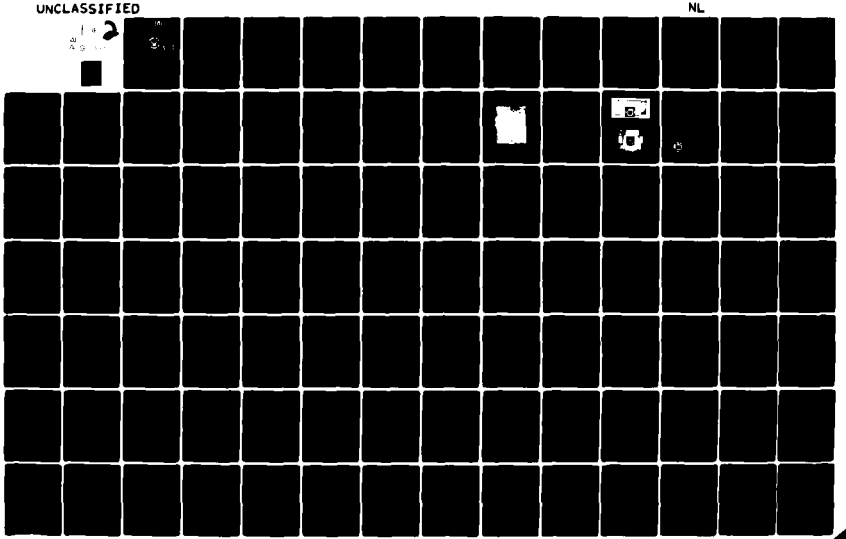
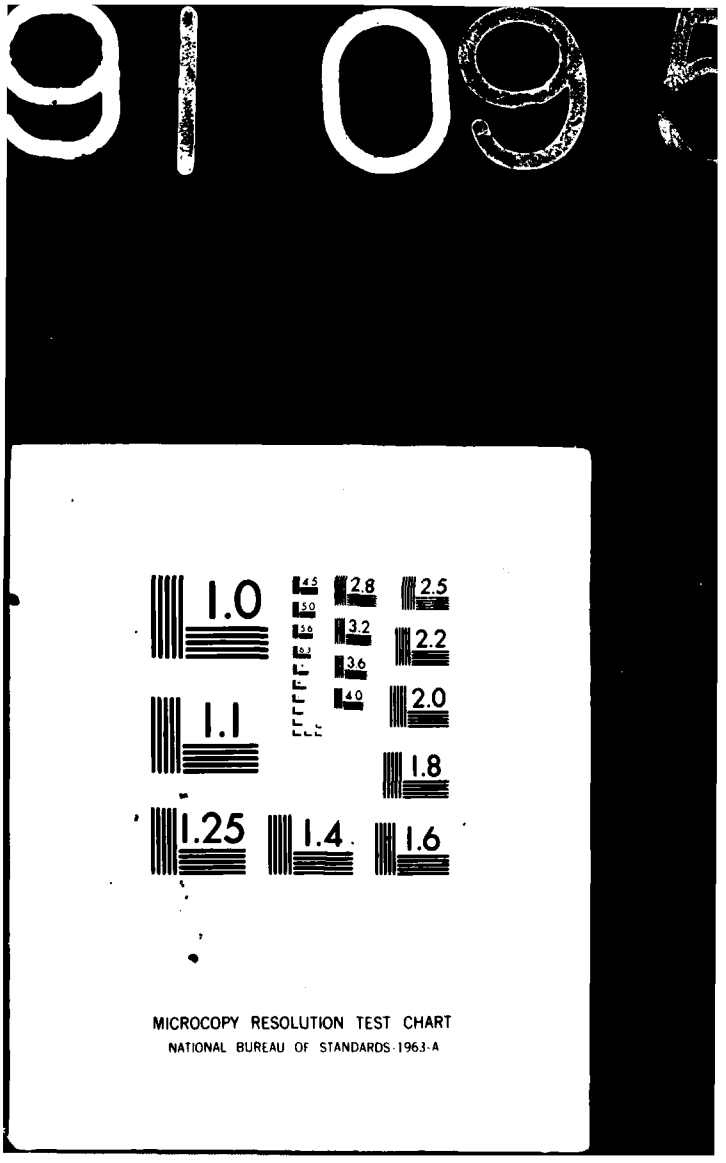
AD-A091 095 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 9/2
A METHOD FOR EVALUATION OF MICROCOMPUTERS FOR TACTICAL APPLICAT--ETC(U)
JUN 80 M MASTRAKAS

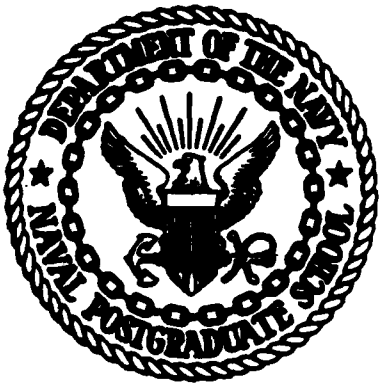
UNCLASSIFIED NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
7.1
8.0
9.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

(4) A METHOD FOR EVALUATION OF MICROCOMPUTERS FOR TACTICAL APPLICATIONS.

by

(10) Marcos Mastrakas

(12) 192

(11) June 1980

Thesis Advisor: U. R. Kodres

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A091095 / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Method for Evaluation of Microcomputers for Tactical Applications	5. TYPE OF REPORT & PERIOD COVERED Master's Thesis: June 1980
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Marcos Mastrakas	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940	12. REPORT DATE June 1980
	13. NUMBER OF PAGES 189
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, California 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)

Microprocessor
Microcomputer
Multi-microcomputer
Tactical application of microcomputer systems
Performance capacity of computers

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

A method is proposed to evaluate the performance of microcomputer systems in a specified tactical application. The computational requirements of a tactical application are specified in terms of performance parameters. The presently marketed microcomputer and multi-microcomputer systems computational performance capacities are also specified in terms of performance parameters. If the performance capacity is not less than the

DD FORM 1 JAN 73 1473 (Page 1)

EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

performance requirement, then the microcomputer system is evaluated as a feasible system suitable for implementing the tactical application. A case study using the attack aircraft A6-E tactical system illustrates the evaluation method.

Approved for public release; distribution unlimited

A Method for Evaluation of Microcomputers for Tactical Applications

by

Marcos Mastrakas
Lieutenant Commander, Hellenic Navy
B.S., Hellenic Naval Academy, 1962

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1980

Author _____

Approved by: _____
Thesis Advisor

_____
Second Reader

_____
Chairman, Department of Computer Science

_____
Dean of Information and Policy Sciences

ABSTRACT

A method is proposed to evaluate the performance of microcomputer systems in a specified tactical application. The computational requirements of a tactical application are specified in terms of performance parameters. The presently marketed microcomputer and multi-microcomputer systems computational performance capacities are also specified in terms of performance parameters. If the performance capacity is not less than the performance requirement, then the microcomputer system is evaluated as a feasible system suitable for implementing the tactical application. A case study using the attack aircraft A6-E tactical system illustrates the evaluation method.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

"In its thirty years of experience with computers, the Navy has discovered--and often rediscovered--one fundamental truth: that people are the vital key to success in the computer age!"

Rear Admiral Frank S. Haak, USN
"Brainware versus Hardware" [1976]

ACKNOWLEDGEMENT

I would like to thank Professor Uno Kodres, who led me in the mysteries of microprocessors/microcomputers. He helped more than his share on this thesis, and is more than an advisor to me.

I. INTRODUCTION

A. GENERAL

The microcomputer technology, which came into existence in 1971, has grown from a low performance four binary digit word length device (Intel 4004) to a high performance sixteen binary digit word length computer, the Motorola MC 68000, in 1979.

The central processing unit (CPU) of the MC 68000 is capable of addressing 16 Megabytes of memory, which is twice the memory capacity of the CRAY-1, the world's fastest and largest computer.

A thirty-two bit word length microcomputer system with sophisticated architecture is soon to be announced. Clearly, the advancements in Large Scale Integrated (LSI) and Very Large Scale Integrated (VLSI) technologies are coming so rapidly that just to keep up with the newest information is becoming a challenge.

The Armed Forces, and the U.S. Navy in particular, have long been pioneers in the development of new systems. The development of large computer systems normally lags behind the state of the art because of the length of development time. Microcomputer systems are virtually nonexistent in tactical applications. This may be attributed to a lack of knowledge of their capabilities and to the rapid advancement of microcomputer technology, coupled with the high risk of trying something new.

The intent of this thesis is to give the user the necessary information and background in the area of microprocessors and microcomputer systems and also to provide a method to evaluate microcomputer systems for tactical applications, i.e., Command Control Communications (C^3), weapon systems, etc. At the moment the only known systematic method for the evaluation of computers for military use is the Military Computer Family (MCF) report in 1976 [3].

B. MAJOR ISSUES

There are several major issues.

1. Processing Needs

Given an application described by parameters, such as:

(1) Number/per sec of loads, stores, compares.

(2) Number/per sec of additions, subtractions, etc.

(3) Number/per sec of floating point operations.

(4) Number/per sec of multiplies, divides.

(5) Number of instructions/bytes in the program and data.

2. Processing Capacity of LSI/VLSI Systems

Stated in terms of parameters:

a. Uniprocessor

(1) Number/per sec of stores, loads, compares.

(2) Number/per sec of additions, subtractions, etc.

(3) Number/per sec of multiplies, divides.

(4) Number/per sec of floating point operations.

(5) Memory sizes: main + auxiliary memory.

b. Multiprocessor

Multi-single board computer on a Multibus [Intel 77].

3. Program Development Capability

a. Program Development Tools

(1) Hosted on large systems. The High Order Language (HOL) Ada used as a program development tool for real-time processing including microprocessor applications.

(2) Hosted on small development systems.

4. Selection Procedure

Selection of one or more processor families which will provide long term support for a large class of Navy applications.

C. STANDARDIZATION

Standards for microprocessors are becoming increasingly important for military and industrial market expansion. However, the standards must be objectively chosen with applications engineers and user communities involved in the decisions and must be "technology transparent" to provide continuity and support for microcomputer development without stifling innovation.

The idea of a standard military microprocessor architecture has the advantages of volume purchasing power and logistic support cost savings; the disadvantage appears to be a risk of not following the commercial mainstream.

The Defense Department's Ada HOL is being developed to serve programming needs for real-time processing including

microprocessor applications, and is an example of one potential microprocessor HOL standard [26].

D. APPLICATION TRENDS AND OPPORTUNITIES

The opportunities have never been greater for microprocessors--in computation, control, logic replacement and special functions for industrial processing and defense systems.

However, a variety of barriers discourages use of microprocessors in many of the lower-volume application areas [26].

The three key barriers to rapid introduction of microprocessors into military, in particular, the Navy systems relate to:

1. Concerns for life-cycle supportability (for both military and industrial applications where anticipated system life is 15 years or longer).

2. The belief that the microcomputer performance in military applications is not adequate.

3. The microcomputer technology is not compatible with previously defined architectural standards. Because of large prior investments in software tools and applications software geared to the established hardware standards, there is reluctance to undertake costly changes.

In terms of user requirements, DOD and industry appear to have more in common than is generally perceived. Some military systems do require radiation hardened devices. Fortunately, the two primary approaches to radiation hardening--I^2L and,

to a lesser extent, silicon on sapphire--may also provide performance and economic advantages. However, a number of military applications, such as radar and electronic counter-measures, require very-high-speed signal processing in contrast to the slower, general-purpose computation more commonly found in commercial applications.

E. COMPOSITION OF THESIS

Section II presents a survey on the characteristics of microprocessors/microcomputers, magnetic bubble memories, Large Scale Integrated (LSI) and Very Large Scale Integrated (VLSI) technologies, information about Department of Defense (DOD) project Very High Speed Integrated Circuit (VHSIC), and some characteristics of different types of computers.

Section III presents a discussion of the use of computer systems in tactical applications (sensor control, C^3, weapons, etc.), and the possibilities of using microcomputers instead of the standard computer systems, i.e., AN/UYK-7,14,20, IBM-370/168, etc.

Section IV describes a method to evaluate the performance of microcomputer systems in a specified tactical application. It also gives to the evaluator the necessary tools in terms of military requirements, software materials, and how to compute the life cycle cost of a given system.

Section V presents the case study of the attack aircraft A6-E tactical system, in order to illustrate the evaluation method.

Section VI makes a summary and presents certain conclusions of the thesis.

Appendices A through D contain important information to aid the evaluator's work.

Tables I through XVIII contain detailed information about μP/μCs, the available OS for μCs, etc.

Finally, a glossary explains the terms and definitions about μP/μC which are used in the text of this thesis and elsewhere.

It is underlined here that this thesis is based solely on an unclassified bibliography.

II. BACKGROUND

During the past 25 years the "computer revolution," as in the past the "industrial revolution," has dramatically changed our world and it promises to bring about even greater changes in the near future.

In the middle 1960's the advent of "microcomputers" has both accelerated and expanded the impact of the "computer revolution" [4].

The microprocessor (μP) is one of the latest developments in computer technology. This device has all the functions of the Central Processing Unit (CPU) of a computer on a tiny piece of silicon. Such a device can fetch instructions from a memory, decode and execute them, perform operations (arithmetic and logical), accept data from input devices, and send results to output devices. A microprocessor, together with a memory and input/output devices, forms the "microcomputer" (figures 1, 2).

The microcomputer (μC) represents a very remarkable achievement of engineering ingenuity and industrial know-how at their best. The cost of simple μP chip is a few dollars, and that of a complete μC, having a power similar to mini and maxi computers, is a few thousand dollars.

New technologies, different architectures, and faster memories are having an impact on the computer. From a laboratory curiosity in 1978, the "Magnetic Bubble Memory" (MBM)

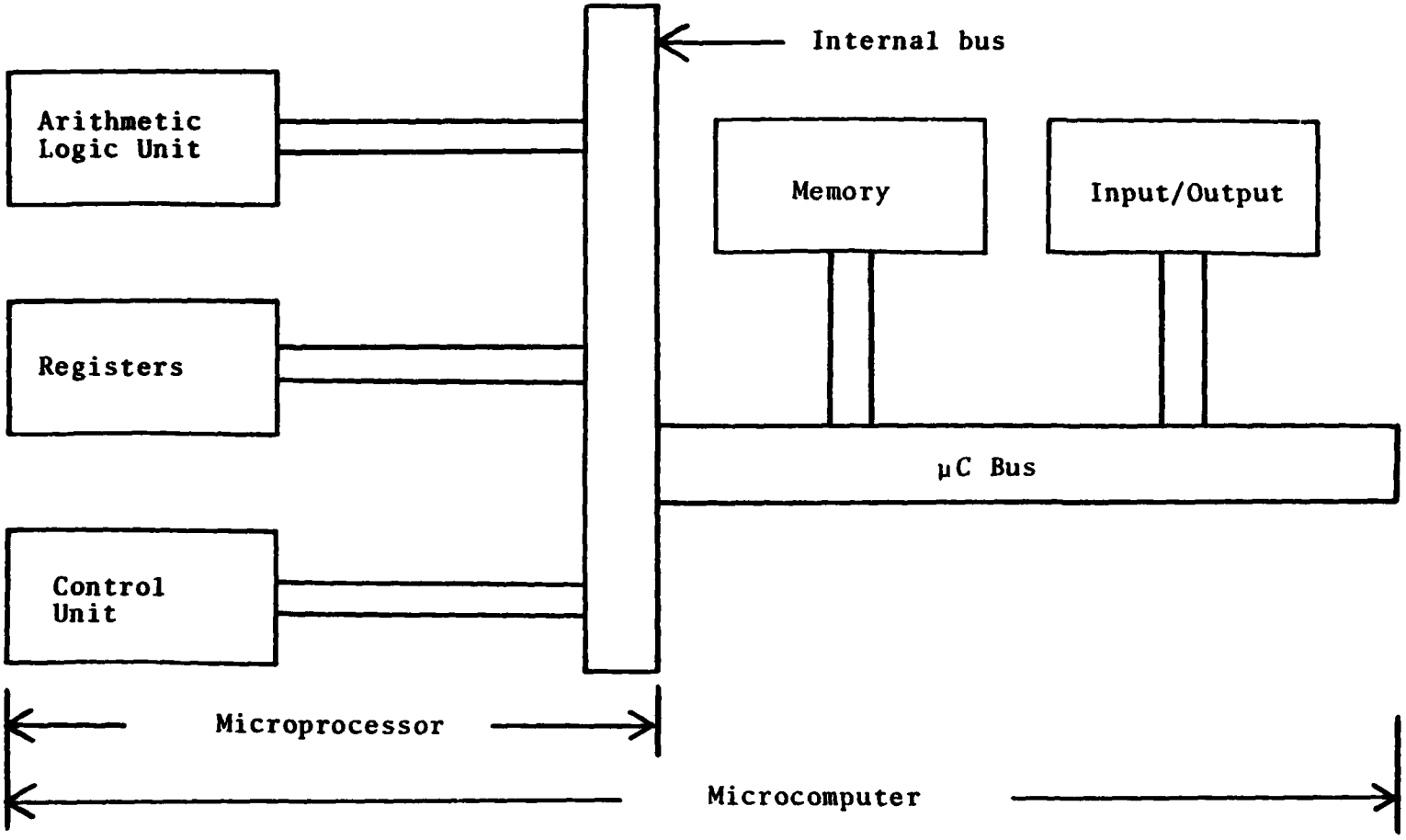


Figure 1. Block diagram of a microprocessor/microcomputer.

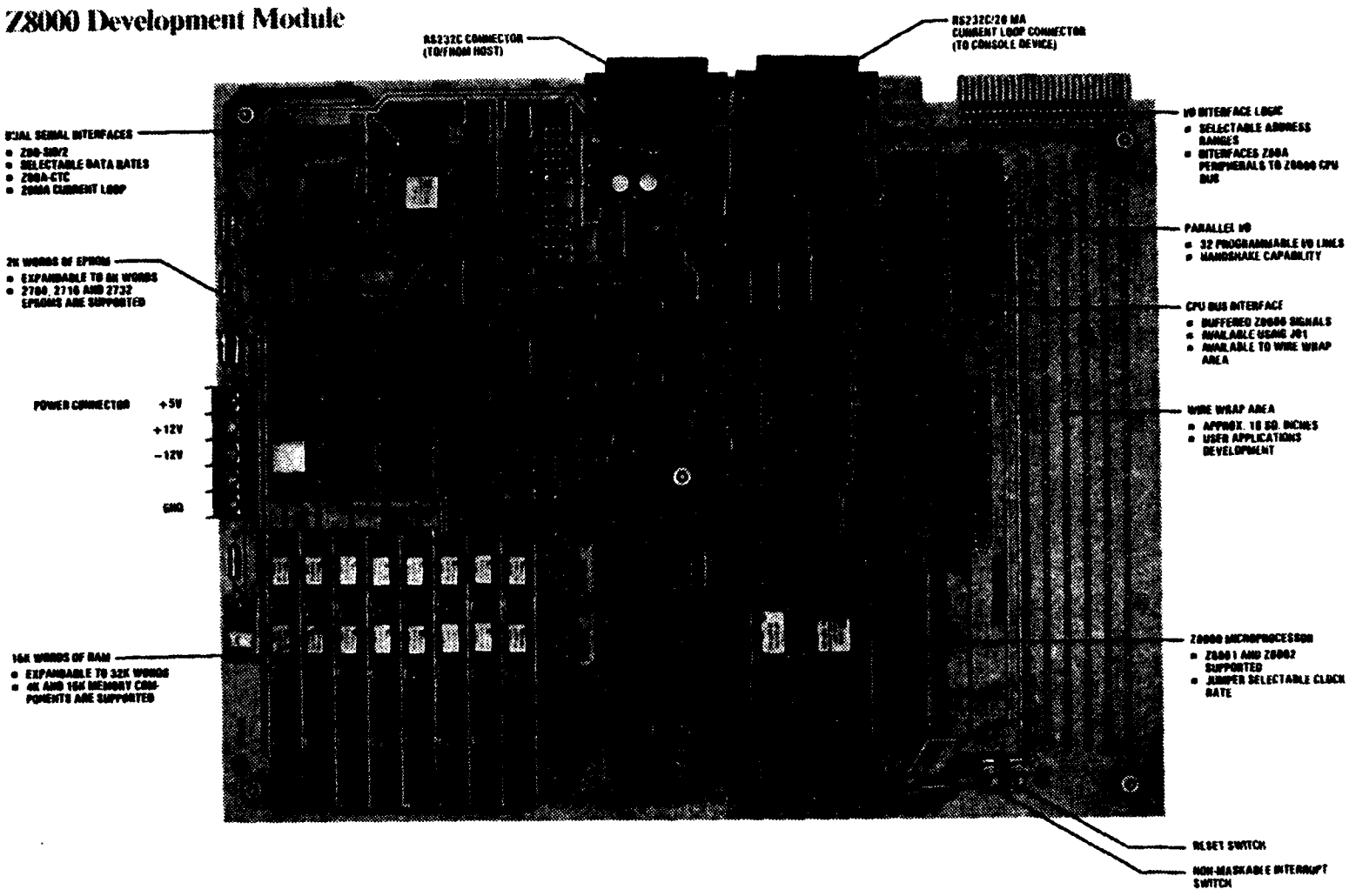


Figure 2. A single board Z8000 microcomputer.

chip (Fig. 3) has evolved rapidly into a viable commercial device. Likewise, the range of available components has grown faster than anyone expected! New techniques will produce devices using the above chips, with capacity from 1 M bits in 1979 to 4 M bits in 1980 and 16 M bits in 1985-87 (i.e., chip T I B 1000) [9].

Magnetic bubble memories are best suited for use in bulk and auxiliary storage systems. Initial applications have been in microcomputer storage. MBM's are becoming cost competitive with small floppy disks. MBM's have high packing density and can be mounted on the same PC board as the CPU.

Aerospace and military applications will use a significant number of MBM devices, which replace mechanical devices (primarily airborne head-per-track disk units). Due to reliability (i.e., with a 10^{-10} probability of error, mean time between errors is 55 hours, but when corrected can be over 500,000 years!), and nonvolatility (i.e., information is maintained even with power loss), bubble memories are ideal for replacing satellite tape recorders (10 Khr MTBF's) [14]. Due to ruggedness, bubble memories are suitable for tanks, backpacks and other severe environmental/mechanical-stress conditions. MBM's can replace special ruggedized cassettes now used for data acquisition and program loading.

The following figures 4 and 5 give characteristics, advantages and disadvantages of MBM's which presently compare to other memories on the memory hierarchy [14].

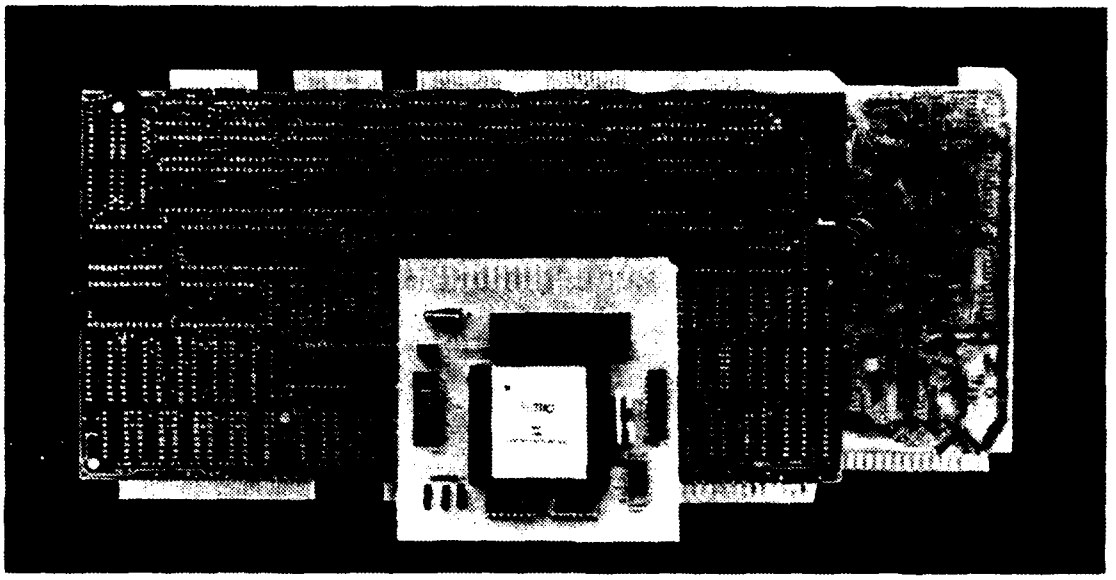


Fig 3a Space saver. Availability of INTEL MBM and supporting LSI family of control and driver circuits reduce space, component count for 1Mbit bubble system by order of magnitude, replacing two printed-circuit boards, four 256 K modules, and about 85 ICs.

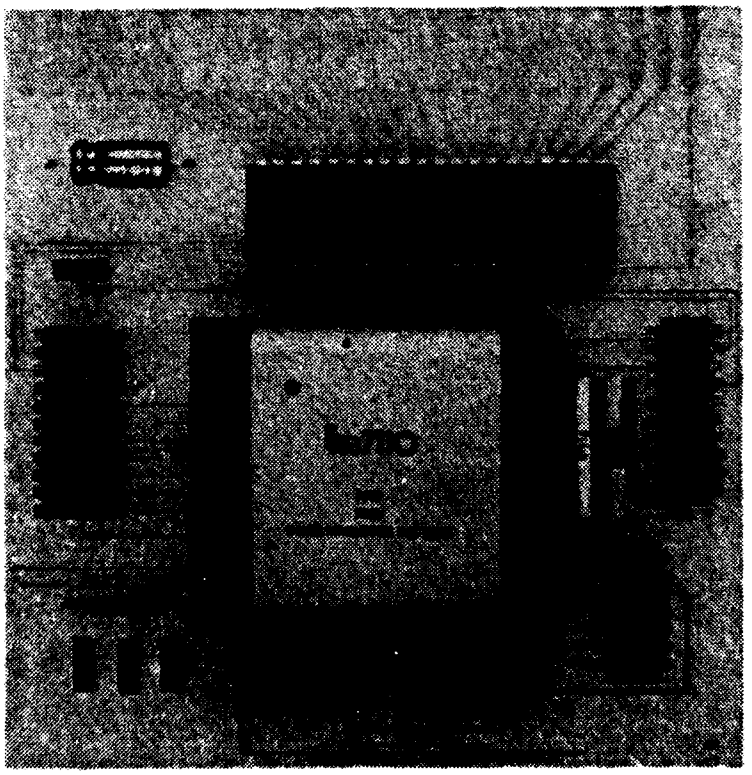


Fig 3b Complete One Megabit Memory System (shown actual size of 16 square inches) consists of 7110 bubble memory device and support electronics.

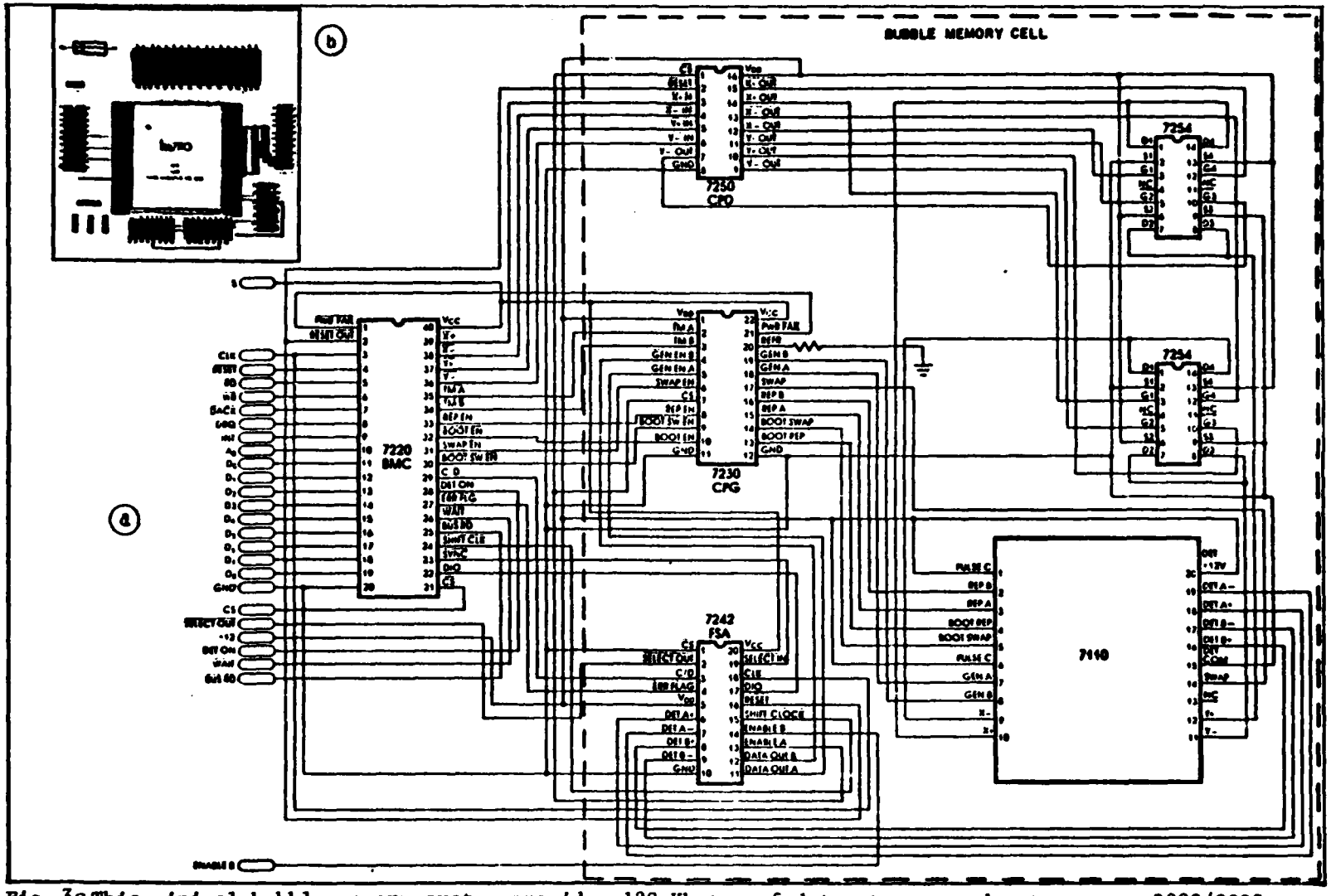


Fig. 3c This minimal bubble memory system provides 128 Kbytes of data storage and mates to an 8080/8085-compatible bus. Just six support packages are needed to make the subsystem operational(a). On a 4.5-sq-in. board, the entire 128-Kbyte memory system can be transported and plugged into another system without any data loss(b).

Memory Characteristics	*Magnetic Bubble Memory*	*Electron Beam Addressable Memory*	*Charge Coupled Device Memory*
Storage Representation	Magnetic domain	Electrical charge	Electrical charge
Bit Density	2×10^6 Bits/in.^{2}	10^7 Bits/in.^{2}	10^6Bits/in.^{2}
Access Mode	Sequential	Quasi-random	Sequential or block addressable
Access Time	0.5–10 msec	10–20 μsec	Sequential: 5–50 μsec Block access: 5 μsec
Transfer Rate	100–500 Kbps	4–8 Mbps per tube	1–10 Mbps.
Power Per Bit (During Memory Operation)	0.4 μW	10 μW	50 μW
Cost Per Bit (1978)	0.2¢	0.005¢	0.05¢
Largest Chip Size Fabricated (1978)	250 Kbit (Texas Instruments, Rockwell International)	120 Mbits per tube 32 (CDC)	64 Kbit (Mnemonics)
Operational Temperature Range	0–50°C	0–50°C	−10° to 80°C
Susceptibility to Electromagnetic Emanations	Can be shielded	Can be shielded	Can be shielded
Susceptibility to Power Surges	Power must be filtered	Power must be filtered	Power must be filtered

Fig. 4 Characteristics of different kind of memories. (From: "Distributed Micro/ Minicomputer Systems," Cay Weitzman, 1980).

ADVANTAGES	DISADVANTAGES
MBMs vs. Floppy Disk	
1) Higher reliability	1) Storage media not readily changed
2) Non-mechanical	
3) Smaller physical volume	
4) Faster access	
5) Simpler interface	
6) Media integrity	
MBMs vs. RAM	
1) Non-volatile	1) Slower access
2) More bits/device	2) Slower transfer rate
3) Reduced board space	
MBMs vs. ROM or PROM	
1) Programmability	1) Slower access
2) More bits/device	2) Slower transfer rate
3) Reduced board space	

Figure 5. Comparison between MBMs and other memories.

Technology has produced, and is continuing to produce, LSI and VLSI (large and very large scale integrated) components of increasing complexity and power. Yet it is difficult to identify the best or even the many ways in which this powerful new tool, VLSI, might be exploited [25].

The design of electronic equipment will change rapidly once VLSI, in the coming years, becomes a key component in the industry [24].

The Pentagon's Very-High-Speed Integrated Circuits (VHSIC) project was initiated by DOD to develop VLSI signal processors with several hundred times higher speed and computing power than today's LSI devices [29].

The goal of the project is pilot production in 1986 of processors containing 250,000 gates, operating at clock speeds of at least 25 MHZ, and performing several million to several billion operations per second. VLSI circuits are also needed to reduce the power consumption, weight, and size of military electronics. These reductions will, in turn, lead to lower life-cycle costs; lower needs for primary power, deck space, air-conditioning, and the like. The costs of primary power and weight in a satellite, for example, are at least $2000/W and $5000/Kg, respectively. About $20 million would be saved on power costs alone by a reduction of 1mW in the average operating power per circuit. The cost of integrated circuits, by contrast, is several hundred thousand dollars [29].

Military equipment that will use VHSIC are the following:

° SONAR - Acoustic signature analyzers used in the BQQ-5 and the BQQ-6 processing subsystem in strategic (Trident) and attack submarines; the MK-48 processor in homing torpedoes; and in the Proteus processor used in antisubmarine warfare aircraft.

° RADAR - Signal processors for radar systems in the E-2C and E-3A airborne early warning systems; in advanced fighters (F-14, F-15, F-18) and interdiction aircraft (A-6) for all weather bombing; and in stand-off target acquisition systems (SOTAS).

° MISSILE GUIDANCE SATELLITES - Processors of radar and infrared sensor data for inertial navigation (Global Position Satellite), and processors for target recognition, proximity fusing, and clutter rejection in air-to-air missiles (Phoenix, Sparrow, Sidewinder), surface-to-air missiles (Patriot, Hawk), and submarine-launched cruise missiles.

° COMMUNICATIONS - Spread spectrum and time dispersion modulators and demodulators, error correction coders and decoders for digital voice transmission (ANDVT) and battlefield communications (REMBASS, Seek Talk, SINGGARS).

° SIGNAL INTERCEPT ANALYSIS- Signal modulation analyzers and signal classifiers for scan receivers (ALR-66, ALR-67).

° ELECTRO-OPTICAL PROCESSORS - Processors of electro-optical data for more detail and for estimation of target trajectories, in such infrared surveillance systems as the Halo satellite.

The microcomputer has basically the same capabilities and limitations as any other computer. These are:

1. Speed: Extremely rapid rates.
2. Flexibility: May be programmed to solve many types of problems.
3. Repetitive Operation: Perform similar operations thousands of times, without becoming bored, tired, or careless.
4. Accuracy: As specified by the programmer.
5. Intuition: Has no intuition. A man may suddenly find the answer to a problem without working on details, but a computer can only proceed as ordered [4].

The microcomputer offers a number of advantages to the design and production engineer, namely: 1) smaller size and weight, 2) greater reliability and flexibility, 3) component standardization, 4) shorter design cycle time, and 5) lower cost.

Economies associated with the "computer-on-a-chip" have resulted in the availability of systems with the functionality and performance of larger computer systems at a cost which is up to two orders of magnitude smaller.

Multi-microcomputer systems are presently designed and will be in a large number of applications, such as control of electric power, nuclear power generating facilities, etc.

The reasons why these systems are useful in many applications are several. Figure 6 tabulates the advantages and disadvantages of multi-microcomputer systems [23].

ADVANTAGES	DISADVANTAGES
1. Increased reliability	1. Increased software complexity
2. Increased survivability	2. More dependence on communications technology
3. Increased distributed processing power	3. Unique expertise needed during design and development phase
4. Increased responsiveness	
5. Increased modularity	
6. System expandability in smaller increments	
7. Lower cost	

Figure 6. Advantages and disadvantages of multi-microcomputer systems.

The March 1979 Quarterly Report for the Chemical Fund, Inc. states that: "Many analysts foresee a proliferation of intelligent electronics so widespread over the next ten years that the 1980's may be known as the Microcomputer Decade."

It is estimated that the annual demand for microcomputers may reach 300 million units by 1984, and triple that number may be in use by the end of this decade.

The following table I and figure 7 present the characteristics of different computer systems and corresponding execution times.

TABLE I

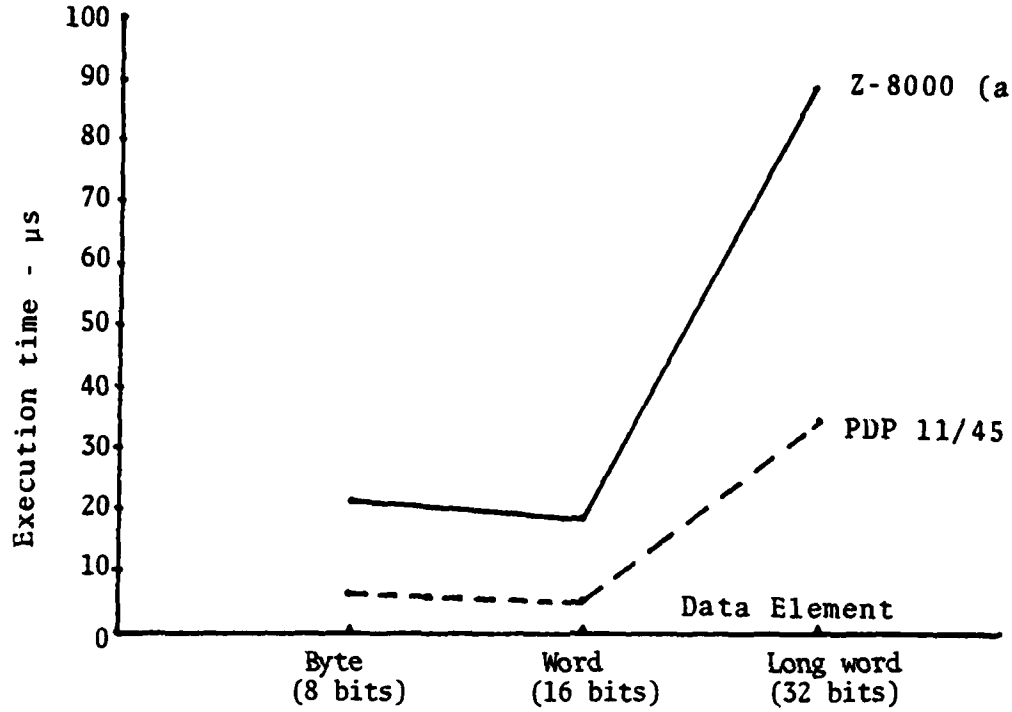
Different Types of Computers

General Characteristics	IBM 370/168	DEC PDP 11/45	UNIVAC AN/UYK-7	UNIVAC AN/UYK-20	ZILOG Z-8000*
Type	Maxi	Mini	Maxi	Mini	Micro
Construction Standard	Comm.	Comm.	Mil.	Mil.	Comm.
Word Length (in bits)	32	16	32	16/32	16
Processor Add Time (in μS)	0.13	0.90	1.50	0.75	0.75
Maximum I/O Data Rate (bytes/sec)	16M	9M	4M	65K	48M
Number of Internal Registers	64	16	8(16)	16(32)	16
Maximum Physical Dimensions (in inches)	Huge	71x30x22	41x24x20	24x18x29	14x2x11
Maximum Weight (in lbs)	out of range	300	527 to 1,139	185	2
Multi-User Op. System	Yes	Yes	Yes	No	Yes
Real-Time O.S.	Yes	Yes	Yes	Yes	Yes
Security	Yes	Yes	Yes	No	No
Memory Capacity (in words)	8.4M	248 K	256 K	65 K	8 M
Software	All Types	Wide Variety	Fortran, CMS-2 Assembly	CMS-2 Assembly	Basic, Cobol, Fortran, Assembly
Approximate Cost (in $)	4.5M	50 K	250 K	80 K	1300

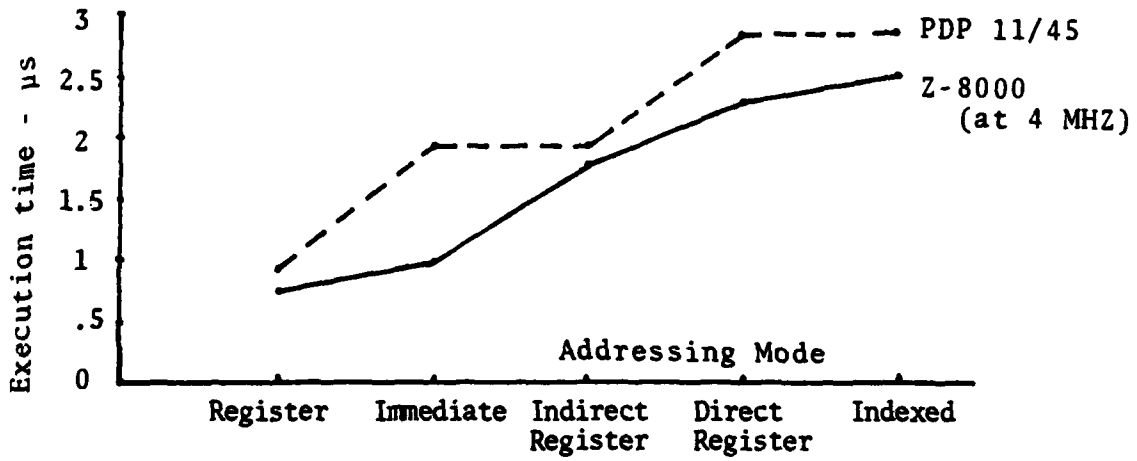
K = Thousands, M = Millions, Comm. = Commercial, Mil. = Military

*This information is for a single-board computer (SBC) and not for a complete microcomputer system.

