AD-A059 600 NAVAL POSTGRADUATE SCHOOL MONTEREY CALIF F/G 9/2
ACQUISITION PLANNING FOR TACTICAL AVIONICS SYSTEMS.(U)
JUN 78 C D ENGLEHARDT

UNCLASSIFIED NL

1 OF 1
AD A059 600

END
DATE FILMED
12-78
DDC

LEVEL II

(2)

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

(6) Acquisition Planning
for
Tactical Avionics Systems.

by (9) Master's thesis /

(10) Cleveland Duane/Englehardt

(11) June 1978 (12) 63p.

Thesis Advisor: U. R. Kodres

Approved for public release; distribution unlimited

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Acquisition Planning for Tactical Avionics Systems		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis June 1978
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Cleveland Duane Englehardt		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93940		12. REPORT DATE June 1978
		13. NUMBER OF PAGES -62-
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, CA 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Microcomputer
Minicomputer
Avionics
Acquisition
Computer Languages
Computer Programming
Computer Architecture

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

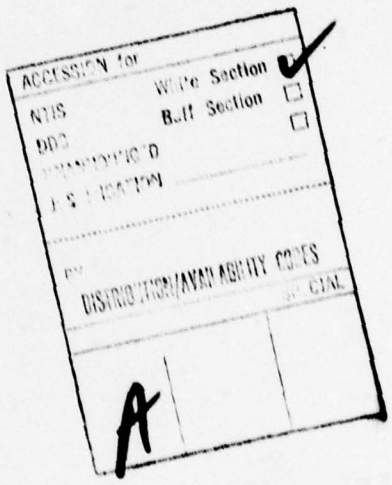
This thesis examines the use of microcomputer technology in tactical avionics systems and its impact on the procurement process of associated hardware and software. The rapid expansion of implementation of large scale integrated circuits in avionics systems aboard tactical military aircraft and missile systems has resulted in some serious potential problems in the areas of development, maintenance and acquisition of

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (continued)

microprocessor-based systems and software. These problems are identified and discussed, and proposed recommendations are made to lessen their undesirable long-range effects.



Approved for public release; distribution unlimited

ACQUISITION PLANNING FOR TACTICAL AVIONICS SYSTEMS

by

Cleveland Duane Englehardt
Lieutenant Commander, United States Navy
B.S.E.E., San Jose State College, 1968
M.S.A.E., Naval Postgraduate School, 1977

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MANAGEMENT

from the
NAVAL POSTGRADUATE SCHOOL
June 1978

Author: C D Englehardt

Approved by: Chas R. Kodres
Thesis Advisor

Second Reader

Chairman, Department of Administrative Sciences

Dean of Information and Policy Sciences

ABSTRACT

This thesis examines the use of microcomputer technology in tactical avionics systems and its impact on the procurement process of associated hardware and software. The rapid expansion of implementation of large scale integrated circuits in avionics systems aboard tactical military aircraft and missile systems has resulted in some serious potential problems in the areas of development, maintenance and acquisition of microprocessor-based systems and software. These problems are identified and discussed and proposed recommendations are made to lessen their undesirable long-range effects.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

A. PURPOSE

The dramatic advancements in semiconductor and microcircuit technology have made it difficult, if not impossible, for managers of major avionics systems procurement programs to remain technically abreast of their contractor counterparts. Without a firm understanding of what is transpiring in the avionics industry technologically, government representatives as well as industrial leaders will be unprepared to deal with the growth of the electronics capabilities. This rapid expansion of technology, presents a serious problem to the long-range planning of future avionics systems because of the widening gap in knowledge between buyer and producer resulting from the rapid growth of the technological base itself.

The goals of this thesis are to:

1. Increase the awareness of the reader in the area of current and future trends in avionics design.
2. Analyze the effects of microcircuit technology on the acquisition of tactical avionics systems and associated software.
3. Present alternatives to the present concepts of avionics systems development, design and procurement processes.

B. ORGANIZATION OF THE THESIS

This thesis assumes that the reader is not familiar with microprocessor terminology, or avionics applications of large scale integrated circuitry. First, we will present background material on the evolution of the microprocessor, its place among computers, and its use as a substitute for dedicated electronic circuitry. Second, we will discuss trends in the design of tactical aircraft avionics systems and the relationship of present and future designs to the microprocessor. The foregoing topics will serve as a primer for the remainder of the thesis, and may be skipped or skim-read by the reader who is familiar with microprocessor applications in avionics systems. Third, we will discuss the effects which the "LSI (Large Scale Integration) Revolution" has had, and is expected to have on the procurement of modern avionics systems. Within this topic, we will attempt to identify potential pitfalls of current concepts in avionics system design. Finally, we will discuss alternatives for dealing with the problems which arise because of the radical changes in technology.

Every effort has been made to maintain brevity, yet accurately convey the intended message in an understandable form avoiding, where possible, the newly created acronyms.

II. MICROELECTRONIC CIRCUIT BACKGROUND

This section is intended for the reader who is not familiar with the world of microprocessors, integrated circuit technology, or computer language structures. Its purpose is to introduce the microprocessor, some of the associated terminology, and project trends in computational hardware and programming language development. Readers already knowledgable in these areas may prefer to proceed to the next section which deals with tactical airborne computer systems. This section serves only as a brief introduction, however, Ref. 1 provides an excellent in-depth description of microelectronics principles and applications.

A. INTEGRATED CIRCUITS

Integrated circuits (IC) have been in common usage throughout the electronics industry since the early nineteen-sixties. In its simplest form the IC is nothing more than a collection of one or more transistors, resistors and capacitors formed on a plane of semiconductor material. It is generally designed to perform a specific function using either digital or analog principles. Typically, integrated circuits are in themselves very small, about one square millimeter or less in area, and are generally packaged in what is termed a Dual Inline Package or DIP. Most military applications utilize what is known as a flat pack which is usually square with pins for circuit board mounting protruding from the four sides of the package. Flat packs have a much lower profile than the DIP and

therefore result in greater circuit board chip density. Figure 1 is a photograph of an actual LSI radom access memory array magnified several hundred times. Figure 2 shows a typical DIP and an equivalent flat pack. This packaging serves to protect the enclosed microcircuit from dust, humidity, physical damage from impact, while it provides a means of connecting the device to outside circuit elements. The encasement also provides a medium for dissipating the heat generated by the internal circuit during normal operation.

There are three generally recognized levels of integration in microcircuit construction:

1. Small Scale Integration (SSI) -- These circuits usually perform simple logical tasks, such as independent boolean operations. SSI circuit packages typically consist of from 1 to 64 transistor and resistor components.

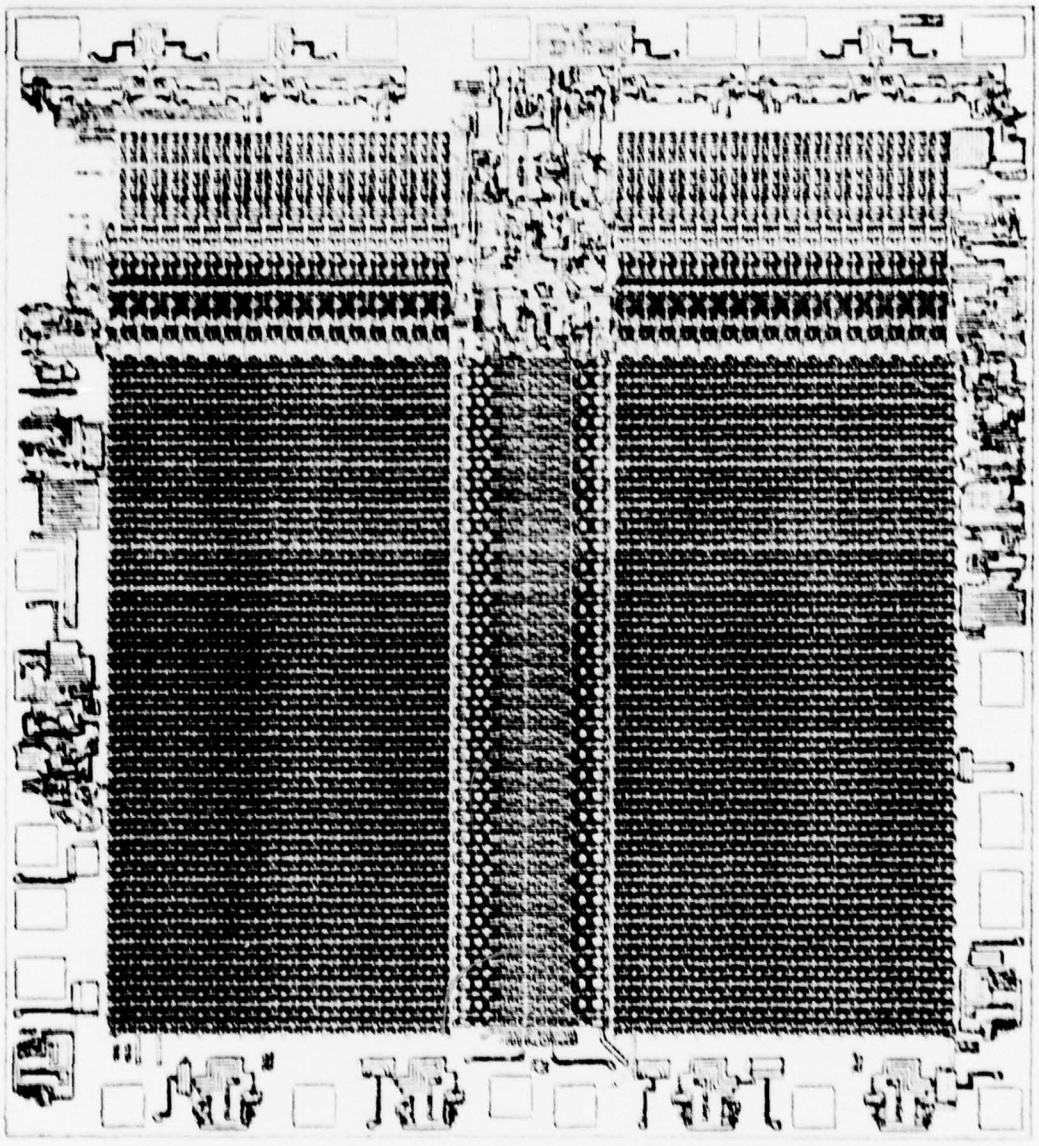
2. Medium Scale Integration (MSI) -- These circuits perform complex digital logic operations such as counting, multiplexing, encoding or decoding. MSI packages usually contain from 65 to 1024 circuit elements.

3. Large Scale Integration (LSI) -- These circuits contain up to 250,000 components and perform extremely complex operations or simply allow for large amounts of data storage in flip-flops. Microprocessors fall into this category of microelectronic circuits, as do large memory arrays. In physical size, however, LSI circuits are typically less than 20 square millimeters in area and are usually packaged in DIPs having 16 to 40 pins.

