NASA CR-137889

June 30, 1976

COMPUTER TECHNOLOGY FORECAST STUDY FOR GENERAL AVIATION

Prepared under Contract No. NAS 2-8971

by

Charles L. Seacord

and

Darrel Vaughn

of

HONEYWELL INC.
Government and Aeronautical Products Division
St. Louis Park, Minn.

For

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

EXECUTIVE SUMMARY

BACKGROUND

NASA has underway a multi-year, multi-faceted program to investigate and develop potential improvements in airframes, engines, and avionics for general-aviation aircraft. As part of this overall program, a general-aviation avionics research and technology program has been initiated. It is the basic premise of this program that by using 1980 electronics technology, many significant improvements can be made in general-aviation avionics system design; for example, the use of digital computers to assist the pilot in performing the functions involved in general aviation, such as navigation, guidance, control, fuel management and communications. Studies have been initiated to define the advanced electronics technology available in the late 1970's and early 1980's upon which an advanced general-aviation avionics system design could be based. The objective of this study was to assemble information that will allow the Government to assess the trends in computer and computer/operator interface technology that may have application to general aviation in the 1980's and beyond.

SCOPE OF STUDY

The study assesses the current state of the art of computer hardware, predicts technical developments in computer hardware through the 1980's, identifies nonaviation large-volume users of computer hardware that might provide a source of low-cost components, and makes recommendations for further NASA work that would accelerate technical development and/or support high-risk developments.

The survey/study was conducted by the Government and Aeronautical Products and Systems and Research Divisions of Honeywell, Inc. Both these

groups are actively engaged in the development of advanced computers for aviation use, and much of the data in this report has been taken from various ongoing programs. In addition, surveys were made of recent technical and commercial data on all elements of computer-based systems, and leading manufacturers of applicable hardware were contacted on current and future technology.

To provide a basis for estimating what future general-aviation avionics requirements may be, a list of current general-aviation aircraft and typical sets of avionics was compiled and grouped into three general classes. The requirements of each class are discussed in regard to the effect on the amount and technical level of avionics required in the future.

Specifically, all types of semiconductor logic and memory devices that are beyond the basic research stage and appear to have some practical applications for general aviation are discussed in the report. Also, in the memory area, all feasible magnetic devices and schemes and several optical read-out/plastic medium arrangements are reviewed. In the case of display elements, several concepts that may have application to general aviation in addition to the familiar cathode ray tube (CRT), light-emitting diodes (LED), liquid crystal display (LCD), and incandescents,are covered.

The various available software generation techniques (e. g. machine and assembly languages, and compilers for higher-order languages) are reviewed for application to general aviation. Potential computer system architectures are described and comments on their use in general-aviation avionics are included. For the sake of brevity, much of the detailed background material on the various technologies is presented in a set of appendices.

An attempt has been made to avoid overstating the potential uses of the technologies; on the other hand, development in all the fields is proceeding rapidly, and the future status may well be better than indicated here. This

study evaluates in a qualitative manner the applicability of current and near-future computer technology to general-aviation avionics. The data should not be used as a basis for specific designs, but it is believed that the trends indicated are correct and useful in planning future development work.

ASSESSMENT OF FUTURE GENERAL-AVIATION AVIONICS REQUIREMENTS

Airplane/Avionics Classification

The term general aviation covers a wide range of aircraft varying from light single piston-engine planes to multi-engine turbofan executive transports. Obviously, the avionics requirements vary widely from one end of this scale to the other, and thus the need for advanced computer technology also varies. Accordingly, general-aviation aircraft have been separated into three classes, the characteristics of which are listed in Table S1. It is interesting to note that the avionics cost as a percentage of the total airplane cost is greatest for the single piston-engine Class C airplanes. This ratio coupled with the facts that 1) the higher relative cost is buying less capability and 2) avionics maintenance for the Class C airplane is probably relatively more expensive, make the provision of desirable avionics for the Class C airplance more critical than for the two higher-level aircraft. Although the cost of avionics for the lower-priced airplanes has the highest cost ratio, it seemed advisable to determine which technologies (type of hardware) are most important to overall avionics cost for all classes of airplanes. An estimate of the cost of the various elements of current general-aviation avionics is shown in Table S2. These figures show that no one type of hardware is dominant (except for the RF elements in the VOR/ILS and communication subsystem). Further, it is indicated that the primary subject of this study (i.e. the computer and memory) is not now more than about a third of the subsystem cost. Thus, even if the computer and memory costs decrease sharply as predicted, the cost of RF hardware, inertial sensors, and to some degree, displays will remain a problem unless digital electronics can be

TABLE S1. – SUMMARY OF AIRPLANE CHARACTERISTICS

Aircraft class	Averages: Airframe cost, $	Averages: Avionics cost, $	Cruise Performance	Mission	Potential computer applications
A) Multi-engine turbine	1 270 000	159 720	R = 1592 nmi V = 345 K T = 4.7 hrs	• Business, company crew • Third-level airline • Charter • Air freight	Flight stabilization, area navigation, flight management, fuel management computations, flight progress computations, display generation, built in test
B) Twin-engine piston	145 000	17 235	R = 1127 nmi Va = 208 K T = 5.4 hrs	• Pleasure • Business Small-business, owner operated Corporate, contract pilot Air taxi, passenger and freight	Flight stabilization, area navigation, flight management, fuel management computations, flight progress computations, display generation, built in test
C) Single-engine piston ≥ 200 hp	42 790	7 287	R = 832 nmi V = 154 K T = 5.4 hrs	• Pleasure • Business Small business, owner operated Air taxi, passenger and freight Charter, passenger and freight	Flight stabilization, area navigation, flight progress computations, display generation
R = Range V = Cruise Speed T = Endurance at Cruise Speed					

used to bring about a marked reduction in the cost of RF elements, sensors, and displays.

TABLE S2. – RELATIVE COSTS OF SUBSYSTEM ELEMENTS (GIVEN IN PERCENT OF TOTAL SYSTEM COST)[a]

Subsystem	Major assemblies					
	RF Elements	Computer and memory	Display	Inertial sensors	Servos	Controls
R-NAV	40	25	25			10
Autopilot/stability augmentation system, 3 axis		35		35	20	10
Flight director		25	30	40		5
INS		25	5	65		5
VOR/ILS and communication	75		10			15

[a] Each system is assumed independent, as flight directors and autopilots must be independent for Category II operation.

Future Avionics Requirements

In terms of data that can be sensed and processed, current avionics can perform all functions required for general-aviation operations. This is not to say, however, that no improvement is needed or forthcoming. Beyond the obvious needs for lower cost and better reliability, there is a very real need to reduce the pilot workload and required skill level. A higher-level Class A or B airplane has an instrument panel and controls similar to those of a commercial transport; it also needs a pilot with equivalent training and skill.

General requirements. – There is then a need for more automation and/or data processing to relieve the pilot of routine but attention-demanding tasks and to give him data suitable for decision making. To oversimplify, the pilot needs an airplane that will go where he points it (Reference 6 illustrates the importance of airplane handling qualities) and computers and displays that will tell him where to point it next. The general requirements for future general-aviation avionics then become:

- Perform the same tasks now being handled (i.e. communication, navigation, stabilization, and data display) with at least equal performance at lower cost of ownership.

- Process data from aircraft and ground sources into a display form immediately useful for pilot decision.

- Keep track of critical factors (e.g. flight time remaining, airspeed, and angle of attack) and display appropriate warnings.

These requirements translate into a need for improvement in cost and reliability of RF elements, sensors, and actuators plus a quantum jump in the use of digital computers and electronic displays. The functions in which digital computers will be increasingly involved are listed in the last column of Table S1. Improved displays will also be required for all of these functions except flight stabilization.

Logic, computing and memory electronics. — In terms of performance (speed, power required, and size), current digital hardware is adequate for all the functions and classes of aircraft shown in Table S1. Obviously, improvements in performance will occur and will be useful, but the most needed improvement is in cost. The actual technology level required for the three classes of aircraft will be essentially the same, but there will, of course, be a considerable difference in the amount of digital hardware. While most digital computing and data processing will be handled by applying one or more of the rapidly proliferating microprocessors and large-scale integrated circuit (LSIC) logic arrays, there will remain those applications in which custom LSIC will be more efficient and cost effective, particularly in low-cost, high-volume devices. The functions shown for Class C airplanes could probably be handled by two simple microprocessors and a few thousand words of memory, whereas the Class A airplane might use 10 microprocessors and 50,000 to 1,000,000 words of memory.

Input/output/display devices. -- Again, as for the computing electronics, the technical level of input, output, and display devices should not vary much

with aircraft class, but the complexity and versatility of these devices will. It is expected that the most widely used input device will be a multipurpose keyboard to input data for navigation, flight management, display format etc. The first widely used displays may be CRTs, but they will be displaced by one of the matrix-type flat-panel displays now under development. Whatever the technology, new formats presenting both command and situation data need to be developed to improve performance of the typical general-aviation pilot.

Computer architecture and software. — Because of the increasing performance and decreasing cost of microprocessors, the use of two or more digital computers in a suite of avionics will become common. In the simpler systems, these processors will be dedicated to particular tasks and will be completely independent, but as the system functions become progressively more complex, there will be a need first for data exchange, then for task sharing, and finally in redundant flight-critical applications, automatic reconfiguration to continue critical tasks and drop noncritical tasks in case of a processor failure.

STATUS AND PROJECTION

The following comments are based on information available in early 1976. Specific quantitative data is taken from that presented in the body of the report; general and qualitative remarks are based on the material in the appendices as well as the report itself. It should be remembered that the projections are based on technical and cost feasibility; the actual widespread application to general aviation will depend on many other factors such as the general economic situation, manufacturer's marketing decisions, and government regulations.