Figure 7 Comparison of execution times between "MINI" PDP 11/45 and "MICRO" Z-8000.



a. Execution times for MULT R, DA



b. Execution times for LDB R, src for various addressing modes.

III. THE USE AND PERFORMANCE REQUIREMENTS FOR COMPUTER SYSTEMS IN TACTICAL APPLICATIONS

A. GENERAL VIEW

The use of computers in tactical applications in the Navy really got off the ground around 1960, with the fleet ballistic missile submarines (SSBN) and the Naval Tactical Data System (NTDS).

The submarine's missile and navigation systems contained several computers, and the NTDS was a complete automation of the ship's Combat Information Center (CIC). The Airborne Tactical Data System (ATDS), an airborne extension of NTDS, followed soon after [10].

Some of the most well known computer systems which are in use today for tactical applications are MAXI and MINI, as IBM 370/168, AN/UYK-7, AN-UYK-14, AN/UYK-20, etc. Although the microcomputer systems are not used for tactical applications, the use of microprocessors as integral components of computer's peripherals, in order to make them more intelligent (i.e., RD-358 magnetic tape, display consoles, etc.) is virtually impossible to enumerate.

The complete spectrum of current tactical applications will be surveyed starting with those of sensor control and ending with those of guided missile control.

B. SENSOR CONTROL

Radar, sonar and electronic warfare devices are the prime examples. These analog devices use computers for converting incoming signals from analog to digital form, and for automatic detection and classification of targets, track correlation and automatic tracking. Without a computer, a typical real-time electronic warfare system depends on the operator's ability to spot significant signals, to analyze and pass the information to CIC. Using the Automatic Detection and Tracking (ADT) radar system, it is possible to track simultaneously hundreds of targets, whereas a human operator is only able to handle up to four targets.

1. Single-Function, Stand-Alone Systems

Most familiar and widely used systems are the navigations systems, such as Satellite, Inertial, OMEGA, LORAN, TACAN, and autopilots. The F-14 uses a computer to control its variable wing angle and the A-7 has an automated in-flight engine monitor system.

C. COMMAND AND DISPLAY

In the area of operations, the most well known computerized systems are the Command and Control (or real-time combat direction) systems. Such systems are the NTDS, ATDS, Marine Tactical Data System (MTDS) and the AEGIS Combat System, detailed in figures 8, 9.

These systems consist basically of the following major components:

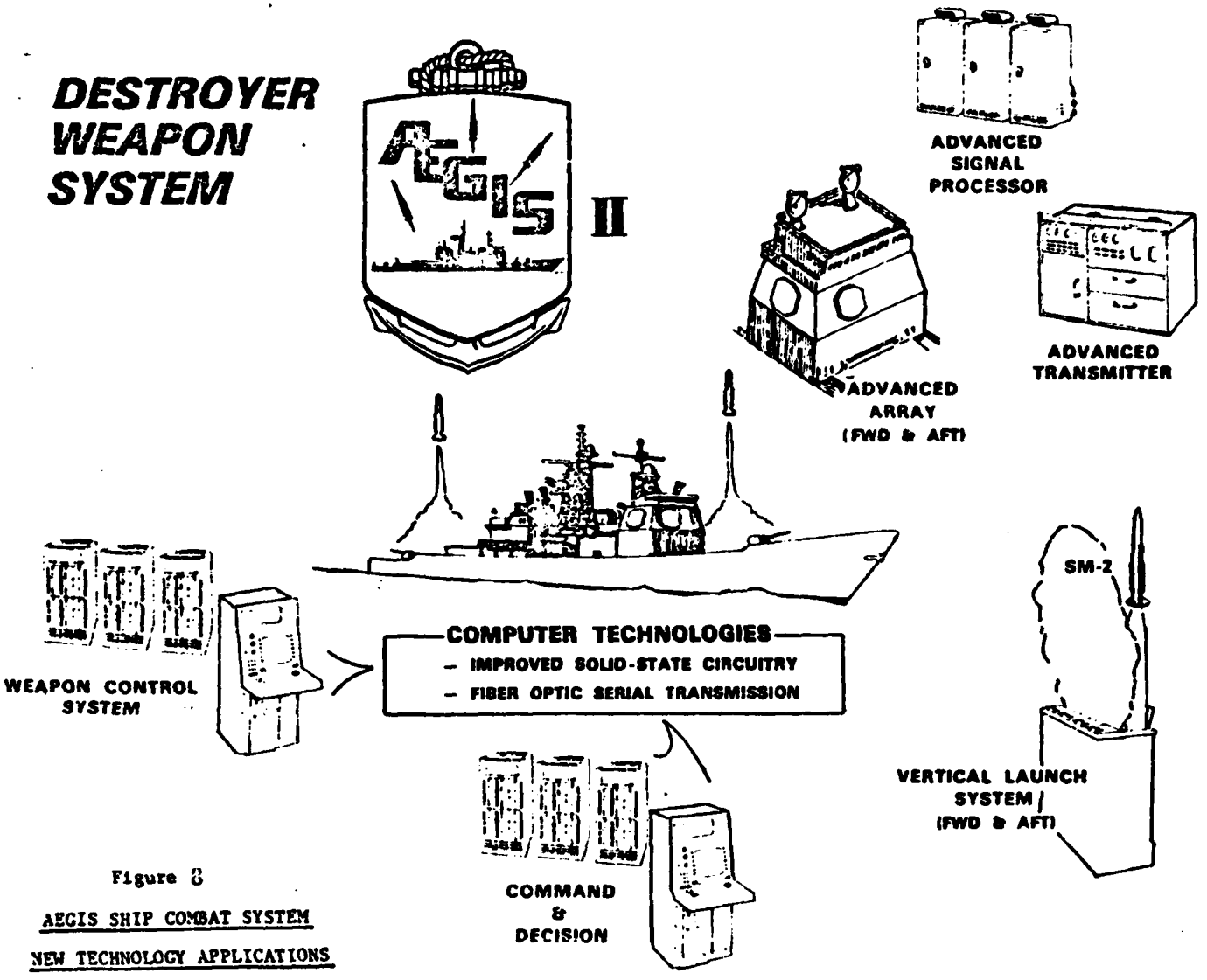


Figure 8

AEGIS SHIP COMBAT SYSTEM

NEW TECHNOLOGY APPLICATIONS

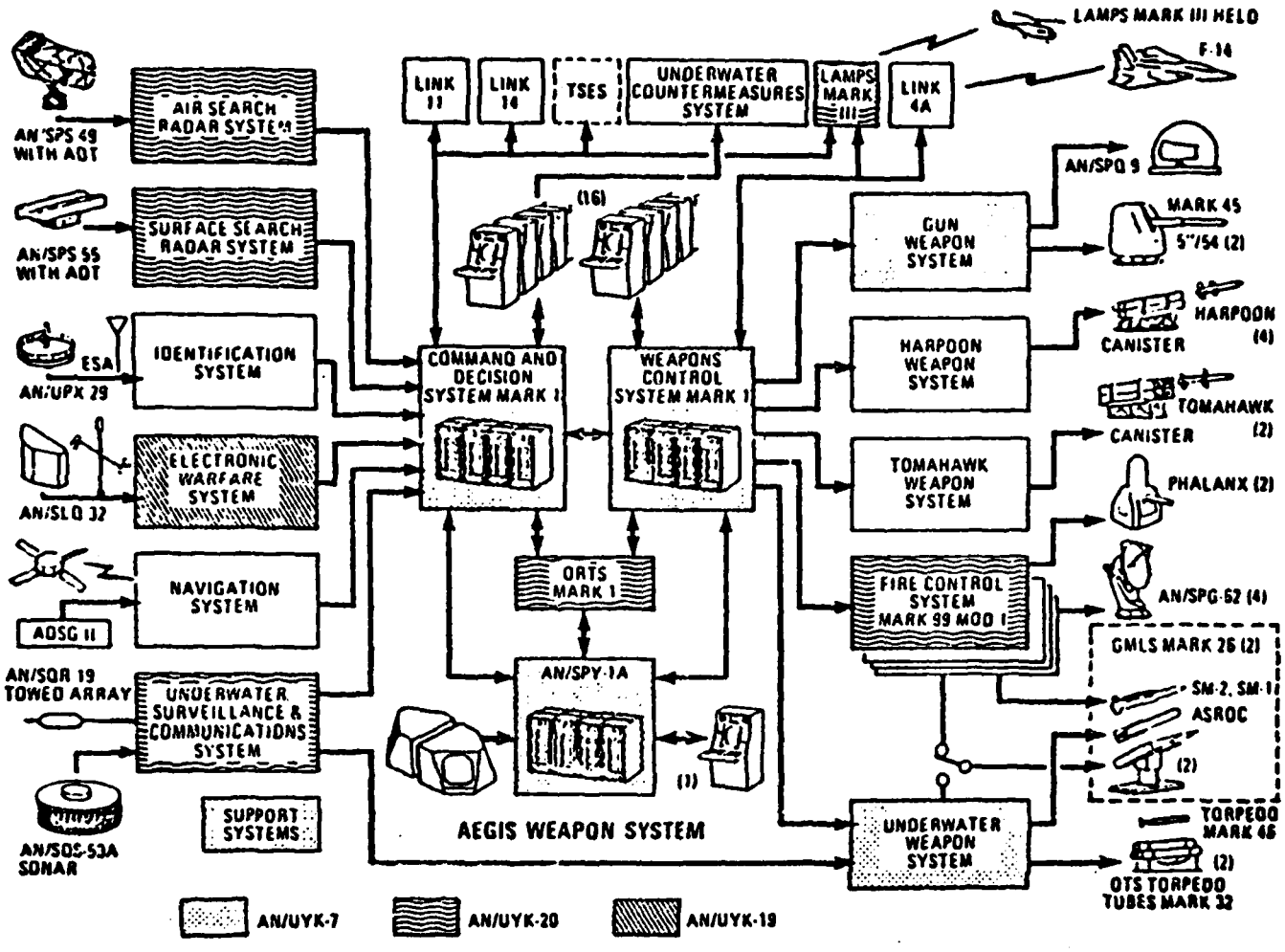


Figure 9

AEGIS SHIP COMBAT SYSTEM BASELINE

1. Analog to Digital Converters

These are used in order to convert the information from the sensors (radar, sonar, EW/ESM, etc.) into digital form and enter directly into the digital computer.

2. Computing Equipment

The heart of these systems is a central computer system(s) complex with various man/machine interface devices, in order to perform necessary calculations, in accordance with stored programs, and provide a picture of the current situation to the operator for further exploitation.

3. Communications Equipment

The crucial need of exchanging information rapidly between the elements of a unit or between different forces, requires the use of high-speed data-links. Via these data-links, the computers can communicate very fast (by having the appropriate data) in order to provide the necessary information for the operations.

4. Communications

Digital computers are excellent communication tools. They can interface directly with dedicated equipment; they can detect and correct errors in the messages; they can be used as controllers and multiplexers for communicating asynchronously through several devices simultaneously. Some examples are:

a) Ship-to-shore communication

b) Ship-to-ship communication

c) Inter-computer data links

d) Automatic and semi-automatic aircraft control links

e) Missile and drone guidance

f) Telemetry

g) Broadcasting tactical information to teletypes.

5. Visual Displays

The human interface for the computed information is the visual console display, where the operator finally has to make his own decisions.

6. Test Equipment

It may sound unusual to use the words "test equipment" in a paper about tactical applications, but it is absolutely necessary to examine this type of equipment because without it, it is not feasible to verify the operational readiness of the equipment used in tactical applications.

The automatic test equipment systems can be made to test anything, from tanks to turbines and to total aircraft systems. A notable example is the "Versatile Avionics Shop Test" (VAST) which determines the status of the system of an aircraft and what maintenance is needed when a plane returns to the aircraft-carrier. The microcomputers in these systems will make them more powerful, faster and cheaper.

D. WEAPONS CONTROL

This type of system obviously must be capable of standing alone, and is a natural application for automatic control by a computer.

Modern naval warfare requires storage of large amounts of information concerning the location, movement, and characteristics of friendly and enemy units. The air warfare problem requires that fire control computers be almost fully automatic with little need for manual operation.

Today, all new combat ships and aircraft are being built with digital fire control systems, and original equipment on most older units is being retrofitted with digital computers. U.S. Navy has adopted two computers to be standard naval ship tactical digital processors, AN/UYK-7 which can be configured for either small or large fire control system applications, and AN/UYK-20 which is designed to meet most of the small and medium-sized needs.

In the area of weapons control, typical tasks performed by today's digital computers include director positioning, automatic target acquisition and tracking, computation of all gun or missile orders, weapons selection, weapons firing, and control of operator's display consoles. It is believed that in the near future all these tasks will be solved by micro or multi-microcomputer systems.

E. GUN CONTROL

This system is responsible for acquiring, locking onto and then tracking the target; solving the ballistics control problem continuously, and finally, guiding the weapon's bearing and elevation, after receiving ship's data concerning course, speed, roll-pitch and wind speed and direction.

Coupled with a scanning radar, a digital computer or microcomputer system can be used to allow the radar to track selected targets while searching for new contacts. By this way, several different contacts can be simultaneously engaged.

F. GUIDED MISSILE CONTROL

In the AEGIS system tests conducted in 1975, two standard missile firings destroyed two low altitude supersonic missile targets simulating attacks on an aircraft and an aircraft carrier. Each target was shot down with only one human action --the closing of the firing key [10].

Digital computers have been combined with phased-array radar. In this radar, the transmitted radar beam is steered in bearing and elevation by varying the phases of the signals transmitted by each of the elements on each antenna face. This radar and the digital computer system associated with it can do all of the following automatically:

- Track-while-scan.
- Conduct a continuous 360-degree search.
- Transmit mid course guidance commands to several missiles in flight, each missile being aimed at a different target. When the missile acquires the target terminal guidance by the missiles own sensors takes over.
- Automatically initiates countermeasures when jamming is detected.

The combination of all the above features in the digital control system gives the missiles capability to achieve far greater ranges than their predecessors.

G. A CRITICAL QUESTION ABOUT MICROCOMPUTERS

The previous paragraphs examined the use of the computer systems in general tactical applications. At this point the question is:

> "Would it be possible, in the near future, to use computers which are similar to today's commercial microcomputers for tactical applications?"

It is believed that even the most pessimistic expert on computer technology will give an affirmative answer, because of the following reasons:

1. The use of the μC in areas of science and engineering similar to tactical applications increased rapidly in the last five years, and much more increase is expected in the future.

2. The general capabilities of the newest μC are approaching the minicomputer and in some areas exceed those of mini computers.

3. The dimensions and the weight of the μC are very small.

4. The possibility of using many (up to 8 recently and 16 in the near future) single-board μCs in parallel (multi-microcomputer systems) in order to achieve the same results as a single maxi or a large mini computer. The advantages are: reducing the cost approximately 10 to 1 and increasing the capabilities of weapons (i.e., surface-to-surface short range missiles).

5. The low hardware cost of any μC system in comparison with mini or maxi computers; and

6. The need for increased effectiveness of weapon systems in order to remain competitive or become superior to adversaries.

For all these reasons above, it is believed that very soon the μC systems will take their proper place in the Armed Forces and in this way even the smallest combat unit can afford a powerful new tool.

IV. METHOD OF EVALUATION

The major questions to be answered when evaluating microcomputer systems for a particular application are:

1. Will the system have enough computing capacity to satisfy the computational requirements?
2. For the systems which have enough computing capacity, will the systems have support software, pass military qualifications, have multiple sources, etc.?

Question 1 above is addressed in two parts:

Computing capacity of uniprocessors.

Computing capacity of multi-microcomputer systems.

Question 2 above is developed in two parts:

Additional important criteria.

Desirable features.

A. COMPUTING CAPACITY OF UNIPROCESSORS

The measures of capacity which significantly limit the computational capacity are:

- Read only memory (ROM) capacity.
- Random access memory (RAM) capacity.
- Auxiliary memory capacity, disks, tapes, etc.

The memory capacity limits have played an important role in the past because the original requirements of the problem changed and caused an overflow which in turn caused the use of more unicomputers or a revision of the original program to fit into the existing memory. In all cases, substantial

unanticipated expenditures were required to adjust the system to the new requirements.

With microcomputer systems, the memory size limits range from 1-2 Kbytes, on single chip computers, to 65 Kbytes on single board computers, to 16 Mbytes in the newest 16-bit systems. Tables VI, IX, X, and XI tabulate the implemented memory sizes for typical microcomputer systems.

A widely used performance capacity measure is Million Instructions Per Second, MIPS. This measure is misleading because of the wide range of instruction execution times with microcomputers. At least four different categories of instructions must be distinguished to give the crudest measures:

- INTEGER Additions, Subtractions and Comparisons.
- INTEGER Multiplications.
- FLOATING Point Additions and Comparisons.
- FLOATING Point Multiplications and Divisions.

The case study in the following chapter distinguishes between twelve basic functions and thus gives a more refined evaluation of a system.

The performance capacity of a system to carry out required calculations depends on the mix of calculations. The commonly used mixes are: Gibson, Real-Time, Message Processing (table IIa) [29].

The attached Basic program was written in order to calculate performance capacities of representative computer systems, standard Military computers and widely used microcomputer

TABLE IIa

Performance Capacities of Different Computer Systems. Three commonly used mixes are summarized below.

System Mix	Gibson	Real-Time	Message Processing
Fixed (single prec.)			
Add/Subtract	0.061	0.16	0.05
Multiply	0.006	0.05	0.005
Divide	0.002	0.02	0.005
Logical			
Compare	0.038	0.12	0.01
Shift (6 bits)	0.044	0.05	0.03
And/Or	0.016	0.04	0.15
Control			
Load/Store	0.312	0.33	0.47
Conditional branch	0.166	0.1	0.14
Increment and store index	0.18	0.04	0.03
Move reg. to reg. No memory refer.	0.053	0.05	0.058
I/O Control			
Programmed I/O transfer	0	0.02	0
Initialize buffered I/O	0	0.01	0.001
Interrupt response and store 4 reg.	0.122	0.01	0.001

systems, using three common instruction mixes (Gibson, Real-Time and Message Processing). Table IIb contains the instruction times (in μs) for these computers.

The results of these calculations are given in table IIc.

In order to answer the question, whether or not a given computer system has the performance capacity to satisfy the requirements of an application, one must characterize the application in terms of the instruction mix and the number of instructions in this mix to be executed each second. If the capacity of the computer exceeds the requirements, the computer system can be used to carry out the calculations. If the capacity of the computer is smaller than the stated requirements, then either a faster computer must be found or a multicomputer system must be used.

B. COMPUTING CAPACITY OF MULTI-MICROCOMPUTER SYSTEMS

The computing capacity of a uniprocessor system is a serious limitation for applications which at the design stage fit comfortably into a uniprocessor, but several years later outgrow the capacity of uniprocessor. Historically this has been the case with tactical systems, such as NTDS. Even for the newest tactical system, AEGIS, the number of computers in the system has been growing.

A uniprocessor system designed during 1960-70 was not designed to be a building block of a multicomputer system. The microcomputer systems, on the other hand, were specifically designed to be building blocks of larger systems.

TABLE IIb

Performance Capacities of Different Computer Systems
Instruction times (in μs) for different computers.

Instruction	AN/UYK 7	AN/UYK 20	AN/UYK 14	MC 6800 (1MHZ)	INTEL 8086 (5MHZ)	ZILOG 8000 (4MHZ)	MC 68000 (8MHZ)
Fixed (s.p.)							
ADD/SUB reg. to register	1.5	1.5	2.0	5.0	0.6	1.0	0.5
MUL (16 bits)	7.5	4.2	5.3	72.0	26.0	18.75	5.4
DIV (16 bits)	14.5	7.2	10.6	98.0	32.0	23.7	10.9
Logical							
COMPARE (reg. immediate)	1.1	1.2	2.0	2.0	3.0	1.75	1.4
SHIFT (6 bits, register)	1.75	3.5	2.7	12.0	2.4	1.75	1.3
AND/OR (reg. imm.)	1.0	1.5	2.0	2.0	3.0	1.75	1.5
Control							
LOAD/STORE	1.5	1.5	0.9	4.0	2.8	1.75	1.5
COND. BRANCH	1.5	2.25	1.75	3.0	1.6	2.75	0.6
INCR. & STORE	1.5	2.24	1.4	4.0	4.0	5.0	1.3
MOVE register to register	1.1	3.6	7.4	2.0	0.4	0.75	0.25
I/O Control							
Programmed I/O transfer	3.5	4.5	3.5	4.0	2.0	2.5	0.5
Initialize buffered I/O	3.25	4.5	5.8	4.0	4.0	4.0	4.0
Interrupt response & store 4 reg.	4.0	4.0	4.2	18.0	21.2	9.0	4.9

Note: Some given times are coming from separate calculations (MUL and DIV for MC6800), despite lack of such instructions in these computers.

```
10      REM  ** PROGRAM WRITTEN IN TRS-80 DISK BASIC.
20      REM  ** THIS PROGRAM CALCULATES PERFORMANCE CAPACITIES OF
30      REM  ** REPRESENTATIVE COMPUTER SYSTEMS USING THREE COMMON
40      REM  ** INSTRUCTION MIXES.  THESE MIXES ARE GIBSON, REAL-
50      REM  ** TIME, AND MESSAGE PROCESSING.
60      CLEAR 5000
70      DIM B(8,13),G(8,13),R(8,13),M(8,13)
80      DIM N(20),P(8,3),C(3,20)
85      INPUT"DO YOU WISH TO CREATE A NEW FILE (Y/N)";A$
86      IF A$ = "N" THEN 670
90      REM  ** INPUT COMPUTER CHARACTERISTICS
100     INPUT"HOW MANY SYSTEMS DO YOU WISH TO EVALUATE ?";N
105     INPUT"ENTER NUMBER OF CONSTANTS";M
110     FOR I = 1 TO N
120       INPUT"ENTER SYSTEM NAME";N$(I)
140       FOR J = 1 TO M
150         PRINT"ENTER SYSTEM CONSTANT  "J
155         INPUT B(I,J)
160       NEXT J
170     NEXT I
180     REM  ** INPUT INSTRUCTION MIX CONSTANTS
181     INPUT"DO YOU WISH TO ENTER NEW MIX CONSTANTS (Y/N)";A$
182     IF A$ = "N" THEN 790
190     FOR I = 1 TO 3
200     IF I = 1 THEN PRINT"ENTER GIBSON CONSTANTS"
210     IF I = 2 THEN PRINT"ENTER REAL-TIME CONSTANTS"
220     IF I = 3 THEN PRINT"ENTER MESSAGE PROCESSING CONSTANTS"
230       FOR J = 1 TO M
240         PRINT"ENTER EVALUATION CONSTANT "J
250         INPUT C(I,J)
260       NEXT J
265     REM ** STORE INSTRUCTION MIX DATA TO DISK
270     NEXT I
271     OPEN"O",1,"MIXDATA/TXT"
272     PRINT#1,M
273     FOR K = 1 TO 3
274       FOR J = 1 TO M
275         PRINT#1, C(K,J)
276       NEXT J
277     NEXT K
```

```
270     CLOSE 1
280     REM  ** PERFORM CALCULATIONS
290     FOR I = 1 TO N
300       FOR J = 1 TO M
310         G(I,J) = B(I,J) * C(1,J)
320         R(I,J) = B(I,J) * C(2,J)
330         M(I,J) = B(I,J) * C(3,J)
340       NEXT J
350     NEXT I
370     FOR I = 1 TO N
380       G1 = 0: R1 = 0: M1 = 0
390       FOR J = 1 TO M
400         G1 = G(I,J) + G1
410         R1 = R(I,J) + R1
420         M1 = M(I,J) + M1
430       NEXT J
440       P(I,1) = G1
450       P(I,2) = R1
460       P(I,3) = M1
470     NEXT I
475     REM ** PRINT OUTPUT
480     LPRINT TAB(35)"TABLE"
485     LPRINT:LPRINT:LPRINT
490     LPRINT TAB(13)"PERFORMANCE CAPACITIES OF DIFFERENT
        COMPUTER SYSTEMS"
495     LPRINT:LPRINT:LPRINT:LPRINT
500     LPRINT TAB(13)"SYSTEM"TAB(29)"GIBSON" TAB(40)
        "REAL-TIME"TAB(53)"MSG. PROCESSING"
510     LPRINT TAB(13)
"-------------------------------------------------------"
        :LPRINT:LPRINT
520     FOR I = 1 TO N
530       LPRINT TAB(13)N$(I)TAB(29)
          P(I,1)TAB(41)P(I,2)TAB(56)P(I,3)
535       LPRINT
540     NEXT I
545     INPUT"DO YOU WISH TO SAVE DATA TO DISK (Y/N)";A$
546     IF A$ = "N" THEN 1000
550     REM  ** STORE DATA IN A DISK FILE
560     OPEN"O",1,"PERFDATA/TXT
565     PRINT#1,N,M
```

```
570     FOR I = 1 TO N
580       PRINT#1,N$(I)
590       FOR J = 1 TO M
600         PRINT#1,G(I,J),R(I,J),M(I,J)
610       NEXT J
620       FOR K = 1 TO 3
630         PRINT#1,P(I,K)
640       NEXT K
650     NEXT I
660     CLOSE 1
665     GOTO 1000
666     REM  ** INPUT SYSTEM PERFORMANCE CHARACTERISTICS
670     OPEN"I",1,"PERFDATA/TXT"
675     INPUT#1,N,M
680     FOR I = 1 TO N
690       INPUT#1,N$(I)
700         FOR J = 1 TO M
710           INPUT#1,G(I,J),R(I,J),M(I,J)
720         NEXT J
730         FOR K = 1 TO 3
740           INPUT#1,P(I,K)
750         NEXT K
760       NEXT I
770       CLOSE 1
780       GOTO 360
785     REM  ** INPUT INSTRUCTION MIX DATA FROM DISK
790     OPEN"I",1,"MIXDATA/TXT"
800     INPUT#1,M
810     FOR K= 1 TO 3
820       FOR J = 1 TO M
830       INPUT#1, C(K,J)
840       NEXT J
850     NEXT K
860     CLOSE 1
870     GOTO 280
1000    END
```

TABLE II c

PERFORMANCE CAPACITIES OF DIFFERENT COMPUTER SYSTEMS

SYSTEM	GIBSON	REAL-TIME	MSG. PROCESSING
AN/UYK-7	1.8336	2.067	1.42955
AN/UYK-20	2.2782	2.1376	1.7785
AN/UYK-14	2.1297	2.32	1.7297
MC-6800	6.337	9.46	4.338
INTEL-8086	4.991	4.192	2.5804
Z-8000	3.43265	3.209	2.00875
MC-68000	1.63175	1.5795	1.2359

Note: The number in the above table refers to the number of instruction mix per microsecond.

A typical multi-microcomputer system consists of single board computers, each of which contains a processor (Proc), input/ output interfaces (I/O), real-time clocks (RTC), interrupt control (IC) and private memory. The single board computers are connected by a system's bus, which allows common shared memory to be used for passing information between such system almost at the same rate as a computer accesses its private memory. Typically, up to sixteen such single board computers

can be put into one system. A representative multi-microcomputer system is shown in figure 10.

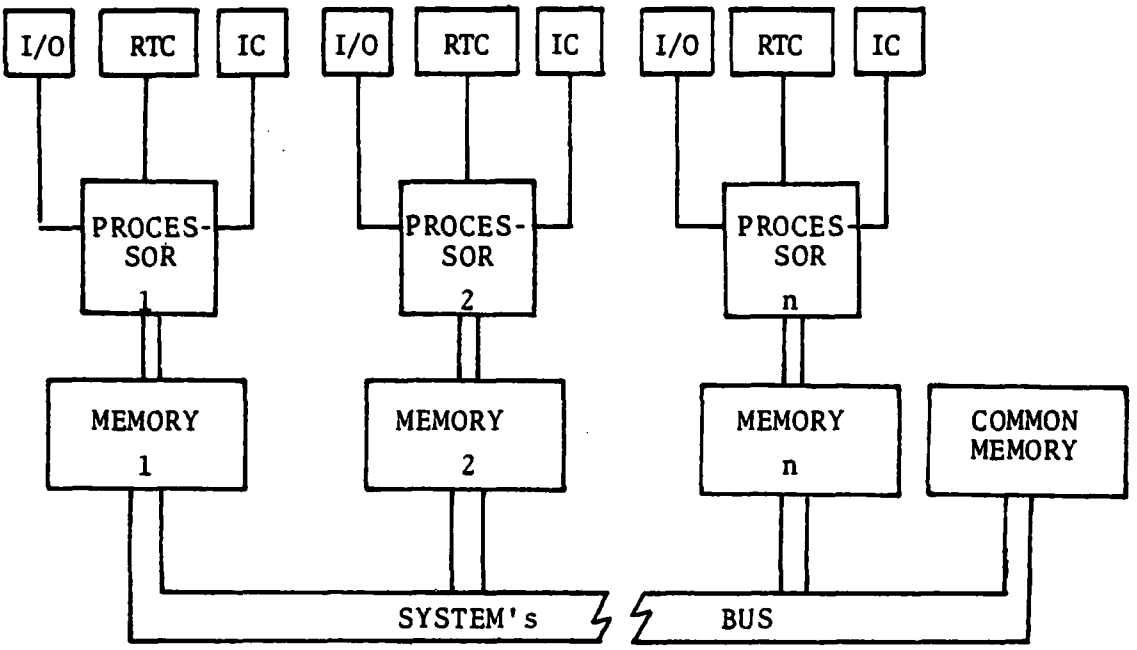


Figure 10. A multi-microcomputer system.

The computational capacity of such systems depends on the single board processors.

Each processor adds memory capacity as well as processing capacity to the system. If the processes can be partitioned among the processors in such a way that each processor is working near its performance capacity and relatively little information is shared between the processors, then n processors yield a performance capacity of n times uniprocessor performance capacity.

If, on the other hand, much information needs to be shared, the system's bus may become a serious bottleneck to the performance capacity. Experimental evidence indicates that if the system's bus is used at 50% or less of its capacity, the performance of each single board processor is not affected. The system's bus usage at 75% capacity caused a loss in performance capacity of about 10% in the system. Use of the bus beyond 80% capacity causes serious losses in performance of the system.

In order to predict the system's bus usage, the number of external variables stored in common memory and the frequency of updates must be estimated. Each time the variable is used by another process resident in another single board computer, another access of the system's bus is necessary. Therefore, the bus use estimate is based on the number of variables multiplied by the frequency of updates and usages multiplied by the bus transaction time of the common memory reference.

In the case study in the following chapter, the system's bus usage is a minor percentage of bus capacity and hence the performance capacity is a multiple of the performance capacities of the individual computers in the system.

Because the performance of the multi-microcomputer system depends on the partitioning of the processes into relatively independent partition elements, the performance capacity is dependent on the application and therefore must be studied in the context of the application. If the bus usage remains less than 50% of bus capacity, the performance capacity is the sum

of the performance capacities of the single board computers in the system.

C. ADDITIONAL IMPORTANT CRITERIA

1. Military Qualifications

a. Dimensions

As it is known, there are a lot of restrictions in warships, submarines, and especially in aircraft and helicopters about additional equipment, because of the restricted space in these units. Some examples of these restrictions are given below:

(1) The door openings in surface ships are 26 x 60 inches.

(2) The diameter of hatch openings in submarines is 21 inches.

Therefore, the smaller the size the more acceptable.

b. Weight

Another important factor which must be very carefully examined is weight, because for any equipment destined for fast boats, aircraft, helicopters, spacecraft, and satellites, equipment must be as small as possible. It must be kept in mind that for an extra pound of weight in an aircraft and similar vehicles, they must add seven pounds weight of additional material to its construction [30]; that means the ratio of extra weight in these cases is seven to one. On the other hand, speaking about fast boats, it must be kept in mind that extra weight on these has a drastic influence on reducing their highest speed.

c. Electrical Power

To operate the microprocessors in most of the new systems, requires a five (5) to twenty (20) volt power supply. However, many of these power supplies are not regulated and ships are famous for having fluctuations, and part or total loss of electrical power. The necessary prerequisites are that the system has a regulated power supply that can withstand fluctuations of 10%, based on the maximum fluctuations that most shipboard electronic equipment must be able to withstand without degradation, and also has a back-up battery with automatic cut-in as power fails. If the system does not have one of those prerequisites, it is not considered capable for shipboard use.

d. Environmental Conditions

These are:

(1) Temperature: It has to withstand temperature ranges of 32 to 100 degrees Farenheit to remain in contention.

(2) Humidity: It has to be operable under relative humidity between 10% and 90%.

(3) Shock Mounts: It has to provide shock mounts for the units.

In order for a microcomputer to be able to withstand these extremes of temperature and humidity, power supply cooling features have to be incorporated into the computer.

2. Reliability

a. Time of Failure/Repair

The minimum "mean Time Between Failures" (MTBF) of 3,000 hours is required for a unit to remain in contention

and receive further evaluation. The maximum of 60 minutes is allowed for "Mean Time To Repair" (MTTR), utilizing the manufacturer's recommended spare parts allowance. It should not be contingent on any extraordinary technical training either.

b. Manufacturers

To determine the best manufacturers of the major components of the system, such as microprocessors and peripherals, is not an easy job, especially when a company is new and small. Therefore, the size of the company, and how long it has been in this kind of business is a very important factor.

3. Maintainability

One of the most important factors to be examined is the availability of replacement parts when a failure occurs to the system. This factor, it can be said, is more important than that of the time between failures.

To simplify maintenance, a diagnostic package, containing the following, must be included:

a. Front panel indications
b. Test points
c. Test programs
d. Test equipment.

In the area of chips that are used, it is more reliable to use the expensive chips that are normally soldered on the boards than those that plug-in.

In order for the system to remain continuously updated, it must be included in maintenance support, which is

the manufacturer's obligation for any software or hardware changes to the system.

If a training program for the maintenance and operator personnel is available, that will be counted as an advantage.

4. Programming Languages for Microprocessors

Programs can be written at various levels. Machine and assembly language do not require much software or hardware support and can produce very efficient programs. These languages, however, are machine-dependent and difficult and time-consuming for the programmer to use. High-level or procedure-oriented languages are not machine-dependent and are significantly easier to use but require a large amount of software and hardware support and usually produce inefficient programs (figure 11).

No less than twenty different microprocessor languages are in use today. Languages can be classified in several ways, considering the use for which they are best suited. Most μC applications seem to be in the area of system programming or control applications (i.e., they require direct access to the machine to perform I/O or control functions or to achieve maximum efficiency). The primary dimension for classifying a language is its distance from assembly language (i.e., its level). There are four categories into which languages are divided: structured assemblers, high-level machine-dependent, high-level μP-oriented and high-level system languages.

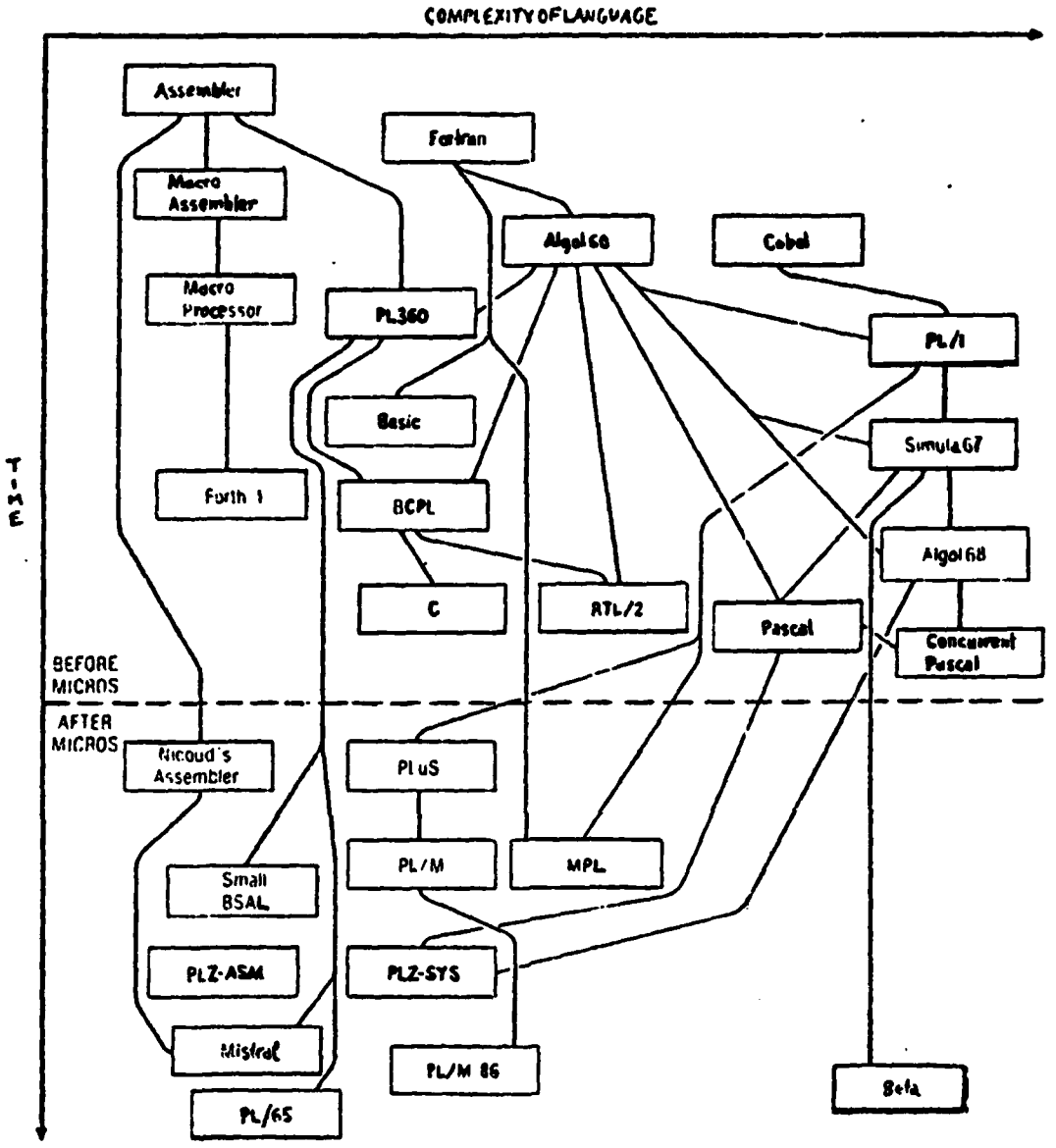


Fig. 11. A genealogy of current languages for microprocessors. A link indicates that the design philosophy of the descendant was strongly influenced by the ancestor but does not imply that one contains the other.