The growth of circuit complexity has been exponential since the discovery of the integrated circuit. Figure 3

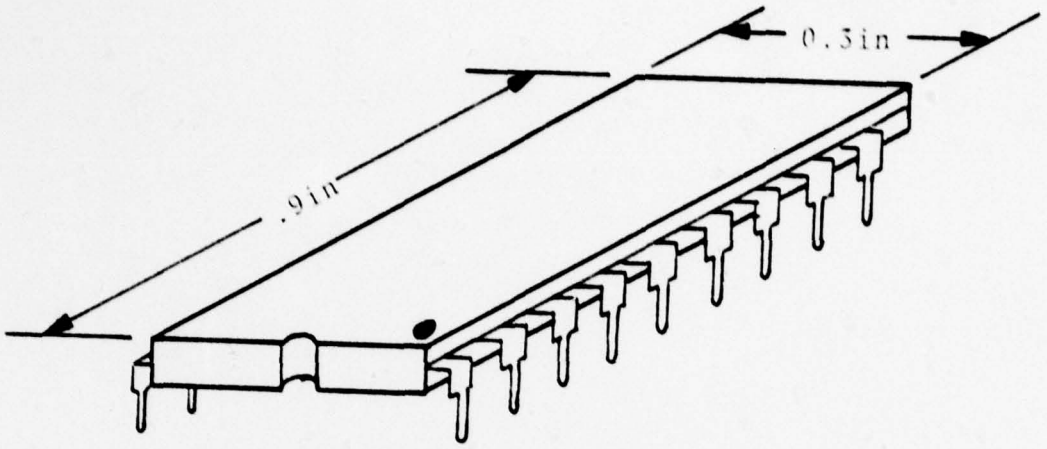
illustrates this growth since the production of the first IC in 1959. The number of components in microelectronic circuits has doubled every year over the last 19 years and the trend can be expected to continue. As integrated circuit technology improves construction techniques, the density of circuit elements on a single chip continues to improve. Also, as manufacturing methods continue to improve, the yield, or percentage of good circuits per production run, increases. For several years now, circuit design has been computer assisted, wafer manufacturing has been computer controlled and production has become less difficult. Most manufacturers have experienced a 20 to 30 percent cost reduction for every doubling of production output due to corporate learning. Figure 4 depicts the microelectronincs industry "learning curve," which again exhibits an exponential behavior.

Bits of memory per integrated circuit package have become accepted as a measure of IC complexity or size. Thus, price per bit is useful in evaluating the cost effectiveness of a microcircuit where memory is considered. Figure 5 shows the current trend in LSI memory circuit costs.

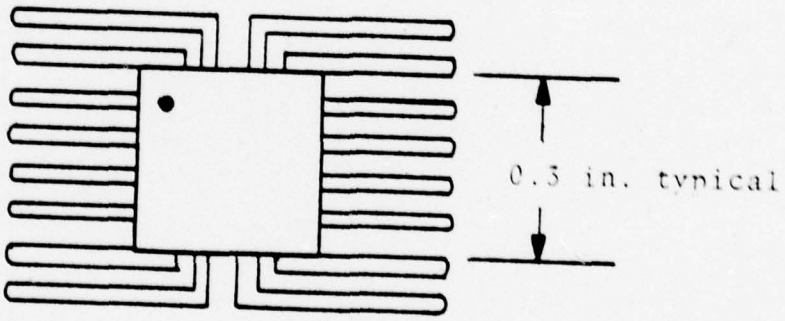


Photomicrograph of 4096-Word by 1-Bit 2107A Dynamic RAM

Figure 1 - LSI EXAMPLE (Reprinted by permission of Intel Corporation copyright 1977)



a. Dual-Inline-Package (DIP)