Logic and Memory

Current status. — Logic and computing devices technically satisfactory for use in general-aviation avionics are available today in the form of monolithic microprocessors or small assemblies of LSIC elements. Some equipment using microprocessors (primarily R-Nav) has been offered for about two years. Digital electronics is not yet being used in all feasible applications because of the time and investment required for manufacturers to switch to digital computing in current systems (e.g. autopilots and flight director computers) and because digital implementation of simple functions is not yet cheaper than the current analog versions. Also, the actual arithmetic processing elements form a minor part of a complete digital computer for an autopilot, for example. In current techniques, the processor and memory would occupy two cards of a six or eight card total, with the remainder being devoted to analog-to-digital-to-analog (A/D/A) conversion, input/output (I/O) conditioning, mode logic, and power supply. Because these functions (with the possible exception of the A/D/A) are peculiar to a specific design, they cannot be bought in standard monolithic LSIC form; thus, the majority of the computer must be made up of assemblies of smaller, standard MSIC, or custom LSIC chips must be designed and procured. Generally such custom LSIC becomes cost-effective whenever the quantities reach the 1000 to 5000 unit range (or more).

Nevertheless, the logic and memory technology for digitally implementing all the functions listed in Table I is available in a variety of forms. Currently, for logic and arithmetic processing, the most common forms of LSIC are complementary metal oxide semiconductor (CMOS) and transistor-transistor logic (TTL). CMOS offers more functions per chip, and TTL more inherent speed. Production devices in low-power Schottky TTL (which as the name implies use less power and generate less heat) are appearing in quantity and are being used wherever speed and performance is important. New technologies such as integrated injection logic (I^2L) and direct coupled transistor-transistor logic (DCT^2L) are in pilot production and show promise of attractive combinations of speed, power, and density (gates per unit area).

Projection. — The many semiconductor techniques will continue to develop, although CMOS and basic TTL are probably nearing a plateau. Low-power Schottky TTL and I^2L appear to offer potential for continued improvement in performance and reduction in cost. Figure S1 shows the history and a projection of the price/performance ratio for one card processors using MSIC and LSIC, as well as current and future 16-bit monolithic microprocessors. The ordinate is a figure of merit, cost (dollars) divided by word length (bits) and multiplied by throughput (KOPS), in which decreasing magnitude indicates improved cost effectiveness. Figure S2 is a similar plot showing the change in cost per bit for the digital memory, which is an integral part of all computers. These figures indicate that by 1980 the cost of the computing (processor plus memory) elements will be about one-fourth what they are now. A satisfactory computer for a three-axis autopilot (16-bit, 5 microseconds add time, and 4K memory) would cost about $800.00, considerably less than corresponding analog circuitry costs today. Inasmuch as the autopilot is one of the more demanding computing tasks, it is evident that it will be both technically and economically feasible to apply digital computing not only where analog circuits have been used, but in new applications where analog devices have been too complex and expensive. Another benefit of the change to digital electronics will be a much lower failure rate, thus increasing safety and decreasing maintenance costs. Data on currently operational digital avionics indicates an increase in mean time between failures (MTBF) of from 4 to 10 times that for similar analog units. This current digital equipment is built using MSIC. Because failure rate is roughly proportional to piece-part count, future equipment using LSIC (and thus fewer parts) will show even more improvement.

Input/Output/Display Devices

Current Status. — Digital inputs in the form of dedicated keyboards have come into use on R-Nav, inertial navigation, and communication equipment in the past few years. Some of the transport-level inertial navigation systems make limited multimode use of the input keyboard (and numerical display), but there are no current examples of use of a single input keyboard to serve several

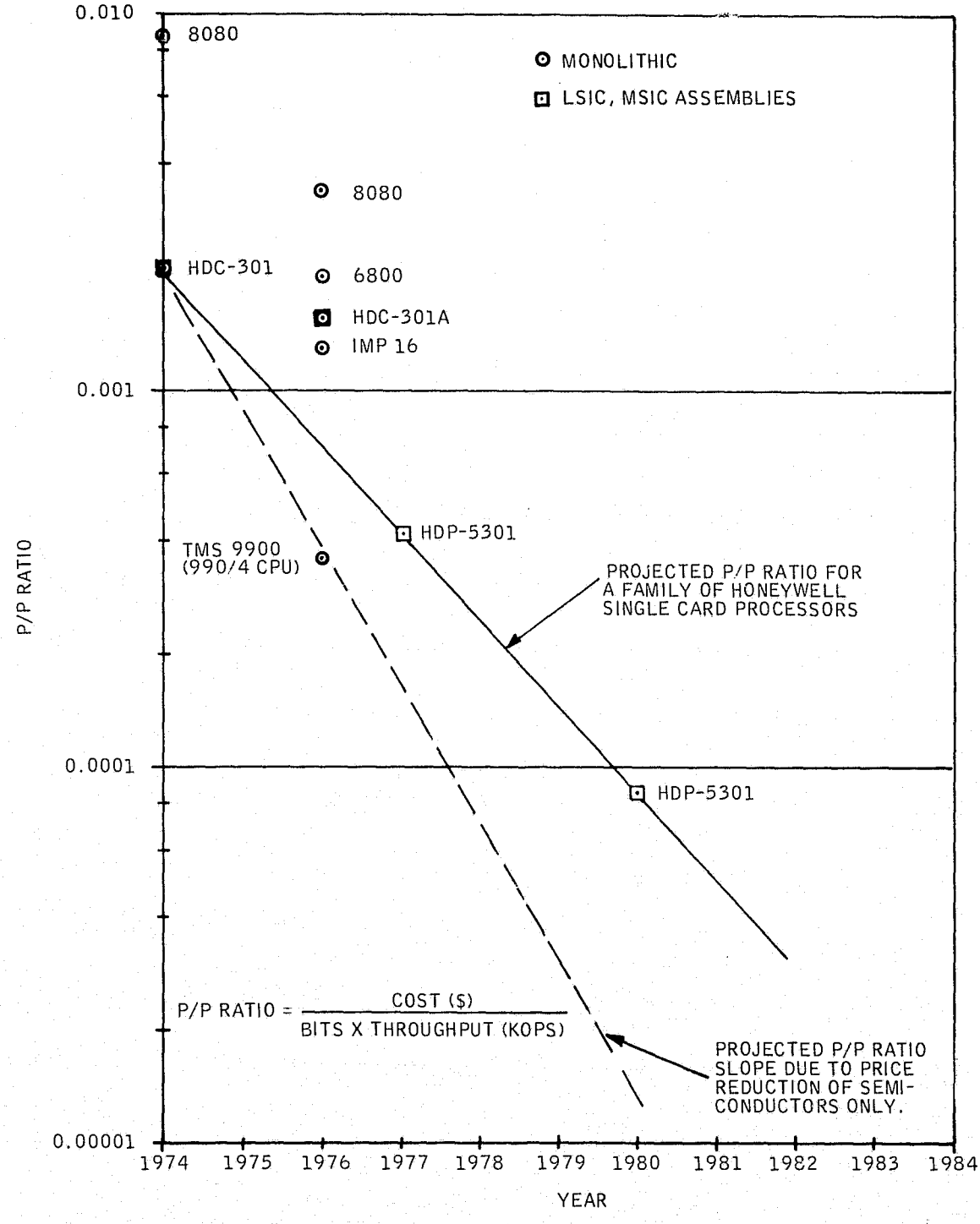
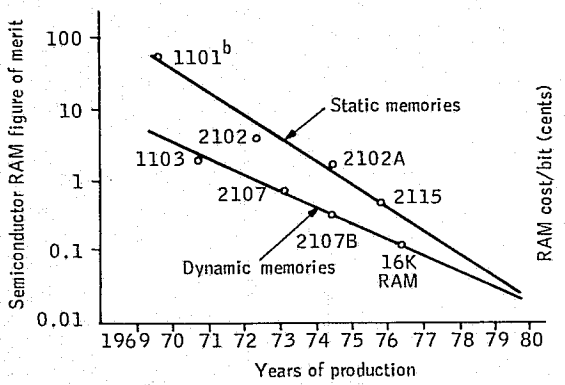


Figure S1. – Projected Price/Performance Ratio for a Family of Honeywell Processors Compared with Past and Present Devices

a. RAM figure of merit (M)[a] versus years of production



a. $M = \frac{\text{Area X Power Dissipation X Access Time}}{\text{Bits}}$

b. The Designations Above Identify a Family of Intel Memory Devices that are Typical of Current Production RAM's.

b. Cost/bit curve

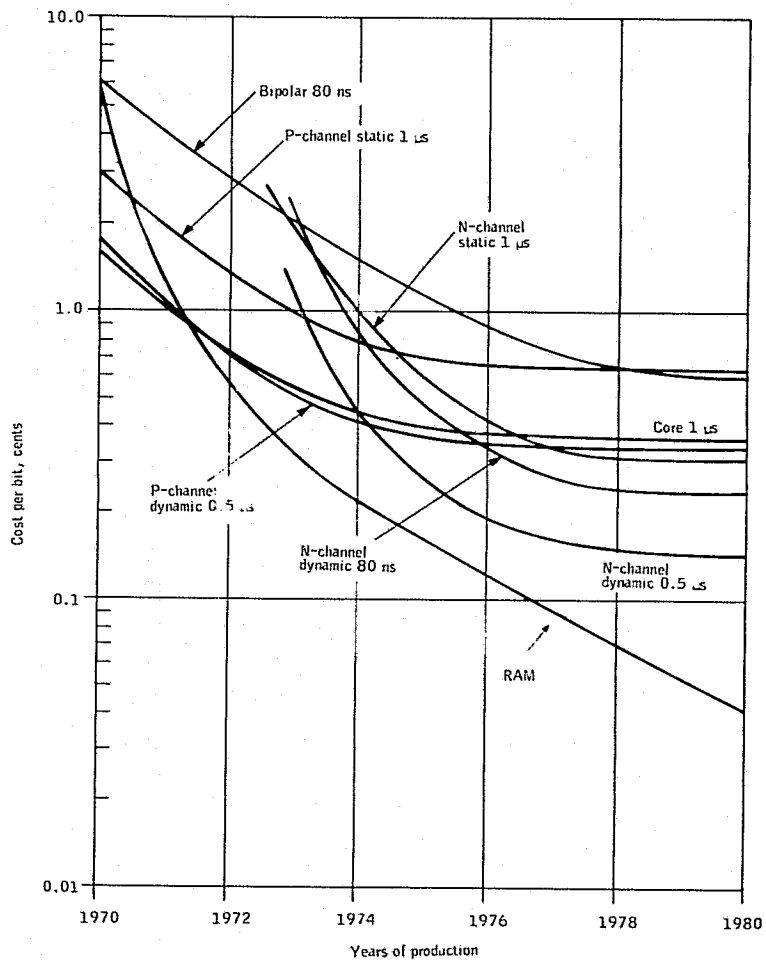


Figure S2. – Graphs Showing Progress in MOS Technology

general-aviation subsystems. Nevertheless, the basic keyboard and accompanying digital processor technology is available and is being used on computer terminals, desk calculators, and various laboratory instruments. For multipurpose inputs to be effectively utilized in general-aviation avionics, the navigation, control and display subsystems must be designed as an integrated system or at least designed to standardized data interface specifications.