(From: "A Survey of Microprocessors Languages," by S. C. Reghizzi, Computer, January 1980.)

a. Structured Assemblers

The main impetus to the introduction of High Level Languages (HLL) arose from the awkwardness and poor readability of assembler programs, shortcomings which led to high development costs and difficulties in updating programs.

On the other hand, the disadvantage with HLL's is that they take more memory space and more time to execute than assembly languages. A lengthy program in HLL can take an hour to compile. The additional execution time is a disadvantage in some real-time applications and the added memory requirements become critical in single-chip computers with limited read-only memory [13].

A modern structured assembler is PLZ-ASM, a language designed by Zilog to interface with higher-level language PLZ-SYS.

b. High-Level Machine-Dependent Languages

These languages are less machine-dependent than structured assemblers, since executive statements and expressions are written in conventional high-level notations rather than in assembly form. The ancestor of these languages is PL 360, designed by Wirth in 1968 for the IBM 360. To understand a PL language, one has to have a pretty good idea of the machine instructions and machine-idiosyncratic conventions.

In this category four languages are available for microcomputers by increasing levels: Smal/80, BSAL/80, Mistral and PL/65.

c. High-Level Microprocessor-Oriented Languages

These languages--that are largely machine-independent--are especially designed for μP. They avoid some of the resource-consuming features of traditional main-frame languages which would cause unacceptable performance degradation.

In this category are: MPL, PLuS, PL/M and PLZ-SYS. Information and characteristics of these languages are given in table III.

d. Higher-Level System Languages

Excluding APL, Cobol and Fortran (as non-system languages), four representative members of this family are: Basic, Pascal, RTL/2 and C, all originally designed for larger machines.

"BASIC" is available on all microsystems, although some Basic implementation have low quality. Several extensions and dialects derived from Basic are currently available on microcomputers (e.g., LLL Basic, which facilitates interfacing with assembly routines and has been successfully used in a real-time environment).

"PASCAL" is perhaps the best choice for achieving machine-independence on 16-bit (but not on 8-bit) machines in noncritical applications. Pascal will likely be available on most new generation machines; this will be facilitated by the general availability of a portable Pascal compiler which can be easily boot-strapped on new machines. The University of California at San Diego has boot-strapped Pascal (or rather a

TABLE III

Information and Characteristics of Languages--MPL, PL/M, PLZ-SYS

a. General information on MPL, PLuS, PL/M, and PLZ-SYS.

LANGUAGE	MPL	PLuS	PL/M	PLZ-SYS
ORIGINALLY DESIGNED FOR	Motorola 6800	Signetics 2650	Intel 8080	Zilog Z-80
YEAR	1976	1974	1975	1977
NOW AVAILABLE OR ANNOUNCED FOR	M6800 M68000		Intel 8086 M6800 (from independent suppliers)	Zilog Z-8000 Olivetti
CROSS COMPILER	for M6800	for 2650	for 8080	Under development by non-Zilog suppliers
RESIDENT COMPILER	Motorola 6800	-	Intel 8080 8085 8086	Zilog Z-80 Olivetti machines
RESIDENT INTERPRETER	-	-	-	Zilog Z-80

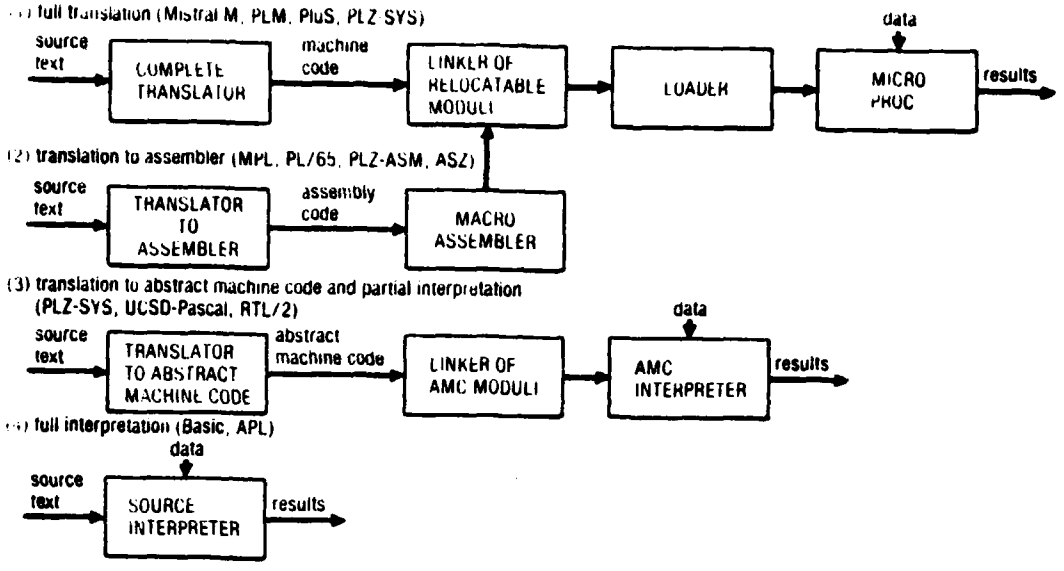
b. Minimum memory requirements for some resident compilers and interpreters. Entries marked with * do not include a standard operating system.

LANGUAGE	MINIMAL MEMORY CONFIGURATION REQUIRED (KBYTES)
PLZ-ASM	32
Smal/80	12*
Mistral M6800	8*
PL65	16
MPL	56
PL/M 8080	64
PL/M 8086	64
PLZ-SYS	48
Basic Zilog	48
Basic Intel 8080	48
LLL Basic	12*
Pascal UCSD 8080	48*
Pascal Texas Instr.	48

c. Simple data types in MPL, PLuS, PL/M, and PLZ-SYS.

LANGUAGE	BIT	1 BYTE BINARY	CHARACTER	WORD (2 BYTES BINARY)	POINTER (2 BYTES)	1 BYTE SIGNED INTEGER (BINARY)	2 BYTES SIGNED INTEGER (BINARY)	SIGNED OR UNSIGNED ASCII CODED DECIMAL	BOOLEAN OR LOGICAL	LABEL
MPL	yes	yes	yes	yes	same as 2 bytes binary	–	–	yes, up to 12 digits	only unnamed Boolean expressions in if and do while statements	yes, label declarations
PLuS	–	yes	yes	same as address	same as address	yes	yes	–	1-byte w/ 2-byte types can be interpreted as true or false under standard conventions	yes
PL/M	–	yes	same as 1-byte binary	same as address	same as address	–	–	–	1-byte or 2-byte types can be interpreted as true or false under standard conventions	yes, label declarations
PLZ-SYS	–	yes	same as 1-byte binary	yes	yes	yes	yes	–	only unnamed Boolean expressions in if statements	yes, restricted to do groups

TABLE III (continued)



d. A spectrum of language processing solutions from full translation to source interpretation.

e. Control structures in MPL, PLuS, PL/M, and PLZ-SYS.

	TESTS			JUMPS		
	2-ways	Multi-ways (*case*-statement)	Interations	*go to*	Procedure calls and returns	Halt
MPL	*if . . then*, *if . . then . . . else*	Computed *goto* (as in Fortran).	*do* i = initial *to* final *by* increment . . .; *do while* Boolean-exp . . .; Parameters restricted to constants or variables.	yes	yes	—
PLuS	*if . . then*, *if . . then else*	One out of n branches is selected according to the value of a control expression in the range 0 . . . n − 1.	*do* i = initial *to* final *by* increment . . .; Parameters may be expressions. *do while* expression . . . *do until* expression . .	yes	yes	yes
PL/M	*if . . then*, *if . . then . . . else*	Same as PLuS.	Same as PLuS, except that there is no *do until* and increment>0.	yes	yes	yes
PLZ-SYS	*if . . then* fi *if . . then else* fi,	The i-th branch is entered if a control expression equals one of K_{i1}, K_{i2} . . . K_{im} (a list of constants), for some $1 \leq i \leq n$. Otherwise the $(n + 1)$-th branch is entered	Never ending loop; *do . . . od*; only loops can be labeled.	Restricted to: *exit* [*from* label] *repeat* [*from* label] where label denotes a loop.	yes	—

TABLE III (continued)

f. Procedures and parameter passing in MPL, PLuS, PL/M, and PLZ-SYS.

	Allowed actual parameters	Parameter transmission	Call by address of 1st instruction	Interrupt procedures	Reentrant (recursive) procedures	Functions	Multiple exits	Multiple return values
MPL	Constants, nonsubscripted variables, array names Parameters are forbidden as control variables in control statements, subscripts, and parameters of an inner procedure.	By value, (by address for arrays and based variables). User specifies transmission via registers.	—	—	no	no	—	yes
PLuS	Expressions, addresses of arrays. Automatic conversion of actual parameters types to formal parameters.	By value (by address for based variables). The last two parameters and result are passed via registers	—	—	no	yes	yes	—
PL/M	Expressions, addresses of arrays or structures	Same as PLuS, but parameters via stack for reentrant procedures	yes	yes	Must be explicitly declared	yes	yes	—
PLZ-SYS	Constants and variables. Pointers to arrays and structures.	By value (by address for pointed variables)			Mutually recursive procedures must be declared in separate modules	yes	no	yes

g. Comparison of languages processing solutions with respect to program execution, program preparation, and portability. Numbers denote qualitative ordering. Note that the evaluation is very much dependent on the source language and on the target machine.

		COMPLETE TRANSLATION	TRANSLATION TO ASSEMBLER	TRANSLATION TO AMC AND AMC INTERPRETATION	SOURCE LANGUAGE INTERPRETATION
program execution	EXECUTION SPEED	1	1	2	3
	SIZE OF OBJECT PROGRAM	3	3	2	1
	SIZE OF RUN-TIME SYSTEM	1	1	2	3
	RUN-TIME DIAGNOSTIC AND DEBUGGING	3	3	2	1
program preparation	OVERALL SIZE OF PROGRAM PROCESSING PACKAGE	3	4 (includes assembler)	2	1
	TIME TO PREPARE A PROGRAM FOR EXECUTION	3	4	2	1
	DIAGNOSTICS AND INTERACTIVITY	2	3	3	1
	MEMORY SIZE TO COMPILE	3	2	1	—
portability	PORTABILITY TO OTHER MACHINE	3 (good with retargeting techniques)	2	1	2

(From: "A Survey of μP Languages," S. C. Reghizzi, Computer, Jan 1980)

subset) on a number of different systems--i.e., Intel, Digital Equipment, Motorola, Texas Instruments and Zilog.

"RTL/2", originally designed by ICI, the British chemical firm for real-time control applications on microcomputers, currently is advertised as a microprocessor language. A Multi-Task System (MTS), mostly written in RTL/2, is available for mini computer DEC-PDP/11. Recently a micro-RTL/2 for Intel, Motorola and Zilog microcomputers was announced. These new compilers will use the time-honored technique of translation to an intermediate language, followed by interpretation (as with UCSD-Pascal) in order to achieve portability.

"C" is available on microcomputers and larger main-frames: a subset, "Tiny C", is available on several microcomputers as an interpreter. C can be compiled into reasonable code on a variety of machines (it was used to write the very successful operating system UNIX), but is not very readable due to the rich set of operators (more than 50), which can be combined in rather peculiar ways.

e. CMS-2 Programming Language

CMS-2 (Compiler Monitor System 2) is a computer programming system developed for the U.S. Navy. Primarily designed for real-time, command and control applications, it is equally applicable to other data processing and scientific problems.

CMS-2 was designed to replace the CS-1 compiler and MS-1 monitor system and developed in the late 1950's and early 1960's. CS-1 was used primarily for NTDS and is

oriented to the CP-642/USQ-20 family of computers, which have been replaced by newer computers, and is too machine-dependent to meet new programming demands.

Within CMS-2 are five components (compiler, loader, librarian, utilities, and assembler) which are controlled by a routine called the Monitor. The standard input is punch cards and the standard output is source cards, binary cards or print lines.

Within a single job, several components of the CMS-2 system may be executed (i.e., compiling a source program for the AN/UYK-7 computer and loading the object code into the computer for execution).

CMS-2 system was designed in 1967 and from then on it is in use in the U.S. Navy as a basic language for tactical data systems. It is available to mini and maxi computer systems but not yet to microcomputer systems.

f. Ada Programming Language

Ada, which is named after the world's first computer programmer, Lady Ada Lovelace, is a very new programming language and is designed for the necessities of Department of Defense (DOD) by CII Honeywell Bull of France. The language has reached the fine-tuning stage, but the first production-quality compiler for the language probably will not be available until early 1982 [17].

The Ada compiler will consist of a root compiler that will produce machine-independent intermediate code and

object code generators that will translate the intermediate code into the machine language of specific computers. The Army has initially specified code generators for AN/GYK 12, PDP 11/70, VAX 11/780 and LITTON L3050. The Air Force asked for IBM 370, Perkin-Elmer Corp. model 8/32, DEC system 10, and the Control Data Corp. model 6600.

While the Army and the Air Force have embraced Ada, the Navy still has reservations [18]. INTEL Corporation reportedly is supporting Ada in its new microprocessor series [32].

Ada is an example of one potential microprocessor HOL standard, because it is being developed to serve programming needs for real-time processing, including microprocessor applications.

5. Operating System

The term "operating system" (O.S.) denotes those program modules within a computer system that govern the control of equipment resources such as processors, main storage, secondary storage, I/O devices, and files (figures 12, 13, 14). These modules resolve conflicts, attempt to optimize performance, and simplify the effective use of the system. They act as an interface between the user's programs and the physical computer hardware [7].

Sophisticated operating systems increase the efficiency and consequently decrease the cost of using a computer. At this point it is noted that, as estimated, 70% of the amount of money that is spent in a year on computer products and

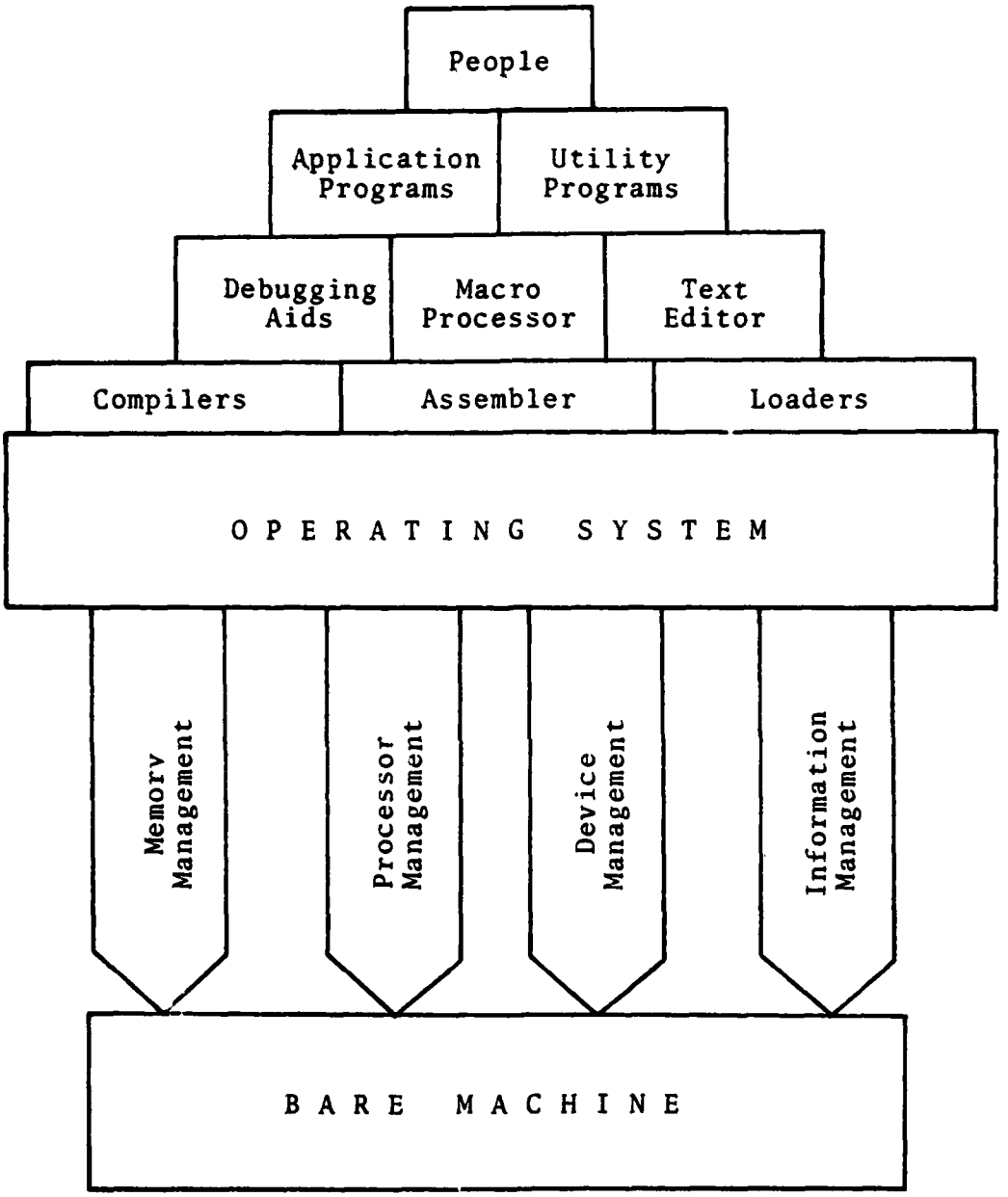


Figure 12 Relationship of O.S. to basic computer hardware.

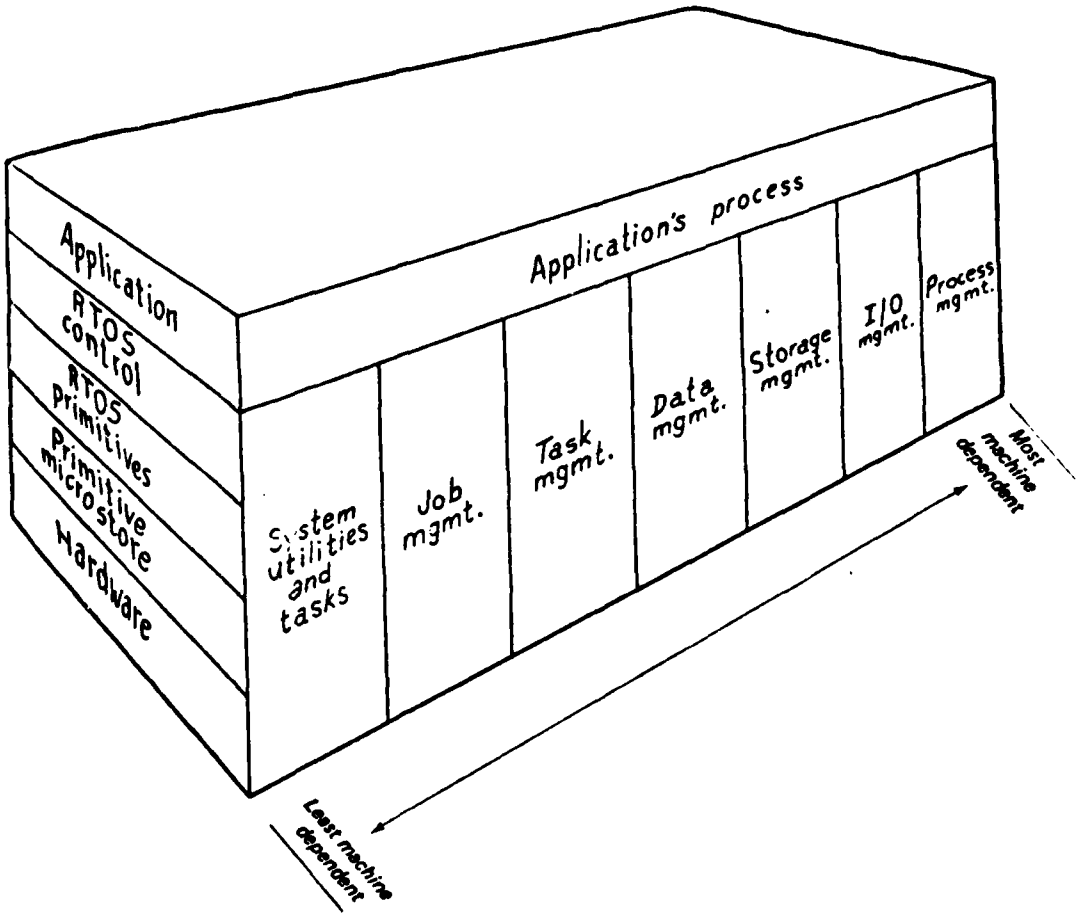


Figure 13 Real-time operating system conceptual approach. (From: "Distributed MicroMinicomputer Systems," Cay Weitzman, 1980)

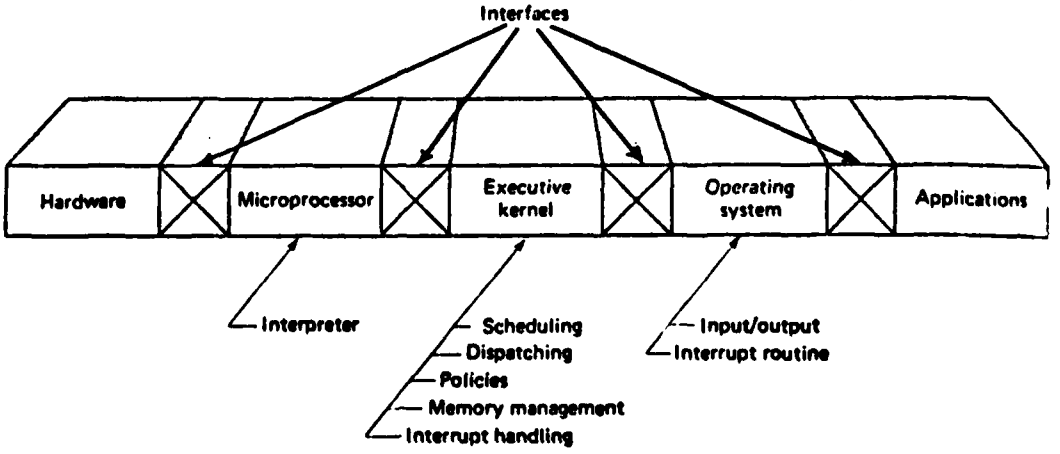


Figure 14 Overview of Kernel layering for general real-time operating system. (From: "Distributed Micro/Minicomputer Systems," Cay Weitzman, 1980)

services goes into software development, data processing salaries, and other software-related activities.

The O.S. is divided into two main categories: (1) developmental, and (2) target, or executive.

In the first category, they belong to O.S. which cover a lot of applications of many kinds of computer systems, and for that reason are very complex and their cost is high. "MULTICS" (Multiplexed Information and Computing Systems) belong to this category, which employs the concept of "rings of protection," links, hierarchical structure, and dynamic linking. Also, from March 1980, there is available, especially for microcomputers, the Taurus O.S. which can spool to multiple printers, has built-in security features with multiple priority levels and is capable of reporting the status of every running job in the system [15].

In the second category, the O.S. are very simple and are designed for a given computer to support specific applications (i.e., fire control system). The advantages of this O.S. are simplicity and low cost.

In order to evaluate the O.S. that are available for different microcomputers, it is necessary to determine what the general application area is (i.e., time-sharing, process control, etc.) and the efficiency of the O.S.

Transportability is also an important consideration relating to software cost and software development time.

The more programming languages and input and output devices that the operating system can support, the more powerful it is and the more desirable it becomes.

The efficiency of an O.S. is also important. An O.S. may be able to support distributed systems or security, but if the necessary time to accomplish it is "long", i.e., 30 seconds delay waiting for feedback of an interactive system seems "long" and one minute seems "too long."

It is necessary to examine any of the following special features of an O.S.:

a. Type of file protection or security provisions.

b. Method of starting and restarting the system.

c. Form of memory retention capability in case of power loss.

d. Error recovery method such that the user will be able to understand them, without reference to publications.

e. Ease of use.

D. DESIRABLE FEATURES

1. Availability

Availability (A) is defined as the percentage of time a computer system is available and is given by the following equation:

$$A = \frac{MTBF}{MTBF + MTTR}$$

For example, given:

MTBF = 3000 hours and MTTR = 48 hours

$$A = \frac{3000}{3000 + 48} = 0.98425 \simeq 98.4\%$$

MTBF = 3000 hours and MTTR = 1.5 hours

$$A = \frac{3000}{3000 + 1.5} = .99950 \simeq 99.9\%.$$

From these examples it can be seen that availability can be improved by increasing the MTBF and/or decreasing the MTTR.

2. Software Material

Software evaluation is so complex a task that it is at present more of an art than a science. The number of factors to be considered is so large and their interaction and implementation so complex that thorough evaluations of any but the simplest products are exceedingly costly and, as a result, are seldom if ever done [12].

Attention will have to be focused on the most important aspects of the evaluation process at the expense of the less important ones. A key question, then, is: "Which aspects are most important?" The answer hinges on two basic points:

(1) The software package must contribute in a non-trivial way to helping the user firm achieve its basic objectives, which generally include a certain type and level of service and long-term survival.

(2) The software package should be able to adapt to changes--most notable, growth of a company's volume of business--and continue to function in a useful way for as long as possible.

Nearly all objectives for software products fall into one or more of the following categories:

(1) Functionality

(2) Capacity

(3) Flexibility

(4) Usability

(5) Reliability

(6) Security

(7) Performance

(8) Serviceability

(9) Ownership

(10) Minimization of operating and maintenance costs

(11) Minimization of purchase and installation costs.

The categories are certainly not mutually exclusive. For example, functionality, performance and capacity are often strongly interrelated. Some examples, by category, follow:

a. Functionality

Do the input transactions, files and reports contain the necessary data elements?

Are all necessary computations and processing steps performed the way you want them performed?

Are the types of ad hoc inquiries (the ones you would like to make) possible?

b. Capacity

Find out if the product will be able to handle your requirements for the size of files, number of data elements, number of table entries, volume of transactions, volume of reports and number of occurrences of certain data elements.

Determine whether response or turnaround time will be adequate.

Check all limitations.

c. Flexibility

Can transaction and report formats be changed easily?

Can screen layouts be changed easily?

How easy is it to add new computations or processing steps?

Can the programs be adapted to new applications?

d. Usability

Does the level of technical knowledge required to use this product properly match the level of knowledge of those who will be using it?

Is the user documentation complete, easy to read, easy to understand and up-to-date?

How readable, informative and easy to interpret are the reports and screen displays produced?

Is training available from the vendor?

e. Reliability

Does the product have a clear, modular design?

Has it been in actual use long enough to make sure that most of its bugs have been cleaned up?

How much of the system will become unusable when a part of it fails?

Does the product rely on any failure-prone hardware or noisy communication links?

Does the product incorporate any features for the detection and self-correction of errors?

Are there errors a user can make which will bring down the system?

What are the recovery capabilities?

f. Security

Does the product incorporate standard types of controls?

Does the product permit adequate back-up procedures?

Does it assist in any way to prevent intruders from extracting sensitive data from files or transaction input streams?

Does it help prevent employees from entering unauthorized transactions or running programs without authorization?

Does the product provide adequate detection and diagnosis of data entry and other types of errors?

Is a standard procedure provided that can be used to verify that the system is functioning properly?

g. Performance

At what rate will the product typically use machine cycles?

At what rate will it be requesting disk accesses?

How much main memory will it require?

If it will be run on a virtual memory system, how much paging activity will result?

How many users can be on the system before it begins to bog down?

h. Serviceability

Are the source programs available? (Often they aren't).

If the vendor will be doing maintenance, how reliable and accessible is the company?

What level and quality of maintenance will the vendor supply? Is this guaranteed in writing?

Will changes to the system invalidate the warranty?

Can the product easily be used in another operating environment? That is, is it portable?

If your staff will be doing the maintenance, are the programs written in a language with which your staff is familiar? Does it use techniques with which staffers may not be familiar?

Are sets of test data available with adequate documentation of how to use them and of what results to expect?

i. Ownership

What kind of rights to the product are you buying?

Can you resell or rent the product to someone else?

If the vendor is making the product especially for you, will it also be marketed to others? If so, will the vendor pay you a royalty on each sale?

Are there restrictions on copying the product or its documentation?

Are there restrictions on the purposes for which the product may be used?

Will you be able to obtain full ownership rights and copies of the source programs if the vendor goes out of business?

j. Minimizing Operating and Maintenance Costs

How much does the vendor charge for maintaining and upgrading the product?

How frequently will maintenance probably be required?

What is the cost per transaction of using this product?

What will the storage costs be?

Will purchase of this product require acquisition of additional hardware or personnel? How much will they cost on a continuing basis?

What will be the usable life span of this product?

k. Minimizing Purchase and Installation Costs

What initial costs are there besides the basic costs of the product?

Will you have to pay shipping costs? Will you pay transportation and lodging costs of the vendor personnel who will install the system and train your staff?

Will there be considerable delay between placement of the order and actual delivery of the product in a ready-to-use state? How much will this cost?

What will conversion costs be?

Any list of objectives should include items from each of the 11 categories. If any one of them is neglected, the adequacy of the evaluation must be suspect.

Reliability and programming costs are currently the most important criteria in software development for microprocessors. The primary objective is to develop a program

that works with a reasonable expenditure of time and money. The major cost in most μP projects is programming time; methods that can minimize the time required to complete a program are especially important. With the availability of larger memories, cheaper hardware, and faster processors, time, memory, and hardware constraints are not as critical in μP software development as they have been in the development of software for larger computers.

3. Ground Rules for Graphics Software Package Evaluation

The main reason to present separately this category of software is the tremendous importance of computer/graphics for Tactical Applications, especially in the area of Command-Control and Communications (C^3). An example of their usefulness today is the use of image processing systems from Military Intelligence personnel, who use Image Processing to see through camouflage and detect objects of military interest in aerial and satellite photographs [22].

A typical computer/graphics system includes a computer (mini or micro), I/O devices and peripherals. The heart of this system is the software package that accepts graphics input and interaction and produces graphics output [21]. Therefore, in order to evaluate this software package, the following important issues must be carefully examined:

a. Simplicity

Features that are too complex for the application programmer to understand, indicate that there is no simplicity.

Anything difficult to explain in the user's manual will almost certainly be difficult to use.

b. Consistency

A consistent graphics system is one that behaves in a generally predictable manner (i.e., function names, calling sequences, coordinate systems, etc. should, without exceptions, follow simple and consistent patterns.

c. Completeness

The system must provide a set of functions that can conveniently handle a wide range of applications; missing functions will have to be supplied by the application programmer, because completeness does not imply comprehensiveness.

d. Robustness

The system, in the case of a serious error of the programmer, should report the error in the most helpful manner possible. Only in extreme circumstances should errors cause termination of execution, since this will generally cause the user to lose valuable results.

e. Performance

Graphics-system performance is often limited by such factors as O.S. response and display characteristics. The system has to maintain consistent performance, so that application programs provide an equally consistent speed of response.

f. Economy

Graphics systems should be small and economical so that adding graphics to an existing application program can always be considered.

4. Expandability

For any microcomputer system, expandability is necessary. The minimum of 32 Kilobytes of RAM is required for the microprocessor to be considered at all. It has to be expandable to a minimum of 64 Kilobytes for serious consideration to be given for Navy use. This is based on projections of future applications requiring more memory in addition to immediate applications.

In evaluating the complete microcomputer system, it is important to examine the ability of the computer to support peripheral devices. Also, in order for the computer to be compatible with different terminals, printers, etc., the number of bits/sec (BAUD) must be able to be switched. The switching system, wires or toggles, must be able to be plugged into different positions on the board.

The computer has to have a serial and parallel port capability for support of both low and high speed peripheral hardware. It is desirable that the system have a mass storage unit capable of holding 1-15 megabytes of data, a terminal, and a printer.

It is also important, as it affects ease of maintenance, the way that the "mother board" can have additional boards added to it. For adverse environments, such as shipboard or airborne use, special springloaded connectors are necessary to insure reliable connections between removable units.

5. Peripheral Devices

Among peripheral devices are included the I/O devices that will interface with the microcomputer (terminals, mass storage devices and printers).

When examining these devices, the following must be considered:

(1) Number of bits per second that are transferred (baud rate).

(2) Access speed of the device.

(3) Amount of overhead necessary in the O.S. to support the device.

(4) If the device, for an effective interface with the computer, needs serial or parallel ports.

a. Analysis of I/O Devices

(1) Terminals. There are two basic categories of terminals. The first of these, called "SMART," which comes with a built-in microprocessor and allows many applications without having to access the main computer (i.e., text processing). The second operates like a typewriter, for I/O between operator and computer.

Both categories come as "hard copy" terminals (including built-in printer) or video terminal (without printer).

Special characteristics to be examined:

(a) Send and receive lower and upper case letters.

(b) Include full 128 ASCII character set.

(c) Video terminals require a minimum of 80 characters per line and 20 lines on the screen.

(d) Hard-copy terminals are required to have capability of 132 characters per line.

It is underlined here that the "video" terminal is faster than the "hard copy" terminal.

(2) Mass Storage Devices. In this category belongs: hard-disk units, "floppy" disks, cassettes, and paper tapes. These units increase the capability of microcomputers by providing a virtual memory of up to several million bytes.

Special characteristics to be examined:

(a) Memory capacity

(b) Access speed

(c) Error rate

(d) Pilferability potential

(e) Method of computer interface

(f) Required "baud rate"

(g) Power requirements

(h) Potential for information loss, due to unit failure caused by power fluctuations or rough seas.

(3) Printer. The type of printers that are available to interface with the microcomputers are: line, dot matrix, friction feed, band, daisy wheel, and teletypes. Special characteristics to be examined:

(a) Number of characters per inch

(b) Line speed

(c) Print quality

(d) Type of ports required for interfacing

(e) "Baud rate" required.

6. Operational Capabilities

It is desirable for the microcomputer system to have the following capabilities:

a. To support a minimum of three users on line at any time (centralized data base).

b. To have a form of write protection, to prevent the data base from any inadvertent change.

c. Direct Memory Access (DMA), which allows a high speed device to communicate directly with on-line memory, is a desirable feature.

d. It is also desirable to have multiply or divide capability, or to have floating point arithmetic capability.

e. I/O interrupts, to control I/O operations, and supervisor call interrupts, which are necessary to perform actual I/O.

7. Throughput

Any meaningful evaluation of computer performance must be based on the execution times of typical programs in typical applications. For example, for the Z-8000, these applications may involve high-level language compilers, operating systems and large data-base management.

8. Life Cycle Cost

a. Definition

Life cycle cost is the methodology for combining the results of all other evaluations into a single evaluation criterion which would be realistic and meaningful to the decision maker.

Unfortunately, the concept of computer productivity or performance is confounded by the great flexibility, both in design and in use, of these machines.

Computer characteristics or capabilities which are essential measures of performance for one user are often irrelevant to another. Additionally, recent architectural advances which allow software to perform previous hardware functions, and vice versa, further blur the concept of "Machine" performance [6]. A total cost model is shown in figure 15.

b. Life-Cost Levels

As it is shown in figure 15, life-cost is divided into five levels and each one of these into two or more phases [23].

At the first level, life-cycle cost is obtained by summing costs incurred during design, development, verification and validation, and the operation and maintenance.

The development phase cost involves documentation, test and implementation of hardware and software, level 2. Hardware implementation cost consists of the cost of acquisition, manufacture, and installation of microcomputer and interconnecting link(s), in case of multi-microcomputer systems.

Cost of data processing is the sum of items shown on level four, which includes the cost of peripherals and facilities. The interconnecting network cost breaks down into cost of links and possible traffic switches (depending on architecture) that may reside either at computer locations or intermediate facilities for relaying and/or switching.

Software cost is generally much more difficult to assess than hardware, and besides that if cost models exist,

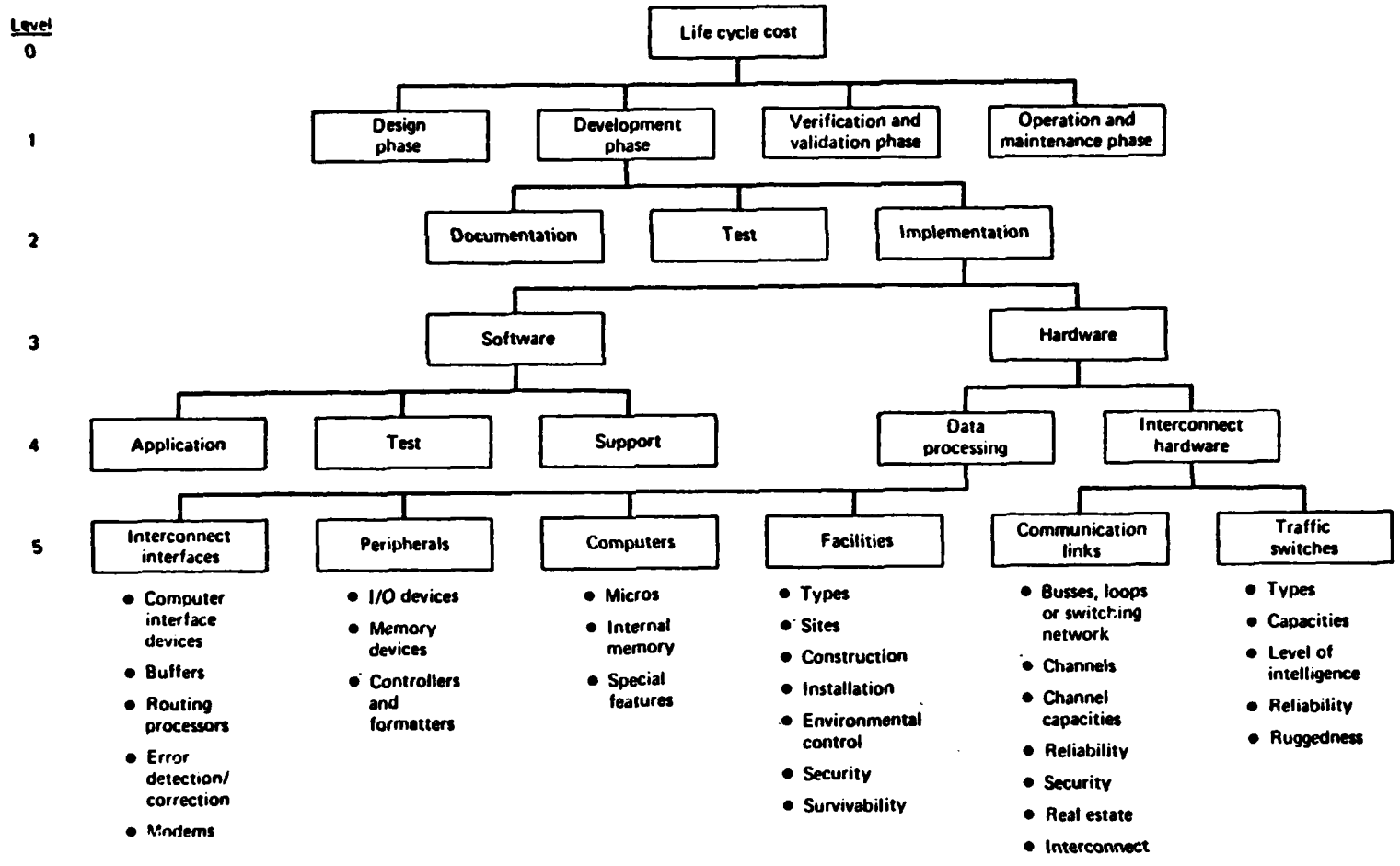


Figure 15 Life-cycle cost associated levels and elements at each level for a microcomputer system. (From: "Distributed Micro/Mini Computer Systems," Cay Weitzman, 1980).

they should be used with caution, as this phase may cost the user 70% of the total cost.

c. Calculations

All costs associated with the four phases on level one can be expressed as follows:

$$C_{MCS} = \sum_{i=1}^{4} (C_{HW} + C_{SW})_i \qquad (1)$$

where, C_{MCS} = total life-cost of microcomputer system

C_{HW} = hardware cost associated with the ith life-cycle phase

C_{SW} = software cost associated with the ith life-cycle phase.