b. Flat-Pack

Figure 2 - INTEGRATED CIRCUIT PACKAGES

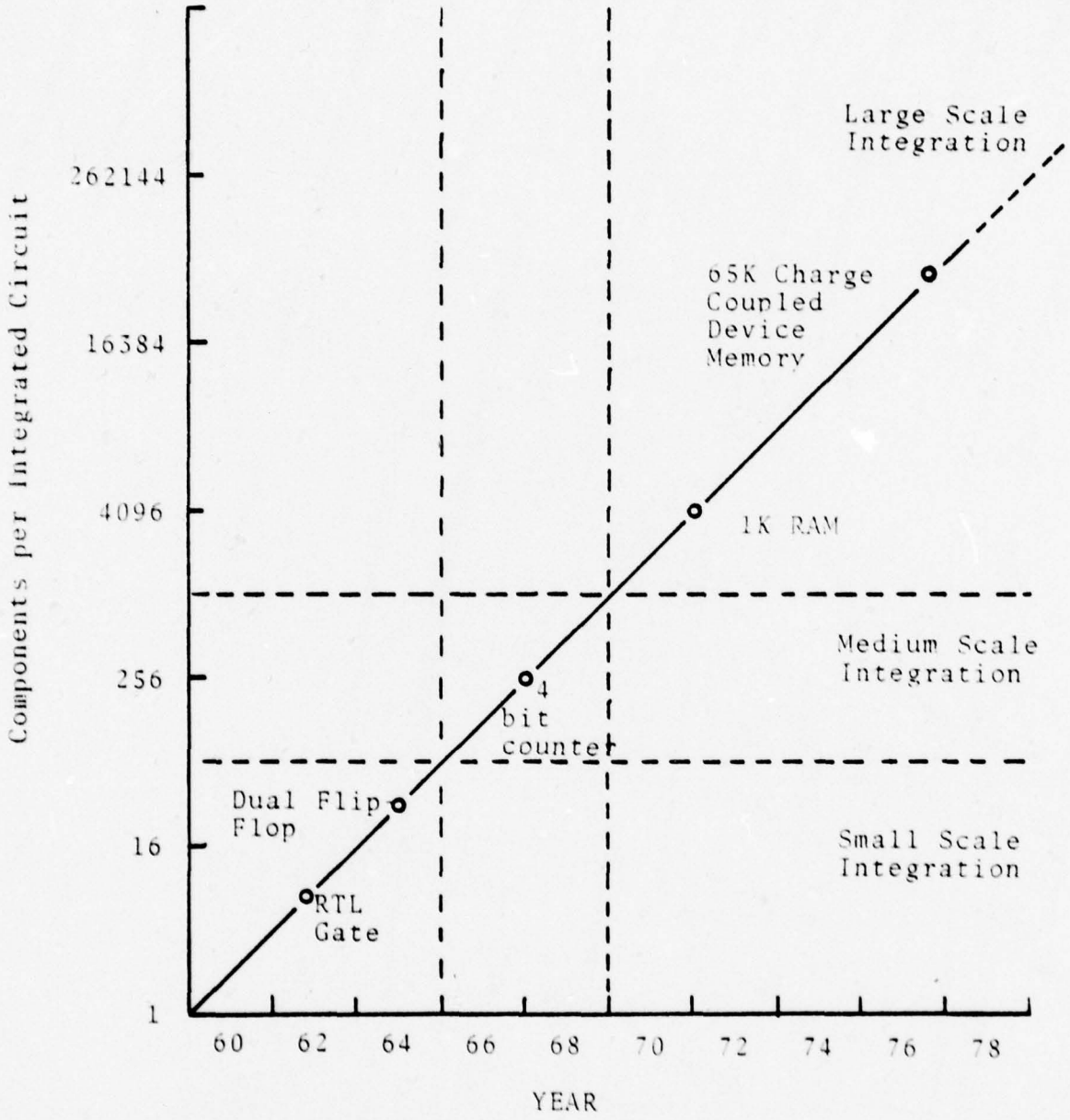


Figure 3 - Cronological Growth of Integrated Circuit Complexity

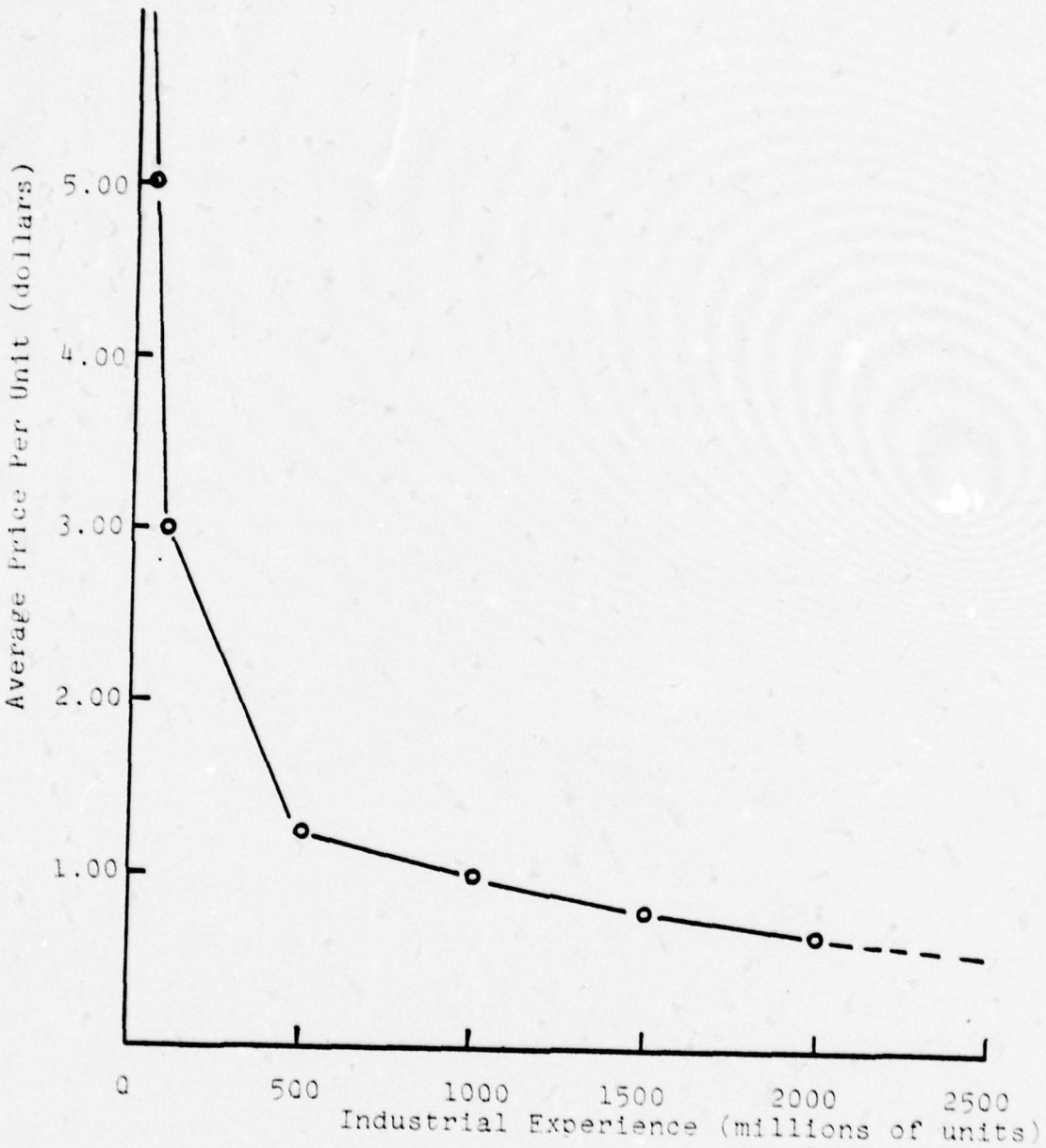


Figure 4 - MICROELECTRONICS INDUSTRIAL LEARNING CURVE

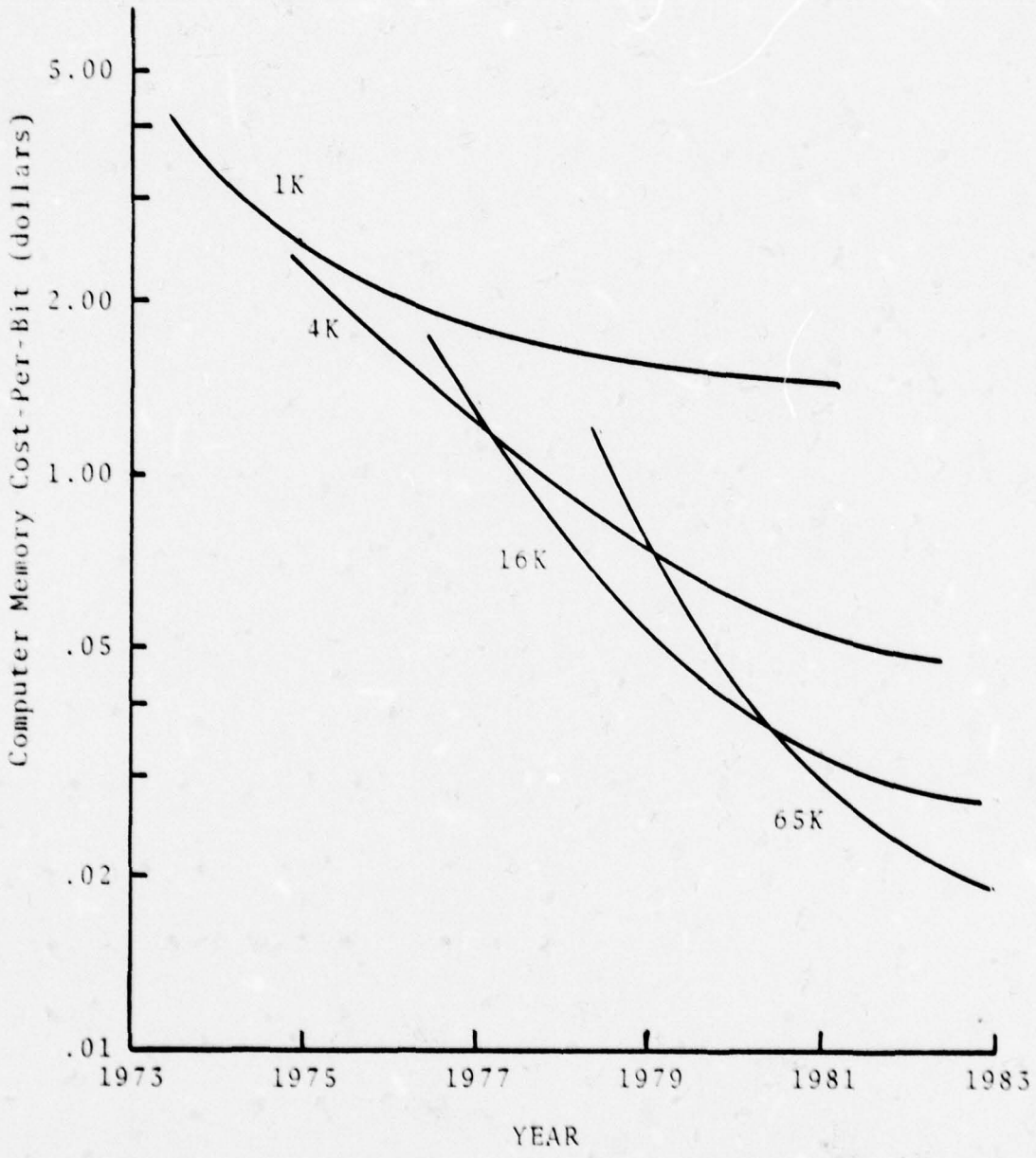


Figure 5 - COST TRENDS IN COMPUTER MEMORY

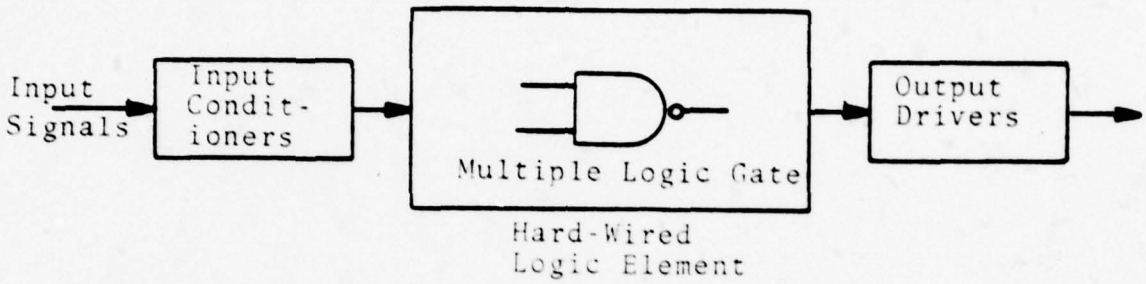
B. PROGRAMMED LOGIC

Before the advent of the microprocessor, the major expense incurred in the development of a certain functional circuit was in the design of the logic required to perform the task. The building blocks of digital circuits were formed from discrete small-scale integrated circuits and some medium scale integrated packages. In many cases, an enormous amount of circuitry was required to perform a relatively simple logical task. Circuit size is generally measured in square inches of circuit board (real estate), number of IC packages required, or the pin count on a single printed circuit board.

One of the most significant contributions of microprocessor technology has been the introduction of programmed logic as an alternative to complex circuit design using descrete small scale integrated circuits. By designing with microprocessors, great flexibility may be designed intc a functional circuit. As modifications to the design become necessary for correctional reasons or simply to enhance performance of a circuit, a change to the program will usually suffice. In the descrete design, however, a minor modification of circuit function generally required extensive redesign of the circuit and complete refabrication of the printed circuit board. This was an extremely expensive factor once the circuit had reached production level.

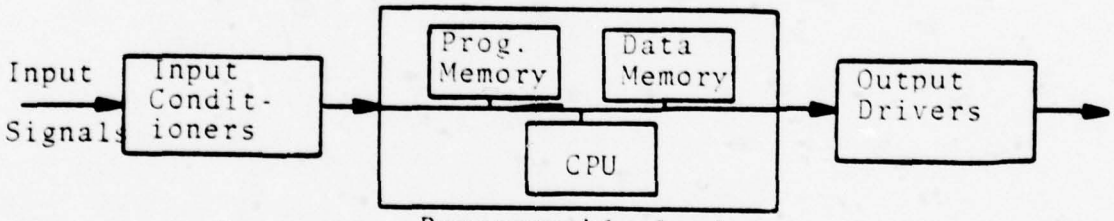
The use of microprocessors and the concept of programmed logic have contributed to the advancement of electronics on a scale comparable to that resulting from the development of the basic transistor. One basic circuit design employing a

microprocessor now serves in a seemingly endless variety of functions, differing only in the program, and the input/output interface circuitry. Figure 6 illustrates this concept.



Hard-Wired
Logic Element

1. High parts count
2. Fixed function
3. Low flexibility
4. High design cost
5. Low reliability
6. High fabrication costs
7. Modification requires significant redesign
8. High power consumption



Programmable Logic
Element

1. Low parts count
2. Variable function
3. Highly flexible
4. Very low design cost
5. High reliability
6. Low fabrication costs
7. Modification requires program change
8. Low power consumption

Figure 6 - PROGRAMMED LOGIC CONCEPT

C. COMPUTATIONAL POWER

Modern computing machines fall into three major categories depending on several distinguishing characteristics. These are: computers (full scale), minicomputers, and the newest form, microcomputers. Classification of a computing machine is generally based on the length of the word and instruction cycle time. It is typical to consider such performance parameters as instruction variety, memory size, arithmetic architecture, instruction execution speed, or even physical size and system complexity when categorizing a computing machine. The standard method of categorizing by word length, however, is illustrated in the following widely recognized classifications:

1. Computer - 32 or more bits per word
2. Minicomputer - 16 to 32 bits per word
3. Microcomputer - 4 to 16 bits per word

In general, a minicomputer is approximately 4 to 10 times as fast as a microcomputer in performing an identical computational task, while the computer is 5 to 10 times as fast as the minicomputer. When considering computational efficiency, it is common to compare relative speed for accomplishing a given task as just illustrated. It is important to note, however, that for some applications, bigger and faster are not always better. Dedicated machine control is a good example of where this might be true. Often, a mechanical system under control of a computer cannot possibly respond to instructions as quickly as the computer is capable of issuing them. The computer therefore

spends much of its time in idle loops waiting for the machine. A slow low-cost processor is better suited to this type of application. Figure 7 graphically illustrates the relative performance, cost, and functional application of the various classifications of computing machines.

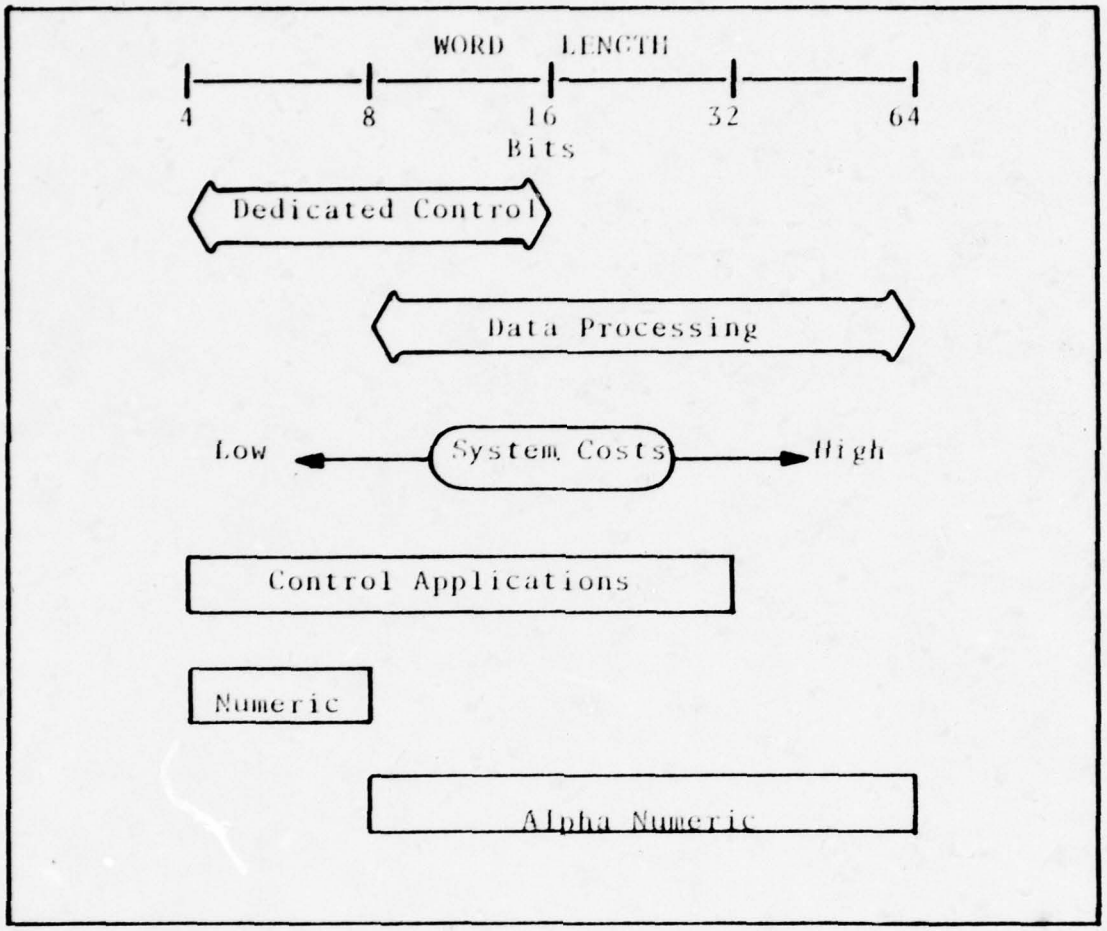


Figure 7 - RELATIVE COST, PERFORMANCE AND APPLICATION OF COMPUTER CATEGORIES

D. MICROCOMPUTER ARCHITECTURE

Mainframe computers, minicomputers, and microcomputers all have certain functional components in common with each other. These components are shown as blocks in Fig. 8.

1. Program Memory

That portion of memory allocated to the storage of instructions to be executed by the Arithmetic Logic Unit is called program memory. This memory area may be volatile random access memory (RAM), or it may be composed of non-volatile programmable read-only memory (PROM). It is even possible for the program memory to consist of both RAM and PROM , in which case, some of the program modules will be transient while others will be permanently resident.

2. Timing and Control

The timing and control logic maintains control over the program counter which points to the location in program memory where the next program instruction is to be found. After completion of an instruction or instruction group, the program counter will normally be incremented to the next succeeding memory address. In the event of a jump or call instruction, however, the program counter will be forced to the destination of the jump.