In the case of displays, all current equipment except incandescants, light-emitting diode (LED) numerical readouts, and weather radar scopes are of the traditional electromechanical type. These, except for the flight director indicator (FDI), are usually dedicated, single-format instruments using meter movements or pressure bellows to move pointers, bars, or scales. FDI's may combine meter movements with servoed or gyro actuated indicators. They are complex, costly to buy, and relatively expensive to maintain. The factors of multiple engines, the various types of navigation aids, redundancy requirements, and the dedicated nature of the displays, all combine to make the instrument panels of the higher-level airplanes crowded and difficult to monitor. Currently, display technology from other uses does not offer solutions to general-aviation display problems. The cathode ray tube (CRT) is the nearest to being immediately useful, but has disadvantages in size, power, and cost unless it can be used to replace several current instruments in a multiformat applications. None of the various flat-plate matrix displays under development are ready for application to general aviation.

Projection. — Multiple-use input keyboards will be increasingly applied as more subsystems are converted to digital implementation. The keys themselves will be identified with integral digitally generated characters, which will be controlled by dedicated mode-select keys. The mode-select keys will control, through microprocessor logic, the activation of the computer routines needed for each mode. Inputs of large amounts of data into avionics computers for particular routes or functions will probably continue to be done with magnetic cards or tape cassettes. If there is a need for high-

speed data memory to be changed, it can be done by using replaceable semiconductor memory cards.

By 1980, electronic displays will be adapted for general aviation. The first application will probably be a CRT electronic attitude director indicator (EADI) with two or more formats, each tailored to a specific flight phase (for example, landing or cruise) or perhaps a combined and multiformat engine/fuel management display. By the mid 1980's, the use of CRT's will be superceded by one of the various digitally driven flat-plate matrix displays. An improtant part of future development will be the combination and/or redesign of traditional formats (many influenced by the predominant meter movement implementation) into new formats that fully exploit the new display techniques in meeting human factors requirements.

Computer Architecture and Software

Current Status. - The digital computers now in use in general aviation are comparatively simple dedicated processors, each with its own memory. There are no redundant computers with automatic error detection and reconfiguration used in general aviation; the technology for such systems is well developed and coming into use in military avionics, but as yet computers have not been used in flight-critical applications in general aviation. Further, because of the limited use of digital computers, there is no use of distributed processing or even any appreciable data exchange between computers.

In regard to software technology, most, if not all, current general-aviation digital computers were programmed in assembly or machine language. Higher-order language compilers are not available (or needed) for most of the rather simple digital processors now in use. Compilers are being developed for the newer, higher performance microprocessors and minicomputers, but they are not yet in widespread use.

Projection. - Computer architecture will become more complex as 1)

dedicated computers are interconnected to exchange data and perform new tasks 2) digital computers are applied to flight critical tasks (for example, autoland control), which require automatic failure detection and reconfiguration to continue operation and 3) as built-in-test techniques are employed. The basic technology for all these developments is available, primarily from military avionics, and it will be a process of evolution, pushed by competition and limited by investment, for them to be applied to general aviation. True distributed processing with load sharing and task redistribution will follow the simpler data exchange, self test, and parallel redundancy techniques and probably will not be used before 1990.

The question of applying higher-order language (and hardware floating point computation) is primarily one of a tradeoff between development costs (software) and production hardware costs. It appears probable that for some time the simpler computers will remain fixed point and be programmed in assembly language. Only when the systems become complex and the digital hardware inexpensive will the development cost overshadow the unit cost of production hardware. There may, however, be considerable use of higher order language in developing more complex computer systems with a conversion to assembly language when functional design and algorithm development is complete.

RECOMMENDATIONS

The information gathered in the study indicates that current and continuing developments in semiconductor technology will satisfy all requirements for computing hardware (arithmetic logic, A/D/A converters, and memory) for general-aviation avionics in the 1980's. There are also current developments in multiprocessor techniques, fault-tolerant computers, software, sensors, and electronic displays that appear to have application to general-aviation avionics, but which will require either continued development and/or adaptation specifically oriented to general-aviation requirements. Also, there is

a need to establish overall functional requirements and candidate configurations for avionic systems that will materially improve the safety and utility of general-aviation aircraft.

To meet these needs, it is recommended that NASA:

- Consider sponsoring programs to develop more cost-effective I/O devices (particularly inertial sensors and command and situation displays) that are designed specifically for general-aviation application
- Pursue work to define the functional requirements for future general-aviation avionic systems and the system configuration(s) that will meet the improved requirements

CONTENTS

ILLUSTRATIONS

TABLES

FOREWORD

This report covers the work conducted under NASA Contract Number NAS2-8672, "Computer Technology Forecast Study for General Aviation". The program was administered under the direction of the Aircraft Guidance and Navigation Branch of the Flight Systems Research Division of NASA-Ames Research Center.

The report covers work performed from August 1975 to April 1976 by the Government and Aeronautical Products Division of Honeywell Inc. under the direction of Charles L. Seacord. The principal contributors were Robert Berg, Douglas Jenson, Stephen Nuspl, Patrick Pratt, Charles Seacord, Ernest Silagyi, and Darrel Vaughn. Mr. Ralph Crumrine acted as a consultant on current general-aviation aircraft and avionics.

COMPUTER TECHNOLOGY FORECAST STUDY FOR GENERAL AVIATION

SECTION 1
SUMMARY

A survey/study to define the type of computer hardware that will be available and suitable for use in general-aviation aircraft avionics during the 1980's has been made. The purposes of this study are:

- To assess the current state of the art in computer technology to define what is suitable for use in low-cost avionics.
- To identify those industrial and commercial uses of computer hardware that would tend to provide sources of low-cost computers.
- To assess the direction and extent of future computer technology developments that would aid the application of electronic computing to low-cost avionics.
- To evaluate the risk associated with the applicable computer technologies and make recommendations for future NASA effort that would aid the development of more effective avionics for general-aviation aircraft.

The study was based on data available in early 1976. Much of the information presented, in particular that concerning semiconductor logic and memory devices, was gathered from sources within Honeywell Inc. Additional information on specialized devices and recent developments was obtained from visits to semiconductor integrated circuit manufacturers, a survey of recent technical publications and applicable manufacturer's device specifications. The results are summarized in this report, which contains condensed examples of the characteristics of the various types of computer hardware. More detailed information on the various technologies is presented in accompanying appendices.

From the study, the following conclusions have been made.

- Current, and probable future, LSIC (large-scale integrated circuit) semiconductor devices will satisfy all general-aviation avionics computing requirements. The most promising bipolar forms for future applications are integrated injected logic (I^2L) and direct coupled transistor-transistor logic (DCT^2L). A number of alternate structures have been postulated to further improve the characteristics of MOS; of these, VMOS and CMOS/SOS appear the most likely technologies for 1980's applications.

- The advantages of LSIC technology, namely, low cost and large volume, are gainfully exploited by two important features of the microprocessor. The microprocessor can be used as a universal standard component for many general-aviation avionics applications and permits the easy formation of multiprocessor networks. The increasing capability and decreasing cost of microprocessors is slowing the application of data bus techniques and encouraging the use of dedicated computers. In addition, low computer costs will promote use of parallel redundancy and extensive self check to such an extent that the failure probability of the computer function will become negligible in relation to that for the complete system.

Among all the memory technologies — core, semiconductor, plated wire, disk, drum and more recent exotic memory types — only semiconductors are projected to show significant improvements in application to general-aviation avionics.

It appears probable that the cost of computing equipment per se (logic and memory) will represent a relatively small portion of a total general-aviation avionics system cost. Conversely, input/output (I/O) devices (for example, inertial sensors, displays, and control servos) are currently, and will continue to be, costly items due to the lack of nonaviation large-volume users of such devices and their need for tailoring to a specific system design.

It is recommended that NASA:

- Consider sponsoring programs to develop more cost - effective I/O devices (particularly inertial sensors, and command and situation displays) designed specifically for general-aviation application.

- Pursue work to define the functional requirements for future general-aviation avionic systems and the system configuration(s) that will meet the improved requirements.