The annual cost of the operational and maintenance phase can be expressed as a fraction of the sum of the total costs of the first three phases:

$$C_{MCS} = (1 + fY) \sum_{i=1}^{3} (C_{HW} + C_{SW})_i \qquad (2)$$

where, f = annual operational and maintenance cost (expressed in "per-year" terms)

Y = microcomputer system life-cycle time period of operational service, number of years.

The hardware cost can be decomposed as follows:

$$C_{HW_i} = (C_{DP} + C_{FAC} = C_{NET})_i \qquad (3)$$

where, C_{DP} = data processing hardware cost associated with phase i.

C_{FAC} = facilities cost associated with phase i.

C_{NET} = communications/interconnect cost associated with phase i.

The total computer hardware cost associated with the implementation aspect of the development phase, in case of multi-microcomputer systems, can be expressed as follows:

$$C_{DP_2} = \sum_{j=1}^{G} C_{DP_{2,j}} \tag{4}$$

where, G = total number of microcomputers used in the system.

$C_{DP_{2,j}}$ = the total hardware computer cost.

Software cost can be decomposed into the cost of developing real-time (RT) and non-real-time (NRT) code, i.e., the software cost term of equation (1) becomes:

$$C_{SW_i} = C_{NRT_i} + C_{RT_i} \tag{5}$$

where, C_{NRT_i} = cost of non real-time machine language instructions (NRT MLI's) during phase i (note that the equation is equally valid using HLL for programming)

C_{RT_i} = cost of real-time machine language instructions (RT MLI's) during phase i.

The cost of developing NRT MLI's can be expressed by the following equation:

$$C_{NRT} = c_{NRT}\, I_{NRT} \tag{6}$$

where, c_{NRT} = cost per non real-time machine language instruction

I_{NRT} = number of NRT MLI's.

and similarly,

$$C_{RT} = c_{RT} I_{RT} \quad (7)$$

where, c_{RT} = cost per RT MLI

I_{RT} = number of RT MLI's.

The total software required during the development and life cycle of the system can be divided into three major categories:

(1) Application, which is the operational software that supports user requirements (i.e., sensors, time table, etc.).

(2) Test, which serves to test the applications software at various levels of development (from algorithm to process) and during integration and validation testing (system level testing). Test software may consist of simulation software used to exercise and test the applications during the development phase by simulating the data processing subsystem, environment, and sensor subsystem and of data reduction and analysis of similar test tool programs, depending on the particular system application.

(3) Support, which consists of programs specially developed in support of a particular system including programs for configuration management, system maintenance, etc.

Hence, all the software required for a micro-computer system can be expressed as follows:

$$SW = SW_{APPLIC} + SW_{TEST} + SW_{SUPPORT} \quad (8)$$

Similar to previous groupings, each element can be further broken into real-time and non real-time software.

The cost per machine language instruction (MLI) has been found, from practical experience, in 1977 to be typically as follows [23].

$$C_{NRT} = \frac{\$42}{NRT\ MLI} \quad (9)$$

$$C_{RT} = \frac{\$120}{RT\ MLI}$$

The above rates include the following charges:

- Direct labor.
- Overhead.
- General and administrative.
- Fee.
- Computer time.
- Documentation.
- Travel.

The rates apply to the following major development phases:

- Requirements definition and analysis.
- Preliminary design.
- Code and debug.
- Testing (algorithm, routine, task subprogram, program, system integration, evaluation and validation, and acceptance by user).

d. Physical Dispersability and Survivability

Physical dispersability and survivability are closely related. Survivability is defined as the probability that a multi-microcomputer or minicomputer system can sustain hardware losses due to hardware failures, software errors, and hostile action and continue to carry out the nominal mission objective, as spelled out by the performance requirements, without degradation. Survivability is thus related to the probability that a system will fail during a particular time interval based on a certain system availability [23]. This attribute is important for systems that may be subject to destruction (i.e., vandalism, sabotage, or other forms of hostile action such as in military environments).

Survivability can be achieved using redundant, dispersed computers and transmission links between computers that are less susceptible to damaging effects or events, e.g., by operating in ways that reduce the probability of damage and by using redundancy techniques to increase the probability of maintaining the required connectivity of the system.

Figure 16 summarizes the relative importance of the key design attributes in some typical multi-microcomputer and minicomputer application areas.

e. Modularity

Modularity for growth is synonymous with adaptability, enhancement, extensibility, changeability, and modifiability. Growth can often be equated to design specifications and, in particular, design specifications for systems where

Attributes \ Application areas	Automotive control	Space, avionics, shipborne control	Medical monitoring	Business, point of sale, electronic funds transfer, transaction processing	Laboratory, scientific data processing	Industrial process control	Data acquisition, file management	Military tactical command and control	Simulation, training
Performance (throughput, response, etc.)	M	H	M	M	H/M	H/M	H/M	H	H
Availability	M	H	H	M	L	H	H	H	M
Reliability	M	H	H	M	L	H	H	H	M
Fault tolerance (path, switching elements, modes, etc.)	L	H	H	H	L	H	M	H	L
Life cycle cost (off the shelf hardware, software, ease of development, etc.)	H	M	H	H	H	H	H	M	M
Modularity/growth	L	M	M	H	H	H	H	H	H
Form factor (volume, weight, power)	H	H	L	L	L	M	L	H	L
Physical dispersability and survivability (failure reconfigurability)	L	H	M	M	L	H	M	H	L

Fig.16 Examples of potential application areas and the relative weighting of attributes for a distributed micro- or minicomputer system(H=very important, M=moderately, L=unimportant or incosequential).

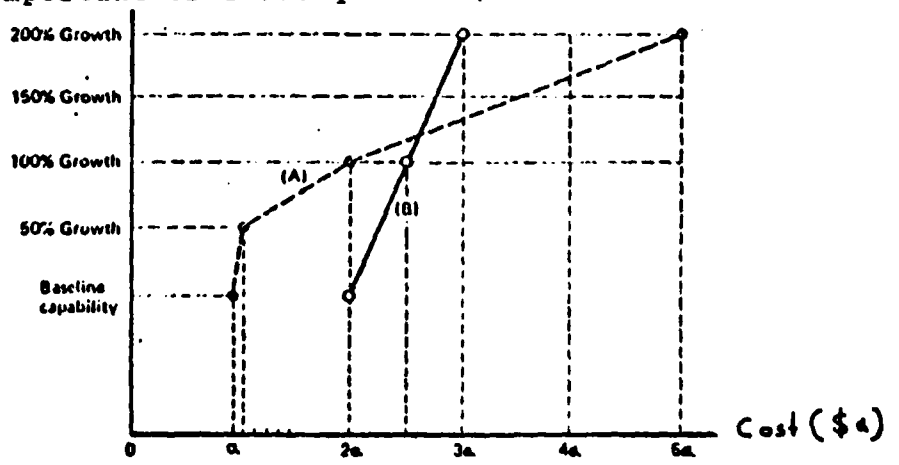


Figure 17. Comparison of two potential distributed computer systems, A and B. As it is seen when A expanded to 3 times its original performance level, is 67% more costly than B.

(Both fig.16,17 are from:"Distributed Micro/Minicomputer systems", Cay Weitzman, 1980)

the exact future needs are difficult to predict [23]. Growth requirements are nevertheless imposed to minimize the system life-cycle cost. This is best illustrated by an example, as shown in figure 17.

V. CASE STUDY

The case study is presented to illustrate by means of a concrete example how the performance requirements are expressed in terms of parameters which are used to estimate the performance of any particular computer system.

The attack aircraft A6-E tactical system was chosen because it serves as a realistic representative system which exists and is suitably documented in references [30] and [31]. Also, the A6-E tactical system provides an example which requires the use of multi-microprocessor system instead of uniprocessor system.

This case study therefore serves to illustrate how the performance analysis is extended to multi-microcomputer systems.

A. THE A6-E OPERATIONAL FLIGHT PROGRAM

The operational flight program of the A6-E aircraft performs the following functions:

a. Navigational calculations.

b. Tracking and ranging calculations.

c. Ballistics calculations.

d. Sensor I/O and steering updates.

1. Description of the Functions

a. Navigational Calculations

The sensors used to generate information for the navigational system are:

(1) Inertial navigational

(2) Doppler radar

(3) Magnetic compass

(4) Airspeed indicator

(5) Altimeter.

The computer program is subdivided into nine segments, each of which is documented as a flowchart page. Table IVa contains the breakdown of each page of the flowchart with a count of each type of instruction, as well as the number of calls to library subroutines.

b. Tracking and Ranging

The purpose of this segment of the program is to establish the position of the target with respect to the aircraft. Tables IVd and IVe give the performance requirements for these calculations.

c. Ballistics Calculations

These determine, from the initial conditions at release, the down range travel and the time of fall of any particular weapon. The ballistic calculations are described in Table IVc. Table IVf contains the attack decision-making process.

d. Sensor I/O and Steering

This segment of the program controls the analog to digital converter. Correction signals to inertial navigation unit, radar antenna control, display control and steering commands are generated by this segment. Table IVb contains the performance parameters.

2. Explanations for Tables IV (a-f)

The floating point operations were chosen to characterize the program, because of the programmer convenience for using floating point operations, whenever available. Because floating point hardware is becoming less expensive and more efficient, programmer convenience is considered more important than the additional hardware expenses.

The column headings refer to the instruction types categorized as follows:

- C - Conditional branch for integer operands
- L/S - Load or store an integer
- CF - Conditional branch for floating point operands
- LFS - Load or store a floating point operand
- FAD - Floating point addition
- FMU - Floating point multiply
- FDV - Floating point divide
- CO - Cosine function
- SI - Sine function
- AT - Arc tangent function
- LN - Logarithmic function
- SQ - Square root function

The two rows of numbers, corresponding to each flowchart page, are the instruction counts in the most frequently executed execution path in the upper row and the most time consuming execution path in the lower row.

The two rows of totals at the bottom of the page accumulate the upper rows for the most frequent execution path, and the lower rows for the worst case execution path.

TABLE IVa

A6-E Navigational Function Complexity Measures

	C	L/S	CF	LFS	FAD	FMU	FDV	CO	SI	AT	LN	SQ
AIR DATA 1	3	9	3	30	10	7	2	0	0	0	1	0
	6	12	3	40	11	8	5	0	0	0	2	0
AIR DATA 2	1	7	0	23	10	7	2	2	0	0	1	1
	2	10	0	33	14	10	2	2	0	0	0	2
AIR MASS	1	3	6	29	11	4	4	1	1	2	0	1
ANGLES	4	6	6	38	11	4	4	1	1	2	0	1
DOPPLER	1	5	0	26	16	6	0	1	1	0	0	0
VELOCITY	2	6	3	33	17	6	1	1	1	0	0	0
SYSTEM	4	2	0	37	10	10	1	1	1	1	0	3
VELOCITY	6	6	0	62	16	14	2	2	2	1	0	3
BARO IN	2	2	3	22	9	6	2	0	2	0	0	0
VERT LP	4	6	5	52	11	7	3	0	2	0	0	0
INERTIAL	2	3	0	27	11	8	2	1	2	2	0	2
ANGLES	2	3	0	27	11	8	2	1	2	2	0	2
PLATFORM	1	3	2	36	14	13	5	1	0	0	0	0
CORRECT1	1	7	2	40	14	13	5	1	0	0	0	0
PLATFORM	1	1	0	58	27	22	6	1	1	0	0	0
CORRECT2	1	1	0	66	31	24	8	1	1	0	0	0
TOTALS	16	35	14	288	118	83	24	8	8	5	2	7
	28	57	19	391	136	94	32	9	9	5	2	8

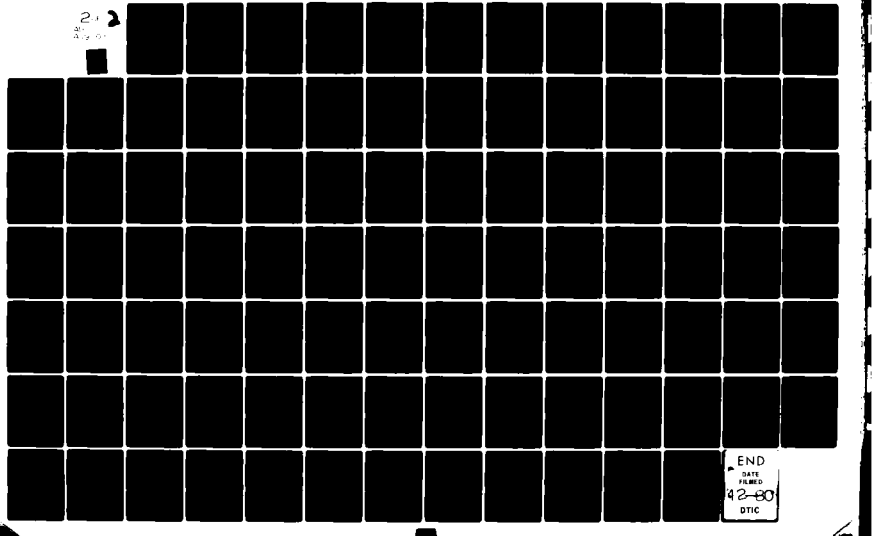
TABLE IVb

A6-E Input/Output and Steering

	C	L/S	CF	LFS	FAD	FMU	FDV	CO	SI	AT	LN	SQ
COMMAND	3	11	0	41	7	3	2	3	1	0	0	0
STEERING1	4	30	0	44	7	3	2	3	1	0	0	0
COMMAND	3	3	1	39	24	7	2	1	2	2	0	0
STEERING2	5	13	4	68	31	20	5	1	4	2	0	0
COMMAND	3	13	3	35	5	6	5	0	0	0	0	0
STEERING3	3	21	3	49	6	7	5	0	0	0	0	0
SAMPLE	0	52	1	0	0	0	0	0	0	0	0	0
INPUTS	0	56	1	0	0	0	0	0	0	0	0	0
INTERRUPT	4	20	0	4	2	0	0	0	0	0	0	0
SERVICE	9	32	0	16	7	0	0	0	0	0	0	0
STEERING	4	8	1	44	10	20	1	1	1	0	0	0
DISPLAY	5	10	1	48	12	21	1	2	1	0	0	0
DISCRETE	0	86	2	2	0	0	0	0	0	0	0	0
OUTPUTS	0	90	2	2	0	0	0	0	0	0	0	0
STEERING	2	10	0	6	1	1	0	0	0	0	0	0
KEY SEL1	5	31	3	22	1	1	0	0	0	0	0	0
STEERING	5	6	0	18	0	2	0	0	0	0	0	0
KEY SEL2	10	19	1	66	0	3	0	0	0	0	0	0
TOTALS	24	209	8	189	49	39	10	5	4	2	0	0
	41	302	15	315	64	55	13	6	6	2	0	0

AD-A091 095 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 9/2
A METHOD FOR EVALUATION OF MICROCOMPUTERS FOR TACTICAL APPLICAT--ETC(U)
JUN 80 M MASTRAKAS

UNCLASSIFIED NL



END
DATE FILMED
9-80
DTIC

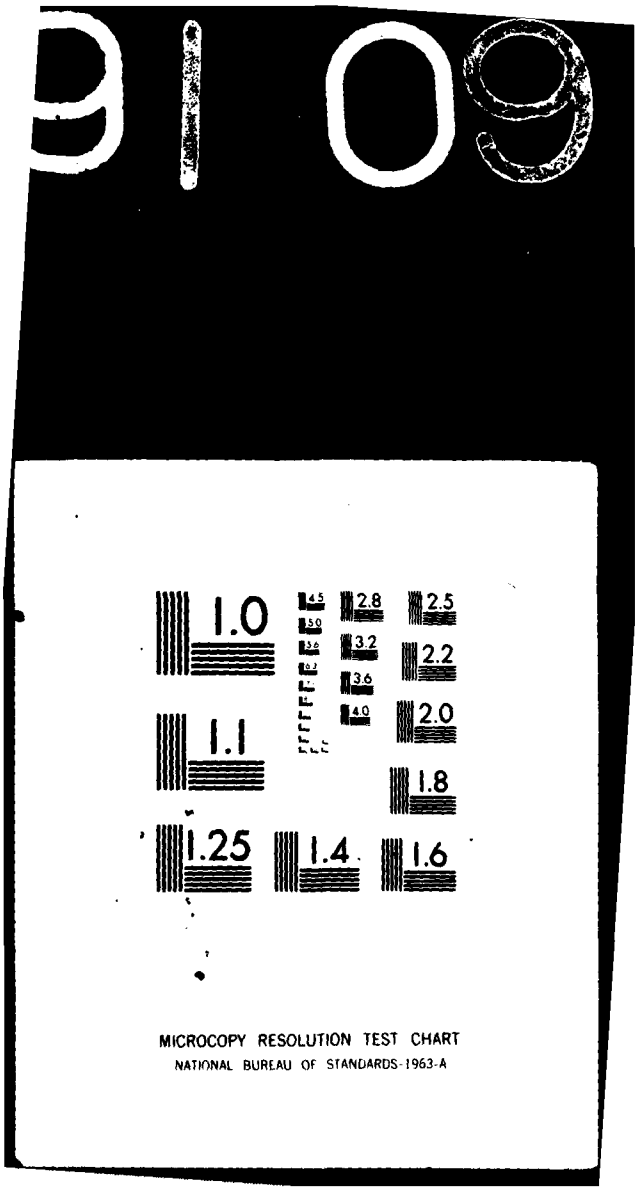

1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

TABLE IVc

A6-E Ballistics Function

	C	L/S	CF	LFS	FAD	FMU	FDV	CO	SI	AT	LN	SQ
ROCKET	0	0	4	48	17	19	5	1	1	0	0	0
ATTACK1	1	5	4	52	17	19	5	1	1	0	0	0
ROCKET	1	18	2	34	17	13	5	1	0	0	0	1
ATTACK2	1	19	2	36	17	13	5	1	0	0	0	1
BOMB	2	4	0	16	2	5	1	0	0	0	0	1
ATTACK1	3	5	0	30	5	9	1	0	0	0	0	1
BOMB	5	5	2	46	21	15	6	2	2	0	0	0
ATTACK2	5	5	4	90	23	17	6	2	2	0	0	0
BOMB	2	2	0	78	43	39	4	3	3	0	0	2
ATTACK3	2	2	1	93	43	40	4	3	3	0	0	2
BOMB	2	4	4	48	23	13	2	1	1	0	0	1
ATTACK4	3	5	6	58	23	13	2	1	1	0	0	1
BOMB	2	2	3	17	3	2	1	1	0	0	0	0
ATTACK5	3	4	6	36	3	3	2	1	0	0	0	0
BOMB	4	10	6	8	3	2	1	0	0	0	0	0
ATTACK6	6	14	9	27	6	6	1	0	0	0	0	0
BOMB	4	4	3	33	11	12	4	1	0	0	0	0
ATTACK7	11	11	4	76	18	14	5	2	0	0	0	0
COMMON	2	2	0	84	41	42	3	0	1	0	0	1
DRAG	4	6	1	85	41	42	3	0	1	0	0	1
TOTALS	24	51	24	412	181	162	32	10	8	0	0	6
	39	76	37	583	196	176	34	11	8	0	0	6

TABLE IVd

A6-E Tracking and Ranging Function

	C	L/S	CF	LFS	FAD	FMU	FDV	CO	SI	AT	LN	SQ
GREAT CIRC	0	0	0	46	10	13	4	4	5	2	0	0
NAVIGATION	0	0	0	46	10	13	4	4	5	2	0	0
TRACK RADR	3	7	5	23	6	6	0	0	0	0	0	0
TESTS	4	10	5	27	6	6	6	0	0	0	0	0
DEPR ANGLE	3	4	7	21	4	3	1	0	1	0	0	0
TRACK-1	3	16	7	21	4	3	1	0	1	0	0	0
TRACK SCAN	7	13	0	44	9	14	2	4	6	2	0	0
TESTS	9	21	0	51	10	14	2	4	6	2	0	0
DEPR ANGLE	5	9	0	11	2	2	2	0	0	0	0	0
TRACK-2	9	21	0	32	10	7	4	0	0	0	0	0
LINE OF	2	8	4	31	9	5	4	3	1	1	0	0
SIGHT RNG1	3	12	5	50	10	6	5	4	1	1	0	0
LINE OF	10	25	2	16	2	2	0	1	1	0	0	0
SITE RNG2	10	45	3	19	2	2	0	1	1	0	0	0
RADAR	7	11	0	18	7	3	3	2	1	0	0	0
RANGING	9	15	0	18	7	3	3	2	1	0	0	0
SHRIKE	4	14	4	36	11	9	3	1	1	0	0	0
RANGING	15	39	0	25	9	3	3	3	3	1	0	1
TOTALS	41	91	22	246	60	57	19	16	16	5	0	0
	62	179	25	289	68	57	28	17	18	6	0	1

TABLE IV.e

A6-E Target Updates

	C	L/S	CF	LFS	FAD	FMU	FDV	CO	SI	AT	LN	SQ
TARGET	6	6	0	120	4	0	0	0	0	0	0	0
INI	7	38	0	140	6	1	0	0	1	0	0	0
TARGET	5	10	0	57	24	14	8	0	0	0	0	0
POS FILTERS1	7	14	0	73	28	18	11	0	0	0	0	0
TARGET	2	4	0	16	4	0	0	0	0	0	0	0
POS FILTERS2	2	4	0	16	4	0	0	0	0	0	0	0
SLEW	5	24	0	8	0	2	0	0	0	0	0	0
UPDATE1	5	26	0	20	0	2	0	0	0	0	0	0
SLEW	8	12	2	49	16	10	4	1	1	0	0	0
UPDATE2	14	24	2	87	18	10	4	1	1	0	0	0
ANGLE	2	4	4	44	12	13	5	2	1	0	0	0
RATES	2	7	5	62	13	16	6	2	1	0	0	0
CURSOR	2	2	1	87	49	32	12	4	4	3	0	2
UPDATES	6	8	1	125	52	37	14	5	4	4	0	4
RADAR	1	10	3	26	0	0	0	0	0	0	0	0
OUTPUTS	1	10	3	26	0	0	0	0	0	0	0	0
TARGET	1	3	0	46	15	11	3	1	1	2	0	1
POS UPDATES	1	3	0	46	15	11	3	1	1	2	0	1
TOTALS	32	75	10	453	124	82	32	8	7	5	0	3
	45	134	11	595	136	95	38	9	8	6	0	5

TABLE IV.f

A6-E Attack Decisions

	C	L/S	CF	LFS	FAD	FMU	FDV	CO	SI	AT	LN	SQ
RESELECT	8	9	0	0	0	0	0	0	0	0	0	0
LOGIC1	9	31	0	0	0	0	0	0	0	0	0	0
RESELECT	5	11	0	0	0	0	0	0	0	0	0	0
LOGIC2	14	30	0	4	0	0	0	0	0	0	0	0
RESELECT	7	8	0	0	0	0	0	0	0	0	0	0
LOGIC3	11	25	0	3	0	0	0	0	0	0	0	0
RESELECT	10	14	0	0	0	0	0	0	0	0	0	0
LOGIC4	20	36	0	0	0	0	0	0	0	0	0	0
RESELECT	8	10	0	0	0	0	0	0	0	0	0	0
LOGIC5	16	30	0	0	0	0	0	0	0	0	0	0
ATTACK	7	11	3	15	2	2	0	0	0	0	1	0
SELECT1	8	25	3	25	6	3	1	0	0	0	1	0
ATTACK	6	23	0	22	5	5	1	1	1	0	0	0
SELECT2	10	40	0	38	7	5	1	1	1	0	0	0
STEP OUT	2	2	7	15	6	1	1	0	0	1	0	0
OF ATTK	9	19	10	18	6	1	1	0	0	1	0	0
ATTACK	5	7	10	16	4	2	0	0	0	0	0	0
VALID1	6	34	11	17	5	2	0	0	0	0	0	0
ATTACK	7	9	4	24	12	2	2	0	0	0	0	0
VALID2	10	20	5	39	18	3	2	0	0	0	0	0
TOTALS	65	104	24	92	29	12	4	1	1	1	1	0
	113	290	29	144	42	14	5	1	1	1	1	0

Tables IV(g-l) tabulate the number of FORTRAN or CMS-2 instructions (if the flowchart were translated into FORTRAN or CMS-2) in the categories of arithmetic (AR), conditional (IF), and control alteration (GO) statements. In addition, the number of assembly language instructions in the actual program and the number of bytes (8 bits) the program occupies in memory are also tabulated.

The entire functional requirements of the A6-E operational flight program are expressed by table IVm.

The purpose of all these tables IV(a-m) is to help the reader to find the components which are used in the calculations of this study for further use.

TABLE IVg

A6-E Navigation Higher level Language Complexity

	AR	IF	GO	TOTAL	ASSEMBLY	BYTES
AIR DATA 1	19	9	6	34	118	242
AIR DATA 2	20	2	1	23	87	168
AIR MASS ANGLES	15	10	2	27	124	272
DOPPLER VELOCITY	16	5	4	25	90	176
SYSTEM VELOCITY	26	4	4	34	115	232
BARO INRT VERT LOOP	25	9	6	40	127	270
INERTIAL ANGLES	16	3	2	21	78	168
PLATFORM CORRECTIONS 1	17	2	2	21	100	200
PLATFORM CORRECTS2	28	1	1	30	243	488
TOTALS	182	45	28	255	1082	2216

TABLE IV.h

A6-E Input/Output and Steering
Higher Level Language Complexity

	AR	IF	GO	TOTAL	ASSEMBLY	BYTES
COMMAND STEERING 1	33	4	4	41	109	242
COMMAND STEERING 2	33	9	7	49	188	430
COMMAND STEERING 3	28	6	5	39	99	318
SAMPLE INPUTS	52	1	1	54	272	482
INTERRUPT SERVICE	30	8	7	45	147	376
STEERING DISPLAY	23	6	4	33	127	252
DISCRETE OUTPUTS	46	2	2	50	254	552
STEERING KEY SEL 1	18	8	5	31		
STEERING KEY SEL 2	38	10	10	58	255	588
TOTALS	301	54	45	400	1451	3240

TABLE IV.i

A6-E Ballistics Function Higher Level Language Complexity

	AR	IF	GO	TOTAL	ASSEMBLY	BYTES
ROCKET ATTACK 1	26	5	3	34	113	252
ROCKET ATTACK 2	16	2	3	21	100	216
BOMB ATTACK 1	13	3	3	19	62	136
BOMB ATTACK 2	38	9	2	49	240	480
BOMB ATTACK 3	29	3	3	35	261	532
BOMB ATTACK 4	22	9	4	35	164	356
BOMB ATTACK 5	15	9	7	31	73	168
BOMB ATTACK 6	17	14	7	38	98	224
BOMB ATTACK 7	38	15	11	64	229	498
COMMON DRAG	21	5	3	29	209	434
TOTALS	235	74	46	355	1549	3296

TABLE IV.j

A6-E Tracking and Ranging Function
Higher Level Language Complexity

	AR	IF	GO	TOTAL	ASSEMBLY	BYTES
GREAT CIRC NAVIGATION	14	0	0	14	82	164
TRACK RADAR TESTS	13	9	7	29	78	174
DEPR ANGLE TRACKING 1	10	12	10	32	102	250
TRACK SCAN TESTS	21	9	5	35	157	338
DEPR ANGLE TRACKING 2	22	9	9	40	112	268
LINE OF SIGHT RANGE 1	26	8	4	38	159	322
LINE OF SIGHT RANGE 2	19	13	7	39	107	244
SHRIKE RANKING	28	13	8	49	138	300
RADAR RANGING	23	13	10	46	131	284
TOTALS	176	86	60	322	1066	2344

TABLE IV.k

A6-E Target Updates Higher Level Language Complexity

	AR	IF	GO	TOTAL	ASSEMBLY	BYTES
TARGET INITIALIZE	50	6	5	61	152	306
TARGET POS FILTERS 1	43	7	5	55	216	490
TARGET POS FILTERS 2	10	2	1	13	15	36
SLEW UPDATE 1	17	5	5	27	50	102
SLEW UPDATE 2	46	16	8	70	189	396
ANGLE RATES	27	7	3	37	136	284
CURSOR UPDATES	38	7	3	48	316	678
RADAR OUTPUTS	15	4	3	22	86	230
TARGET POS UPDATES	18	1	0	19	88	176
TOTALS	264	55	33	352	1248	2698

TABLE IV.1

A6-E Attack Decisions Higher Level Language Complexity

	AR	IF	GO	TOTAL	ASSEMBLY	BYTES
RESELECT LOGIC 1	14	10	5	29		
RESELECT LOGIC 2	24	15	12	51	The totals for these five lines appear below	
RESELECT LOGIC 3	21	11	6	38		
RESELECT LOGIC 4	14	20	8	42		
RESELECT LOGIC 5	10	16	6	32	583	1324
ATTACK SELECT 1	29	11	4	44	161	390
ATTACK SELECT 2	28	10	6	44	100	236
STEP OUT OF ATTACK	11	19	5	35	81	208
ATTACK VALID 1	25	17	12	54	145	342
ATTACK VALID 2	26	15	5	46	148	360
TOTALS	202	144	69	415	1218	2860

TABLE IV.m

Summary of A6-E Program Segments

Subprograms	Ixstructions				Variables		External	
	S	M	L	X	Int	Real	Int	Real
Navigational	51	302	225	29	18	125	3	47
Function	85	405	262	31	18	125	3	47
Tracking &	239	730	374	64	50	192	7	7
Ranging	433	954	439	71	50	192	7	7
Ballistics	234	552	420	28	20	170	18	16
Calculations	518	793	467	29	20	170	18	16
Sensor I/O &	233	198	98	11	87	103	17	16
Steering	343	330	132	14	87	103	17	16

The headings of this table are described as follows:

- S - Short instructions, 16 bit integers.
- M - Medium length instructions: load, store, and compare floating point quantities.
- L - Long floating point instructions: FAD, FMU, FDV.
- X - Subprograms which calculate sines, cosines, etc.
- INT - Number of integer variables in the program.
- REAL - Number of floating point variables in the program.
- EXT - Number of variables which are used by other programs external to the named one.

B. PERFORMANCE ESTIMATES FOR UNIPROCESSOR SYSTEMS

The attached FORTRAN program was written in order to estimate the average and the worst case performance for each of six major functions of the A6-E operational program.

The performance requirements are expressed in terms of the fundamental operations listed as columns in the tables IV(a-f). The first row of each row pair corresponds to the number of operations required in the most frequently executed path of each subfunction. The second row of the pair corresponds to the path which requires the maximum or worst case execution time. The totals at the bottom of each page accumulate the corresponding values of each subfunction to give the totals for the major functions.

The performance estimate of any computer is calculated by multiplying the execution times of the basic operations for that particular computer, with the number of times that basic operation is executed. The worst case execution time is determined for one execution of the major function by taking the totals in the bottom row of each table and multiplying these values by the corresponding execution times of the specified computer.

Each of the major functions must be executed a specified number of times per second. The repetition rate depends on the function and varies from fifty times per second (50 HZ) to five times per second (5 HZ). In order that the performance of a computer be acceptable, the worst case execution times multiplied by the repetition rates must not exceed

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                   C
C   THIS PROGRAM CALCULATES THE TOTAL TIME WHICH A GIVEN            C
C   SYSTEM NEEDS TO SOLVE THE OPERATIONAL FLIGHT PROGRAM            C
C   OF THE AIRCRAFT A6-E. GIVEN THE FUNCTIONS AND THE               C
C   NECESSARY COUNTS FOR EACH OF THE INSTRUCTIONS.                  C
C                                                                   C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C   TIME : CONTAINS TIMES OF REPETITION OF EACH INSTRUCTION
C   NAME : CONTAINS NAMES OF OPERATIONS
C   NM   : CONTAINS EXECUTION TIMES OF INDIVIDUAL INSTR.
C   RES  : TO STORE EXECUTION TIMES OF PROGRAMS
C
      REAL*4 NAME(12),RES(6,3),NM(3,12),COMP(4),TOTRES(3)
      INTEGER TIME(6,12),ANSWER
C
C   THE FOLLOWING DATA REFER TO TOTALS OF EACH INSTR.
C
      DATA TIME/28,41,35,62,45,113,57,302,76,179,134,290,
     119,15,37,25,11,25,351,215,583,285,595,144,
     1136,64,196,68,136,42,94,55,176,57,95,14,
     132,13,34,28,38,5,9,6,11,17,9,1,
     19,6,8,18,8,1,5,2,0,6,6,1,
     12,0,0,0,0,1,8,0,6,1,5,0/
C
C   COLUMN HEADINGS :
C   C   CONDITIONAL BRANCH FOR INTEGER OPERANDS
C   L/S LOAD OR STORE AN INTEGER
C   CF  CONDITIONAL BRANCH FOR FLOATING POINT OPERAND
C   LFS LOAD OR STORE A FLOATING POINT OPERAND
C   FAD FLOATING POINT ADDITION
C   FMU FLOATING POINT MULTIPLY
C   FDV FLOATING POINT DIVIDE
C   CO  COSINE FUNCTION
C   SI  SINE FUNCTION
C   AT  ARC TANGENT FUNCTION
C   LN  LOGARITHMIC FUNCTION
C   SQ  SQUARE ROOT FUNCTION
C
      DATA NAME/' C  ',' L/S',' CF ',' LFS',' FAD',' FMU',
     1' FDV',' CO ',' SI ',' AT ',' LN ',' SQ '/
C
100   CONTINUE
310   FORMAT(12A4)
      READ(5,310)COMP
      READ(5,201)((NM(I,J),I=1,3),J=1,12)
201   FORMAT(3F10.2)
22    FORMAT(///,11X,'INSTRUCTION',8X,'EXECUTION TIMES FOR',
     1//,25X,'AN/LYK-7',3X,'Z-8000',3X,'INTEL-8086',/,
     111X,'-----------------------------------------')
      WRITE(6,22)
C
C   CALCULATION OF EXECUTION TIME FOR EACH OF THE
C   FOLLOWING SIX(6) SUBPROGRAMS :
C     NAVIGATIONAL
C     I/O AND STEERING
C     BALLISTICS
C     TRACKING AND RANGING
C     TARGET UPDATES
C     ATTACK DECISIONS
C
C
C    CALCULATION OF EXCUTION TIME OF EACH SUBPROGRAM
C
      DO 401 J=1,3
401   TOTRES(J)=0.0
      DO 100 J=1,6
         DO 100 I=1,3
            RES(J,I)=0.0
      DO 102 K=1,12
102      RES(J,I)=RES(J,I)+TIME(J,K)*NM(I,K)
```