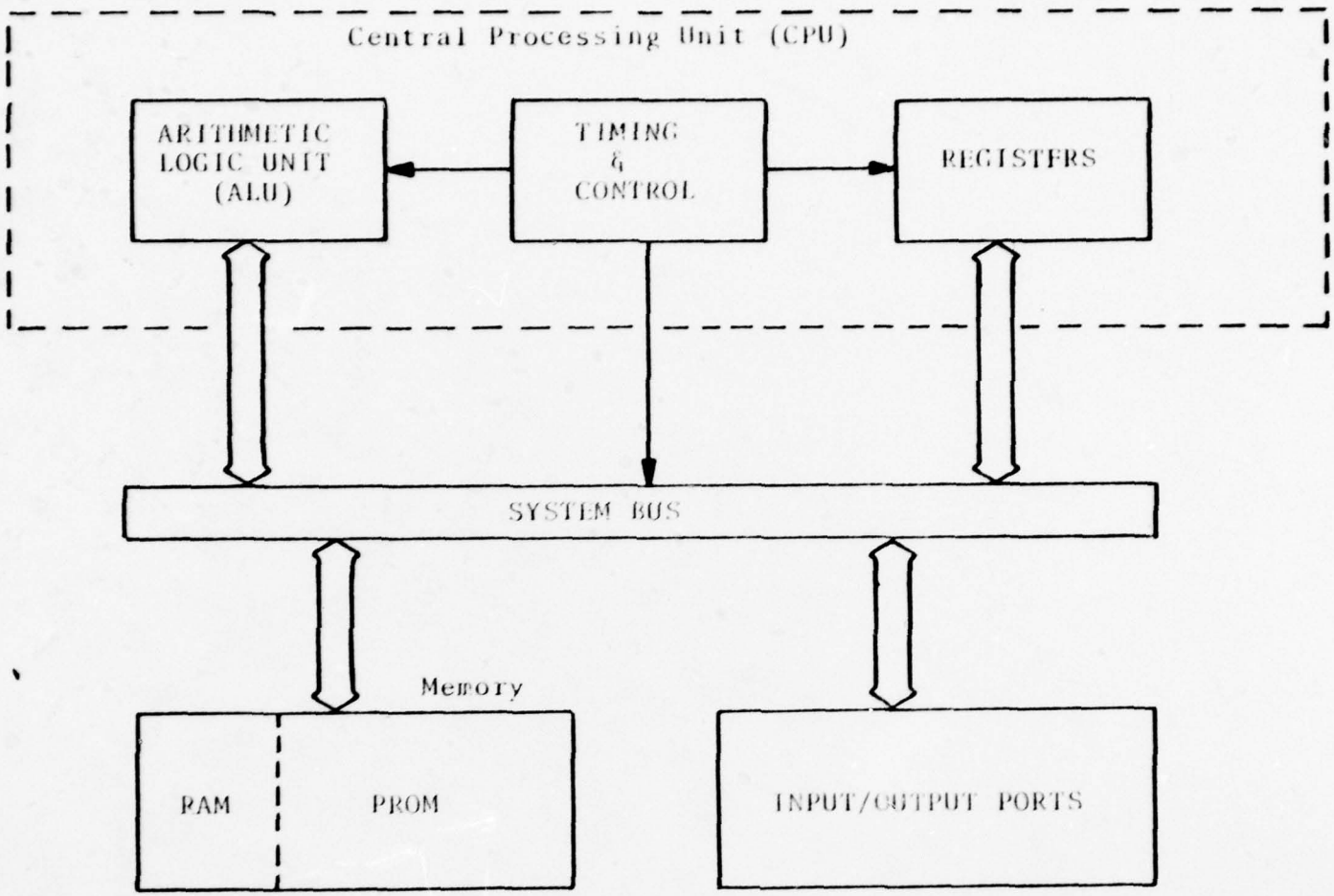


Figure 8 - FUNCTIONAL BLOCK DIAGRAM OF GENERALIZED COMPUTING MACHINE

3. Arithmetic Logic Unit (ALU)

The ALU fetches an instruction from the program memory such as an add, subtract, jump or any one of many possible simple instructions. It then performs the function as specified by the instruction and proceeds to fetch and perform the next instruction. The ALU contains most of the complex circuitry found in any computer, and traditionally has been a high-cost component of a computer.

4. Data Memory

Data memory, commonly called random access memory (RAM), is used on an as needed basis by the ALU as temporary work space to store intermediate results while performing a program sequence. It usually exhibits very fast access time, on the order of 20 to 500 nanoseconds.

5. Input/Output (I/O) Interface

The input/output interface provides connection between the computer and the various peripheral devices which make it useful. Such devices as line printers, tape drives, disk drives, solenoid controls, analog converters, switches, plotters, cathode ray tube (CRT) displays, or any conceivable electromechanical device must be made electrically compatible to the computer through the interface circuitry.

E. SINGLE-CHIP COMPUTERS

Currently, several microelectronics manufacturing firms are producing so-called single-chip microprocessors which contain all of the aforementioned functional elements in one LSI package. One such device is the Intel 8748 shown in Fig. 9. This device is typically priced in the 40 to 50 dollar range, with derivatives of the same processor priced as low as 3 dollars in large quantities. Given the appropriate program, the microcomputer can perform all the functions of a mainframe computer at a much reduced speed, of course, but at a tremendously reduced cost. Current trends are to pack more power into the single-chip microprocessor by increasing the internal memory size, improving speed and increasing I/O flexibility. As the single-chip microcomputers increase in internal program and data memory size, they become sufficiently powerful to perform any of the avionics functions. Thus, a collection of such devices could form a computational system equivalent to the present minicomputer. Judging from the recent rate of growth of single-chip computer capabilities, it is anticipated that this prospect will become practicable within just a few years.

intel

PRELIMINARY

8048/8748/8035

SINGLE COMPONENT 8-BIT MICROCOMPUTER

***8048 Mask Programmable ROM**
***8748 User Programmable/Erasable EPROM**
***8035 External ROM or EPROM**

- **8-Bit CPU, ROM, RAM, I/O in Single Package**
- **Interchangeable ROM and EPROM Versions**
- **Single 5V Supply**
- **2.5 µsec and 5.0 µsec Cycle Versions All Instructions 1 or 2 Cycles.**
- **Over 90 Instructions: 70% Single Byte**
- **1K x 8 ROM/EPROM 64 x 8 RAM 27 I/O Lines**
- **Interval Timer/Event Counter**
- **Easily Expandable Memory and I/O**
- **Compatible with MCS-80™ Peripherals**
- **Single Level Interrupt**

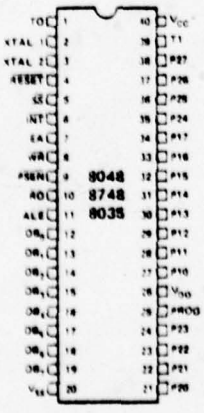
The Intel® 8048/8748/8035 is a totally self-sufficient 8-bit parallel computer fabricated on a single silicon chip using Intel's N-channel silicon gate MOS process.

The 8048 contains a 1K x 8 program memory, a 64 x 8 RAM data memory, 27 I/O lines, and an 8-bit timer/counter in addition to on board oscillator and clock circuits. For systems that require extra capability, the 8048 can be expanded using standard memories and MCS-80™ (8080A) peripherals. The 8035 is the equivalent of an 8048 without program memory.

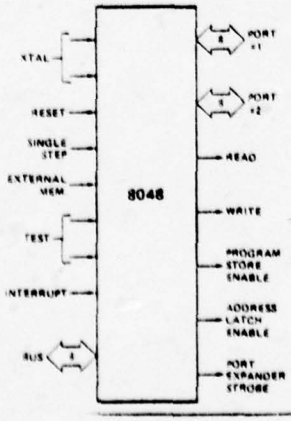
To reduce development problems to a minimum and provide maximum flexibility, three interchangeable pin-compatible versions of this single component microcomputer exist: the 8748 with user-programmable and erasable EPROM program memory for prototype and preproduction systems; the 8048 with factory-programmed mask ROM program memory for low-cost high volume production; and the 8035 without program memory for use with external program memories.

This microprocessor is designed to be an efficient controller as well as an arithmetic processor. The 8048 has extensive bit handling capability as well as facilities for both binary and BCD arithmetic. Efficient use of program memory results from an instruction set consisting mostly of single byte instructions and no instructions over two bytes in length.

PIN CONFIGURATION



LOGIC SYMBOL



BLOCK DIAGRAM

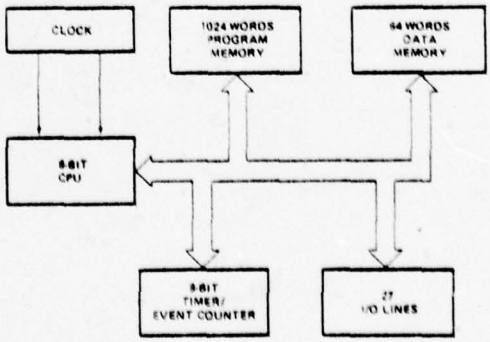


Figure 9 - INTEL 8748 SINGLE-CHIP MICROCOMPUTER (Reprinted by permission of Intel Corporation copyright 1977)

F. PROGRAMMING LANGUAGES

1. Low-level Languages

a. Machine Code

The computing machine must be provided with a series of instructions which the arithmetic logic unit is to perform. The representation of the program as it exists in executable form within the program memory is called machine code. If a computing machine has a large instruction repetoire, programming in machine code is a very cumbersome process. This is because the instructions exist as numbers and memorization of the instruction set is difficult. Most microprocessors have instruction sets exceeding 70 operation codes. Of significance here, is the fact that different computers, minicomputers, and microprocessors each have distinctly different instruction sets. Machine language programming is rarely utilized except during initial development of a system design. This is the most rudimentary form of programming on any computer, and although infrequently used, it is usually well understood by persons engaged in low-level language programming.

b. Assembly Languages

The second most basic level of computer programming employs assembly languages. As in machine language programming, the language itself is usually

processor dependent. The assembly language is essentially similar to machine code, however, instead of using actual machine interpretable numbers for input, the programmer uses mnemonics. Mnemonics are abreviated alphanumeric words which serve as memory aids for programming on the single instruction level. Once the program is written, it is processed by another program called an assembler, which converts the mnemonic instructions into machine code. Assembly language, although still considered a low-level programming language, is in very common usage. When a new processor is introduced, most programming will be accomplished in assembly language until higher level languages are adapted to the processor. This is primarily due to the relative ease of developing the assembler program compared to construction of a higher-level language compiler.

2. Medium-Level Languages

The medium-level languages, such as PL/M and PL/Z, offer the programmer additional programming development facilities. Generally they are capable of translating English-like statements and mathematical equations into machine code program segments. This capability is usually limited to integer arithmetic operations such as add, subtract, multiply, and divide. Floating-point operations are not included within the language and all operations are byte or double byte structured rather than word or field oriented. Medium-level languages are useful in microprocessor applications which require a more extensive program in which documentation quality is important.

3. High-Level Languages

This group of computer programming languages includes most of the well-known languages such as FORTRAN, COBOL, PASCAL, BASIC and CMS-2, the Navy's standard tactical computer language. Languages within this category are capable of translating complex algebraic equations into executable machine code. This allows large programming tasks to be accomplished with less effort. Beyond this obvious advantage, the use of high-level languages enables program transferability between different computers, assuming that a compiler program exists for each computer of distinct architecture. Usually, the transfer of programs can be accomplished with little or no modification of the original program. Although the programming process becomes much more efficient using a high-level language, the execution efficiency of the compiler produced code is reduced over that of the low or medium-level languages.