SECTION 2
INTRODUCTION

BACKGROUND

NASA has underway a multi-year, multi-faceted program to investigate and develop potential improvements in airframes, engines, and avionics for general-aviation aircraft. Reference 1 contains a summary of past work in this area. As part of this overall program, a general-aviation avionics research and techology program has been initiated. It is the basic premise of this program that by using 1980 electronics technology, many significant improvements can be made in general-aviation avionics system design; for example, the use of digital computers to assist the pilot in performing the functions involved in general aviation, such as navigation, guidance, control, fuel management, and communications. Studies have been initiated to define the advanced electronics technology available in the late 1970's and early 1980's upon which an advanced general-aviation avionics system design could be based. The objective of this study was to assemble information that will allow the Government to assess the trends in computer and computer/operator interface technology that may have application to general aviation in the 1980's and beyond.

PURPOSE

The study assesses the current state of the art of computer hardware, predicts technical developments in computer hardware through the 1980's, identifies nonaviation large-volume users of computer hardware that might provide a source of low-cost components, and makes recommendations for

further NASA work that would accelerate technical development and/or support high-risk developments.

CONDITIONS AND SCOPE OF STUDY

The survey/study was conducted by the Aircraft Flight Systems group of the Government and Aeronautical Products Division of Honeywell, Inc. with the participation of the Computer Technology Group of the Systems and Research Division of Honeywell. Both these groups are actively engaged in the development of advanced computers for aviation use, and much of the data in this report has been taken from various on-going programs. In addition, surveys were made of recent technical and commercial data on all elements of computer-based systems, and leading manufacturers of applicable hardware were contacted on current and future technology.

In projecting future avionic requirements, considerable dependence was placed on interviews with active pilots and reviews of current general-aviation publications. In logic, memory, and display the range of technologies covered by the study is wide, covering several areas that currently appear to have little application to avionics, but are included because they may have application in the future.

Specifically, all types of semiconductor logic and memory devices that are beyond the basic research stage and appear to have some practical applications are included. In the memory area, all feasible magnetic devices and schemes and several optical readout/plastic medium arrangements were reviewed. In the case of display elements, several concepts in addition to the familiar cathode ray tube (CRT), light-emitting diodes (LED), liquid crystal display (LCD) and incandescents are covered.

The application of various approaches to generating computer software (e.g., machine and assembly language, compilers for higher-order

languages) has been considered in light of the peculiar requirements of the general-aviation avionics industry, and the manner in which the required computing capability is deployed in a total avionic system has been reviewed, particularly in regard to potential improvements in overall system integrity and ease of maintenance. Factors considered in this regard are busing, distributed versus central arithmetic and memory units, and the use of various self-check or fault-isolation techniques.

Because of the large volume of background material collected in the study, this report has a number of appendices which contain pertinent background information that may not be generally available as reference material. It should be noted that much of the data presented in the appendices concerning logic and memory elements has been taken directly from other projects or studies. Development work in this field is proceeding rapidly and there are advances being made continually. In general, the approach in projecting the future status of the various technologies has been conservative. Any quantitative study and specific system designs should be based on the latest information rather than that contained in the appendices to this report.

SIGNIFICANCE OF THIS STUDY

This study does not provide quantitative design data nor does it predict the configuration of future general-aviation avionic systems; the intent was to evaluate in a qualitative manner (backed up by quantitative data) the applicability of current and future computer technology to general-aviation avionics. It is believed that the conclusions reached are valid and rather obvious in regard to logic and memory devices; however, in the cases of sensors and displays, the direction of growth is uncertain. Even so, the data presented can be used as a basis for formulating succeeding studies and/or development work toward the ultimate objective.

SECTION 3
LIST OF ABBREVIATIONS AND ACRONYMS

Abbreviations:

A	ampere
cd	candela
ft-L	foot-lambert
hp	horsepower
hr	hour
Hz	hertz
K	knots
lm	lumen
m	meter
mil	1 mil = 0.001 in.
μsec	microsecond
msec	millisecond
n. mi.	nautical mile
nsec	nanosecond
sr	steradian
V	volts
W	watt

Acronyms:

A/D	analog to digital
ASC II	American standard code for information exchange
ATC	air traffic control
ATCRBS	air traffic control radar beam system
BEAMOS	beam addressed metal oxide semiconductor wiring
BORAM	block oriented random access memory
CAS	collision avoidance system
CCD	charge-coupled device
CHL	current-hogging logic(same as I^2L)
C^3L	complementary current logic (same as I^2L)

CML	current mode logic
CMOS	complementary metal oxide semiconductor
CRT	cathode ray tube
3D	triple diffusion device structure
D/A	digital to analog
DABS	discrete address beacon system
DCT^2L	direct coupled transistor - transistor logic
DIMOS	dielectric isolated metal oxide semiconductor
DME	distance measuring equipment
DMOS	double diffused metal oxide semiconductor
DTL	diode transistor logic
EAROM	electrically alterable read only memory
ECD	electrochromatic display
ECL	emitter-coupled logic
EFL	emitter follower logic
ELD	electroluminescent device
EPI	epitaxial collector structure
EPID	electrophoretic display
FAR	federal aviation regulations
FET	field effect transistor
GPD	gas plasma device
Group I, II and III terminals	classification of airports in regard to landing restrictions
HOL	high-order language
IC	integrated circuit
IFR	instrument flight rules
I^2L	integrated injection logic
ILS	instrument landing system
INS	inertial navigation system
I/O	input/output
JTL	Josephson tunneling logic
KOPS	thousand operations per second
LCD	liquid crystal display
LED	light-emitting diode

LSIC	large-scale integrated circuit
MECL	Motorola emitter coupled logic
MLS	microwave landing system
MNOS	metal nitride oxide semiconductor
MOS	metal oxide semiconductor
MOSFET	metal oxide semiconductor field effect transistor
MSIC	medium-scale integrate circuit
MTL	merged transistor logic (same as I^2L)
NDRO	nondestructive read out
NMOS	N-channel metal oxide semiconductor
PLZT	lead, lanthanum, zirconium, and titanium ceramics
PMOS	P-channel metal oxide semiconductor
RF	radio frequency
R-NAV	area navigation
RTL	resistor transistor logic
SBTTL	Schottky barrier transistor-transistor logic
SFL	substrate fed logic (same as I^2L)
SOS	silicon on substrate
TFT	thin-film transistor
T^2L	transistor-transistor logic
T^3L	transistor-transistor-transistor logic TTL with PNP emitter follow input
T^2LS	transistor-transistor logic (Schottky)
TRL	transistor-resistor logic
TTL	transistor-transistor logic
V/F	voltage to frequency
VFR	visual flight rules
VMOS	V-groove metal oxide semiconductor
VOR	VHF omni-directional radio range

SECTION 4
ASSESSMENT OF FUTURE GENERAL-AVIATION REQUIREMENTS

GENERAL-AVIATION REQUIREMENTS

Airframe/Avionics Classification

Currently available general-aviation aircraft form a continuous progression from single-place pure sport planes to multi-engine fan jets with transoceanic capability. Any grouping of all these types of aircraft is bound to be arbitrary, but grouping into subsets is necessary because the airplanes and the avionic equipment obviously differ greatly. The approach used in this study is to start with those airplanes that have the minimum capability to contribute to the national transport capability and work up to the business jet class. There are, of course, a number of actual commercial transport airplanes used as general-aviation aircraft by large companies or wealthy individuals, but these are ususally flown, equipped, and serviced in the same manner as commercial transports and are not considered in this study. Between the minimum capability aircraft (Class C) and the multi-engine turbine-powered aircraft (Class A aircraft, which in many cases carry avionics suites very similar to those of commercial aircraft), the twin-engine piston aircraft (Class B) have been lumped as comprising a group with requirements for more than minimum avionics capability, but still not justifying the cost of more sophisticated avionics. The criterion for the lowest class (i.e., 200 horsepower or greater) may seem to be restrictive in that many lower powered aircraft having cross-country transportation capability are excluded. The reasoning here is that the function of the airplane (cross-country transportation) will probably require that these lower powered airplanes carry avionics approximately equal to the higher powered aircraft, and thus a separate class of aircraft/avionics is not

needed. Types of aircraft specifically excluded from any of these groups, even though some of them have more than 200 horsepower, are trainers, sport and acrobatic planes, agricultural aircraft, and helicopters.

Table 1 presents a summary chart showing airplane characteristics, type of use, air traffic control (ATC) environment, and weather capability for each of the three classes of aircraft, as well as an indication of where computer technology may be effectively applied. Table 2 gives more detailed data on each of the three classes of aircraft in the form of maximum, minimum, and average values of airframe cost, cruise speed, cruise range, and cruise endurance. Table 3 lists the avionics associated with each type of aircraft, and Table 4 lists the specific aircraft which make up each class.

As noted, this classification is arbitrary and obviously different groupings could be made. However, the current study is concerned with the application of computer technology to future general-aviation avionics rather than with the details of the aircraft and avionics themselves. Also it is felt that the future application of computer technology to the various levels of avionics will differ in degree rather than principle. Therefore, three classes of aircraft and accompanying avionics were chosen as sufficient to characterize the entire range. In the process of classifying the aircraft and avionics, considerable use was made of the data presented in Reference 2, which gives detailed statistics on general-aviation aircraft and avionics.