```
C
C  CALCULATION OF TOTAL EXECUTION TIME
C
100    TOTRES(I)=TOTRES(I)+RES(J,I)
       L=1
       DO 399 I=1,12
       WRITE(6,555)NAME(I),NM(1,I),NM(2,I),NM(3,I)
555    FORMAT('0',12X,A4,3X,F10.2,2(2X,F10.2))
399    CONTINUE
C
111    FORMAT('1')
       WRITE(6,111)
150    FORMAT(' ',//,' ','RESULTS(IN MICROSEC) FOR:',
      11X,'AN/UYK-7',5X,'Z-8000',4X,'INTEL 8086',/,
      1'-------------------------------------------',
      1'---------------')
       WRITE(6,150)
       DO 200 J=1,6
99     FORMAT(/,' ','PROGRAM',2X,I2,3X,'TAKES',3X,F11.2,
      12(2X,F11.2))
       WRITE(6,99)J,(RES(J,L),L=1,3)
200    CONTINUE
400    FORMAT(///,' ','TOTAL EXECUTION TIME',1X,F12.2,
      12(1X,F12.2))
       WRITE(6,400)(TOTRES(J),J=1,3)
444    FORMAT('1')
       WRITE(6,444)
       STOP
       END
$ENTRY
```

INSTRUCTION	EXECUTION TIMES FOR		
	AN/UYK-7	Z-8000	INTEL-8086
C	2.00	2.00	1.00
L/S	1.50	0.75	4.00
CF	3.00	10.00	10.00
LFS	3.00	10.00	13.00
FAD	6.25	7.50	9.00
FMU	10.00	22.50	18.00
FDV	17.00	48.75	39.00
CO	162.50	137.50	110.00
SI	162.50	137.50	110.00
AT	162.50	137.50	110.00
LN	200.00	162.50	130.00
SQ	15.00	45.00	36.00

RESULTS(IN MICROSEC) FOR:			AN/LYK-7	Z-80CC	INTEL 8086
PROGRAM	1	TAKES	7963.CC	12741.25	12771.00
PROGRAM	2	TAKES	4971.00	7884.75	5107.00
PROGRAM	3	TAKES	8792.5C	16305.CC	16856.CC
PROGRAM	4	TAKES	9483.CC	12238.25	12061.00
PROGRAM	5	TAKES	8367.50	14648.00	15552.00
PROGRAM	6	TAKES	2355.CC	3622.25	4720.00
TOTAL EXECUTION TIME			41932.CC	67439.5C	71067.CO

one second. If one second is exceeded, only a multicomputer solution is possible. The worst case execution time estimates for each major function are tabulated in table V. A uni-processor performance evaluation is shown on table VI.

TABLE V

Execution Time Estimates for Each Major Function

Function	Rep. Rate (HZ)	COMPUTER AN/UYK-7	Z-8000 Enhanced	INTEL-8086 Enhanced
Navigational	50	398150 μs	637062 μs	638550 μs
I/O and Steering	25	124275	197119	227675
Ballistics	Demand (about 30)	263775	489150	505680
Tracking and Ranging	12.5	118537	152978	150762
Target Updates	12.5	104594	183100	194400
Attack Decisions	5	11775	18111	23600
Total (in seconds)		1.021106	1.677520	1.740667

Notes: (1) Enhanced - includes a floating point chip (i.e., INTEL-8087).

(2) The above numbers represent the results of the execution times (coming from the attached FORTRAN program) multiplied by the repetition rate of each subprogram. The total execution time gives an indication if the system is acceptable (time less than 1 second) or not (in case of μC that means solution must come from multi-microcomputer system.) This case is presented in the following pages.

TABLE VI

Uni-Microprocessors Performance Evaluation

SPECIFICATIONS	Z-80	8080	MC6809	8086	Z8000	MC68000
Data word size	8 bits	8 bits	8/16 bits	16 bits	16 bits	16 bits
Address bus size	16 bits	16 bits	16 bits	20 bits	23/16 bits	23 bits[1]
Addressing range	65 Kbytes	65 Kbytes	64 Kbytes	1 Mbytes	8 Mbytes/ 64 Kbytes	16 Mbytes
Instruction word size	1 to 3 bytes	8 to 24 bits	1 to 4 bytes	1 to 6 bytes	2 to 6 bytes	2 to 10 bytes
Shortest instruction/time (in μs)	(Read reg. to reg.) 1.0	(add reg.) 2	(many) 2	(many) .40	(many) .75	(many) 0.5
Number of basic instructions	158	78	59	97	110+	56
Longest instruction/time (in μs)	(set bit at address) 5.75	(swap) 9	(interrupt) 6	(div) 37.8	(div) 90.0	N/A[2]
Clock frequency	5 KHZ/4.5 MHZ	2/3 MHZ	100 KHZ/2 MHZ	8/5 MHZ	4 MHZ	1 MHZ
Clock phases/ voltage swing	1.5 V	2/9 V	1/TTL	1/TTL	1/TTL	1/TTL
Dedicated I/O control lines	5	None	3	3	3	6
Package	40 pin DIP	40 pin DIP	40 pin DIP	40 pin DIP	40 or 48 pin DIP	64 pin DIP
Power requirements	5V/90mA	12V/40mA	5V/250mA	5V/275mA	5V/300mA	5V/300mA
Price (in $)	30	32	36	87	140	249

Notes: (1) The 24th bit is generated external to the processor. (2) Not applicable

C. PERFORMANCE ESTIMATES FOR MULTI-MICROCOMPUTER SYSTEMS

If the performance requirements exceed the processing capacity of the uniprocessor, then it is possible to implement the system by using multi-microcomputer systems. The most available architectures which are designed specifically for real-time applications are the single board computers

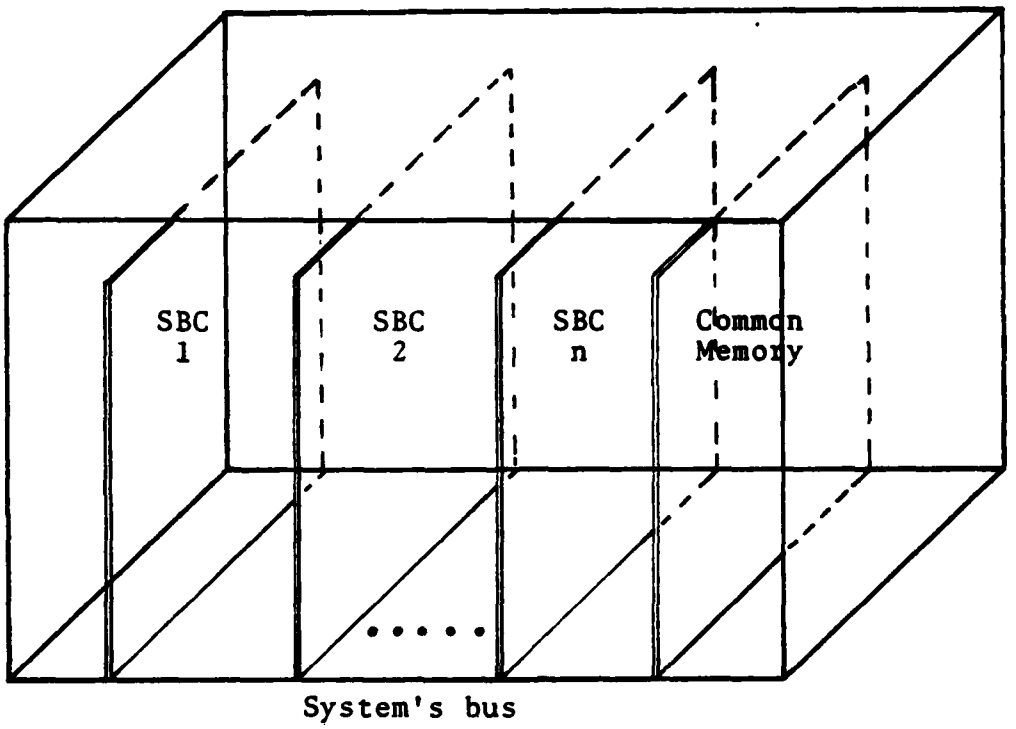


Figure 18 A typical multi-microcomputer system.

connected by a system's bus to common memory. Figure 18, above, depicts a typical architecture.

Because each Single Board Computer (SBC) contains its own memory for storing programs and private data, the system's bus is only used when external data is accessed. If the computers are dedicated to distinct functional tasks, each

computer performs independently of the others and only external data needed for computation and synchronization of the processes is communicated on the system's bus. If the functional processes are relatively independent, the performance of the system is the sum of the performances of the component parts. If, however, common data is frequently accessed by all of the functional processes, the system's bus may become the bottleneck of such a multi-microcomputer system. In order to evaluate the system's bus performance, experiments were designed and the results are summarized below.

When the system's bus is used at 50% of capacity, no observable degradation in the system's performance is observed. At 75% of bus capacity, approximately 10% loss of the system's performance was observed. The system's degradation becomes serious above 80% of capacity.

Each application must be studied individually in order to partition the system into functional elements so that a single board computer is able to satisfactorily perform the computations and communicate information needed externally to common memory.

With the case study of A6-E operational flight program, the following partition, table VII of the functional program was found satisfactory.

The results of table VII are that using three (3) single board computers in parallel (INTEL 8086/8087) [33], the percentage of the system's bus use is very low (about 4.87% of

TABLE VII

Functional Partitioning of the
A6-E Operational Flight Program

SBC #	Functions	Estimated Execution Time/Sec. For INTEL-8086		External Variables Access Time	% of Bus Use
1	Navigational	0.64 sec		29100 µs/sec	2.9%
2	° I/O Steering	0.23	0.59	10731	1.07%
	° Attack Decisions	0.02			
	° Target Update	0.19			
	° Tracking and Ranging	0.15			
3	Ballistics	0.51		9000	0.9%
	TOTAL	1.74 sec		--	4.87%

Note: The value of External Variable Access Time includes the number of words per second multiplied by the repetition rate of the function and by the execution time for two more instructions of INTEL 8086, which is 6.0 µs.

capacity) and therefore, according to the previous assumptions, this system is acceptable.

D. SUMMARY

The case study presents an implemented system, A6-E operational flight program, as a concrete example to illustrate the proposed method of computer evaluation. Whenever a system has been implemented, an accurate estimate of the computational requirements can be made. The estimates were based on the "worst case" computational requirements.

From table V, none of the uniprocessors have a computational capacity to match the requirements. A multi-computer solution, therefore, must be used. Table VII illustrates a three computer solution to the problem. Although a two computer solution would also have been possible for the *INTEL* 8086, the computational capacity of each computer would be used at 80% level. Future growth of the system would be severely limited. The three computer solution would allow less restricted future growth as well as more reliable performance. If any one computer fails, the two remaining computers can share the computational load without degradation in performance.

VI. CONCLUSIONS

This thesis provided a method to evaluate microcomputer systems for Tactical Applications, presented a Case Study to illustrate by means of a concrete example how the performance requirements are expressed in terms of parameters which are used to estimate the performance of any particular computer system, and contained important information to aid in the evaluator's work, including tables in the Appendix which list microcomputer characteristics.

The purpose of this chapter is to emphasize the main difficulties, make recommendations, and draw conclusions about the use of microcomputer systems in tactical applications.

The evaluation of microcomputer performance capacities indicates that whereas the eight (8) bit microcomputers are deficient in arithmetic capability, the newest sixteen (16) bit microcomputers enhanced by floating point arithmetic units are close rivals to the Navy standard real-time computers, AN/UYK-7 and AN/UYK-20. Because the single board computers are designed to be building blocks of multi-microcomputer systems, they provide a greater flexibility for system's expansion and reliability. The computational capacity of multi-microcomputer systems on a system's bus can be easily expanded to meet the requirements of most presently conceived real-time combat applications.

The three key barriers to rapid introduction of microprocessors into Military and industrial systems relate to:

° Concerns for life-cycle supportability (for both military and industrial applications where anticipated system life is fifteen (15) years or longer) [26].

° The belief that the microcomputer performance in military applications is not adequate.

° The microcomputer technology is not compatible with previously defined architectural standards. Because of large prior investments in software tools and applications software geared to the established hardware standards, there is reluctance to undertake costly changes.

Certain recommendations can be made regarding the use of microcomputer systems in tactical applications. These are:

° Adaption of standard architectural families, enabling the development community to partially overcome [26] the development cycle delay and take advantage of the logistics maintenance and upgrade process, wherein old circuit boards are replaced with new boards containing microprocessors. The benefits in reliability, power, weight, cost, and self-diagnosis capability could be significant.

° Initiating training programs designed to educate end-users of microcomputer systems. Microprocessors would certainly be used more widely in new designs if designers felt comfortable using them. Design aids, such as software development systems, logic analyzers, emulators, and High Order Languages will help. In addition, some stability in

languages (i.e., DOD instructions about Ada) and architectures is needed.

° Some microprocessor standardization can be beneficial and needs to be proposed, and the microprocessor industry is becoming increasingly willing to accept it.

Central to this need for standards is user concern for life-cycle supportability of industrial and military electronic systems which contain embedded microprocessors and microcomputers. Standards, such as bus interfaces and protocols, language, component interchangeability, or functional description can play a key role in overcoming these concerns, as long as the standards are intelligently applied and not unilaterally decreed.

The major conclusions produced by this work are that single board microcomputer systems, and multi-microcomputer systems are a flexible and economical solution to tactical systems' implementations.

For those users who have no prior investment, the main criteria for selection of microcomputer-based systems are:

° Performance capacity of the processor.

° Software support from various sources.

° Manufacturer's risk.

For those users with prior investment, the following conversion costs need be considered:

° Change of High Order Language, or

° Compiler conversion.

The Armed Forces can influence the microprocessor/microcomputer industries in two ways:

- By seeding money to influence basic technology directions, and
- By consolidating numerous small-quantity purchases into bulk buys, using functional standards to achieve commonality.

It is believed that the use of microcomputer systems, while considering the disadvantages which follow every new technology, will increase the effectiveness of weapon systems and will allow the Armed Forces to remain competitive or become superior to potential adversaries.

APPENDIX A

REQUIRED TIME TO SELECT A MICROCOMPUTER

From previous experience, in the U.S. Navy, the time to select a computer system (other than microcomputer) fluctuates between 10 and 24 months [10].

Why does it take such a long time?

This is a reasonable question from many people without the necessary background in this field. Nevertheless, the answer to the above question is that the selection process must solve complex problems and, as the problems become more complex, so do the tools we employ. Complexity does not necessarily mean delay, but very often it accounts for a large part of the necessary time for the solution of the problem.

In the area of microcomputers, it is believed that the selection time is between 6 and 12 months, as outlined in Table VIII.

It is necessary for the selection of a microcomputer system, as for any other equivalent equipment, to form two individual groups: the Source Selection Advisory Council (SSAC) (consisting of the procurement contracting officer, the representative of CNO, and one or two members of the user), and the Source Selection Evaluation Board (SSEB) (consisting of the Project Leader and two or three technical advisors from the user, with necessary experience).

TABLE VIII

Required Time to Select a Microcomputer

Selection Steps	Estimated time in days Minimum - Maximum
1. Draft request for proposals for approved project	30 - 60
2. Release of draft for comments	30 - 30
3. Revision of request for proposals	20 - 20
4. Response to request for proposals	40 - 80
5. Evaluation of proposals and benchmarking	30 - 90
6. Administrative time after evaluation	20 - 45
7. Installation of equipment after contract award	20 - 40
TOTAL	180 - 365

The SSAC assures adherence to the headquarters policy and requirements. It reviews and approves the solicitation document, the selection plan, and the recommendations of the SSEB [10].

Responsibility for policy and specially configured hardware and programs for tactical systems belongs to the Tactical Digital Systems Office (TADSO) of CNO.

1. The Selection Process

The selection process has the following steps:

a. Request for Proposals (RFP)

As its name implies, this form invites firms to submit proposals to satisfy the requirements. This is a

formal document, with all the necessary descriptions of requirements, contractual terms and conditions, specific regulations, and finally a technical description of the requirements. The emphasis is on "requirements": data processing to be done, reliability, type of function wanted, etc.

b. Response to the RFP

The time for response from vendors depends on the complexity of the given requirements; usually this time is from 1 to 3 months. During this period the vendors are organizing "bid teams" and the user benchmarks. At the specified date and time, the RFPs are sent to the selection office. At this point it is underlined that corrections to the user's requirements means that an extension must be given to the vendors. It must also be kept in mind that even slight changes in wording may cause a protest or lead to an interpretation disadvantageous to the user.

c. Evaluation of Proposals

At this step, the following take place.

(1) From the responses received, which must arrive on time in order to be valid, is established how many vendors are going to bid.

(2) The user starts to work on the proposals.

(3) If the proposal fails, after the first validation against technical literature, the vendor is declared nonresponsive and is notified that he will not be considered further (e.g., if the RFP requires the expansion of the main

memory and the vendor's technical justification proves that does not satisfy). By this way, the number of vendors for further evaluation is narrowed.

(4) The benchmarking, following the above step, gives an added measure for the evaluation.

d. Benchmarks

Benchmarking is only one of a large number of tools available to us for measuring the performance of machines and working out effectively which machine to get. Some tools for the assessment of computer performance are the following:

(1) Mathematical models, based on operations research or more specialized mathematical techniques (Markov model or decision theory).

(2) Simulation models.

(3) Programs designed to test specific functions.

(4) Hardware monitors.

(5) Software assessment, such as examining the operating system or using trace programs.

(6) Cost analysis, which is kept entirely separate from the technical evaluation.

e. Selection

At this point, the winning vendor is selected, after the proposals have passed the technical evaluation, benchmarking, cost analysis, and check that both parties agree to the contractual terms and conditions.

f. Contract Award

This is the final step of the selection process. The user must verify the winning vendor that the business

decision has been made, clearances and approvals have been obtained and the authorized person may award the contract.

The selection process is lengthy, costly, and difficult. However, it is necessary to follow step-by-step all the procedures described above if it is desired to avoid a failure.

APPENDIX B

CHECKLIST FOR EVALUATING MICROCOMPUTER VENDOR POLICIES

Payoff Idea

There are two sides to every coin. On one hand, selecting a microcomputer system for a specific application or job requires consideration of hardware and software performance. On the other hand, vendor support may be crucial in the form of services (both pre- and post-installation), hardware maintenance contracts, software maintenance practices, and educational programs. Buying a microcomputer based on hardware and software performance alone, without considering other vendor policies, is precarious. This paper presents pertinent categories for vendor comparison and a checklist for evaluating any vendor's offerings. These questions can also be submitted to a vendor when a competitive analysis must be performed.

Evaluation Methodology

In order for buyers to evaluate vendor policies effectively, current information must be available. This paper assembles a checklist of questions that may be used to structure the information collection process. The buyer should first read the checklist in order to become familiar with the specific information used to compare vendor policies, and then choose those questions most pertinent for the desired project or application. Potential vendors should then be selected and asked to respond, in writing, to these questions, in sufficient time.

Using this checklist properly should guarantee thorough information collection, and define the scope and cost of services available [27].

CHECKLIST QUESTIONS

The evaluator should request the following information from the vendor.

Hardware Sales

Policies regarding a manufacturer's pricing structure are one of the primary considerations made before purchasing any computer system. This section contains a series of questions pertaining to vendor sales posture, available discounts, trade-in allowances, Original Equipment Manufacturer (OEM) equipment resale, etc.

1. Do you rent, lease, or sell your equipment?
2. If you provide a lease, is it (a)
   - ° Monthly, within a 90-day cancellation?
   - ° One-year lease?
   - ° Two-to-four-year lease?
   - ° Full payout lease?
   - ° Lease/other terms?
3. Do you offer quantity discounts to end-users on your equipment?
4. What is the range of your discounts for the following quantities:

End-user %

- 3 or less _____
- 4 - 10 _____
- 11 - 25 _____
- 26 - 50 _____
- 51 - 99 _____
- 100 plus _____

5. Where is the discount based? Specify.

6. Do you charge separately for:

   - Hardware?
   - Maintenance?
   - Installation?
   - Documentation?
   - Program support?
   - Education?

7. Do you pay for the shipping costs?

7a. What are your specified delivery times

   - Immediately?
   - 30 days?
   - 60 days or less?
   - 90 days or more? Specify.

8. Do you accept trade-in of old equipment on new?

8a. If yes, what is your allowance structure? Specify.

8b. Will you sell used and/or reconditioned hardware?

8c. If yes, how is it discounted? Specify.

9. Will you sell OEM designated equipment to an end user?

9a. Is the OEM equipment normally sold under your name or that of the OEM? Specify.

Hardware Warranty

Another important consideration is the extent and breadth of the hardware warranty. The duration, the types of service included, etc., are important and useful.

1. Do you warrant your hardware?

1a. If yes, parts only, or parts and labor? Specify.

1b. If yes, for what length of time?

°30 days?

°60 days?

°Six months?

°One year or more? Specify.

2. When does the warranty begin?

°Date of shipment?

°Date of receipt?

°Other? Specify.

3. Do you provide warranty service outside of normal working hours?

3a. If yes, is it billable or non-billable?

3b. Do you bill extra for travel?

4. Do you charge for situations in which no defect is found?

5. How is attached OEM equipment warranted? Specify.

Hardware Maintenance

After the warranty period expires, contract and/or per-call maintenance must be carefully considered in order to plan for most contingencies. The availability of maintenance on foreign peripherals is also important, as are the type and cost of maintenance education that is offered.

1. Do you offer maintenance at the customer's location on:
   - °A contractual basis?
   - °An hourly basis?

1a. If yes, do you charge extra for:
   - °Weekends?
   - °Nights?
   - °Cases in which no trouble is found?
   - °Operator errors?
   - °Other? Specify.

2. Do you charge to correct a design defect?

3. Do you offer maintenance education?

3a. If yes, is it:
   - °Billable and/or non-billable?
   - °Available at the customer's location?
   - °Available at multiple locations? At how many?

Software Products

Software support is also a primary consideration, especially in terms of the cost of license fees and the extent of the license.

1. Is your firm's software copyrighted?
2. Do you sell or license your software?
3. Is there a license fee?

3a. If yes, how is it paid?

- °Monthly?
- °Yearly?
- °One-time charge?
- °Other? Specify.

4. Do you offer software discounts if a customer upgrades from one licensed product to another?
5. Is software under development ever made available to users?

5a. If yes, under what circumstances?

Software Distribution

The form in which the software is distributed is often important to the system user.

1. On what media does your firm distribute software?

- °Diskettes?
- °Other? Specify.

2. How is your operating system software delivered?

- °Executable?
- °Object?
- °Source?

3. How many copies of related software publications are provided free with the software?

Software Maintenance

Once the system is in place, it must be maintained.

1. Is software maintenance available?

1a. If yes, is it included in the software price?

1b. If it is priced separately, how is it paid?

°Monthly?

°Yearly?

°Other? Specify.

2. Is a hardware maintenance contract a prerequisite for software maintenance?

3. Does the user pay for software functional upgrades?

4. While supported by a maintenance agreement for a typical operating system, what does a user automatically receive from the firm? Specify.

5. What must a user do to receive a new version or release of a product to which he is licensed? Specify.

6. For how many months may a user be covered by a maintenance contract after a new release/version is available if he does not upgrade?

7. What types of on-site assistance/maintenance are available? Specify.

8. Who would perform this on-site software service?

°Salesman?

°Software engineer?

°Other? Specify.

9. Does your firm offer software education?

9a. If yes, how is it paid (fee, credit with license, etc.)?

APPENDIX C

AN INTERACTIVE COMPUTER GRAPHICS (ICG) SYSTEM SELECTION CHECKLIST

Once you have determined your functional requirements, you can use the following checklist to write a system specification.

RESOLUTION	I/O, display, software, overall.
ACCURACY	I/O, display, overall.
REPEATABILITY	I/O.
COLOR	I/O, display (number, shades, brightness).
SPEED	I/O, display, software, tape, disk, printer, overall.
MUST INTERFACE WITH	Other systems, software devices.
SOFTWARE	General graphics, applications, languages.
HARDWARE FUNCTIONS	All devices (specific requirements).
THROUGHPUT	Overall, units of work/units of time.
HARD COPY OUTPUT	Needs.
INPUT	Range of data sources.
PHYSICAL ENVIRONMENT	Temperature, humidity, lighting, static, dirt, noise.
DELIVERY TIME	Hardware, software (when the system will be operational).
PERSONNEL CONSTRAINTS	How many operators, with what skills will be available?
RELIABILITY	Maximum down time of the system.
BENCHMARKS	Specific tools to check the system.
COST	Be careful, THE CHEAPEST IS NOT ALWAYS THE BEST!

APPENDIX D

USEFUL TABLES

1. Tables IX, X, and XI provide performance summaries of the following three major classes of processor components respectively:

a. Microprocessor--a general purpose processor on one to three chips. It contains an arithmetic and logic unit (ALU), an address and data bus (sometimes multiplexed), a built-in instruction set, and the basic control logic.

b. Microcomputer--an all-in-one general purpose processor that is similar to the microprocessor except that the chip contains the main control program in read-only memory (ROM) and possibly a clock oscillator, some input/output (I/O) capability, and some read-write random access memory (RAM).

c. Bit Slice--an enhanced subsection of the microprocessor's ALU. Besides the slice, which is available in two and four-bit sections, other circuits must be added to provide the basic control logic.

There are, of course, other classes of μP components-dedicated controller chips that are either custom designed or preprogrammed for specific applications, i.e., calculator chips, floppy-disc controllers, etc.

2. Table XII is a directory of μPs by primary and alternate vendors.

3. Table XIII shows the vendors that are vying for a share of the bubble-memory business and each is developing its own support components. The fact is that the range of available components has grown faster than anyone expected.

4. Table XIV - Operating Systems for μC

As figure 19 illustrates, an OS is divided into layers; innermost are the essential functions that sustain the μP's very life. Then comes a group of subprograms that permit interaction with the outside world; then a number of utilities designed to fit a μP to its applications.

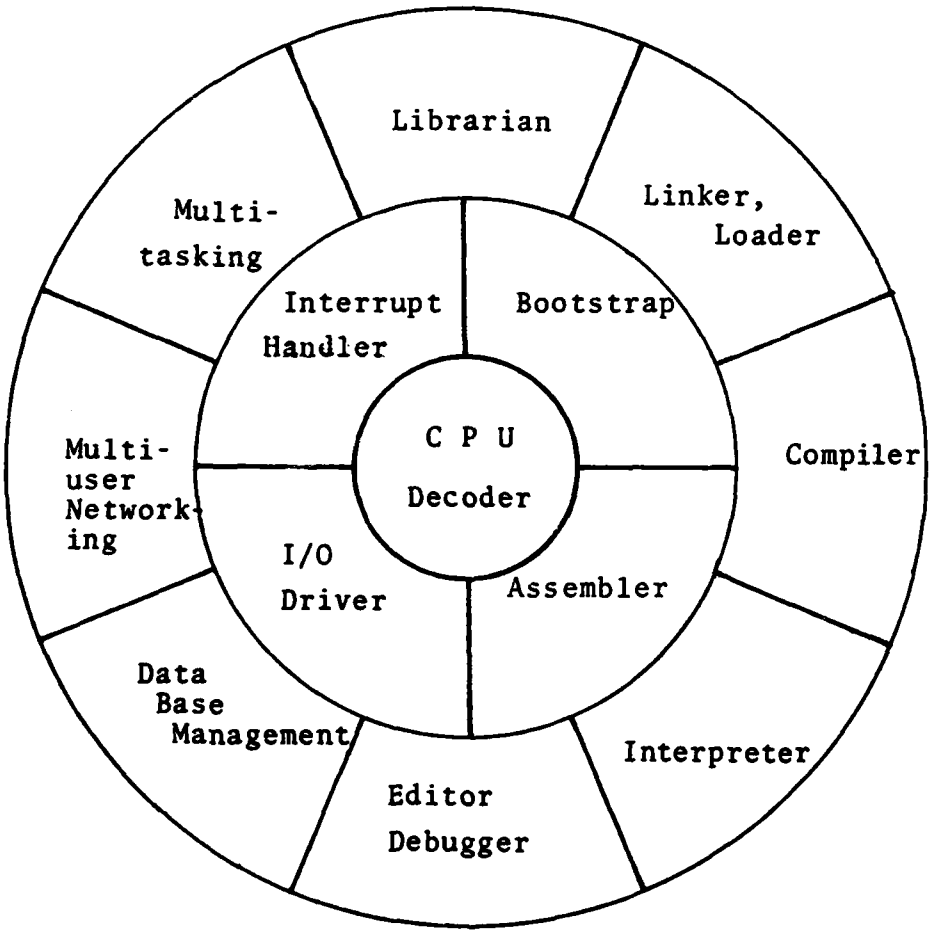


Figure 19 Layers of the operating system.

The "kernel" (1st layer)--typically on-chip ROM that governs CPU activities--is normally inaccessible to the programmer.

Utilities (2nd layer) may be on the μP chip or on separate chips, but they are normally ROM-based as well.

The bulk of most general purpose OS code falls into the outer circle (3rd layer).

Table XIV summarizes a number of commercially available operating systems, some dedicated (e.g., for development systems), some general purpose. The listings are arranged alphabetically.

5. Tables XV(a-i) present briefly the semiconductor technologies currently used in μPs.

6. Table XVI presents the μP/μC families, for a first-pass evaluation.

7. Table XVII gives, by curves, the production volumes for many μPs and μCs from 1977 up to 1979.

8. Table XVIII gives the original source μP/μC manufacturers and most of the alternate source vendors' addresses, for additional information, if needed.

TABLE IX. General-Purpose Microprocessors

Original source manufacturer	Processor	Process technology	Word size (data/instruction)	Direct addressing range (words)	Number of basic instructions	Maximum clock frequency (MHz)/phases	Instruction time shortest/longest[2] (μs)	TTL compatible	BCD arithmetic	On-chip interrupts/levels	Number of internal general-purpose registers	Number of stack registers	On-chip clock	DMA capability	Specialized memory & I/O circuits avail.	Prototyping system avail.	Package size (pins)	Voltages required (V)	Assembly language development system	High level languages	Time sharing cross software	Comments
Motorola	MC14500	CMOS	1/4	0	16	1/1	1/1	Yes	No	Yes/1	1	0	Yes	No	No[4]	No	16	3 to 18	No	No	No	Needs external program counter
Intel	4004	PMOS	4/8	4k	46	0.74/2	10.8/21.6	No	Yes	Yes/1	16	3x12	No	No	Yes	No	16	15	Yes	Yes	Yes	Superseded by 4040
Intel	4040	PMOS	4/8	8k	60	0.74/2	10.8/21.6	No	Yes	Yes/1	24	7x12	No	No	Yes	Yes	24	15	Yes	Yes	Yes	General purpose 4 bit μP
NEC Microcomputers	μPD555	PMOS	4/10	1920 X10	72	0.2/1	10/20	Yes	Yes	Yes/2	96x4	4	Yes	No	No	Yes	64	−10	Yes	No	Yes	ROM less version of μPD548
National Semiconductor	COP402	NMOS	4/8	1k	49	1/1	4	Yes	Yes	Yes/3	64x4	RAM	Yes	No	Yes	Yes	40	4.5 to 6.3	Yes	No	Yes	The 402, 402M and 404L are ROM less versions of COP420 and 440
National Semiconductor	COP402M	NMOS	4/8	1k	49	1/1	4	Yes	Yes	No	64x4	RAM	Yes	No	Yes	Yes	40	4.5 to 6.3	Yes	No	Yes	Single chip μC. All three have serial I/O and event counting capability, as well
National Semiconductor	COP404L	NMOS	4/8	2k	49	0.25/1	16	Yes	Yes	Yes/3	128x4	RAM	Yes	No	Yes	Yes	40	4.5 to 9.5	Yes	No	Yes	As 20 I/O lines
NEC Microcomputers	μPD556	PMOS	4/8	2k	80	0.44/1	4.5/9	Yes	Yes	Yes/2	96x4	3	Yes	No	No	Yes	64	−10	Yes	No	Yes	ROM less version of μPD546
Panasonic	MN1458	NMOS	4/8	1k	66	0.3/1	10/20	Yes	Yes	Yes/1	64x4	RAM	Yes	No	Yes	Yes	40	5	Yes	No	Yes	ROM less version of MN1402, but 66 instructions and come in a 40 pin package
Panasonic	MN1459	NMOS	4/8	2k	75	0.3/1	10/20	Yes	Yes	Yes/1	64x4	RAM	Yes	No	Yes	Yes	64	5	Yes	No	Yes	ROM less version of MN1400
Panasonic	MN1459A	NMOS	4/8	2k	75	0.3/1	10/20	Yes	Yes	Yes/1	128x4	RAM	Yes	No	Yes	Yes	64	5	Yes	No	Yes	ROM less version of MN1400, but 128 nibbles of on chip RAM
Panasonic	MN1599	NMOS	4/8	4k	125	1/1	2/4	Yes	Yes	Yes/4	256x4	RAM	Yes	Yes	Yes	Yes	64	5	Yes	No	Yes	ROM less version of MN1564, chip has 12 4 bit I/O ports
Fairchild	2 chip F8 (3850)	NMOS	8/8	64k	69	2/1	2/13	Yes	Yes	Yes/1	64	RAM	Yes	Yes	Yes	Yes	40	5,12	Yes	Yes	Yes	Usually used with program storage unit
General Instrument	8080	PMOS	8/8	1k	48	0.8/2	1.25/3.75	No	Yes	Yes/1	48	0	No	No	Yes	Yes	40	5,−12	No	Yes	Yes	Predecessor of F8
Intel	8008	PMOS	8/8	16k	48	0.8/2	12.5/37.5	No	Yes	Yes/1	6	7x14	No	No	Yes	Yes	18	5,−9	Yes	Yes	Yes	Predecessor of 8080, still in wide use
Intel	8035/8039	NMOS	8/8	64k	96	6/1	2.5/5	Yes	Yes	Yes/1	64	RAM	Yes	Yes	Yes	Yes	40	5	Yes	Yes	Yes	ROM less versions of 8048/8049
Intel	8080A	NMOS	8/8	64k	78	2.6/2	1.5/3.75	Yes[3]	Yes	Yes/1	8	RAM	No	Yes	Yes	Yes	40	5,12,−5	Yes	Yes	Yes	By and large, still the most popular
Intel	8085	NMOS	8/8	64k	80	5.5/1	0.8/5.2	Yes	Yes	Yes/4	8	RAM	Yes	Yes	Yes	Yes	40	5	Yes	Yes	Yes	8080 code compatible, has built in clock
MOS Technology	MCS 650X	NMOS	8/8	64k	56	4/1	0.5/3.5	Yes	Yes	Yes/1	0	RAM	Yes	No	Yes	Yes	40	5	Yes	Yes	Yes	Provides 13 addressing modes
MOS Technology	MCS 651X	NMOS	8/8	64k	56	4/2	0.5/3.5	Yes	Yes	Yes/1	0	RAM	No	No	Yes	Yes	40	5	Yes	Yes	Yes	Similar to 650X but needs 2φ clock
Mostek	3874	NMOS	8/8	64k	70+	4/1	1/6.5	Yes	Yes	Yes/4	64	RAM	Yes	No	Yes	Yes	40	5	Yes	No	Yes	ROM less piggy back version of 3870, accepts UV EPROMS on top
Motorola	6800	NMOS	8/8	64k	72	2/2	1/2.5	Yes	Yes	Yes/1	0	RAM	No	Yes	Yes	Yes	40	5	Yes	Yes	Yes	Available in depletion load version
Motorola	6802/6808	NMOS	8/8	64k	72	2/1	2/5	Yes	Yes	Yes/1	128/0	RAM	Yes	Yes	Yes	Yes	40	5	Yes	Yes	Yes	6802 has 128x8 on chip RAM, 6808 has no RAM
Motorola	6801	NMOS	8/8	64k	82	3.58/1	2/12	Yes	Yes	Yes/1	128	RAM	Yes	Yes	Yes	Yes	40	5	Yes	Yes	Yes	ROM less version of 6801 single chip μC
Motorola	6809	NMOS	8/8	64k	59	2/1	2/5	Yes	Yes	Yes/1	0	RAM	Yes	Yes	Yes	Yes	40	5	Yes	Yes	Yes	Enhanced 6800 command set
National Semiconductor	INS8060	NMOS	8/8	4k	46	4/1	5/22	Yes	Yes	Yes/1	8	RAM	Yes	Yes	No[4]	Yes	40	5	Yes	Yes	Yes	Has handy daisy chain capability
National Semiconductor	NSC800	CMOS	8/8	64k	150+	8/1	0.5/2.88	Yes	Yes	Yes/5	14	RAM	Yes	Yes	Yes	Yes	40	3 to 12	Yes	Yes	Yes	Executes Z80 instructions and has 8085 bus structure
National Semiconductor	INS8040	NMOS	8/8	64k	96	11/1	1.4/2.8	Yes	Yes	Yes/1	256	RAM	Yes	Yes	Yes	Yes	40	5	Yes	Yes	Yes	ROM less version of INS8050 single chip μC
National Semiconductor	INS8070	NMOS	8/8[7]	64k	74	4/1	3/1000[8]	Yes	No	Yes/2	9	RAM	Yes	Yes	Yes[4]	Yes	40	5	Yes	Yes	No	ROM less version of INS8072, 64 bytes of on-chip RAM
NEC Microcomputers	μPD 8080A	NMOS	8/8	64k	78	2/2	1.92/8.16	Yes[3]	Yes	Yes/1	8	RAM	No	Yes	Yes	Yes	40	5,12,−5	Yes	Yes	Yes	Pin compatible but does BCD subtraction
RCA	1802	CMOS	8/8	64k	91	6.4/1	2.5/3.75	Yes	Yes	Yes/1	16	RAM	Yes	Yes	Yes	Yes	40	3 to 12	Yes	Yes	Yes	Superseded two-chip version
RCA	8085AC	CMOS	8/8	64k	80	5.5/1	0.8/5.2	Yes	Yes	Yes/4	8	RAM	Yes	Yes	Yes	Yes	40	5	Yes	Yes	Yes	CMOS equivalent to 8085A, and pin compatible
Signetics	2650	NMOS	8/8	32k	75	2/1	1.5/6	Yes	Yes	Yes/1	7	8x15	No	Yes	Yes	Yes	40	5	Yes	Yes	Yes	There are 1.25 and 2-MHz versions

TABLE IX (Continued)

Original source manufacturer	Processor	Process technology	Word size (data/instruction)	Direct addressing range (words)	Number of basic instructions	Maximum clock frequency (MHz)/phases	Instruction time shortest/longest[2] (μs)	TTL compatible	BCD arithmetic	On-chip interrupts/levels	Number of internal general-purpose registers	Number of stack registers	On-chip clock	DMA capability	Specialized memory & I/O circuits avail	Prototyping system avail.	Package size (pins)	Voltages required (V)	Assembly language development system	High level languages	Time-sharing cross software	Comments
Signetics	8A330	Bi-polar	8/16	8k	NA[10]	4/1	0.25	Yes	No	No	8	0	No	No	No[4]	Yes	50	5	Yes	No	Yes	Intended for high-speed controllers
Zilog	Z80	NMOS	8/8	64k	150+	4/1	1/5.75	Yes	Yes	Yes/1	14	RAM	No	Yes	Yes	Yes	40	5	Yes	Yes	Yes	8080 instructions are a subset
Intersil	6100	CMOS	12/12	4k	81	4/1	2.5/5.5	Yes	No	Yes/1	0	RAM	Yes	Yes	Yes	Yes	40	4 to 11	Yes	Yes	Yes	Emulates PDP-8 instruction set
Toshiba	T3190	PMOS NMOS	12/12	4k	108	2.5/1	10/30	Yes	No	Yes/8	8	RAM	Yes	Yes	Yes	Yes	36	5,-5	Yes	Yes	Yes	Has multiply and divide inst.
Advanced Micro Devices	Am29116	ECL	16/16	64k	38+	10/1	0.1/0.2	Yes	No	No	32	32	No	Yes	Yes	Yes	48	5	Yes	No	Yes	Control-oriented microprogrammable CPU, can generate CRC bits
Data General	mN601	NMOS	16/16	32k	42	8.33/2	1.2/29.5	Yes	No	Yes/1	4	RAM	Yes	Yes	Yes	No	40	5,10,14,-4.25	Yes	Yes	Yes	Emulates NOVA instruction set