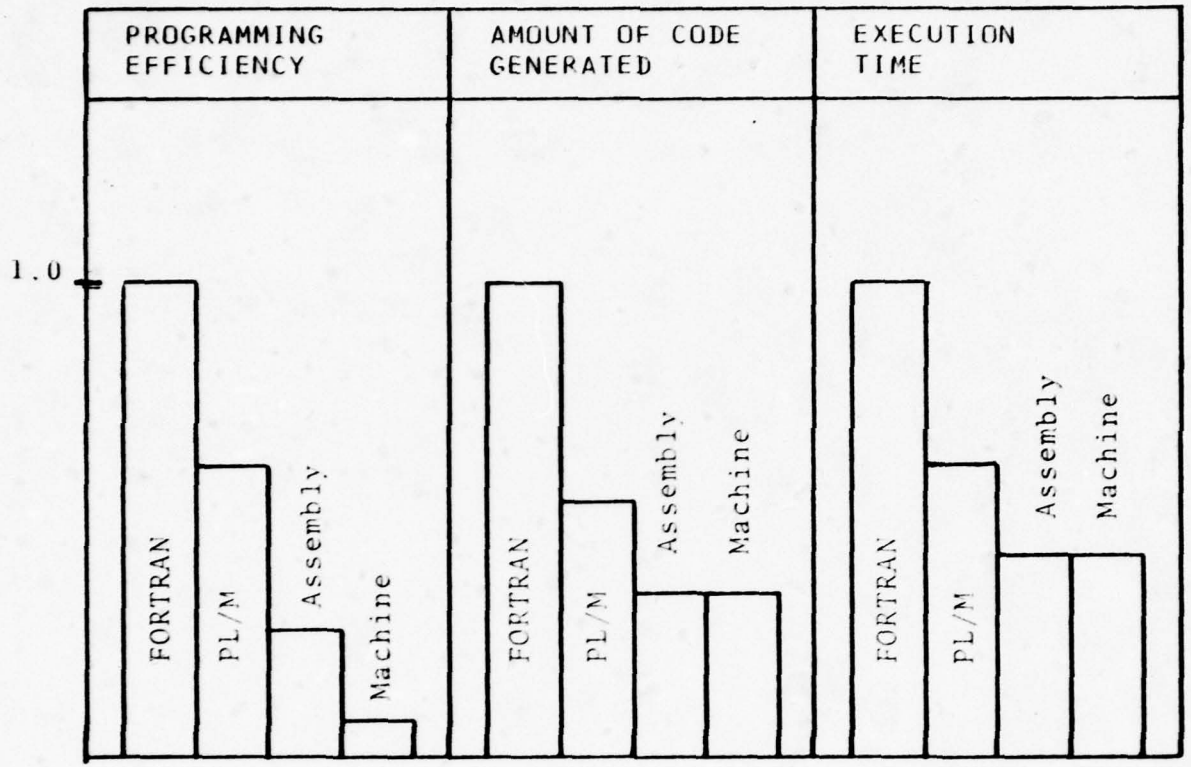


Figure 10. COMPARISON OF LANGUAGE LEVELS

III. TACTICAL AIRBORNE COMPUTER SYSTEMS

A. TYPICAL DEPLOYED AIRCRAFT SYSTEMS

The technological capability of miniaturizing a computer to a size and weight compatible with tactical jet aircraft caused immediate application in Naval and Air Force avionics designs. In the Navy's A-7E, for example, a relatively small (approximately two cubic feet) general purpose central computer controls many of the cockpit visual displays, performs navigational computations, performs ballistic calculations and effects automatic weapon release for several different types of guided and unguided weapons. The A-6E on-board computer is very similar in function to that of the A-7E in that it consists of one central computer surrounded by many analog or digital peripheral devices. Figure 14 shows the A-7E avionics system in block diagram form.

The A-6E and A-7E systems effected major advances in tactical aircraft weapon delivery accuracy and overall aircraft mission performance. These aircraft demonstrated that any future tactical aircraft must be equipped with some form of a digital computing system.

Subsequent Naval aircraft, and the number of computers incorporated in their avionics systems include: E-2 (3), S-3 (5), and the F-14 (6). The number of computers on board each aircraft shown in parentheses does not include embedded microprocessors, that is, microprocessor devices used in

peripheral equipment and distributed throughout the aircraft. The F-14, for example, has a separate computer dedicated to each of the following functions: navigation, combat system, engine monitoring and control system, displays control, wing sweep control, and weapon delivery control.

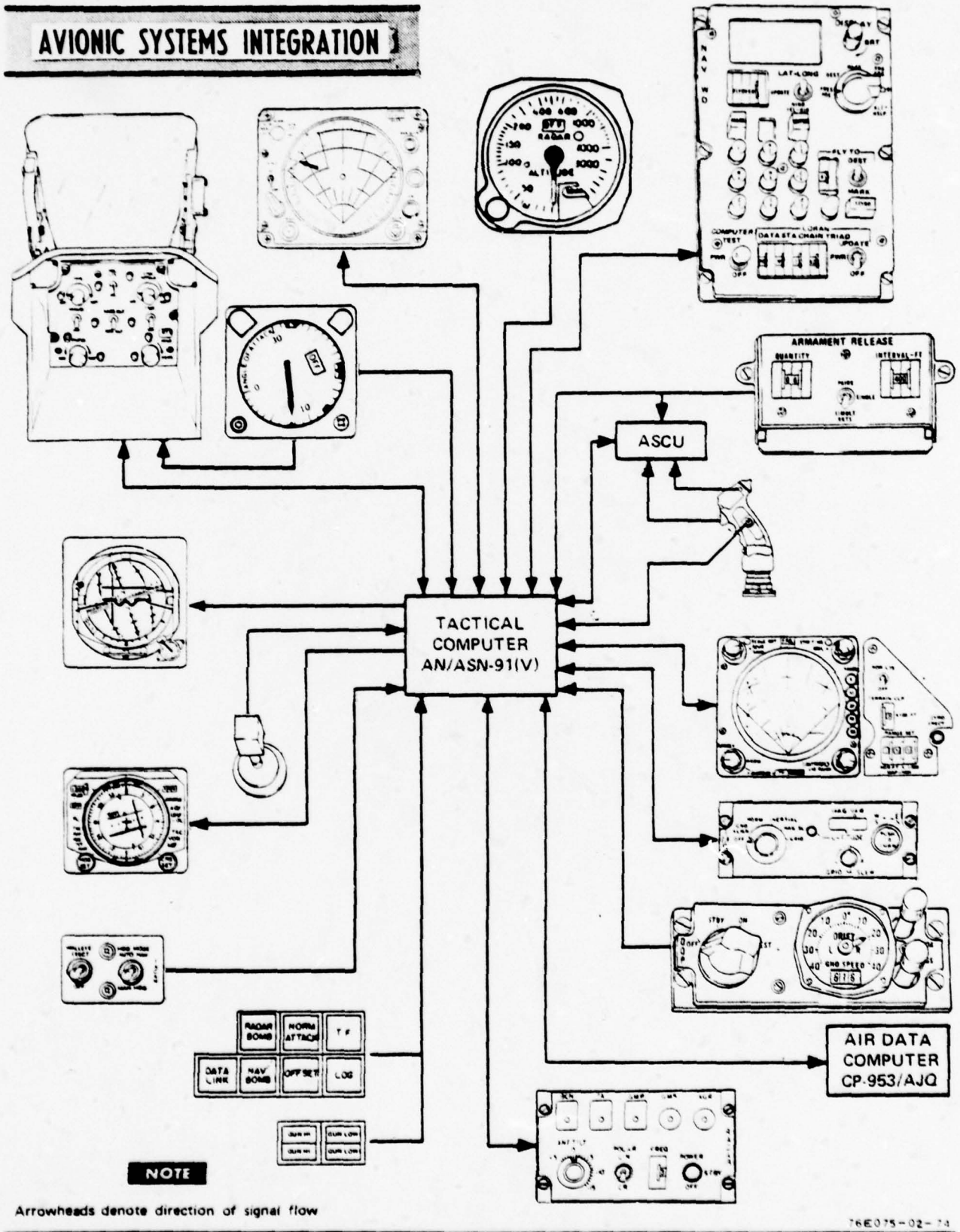


Figure 11 - TYPICAL TACTICAL AIRCRAFT COMPUTERIZED AVIONICS SYSTEM (A-7E)

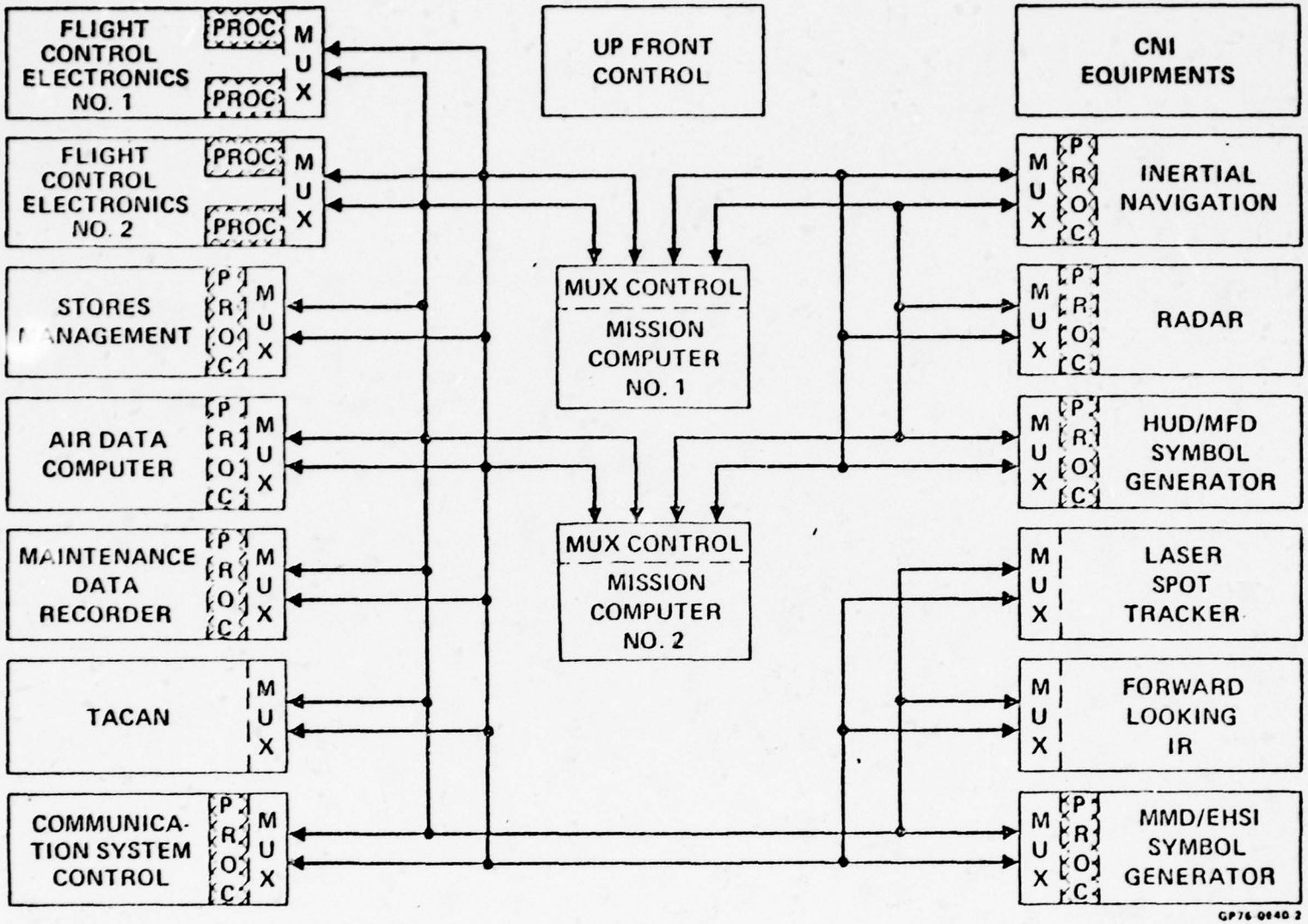


Figure 12 - F-18 AVIONICS SYSTEM

E. NAVY STANDARDIZATION OF TACTICAL COMPUTERS

With the rapidly growing proliferation of computing devices, it became apparent that controls must be placed on the variety of computing machines utilized in Naval aircraft and other avionics systems. In order to minimize logistical and maintenance problems, the Navy planned for standard computers to be used in all of its tactical systems. This standardization plan provided for shipboard and shore-based large scale computers, the AN/UYK-7, minicomputers, the AN/UYK-20; as well as the standard AN/AYK-14 for airborne systems.

In addition to standardization of the computing machines, the Navy also standardized its programming language named CMS-2. This high-level language family in its various forms, was intended to serve as a common language link between the AN/UYK-7, AN/UYK-20 and the AN/AYK-14. Due to hardware differences, among the three computer types, complete program transferability was never achieved. The CMS-2 compilers and cross-compilers convert CMS-2 language programs into the machine language required by the intended computer.