In an effort to substantiate the recommendations for future research (discussed in Section 8), a cursory analysis of the cost of typical general-aviation subsystems was made. The object was to assess the influence of the cost of various types of hardware (e.g., RF receivers, computers and memory, inertial sensors) on the cost of the complete subsystem. The relative costs were estimated from manufacturers' parts lists where available or from design data and typical parts costs where the actual prices were not available. The figures presented in Table 5 are typical rather than exact

TABLE 1. – CHARACTERISTICS OF TYPICAL CURRENT GENERAL-AVIATION AIRCRAFT

Aircraft class	Averages					No. of pilots	Mission
			Cruise				
	Airframe cost, $	Avionics cost, $	Range, n. mi.	Speed, K	Endurance, hr		
A) Multi-engine turbine	1 270 000	159 720	1 592	345	4.7	2	• Business, company crew • Third-level airline • Charter • Air freight
B) Twin-engine piston	145 000	17 235	1 127	208	5.4	1-2	• Pleasure • Business Small-business, owner operated Corporate, contract pilot Air taxi, passenger and freight
C) Single-engine piston ≥ 200 hp	42 790	7 287	832	154	5.4	1	• Pleasure • Business Small business, owner operated Air taxi, passenger and freight Charter, passenger and freight

TABLE 1. – CHARACTERISTICS OF TYPICAL CURRENT GENERAL-AVIATION AIRCRAFT (Concluded)

Aircraft class	ATC environment	Weather	Computer applications
A) Multi-engine turbine	• Controlled air space Enroute Terminal groups I, II, III Tower • Uncontrolled air space Seldom used	• VFR • IFR Known icing, thunderstorms; category I and II approaches	Flight stabilization, area navigation, flight management, fuel management computations, flight progress computations.
B) Twin-engine piston	• Controlled air space Enroute Terminal groups I, II, III Tower • Uncontrolled air speed Occasional user	• VFR • IFR Light icing, scattered thunderstorms; cloud ceilings and visibilities down to category I minimums	Flight stabilization, area navigation, flight management, fuel management computations, flight progress computations.
C) Single-engine piston ≥ 200 hp	• Controlled air space Enroute Terminal groups I, II, III Tower • Uncontrolled	• VFR • IFR No known icing, or thunderstorms; ceilings and visibility down to category I minimums	Flight stabilization, area navigation, flight progress computations

TABLE 2. - AIRCRAFT CLASS STATISTICS

Class A - Multi-engine turbine units surveyed: 32				
	Airframe cost, $	Speed, K	Cruise Range, n.mi.	Endurance, hr
Minimum	385 000	170	828	2.4
Maximum	4 000 000	495	3 642	8.3
Average	1 270 000	345	1 592	4.7

Class B - Twin engine - piston units surveyed: 24				
	Airframe cost, $	Speed, K	Cruise Range, n.mi.	Endurance, hr
Minimum	63 000	170	760	4.0
Maximum	241 000	260	1 440	6.5
Average	145 000	208	1 127	5.4

Class C - Single engine ≥ 200 hp units surveyed: 32				
	Airframe cost, $	Speed, K	Range, n.mi.	Endurance, hr
Minimum	25 000	130	466	3.4
Maximum	85 000	220	1 485	7.8
Average	42 790	154	832	5.4

TABLE 3. – AVIONICS

Class C Single engine, > 200 hp, avionics cost as percent of airplane cost = 17%				
Quantity	Unit	Use in percent	Cost in dollars	Average cost in dollars
1	#1 Nav/communication	100	2 000	2 000
1	#2 Nav/communication	70	1 000	700
1	Automatic direction finder	80	1 000	800
1	Transponder	100	600	600
1	Distance measuring equipment (DME)	50	1 000	500
1	Audio/marker	100	270	270
1	Emergency location transmitter	100	125	125
1	Autopilot	50	2 000	1 000
1	Radar altimeter	10	2 000	200
1	Radio telephone	2	1 600	32
1	R Nav	3	2 000	60
			Total	7 287

Class B Twin engine – piston, avionics cost as percent of airplane cost = 12%				
Quantity	Unit	Use in percent	Cost in dollars	Average cost in dollars
1	#1 Nav/communication	100	2 600	2 600
1	#2 Nav/communication	90	2 000	1 800
1	Glideslope	90	1 500	1 350
1	Automatic direction finder	100	1 400	1 400
1	Transponder	100	600	600
1	Distance measuring equipment (DME)	60	2 500	1 500
1	Audio marker	100	270	270
1	Emergency location transmitter	100	150	150
1	Autopilot	75	5 000	3 750
1	Radar altimeter	25	2 000	500
1	Radar weather	25	7 500	1 875
1	Radio telephone	15	1 600	240
1	R-Nav	10	2 000	200
1	Integrated display	25	2 000	500
1 lot	Adapters, interface units, converters optional display	25	2 000	500
			Total	17 235

TABLE 3. - AVIONICS (Concluded)

Class A Multi-engine turbine, avionics cost as percent of airplane cost = 12.5%				
Quantity	Unit	Use in percent	Cost in dollars	Average cost in dollars
2	Communication	100	3 000	6 000
1	Navigator	100	1 600	1 600
2	Glideslope marker beacon	100	1 500	3 000
2	Automatic direction finder	100	3 370	6 740
2	Remote magnetic indicator	100	2 300	4 600
2	Transponder	100	2 190	4 380
2	Distance measuring equipment (DME)	100	3 900	7 800
1	Audio	100	1 800	1 800
1	Autopilot	100	10 000	10 000
2	Flight directors	100	24 300	48 600
1	R-Nav	100	12 000	12 000
2	Encoding altimeter	100	5 360	10 720
1	Weather radar	100	18 000	18 000
2	Radar altimeter	100	4 700	9 400
1	Radio telephone	100	2 800	2 800
1	Emergency location transmitter	100	280	280
1 lot	Control heads Adaptors Interface units Converters Optional displays Intercoms, etc.	100	12 000	12 000
			Total	159 720

TABLE 4. - AIRCRAFT USED IN SURVEY

Class C Single engine > 200 hp		
Manufacturer	Model	Cost in dollars
Cessna	Skywagon 180	25 700
Cessna	Skywagon 182	26 700
Cessna	Skylane	27 950
Cessna	Skywagon 185	30 175
Cessna	Stationair	36 375
Cessna	Skywagon 207	37 250
Cessna	Turbo Stationair	41 625
Cessna	Turbo Skywagon 207	42 725
Cessna	Cardinal RG	30 950
Cessna	Centurian	47 950
Cessna	Turbo Centurion	53 150
Piper	Cherokee Pathfinder 235	27 070
Piper	Cherokee Six 260	32 170
Piper	Cherokee Six 300	35 740
Piper	Arrow II	27 690
Beech	Sierra 200 B24R	30 250
Beech	Bonanza F33A	55 400
Beech	Bonanza V35B	55 400
Beech	Bonanza A36	59 000
Mooney	Chaparral M20E	28 500
Mooney	Executive M20F	30 500
Bellanca	Super Viking 17-30A	38 990
Bellanca	Super Viking 17-31A	41 110
Bellanca	Turbo Viking 300A	48 545
Bellanca	Skyrocket II 19-25	80 000
Rockwell	Commander 112A	33 500
Aerospatiale	Rallye 220 GT	33 000
Helio	Super Courier H-295	81 740
Helio	Tri-gear Courier HT-295	85 869
Navion	Range Master H	46 250
Trident	Tri Gull 320	59 500
Lake	Buccaneer	37 950

TABLE 4. - AIRCRAFT USED IN SURVEY (Continued)

Class B Twin engine - piston		
Manufacturer	Model	Cost in dollars
Cessna	Skymaster	63 300
Cessna	310	89 950
Cessna	Pressurized Skymaster	94 100
Cessna	Turbo 310	107 500
Cessna	402B	138 500
Cessna	340	143 950
Cessna	414	174 950
Cessna	421B	229 950
Piper	Seneca II	63 995
Piper	Aztec E	88 200
Piper	Turbo Aztec E	99 095
Piper	Turbo Navajo B	139 100
Piper	Navajo Cheftain	167 995
Piper	Pressurized Navajo	241 400
Beech	Baron B55	89 000
Beech	Baron E55	109 500
Beech	Baron 58	128 000
Beech	Duke B60	219 450
Beech	Queen Air B80	240 925
Rockwell	Shrike Commander	128 150
Rockwell	Commander 685	229 440
Ted Smith	Aerostar 600	138 800
Ted Smith	Aerostar 601	156 200
Ted Smith	Aerostar 601P	198 900

TABLE 4. - AIRCRAFT USED IN SURVEY (Concluded)

Class A Turbine - multi-engine		
Manufacturer	Model	Cost in dollars
Cessna	Citation	795 000
Piper	Navajo Cheyenne	459 000
Beech	King Air C90	476 500
Beech	King Air E90	593 000
Beech	AirLiner B99	691 500
Beech	King Air A100	733 750
Beech	Super King Air	806 000
Beech	BH-125-600	1 886 000
Rockwell	Turbo Commander 690A	473 000
deHavilland	Twin Otter	595 000
Mitsubishi	MU-2M	669 625
Mitsubishi	MU-2L	767 625
Swearinger	Merlin IIIA	725 000
Swearinger	Merlin IVA	815 000
Swearinger	Metro II	815 000
Shorts	SD3-30	1 400 000
Gates	Learjet 24D	941 500
Gates	Learjet 25B	1 071 500
Gates	Learjet 25C	1 138 000
Gates	Learjet 35	1 395 000
Gates	Learjet 36	1 445 000
Aerospatiale	SN601 Corvette	1 300 000
IAI	1124 Westwind	1 550 000
Dassault	Falcon 10	1 575 000
Dassault	Falcon 20E Cargo	2 200 000
Dassault	Falcon 20E	2 225 000
Dassault	Falcon 20F	2 475 000
Rockwell	Sabre 60	1 690 000
Rockwell	Sabre 75A	2 195 000
Lockheed	Jet Star II	3 750 000
Grumann	Gulfstream II	3 900 000
Shorts Harland	Sky Van 3	750 000

TABLE 5. – RELATIVE COSTS OF SUBSYSTEM ELEMENTS (GIVEN IN PERCENT OF TOTAL SYSTEM COST)[a]

Subsystem	Major assemblies					
	Receivers	Computer and memory	Display	Inertial sensors	Servos	Controls
R-NAV	40	25	25			10
Autopilot/stability augmentation system, 3 axis		35		35	20	10
Flight director		25	30	40		5
I-NAV		25	5	65		5
ILS and communication	75		10			15

[a] Each system is assumed independent, as flight directors and autopilots must be independent for Category II operation.

for any specific model of subsystem; the totals would not be changed much by ±5 percent variations in any of the subassembly costs. The results indicate that while computer and memory costs are significant, they are by no means dominant, comprising no more than 35 percent of any of the system costs. RF receivers, inertial sensors, and displays all have approximately an equal overall cost effect on a complete avionics system. Therefore, while the computation function per se is an important part of avionics subsystem costs, there are also other elements that must receive equal or greater emphasis in the effort to reduce total avionics costs.