Data General	mN602	NMOS	16/16	64k	82	8.3/2	2.4/53	Yes	No	Yes/16	4[6]	RAM	No	Yes	Yes	Yes	40	3,12,±5	Yes	Yes	Yes	Executes the NOVA instruction set and addresses double the memory of the mN601
Fairchild	9440	I²L	16/16	64k[5]	42	12/1[9]	1.25/3.5	Yes	No	Yes/16	4	RAM	Yes	Yes	Yes	Yes	40	5	No	No	No	Emulates NOVA instruction set
Fairchild	9445	I²L	16/16	64k	100	20/1[9]	0.3/5.7	Yes	No	Yes/16	4	RAM	No	Yes	Yes	Yes	40	5	Yes	Yes	Yes	Executes NOVA 3 and 4 instruction sets
Ferranti	F100L	Bi-polar	16/16	32k	153	14/1	1.19/14	Yes	No	Yes/1	RAM	RAM	No	Yes	Yes	Yes	40	5,1.2	Yes	Yes	Yes	Can do double word operations
General Instrument	CP1600/1610	NMOS	16/16	64k	87	4/2	1.6/4.8	Yes	No	Yes/1	8	RAM	No	Yes	Yes	Yes	40	5,12,-3	Yes	Yes	Yes	All internal registers can be accumulators
Intel	8086	NMOS	16/16	1M[5]	97	5/1	0.4/37.8	Yes	Yes	Yes/1	8	RAM	Yes	Yes	Yes	Yes	40	5	Yes	Yes	Yes	Has 24 addressing modes
Intel	8088	NMOS	16/16[1]	64k[5]	97	5/1	0.4/37.8	Yes	Yes	Yes/1	8	RAM	Yes	Yes	Yes	Yes	40	5	Yes	Yes	Yes	8 bit bus version of 8086 microprocessor
Motorola	MC68000	NMOS	16/16	16M[5]	61	8/1	0.5/NA[10]	Yes	Yes	Yes/1	16	RAM	No	Yes	Yes	Yes	64	5	Yes	Yes	Yes	Has 32 bit wide internal structure
National Semiconductor	INS8900	NMOS	16/16	64k	45	2/1	2.5/5	Yes	Yes	Yes/6	4	10x16	No	Yes	Yes	Yes	40	5	Yes	Yes	Yes	Architecture intended for data handling
National Semiconductor	NS16008	NMOS	16/16	64k[5]	78/100+	NA[10]	NA[10]	Yes	Yes	Yes	8	RAM	No	Yes	Yes	Yes	40	5	Yes	Yes	No	8 bit bus version of dual language (8080/native) CPU, has internal 16 bit bus
National Semiconductor	NS16016	NMOS	16/16	64k	78/100+	NA[10]	NA[10]	Yes	Yes	Yes	8	RAM	No	Yes	Yes	Yes	40	5	Yes	Yes	No	Full 16 bit version, offers 8080A and native instruction sets
National Semiconductor	NS16032	NMOS	16/16	16M[5]	100+	NA[10]	NA[10]	Yes	Yes	Yes	8	RAM	No	Yes	Yes	Yes	48	5	Yes	Yes	No	Expanded 16 bit version with eight 32 bit registers, six 24 bit registers and two 16 bit registers; can address 16 Mbytes
Panafacom	MN1610	NMOS	16/16	64k	33	2/2	2/6	Yes[3]	No	Yes/3	5	RAM	No	Yes	Yes	No	48	5,12,-3	Yes	No	No	
Texas Instruments	TMS9980/9981	NMOS	16/16[1]	8k	69	4/4	3.2/49.6	Yes[3]	No	Yes/4	16	RAM	Yes	Yes	Yes	No	40	5,12,-5[11]	Yes	Yes	Yes	The 9981 requires external clock
Texas Instruments	TMS9985	AMOS	16/16[1]	32k	68	5/1	2.4/50	Yes	No	Yes/4	RAM	RAM	Yes	Yes	Yes	Yes	40	5	Yes	Yes	Yes	ROM less version of 9940, with buses
Texas Instruments	TMS/SBP9900	NMOS/ I²L	16/16	32k	69	4/4	2/31	Yes[3]	No	Yes/16	16	RAM	No	Yes	Yes	No	64	5,12,-5	Yes	Yes	Yes	Emulates 990 mini instructions
Western Digital	WD 16	NMOS	16/16	64k	116	2.5/4	2.1/720	Yes	Yes	Yes/16	6	RAM	No	Yes	Yes	Yes	40	5,±2,-5	Yes	Yes	No	Very similar to DEC LSI 11
Western Digital	Pascal Microengine	NMOS	16/16	64k	150+	3/4	2.4/300[8]	Yes	Yes	Yes/4	RAM	RAM	No	Yes	Yes	Yes	40	+12,±5	Yes	Yes	Yes	Five-chip set directly executes Pascal p-code
Zilog	Z8000	NMOS	16/16	48M[5]	110+	8/1	0.75/10	Yes	Yes	Yes/1	16	RAM	No	Yes	Yes	Yes	40/48	5	Yes	Yes	Yes	40-pin version is the Z8002; 48-pin, the Z8001

1. Has 8-bit external buses and 16-bit internal buses. 2. With maximum clock. 3. Except clock lines. 4. Standard TTL or MOS circuits will suffice. 5. Range in bytes. 6. Frame pointer too. 7. Double-precision 16-bit operations available. 8. String search. 9. Clock internally divided by 4 or 6, depending on instruction. 10. Not applicable. 11. 9980 only. (From: Electronic Design 24, Nov. 22, 1979)

TABLE X. All-In-One Processors

Original Source Manufacturer	Device	Process technology	Word size of ALU (bits/nibbles)	On-chip RAM size	On-chip ROM/PROM size (words)	On-chip battery expansion	Number of basic instructions	Maximum clock frequency (kHz)	On-chip clock	Instruction time (microsec/machine cycle)	TTL compatible	LED interface	On-chip interrupt levels	Software vectored interrupt levels	Several purpose external registers	Number of I/O lines	Additional special purpose circuits	Package size (DIP pins)	Voltages required (V)	Preferred system goal	Assembly language programming system	High level language programming system	Turn-key control software	Comments
AMI	S2000	NMOS	4/8	64×4	1024×8	No	51	1000	Yes	4.5/9	Yes	Yes	Yes/1	RAM	RAM	29	No	40	9	No	Yes	No	No	Includes display drivers and switch interface
	S2150	NMOS	4/8	64×4	1536×8	No	51	1000	Yes	4.5/9	Yes	Yes	Yes/1	RAM	RAM	29	No	40	9	No	Yes	No	No	Expanded ROM version
	S2200	NMOS	4/8	128×4	2048×8	Yes	59	1000	Yes	4.5/9	Yes	Yes	Yes/3	5	RAM	29	No	40	9	Yes	Yes	Yes	Yes	Both the 2200 and 2400 include an 8 bit a/d
	S2400	NMOS	4/8	128×4	4096×8	Yes	59	1000	Yes	4.5/9	Yes	Yes	Yes/3	5	RAM	29	No	40	9	Yes	Yes	Yes	Yes	converter and an 8 bit d/a converter. All S2000 processors also come in an "A" version that can directly drive vacuum fluorescent displays
Exan International Hitachi	SX-250	PMOS	4/8	64×4	1024×8	Yes	61	400	Yes	20/20	No	Yes	Yes/1	1	RAM	16	No	28	10 to 20	Yes	Yes	Yes	Yes	Has touch switch interface
	HMCS42	PMOS	4/10	32×4	512×10	No	74	700	Yes	10	No	Yes	No	RAM	RAM	22	No	28	−10	No	Yes	Yes	Yes	Minimal I/O version
	HMCS43/43C	PMOS/ CMOS	4/10	80×4	1024×10	No	74	700/500	Yes	10	No/Yes	Yes	Yes/2	RAM	RAM	32	No	42	−10/+5	No	Yes	Yes	Yes	Available in CMOS and PMOS versions
	HMCS44	PMOS	4/10	160×4	2048×10	No	69	700	Yes	20	No	Yes	Yes/2	RAM	RAM	31	Yes	42	−10	Yes	Yes	Yes	Yes	Easily handles display driving
	HMCS45	PMOS	4/10	160×4	2048×10	No	69	700	Yes	20	No	Yes	Yes/2	RAM	RAM	40	Yes	56	−10	Yes	Yes	Yes	Yes	Comes in 56 pin flat package
ITT Semiconductor Roboral Semiconductor	7150	PMOS	?	?	N/A[a]	?	?	25	Yes	?	?	?	?	?	?	14	Yes	4/18/24	−15	No	No	No	No	Designed for washing machines
	MMS7109	PMOS	4/8	5×32	N/A[a]	Yes	70	400	No	1220/1.5	Yes	Yes	Yes/1	4	1	11	Yes	28	9	No	No	No	No	Has scientific calculation ability
	COP410L	NMOS	4/8	32×4	512×8	Yes	40	250	Yes	16/32	Yes	Yes	No	2	RAM	16	Yes	24	4.5 to 9.5	Yes	Yes	No	Yes	All COPs processors include serial I/O and *event counting capability. Major differences* between models include the I/O arrangements
	COP411L	NMOS	4/8	32×4	512×8	Yes	40	250	Yes	16/32	Yes	Yes	No	2	RAM	15	Yes	20	4.5 to 9.5	Yes	Yes	No	Yes	
	COP420	NMOS	4/8	64×4	1024×8	Yes	49	1000	Yes	4/8	Yes	Yes	Yes/1	3	RAM	20	Yes	28	4.5 to 6.3	Yes	Yes	No	Yes	
	COP420L	NMOS	4/8	64×4	1024×8	Yes	49	250	Yes	16/32	Yes	Yes	Yes/1	3	RAM	20	Yes	28	4.5 to 9.5	Yes	Yes	No	Yes	
	COP420C	CMOS	4/8	64×4	1024×8	Yes	49	250	Yes	16/32	Yes	Yes	Yes/1	3	RAM	20	Yes	28	2.4 to 6.3	Yes	Yes	No	Yes	
	COP421	NMOS	4/8	64×4	1024×8	Yes	49	1000	Yes	4/8	Yes	Yes	No	3	RAM	16	Yes	24	4.5 to 6.3	Yes	Yes	No	Yes	
	COP421L	NMOS	4/8	64×4	1024×8	Yes	49	250	Yes	16/32	Yes	Yes	No	3	RAM	16	Yes	24	4.5 to 9.5	Yes	Yes	No	Yes	
	COP421C	CMOS	4/8	64×4	1024×8	Yes	49	250	Yes	16/32	Yes	Yes	No	3	RAM	16	Yes	24	2.4 to 6.3	Yes	Yes	No	Yes	
	COP440	NMOS	4/8	128×4	2048×8	Yes	49	1000	Yes	4/8	Yes	Yes	Yes/1	3	RAM	32	Yes	40	4.5 to 6.3	Yes	Yes	No	Yes	
	COP444L	NMOS	4/8	128×4	2048×8	Yes	49	250	Yes	16/32	Yes	Yes	Yes/1	3	RAM	20	Yes	28	4.5 to 9.5	Yes	Yes	No	Yes	
NEC Microcomputers	μPD548	PMOS	4/10	96×4	1920×10	Yes	72	200	No	10/20	Yes	Yes	Yes/2	4	RAM	35	No	42	−10	Yes	Yes	No	Yes	Well suited for POS and ECR applications
	μPD546	PMOS	4/8	96×4	2000×8	No	80	440	Yes	4.5/9	Yes	Yes	Yes/1	3	RAM	35	No	42	−10	Yes	Yes	No	Yes	TTL compatible I/O lines
	μPD553	PMOS	4/8	96×4	2000×8	No	80	440	Yes	4.5/9	Yes	Yes	Yes/1	3	RAM	35	No	42	−10	Yes	Yes	No	Yes	I/O handles −35 V vacuum fluorescent drive
	μPD650	CMOS	4/8	96×4	2000×8	No	80	440	Yes	4.5/9	Yes	Yes	Yes/1	3	RAM	35	No	42	+5	Yes	Yes	No	Yes	CMOS version of 546 (4% of the power)
	μPD547	PMOS	4/8	64×4	1000×8	Yes	58	440	Yes	4.5/9	Yes	Yes	Yes/1	1	RAM	35	No	42	−10	Yes	Yes	No	Yes	TTL compatible I/O lines
	μPD547L	PMOS	4/8	64×4	1000×8	Yes	58	180	Yes	11/22	Yes	Yes	Yes/1	1	RAM	35	No	42	−8	Yes	Yes	No	Yes	Low power version of 547 (half the current)
	μPD552	PMOS	4/8	64×4	1000×8	No	58	440	Yes	4.5/9	Yes	Yes	Yes/2	1	RAM	35	No	42	−10	Yes	Yes	No	Yes	I/O handles −35 V vacuum fluorescent drive
	μPD651	CMOS	4/8	64×4	1000×8	Yes	58	440	Yes	4.5/9	Yes	Yes	Yes/1	1	RAM	35	No	42	+5	Yes	Yes	No	Yes	CMOS version of 547 (4% of the power)
	μPD550	PMOS	4/8	32×4	640×8	No	58	440	Yes	4.5/9	Yes	Yes	Yes/1	1	RAM	21	No	28	−10	Yes	Yes	No	Yes	I/O handles −35 V vacuum fluorescent drive
	μPD554	PMOS	4/8	32×4	1000×8	No	58	440	Yes	4.5/9	Yes	Yes	Yes/1	1	RAM	21	No	28	−10	Yes	Yes	No	Yes	I/O handles −35 V vacuum fluorescent drive
	μPD652	CMOS	4/8	32×4	1000×8	No	58	440	Yes	4.5/9	Yes	Yes	Yes/1	1	RAM	21	No	42	+5	Yes	Yes	No	Yes	CMOS version of 550 (5% of the power)
	μPD551	PMOS	4/8	64×4	1000×8	Yes	58	440	Yes	4.5/9	Yes	Yes	Yes/1	2	RAM	28	No	40	−10	Yes	Yes	No	Yes	Includes a/d converter with 2% resolution, 4% accuracy
Panasonic	MN1400	NMOS	4/8	64×4	1024×8	No	75	300	Yes	10/20	Yes	Yes	Yes/1	2	RAM	30	No	40	5	No	Yes	Yes	Yes	Complete all in one controller
	MN1402	NMOS	4/8	32×4	768×8	No	57	300	Yes	10/20	Yes	Yes	Yes/1	2	RAM	19	No	28	5	No	Yes	Yes	Yes	Smaller I/O version of 1400
	MN1403	NMOS	4/8	16×4	512×8	No	50	300	Yes	10/20	Yes	No	Yes/1	2	RAM	13	Yes	18	5	Yes	Yes	No	Yes	All the processors in the MN1400 family are
	MN1404	NMOS	4/8	16×4	512×8	No	48	300	Yes	10/20	Yes	No	Yes/1	2	RAM	10	Yes	16	5	Yes	Yes	No	Yes	available in at least one other technology. The
	MN1405	CMOS	4/8	128×4	2048×8	No	75	300	Yes	10/20	Yes	No	Yes/2	2	2+RAM	34	Yes	40	5	Yes	Yes	No	Yes	18 and 16 pin versions are about the smallest
	MN1430	PMOS	4/8	64×4	1024×8	No	75	200	Yes	15/30	No	No	Yes/1	2	RAM	30	Yes	40	−15	Yes	Yes	No	Yes	μCs available, although most versions still
	MN1432	PMOS	4/8	32×4	768×8	No	57	200	Yes	15/30	No	No	Yes/1	2	RAM	19	Yes	28	−15	Yes	Yes	No	Yes	retain at least 2/3 of the instructions available
	MN1435	PMOS	4/8	128×4	2048×8	No	75	200	Yes	15/30	No	No	Yes/2	2	2+RAM	34	Yes	40	−15	Yes	Yes	No	Yes	on the 1405. The CMOS versions can also be
	MN1450	CMOS	4/8	64×4	1024×8	No	75	500	Yes	6/12	Yes	No	Yes/2	2	RAM	30	Yes	40	4.25 to 6	Yes	Yes	No	Yes	with operating voltages of up to 10 V.
	MN1453	CMOS	4/8	16×4	512×8	No	50	500	Yes	6/12	Yes	No	Yes/1	2	RAM	13	Yes	18	4.25 to 6	Yes	Yes	No	Yes	
	MN1454	CMOS	4/8	16×4	512×8	No	48	500	Yes	6/12	Yes	No	Yes/1	2	RAM	10	Yes	16	4.25 to 6	Yes	Yes	No	Yes	

TABLE X (Continued)

Original Source Manufacturer	Device	Process technology	Word size or bit (data/instr.)	On chip RAM bits	On chip ROM/PROM size (words)	Off chip memory expansion	Number of basic instructions	Maximum clock frequency (kHz)	On chip clock	Instruction time (shortest/longest) µs	TTL compatible	BCD arithmetic	On chip interrupts/no	Subroutine nesting levels	General purpose internal registers	Number of I/O lines	Additional output current drivers	Package size (DIP pins)	Voltages required (V)	Prototyping system avail.	Assemble language programming system	High level language programming system	Time sharing cross software	Comments
	MN1455	CMOS	4/8	64×4	2048×8	No	75	500	Yes	6/12	Yes	No	Yes/2	2	2+RAM	34	Yes	42	4.25 to 6	Yes	Yes	No	Yes	
	MN1542	NMOS	4/8	152×4	2048×8	Yes	124	1000	Yes	2/4	Yes	Yes	Yes/4	16	4+RAM	24	Yes	40	5	Yes	Yes	No	Yes	All MN1500 family processors include an 8 bit counter/timer and an 8 bit serial shift register. All I/O lines are bidirectional and the chips also have a power down mode to minimize power dissipation
	MN1544	NMOS	4/8	256×4	4096×8	Yes	124	1000	Yes	2/4	Yes	Yes	Yes/4	16	4+RAM	24	Yes	40	5	Yes	Yes	No	Yes	
	MN1562	NMOS	4/8	152×4	2048×8	Yes	124	1000	Yes	2/4	Yes	Yes	Yes/4	16	4+RAM	48	Yes	64	5	Yes	Yes	No	Yes	
	MN1564	NMOS	4/8	256×4	4096×8	Yes	124	1000	Yes	2/4	Yes	Yes	Yes/4	16	4+RAM	48	Yes	64	5	Yes	Yes	No	Yes	
Rockwell	PPS-4	PMOS	4/8	0	0	Yes	50	200/400 Two clocks	No	5/15	No	Yes	Yes/1	2	1	12+	Yes	42	−17/+5, −12	Yes	Yes	No	Yes	Combination ROM/RAM/I/O available
	PPS 4/2	PMOS	4/8	0	0	Yes	50	200/400	Yes	5/15	No	Yes	No	2	1	12+	Yes	42	−17/+5, −12	Yes	Yes	No	Yes	Same as PPS 4 but has internal clk
PPS-4/1	MM77/MM77L	PMOS	4/8	96×4	1344×8	RAM only	50	100/4	Yes	10/40	Yes	Yes	Yes/1	2	2+RAM	31	Yes	42	−15/+5, −10	Yes	Yes	No	Yes	I/O includes serial channel
	MM78/MM78L	PMOS	4/8	128×4	2048×8	RAM only	50	100/4	Yes	10/40	Yes	Yes	Yes/1	2	2+RAM	31	Yes	42	−15/+5, −10	Yes	Yes	No	Yes	Software compatible with 77
	MM76	PMOS	4/8	48×4	640×8	RAM only	50	100/4	Yes	10/40	Yes	Yes	Yes/1	1	1+RAM	31	Yes	42	−15/+5, −10	Yes	Yes	No	Yes	Primarily used for keyboard display
	MM76C	PMOS	4/8	48×4	640×8	RAM only	50	100/4	Yes	10/30	Yes	Yes	Yes/1	1	1+RAM	39	Yes	52	−15/+5, −10	Yes	Yes	No	Yes	Has high speed counter
	MM76E	PMOS	4/8	48×4	1024×8	RAM only	50	100/4	Yes	10/30	Yes	Yes	Yes/1	1	1+RAM	31	Yes	42	−15/+5, −10	Yes	Yes	No	Yes	Larger ROM than MM76
	MM76L	PMOS	4/8	48×4	640×8	RAM only	50	100/4	Yes	10/30	Yes	Yes	Yes/1	1	1+RAM	31	Yes	40	6 to 11	Yes	Yes	No	Yes	Low power version of 76
	MM76EL	PMOS	4/8	48×4	640×8	RAM only	50	100/4	Yes	10/30	Yes	Yes	Yes/1	1	1+RAM	31	Yes	42	−15/+5, −10	Yes	Yes	No	Yes	Expanded ROM version
	MM75	PMOS	4/8	48×4	670×8	RAM only	50	100/4	Yes	10/40	Yes	Yes	Yes/1	1	1+RAM	22	Yes	28	−15/+5, −10	Yes	Yes	No	Yes	Reduced I/O version of 76
Texas Instruments	TMS 1000	PMOS	4/8	64×4	1024×8	No	43	400	Yes	15/15	Yes	Yes	No	1	2+RAM	23/25	Yes	28/40	9 or 15	Yes	Yes	No	Yes	Aside from two supply versions, a 35 V vacuum fluorescent drive version is available (TMS 1070/1270)
	TMS 1000C	CMOS	4/8	64×4	1024×8	No	43	1000	Yes	6/6	Yes	Yes	No	3	2+RAM	22/32	Yes	28/40	3 to 6	Yes	Yes	No	Yes	CMOS version of TMS-1000
	TMS 1100	PMOS	4/8	128×4	2048×8	No	40	400	Yes	15/15	Yes	Yes	No	1	2+RAM	23/28	Yes	28/40	9 or 15	Yes	Yes	No	Yes	Also has VF drive versions (TMS 1170/1370) and is pin compatible with TMS-1000
	TMS 1018	PMOS	4/8	64×4	1024×8	No	43	400	Yes	15/15	Yes	Yes	No	N/A[1]	N/A[1]	N/A[1]	N/A[1]	28	15	N/A[1]	N/A[1]	N/A[1]	N/A[1]	Dedicated number cruncher
	TMS1022	PMOS	4/8	64×4	2048×8	No	43	400	Yes	15/15	Yes	Yes	No	N/A[1]	N/A[1]	N/A[1]	N/A[1]	28	15	N/A[1]	N/A[1]	N/A[1]	N/A[1]	Dedicated CB PLL controller
	TMS1117	PMOS	4/8	128×4	2048×8	No	43	400	Yes	15/15	Yes	Yes	No	N/A[1]	N/A[1]	N/A[1]	N/A[1]	28	15	N/A[1]	N/A[1]	N/A[1]	N/A[1]	Dedicated microwave oven controller
	TMS1121	PMOS	4/8	128×4	2048×8	No	42	400	Yes	15/15	Yes	Yes	No	N/A[1]	N/A[1]	N/A[1]	N/A[1]	40	15	N/A[1]	N/A[1]	N/A[1]	N/A[1]	Dedicated appliance timer/controller
	TMS1400	PMOS	4/8	128×4	4096×8	No	41	550	Yes	11/11	Yes	Yes	No	3	2+RAM	19	Yes	28	9	Yes	Yes	No	Yes	VF drive version available (1470) and 15 V supply model coming
	TMS1600	PMOS	4/8	64×4	512×8	No	41	550	Yes	11/11	Yes	Yes	No	3	2+RAM	32	Yes	40	9	Yes	Yes	No	Yes	VF drive version available (1670) and 15 V supply model coming
	TMS1700	PMOS	4/8	64×4	512×8	No	43	400	Yes	15/15	Yes	Yes	No	3	2+RAM	21	Yes	28	9 or 15	Yes	Yes	No	Yes	Reduced ROM and I/O version of TMS 1000
Toshiba	T3444	NMOS	4/8	16×8	256×24	Yes	3	800	Yes	3	Yes	3	Yes/1	8	RAM	16	No	40	5	Yes	Yes	Yes	Yes	Intended for dedicated controllers
	T3472	NMOS	4/8	16×4	256×24	Yes	67	1000	Yes	33/360	Yes	Yes	Yes/2	8	RAM	16	Yes	42	5	Yes	Yes	Yes	Yes	Designed for keyboard/display interfacing
Western Digital	1872	PMOS	4/10	32×4	512×10	No	37	150	Yes	6.25/12.5	Yes	Yes	Yes/1	1	RAM	27	No	40	12	Yes	Yes	Yes	Yes	RAM holds BCD numbers
Fairchild	F38E70	NMOS	8/8	64×8	2048×8	Yes	70+	4000	Yes	1/6.5	Yes	Yes	Yes/4	RAM	RAM	32	Yes	40	5	Yes	Yes	Yes	Yes	On chip UV EPROM instead of mask ROM version of 3870
	F3878	NMOS	8/8	64×8	4096×8	Yes	70+	4000	Yes	1/6.5	Yes	Yes	Yes/4	RAM	RAM	32	Yes	40	5	Yes	Yes	Yes	Yes	Similar to Mostek 3872 but has no standby RAM
General Instrument	PIC1645	NMOS	8/12	24×8	256×12	No	30	1000	Yes	4/8	Yes	Yes	Yes/1	2	RAM	4	Yes	24	5	Yes	Yes	Yes	No	Minimal I/O family members
	1650	NMOS	8/12	32×8	512×12	No	30	1000	Yes	4/8	Yes	Yes	Yes/1	2	RAM	32	Yes	40	5	Yes	Yes	Yes	No	32 programmable I/O lines
	1655	NMOS	8/12	32×8	512×12	No	30	1000	Yes	4/8	Yes	Yes	Yes/1	2	RAM	8	Yes	28	5	Yes	Yes	Yes	No	Reduced I/O version of 1650
	1670	NMOS	8/12	48×8	1024×12	No	30	1000	Yes	4/8	Yes	Yes	Yes/1	2	RAM	32	Yes	40	5	Yes	Yes	Yes	No	Increased memory version
Intel	8021	NMOS	8/8	64×8	1024×8	No	70	3000	Yes	10/20	Yes	Yes	Yes/1	RAM	RAM	21	No	28	5	Yes	Yes	Yes	Yes	Minimal I/O CPU version
	8022	NMOS	8/8	64×8	2048×8	No	70	3000	Yes	10/20	Yes	Yes	Yes/1	RAM	RAM	27	No	40	5	Yes	Yes	Yes	Yes	Contains two a/d converter channels
	8041/8741	NMOS	8/8	64×8	1024×8	Yes	90	6000	Yes	2.5/5	Yes	Yes	Yes/1	RAM	RAM	18	Yes	40	5	Yes	Yes	Yes	Yes	8041 has a ROM and 8741 a UV EPROM
	8048/8748	NMOS	8/8	64×8	1024×8	Yes	96	6000	Yes	2.5/5	Yes	Yes	Yes/1	8	RAM	27	Yes	40	5	Yes	Yes	Yes	Yes	8748 has UV EPROM
	8049	NMOS	8/8	128×8	2048×8	Yes	96	11000	Yes	1.4/2.8	Yes	Yes	Yes/1	8	RAM	27	Yes	40	5	Yes	Yes	Yes	Yes	Enlarged memory version of 8048
Intersil	87C41	CMOS	8/8	64×8	1024×8	Yes	90	6000 (5V)	Yes	2.5/5	Yes	Yes	Yes/1	8	16+RAM	18	Yes	40	5 to 10	Yes	Yes	Yes	Yes	CMOS equivalent to Intel 8041, dissipates 50 mW
	80C48/87C48	CMOS	8/8	64×8	1024×8	Yes	96	6000 (5V)	Yes	2.5/5	Yes	Yes	Yes/1	8	RAM	27	Yes	40	5 to 10	Yes	Yes	Yes	Yes	CMOS equivalent to Intel 8048/8748, dissipates 50 mW
Mostek	3870	NMOS	8/8	64×8	2048×8	Yes	70+	4000	Yes	1/6.5	Yes	Yes	Yes/4	RAM	RAM	32	Yes	40	5	Yes	Yes	Yes	Yes	Has 16 bit prog timer
	3872	NMOS	8/8	128×8	4096×8	Yes	70+	4000	Yes	1/6.5	Yes	Yes	Yes/4	RAM	RAM	32	Yes	40	5	Yes	Yes	Yes	Yes	Double the memory capacity of 3870.

TABLE X (Continued)

Original Source Manufacturer	Device	Process technology	Word size in bits (data and I)	On-chip RAM bytes	On-chip ROM/PROM bytes (words)	Off-chip memory expansion	Number of basic instructions	Maximum clock frequency (MHz)	On-chip clock	Instruction time (cycles/usec) at	TTL compatible	BCD arithmetic	On-chip interrupts/levels	Subroutine nesting levels	General purpose internal registers	Number of I/O lines	Additional special support controls	Package size (DIP pins)	Voltages required (V)	Prototyping system avail	Assembly language programming system	High level language programming system	Time sharing, cross software	Comments
Motorola	3873	NMOS	8 8	64×8	2048×8	Yes	70+	4000	Yes	1/6.5	Yes	Yes	Yes/4	RAM	RAM	32	Yes	40	5	Yes	Yes	Yes	Yes	Some I/O lines dedicated as serial port
	3876	NMOS	8/8	128×8	2048×8	Yes	70+	4000	Yes	1/6.5	Yes	Yes	Yes/4	RAM	RAM	32	Yes	40	5	Yes	Yes	Yes	Yes	Same as 3870 but double the RAM
	6801/68701	NMOS	8/8	128×8	2048×8	Yes	82	3500	Yes	2/12	Yes	Yes	Yes/1	RAM	RAM	31	Yes	40	5	Yes	Yes	Yes	Yes	6801 has masked ROM. 701 has UV EPROM
	6805/705	NMOS	8/8	64×8	1100×8	Yes	61	3500	Yes	2/4	Yes	Yes	Yes/1	0	RAM	20	Yes	28	5	Yes	Yes	Yes	Yes	Low cost 8 bit μC has 8 bit timer and prescaler. 705 version has UV EPROM
	6805R2	NMOS	8/8	64×8	2048×8	Yes	61	3500	Yes	2/4	Yes	Yes	Yes/1	0	RAM	20	Yes	28	5	Yes	Yes	Yes	Yes	CPU includes 8 bit a/d converter (available in mid 1980)
	146805	CMOS	8/8	64×8	1100×8	Yes	61	3500	Yes	2/4	Yes	Yes	Yes/1	0	RAM	20	Yes	28	5	Yes	Yes	Yes	Yes	CMOS version of the 6805
National Semiconductor	INS8050	NMOS	8/8	256×8	4096×8	Yes	96	11000	Yes	1.4/2.8	Yes	Yes	Yes/1	8	RAM	27	Yes	40	5	Yes	Yes	Yes	Yes	Enlarged proprietary version of Intel 8049 processor with transparent improvements
	INS8072	NMOS	8/8	64×8	2560×8	Yes	74	4h	Yes	3/1000	Yes	Yes	Yes/2	RAM	RAM	0	Yes	40	5	Yes	Yes	Yes	Yes	Three state data/address bus for user selectable I/O
RCA	CDP1834	CMOS/SOS	8/8	64×8	2048×8	Yes	113	8000	Yes	2/3	Yes	Yes	Yes/1	RAM	RAM	13	Yes	40	5 to 10	Yes	Yes	Yes	Yes	Compatible with 1802 software
Rockwell	PPS-8	PMOS	8/8	0	0	Yes	100	256/4	No	4/12	No	Yes	Yes/3	16	2	35	Yes	42	−12/+5,−12	Yes	Yes	No	Yes	Combination RAM.ROM/I/O support
	PPS 8/2	PMOS	8/8	0	0	Yes	100	200/4	No	5/15	No	Yes	Yes/3	16	2	15	Yes	42	−12/+5,−12	Yes	Yes	No	Yes	I/O chip includes clock
	65500/1	NMOS	8/8	64×8	2048×8	Yes	56	2000	Yes	1/3.5	Yes	Yes	Yes/1	RAM	RAM	32	Yes	40	5	Yes	Yes	Yes	Yes	Single chip version 6502
Zilog	Z8	NMOS	8/8	144×8	2048×8	Yes	47	8000	Yes	1.5/3.75	Yes	Yes	Yes/6	RAM	RAM	32	Yes	40	5	Yes	Yes	Yes	Yes	Has two counter/timers and UART
Texas Instruments	TMS 9940E/ 9940M	NMOS	②	128×8	2048×8	No	68	5000	Yes	2/452	Yes	Yes	Yes/4	64	RAM	16	No	40	5	Yes	Yes	Yes	Yes	Two versions available, one has a 2 k EPROM, the other 2 k ROM
Intel	2920	NMOS	25/25	40×25	192×24	No	21	2500	Yes	8.4/8.4	Yes	N/A[1]	N/A[1]	0	RAM	12	No	28	5,−5	Yes	Yes	No	Yes	Analog processor accepts four analog inputs and delivers up to eight digitally processed analog outputs

1. Not applicable. 2. External 8 bits, internally 16 bits. 3. User defined. 4. Not available.

(From: Electronic Design, 24, Nov. 22, 1979)

TABLE XI

Bit-Slice Families

Original Source Company	Series	Process technology	ALU part number	ALU word size (bits)	Number of ALU instructions	Can ALU do BCD arithmetic?	Maximum ALU clock rate (MHz)	General-purpose registers in ALU	ALU package size (DIP pins)	Microprogram sequencer number	Number of address bits	Maximum sequencer clock rate (MHz)	Number of sequencer commands	Sequencer stack size	Sequence package size (DIP pins)	Are parts TTL-compatible?	Voltages required (V)	Prototyping system available	Development software available	Specialized support circuits available	Comments
Advanced Micro Devices	2900	STTL	2901A	4	16	No	16.67	16	40	2909/11	4	10	12	4×4	28/20	Yes	5	Yes	Yes	Yes	Has widest number of second sources
		STTL	2903	4	25	No	10	16	48	2910	12	10	16	5×12	40	Yes	5	Yes	Yes	Yes	ALU has nine more instructions than 2901, including multiply and divide.
Fairchild	Macrologic	STTL/CMOS	9405/34705	4	64	No	10	8	24	9406	4	10	4	16×4	24	Yes	5	Yes	Yes	Yes	CMOS version (34705) operates at 2 MHz.
	F100220	ECL	F100220	8	27	Yes	50	1	68[1]	F100224	N/A[2]	N/A[2]	N/A[2]	N/A[2]	N/A[2]	N/A[2]	−4.2 to −5.7	1979	1979	1980	Only 8 bit slice, sub-ns instructions
Motorola	10800	ECL	10800	4	100+	Yes	20	0	48	10801	4	20	16	4×4	48	No	−2, −5.2	Yes	Yes	Yes	Fastest 4-bit slice available
Signetics	3000	STTL	3002	2	40	No	10	11	28	3001	9	10+	11	0	40	Yes	5	Yes	Yes	Yes	Only 2-bit ALU available
Texas Instruments	SBP-0400A	I²L	SBP-0400	4	512	No	5	10	40	74S482	4	20	64	4×4	20	Yes	Current	Yes	No	No	Has pipeline register
	SBP-0401A	I²L	SBP-0401	4	512	No	5	10	40	74S482	4	20	64	4×4	20	Yes	Current	Yes	No	No	Does not have pipeline register
	54/74S481	STTL or LSTTL	S481 LS481	4	24,780	No	10	0	48	74S482	4	20	64	4×4	20	Yes	5	Yes	No	Yes	Very flexible instruction set

1. Leadless carrier. 2. Not available. (From: Electronic Design 24, Nov. 22, 1979)

TABLE XII

Processor Alternate Source Directory

Generic type number	Data word size (bits)	Technology	Original source	Alternate sources
1600, 1610	16	NMOS	General Instrument	EM&M Semiconductor
1650, 1655	8	NMOS	General Instrument	EM&M Semiconductor
1802, 1803	8	CMOS	RCA	Hughes** & Solid State Scientific
1872	4	PMOS	Western Digital	None
2650	8	NMOS	Signetics	Advanced Memory Systems National Semiconductor
2900	4	STTL	Advanced Micro Devices	Fairchild, Monolithic Memories, Motorola, National Semiconductor, Raytheon, Sescosem, Signetics
3000	2	STTL	Intel	Signetics
3850	8	NMOS	Fairchild	Mostek,** Motorola **
3859	8	NMOS	Fairchild	Discontinued
3870	8	NMOS	Mostek	Fairchild, Motorola
4040/4004	4	PMOS	Intel	National Semiconductor
5701/6701	4	STTL	Monolithic Memories	ITT Semiconductor
6100	12	CMOS	Intersil	Harris Semiconductor **
6400	8	NMOS	Motorola	None
6500	8	NMOS	MOS Technology	Rockwell, Synertek
6800, 68A00, 68B00	8	NMOS	Motorola	American Microsystems,** Fairchild, Fujitsu, Hitachi, Sescosem
68000	16	NMOS	Motorola	AMI,** Fairchild
6801	8	NMOS	Motorola	None*
6802	8	NMOS	Motorola	None
6809	8	NMOS	Motorola	None*
7150	4	PMOS	ITT Semiconductor	None
8000	8	PMOS	General Instrument	AEG, SGS-ATEs
8008	8	PMOS	Intel	None
8035, 8048, 8748	8	NMOS	Intel	Advanced Micro Devices,** RCA, Intersil, NEC,** Signetics,** NEC **
8041, 8741	8	NMOS	Intel	None
8080A	8	NMOS	Intel	Advanced Micro Devices, NEC, National Semiconductor, Signetics, Texas Instruments
8085	8	NMOS	Intel	Advanced Micro Devices,** NEC, RCA
8086	16	NMOS	Intel	None
8900	16	PMOS	National Semiconductor	Discontinued
9002	8	NMOS	Electronic Arrays	Actually an alternate source for 8080
9080	8	NMOS	Advanced Micro Devices	Signetics
9405, 34705	4	STTL/CMOS	Fairchild (Macrologic)	None
9440	16	I^2L	Fairchild	

TABLE XII (Continued)

Generic type number	Data word size (bits)	Technology	Original source	Alternate sources
9900	16	I^2L/NMOS	Texas Instruments	American Microsystems** (NMOS version), SMC**
9940	16	NMOS	Texas Instruments	None*
9980	16	NMOS	Texas Instruments	None
10800	4	ECL	Motorola	None
14500	1	CMOS	Motorola	None
8X300	8	STTL	Signetics	None
74S481	4	STTL	Texas Instruments	None
100 K	8	ECL	Fairchild	None*
COPS	4	PMOS	National Semiconductor	None
F8	8	NMOS	Fairchild	See 3850
F100L	16	Bipolar	Ferranti	None
IMP-4, 8, 16	4, 8, 16	PMOS	National Semiconductor	Rockwell
Macrologic	4	STTL/CMOS	Fairchild	See 9405, 34705
MCP-1600/WD-16	16	NMOS	Western Digital	None
Micromachine	8	NMOS	Fairchild	See 3859
mN601	16	NMOS	Data General	None
MN1400	4	NMOS	Panasonic	None
MN1610	16	NMOS	Panafacom	None
PACE	16	PMOS	National Semiconductor	Rockwell
PPS-4	4	PMOS	Rockwell	National Semiconductor
PPS-4/1	4	PMOS	Rockwell	None
PPS-4/2	4	PMOS	Rockwell	None
PPS-8	8	PMOS	Rockwell	National Semiconductor
PPS-8/2	8	PMOS	Rockwell	None
S2000	4	NMOS	American Microsystems	None
SBA	1	NMOS	General Instrument	None
SBP0400A/ 0401A	4	I^2L	Texas Instruments	None
SC/MP, SC/MPII	8	NMOS/PMOS	National Semiconductor	Rockwell, Signetics, Western Digital
SMS-300	8	STTL	Scientific Microsystems	Signetics
SX200	4	PMOS	Essex International	None
T3190	12	PMOS, NMOS	Toshiba	None
T3444	4	NMOS	Toshiba	None
T3472	4	NMOS	Toshiba	None
TMS1000, 1100, 1200, 1300	4	CMOS, NMOS or PMOS	Texas Instruments	Motorola (for CMOS version)
uCOM 42	4	PMOS	NEC	None
uCOM 43, 44, 45	4	PMOS	NEC	None
Z8	8	NMOS	Zilog	None**
Z80	8	NMOS	Zilog	Mostek, Sharp, NEC, SGS
Z8000	16	NMOS	Zilog	Rockwell**

* This product is still in development. ** Licensed.

(From: "Microprocessor Data Manual," D. Bursky, 1978)

TABLE XIII. Bubble-Memory Devices

Manufacturer	Model No.	Net capacity	Organization	Package	Sample availability	Sample cost	Special support ICs
Fujitsu	FBM0102	64 kbits	Major/minor	18-pin	Stock	$100[3]	In development
	FBM0201	64 kbits	Serial/parallel single-loop	18-pin	Stock	$100[3]	In development
	FBM0301	256 kbits	Major/minor block-replicate	16-pin	Stock	$500	In development
Hitachi	H3M	64 kbits	No information available				
	H9M	256 kbits					
	H8M	1 Mbit					
Intel	7110	1 Mbit	Major/minor block-replicate	Leadless	30 days	$2000	Yes
Motorola[1]	R3M256	256 kbits	Major/minor block-replicate	18-pin	Late 1980	—	In development
National Semiconductor	N9M2256	256 kbits	Major/minor block-replicate	16-pin	Early 1980	—	In development Early 1980
Plessey	PB064/S1	64 kbits	Serial shift register (no loop)	12-pin	Stock	—[4]	In development
	PB256	256 kbits	Major/minor block-replicate	18-pin	Mid-1980	—	In development
Rockwell	R3M256	256 kbits	Major/minor block-replicate	18-pin	Stock	$500	In development
		1 Mbit	Major/minor block-replicate	18-pin	Late 1979	—	In development
Siemens	RBM256	256 kbits	Major/minor block-replicate	18-pin	Late 1980	—	In development
Texas Instruments	TIB0203	92 kbits	Major/minor	14-pin	Stock	$100	Yes
	TIB0303	254 kbits	Major/minor block-replicate	18-pin[2]	Stock	$500	Yes
	TIB0250	256 kbits	Major/minor block-replicate	24-pin	2nd qtr 1980	—	2nd qtr 1980
	TIB0500	512 kbits	Major/minor block-replicate	24-pin	4th qtr 1979	$2100[5]	2nd qtr 1980
	TIB1000	1 Mbit (512 k × 2)	Major/minor block-replicate	24-pin	4th qtr 1979	$3100[5]	2nd qtr 1980

1. Alternate source for Rockwell ROM256
2. 70-mil centers (an 18-pin, 100-mil version is also available)
3. 100-unit quantities.
4. Only board-level products will be available ($2997, 64 Kbytes, SBC-80 compatible)
5. Board-level evaluation subsystems with support IC's.

(From: Electronic Design, 24, Nov. 22, 1979)

TABLE XIV

Operating Systems for Microcomputers

Company	OS model	Target μCo Model(s)	μPs used	Mod.e	Number of simultaneous users / μPs	Multi-tasking	Net-working	Real-time	RAM (max)	Data-base	Supported Languages: Assemblers	Interpreters	Compilers	Supported Peripherals	Comments	Notes
AMI, 3800 Homestead Rd, Santa Clara, CA 95051	TDOS	6800 boards	6800	F, I	1	N/A	No	No	No	56 kb	2 Mb	6800, 2000	Basic		CRT, line printer, RS232	UCSD compatible	
	Pascal	Convac, TI 990/4, TEK 8002	6800, 9900, Z80	F	1	N/A	No	No	No	56 kb	2 Mb	6800, 01, 05, 09, 9900, 9940 2,000, 2200	Basic	Pascal			
Altek Corp., 2150 Industrial Pkwy., Silver Spring, MD 20904	AMOS[1]	MCZ family with R10	Z-80	M[2]	[3]	1	No	No	No	64 kb	N/A	[3]		[3]	Tape drive[2] 800 BPI NRZI, 1600 BPI PE	Full DMA EBCDIC coding	[1] File system similar to ZDOS [2] 9-track [3] 6 drives max
Cascade Data 6300 28th St., S.E. Grand Rapids, MI 49508	Concept-OS	Concept-IV	Z80	F[1]	1	N/A	No	1	Yes	64 kb	5 Mb (4 drives)	Z80 macro	Basic	Cobol	Intelligent CRT (25 Lines x 80) 2 ser. channels 1 par prtr port	BCD operations, print, CRT macros, business oriented	[1] Double sided, double density
Control Systems, 1317 Central Ave., Kansas City, KS 66102	UCSD Pascal	6800 boards	6800, 6809, 68000[1]	R, F	1	1	[2]	No	Yes	56 kb	0.4 Mb[3]	6800, 09, Z80 8080, 6502 9900, PDP 11	P-code	Pascal Basic	32 ser. chan. max 512 discrete I/O max	MT 809 is similar to RT 68 but for 6809, also includes intertask communication	[1] Future [2] Future 11 Mb [3] Floppy or hard disk
	MT 809	6809 boards		R	16	1	Yes	No	Yes	60 kb	[3]	6809	N/A	N/A			
Cromemco Inc., 280 Bernardo Ave., Mountain View, CA 94043	CDOS	System Two System Three	Z80A	F[1] H[2]	1	1	No	No	No	512 kb	80 Mb (8 disks)	Z80	Extended & Structured Basic	Fortran IV Ratfor Cobol	CRT terminal, Dot-matrix pr., Character pr., a/d d/a[3]	Supports 5 & 8 in. floppy disks and 8 in. Winchester hard disk	[1] 5 and 8 in. [2] 8 in. [3] Opto-isolated I/O
	Multi-user CDOS		Z80A	F[1] H[2]	7	1	Yes	No	No	512 kb	80 Mb (8 disks)	Z80	Multiuser Basic	Fortran IV Ratfor Cobol			
Data General, Rte 9, Westboro, MA 01581	MP/OS	μNova family	MN601, 602	F	18[1]	N/A	Yes	N/A	Yes	64 kb	100 Mb (8 disks)	Macro	N/A	MP/Pascal MP/Fortran	6 async ports, 8 line prt,[2] 4 disks[3]	Full development & runtime. Development on disk or diskette-based μNovas or Nova-Eclipse mini's[5]	[1] Application-limited development 1 user typical. 3+ [2] Also a/c & c/a DIO, mag & paper tape user defined devices [3] Runtime only [4] Or 6 diskettes [5] MP OS compatible w AOS
	DOS	MN601 AP100 MP/200	MN601, 602	F	18[1]	N/A	Yes	X-25	Yes	64 kb	44 Mb (4 disks)	Macro	Ext., Bus., multiuser Basic	Fortran IV DG/L[4]	18 async ports 2 line prt[2] 8 disks[3]		
	RTOS	μNova family	MN601, 602	F	18[1]	N/A	Yes	X-25	Yes	64 kb	10 Mb (1 disk)	Macro[4]	N/A	Fortran IV	3 async ports 1 line prtr.[2] 1 disk	Compatible with DOS/RDOS runtime only	
	MBC/M	μNova family	MN601, 602	F	18[1]	N/A	Yes	N/A	Yes	64 kb	no disk support	Macro[3]	N/A	N/A	1 async port 1 par (32 bit)[4]	PROM-based OS for μNova boards. Develop on MP/OS, DOS, AOS	
Digital Equipment 146 Main St., Maynard, MA 01754	RT-11	LSI-11/12	LSI-11/23	F	4	1	No	Yes	Yes	64 kb	21 Mb (4 disks)	PAL-11 Macro-11	Minc Basic Focal-11	Fortran IV Basic+2	CRT, line printer RE 488 Bus, RT clock 4 ser channels	Compatible with PDP-11 mini family. Full PDP-11/30 instruction set	
	RSX-11M	LSI-11/23	N/A	F	16	1	Yes	Yes	Yes	256 kb	21 Mb (4 disks)	PAL-11 Macro-11	Minc Basic Focal-11	Fortran IV Basic+2			
Digital Research, P.O. Box 579 Pacific Grove, CA 93950	CP/M 2.0	8080/280 boards	N/A	F[1]	N/A	N/A	No	N/A	No	64 kb	128 kb (16 disks)	8080†	†	†	User defined console, reader, punch	†Much add'l software av. from outside vendors (Basic, Fortran, Cobol, Communications, Business Applic. etc.) MP/M is upward compatible from CP/M	[1] Field-definable [2] 16+8x48 kb with optional bank select [3] (64K 8"/8) [4] (max 8MB each)
	MP/M	8080/280 boards	N/A	F[1]	16	N/A	Yes	N/A	Yes	400 kb[2]	128 Mb (16 disks)	8080†	†	†			
E&L Instruments, 61 First St., Derby, CT 06418	MMD	MMD-1, 2	8080A, -1, -2	C, R	1	1	No	No	Yes	64 k	None	8080[1]	2-k Basic[2] 8-k Basic		1 ser. RS232, 20 mA (110 9600) 1 cassette I/C (300 1200)	Simple, compatible with Bugbook tutorials	[1] In development stage [2] Available in ROM
Forth Inc. 815 Manhattan Ave., Manhattan Beach, CA	PolyForth	Intel, DEC TI	8080/5/6 LSI-11 9900	H, P	[3]	N/A	Yes	N/A	Yes	64 k	[3]	8080/5/6, 1802, 9900, 280 68000	Forth	Forth	CRT disk, Printer	Interactive structured programming detonary structure target computer	
	Micro-Forth	Zilog, RCA	Z80 6800 1802		1	N/A	Yes	N/A	Yes	64 k							
GenRad/Futuredata 6151 W. Century Blvd., Ste 1124, Los Angeles, CA 90045	NDOS	8080-5/6, Z80, 6800 6802 boards		F	1	1	No	No	Yes	64 kb or less	4 Mb	8080-5/6, Z80 6800/02	Basic	Pascal	CRT (fast DMA) 1 par port[1] 2 ser		[1] Of RDOS types [2] NDOS supports 8 CRTs max [3] Continuous
	NDOS	Z80 boards	Z80	F	8[1]		No	Yes	Yes	64 kb	4 Mb						

TABLE XIV (Continued)

Company	OS model	Target µCo Model(s)	µPs used	Mode[a]	Number of simultaneous users	Number of simultaneous µPs	Multi-tasking	Net-working	Real-time	RAM (max)	Disk+ base	Supported languages: Assemblers	Supported languages: Interpreters	Supported languages: Compilers	Supported Peripherals	Comments	Notes
Gnat Computers, 7895 Convoy Court, Unit 6, San Diego, CA 92111	GS10	Gnat Spc. 10	Z80A	F	1	2[1]	1	N/A	N/A	56 k	16M (H) 0.8M (F)	8050, Z80, 1802 etc.	Forth, MBasic, Pascal	C, Fortran	Int. CRT (20 x 80), 3 ser ports, 1 par IEEE 488 (24 bits)	Hardware floating point	[1] Optional [2] With 8901
Hoeltron Corp., 700 W. Badger Rd., Madison, WI 53713	OS-80	MLZ-80	Z80	F,R,H	1	8	Yes	No	Yes	64 kw	128 Mb	8080, Z80	XYBasic	CP/M, MP/M, CBasic, Others	CRT, printer, 2 SIO, 4 PIO, floppy, I/F[2]	On-card floppy disk interface, OS-81 plus math processor	[1] Depends on memory [2] Use MLP-80 based MDS [3] OS-81 has 4 SIO, 2 PIO
	OS-81	MLZ-81	Z80	F,R,H	1	8	Yes	No	Yes	64 kw	128 Mb	8080, Z80	XYBasic				
	OS-DAQ	MLZ-DAQ	Z80	R	N/A	8	No	No	Yes	64 kw	N/A	[3]	[3]	[3]	CRT, printer, 2 SIO, 4 PIO	On-card 32 channel A/D converter	
Industrial Programming, 9 Northern Blvd., Greenvale, NY 11548	MTOS-11	PDP-11 based	LSI-11, PDP-11	F[1,2]	1[3]	1	Yes	No	Yes	64 kb	No	N/A	N/A	N/A	CRT, floppy, line printer, iSBC 310(80/86), iSBC 711(80/86), TI 700(8088)	Small size, fast interrupt response, unlimited number of tasks	[1] Only source language available [2] Source and object forms available [3] Per console [4] Dynamic debugger included
	MTOS-80	8080-based	8080, 8085, Z80	F[1,2]	1[3]	1	Yes	Yes	Yes	64 kb	No	N/A	N/A	N/A			
	MTOS 68	6800-based	6800, 6801, 6801	F, R[4]	1[3]	1	Yes	No	Yes	64 kb	No	N/A	N/A	N/A			
	MTOS 86	8086-based	8086, 8088	F, R[4]	1[3]	16	Yes	Yes	Yes	1 Mb	ISIS files	N/A	N/A	N/A			
Intel Corp., 3065 Bowers Ave., Santa Clara, CA 95051	ISIS-II	Intellec	8048, 80, 85, 86	F	1	N/A	No	No	Yes	1 Mb	15 Mb	8048, 80, 85, 86, 89	Basic-80, Cobol-80	Fortran-80, PL/M 80, PL/M 86	CRT, diskettes	RMZ-80 supports math board, disk (floppy & hard types), ser./parallel I/O controller	
	RMX 80	SBC 80/xx	8080/8085	R	1	N/A	Yes	No	Yes	64 k	80 Mb	N/A	Fortran, Basic	N/A			
Ithaca Intersystems, 1650 Hanshaw, Ithaca, NY 14850	IX3FDOS	DPS-1	Z80	F	1	1	Yes	No	Yes	64 kb	120 Mb (4 disks)	Z80 macro	Basic	Pascal	CRT (16 x 64), printers[1], F or H disk	Pascal re-entrant & RO*table interface to OS through library	[1] 1 line printer, 1 char. printer