C. AN/AYK-14 AIRBORNE TACTICAL COMPUTER

The AN/AYK-14 is the Navy's designated airborne general purpose computer. It is implemented using bit-slice (described below) large scale integrated circuit technology, and was designed to be functionally equivalent to the physically larger AN/UYK-20 minicomputer. Because the

AN/AYK-14 uses the same operation codes as the AN/UYK-20, program transfer between the two computer types is usually possible. This feature was one of the major goals of the standardization program.

The bit-slice architecture is primarily responsible for this designed-in capability in the case of the AN/AYK-14. Using Advanced Micro Devices 2900 series bit-slice microprocessor enabled designers to construct a customized computer with user defined operation codes and performance. In short, bit-slice techniques allow the user to define the architecture of the processor through variations of component interconnection and micro-programming code. Figure 13 shows a typical bit-slice processor consisting of several compatible single-chip components. Expansion of the bit-slice system is accomplished by annexing additional central processing elements (CPE), one for every two or four bits. Each CPE is capable of executing only a few basic instructions including two's complement arithmetic, boolean operations, shifting left or right, and bit and zero checking. The microprogram control unit feeds the central processing elements with the desired sequence of microinstructions derived from the microprogram memory. In this way, the actual internal functioning of the "customized" microprocessor is determined by the designer. Thus, the microprocessor can be designed to respond to most predetermined sets of instructions.

Another of the major advantages of the bit-slice machine is its expandability in terms of word length without significant loss of speed. Each central processing element executes the same instruction sequence but only operates on a two or four bit slice of the computer word. The variable word length can effectively increase throughput rate and thereby improve the overall computational power of the computing machine being designed.

intel

3001

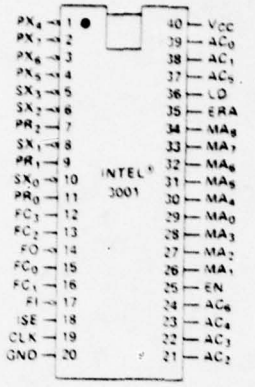
MICROPROGRAM CONTROL UNIT

The INTEL® 3001 Microprogram Control Unit (MCU) controls the sequence in which microinstructions are fetched from the microprogram memory. Its functions include the following:

- Maintenance of the microprogram address register.
- Selection of the next microinstruction based on the contents of the microprogram address register.
- Decoding and testing of data supplied via several input busses to determine the microinstruction execution sequence.
- Saving and testing of carry output data from the central processor (CP) array.
- Control of carry/shift input data to the CP array.
- Control of microprogram interrupts.

- High Performance – 85 ns Cycle Time
- TTL and DTL Compatible
- Fully Buffered Three-State and Open Collector Outputs
- Direct Addressing of Standard Bipolar PROM or ROM
- 512 Microinstruction Addressability
- Advanced Organization
  - 9-Bit Microprogram Address Register and Bus
  - 4-Bit Program Latch
  - Two Flag Registers
- Eleven Address Control Functions
  - Three Jump and Test Latch Functions
  - 16-way Jump and Test Instruction Bus Function
- Eight Flag Control Functions
  - Four Flag Input Functions
  - Four Flag Output Functions
- 40 Pin DIP

PACKAGE CONFIGURATION



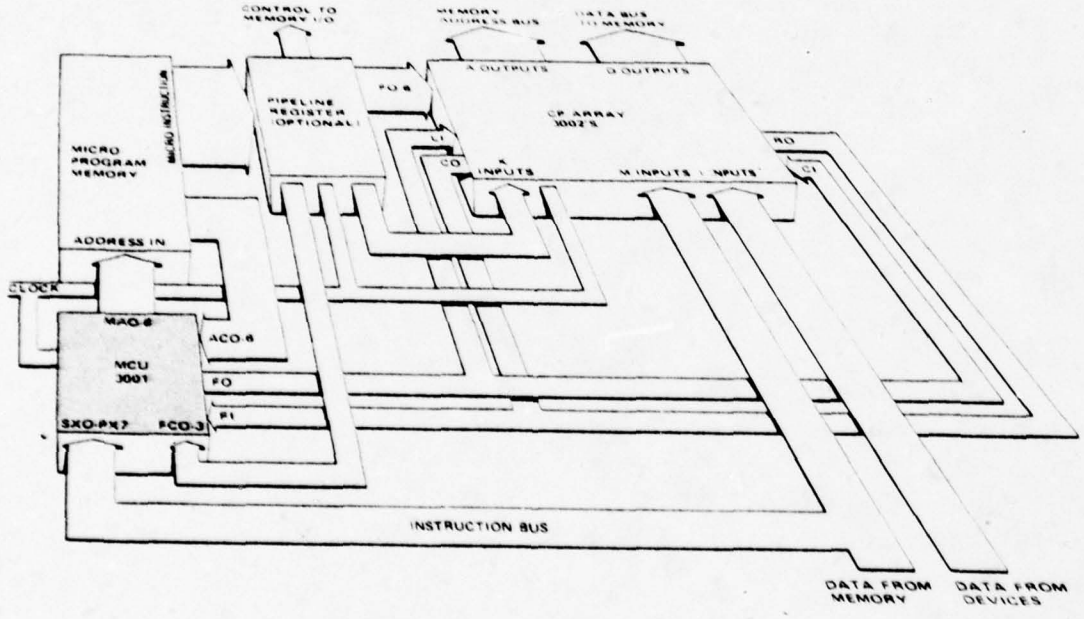


Figure 1. Block Diagram of a Typical System

Figure 13 - BIT-SLICE MICROPROCESSOR ARCHITECTURE
(reprinted by permission of Intel Corp. copyright 1977)

The major disadvantage of such a system lies in the additional level of design which must be performed in order to obtain a workable computer. The consequence of this is that each developer must invest a great deal in software development tools just to bring the processor to the usable level of operation. Large effort must be expended in developing operation codes, assembly language mnemonics, assemblers, cross-assemblers, compilers and debugging programs before any real application of the system can be realized. The AN/AYK-14 development has surpassed this level and indeed, much has been invested in the system development program.

1. Alternative Avionics Systems

In the previous section we examined the historical development of computerized tactical avionics systems. Recent developments in microprocessor architecture, namely distributed processing and concurrent processing techniques, have suggested alternatives to the conventional central computer concept presently employed in tactical aircraft systems. This section describes one generalized alternative design which can be easily expanded or diminished to suit the requirements of the particular airframe or tactical mission. This design concept will be used as a comparison model in subsequent analyses and discussion.

a. Single-Board Computer (SBC)

From the single-chip microprocessor, several manufacturers, led by Intel Corporation and Texas Instruments, have produced single-board computers. These are typically constructed on a single printed circuit card generally less than 80 square inches in area and composed of several MSI and LSI components. Each single-board computer contains a central processing unit, random access memory, program memory, parallel input/output ports, serial input/output ports, and a multiplexed bus interface for common memory access and inter-board communication. The current trend in single-board computer design is to reduce the number of discrete components while increasing the amount of on-board memory and peripheral communication capability. Figure 14 shows a typical single-board computer and its associated block diagram.

Very recently, several LSI manufacturers have marketed single-chip "computers", which implement all of the SBC features in a single integrated circuit package. Intel's 8048 series is such a family of devices. The Intel 8022 for example, contains 64 bytes of RAM, 2048 bytes of PROM, three parallel ports, and even an analog to digital conversion section. The trend in the industry is to continue packing more memory and speed into the single-chip devices. It is anticipated that concurrent processing will soon be possible on a single printed circuit board by arranging several single-chip computers on a card.

intel

SBC 80/20 SINGLE BOARD COMPUTER

8080A CPU

2K bytes static read/write memory

Sockets for 4K bytes of erasable reprogrammable or masked Read-Only-Memory

48 programmable parallel I/O lines with sockets for interchangeable line drivers and line terminators

Programmable synchronous/asynchronous RS232C compatible serial interface with fully software-selectable baud rate generation

Full Multi-Master Bus control logic which allows up to 16 masters to share system bus

Eight-level programmable interrupt control

Two programmable 16-bit BCD and binary timers

Auxiliary power bus, memory protect, and Power-Fail Interrupt control logic provided for battery back-up RAM requirements

Compatible with optional memory and I/O expansion boards

The SBC 80/20 is a member of Intel's complete line of OEM computer systems which take full advantage of Intel's LSI technology to provide economical, self-contained computer based solutions for OEM applications. The SBC 80/20 is a complete computer system on a single 6.75 X 12-inch printed circuit card. The CPU, system clock, read/write memory, nonvolatile read-only-memory, I/O ports and drivers, serial communications interface, priority interrupt logic, two programmable timers, multi-master bus control logic, and bus expansion drivers all reside on the board.

Intel's powerful 8-bit n-channel MOS 8080A CPU, fabricated on a single LSI chip, is the central processor for the SBC 80/20. The 8080A contains six 8-bit general-purpose registers and an accumulator. The six general-purpose registers may be addressed individually or in pairs, providing both single and double precision operators. Minimum instruction execution time is 1.86 μsec.

The 8080A has a 16-bit program counter which allows direct addressing of up to 65,536 bytes of memory. An external stack, located within any portion of read/write memory, may be used as a last-in/first-out storage area for the contents of the program counter, flags, accumulator, and all of the six general-purpose registers. A 16-bit stack pointer controls the addressing of this external stack. This stack provides subroutine nesting that is bounded only by memory size.

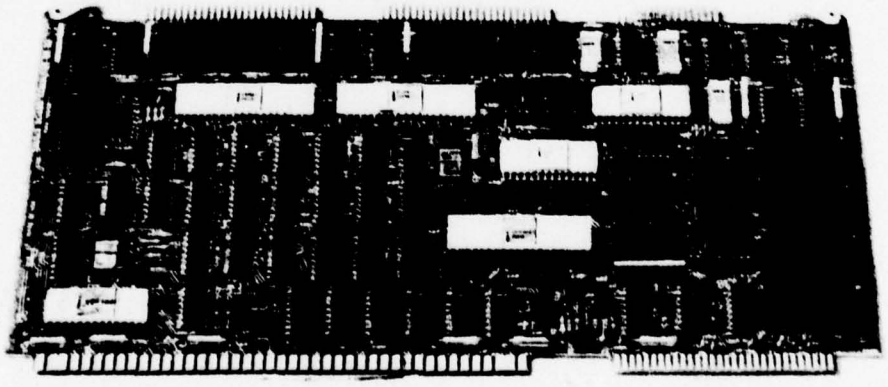


Figure 14 - INTEL SBC80/20 SINGLE-BOARD COMPUTER (Reprinted by permission of Intel Corporation copyright 1977)

b. Concurrent processing

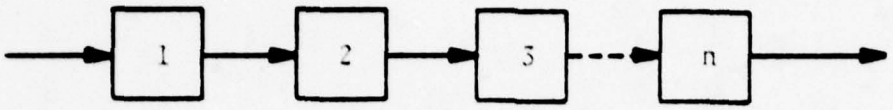
Concurrent processing is simply the technique of putting more than one processor or CPU to work at performing a job normally done by one computer. There are several ways in which this may be accomplished depending on the interconnection structure of the processors used. Figure 15 depicts three possible arrangements of several processors configured for concurrent processing.

The processing power of such a system depends on redundancy of central processing units. Each processing unit performs a function independent of the others but communicates with others, either directly or through means of a common memory block. By sharing the computing load, the throughput or the overall speed of the system, depends on the number of processing elements in the system, the information transfer rate on the system bus and the actual bus usage factor.