Air Traffic Control Environments

The ATC environments given in Table 1 for each of the classes of aircraft differ very little in absolute requirements, but differ considerably in the emphasis on the various requirements. Class C aircraft are expected to be used a significant amount of time in uncontrolled air space and infrequently in controlled air space. They are not expected to use Group I terminals except in unusual circumstances. The ATC environments for Classes A and B are essentially the same with only a minor difference in the expected use of uncontrolled air space. Essentially all three categories must be able to operate in all types of ATC environment. Probably the biggest difference in the functional requirements of the avionics comes from the airframe characteristics themselves, i.e., the higher speed, altitude, and fuel consumption of the higher-level aircraft. The obvious trend for the future will be for more positive controlled air space, probably requiring user aircraft to be equipped for ATCRBS and ultimately DABS. The end result will be that most useful cross-country missions will encounter positive controlled air space, and all general aviation will have to be equipped with the proper ATC equipment.

Weather Conditions

Weather conditions per se will have little effect on avionics beyond the obvious fact that poor visibility due to any sort of weather creates the need

for instrument flight rules (IFR) and their associated equipment. Two other weather characteristics which affect aviation are turbulence and icing; however, the capability of operating in these conditions is far more dependent on the airplane than on the avionics. No amount of avionics will prevent ice from forming on an airplane that does not have adequate deicing features. The only part that avionics could play in combatting icing would be in the reporting of icing conditions by ground stations or other aircraft. In regard to turbulence, there can be considerable improvement in handling qualities in mild turbulence by use of stability augmentation. Flight in turbulence that approaches the structural capability of general-aviation aircraft (FAR, part 23), would be intolerable to the aircraft occupants; again, the role of avionics would be in providing information to enable the pilot to avoid turbulent areas of severe turbulence.

VARIATIONS OF APPLICABLE/REQUIRED COMPUTER TECHNOLOGY FOR EACH CLASS

General Considerations

There are reasons to think that the level of technology would be the same for all classes of aircraft. For example, all classes must operate in the same controlled air space, must at sometime encounter similar weather, and must use the same ground navigation facilities. However, the differences in the areas of pilot qualification/work load and the criticality of the avionics functions and the mission itself require different capabilities. Class A (turbojet) aircraft and the missions for which they are intended definitely require more highly trained pilots and impose a higher work load on the pilot than do Class C aircraft. Further, because of low altitude fuel consumption and airstrip requirements, it is far more critical for a business jet to fly a direct course to the intended airport than for a single-engine propeller airplane. Therefore, avionics for high-performance aircraft require greater mission reliability, automaticity, precision (particularly in flight-control

equipment), and unfortunately complexity. These differing requirements affect the level of avionics technology required for the various aircraft classes.

Variations in Required Technology Levels

The relationship of technology to aircraft class can best be illustrated by use of the following breakdown:

- Logic, memory, and I/O devices
- Input and display devices
- RF electronics
- Inertial sensors and servos
- Computer architecture and software
- Overall avionic system configuration

Logic and memory. — Logic and memory types will differ little from one class to the other. Several studies (e.g. 3 and 4) and recent industry experience have shown that 16-bit-word digital processors are necessary and sufficient for most navigation and flight control functions. Twelve-bit processors could be used for flight control for low-performance aircraft (Class C), but this is a comparatively minor application and thus far the tendency is to move from 8 bits to 16 bits in terms of the digital electronics if more precision is needed. There may well be several types of actual semiconductor techniques in use because, as shown in Section 5, no one technique has clear and overriding advantages over the others. There are significant differences in inherent gate speed and power consumption, but current devices demonstrate that at least metal oxide semiconductor (MOS), transistor-transistor logic (TTL), and emitter-coupled logic (ECL) devices are suitable and available for avionics use.

On the other hand, there will be significant differences in the amount of throughput or memory that is needed. For example, there is on the market today a low-cost (10 way point) R-Nav that uses a microprocessor with about

5 KOPS (thousands of operation per second) throughput and 1024 bits of memory. In contrast, a sophisticated airline-quality R-Nav may have a throughput of 50 KOPS and 192,000 bits of program, scratchpad, and data memory. Also the amount of logic and memory required for flight control will vary with the type of aircraft. A light plane could probably get by with 60-KOPS throughput and a 64,000-bit memory for a complete three-axis autopilot, while a redundant autopilot for Category II landing in a business jet would need 96,000 bits of memory and 150 KOPS in both channels. Probably the greatest computing capacity demand envisioned would be in the use of an integrated navigation/autopilot system using strapdown inertial sensors; 300-KOPS throughput and 256,000 bits of memory would be required. As the data in Section 5 show, both the basic computing logic and memory devices to meet these requirements in suitably small and potentially inexpensive units are now available. The final choice of hardware will probably be dictated by dollars/function rather than the level of technology itself.

Input and display devices. — Here the simpler airplane, equipped with less avionics, will need fewer and less complex input and display devices than the high-level turbojet. For Class C, the first computers having pilot inputs and displays may well be in an R-Nav set or perhaps an engine control/fuel/status monitoring system. In either case, a simple keyboard input and an alpha-numeric plus warning light display would suffice. These requirements could be met satisfactorily with current keyboards and LED, LCD or incandescent displays. An added refinement might be a map display for the R-Nav; this would require a CRT display or one of the newer matrix panels discussed later.

Class A turbojets have an instrument panel similar to a commercial airliner, and the need for improved displays is well recognized. In the 1980's these aircraft will have several computers requiring complex input devices. There are already space, power, and readability problems with inputs and displays on sophisticated aircraft, and multi-use inputs coupled with multi-format displays will be welcome solutions, particularly since the high-level aircraft will not be so sensitive to the cost likely to accompany the first use of advanced input and display devices.

RF electronics. — RF electronics, although not nominally a computing element, is included here because the data in Table 5 show it to be an important cost factor and because the performance requirements for airborne receivers, and to some degree transmitters, remain fairly constant regardless of the class of aircraft involved. Briefly, this is because bandwidths and signal formats are dictated largely by ground equipment and regulation rather than by aircraft type or mission. There will be differences in RF equipment, but they will be in the areas of reliability, maintainability, and service life. Reference 5 describes this situation and gives estimated costs for microwave landing system (MLS) receivers for both low-cost general aviation and air carrier use. The ratio of prices is less than 3:1 ($1,750 versus $4,500); this is far lower than would be the case for an autopilot or the primary flight displays where the ratio will be nearer 10:1. These differing cost ratios indicate that there is a real need to develop true low-cost RF electronics and that the performance requirements for flight control hardware do vary considerably with the type of aircraft.

Inertial sensors and control servos. — Neither of these items are, strictly speaking, part of avionic computers, but they are discussed here because better and cheaper stability augmentation systems and autopilots would greatly contribute to reducing pilot work load, particularly in IFR conditions. Reference 6 gives an excellent example of the benefits of stability augmentation.

The technical level of both inertial sensors and servos will vary considerably with the class of aircraft. Inertial sensors for general-aviation flight control and display purposes generally consist of angular displacement and rate sensors. In a Class C airplane, a rate threshold of 0.25 degree/second and an attitude threshold of 0.5 degree would be tolerable. These figures correspond to those found in panel-mounted (for display) gyros, and in several current light-plane autopilots the gyro signals are taken from electrical pickoffs on panel instruments. On the other end of the scale, a Class A turbojet autopilot will need thresholds below 0.1 degree/second for rate and 0.25 degree for attitude. There will also be a need for the turbojet sensors

to have better dynamic performance (i.e., damping and natural frequency). Quantitatively the performance differs by a ratio of about 3.5:1, but the cost will vary much more, probably at least 5:1. Some top-level Class A turbojets will carry INS using very precise gyros and accelerometers that are priced accordingly (approximately $50,000 per set). In all cases, current inertial sensors, particularly gyros, have relatively high maintenance costs. Current performance levels are satisfactory, but there is an obvious need to apply advanced technology to reduce purchase and maintenance costs.

Somewhat the same comments apply to control surface servos. Because of the low control sensitivity and low hinge moments typical of Class C airplanes, servos with low dynamic response and torque capability can be used. Even so, there is a premium on low weight and power consumption that makes most industrial actuators unsuitable; and thus there is a need for semi-custom designs. A few light-plane autopilots have used penumatic servos, but with the development of solid-state power amplifiers the trend has been more to electrics.

Servos for Class A aircraft are usually similar but more advanced in all aspects of performance (e.g., response, resolution and torque). Two obstacles to the use of the generally higher-performance hydraulic servos are 1) the lack of powered surfaces, which makes the use of series servos difficult and 2) the expense and maintenance associated with aircraft hydraulics. (A series servo is one that drives the control surface without causing any motion of the pilot's controls.) It should be noted that while general-aviation aircraft servos are not the most advanced available, they are performing satisfactorily in their current roles. However, the exploitation of advanced control techniques such as ride quality improvement and relaxed airframe stability will require improved performance.

Computer architecture and software. — The level of technology in these areas will vary with the class of aircraft for many years because the more simple avionics suites for Class C aircraft will not have a critical need for the features that more advanced techniques offer. The advanced techniques

for architecture and software (as described in Section 5) consist of multiple processors cooperatively handling tasks too large for any one processor, and higher-order languages for creating the software. Digital electronics are now being used in low-cost avionics (e.g., in R-Nav and distance measuring equipment), but there is little need for complex intercommunication and no lack of computing capacity in available single microprocessor-based computers. As digital computers are applied to more functions such as flight and engine control, there may be a need for the crossfeeding of some data, but this form of communication is relatively simple and does not involve the advanced techniques discussed in Section 5. One possible new function in which advanced computer architecture would be involved may be a central preflight and maintenance test system. As the separate avionics become digitized, a central checkout system becomes more feasible in principle, but is still subject to the practical considerations of standardization of system configurations and box interfaces as well as the level of field maintenance technology.

