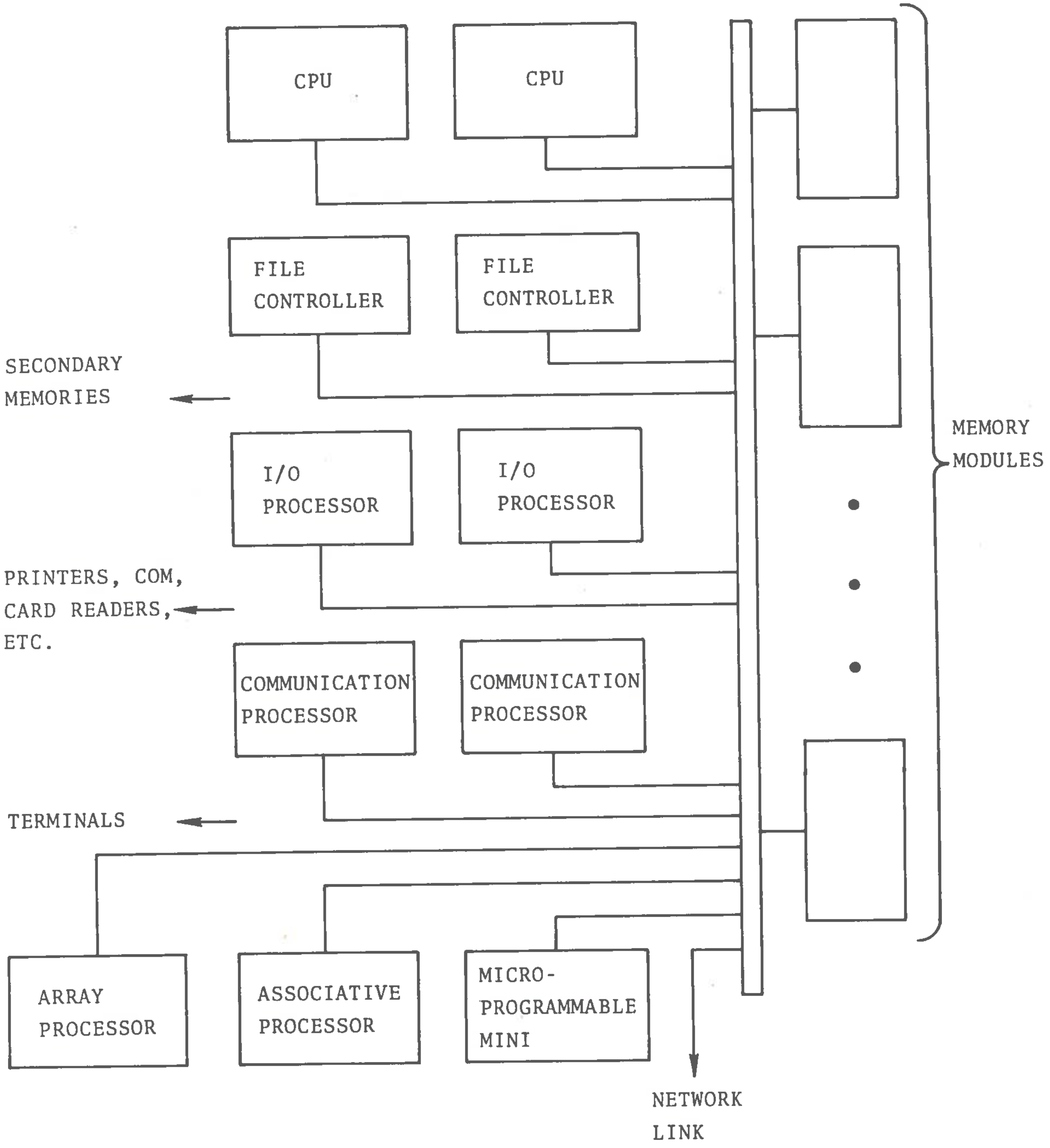


Figure 6-6. Computer of the 1977 Era

the software techniques required to control and coordinate such a system are not yet well understood. It may turn out that software overhead for a minicomputer multiprocessor is too great to justify the benefits of flexibility. Secondly, the interconnection overhead (i.e. crossbar switch cost) is very high. Bell and Newell (Ref. 3) argue in favor of a minicomputer multiprocessor, but concede that a 16x16 switch will cost from four to sixteen times as much as a single processor. It has been proposed (Ref. 28) that a mini net be built out of modules (consisting of a processor and storage) which are connected by a bus interfaced through a multiprocessor communication adaptor. This scheme of processor storage modules (PSM) does eliminate the need for a crossbar switch but it also takes away the benefits of a large shared memory.