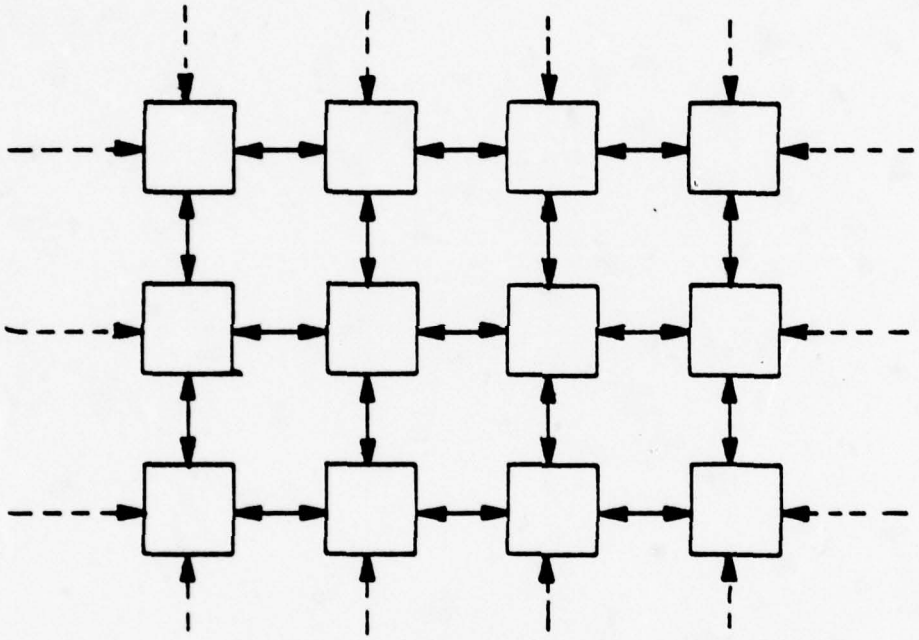
An all important aspect of a concurrent processing system is the degree of homogeniety achieved by using common building blocks in its implementation. This homogeniety of components can effect substantial reductions in the life-cycle cost of a computing system. The cost reduction factors are discussed in further detail in the forthcoming sections of this thesis.

As a model for further discussion, we will refer to the generalized avionics processing system shown in Fig. 16. The structure shown provides for several affinity groups of concurrent processors, possibly distributed physically throughout the aircraft, each communicating to the other via high-speed fiber optic cable. Within each

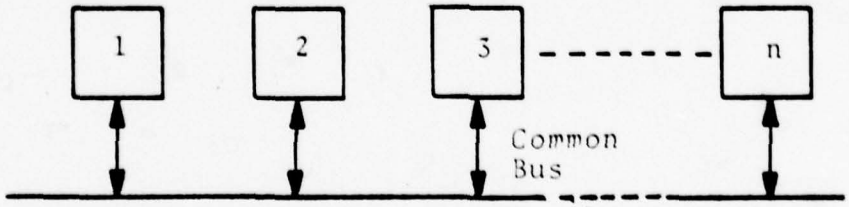
affinity group, there are several single-board computers each performing a specialized function. For example, SBC number 1 may perform the navigation computations by taking data from the inertial navigation gyros and accelerometers and passing information to be displayed on the pilot's head-up-display (HUD) unit to the common memory "mail box". SBC number 4, which controls the cockpit displays picks up the data deposited by SBC number 2 and sends the appropriate control signals to the video displays in the cockpit. Likewise, the weapons delivery computer can access the same data deposited by number 2 and perform ballistic calculations and automatically control bomb delivery.



a. Pipeline Configuration -- Data are passed along from one processor to another, each processor performing a part of the overall operation.



b. Array Configuration -- Each processor performs the same operation depending on neighboring processors for data.



c. Independent Configuration -- Each processor performs an independent function under its own program. This is the most versatile form of concurrent processing.

Figure 15 - CONCURRENT PROCESSING TECHNIQUES

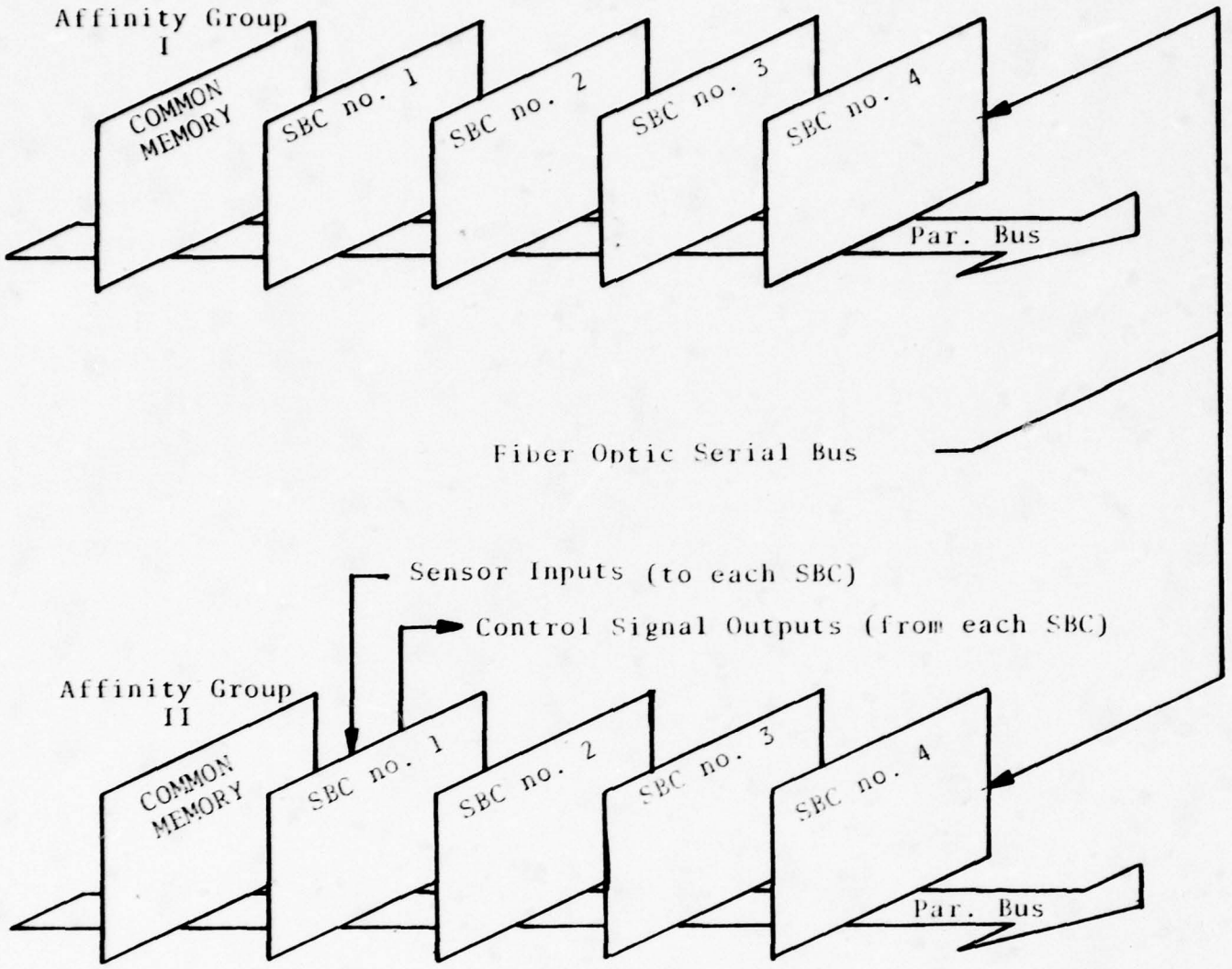


Figure 16 - GENERALIZED CONCURRENT PROCESSING AVIONICS SYSTEM

IV. LONG-RANGE PLANNING

A. PROBLEM STATEMENT

One of the major goals of the Navy's tactical computer standardization program discussed in the previous section, was to reduce costs through elimination of distinct computers in tactical computer applications and to allow common support software procurement for these machines. Indeed, the basic concept is sound, however, a major flaw exists in the implementation of the plan. That is, the plan does not provide adequately for technological growth on a scale experienced during the "LSI revolution."

New LSI devices are being developed at an alarming rate, each with more capabilities than their predecessors. Circuits are literally obsolete within one or two years of initial production. Similarly, advances in programming techniques are taking place but at a much slower rate. Program maintenance and development is certain to become more important as time progresses. A third related area of progress has been in the architecture of the overall computing machine. Advances in data bus structures, fiber optic communications, distributed processing, concurrent processing, and array processing have opened new horizons in the data processing field. The development of these and other devices or techniques have changed circumstances so dramatically, that the AN/AYK-14 standardization plan may not be long-lived. It is the author's opinion that the newly achievable architectures will be incorporated in the

next generation of tactical military aircraft.

It is the purpose of the remainder of this thesis, therefore, to examine the various alternatives made available by LSI technological advances, and relate the cost, performance and growth benefits of each with those of the present Navy standardization plan.

The concurrent processing computer design concept presented in the previous section as an alternative to the centralized computer, must be considered as a prime candidate for incorporation in the next generation of tactical military aircraft. In this section, we will explore some of the major cost/performance tradeoffs between the concurrent processing and the centralized computer design concepts. Before beginning this discussion, however, it must again be emphasized that the field of microelectronics is still growing at an accelerating rate. Any long-range planning effort must take this fact into account. It is quite probable that within the next decade, an altogether new alternative to either the centralized or concurrent processing techniques will emerge. Another important factor is the ever increasing impact of programming. Programming costs are continuing to become a larger proportion of the overall system cost.

We will begin by identifying some of the cost and performance factors associated with the two alternative models. It is hoped that the following discussion will aid in the decision making for those involved in the long-range planning effort for future avionics systems acquisition programs.

B. METHODOLOGY

Recent studies by Genovese [4], and Kodres, Buttinger, Hamming and Jones [2] have concluded through cost and performance analyses, that a homogeneous microprocessor-based concurrent processing system could be up to twice as cost-effective as a central minicomputer. The methodology used herein will be to present a narrative rationale which contributes to the support of the alternative concurrent processing avionics system. We will look at the broad categories of acquisition and maintenance cost considerations of both hardware and software. In the final chapter, we will draw appropriate conclusions and present possible courses of action to be considered as viable in the procurement processes of future avionics computer systems.

We will work under the assumption that the single-board computer modules used in our alternative model would achieve Navy-wide standardization status with hopes of DOD or possibly even industrial standardization. In any case, the components used in the single-board computer modules would be commercially available, high-production, low-cost components. It is important to note, that since each module would be capable of functioning as an independent device, it could find an even wider application than that of avionics systems. Some systems may require ten or more identical modules while others may only require one or two, depending on the complexity of the function. This would result in a much higher production base than the pursuit of the specialized central computer alternative. This predicted higher production base will be the over-riding argument in favor of the concurrent processing alternative in the

forthcoming discussion.

C. HARDWARE COST FACTORS

1. Hardware Acquisition

The research and development phase of the acquisition process of an airborne computer system will be significantly affected by choice of computer architectures. In the case of the AN/AYK-14, this phase is mostly history, whereas for the concurrent processing model, we can only estimate the effects of its implementation on procurement costs.

The Navy's AN/AYK-14 standard, although incorporating the latest and fastest in bit-slice microprocessors, is unique in that the design was forced to accomodate the AN/AYK-20 instruction set and software. It therefore became a "Navy" computer with production estimated at most to be approximately 6500 units [3]. This estimation assumes that the AN/AYK-14 will definitely be installed in the F-18, LAMPS, MKIII, IEWS, and numerous other airborne applications. A more conservative estimate, allowing for program discontinuances and project funding cutbacks would place the figure at about 3000 units. A committment has been made to the AN/AYK-14 program and time will reveal the actual cost per computer. Present estimates place the acquisition costs alone at over $50,000 per copy. [2]. The production base for the present Navy program will be solely determined by the procurement action of the Department of Defense.

It stands to reason, however, when dealing with

devices which are commonly used throughout industry, the 8080 microprocessor for example, the unknowns in a system's development are significantly reduced. The risk of failing to meet design objectives are substantially lessened. By working with a smaller, less complex module, the development of the single-board computer itself would be a comparatively minor task. Most of the expense in developing a working single-board computer is in the design and development of the LSI components themselves. These costs, however, are shared by industrial users and again become insignificant for large production items. The hardware research and development costs to be considered, then, are in the design and development of the various systems formed from the single-board computer modules. This includes interface to the many peripheral systems attached to the computer. The engineering task of developing a working concurrent processing system is currently being performed by the private industrial sector out of commercial interest in the technique. The single-board computer module capable of performing concurrent processing has been on the market for over one year as of this writing. Intel Corporation's SBC-80/20 computer can operate in a system containing up to 16 individual master computers sharing the same bus structure and a common memory.