In regard to software, the desirability of higher-order languages (HOL's) for production systems has been a subject of controversy for several years. It is true that the use of a HOL will reduce programming time, particularly when the programmer is not familiar with the processor being used and when the HOL compiler is efficiently and completely developed. What is sometimes not considered is that in avionics computers the I/O and I/O control often affect the timing of operations to the point that a HOL compiler for a particular processor must be tailored to fit the specific application of that processor in a computer. To the extent that this is true it would create problems (or at least design restraints) for an avionics manufacturer who bought a processor and a HOL compiler to use in constructing his own computer. In general, it seems doubtful that for simple digital applications the HOL will be the better approach; however, as the digital capability (throughput and memory) is increased, the HOL approach will become more attractive and may be desirable, particularly for the manufacturer who buys a processor or computer to put into his hardware.

Overall Avionic System Configuration. – The application of the technologies discussed to general-aviation avionics will undoubtedly improve the performance and cost characteristics of the various subsystems, but the total potential benefit will not be realized unless the combined airframe/avionics/pilot system is considered as a whole. If it is assumed that a Class B or C aircraft that is being flown by a pilot with less average training and experience should perform its mission with the same safety and dependability that is expected of a Class A aircraft, then it would appear that the avionic system effectiveness must really be better in the Class C aircraft than in the Class A aircraft. Admittedly, the airplane may be more forgiving and the landing category requirements lower, but the allowable cost and complexity should also be lower. These considerations indicate that careful attention should be given to defining what the total system functional requirements should be and what system configurations will most efficiently meet those requirements.

To obtain the desired improvements in safety, dependability, and cost within the projected aircraft and ATC environment will probably require significant innovation and sophistication (but not complexity) in the overall system design. Thus, the level of overall avionic system technology will need to be higher for all classes of general-aviation aircraft.

SECTION 5
ASSESSMENT OF CURRENT TECHNOLOGY

This section presents a review of the current status and trends of many types of:

Semiconductor Technology
Component and Module Technology
Computer System Architecture Technology
Software Technology
Operator/Computer Interface Technology

In general, all devices or techniques that are beyond the basic research stage and appear to be practical are reviewed. For a variety of reasons some of these technologies may not be applicable to general-aviation, but are included for the sake of completeness. Wherever it is clear whether or not the technology is applicable to general aviation, such a distinction is noted.

SEMICONDUCTOR TECHNOLOGY

Circuit Technology

Advances in the semiconductor art over the last five years have been impressive. The basis for much of the recent progress in integrated circuit (IC) technology has been advances in photolithographic, device isolation, and ion-implantation processing schemes. Semiconductor chips with 14,000 transistors are available as off-the-shelf items, and microprocessors, hand-held calculators, and complex memory chips are readily available. Research into newer methods, such as electron-beam and ion-beam implantation pattern generation, promise more improvement. To prevent mask damage due to contact printing, efforts are also underway in projection printing and

"near contact" printing. Feeding these developments is an array of technoligies remarkable for their diversity and ability to enhance circuit performance.

As shown in Figure 1, most of these technologies still employ silicon as the base material and are primarily of two basic types: bipolar and MOS. The bipolar technologies can be broken down into at least 16 distinct circuit forms, based on three approaches to circuit construction:

- Epitaxial collector techniques
- Triple diffusion techniques
- Oxide-isolation techniques

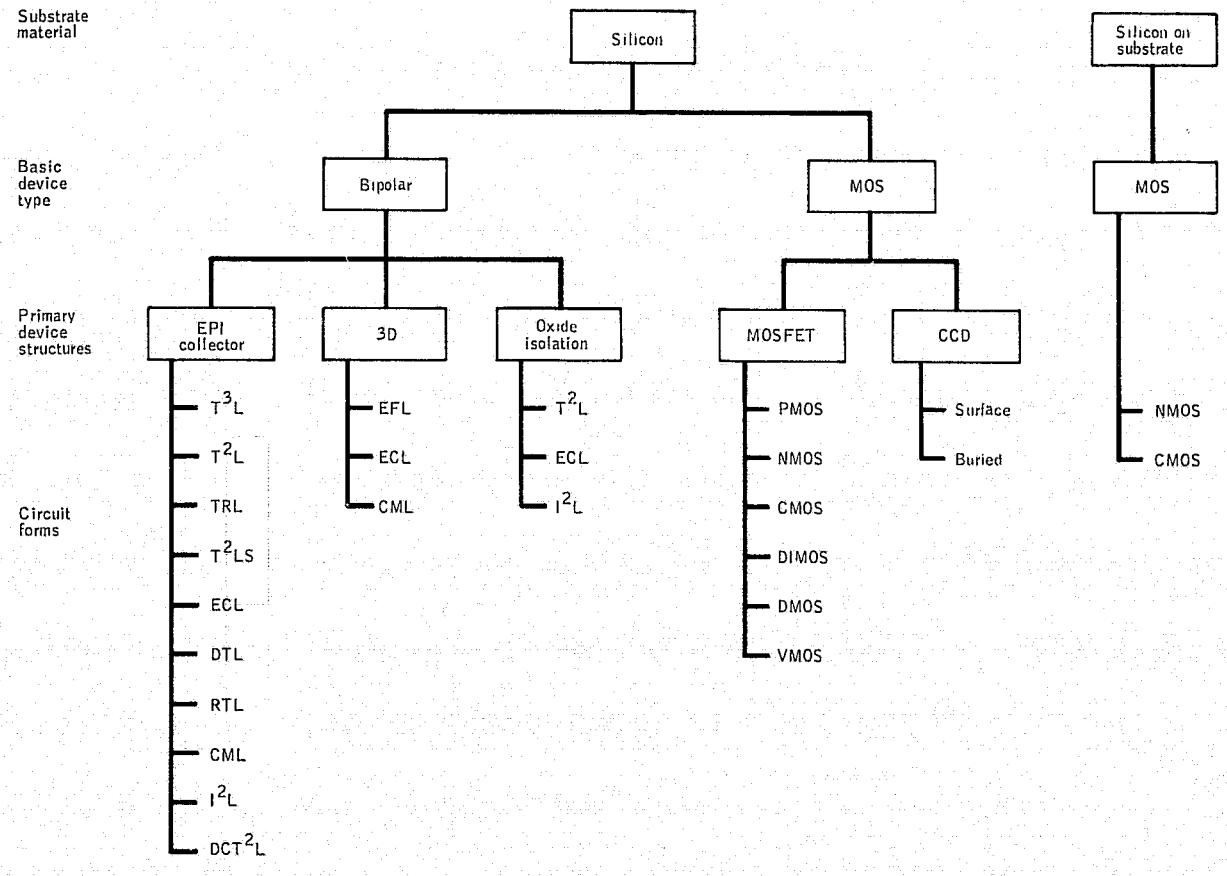


Figure 1. – Semiconductor Technologies

The circuit design characteristics of each of these semiconductor technologies are summarized in Appendix A.

The most promising new developments for digital logic are presently being made in high-density, low-power bipolar logic, where a number of circuit forms are approaching the density levels of present-day MOS.

MOS is being fabricated on both bulk silicon and on sapphire substrates. While MOS has historically been classified as a very dense and low power but slow circuit form, new developments are making MOS faster, approaching today's most popular bipolar circuit forms. The recent emergence of charge-coupled devices will allow significant advances to be made in the areas of signal processing, imagery, and digital memories.

To achieve good yields at high levels of integration in a given technology, three factors must be minimized:

- Device area
- Device power
- Process complexity

Figure 2 shows the gate delay and the required chip area for various semiconductor technologies. Note that some of the high-speed bipolar technologies can be fabricated with very small area consumption. Figure 3 shows the speed versus power relationship for a typical two-input gate for a number of IC technologies. Most technologies widely available today are outside the 10 picojoule line. Process complexity is to some degree a function of the number of mask steps and diffusions required. These are given in Table 6, along with the total number of process steps where known.