6.3 COMPUTER NETWORKS

6.3.1 Introduction to Networks

A computer network is a set of autonomous dependent or independent computing systems that are interconnected to permit interactive sharing of certain resources. Basically, a network makes local resources available to remote users without degradation of service. In practice there is a continuum of configurations ranging from a single computer with a number of remote teletypes to a decentralized network of many independent heterogeneous computers that are time-sharing systems in their own right. The definition of network given here precludes the single time-shared system and even the time-shared system with intelligent terminals but includes the increasingly popular computer networks that have a central "work horse" computer and one or more smaller remote machines to do some local processing and to feed the large computer.

The latter type of system is commonly called a centralized network because it is made up of a central machine with a number of satellites (see Figure 6-7). A more general type of network, composed of independent systems each of which could have a number of terminals, is called a decentralized network (see Figure 6-8). It is conceivable that a centralized network could also be a part of a decentralized net. Either type is categorized as a homogenous network if the computers in the network are compatible machines in terms of work length, instruction sets, operating systems, etc. This normally implies that all computers belong to one series of a particular manufacturer's product. If different manufacturers machines or incompatible machines from one manufacturer are used, the network is classified as heterogeneous.

6.3.2 Centralized Networks

There are numerous motivations for a centralized computer network. The trend toward the large central computer with several satellites has arisen primarily to allow several organizations (usually universities) to share computer resources for cost savings. Rather than supporting several medium scale computer one large computer is shared and the cost per instruction is thereby reduced.[1] A shared computer has the added benefit of making available larger primary and secondary memories to the user and of cutting software system maintenance costs. These of course are the same types of arguments that were given for time-shared computers in the mid 1960's. Their resurgence in polemics over centralized networks reflects the fact that although commercial time-sharing is becoming more widespread and profitable (from 28 profitable firms in 1971 to 40 in 1972) the time-shared utility once predicted has not appeared. Many users prefer a batch or remote batch type of operation or find the higher cost of time-sharing unacceptable. A shared facility can support this kind of a user and give him the benefits of a large computer. It should be added that the trend in centralized networks is to provide full or limited time-sharing in addition to the normal services.

The simplest kind of network is a homogeneous centralized network. A typical example is the Triangle University Computer Center (TUCC) which has been serving Duke, North Carolina State, and North

[1] In Section 6.2.4, it was argued that cost per instruction is lower for small and large machines than for medium scale machines.