Microbar, 711 Semoran Blvd., Altamonte Springs FL 32701	DEC-OS	DEC PDP family	PDP-11	F, R	9[1]		Yes	Yes	Yes	312 Mb[2]	5 Tb[3]	Full language processor and operating system called Microbar			Serial devices, CRTs, multiple attach command	10-100 times faster than DEC Basic or Dibol (no compilation required)	[1] In 64 kb [2] On LSI-11 [3] On PDP-11, LSI-11/23
Microgamma Systems, 413 Potter, Wichita, KS 67206	MTS Basic-1	SBC family	8080, 8085	F[1]	1	1	No	No	N/A	65 kb	[2]	8080	Basic		CRT, printer, I/O channel (user defined)	Low cost, easy to incorporate into user system; I/O expansion capability	[1] Some routines in ROM [2] Depends on SBC appl. [3] 3 floppy disks
Mostek Corp., 1215 W. Crosby Rd., Carrollton, TX 75006	FLP80 DOS	Z80 family	Z80	F	1	1	No	No	Yes	64 kb	2 Mb	3870[1,2] Z80[2]	Basic, Pascal	Fortran	Terminal, line printer, floppy disk, PROM prog[2] serial Intel[2] card, paper[2] tape reader[2] Silent 700[2]	OS for Mostek Matrix development system	[1] Cross-assembler [2] Macroassembler [3] FLP80 DOS only
	FLP80DOS/MDX	Z80 family	Z80	F	1	1	No	No	Yes	64 kb	2 Mb	3770[1,2] Z80[2]	Basic, Pascal	Fortran		Development OS for STD-Z80-based MD boards	
	MD DOS	Z80 family	Z80	F	1	1	No	No	Yes	64 kb	2 Mb		Basic			Production OS for STD-Z80-based MD boards	
	MITE-80	Z80 family	Z80	F, R	1	1	Yes	No	Yes	64 kb	2 Mb					Multitasking OS for STD-Z80-based MD boards	
	DDT-80 ASMB 80	Z80 family	Z80	R	1	1	No	No	Yes	54 kb		Z80			Silent 700	ROM-based OS, Assembler, editor, linker for use w/ MD or SD µC boards	
	HDC BOMOS	Z80 family	Z80	F, H	1	1	No	No	Yes	64 kb	10 Mb		Basic, Pascal	Fortran	N/A	Hard disk based Matrix dev. system. Also for SD boards	

TABLE XIV (Continued)

Company	OS model	Target µCs Model(s)	µPs used	Media*	Number of simultaneous users	Number of simultaneous µPs	Multi-tasking	Net-working	Real-time	RAM (max)	Data-base	Supported languages: Assemblers	Supported languages: Interpreters	Supported languages: Compilers	Supported Peripherals	Comments	Notes
Motorola Semiconductor Products Inc., P.O. Box 20912 Phoenix, AZ 85018	MDOS 6800	EXORciser EXORterm	6800	F	1	1	No	No	No	64 kb	2 Mb[1]	6800	Basic	MPL Fortran Cobol	CRT, EXORdisk II/III, Printer	Relocatable Macroassembler & CRT Editor	[1] 4 diskettes [2] One-line assembler/disassembler [3] Also arithmetic module a/d, d/a micromodule
	MDOS 6809	EXORciser EXORterm	6809	F	1	1	No	No	No	64 kb	2 Mb[1]	6800	Basic, Pascal	MPL, Basic Fortran Pascal			
	SUPERbug	Micromodule 19	6809	R	1	1	No	No	No	64 kb	N/A	[2]	Superlink (Linker)		CRT, TTY, Silent 700 IEEE 488[3]		
National Semiconductor Corp., 2900 Semiconductor Dr. Santa Clara, CA 95051	Starplex DOS	8080, 85, Z80, others[1]	8080A	F	1	[2]	No	No	Yes[3]	64 kb	4 Mb, 4 disks	[4]	Basic	Fortran	CRT (24 x 80) 4 floppy dr. RS232C ser. int. printers PROM progr	Powerful editor, prompting menus. BLC/SBC board level computer support. Starlink (Starplex to Intel MDS communications link)	[1] 8048/49/50, 8070 [2] Multiprocessing ISE [3] Real-time in syst. emul. [4] Assemblers (see [1])
Northern Telecom Systems Corp., Data Park, Minneapolis, MN 55440	OS-85	Model 85	FML-11	[1]	15	1	8 tasks	No	Yes[2]	256 kb	245 Mb, 4 disks	N/A	Cobol RPG-II	Cobol RPG II	CRT (1920 char) 5 printers, 4 mag tapes	Optional full page text-editor, 3270 comm., input & output spooling[3]	[1] Mag. tape 5 Mb disk cartr. [2] 0.5-1.5 s response [3] 2 max data channels for 7x products [4] 2 foregrd, 1 backgrd [5] Proprietary language [6] Omni-Task Multiple 8085 [7] 8 foregrd, 8 backgrd [8] 10 fixed, 298 removable [9] Remote data station
	OS-405	Model 405	8085 types	F	2[4]		3 tasks	Yes	Yes	64 kb	1 Mb	N/A	Cobol, Basic Tal 2000[5]		2 CRTs (24 x 80) 2 comm. ports, 1 printer	Datalink; attach 9 CPUs together, share disk & peripherals—mag tape, BSC/SDLC, async printers from 60 ch/s to 600 lines/min.	
	OS-445	Model 445	8085 types	C,[1] F	8[7]		16[6] tasks	Yes	Yes	256 kb	308 Mb[9]	N/A	Cobol, Basic Tal 2000[5]	Cobol, Basic, Tal 2000[5]	8 CRTs (24 x 80), 3 comm ports, 8 printers[8]		
Ontel Corp., 250 Crossways Park Dr., Woodbury, NY 11797	MDOS	OP-1/50/64/70, OP-1/R/HS/RW	8080, 8085-A2	F[1] H[2]	1	3	No	Yes[3]	Yes	64 kb	1.2 Mb						[1] 5 or 8 in. floppy [2] 5 20-Mb disk [3] Slave in 16 term net [4] Slave or master [5] 16 terms share disk [6] Remote data base via virtual disk diskette [7] Proprietary language [8] Not for VHMOS
	HDOS	OP-1/50/64/70	8080, 8085 A2	H[2]	1	3	No	Yes[3]	Yes	64 kb	160 Mb, 8 disks	Macro, linking, relocating	Basic Word[7] Dear[7]	Fortran IV, Pascal, OPL[7]	CRT, keyboard line & char. printers, tape drive[8]	Exec processor, copy, dup, system, format, list, edit, diagnostics, debugger, compare etc.	
	VHMOS	OP-1/64, OP-1/R/RW	8080, 8085 A2	R	1	3	No	Yes[3]	Yes	64 kb	160 Mb[6]						
	MFTOS	OP 1/50, 64, 70	8080, 8085 A2	F[1] H[2]	16[5]	3	255 tasks	Yes[4]	Yes	64 kb	160 Mb						
Percom Data Co., 211 N. Kirby Rd., Garland, TX 75042	PSYMON	[1]	6809	R	1	1	No	No	Yes	1 kw[4]	N/A	N/A	N/A	N/A	CRT (RS-232), cassette (KC), printer	User-modif. of commands: hardw., easily adaptable: 15 utilities for program dev.; EPROM avl. for SBC9	[1] Percom 6809 EXORciser, Adapter modif. Creative Microsystems EXORciser, Adapter modif. SWTP MP-A2, Percom SBC 9 [2] EXORciser, SWTP MF-68, Smoke Signal BFD 68, Percom LFD-family [3] Minifloppy [4] Auto-links to second ROM via expansion routine [5] I/O devices, peripherals treated as disk files
	INDEX	[2]	6800	F[3]	1	1	No	No	Yes	8 kw	N/A	N/A	Strubal, Super-Basic	N/A	I/O-independent[5]	INDEX incl. file mgt system; periphs serviced by interrupts, user-expandable; supports serial ASCII	
	MPX	EXORciser, Percom LFD	6800	F[3]	1	1	No	No	Yes	0.5 kw	N/A	6800		N/A	Cassette, CRT printer	Preallocated, random access, contiguous file structures	
	MICRODOS	TRS-80	Z80	F[3]	1	1	No	No	Yes	7 kw	N/A	N/A	Level-II Basic	N/A	TRS 80 peripherals	Hex constants, 10 user-defined functions, comes w. file mgt., disk util., applic. prgms (Basic)	
Process Computer Systems, 750 N. Maple Rd., Saline, MI 48176	3800B/2	3800B/2	Z80	F, R	1	1	Yes	PCS, party-line, up-load, down-load	Yes	64 kb	512 kb	Z80	Fortran, Ind. Pascal	Pascal	CRT Keyboard Printer 2 floppies EPROM burner communications	Extended memory, limited graphics	
	3900A	3900A	Z80	F, R	1	1	Yes		Yes	64 kb	512 kb	N/A	N/A	N/A			
	3905A	3905A	Z80	F, H	1	1	Yes		Yes	96 kb	512 kb	N/A	N/A	N/A			
RCA Solid State, Rte 202, Somerville, NJ 08876	1802 OS	RCA boards	1802	F	1	1	No	No	No	62 kb	0.5 Mb[1]	1802, 1804	Basic 1, 2, µForth	Basic-1, PL/M	CRT, TTY printer	Macroassembler, math library (opt.)	[1] 2 diskettes
Ryan McFarland, 609 Deep Valley Dr., Rolling Hills Estates, CA 90274	RM990	TI990/5 10	TI9900	C, F, R	1	1	Yes	No	Yes	64 kb[1]	1 Mb 4 disks	N/A	Process control Basic		CRT, printer minicartridges Process I/O	Up to 256 tasks, designed for process control	[1] For TI9900, 2 Mb for TI 990/10
	RM Cobol	TI-990/5	TI9900	F, R	4	1	Yes	No	Yes	64 kb	4 disks—4 Mb	N/A		Cobol		Designed to support Cobol Multikey index files [illegible]	

TABLE XIV (Continued)

Company	OS model	Target µCs model(s)	µPs used	Media*	Number of simultaneous users	Number of simultaneous µPs	Multi-tasking	Net-working	Real-time	RAM (max)	Data-base	Supported languages: Assemblers	Supported languages: Interpreters	Supported languages: Compilers	Supported Peripherals	Comments	Notes
R2E of America, 47 Bedford St SE Minneapolis, MN 55414	BAL	80/30, 40, 50, 60	8080/Z80	N/A	8	8	Yes	No	No	512 kb	1 Gb[1]	8080, Z80	MBasic	Bal Basic, Forth 80	8 CRTs GPIB	Available with AMD9511 for fast math	[1] 8 drives
	SYSMIC	80/30, 40	8080/Z80	N/A	1	1	No	No	No	64 kb	1 Gb[1]	8080, Z80		Forth 80	11 CRT, 1 ser chan, GPIB		
	MTS	80/50, 80/60	Z80	N/A	16	16	Yes	Yes	Yes	960 kb	1 Gb[1]	8080, Z80	MBasic	MBasic, Fortran 80, Cobol 80	16 CRTs, GPIB		
Software Dynamics, 2111 W Crescent, Suite G, Anaheim, CA 92801	SDOS	Motorola 6800	6800/02/09	F,H[1]	1[2]	1	Yes	Future	Yes	64 kb	2.5 Gb per drive	6800	EDIT	Basic w. 10 digit BCD	CRT, Line printers, Diablo hytype, disk drives	Interrupt-drive I/O, any mixture of drives, automatic disk read-ahead, random/sequential access to dynamically grown files of 2.5 Gb	[1] Winchester 10 Mb cartridge disk, IBM compat floppies [2] Multi terminal avl soon.
Technical Systems Consultants Box 2574, 1208 Kent Ave. W Lafayette, IN 47906	FLEX	Motorola	6800, 6809	F	1	1	No	No	Yes	56 kB	20 Mb (1 disk)	6800, 6809[1]	Basic, ext. Basic[2]	N/A	Terminal, disks	Also supports debug package, sort/merge, text formatting, fast interpreter	[1] Native and cross-assemblers [2] 16 digit precision
Texas Instruments, P.O. Box 1444, Houston, TX 77001	TX999	990/4	TMS 9900	F	1	2	Yes	No	Yes	32 kw	1 Mb (4 disks)	9900[1]	Basic	Fortran	CRT (24x80), CRT (12x80), Line printer PROM burner, AMPL emulator	AMPL option for emulation and logic-state tracing	[1] User-expandable [2] Avl summer 1980 [3] 2-pass assembler
	TIPMX	990/4	TMS 9900	F,H,R	2	2	Yes	Yes[2]	Yes	64 kb	1 MB (4 disks)	9900[1]				Multiprogramming, math package	
	Basic/OS	990/4, board	TMS 9900	F,H,R	2	2[3]	Yes	Yes[2]	Yes	64 kb	0.5 Mb (2 disks)	9900[1] Basic	Basic			IF/THEN construct, math pack., direct access to bit level	
	Pascal/OS	990/4	TMS 9900	F,H	2	2	Yes	Yes[2]	Yes	64 kb	1 Mb (4 disks)	9900[1]	Pascal	Pascal		Multiprogramming, native code or interpretive, math pack, direct bit-level access	

N/A: Not Applicable

(From: Electronic Design 24, Nov. 22, 1979)

TABLE XV

Semiconductor Technologies

The purpose of this table is to present briefly the semiconductor technologies currently used in microprocessors.

a. Characteristics of a Semiconductor Technology

1. Speed
2. Density
3. Cost
4. Power consumption
5. Noise immunity
6. Ruggedness
7. TTL compatibility
8. Maturity or experience

b. Comparison of Semiconductor Technologies

Technology	Speed 1 = fast	Density 1 = complex	Cost 1 = cheap	Power 1 = lower	Ruggedness 1 = most	Experience 1 = long	TTL Compatibility
CMOS	3	3	3	1	1	4	Yes
ECL	1	6	6	6	6	5	No
TTL Scho.	2	5	3	5	2	1	Yes
I^2L	3	3	3	2	3	6	?
NMOS	5	1	2	3	4	3	Sometimes

c. Application Requirements and Appropriate μP Technologies

Requirement	Most Suitable Technologies
Low cost	PMOS, NMOS
Small size	PMOS, NMOS
High speed	ECL, TTL Schottky
Low power consumption	CMOS
Rugged environments	CMOS
Compatibility with: TTL	TTL Schottky, CMOS
CMOS	TTL Schottky, CMOS
ECL	ECL
Wide availability	PMOS, NMOS
Standard parts in same technology	TTL Schottky, CMOS, ECL
Large memories in same technology	PMOS, NMOS, TTL Schottky
Most support	PMOS, NMOS

TABLE XV (Continued)

d. Typical Characteristics of Semiconductor Technologies

Technology	Typical Gate Size (sq.mils)*	Typical Delay/Gate (nanosec.)	Typical Power Gate	Typical Cost/Gate (cents)	Typical Noise Immunity (volts)
PMOS	3	100 ns	0.2 mW	0.1-2	1.0
NMOS	2	50 ns	0.2 mW	0.1-2	0.4
CMOS	12	25 ns	10 μW	10-30	4.0
TTL	13	10 ns	10 mW	5-15	0.4
Schottky TTL	5	3 ns	20 mW	25	0.3
Low Power Schottky TTL	5	10 ns	2 mW	25	0.3
ECL	8	2 ns	30 mW	30-40	0.125
I^2L	1	25 ns	50 μW	5-50	0.2

* One square mil. = 6.45×10^{-6} square centimeters.

e. Types of Semiconductor Memory

Memory Type	Variability	Technologies	Volatility	Typical Size (bits)	Use
ROM	Fixed	All	Nonvolatile	16K	Program memory, tables
PROM	Programmed once	Most	Nonvolatile	1K to 16K	Program memory, tables
EPROM	Can be reprogrammed	MOS	Nonvolatile	2K to 32K	Program memory, tables
EAROM	Can be reprogrammed	Few	Nonvolatile	1K	Program memory, tables
RAM	Variable	All	Volatile	1K to 16K	Data
Shift register	Variable	All	Volatile	8-256	Input/output
Buffer	Variable	Most	Volatile	32-256	Input/output

f. Available Semiconductor RAMs

Technology	Maximum Size (bits)	Typical Size (bits)	Typical Access Time	Dynamic or Static	Typical Cost
PMOS	4K	1K	1 μs	Both	$3/1K bits
NMOS	16K	1K	300-500 ns	Both	2/1K bits
CMOS	1K	256	100-500 ns	Static	20/1K bits
Schottky TTL	1K	256	50-100 ns	Static	20/1K bits
ECL	1K	128	10-50 ns	Static	15/128 bits
I^2L	4K	-	100 ns	Static	-

TABLE XV (Continued)

g. Available Semiconductor ROMs

Technology	Maximum Size (bits)	Typical Size (bits)	Typical Access Time	Typical Cost
PMOS	64K	16K	800 ns 1.5 µs	$25/16K
NMOS	64K	16K	100 ns 1 µs	25/16K
CMOS	12K	1K	100-500 ns	50/12K
Schottky TTL	16K	1K	50-100 ns	10/1K

h. Available PROMs

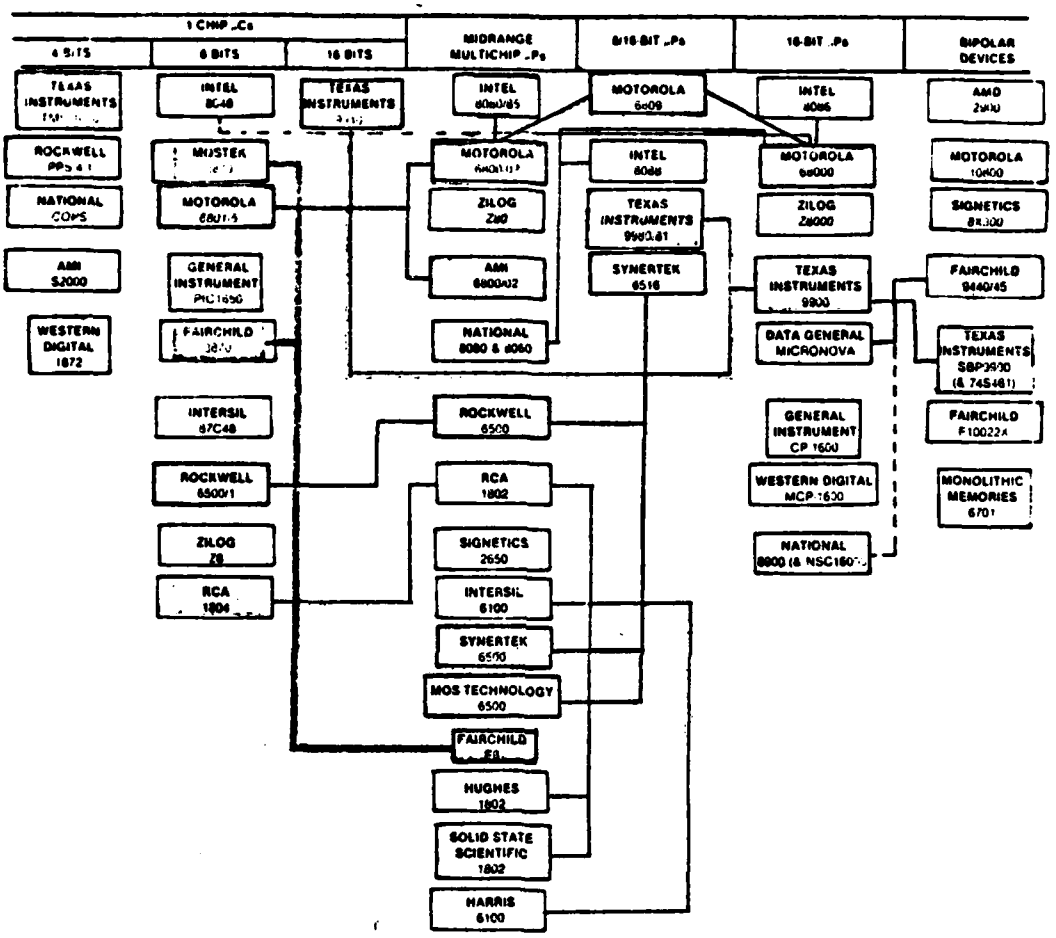
Technology	Sizes (bits)	Typical Access Times	Typical Programming Time	Typical Cost
PMOS	1K-8K	1 - 2 µs	2 min	$20/2K
NMOS	1K-32K	300 ns - 1 µs	2 min	20/8K
TTL	512-4K	50 - 100 ns	30 s	22/1K
Schottky TTL	512-16K	50 - 100 ns	30 s	6/1K
ECL	1K	10 - 50 ns	--	--

i. Typical Shift Registers

Technology	Sizes (bits)	Maximum Operating Frequency (MHZ)	Cost
PMOS	up to 2K	1	$5/1K
NMOS	up to 2K	1-10	$5/single 1K dynamic
CMOS	up to 200	5	$6/200 bits
Schottky TTL	4 or 8	50-70	$2-$10
ECL	4 or 8	100	$10-$15

TABLE XVI

μP/μC-Chip Families

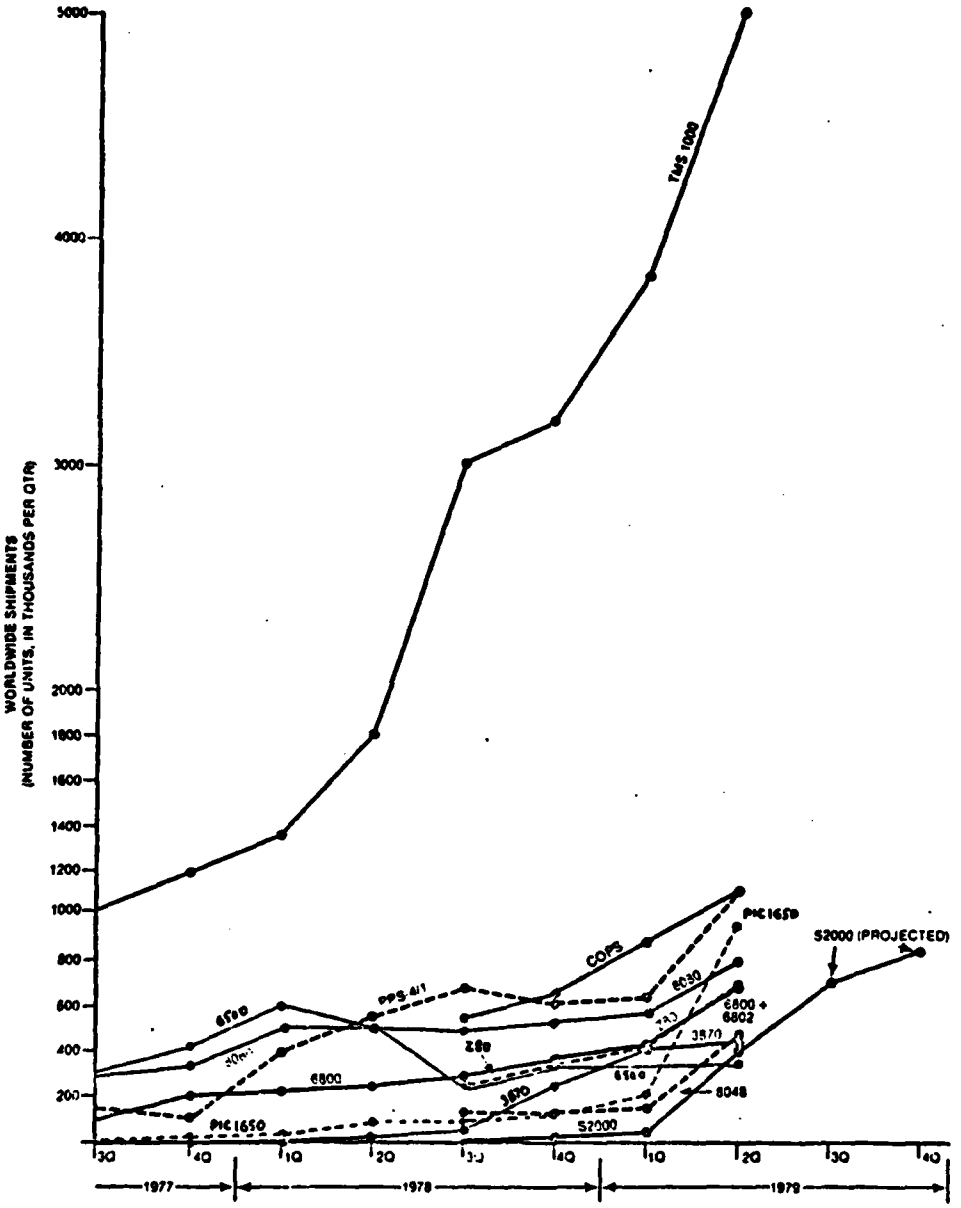


Worksheet for a first-pass evaluation of μP/μC-chip families. The attempt at such an evaluation-reflected in the rankings within each application area-hinges on the criterion that high-volume production makes a μP or μC a safe bet; technical appeal is the secondary criterion. Lines connecting chips in the different application areas show which families have the broadest application flexibility (solid link with relatively strong family relationship and dashed link chips with weaker ties.)

(From: EDN, October 20, 1979)

TABLE XVII

Production Volumes of µPs and µCs



Production volumes for many µPs and µCs soared in 1979. The older PMOS 1-chip µCs continue to lead in volume, with the TMS 1000 remaining ahead of all competition by a wide margin. In the 8-bit 1-chip-µC field, Intel's 8048 is coming on strong and could surpass the 8080 in overall profitability. In the 16-bit-µP field, trends haven't developed with regard to the 8086, Z8000 and MC68000. (From: EDN, Oct. 20, 1979).

TABLE XVIII

LIST OF ORIGINAL-SOURCE μP/μC MANUFACTURERS

Name	Address	Telephone
Advanced Micro Devices	901 Thompson Pl Sunnyvale, CA 94086	(408)732-2400
AEG Telefunken	6 Frankfurt 70 AEG Hochhaus, Federal Republic of Germany	---
American Micro-Systems Inc.	3800 Homestead Rd, Santa Clara, CA 95051	(408)246-0330
Analog Devices, Inc.	P.O. Box 208, Norwood, MA 02062	(617)329-4700
Burr-Brown	P.O. Box 11400, Tucson, AZ 85734	(602)294-1431
Data General Corp.	9 Route, Westboro, MA 01581	(617)366-8911
EMM Semiconductor	3883 N. 28th Ave. Phoenix, AZ 85017	(602)263-0202
ESSEX International	301 Alpha Drive Pittsburgh, PA 15328	(412)963-9322
Fairchild Semi-conductor	464 Ellis St. Mountain View, CA 94040	(415)962-3541
Ferranti Electric	E. Bethpage Road Plainview, NY 11803	(516)293-8383
Ferranti Ltd.	Western Rd., Bracknell Berkshire RG121RA England	---
Fujitsu America, Inc.	1208 E. Arques Ave. Sunnyvale, CA 94086	(408)739-3200
Fujitsu, Ltd.	6-1 Marunochi 2 Chome Chiyoda-Ku, Tokyo, Japan	---

Name	Address	Telephone
General Instrument Corp.	600 W. John Street Hicksville, NY 11802	(516)733-3130
Harris Semiconductor	P.O. Box 883 Melbourne, FL 32901	(305)727-5407
Hitachi America Ltd	707 W. Algonquin Rd. Arlington Heights I1 60005	(312)593-7660
Hitachi Ltd	Nippon Bldg. No. 602 2-Chome, Ohtemachie, Chiyoda-Ku, Tokyo 100 Japan	---
Hughes, Solid State Division	2601 Campus Drive Irvine, CA 92715	(714)752-6396
Intel Corp.	3065 Bowers Ave. Santa Clara, CA 95051	(408)987-8080
Intersil Inc.	10900 N. Tantau Ave. Cupertino, CA 95014	(408)996-5000
ITT Semiconductor	74 Commerce Way Woburn, MA 01801	(617)935-7910
ITT Semiconductor	Maidstone Road Footscray, Sidcup Kent, England	---
Matrox Electronic Systems Ltd	2795 Bates Rd. Montreal, Quebec H35-1B5, Canada	(514)481-6838
Micro Networks Corp.	324 Clark St. Worcester, MA 01606	(617)852-5400
Mitel Semiconductor	18 Airport Blvd. Bromont, Quebec, JOE-1LO, Canada	(514)534-2321
Mitel Semiconductor	2321 Morena Blvd. Suite M, San Diego CA 42110	(714)276-3421
Monolithic Memories, Inc.	1165 E. Arques Ave. Sunnyvale, CA 94086	(408)739-3535

Name	Address	Telephone
MOS Technology, Inc.	950 Rittenhouse Rd. Norristown, PA 19401	(215)666-7950
Mostek Corp.	1215 W. Crosby Rd. Carrollton, TX 75006	(214)242-0444
Motorola Semiconductor	5005 E. McDowell Rd. Phoenix, AZ 85008	(602)244-6900
Motorola Integrated Circuits	3501 Ed Bluestein Blvd. Austin, TX 78721	(512)928-6800
National Semiconductor Corp.	2900 Semiconductor Dr. Santa Clara, CA 95051	(408)737-5000
NEC Microcomputers, Inc.	173 Worcester St. Wellesley, MA 02181	(617)237-1910
Panafacom Ltd.	2-10-16 Jiyuzaoka Mezuro-Ku, Tokyo, Japan	---
Panasonic	50 Meadowland Parkway Secancus, NJ 07094	(201)348-7276
Philips Industries, Electronic Comp. and Materials Div.	P.O. Box 523, Eindhoven The Netherlands	---
Raytheon Semiconductor Div.	350 Ellis St., Mountain View, CA 94042	(415)968-9211
RCA Solid State Div.	P.O. Box 3200, Rte. 202 Somerville, NJ 08876	(201)685-6423
Rockwell International	P.O. Box 3669, RC01 Dept. 720, Anaheim, CA 92803	(714)632-2321
Scientific Micro Systems	520 Clyde Ave., Mt. View, CA 94043	(415)964-5700
SGS-ATES	Via C Olivetti 1/20041 Agrate Brianza, Italy	---
Sharp	22-22 Nagaike-Cho, Abeno-Ku, Osaka 545 Japan	---
Sharp Electronic	10 Keystone Place Paramus, NJ 07652	(201)265-5600

Name	Address	Telephone
Siemens AG. Central Information Department	Oskar-Von-Miller Ring 18, D-8000, Munich 2, Federal Republic of *Germany*	---
Siemens Corp.	186 Wood Ave., South Iselin, NJ 08830	(201)494-1000
Signetics	811 E. Arques Ave. Sunnyvale, CA 94086	(408)739-7700
Solid State Scientific Inc.	Montgomeryville Industrial Park Montgomeryville, PA 18936	(215)855-8400
Standard Microsystems	35 Marcus Blvd. Hauppauge, NY 11787	(516)273-3100
Synertek	3050 Coronado Drive Santa Clara, CA 95051	(408)241-4300
Texas Instruments	13500 North Central Expressway, M/S 308 Dallas, TX 75222	(214)238-2011
Thomson-CSF, Sescosem	101 Boulevard Murat 75781 Paris Cedex 16 France	---
Toshiba America, Inc.	2900 McArthur Blvd. Northbrook, IL 60062	(312)564-5140
Toshiba Transistor Works	1 Komukai, Toshiba-cho Kawasaki-shi, Kanaganaken, Japan	---
TRW LSI Products	P.O. Box 1125, Redondo Beach, CA 90278	(213)535-1831
Western Digital Corp.	3128 Red Hill Ave. Newport Beach, CA 92663	(714)557-3550
Zilog Microcomputers	*10460 Bubb Rd.* Cupertino, CA 95014	(408)446-4666

GLOSSARY

Access Time - The delay between the time when a memory receives an address and the time when the data from that address is available at the outputs.

Accumulator - A register that is the source of one operand and the destination of the result for most arithmetic and logical operations.

Active-High - The active state is the one state.

Active-Low - The active state is the zero state.

Ada - High order level language designed for the necessities of DOD.

Address - The identification code that distinguishes one memory location or input/output port from another and that can be used to select a specific one.

Addressing Methods (Modes) - The methods for specifying the addresses to be used in an instruction. Common addressing methods include direct, indirect, indexed, relative, and stack.

ALGOL - Algorithmic Language, a widely used high-level language designed for systems and scientific applications.

Analog - Continuous signal or representation of a quantity that can take any value.

Anode - Positive terminal.

Architecture - Structure of a system. Computer architecture often refers specifically to the CPU.

Arithmetic-Logic Unit (ALU) - A device that can perform any of a variety of arithmetic or logical functions under the control of function inputs.

Arithmetic Shift - A shift operation that preserves the value of the sign bit (most significant bit).

Arm - See Enable, but particularly applied to enabling interrupts.

ASCII - American Standard Code for Information Interchange, a 7-bit character code widely used in computers and communications.

Assembler - A computer program that converts assembly language programs into a form (machine language) that the computer can understand. The assembler translates mnemonic instruction codes into binary numbers, replaces names with their binary equivalents, and assigns locations in memory to data and instructions.

Assembly Language - A programming language in which the programmer can use mnemonic instruction codes, labels, and names to refer directly to their binary equivalents. The assembler is a low-level language, since each assembly language instruction translates directly into a specific machine language instruction.

Asynchronous - Operating without reference to an overall timing source, that is, operating at irregular intervals.

Attached Input/Output - An addressing method for input/output ports that identifies the ports either directly (if the port is attached to the CPU) or from the address in memory to which the port is attached. The port is usually selected with special instructions that are decoded either in the CPU or in the memory section. Systems using attached I/O are frequently based on LSI devices that combine memory, input/output, and processor functions.

Auto-Correlation Functions (ACF) - Output of the transmitted signal.

Autoindex - An index register that is automatically incremented or decremented with each use.

Auxiliary Carry Bit - See Half-Carry Bit.

Bank - A directly addressable set of registers or memory locations. The register or other storage device that selects banks is called a "bank switch."

Baud - A communications measure for serial data transmission, bits per second but including both data bits and bits used for synchronization, error checking and other purposes.

Baud Rate Generator - A device that generates the proper timing interval between bits for serial data transmission.

Baudot Code - A 5-bit character code used in telegraphy and some communications terminals.

BCD (Binary-Coded Decimal) - A method for representing decimal numbers whereby each decimal digit is separately coded into a binary number.

Benchmark Program - A sample program used to evaluate and compare computers.

Bidirectional - Capable of transporting signals in either direction.

Binary - A binary digit, possible values zero or one.

Bit Manipulation (or Bit Banging) - The examination and changing of single bits or small groups of bits within a word.

Bit Slice - A section of a CPU that may be combined in parallel with other such sections to form complete CPUs with various word lengths.

Bootstrap Loader (or Bootstrap) - Technique for loading first instructions of a program into memory and then using these instructions to bring in the rest of the program. The first instructions (called the "bootstrap") may reside in a special read-only memory.

Borrow - A status bit that is one if the result of an unsigned subtraction was negative.

Bottom-Up Design - A design method in which parts (or modules) of a system are designed and tested separately before being combined.

Bounce - Moving back and forth between states before reaching a final state.

Branch Instruction - See Jump Instruction.

Breakpoint - A location specified by the user at which program execution is to end temporarily. Used as an aid in program debugging.

Bus - A group of parallel lines that connect two or more devices.

Bus Contention - A situation in which two or more devices are trying to place data on a bus at the same time.

Bus Driver - A device that amplifies outputs sufficiently so that they can be recognized by the devices on a bus.

Bus Isolation - Buffering parts of the bus away from other parts with buffers and drivers.

Bus Transceiver - A device that acts as both a bus driver and bus receiver; that is, it interfaces a bidirectional bus to two unidirectional buses.

Byte - The basic grouping of bits that the computer handles as a unit, most often eight bits in length.

Call - See Subroutine.

Carry Bit - A status bit that is one if the last operation generated a carry from the most significant bit.

Cartridge (or 3M Mag-Tape Cartridge) - A compact, enclosed package of magnetic tape that uses 1/4-inch tape and records 1600 bits per inch at 30 in./s on four tracks.

Cassette - An enclosed package of magnetic tape usually housed in a plastic container. Both audio and digital versions exist; the digital ones are more reliable and more expensive. The standard unit is the Philips-type cartridge, which consists of 282 feet of 0.015-in. magnetic tape, phase encoded at 800 bits per inch.

Cathode - Negative terminal.

Central Processing Unit (CPU) - The control section of a computer. It contains the arithmetic unit, registers, instruction-decoding mechanism, and timing and control circuitry.

Checksum - A logical sum of data that is included in a record as a guard against recording or transmission errors.

Chip - A substrate containing a single integrated circuit.

Clear - Set state to zero; an input to a device that sets the state to zero.

Clock - A regular timing signal that governs transitions in a system.

CMOS - Complementary metal-oxide semiconductor, a logic family that uses complementary N-channel and P-channel MOS field-effect transistors to provide high noise immunity and low power consumption.

Coding - The writing of programs in a language that is comprehensible to a computer system.

Common-Anode Display - A multiple display in which signals are applied to the cathodes of the individual displays and the anodes are tied together to the power supply; uses negative logic (i.e., a logic zero turns a display on).

Common-Cathode Display - A multiple display in which signals are applied to the anodes of the individual displays and the cathodes are tied together to ground. Uses positive logic (i.e., a logic one turns a display on).

Common I/O - Uses the same lines for input and output.

Communications Register Unit (CRU) - The general-purpose command-driven hardware interface of TI's 990/9900 μP family.

Comparator - A device that produces outputs that show whether one input is greater than, equal to, or less than the other input. Both analog and digital comparators exist.

Compiler - A program that converts a program in a high-level or procedure-oriented language into an assembly or machine language program.

Computer-aided design (CAD).

Concurrent characteristics - One of several characteristics associated with the definition of a process in μP PASCAL, including the process priority, the stack size and the heap size.

Condition Code (or Flag) - A single bit that indicates a condition within the computer, often used to choose between alternate instruction sequences.

Condition Code Register - A register that contains one or more condition codes.

Control Memory - A memory that holds microprograms--that is, a memory used to decode computer instructions.

Core Memory - A magnetic memory that can be magnetized in one of two directions so as to represent a bit of data.

Counter - A clocked device that enters a different state after each clock pulse (up to its capacity) and produces an output that reflects the total number of clock pulses it has received. Counters are also referred to as "dividers," since they divide the input frequency by "n" where "n" is the capacity of the counter.

Cross-Assembler - An assembler that runs on a computer other than the one for which it assembles programs.

CRT - Cathode-ray tube.

Current Page - The page of memory on which the present instruction is located.

Current-Loop Interface (or Teletype Interface) - An interface that allows connections between digital logic and a device that uses current-loop signals--that is, typically the presence of 20 mA in the loop is a logic one and the absence of that current is a logic zero.

Cycle Stealing - Using a cycle during which the CPU is not accessing the memory for a DMA operation.

Cycle Time - Time interval at which a set of operations is repeated regularly in the same sequence.

Cyclic Redundancy Check (CRC) - An error-detecting code generated from a polynomial that can be added to a data record or sector.

Daisy-Chain - An input or output method whereby signals pass from one device to another until accepted or blocked. Activity near the control unit for the chain will block activity farther from the control unit.

Data Acquisition System - A system that will accept several analog inputs and produce corresponding digital data. The system usually includes sample and hold circuitry, multiplexers, and converters.

Data Fetch Cycle - A computer operation cycle during which data is brought from memory to the CPU.

Data Pointer (or Pointer) - A register or memory location that holds an address rather than the data used.

Debounce - Convert the output from a contract with bounce into a single, clean transition between states.

Debounce Time - The amount of time required to debounce a closure.

Debug - To eliminate programming errors, sometimes referred to as verifying the program.

Debugger (or Debug Program) - A program that helps in finding and correcting errors in a user program.

Decade Counter - A counter with ten different states.

Decimal Adjust - An operation that converts a binary arithmetic result to a decimal (BCD) result.

Decoder - A device that produces unencoded outputs from coded inputs.

Delay Time - The amount of time between the clocking signal and the actual appearance of output data, or the time between input and output.

Demultiplexer - A device that directs a time-shared input to one of several possible outputs, according to the state of the select inputs.

DOD - Department of Defense.

Destructive Readout (DRO) - The contents cannot be determined without changing them.

Development System - A special computer system that includes hardware and software specifically designed for developing programs and interfaces.

Device, Logical - An entity with which programs can perform device-independent I/O.

Device, Physical - An entity that communicates with programs via a CRU or memory-mapped I/O and interrupts.

Diagnostic (Program) - A program that checks part of a system for proper operation.

Digital - Having discrete levels, quantized into a series of distinct levels.

Direct Addressing - An addressing method whereby the address of the operand is part of the instruction.

Directly Addressable - Can be addressed without changing the contents of any registers or bank switches.

Direct Execution - A method whereby the computer directly executes statements in a high-level language rather than translating those statements into machine or assembly language.

Direct Memory Access (DMA) - An input/output method whereby an external controller directly transfers data between the memory and input/output sections without processor intervention.

Disable - Prohibit an activity from proceeding or a device from producing data outputs.

Disable Time (Output) - The amount of time required for an active tri-state output to enter the third or open-circuit state.

Disarm - See Disable, but particularly applied to disabling interrupts.

DFT - Discrete Fourier Transforms.

Disk Operating System (DOS) - An operating system that transfers programs and data to and from a disk, which may be either flexible or fixed-head; the operating system may itself be largely resident on the disk.

Diskette - See Floppy Disk.

Divider - See Counter.

Dual Inline Package (DIP or Bug) - A semiconductor chip package having two rows of pins perpendicular to the edges of the package, sometimes called a "bug," since it appears to have legs.

Dynamic Memory - A memory that loses its contents gradually without any external causes.

EAROM - Electrically alterable ROM, a nonvolatile RAM, often with a relatively long write time.

EBCDIC - Expanded Binary-Coded Decimal Interchange Code, an 8-bit character code often used in large computers.

ECL - Emitter-coupled logic, a high-speed bipolar technology often used in computer mainframes.

Editor - A program that manipulates text material and allows the user to make corrections, additions, deletions, and other changes.

Effective Address - The actual address used by a particular instruction to fetch or store data.

Emulator - A microprogrammed copy of an existing system.

Enable - Allow an activity to proceed or a device to produce data outputs.

Encoder - A device that produces coded outputs from unencoded inputs.

EPROM (or EROM) Erasable PROM, a PROM that can be completely erased by exposure to ultraviolet light.

Error-Correcting Code - A code that can be used by the receiver to correct errors in the messages to which the code is attached; the code itself does not contain any additional message.

False Start Bit - A start bit that does not last the minimum required amount of time, usually caused by noise on the transmission line.

Fan-In - The number of inputs connected to a gate.

Fan-Out - The maximum number of outputs of the same family that can be connected to a gate without causing current overload.

FFAT - Fast Fourier Transforms.

Field-Programmable Logic Array (FPLA) - A programmable logic array that can be programmed by the user.

FIR - Finite Impulse Response, filters.

Firmware - Microprograms, usually implemented in read-only memories.

Fixed-Instruction Computer - A computer for which the manufacturer determines the instruction set. As opposed to microprogrammable computer.

Fixed Memory - See ROM.

Flag - See Condition Code.

Flatpack - A semiconductor chip package in which the pins are in the same plane as the package rather than perpendicular to it as in a DIP.

Flip-Flop - A digital electronic device with two stable states that can be made to switch from one state to the other in a reproducible manner.

Floating - Not tied to any logic level, often applied to tri-state outputs that are in the high-impedance state. TTL devices usually interpret a floating input as a logic one.

Floppy Disk (or Flexible Disk) - A flexible magnetic surface that can be used as a data storage device; the surface is divided into sectors. An IBM-compatible floppy disk is one that uses formatting and sectoring techniques originally introduced by IBM. The individual floppy disk is sometimes called a "diskette."

Flowchart - A graphical representation of a procedure or computer program.

FORTRAN - A high-level (procedure-oriented) programming language devised for expressing scientific problems in algebraic notation. Short for Formula Translation Language.

FTR - Functional Throughout Rate.

Gate - A digital logic element where the binary value of the output depends on the values of the inputs according to some logic rule.

GaAs - Gallium-Arsenide gates.

General-Purpose Interface Bus (GPIB or Hewlett-Packard Bus) - A standard interface for the transmission of parallel data in a network of instruments. The GPIB has 8 data lines, 8 control lines, and 8 ground lines.

General-Purpose Register - A register that can be used for temporary data storage.

Gray Code - A binary code sequence in which only one bit changes in a transition to the next higher or lower value.