Militarizing the hardware clearly becomes an important cost issue. Militarized versions of many LSI components are now becoming available from the larger manufacturers such as Intel and Texas Instruments. The manufacturers are correctly anticipating greater usage of microprocessors, LSI memories and peripheral interface components in military applications. Industrial users, namely the automobile industry, have need for miltary hardened components in harsh environment applications. Conformance to Mil standard 833 is becoming common place among LSI suppliers. Of course, a premium price is placed

on militarized, JAN approved devices, but, as usage increases, methods of improving yield will surely reduce the added costs.

The point should be clear by now. If the Navy were to implement a computer system which utilized the industrial standard components, the research and development costs would be for the most part borne by the entire industrial complex. The cost savings to the government would be substantial as expressed in Refs. 2 and 4.

The same reasoning holds true for other cost factors associated with acquisition. These include: fabrication, test equipment, development and production planning, technical manuals, training materials, initial training courses, and others.

2. Hardware Maintenance

Maintaining a deployed avionics system is a complex activity and entails the upkeep and support of nearly every electronic device in an aircraft. We could consider the support of instrumentation, various sensors, radar, communications, and inertial navigation subsystems in addition to numerous other subsystems in treating the maintenance costs of the total avionics system. For the purposes of this thesis, however, we will put aside the question of total support and concentrate only on those aspects affected by the choice of computer system architecture.

As in the acquisition cost factor discussion, the cost of maintaining the hardware of a composite microprocessor-based computer system is greatly influenced by the use of industry supported components. Since modules

are relatively simple in construction, varying only in program content, trouble shooting is simplified on an organizational level by using replacement techniques. Intermediate maintenance facilities would maintain module test equipment which could accomodate all variations of the basic module by running diagnostic programs to verify correct operation of components on the board. Should simple automated testing of a module fail to identify the malfunctioning component, or if damage were extensive, the relatively small cost of each module would justify discard and replacment from spare parts. This concept avoids high-level training requirements for both organizational and intermediate level maintenance activity personnel. Similarly, the test equipment required at either level is minimized.

Another important aspect of maintainability is the mode of failure which may be expected from use of the modular system. Since the system with modules removed is extremely simple in construction, consisting mostly of mounting hardware and power supply components, nearly all failures can be expected to occur within a module. Failure of a single module would not have a great effect on the overall performance of the mission computer. Thus, graceful degradation is clearly accomplished without special design effort. Should a particular module be responsible for a critical function, it alone could be duplicated within the system. This selective redundancy is easily more desirable than duplicating the entire mission computer as required with the central computer concept. A case in point is the F-18 which makes use of two AN/AYK-14 computers to retain mission reliability at an acceptable level.

Reliability testing of the Intel SBC-80/10 single-board computer has resulted in a mean time between failure of 91,739 hours with a 90% confidence factor. The

tests were conducted under accelerated life conditions on a commercial (non-militarized) version of the board. Reference 2 derated this figure to 25,000 hours MTBF to allow for a 55°C operating temperature. This results in a 10 year life expectancy when the equipment is continuously operated at 25°C.

D. SOFTWARE COST FACTORS

The software of a system follows a procurement process which in many ways is similar to that for the hardware. That is, there are both acquisition and maintenance costs to be considered. As with the hardware costs, software costs are closely related to the production base of the hardware. The costs of software procurement and maintenance are also a function of the breadth of use of the software development tools. In this section we discuss some of the more important aspects of software acquisition and maintenance as related to the choice of computer hardware architectures.

1. Software Acquisition

The effort and expense invested in the software acquisition phase of a computer system are in many respects similar to the acquisition of hardware. The importance of proper software planning has been typically underestimated by hardware oriented systems planners. This is in part due to the relative newness of programmed logic concepts. As discussed in Chapter II, the intelligence of a "smart" circuit exists within the program memory. The ratio of software to hardware cost is growing constantly as microcircuit devices become more common. While hardware costs drop due to ever-improving production techniques, the

cost of producing software continues to be a human intensive activity. Thus, equivalent cost reductions have not been attainable in this area. Much of the expense of software development is comprised of build-up costs of the program development tools such as assemblers, compilers, cross-compilers, emulators, and documentation of such tools. Another major cost factor is in the education of programmers, operators and maintenance personnel in the use of various languages and other development tools.

By adapting to industrial standards of software development, a savings similar to that of utilizing industry standard hardware could be realized. At present, the Navy is trying to enforce the use of CMS-2 as a standard high-level programming language. Consequently, the Navy is the only agency using the language. Education, documentation, and development costs must all be borne by the Navy at great expense. Many microprocessor-based computers have experienced broad usage to an extent that many common high-level languages have been independently adapted by the private sector for their use. Industry has recognized the need for high-level programability of the microcomputer devices in increasing the productivity of the human element in software development. Due to the large number of users, the cost of development of the programming tools is widely distributed. As a consequence, a very capable FORTRAN IV, COBOL, PASCAL, PL/M, APL or BASIC compiler can be typically purchased for less that $2,000. This makes the investment of several million dollars in development of the CMS-2 language standard seem rather wasteful.

Development tools are not the only factor affecting acquisition costs however, the importance of shifting to an industry standard language becomes more pronounced when we consider the impact of rapidly changing hardware.

A major motive behind enforcing a high-level language is to promote software transferability from machine to machine. The idea is that as hardware changes, we should be able to retain some of the programming effort already invested. It must be recognized, however, that only a portion of any program can be practically transferred to a computer of radically different architecture. This assumes, of course, that programs themselves are modularized such that some routines are hardware independent. An example would be high-level mathematical functions such as floating-point routines or various transcendental functions such as sine, cosine or logarithms. Entire operational functions such as ballistic calculation routines could possibly be transferred from one generation of machine to another, assuming input parameters did not vary during the transition. Other routines become hardware dependent and would not be usable in most cases unless a high degree of standardization existed in communication protocol. This is an unlikely happenstance.

Of course, these concepts are equally applicable to the centralized computer alternative, but, it is the author's opinion that the use of industrial standard software development languages would prove more beneficial if industial standard hardware were also adopted.

2. Software Maintenance

It is somewhat difficult to divorce the maintenance aspects of software from the acquisition process. Much of the argument presented in favor of the modular computer alternative in the previous section holds true for maintenance. Very little software maintenance is typically conducted at the depot, intermediate or organizational level. Program upkeep would be, as in the case of the A-7E

and other tactical aircraft, effected at one of the Naval laboratories, such as Naval Weapons Center, China Lake, California. Program modifications are distributed, after extensive evaluation, to the tactical squadrons on a fleet-wide basis. Organizational maintenance personnel incorporate the changes under supervision of a revision team from the cognizant support facility. This is accomplished, in the case of core memory systems by "reading in" the new program from magnetic or paper tape. In the case of single-board computer systems, this function would involve "burning" new program memory PROMs and installing them in place of the outdated PROMs which could then be recycled. One effect of the modular concept is that program modifications can be implemented in one module without affecting the integrity of the other modules in the system. Certain program alterations may require the field replacement of only one or two PROMs on a single-board computer. Management of field changes to the software would be more difficult in this case since extra care must be exercised to prevent mixing old PROM versions with the new ones.

V. CONCLUSION

The foregoing discussion presents an argument in favor of moving away from high-cost centralized airborne tactical computers toward incorporating a moderate cost modularized concurrent processing architecture using low-cost industrial standard components or modules. References 2 and 4 indicate that such a move is economically sound. In the problem statement of Chapter IV, we implied that the present avionics procurement program, in pursuing the centralized computer alternative, does not adequately provide for a technological growth of the magnitude currently being experienced in the semiconductor industry. The consensus of opinion within the Navy places a large importance on the sunk costs invested in the CMS-2 language and tactical computer standardization program. Reference 2 recommends a departure from the present acquisition plan. It is the author's opinion that an opportunity exists for the government to benefit from current industrial activities in this area, rather than the traditional situation in which industry benefits from government sponsored research and development programs. It is believed that military tactical avionics programs can remain state-of-the-art while capitalizing on the natural industrial collective tendency to minimize internal costs while advancing the technological base through competition for the consumer market. As new devices become available, the industry will inevitably devise the appropriate software development tools, documentation and training programs, usually on a timely basis. There seem to be few economical arguments against the government becoming simply another consumer where computing equipment is concerned. It has already been

pointed out that an inexpensive microcomputer can perform the same function as its expensive minicomputer counterpart.

The idea that military avionics should be allowed to lag behind the industry may, at first, seem inconsistant with current concepts. Considering the rate of change in the microelectronics technology, and the fact that government sponsored research and development projects are usually no longer state-of-the-art by the time of full-scale production, this may not be such an unreasonable proposal. It is believed that the government represents a large enough portion of the consumer market to effectively influence the direction which LSI manufacturers will pursue in developing new and, hopefully, compatible devices.

In summary, then, the following recommendations are presented as methods of taking advantage of current and projected trends in microelectronic developments:

1. The Navy or joint services should form a project office to further evaluate the consequences and possible benefits of incorporating a concurrent processing computer system utilizing industry standard components in the next generation of tactical aircraft.

2. The Navy should strongly consider phasing out the CMS-2 language in favor of languages more widely supported by the computer manufacturers: PASCAL, BASIC, FORTRAN.

3. As the computer and microelectronic circuit manufacturers develop new techniques and devices, Navy avionics program managers should evaluate the advancements for possible incorporation into tactical aircraft under design. Avionics acquisition programs should maintain a consumer posture when considering computer systems.

4. Programming should be in high-level languages, when

possible, to permit carry over of programs to the next generation of computers.

5. Programs should be modularized into hardware independent and hardware dependent partitions where possible to facilitate software transferability.

6. Standardized communications protocols should be developed for both serial and parallel bus structures so that succeeding generations of computer modules would tend to evolve along set guidelines. This would have a stabilizing effect on both hardware and software compatability, thus, increasing system life and reducing transition costs.

BIBLIOGRAPHY

1. "Microelectronics," Scientific American, special issue on microelectronics, v. 237, no. 3, September 1977.

2. Kodres, U.R., Buttinger, J. D., Hamming, R. W., and Jones, C. R., A Study of Alternatives for VSTOL Computer Systems, Naval Postgraduate School, December, 1977.

3. Cooper, J. D., "Computers in Tactical Systems," Computers in the Navy, J. Prokop, Ed., Naval Institue Press, Annapolis, MD, 1976, pp. 122-123.

4. Genovese, D. H., "An economic Analysis of Life Cycle Military Manpower Maintenance and Training Requirements in Avionics Minicomputer and Microcomputer Systems," Master's Thesis, Naval Postgraduate School, March 1978.

5. Capece, R. P., "Programming Struggle Goes On," Electronics, pp. 92-94, 11 May 1978

6. "Military's Eye Set on Commercial Microprocessors for its Programs," Electronics, pp. 44-46, 2 March 1978.

7. Moss, D,. "Multiprocessing Adds Muscle to uPs," Electronic Design, pp. 238-244, 24 May 1978.

INITIAL DISTRIBUTION LIST

		No, Copies
1.	Defense Documentation Center Cameron Station Alexandria, VA 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93940	2
3.	Department Chairman, Code 54 Department of Administrative Sciences Naval Postgraduate School Monterey, CA 93940	1
4.	Professor U. R. Kodres, Code 52Kr Department of Computer Science Naval Postgraduate School Monterey, CA 93940	1
5.	CDR A. C. Crosby, Code 54Cw Department of Administrative Sciences Naval Postgraduate School Monterey, CA 93940	1
6.	LCDR Cleveland D. Englehardt, USN 27 Greenbrae Court El Sobrante, CA 94803	1