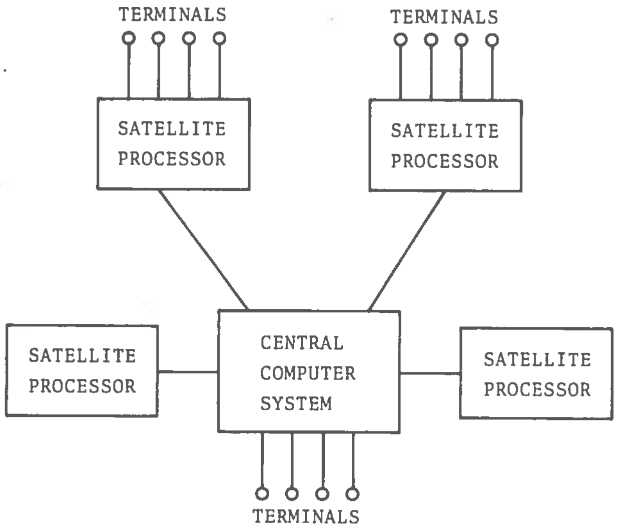


Figure 6-7. Centralized Computing Network

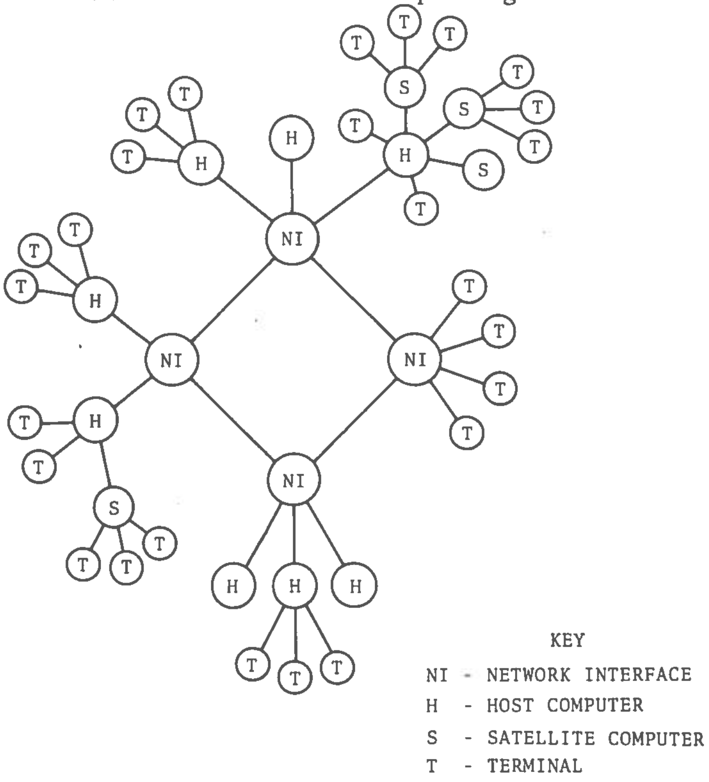


Figure 6-8. Decentralized Computing Network

Carolina Universities since 1966. The heart of the TUCC network is an IBM 360/75. Three high-speed terminals (360/40's and 50's) do local batch work in addition to the network telecommunications.[1] These modes are connected to the central facility by a leased 50 kilobit line. The network also has medium speed terminals (1130's and IMB 2780's) and low speed terminals (teletypes, 2741's, IBM 1050's) at the three universities and at local schools and colleges. The network is quite straight forward in that it uses off-the-shelf hardware and the vendor supplied IBM OS/360 operating system with minimal extensions.

6.3.3 Decentralized Networks

Decentralized computer networks are more difficult to implement, both from a software interface and hardware or communications standpoint. They do offer a number of advantages that are not found in the simpler centralized networks. It is possible to do sharing of workloads, data, programs, and specialized hardware. Workload sharing refers to the ability of one computer in the network to absorb heavy loads from another system during peak usage periods. Data sharing involves the sending of programs to a remote computer to use data resident at that site. Program sharing is the sending of data to a remote computer to be processed by programs implemented on that computer. Hardware sharing refers to the fact that specialized hardware at one installation can be used remotely by the users of any other installation. For example, in the ARPANET a trillion bit store and the ILLIAC IV at the NASA Ames node will soon be available to the entire ARPANET community. As a network gets larger and the number of people who are producing useful resources increases, the various forms of sharing increase the value of the network.

A number of homogeneous and heterogenous decentralized networks have been constructed. Commercial firms have generally developed

[1] To get a feeling for the relative power of the machines it should be noted that a 360/75 is roughly equivalent to six 360/50's.

homegeneous nets since these avoid problems of program, data, and operating system incompatibility. Typical examples are the Cybernet system that connects Control Data Corporation's data centers and the Tyme Share Inc. Tymnet data communications network. Cybernet uses CDC 6600's to provide the primary computing capability while CDC 3300's act as front ends and data concentrators. Tymnet has a 40,000 mile ring of leased telephone lines connecting 22 computers. This network has both DEC PDP-10's and XDS-940's, but users generally access only one kind of machine.

Heterogeneous decentralized networks present additional problems in establishing meaningful communication between machines, but they also offer additional benefits. Because different computers are in the net, the user can choose the machine best suited (in a hardware and software sense) for his particular application. Because standards are established to solve the inter-computer communication problems between unlike machines, it is easier to add specialized hardware to a heterogeneous net.

The prime example of a heterogenous decentralized network is the ARPANET, an experimental network designed to explore network technology. The ARPANET connects some twenty-odd universities and private firms involved in computer science research for the Department of Defense's Advanced Research Projects Agency (ARPA). The ARPANET is made up of two parts, the computers that are connected (called hosts) and the communications portion of the network. Computers in the network include standard machines (PDP-10's, IBM 360/65, 360/67, 360/75, 360/91, Burrough B6500, XDS-940, etc) as well as some more unique computers like the Multics GE-645 and the TX-2 at MIT and the ILLIAC IV (together with a trillion bit Unicon memory) at NASA Ames.

The host computers have, in addition to their normal operating systems, a resident Network Control Program (NCP). The NCP handles connections between programs in different hosts and monitors network usage. Each host is connected to a network interface called an Interface Massage Processor (IMP). One of these IMPs (a modified Honeywell DDP-516) can handle up to four hosts. IMPs are connected via 50 kilobit leased lines in a manner that permits at least two distinct paths between any pair of IMPs for improved reliability. Whenever a message for another host is generated, it is sent to the IMP together with the address of the recipient. The message is then routed to the receiving host's IMP by the store and forward network of IMPS and the high speed links between IMPS. Many of the hosts in the ARPANET are time-shared computers and, to give remote users access to another host without going through their local computers and burdening the NCP, a special kind of network interface, called a Terminal Interface Processor (TIP), that contains a miniature version of an NCP was designed. It is possible for a user at a terminal to be tied directly to his local TIP and establish contact, via the network, with any available host.