Half-Carry (or Auxiliary Carry) Bit - A status bit that is one if the last operation produced a carry from bit 3 of an 8-bit word. Used on 8-bit microprocessors to make the correction between binary and decimal (BCD) arithmetic.

Hardware - Physical equipment forming a computer system.

Heap - A data area holding dynamically allocated variables.

Hex - (1) Containing six distinct logic elements, as in hex buffers; (2) abbreviation for hexadecimal or base 16.

Hexadecimal - Number system with base 16. The digits are the decimal numbers 0 through 9, followed by the letters A through F.

High-Impedance State - See Tri-State.

HLL (High-Level Language, or Procedure-Oriented Language) - A programming language for which the statements represent procedures rather than single machine instructions. FORTRAN, COBOL, and BASIC are three common high-level languages. A high-level language requires a compiler that translates each statement into a series of machine language instructions.

HOL - High Order Language (same as HLL).

Hold Time - The amount of time after the end of an activity signal during which some other signal must be stable to ensure the achievement of the correct final state.

IEEE Standard 488 Bus - See General-Purpose Interface Bus.

Immediate Addressing - An addressing method in which the operand is part of the instruction itself.

Immediate Data - Data that is part of the instruction that uses it.

Implied (or Inherent) Addressing - The operation code itself specifies all the required addresses.

In-Circuit Emulator - A device that allows a prototype to be attached to a development system for testing and debugging purposes.

Index Register - A register that can be used to modify memory addresses.

Indexed Addressing - An addressing method in which the address included in the instruction is modified by the contents of an index register in order to find the actual address of the data.

Indirect Addressing - An addressing method in which the address of the data, rather than the data itself, is in the memory location specified by the instruction.

Input/Output (Section) - The section of the computer that handles communications with external devices.

Instruction - A group of bits that defines a computer operation and is part of the instruction set.

Instruction Cycle - The process of fetching, decoding, and executing an instruction.

Instruction Execution - The process of performing the operations indicated by an instruction.

Instruction (Execution) Time - The time required to fetch, decode, and execute an instruction.

Instruction Fetch - The process of addressing memory and reading an instruction word into the CPU for decoding.

Instruction Length - The number of words of memory needed to store a complete instruction.

Instruction Repertoire - See Instruction Set.

Instruction Set - The set of general-purpose instructions available with a given computer--that is, the set of inputs to which the CPU will produce a known response during the instruction fetch cycle.

Integrated Circuit (IC) - A complete circuit on a single substrate or chip.

I^2L - Integrated-injection logic, a bipolar technology that uses only transistors (both vertical and lateral) to provide moderate speed, low power consumption, and high density.

Intelligent Terminal (or Smart Terminal) - A terminal that has some data processing capability or local computing capability.

Interpreter - A program that fetches and executes instructions written in a high-level language. An interpreter executes each instruction as soon as it reads the instruction; it does not produce an object program, as a compiler does.

Interrupt - A computer input that temporarily suspends the normal sequence of operations and transfers control to a special routine.

Interrupt-Driven System - A system that depends on interrupts to handle input and output or that idles until it receives an interrupt.

Interrupt Mask (Interrupt Enable) - A mechanism that allows the program to specify whether interrupts will be accepted.

Interrupt Service Routine - A program that performs the actions required to respond to an interrupt.

Inverter - A logic device that complements the input.

Isolated Input/Output - An addressing method for I/O ports that uses an addressing system distinct from that used by the memory section.

Jump Instruction - An instruction that places a new value in the program counter, thus departing from the normal one-step incrementing. Jump instructions may be conditional; that is, the new value may only be placed in the program counter if certain conditions are met.

Jump Table - A table that contains the addresses of routines to which the computer can transfer control.

K - 2^{10} or 1024 words, a unit of memory.

Keyboard - A collection of key switches.

Keyboard Encoder - A device that produces a unique output code for each possible closure on a keyboard.

Keyboard Scan - The process of examining the rows and columns of a matrix keyboard to determine which keys have been pressed.

Kilobit - 1000 bits.

Label - A name attached to a particular instruction or statement in a program that identifies the location in memory of the object code or assignment produced from that instruction or statement.

Large-Scale Integration (LSI) - An integrated circuit with complexity equivalent to over 100 ordinary gates.

Latch - A temporary storage device controlled by a timing signal. The contents of the latch are fixed at their current values by a transition of the timing signal (clock) and remain fixed until the next transition.

Light-Emitting Diode (LED) - A semiconductor device that emits light when biased in the forward direction.

Linear Select - Using coded bus lines individually for selection purposes rather than decoding the lines. Linear select requires no decoders but allows only "n" separate devices to be connected rather than 2^n, where n is the number of lines.

Linking Loader - A loader that will enter a series of programs and subroutines into memory and provide the required interconnections.

Loader - A program that reads a user or system program from an input device into memory.

Logic Analyzer - A test instrument that detects and displays the state of parallel digital signals.

Logic Design - Design using digital logic circuits.

Logic Shift - A shift operation that places zeros in the empty bits.

Logical Sum - A bit-by-bit EXCLUSIVE-ORing of two binary numbers.

Lookahead Carry - A device that forms the carry bit from a binary addition without using the carries from each bit position.

Loop - A self-contained sequence of instructions that the processor repeats until a terminal condition is reached. A conditional jump instruction can determine if the loop should be continued or terminated.

Low-Level Language - A language in which each statement is directly translated into a single machine language instruction. See Assembly Language and Machine Language.

Low-Power Schottky TTL - A low-power variant of standard TTL.

Machine Code - See Machine Language.

Machine Cycle - The basic CPU cycle. One machine cycle is the time required to fetch data from memory or execute a single-word operation.

Machine Language - The programming language that the computer can directly understand with no translation other than numeric conversions. A machine language program can be loaded into memory and executed. The value of every bit in every instruction in the program must be specified.

Macro - A name that represents a sequence of instructions. The assembler replaces a reference to the macro with a copy of the sequence.

Macroassembler - An assembler that has facilities for macros.

Macroinstruction - An overall computer instruction fetched from the main memory in a microprogrammed computer.

MBM - Magnetic Bubble Memory.

Majority Logic - A combinational logic function that is true when more than half the inputs are true.

Mark - The one state on a serial data communications line.

Mask - (1) A glass photographic plate that defines the diffusion patterns in integrated circuit production. (2) A bit pattern that isolates one or more bits from a group of bits.

Maskable Interrupt - An interrupt that the system can disable.

Matrix Keyboard - A keyboard in which the keys are connected in rows and columns.

MTBF - Mean time between failures.

MTTR - Mean time to repair.

Medium-Scale Integration (MSI) - An integrated circuit with a complexity of between 10 and 100 gates.

Megabit - One million bits.

Memory (Section) - The section of a computer that serves as storage for data and instructions. Each item in the memory has a unique address that the CPU can use to fetch it.

Memory Address Register (or Storage Address Register) - A register that holds the address of the memory location being accessed.

Memory-Mapped Input/Output - An addressing method for I/O ports that uses the same addressing system as that used by the memory section.

Meta-Assembler - An assembler for which the input instruction patterns can be defined and that can, therefore, assemble programs for different computers.

Microassembler - An assembler specifically designed for writing microprograms.

Microcomputer - A computer whose CPU is a microprocessor. A microprocessor plus memory and input/output circuitry.

Microcontroller - A microprogrammed control system without arithmetic capabilities.

Microinstruction - One of the words in a control memory--that is, one of the organized sequence of control signals that form the instructions at the control level.

Microprocessor - The central processing unit of a small computer, implemented on one or a few LSI chips.

Microprocessor Analyzer - A piece of test equipment that can be used to trace and debug the operations of a microprocessor.

Microprogram - A program written at the control level and stored in a control memory.

Microprogrammable - Having a microprogrammed control function that the user can change. That is, the user can add, enter, or replace microprograms.

Microprogrammed - Having the control function implemented through microprogramming.

Microprogramming - The implementation of the control function of a processing system as a sequence of control signals that is organized into words and stored in a control memory.

Mnemonics - Symbolic names or abbreviations for instructions, registers, memory locations, etc., which suggest their actual functions or purposes.

Modem - Modulator/demodulator, a device that adds or removes a carrier frequency, thereby allowing data to be transmitted on a high-frequency channel or received from such a channel.

Modular Programming - A programming method whereby the entire task is divided into logically separate sections or modules.

Monitor - A simple operating system that allows the user to enter or change programs and data, to run programs, and to observe the status of the various sections of the computer.

Monostable Multivibrator (or One-Shot) - A device that produces a single pulse of known length in response to a pulse input.

MOS - Metal-oxide semiconductor, a semiconductor process that uses field-effect transistors in which the current is controlled by the electric field around a gate.

Multiplexer (or Selector) - A device that selects one of several possible inputs to be placed on a time-shared output bus according to the state of the select inputs.

Multiprocessing - Utilizing two or more processors in a single system, operating out of a common memory.

MAC - Multiplier-accumulator.

Nanosecond - 10^{-9} second, abbreviated ns.

Negative Logic - Circuitry in which a logic zero is the active or ON state.

Nesting - Constructing subroutines or interrupt service routines so that one transfers control to another and so on. The nesting level is the number of transfers required to reach a particular routine without returning.

Nibble - A sequence of four bits operated on as a unit.

N-Key Rollover (NKRO) - Resolving any number of simultaneous key closures into consecutive output codes.

NMOS - N-channel metal-oxide semiconductor, a logic family that uses N-channel MOS field-effect transistors to provide high density and medium speed.

Noise Margin - The noise voltage required to make logic circuits malfunction.

Nondestructive Readout (NDRO) - The contents of the device can be determined without changing those contents.

Nonmaskable Interrupt - An interrupt that the system cannot disable.

Nonvolitile Memory - A memory that does not lose its contents when power is removed.

No-Op (or No Operation) - An instruction that does nothing other than increment the program counter.

Object Program (or Object Code) - The program that is the output of a translator program, such as an assembler or compiler. Usually a machine language program ready for execution.

Octal - Number system with base 8. The digits are the decimal numbers 0 through 7.

Offset - A number that is to be added to another number to calculate an effective address.

One-Address Instruction - An instruction in which only one data address must be specified. The other data, if necessary, is presumed to be in the accumulator.

One's Complement - A bit-by-bit logical complement of a binary number.

One-Shot - See Monostable Multivibrator.

On-Line System - A computer system in which information reflecting current activity is introduced as soon as it occurs.

Open-Collector Output - A special output that is active-low but not high. Such outputs can be wire-ORed to form a bus employing negative logic.

Operating System - System software that controls the overall operation of a computer system and performs such tasks as memory allocation, input and output distribution, interrupt processing, and job scheduling.

Operation Code (Op Code) - The part of an instruction that specifies the operation to be performed during the next cycle.

Optoisolator - Semiconductor device consisting of an LED and a photodiode or phototransistor in close proximity. Current through the LED causes internal light emission that forces current to flow in the phototransistor. Voltage differences have no effect because the devices are electrically separated.

Overflow Bit - A status bit that is one if the last operation produced a two's complement overflow.

Overlay - The section of a program that is actually resident in memory at a particular time. A large program can be divided into overlays and run on a computer having limited memory but backup storage for the rest of the program.

Page - A subdivision of the memory section.

Page Zero - The first page of memory; the most significant address bits (or page number) are zero.

Parallel - More than one bit at a time.

Parity - A 1-bit code that makes the total number of one bits in the word, including the parity bit, odd (odd parity) or even (even parity).

Parity Bit - A status bit that is one if the last operation produced a result with even (if even parity) or odd (if odd parity) parity.

Passing Parameters - See Subroutine.

Pipelining - Overlapping cycles so that different parts of consecutive cycles are performed at the same time.

PL/I - Programming Language 1, a high-level language developed by IBM that combines many of the features of earlier languages, such as ALGOL, COBOL and FORTRAN. Many versions exist for microprocessors, such as PL/M, MPL, SM/PL, and PLμS.

PMOS - P-channel metal-oxide semiconductor, a logic family that uses P-channel MOS field-effect transistors to provide high density and low speed.

Pointer - Register or memory location that contains an address rather than data.

Polling - Determining the state of peripherals or other devices by examining each one in succession.

Pop (or Pull) - Remove an operand from a stack.

Port - The basic addressable unit of the computer input/output section.

Power-On Reset - A circuit that automatically causes a RESET signal when the power is turned on, thus starting the system in a known state.

Pre-empted Process - A process which, because of scheduling policy, must relinquish the processor to another process.

Printed Circuit Board (PC Board) - A circuit board in which the connections are made by etching with a mask.

Priority Interrupt System - An interrupt system in which some interrupts have precedence over others--that is, will be serviced first or can interrupt the others' service routines.

Procedure-Oriented Language - See High-Level Language.

Program - A sequence of instructions properly ordered to perform a particular task.

Program Counter - A register that specifies the address of the next instruction to be fetched from program memory.

Program Library - A collection of debugged and documented programs.

Programmable Interface - An interface device that can have its active logic structure varied under program control.

Programmable Logic Array (PLA) - An array of logic elements that can be programmed to perform a specific logic function; like a ROM except that only certain addresses are decoded.

Programmable Timer - A device that can provide various timing modes and intervals under program control.

Programmed Input/Output (I/O) - Input/output performed under program control without using interrupts or direct memory access.

PROM - Programmable read-only memory, a memory that cannot be changed during normal operation but that can be programmed by the user under special conditions. The programming is generally not reversible.

PROM Programmer - A piece of equipment that is used to change the contents of a PROM.

Prototyping System (or Development System) - A hardware system used to breadboard a computer-based product. Contains the computer plus the software and hardware required for efficient development.

Pseudo-Operation (or Pseudo-Instruction) - An assembly language operation code that directs the assembler to perform some action but does not result in a machine language instruction.

Pull - See Pop.

Pullup Resistor - A resistor connected to the power supply that ensures that an otherwise open circuit will be at the voltage level of the power supply.

Pulse Generator - A device that produces a single pulse or a series of pulses of predetermined length in response to an input signal.

Push - Enter an operand into a stack.

Queue (or FIFO) - A set of registers or memory locations that are accessed in a first-in, first-out manner. That is, the first data entered into the queue will be the first data read.

RAM - Random-access (read/write) memory, a memory that can be both read and altered (written) in normal operation.

Random Access - All internal storage locations can be accessed in the same amount of time.

Real Time - In synchronization with the actual occurrence of events.

Real Time System - A real-time system is one which reacts to the environment by receiving data, processing it, and returning results sufficiently quickly to affect the reaction to the environment at that time.

Real Time Clock - A device that interrupts a CPU at regular time intervals.

Recursive Subroutine - A subroutine that calls itself as part of its execution.

Re-entrant Subroutine - A subroutine that can be executed correctly even while the same routine is being interrupted or otherwise held in abeyance.

Refresh - The process of restoring the contents of a dynamic memory before they are lost.

Register - A storage location used to hold bits or words inside the CPU.

Register Direct Addressing - An addressing method that is the same as direct addressing except that the address is in a register rather than in a memory location.

Relative Addressing - An addressing method in which the address specified in the instruction is the offset from a base address. The base address may be the contents of the program counter or a base register. Relative addressing allows programs to be easily relocated in memory.

Relocatable - Can be placed in any part of memory without changes--that is, a program that can occupy any set of consecutive memory addresses.

Reset - A signal that starts a system in a known state.

Resident Software - Software that can run on the computer itself, unlike cross-assemblers or cross-compilers, which must run on another computer.

Ripple Carry - Forming the carry bit from a binary addition by using the carries from each bit position.

ROM - Read-only memory, a memory that contains a fixed pattern of data permanently defined as part of the manufacturing process.

ROM Simulator - A device that allows read/write memory to act like ROM during system development; the simulator usually has special display and debugging features.

Routine - A program or subprogram.

RS Flip-Flop - A flip-flop that can be placed in the 1 state by a signal on the SET input or in the 0 state by a signal on the RESET input.

RS-232 - A standard interface for the transmission of serial digital data.

Schmitt Trigger - A circuit used to produce a single, sharp transition (i.e., a pulse) from a slowly changing input.

Schottky TTL - A high-speed variant of standard TTL.

Scratch-Pad Memory - Memory locations or registers that are used to store temporary or intermediate results.

Second Source - A manufacturer who supplies a device or product originated by another manufacturer.

Self-Assembler - An assembler that runs on the computer for which it assembles programs.

Self-Checking Number - A number in which some of the digits serve to check for possible errors in the other digits and do not contain any additional information.

Self-Test - A procedure whereby a system checks the correctness of its own operation.

Separate I/O - Uses different lines for input and output.

Sequencer - A device that controls the ordering in time of the states of a system or the order in which instructions are executed.

Serial - One bit at a time.

Serial-Access - A storage device (such as a magnetic tape) from which data can only be reached or retrieved by passing through all intermediate locations between the desired one and the currently available one.

Set - Make state a logic one.

Setup Time - The time, prior to a clock transition, during which data must be stable for proper operation.

Seven-Segment Code - The code required to represent decimal digits or other characters on a seven-segment display.

Seven-Segment Display - A display made up of seven separately controlled elements that can represent decimal digits or other characters.

Shift Register - A clocked device that moves its contents one bit to the left or right during each clock cycle.

Sign Bit - The most significant bit of a register or memory location; a status bit that is one if the most significant bit of the result of the previous operation was one.

Sign Extension - The result of a right arithmetic shift that copies the sign bit into the succeeding less significant bits.

Sign-Magnitude Number - A number in which the most significant bit represents the sign or polarity and the remaining bits represent the magnitude.

Signal Conditioning - Making a signal compatible with the input requirements of a particular device through buffering, level translation, amplification, etc.

Signature Analysis - A method whereby faults can be found in bus-oriented digital systems by examining the time histories of signals at particular nodes.

(Software) Simulator - A computer program that follows the actions of a system in detail and that can be used for debugging or testing.

Sink Current - The ability of a device to accept current from external loads.

Small-Scale Integration (SSI) - An integrated circuit with a complexity of ten gates or less.

Snapshot - Record of the entire state of a system at a particular point in time.

(Computer) Software - Computer programs.

Software Interrupt - See Trap.

SOS - Silicon-on-sapphire, a faster MOS technology that uses an insulating sapphire substrate.

Source Program - Computer program written in an assembly or high-level language.

Space - The zero state on a serial data communications line.

SPDT Switch - Single-pole, double-throw switch with one common line and two output lines.

SPST Switch - Single-pole, single-throw switch with one common line and one output line.

Stack - A sequence of registers or memory locations that are used in a last-in, first-out manner--that is, the last data entered is the first to be removed and vice versa.

Stack Addressing - An addressing method whereby the data to be used is in a stack.

Stack Frame - A contiguous data area allocated for every activation of a routine; holds parameter values, local variables, temporary variables and return linkage information.

Stack Pointer - A register or memory location that is used to address a stack.

Stand-Alone System - A computer system that does not require a connection to another computer.

Standard Teletypewriter - A teletypewriter that operates asynchronously at a rate of ten characters per second.

Standby (or Quiescent) Power - The amount of power required to maintain the contents of a memory when it is not being accessed.

Start Bit - A one-bit signal that indicates the start of data transmission by an asynchronous device.

State Counter - A counter that contains the number of states that have occurred in the current operation.

Static Memory - A memory that does not change its contents without external causes, opposite of dynamic memory.

Status Bit - See Condition Code.

Status Register (or Status Word) - A register whose contents reflect the current status of the computer; may be the same as condition code register.

Stop Bit - A one-bit signal that indicates the end of data transmission by an asynchronous device.

Strobe - A one-bit signal that identifies or describes another set of signals and that can be used to clock or enable a register.

Structured Programming - A programming method whereby all programs consist of structures from a limited but complete set; each structure should have a single entry and a single exit.

Subroutine - A subprogram that can be reached from more than one place in a main program. The process of passing control from the main program to a subroutine is a "Subroutine Call" and the mechanism is a "Subroutine Linkage." The data and addresses that the main program makes available to the subroutine are "Parameters," and the process of making them available is called "Passing Parameters."

Subroutine Call - See Subroutine.

Synchro-To-Digital Converter - A device that converts an analog angle to a corresponding digital value.

Synchronous Operation - Operating according to an overall timing source, i.e., at regular intervals.

Synchronization - Making two signals operate according to the same clocking signal.

Syntax - The rules governing sentence or statement structure in a language.

Teleprinter - See Teletypewriter.

Teletypewriter (TTY) - A device containing a keyboard and a serial printer that is often used in communications and with computers.

Terminal - An input/output device at which data enters or leaves a computer system.

Time-Shared Bus - A bus that is used for different purposes at different times.

Top-Down Design - A design method whereby the overall structure is designed first and parts of the structure are subsequently defined in greater detail.

Trap - An instruction that forces a program to jump to a specific address, often used to produce breakpoints or to indicate hardware or software errors.

Tri-State (or Three-State) - Logic outputs with three possible states--high, low, and an inactive (high-impedance or open-circuit) state that can be combined with other similar outputs in a busing structure.

Tri-State Enable (or Select) - An input that, if not active, forces the outputs of a tri-state device into the inactive or open-circuit state.

TTL (Transistor-Transistor Logic) - The most widely used bipolar technology for digital integrated circuits. Popular variants include high-speed Schottky TTL and low-power Schottky (or LS) TTL.

TTL-Compatible - Uses voltage levels that are within the range of TTL devices and can be used with TTL devices without level shifting, although buffering may be necessary.

2-Key Rollover (2KRO) - Resolving two (but not more) simultaneous key closures into two consecutive output codes.

Two's Complement - A binary number that, when added to the original number in a binary adder, produces a zero result. The two's complement is the one's complement plus one.

Two's Complement Overflow - A situation in which a signed arithmetic operation produces a result that cannot be represented correctly--that is, the magnitude overflows into the sign bit.

Unbundling - Pricing certain types of software and services separately from the hardware.

ULA - Uncommitted Logic Array.

Universal Asynchronous Receiver/Transmitter (UART) - An LSI device that acts as an interface between systems that handle data in parallel and devices that handle data in asynchronous serial form.

Universal Synchronous Receiver/Transmitter (USRT) - An LSI device that acts as an interface between systems that handle data in parallel and devices that handle in synchronous serial form.

Urgency - The degree to which a process requires attention; determined by the processes' priority.

Utility Program - A program that provides basic functions, such as loading and saving programs, initiating program execution, observing and changing the contents of memory locations, or setting breakpoints and tracing.

UVPROM (or UVROM) - See EPROM.

Vectored Interrupt - An interrupt that provides the CPU with an identification code that the CPU can use to transfer control to the corresponding service routine.

VHSIC - Very-High-Speed-Integrated Circuits, project of DOD to develop VLSI signal processors.

Very-Large-Scale Integration (VLSI) - An integrated circuit (similar to LSI) with complexity equivalent to over 100,000 gate RAM chips by 1980 and million-gate chips by the year 2000.

Volatile Memory - A memory that loses its contents when power is removed.

Wired-OR - Connecting outputs together without gates to form a busing structure; requires special outputs of which only one is active at a time.

Word - The basic grouping of bits that the computer can manipulate in a single cycle.

Word Length - The number of bits in the computer word, usually the length of the computer's data bus and data and instruction registers.

Working Register - See General-Purpose Register.

Zero Bit - A status bit that is one if the last operation produced a zero result.

LIST OF REFERENCES

1. Bhattacharyya, J.C., "Criteria for Selection of Microprocessor for Military Applications," IETE, V. 24, No. 3/4 March/April 1978.

2. Leventhal, L.A., Introduction to Microprocessors: Software, Hardware Programming. Prentice-Hall, 1978.

3. Burr, W.E. and Coleman, A.H. (eds.), "Final Report of the CFA Selection Committee," ECOM, September 1977.

4. Eckhouse, R.H., Jr. and Morris, L.R., Minicomputer Systems, Organization, Programming and Applications (PDP-11). Prentice-Hall, 1979.

5. Korn, G.A., Microprocessors and Small Digital Computer Systems for Engineers and Scientists. McGraw-Hill, 1977.

6. Gale, E.G. and Gremillion, L.L., "Price/Performance Patterns of U.S. Computer Systems," ACM, V. 22, No. 22, April 1979.

7. Madnick, E.S. and Donovan, J.J., Operating Systems. McGraw-Hill, 1974.

8. Schell, R.R., "Computer Security: The Achilles' Heel of the Electronic Air Force?" Air University Review, January-February 1979.

9. Electronic Design, "Bubbles--More Capacity, Components," Vol. 27, No. 24, 22 November 1979.

10. Prokop, J., Computers in the Navy, Naval Institute Press, 1976.

11. Burr, E.W. and Smith, R.W., "Comparing Architectures," Datamation, February 1977.

12. Johannsen, C., "Choosing Software," Computerworld, 28 January and 4 February 1980.

13. Bernhard, R., "Computers: Emphasis on Software," IEEE Spectrum, January 1980.

14. Snigier, P., "Magnetic Bubble Memories," Digital Design, December 1979.

15. Computerworld, "Multitasking Business Micro Handles 48 Users," January 28, 1980, p. 67.

16. Reghizzi, C.S., Corti, P., and Darpa, A., "A Survey of Microprocessor Languages," Computer, January 1980.

17. Glass, L.R., "Ada Programming Language Gets Fine-Tuning," Computerworld, November 19, 1979, p. 21.

18. Hurwitz, J. and Klunucan, P., "Mini-Micro World," Mini-Micro Systems, December 1979, p. 35.

19. Bursky, D., "Microprocessor Data Manual," Electronic Design 24, 22 November 1979.

20. Cushman, H.R., "μP/μC Chip Directory," EDN, 20 October 1979.

21. Newman, M.W. and Sproull, F.R., Principles of Interactive Computer Graphics, McGraw Hill, 1979.

22. Orr, J., "Interactive Computer Graphics Systems," Mini-Micro Systems, December 1979.

23. Weitzman, C., Distributed Micro/Minicomputer Systems, Prentice-Hall, 1980.

24. Computerworld, "Japanese Ready Semi Industry for VLSI," 24 March 1980, p. 59.

25. Christiansen, D., "Technology '80," IEEE Spectrum, January 1980, p. 30.

26. Nyman, H.T. and Hazan, P., "Dedicated Microcomputers for Defense and Industry," Computer, February 1979, pp. 85-88.

27. Auerbach Publishers, Inc., "Checklist for Evaluating Minicomputer Vendor Policies," (2-04-11), 1977.

28. Lucas, C.H., Jr., "Performance Evaluation and Monitoring," Computer Surveys, Vol. 3, No. 3, September 1971.

29. Summey, W.L., "VLSI with a Vengeance," IEEE Spectrum, April 1980, p. 24.

30. Kodres, U.R., Hamming, W.R., Buttinger, D.J. Jones, R.C., A Study of Alternatives for VSTOL Computer Systems, Naval Postgraduate School, Monterey, CA, NPS52-78-001, April 1978.

31. NAVAIR 01-85 ADF-2-10.1.1 Technical Manual, Navy Model A-6E Aircraft. Integrated Weapons System Theory, NAVAIR, 1 September 1971.

32. Hirsh, P., "Development of Ada Gets Mixed Reviews," Computerworld, 26 May 1980, p. 20.

33. Palmer, J., Nave, R., Wymore, C., Koehler, R., and McMinn, C., "Making Mainframe Mathematics as Accessible to Microcomputers," Electronics, 8 May 1980, pp. 114-121.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 52 Department of Computer Science Naval Postgraduate School Monterey, California 93940	1
4.	Professor Uno Kodres, Code 52 Ko Department of Computer Science Naval Postgraduate School Monterey, California 93940	1
5.	Professor B. J. Carey, Code 52 Ca Department of Computer Science Naval Postgraduate School Monterey, California 93940	2
6.	Hellenic Navy General Staff Education Department/3 Stratopedon Papagou Holargos, Athens, GREECE	3
7.	LCDR M. Mastrakas HN Asty Egyptioton 4A Kifisia, Athens, GREECE	5
8.	LCDR A. Koutsotolis HN SMC 2738 Naval Postgraduate School Monterey, California 93940	3
9.	Captain S.Tsavdaris HA SMC 2819 Naval Postgraduate School Monterey, California 93940	2


DAT
FILM