The design of the ARPANET, particularly in the telecommunication area, indicates the direction that computer networks are expected to take in the next five to ten years. The two most important concepts are (1) the complete separation of the communication function from the host computers and (2) the replacement of switched messages with packets. The latter means that instead of going through a switchboard to establish a connection between sender and recipient and then sending the message over this established link, the IMP divides a message into a number of packets (at most 1000 bits in the ARPANET) and attaches the recipient's address to the packet. Those packets are then routed through the network from IMP to IMP until the destination is reached. Packets, which might have arrived via different paths, are reassembled at the destination IMP and sent to the proper host. It is interesting to note that one such store and forward communication system could be shared by several distinct computer networks.

The ARPANET currently uses 50 kilobit point-to-point leased lines but it is planned to add the ALOHA network in Hawaii to the ARPANET. This is of considerable interest from the view of communication technology since data will be sent to and from Hawaii via satellite and since the ALOHA system itself uses terrestial microwave broadcast channels. The progress of this satellite network link will be watched with considerable interest despite a recent study by the technology group of Salmon Brothers that predicts that hardware and software problems will not make satellite data communication a cost-effective alternative to terrestial facilities in the near future (Ref. 20).

6.3.4 The Future of Networks

Until the late 1960's insufficient attention was given to the notion of computer networks that allow users to communicate with one another and build upon or extend each other's work. Prime reasons were limitations in hardware and software compatibility and poor data communication facilities. This state of affairs has been changing for a number of reasons: third generation systems provide intra-system compatibility; great advances have been made in data communications (lower rates, improved capability and reliability, digital multiplex or and modem advances, and the introduction of private microwave links); and because considerable experience has been gained from prototype networks. Non-government network data services are growing at an annual rate of forty percent and by 1975 it is expected that data services accessed via telecommunications will have one third of the market share.[1]

By 1977 it is expected that a computer center will be able to buy its way into a decentralized network. More and more pressure is being placed on the ARPANET, for example, to allow additional hosts into the system. It is expected that in 1974 the ARPANET will be sold to a common carrier or to a communications oriented firm or else will be transferred to another government agency for maintenance and administration. This would signify a change in the ARPANET status from a research vehicle to a viable user tool.

6.4 CONCLUSION

In this section we have attempted to show that the process centered view of computing is outdated. At the level of the single computer, memories are arranged in a hierarchical fashion and

[1] These figures are due to Dr. Ruth Davis of the National Bureau of Standards.

control is functionally distributed throughout the machine. Instead of worrying solely about keeping the CPU busy, efforts are made to properly balance and coordinate different parts of the system.

At a higher level, a definite decentralized hierarchical computer organization is apparent. Computation is done at the terminal, at a buffer computer, at the local host computer, and at remote hosts in a national network. L.G. Roberts of ARPA argues that with today's technology and at the expense of one or two television channels, it is possible to provide computer access from pocket-sized hand held terminals to a large part of the population (Ref. 27).

The dividing line between the traditional computer building blocks will become less and less clear from the designers view point. To the user, the hierarchical structure will become totally transparent. He will use the system of the future as he would a utility such as the telephone. The decentralized networks being developed today are a definite manifestation of this change of computing from a goods to a service industry.

REFERENCES

1. Auerback, I.L., "Technological Forecast 1970" in Gruenberger, F. (ED) Expanding Use of Computers in the 70's, Prentice Hall, Inc., Englewood Cliffs, N.J., 1971.

2. Ball, C.J., "Communications And The Minicomputer", Computer September/October 1971.

3. Bell, C.G. and Newell, A., "Possibilities For Computer Structures 1971", Proc. AFIPS FJCC 1971, p. 387.

4. Bell, C.G. and Grayson, J., "A Comparative Hardware-Software Design Study Using DEC Register Transfer Modules (RTM)", in Proceedings of the 1971 IEEE International Computer Society Conference, September 22-24, Boston, Mass.

5. Bell, C.G. and Grayson, J., "The Register Transfer Module Design Concept", Computer Design, May 1971, pp. 87-94.

6. Davidow, W.H., "The Rationale For Logic From Semiconductor Memory", Proc. AFIPS SJCC 1972.

7. Dennis, S.F. and Smith M.G., "LSI - Implications For Future Design and Architecture", Proc. AFIPS SJCC 1972, p. 343.

8. Farber, D.J., "Networks: An Introduction", Datamation, April 1972.

9. Flynn, M.J. and Rosin, R.F., "Microprogramming: An Introduction and A Viewpoint", IEEE Trans. on Computers, Vol. C-20, No. 7, July 1971, pp. 727-731.

10. Flynn, M.J., "Toward More Efficient Computer Organizations", Proc. AFIPS SJCC 1972, p. 1211.

11. Glushkov, V., "Computer Developments and Prospects", NAUKA I. ZHIZN No. 2, 1971, pp. 59-64.

12. Graham, W.R., "The Impact of Future Developments in Computer Technology" Rand Corporation, June 1970.

13. Gross, W.A., "Ultra-Large Storage Systems Using Flexible Media, Past, Present, and Future", Proc. AFIPS SJCC 1972, p. 957.

14. Gruenberger, F. (ED), Expanding Use of Computers in the 70's, Prentice Hall, Inc., Englewood Cliffs, N.J. 1971.

15. Hobbs, L.C., Theis, D.J., Trimble, J., Titus, H., Highberger, I. (EDS) Parallel Processor Systems, Technologies, and Applications, Spartan Books, N.Y. 1970.

16. Hootman, J.T., "The Computer Network As A Market-Place", Datamation, April 1972, p. 43.

17. Irwin, J.W., Cassie, J.V., and Oppeboen, H.C., "The IBM 3803/ 3420 Magnetic Tape Subsystem", IBM Journal of Research and Development Vol. 15, No. 5, September 1971.

18. Keyes, R.W., "Physical Problems and Limits in Computer Logic", IEEE Spectrum, May 1969.

19. Keyes, R.W., "Physical Limit on Computing Devices", Proc. 1971 IEEE Computer Conf.

20. LaBlanc, R.E. and Himsworth, W.E., "Data Communications in 1980 - A Capital Market View," Proc. AFIPS SJCC 1972, p. 611.

21. Liptay, J.S., "The Cache", IBM Systems Journal, Vol. 7, No. 1, 1968, pp. 15-21.

22. Meade, R.M., "How A Cache Memory Enhances A Computer's Performance", Electronics, Jan. 17, 1972, pp. 58-63.

23. Moore, G.E., "Semiconductor RAM's - A Status Report", Computer, March/April 1971.

24. Peck, P.L., "Effective Corporate Networking, Organization and Standardization", Proc. AFIPS FJCC 1971, pp. 561-569.

25. Riley, W.B., "Wanted For The 70's: Easier-to-Program Computers", Electronics, Sept. 13, 1971, pp. 61-84.

26. Roberts, L.G. and Wessler, B.D., "Computer Network Development to Achieve Resource Sharing", Proc. AFIPS SJCC 1970, p. 543.

27. Roberts, L.G., "Extension of Packet Communication Technology To A Hand Held Personal Terminal", Proc. AFIPS SJCC 1972, p. 295.

28. Seligman, L, "LSI and Minicomputer System Architecture", Proc. AFIPS SJCC 1972, p. 767.

29. Stone, H.S., "Pipeline Push-Down Stack Computer" in Hobbs, L.C. ET AL (EDS) Parallel Processor Systems, Technologies, and Applications Spartan Books, N.Y. 1970.

30. Thurber, K.J. and Berg, R.O., "Universal Logic Modules Implemented Using LSI Memory Techniques", Proc. AFIPS FJCC 1971, p. 177.

31. Uncapher, K.W., "The Rand Video Graphics System - An Approach To A General User - Computer Graphics System" 20th AGARD Avionics Panel Technical Symposium on Data Handling Devices Istanbul, Turkey, Jun. 1970.

32. Ware, W.H., "The Ultimate Computer", IEEE Spectrum, March 1972.

33. Waks, D.J. and Kronenberg, A.B., "The Future of Minicomputer Programming", Proc. AFIPS SJCC 1972, pp. 103-109.

34. Weber, E., Teal, G.K. and Schillinger, A.G. (EDS) Technology Forecast For 1980 Van Nostrand Reinhold Co. 1971.

35. Winograd, S., "On The Time Required to Perform Addition", JACM, Vol. 12, pp. 277-285, Apr. 1965.

36. Winograd, S., "On The Time Required to Perform Multiplication", JACM, Vol. 14, pp. 793-802, Oct. 1967.

37. Withington, F.G., "The Next (And Last?) Generation" Datamation May 1972, p. 71.