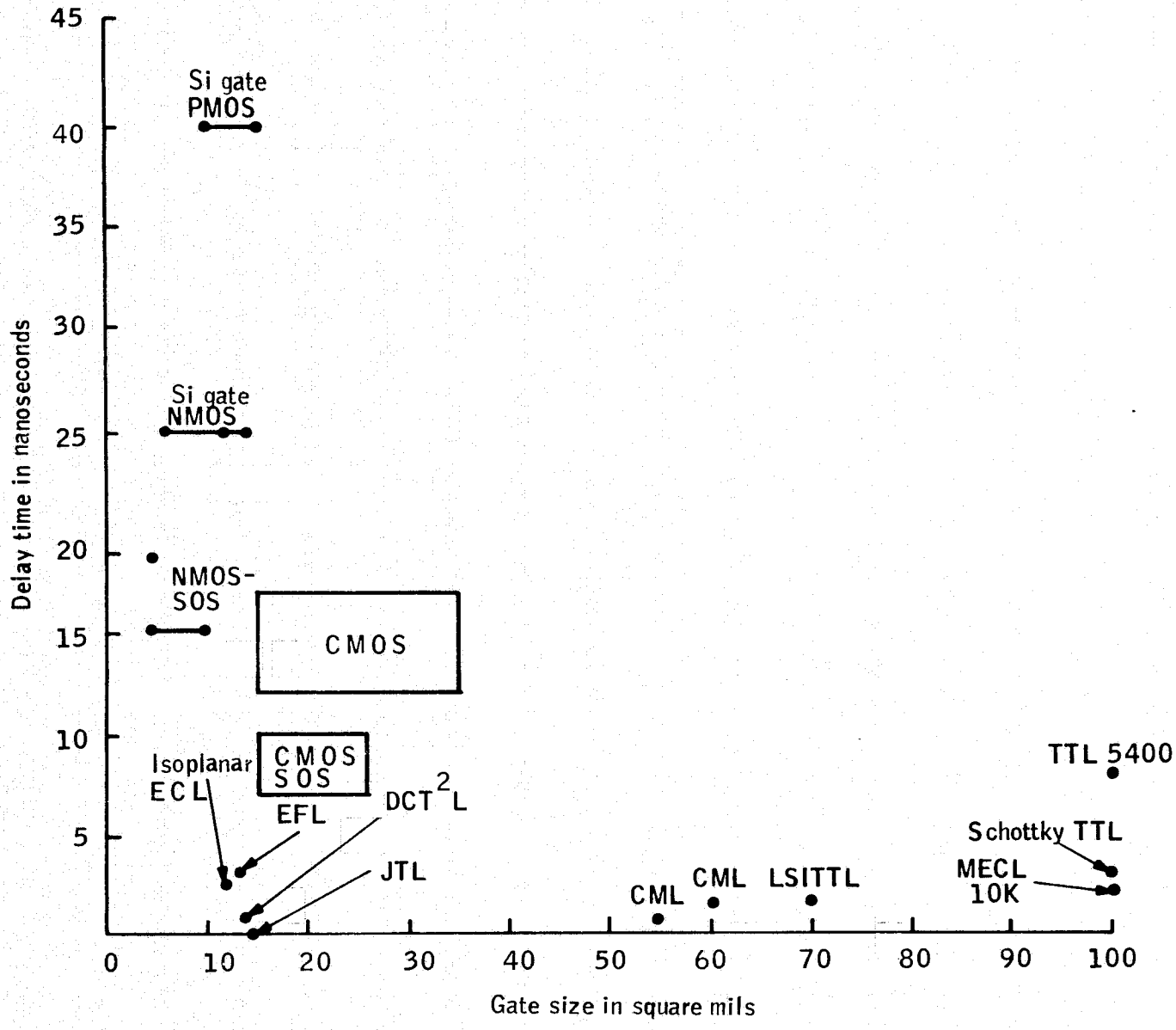


Figure 2. – Gate Delay versus Gate Size (1975)

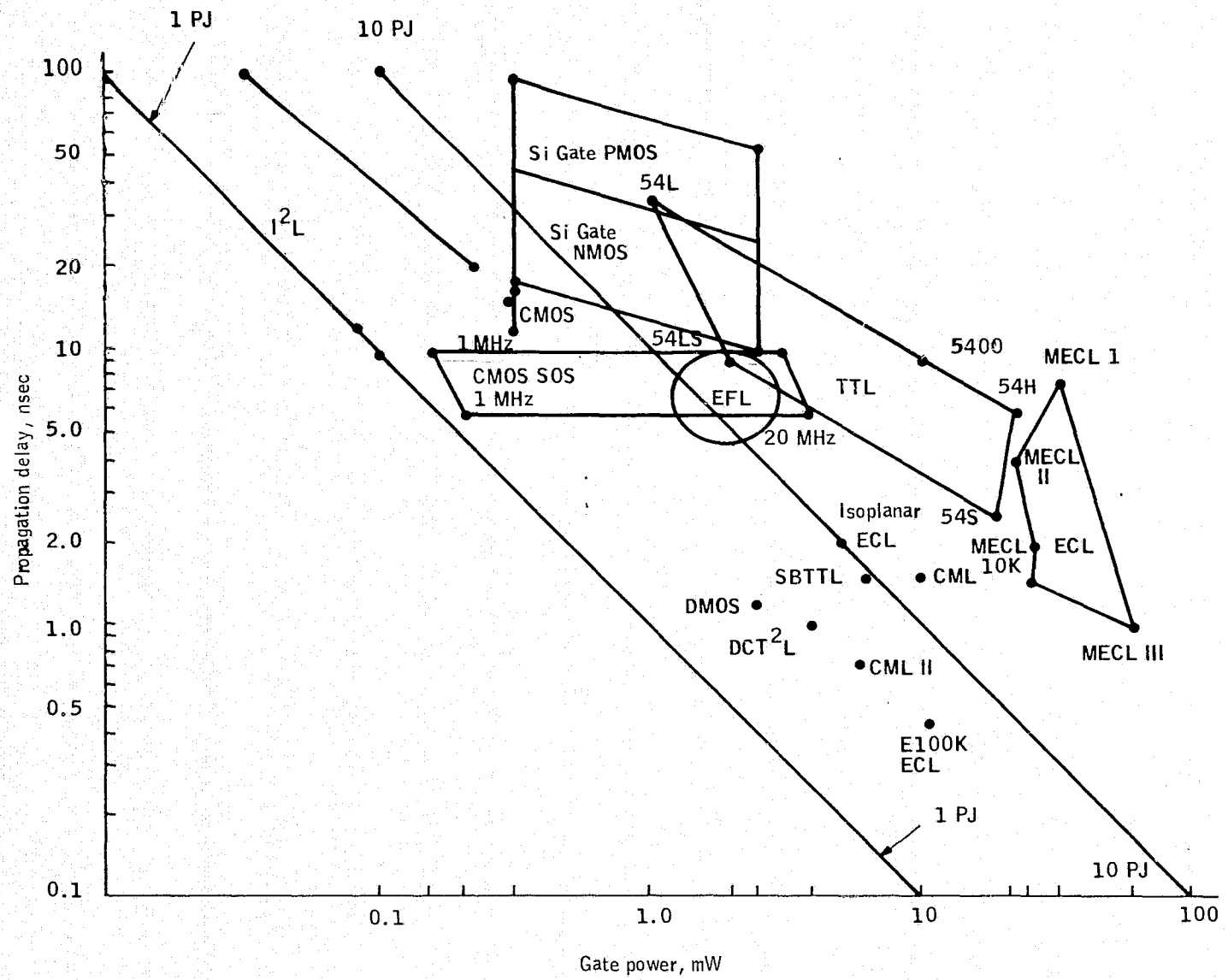


Figure 3. – Gate Delay versus Gate Power (1975)

TABLE 6.–MEASURES OF PROCESS COMPLEXITY

Circuit form	Mask steps	Diffusions	Process steps
TTL (low power)	7	4	60-65
TTL std	7	4	60-97
TTL Schottky			60-74
TTL LSI	7	4	
PMOS metal gate	4	1	29
NMOS (Si gate)	5-7	3	23-34
NMOS (SOS)	5-7	3	30-40
CMOS	7	3	37
CMOS (SOS)	7	3	30-40
I^2L	4	2	
EFL	5	3	
ECL (isoplanar)	6	3	
VMOS	3-4	1	
CML	7	5	
DCT^2L	7	5	
DMOS		2	

From this table it is apparent that the bipolar processes commonly in use today are significantly more complex than those used to fabricate MOS technologies. More recent bipolar technologies, such as I^2L, EFL, and isoplanar ECL, require fewer mask steps and diffusions, and thus promise higher yields on LSIC chips.

Guidelines for the Effective Application of LSIC

Until the early 1970's, the only implementation tools available for the digital portion of a system consisted of IC's, which contained an average of about four logic functions per package. As a result, the cost, size, weight, power and reliability of a system were almost a direct function of the number

of logic gates in the system. Thus, a heavy emphasis was placed on minimizing the number of gates. Because the logic complexity per package was so limited, the design of standard multi-usage devices was a simple task.

With LSIC, hundreds of logic functions can be contained in a single package. Designing functional elements of this complexity that have universal appeal is difficult. Nevertheless, during 1974 and 1975 many complex, but standard LSIC devices that perform significant digital computing functions have appeared, and it has become relatively easy to assemble efficient digital processors using standard LSIC's. The proper use of the new implementation tools provided by LSIC requires that new design approaches be used and that the system designer be fully aware of the advantages and limitations inherent in LSIC. To define a system ideally suited to LSIC implementation, a designer must consider the following guidelines:

- Replace analog and mechanical devices with digital circuitry
- Increase the use of semiconductor memories
- Use functional and computational parallelism
- Replace software by hardware
- Increase system fault tolerance
- Reduce system interconnections

Once the system has been properly defined and an implementation approach selected, the system must be properly partitioned into unique LSI circuits. The goals for proper system partitioning are to:

- Maximize the use of standard LSIC chips
- Minimize the number of unique part types
- Minimize the total number of parts required
- Maximize the gate-to-pin ratio of each part
- Ensure economically testable chips

Appendix B discusses each of these guidelines in detail, and presents clarifying examples.

A major benefit of the use of digital rather than analog electronics is a much lower failure rate, thus an increase in safety and a decrease in maintenance costs. Data on currently operational digital avionics indicates an increase in mean time between failures (MTBF) of from 4 to 10 times that for similar analog units. This current digital equipment is built using MSIC. Because failure rate is roughly proportional to piece-part count, future equipment using LSIC (and thus fewer parts) will show even more improvement. Appendix C gives further data on LSIC reliability.

COMPONENT AND MODULE TECHNOLOGY

Microcomputer Technology

A microcomputer can be discussed on three distinguishable levels of sophistication: the component level, card level, and system level. In Figure 4 the microcomputer is divided into hardware and software and the various elements that constitute the three levels of sophistication. These elements and terms used in describing microcomputers are defined in the next pages.

Microprocessor. — The microprocessor is part of the lowest or component level and corresponds to a rudimentary central processing unit. In a fixed-instruction system, it contains the control logic and arithmetic logic sections, and in a microprogrammed system, it contains an additional control memory section. The processor portion of a computer can be purchased in one of the following three forms, all which are sometimes referred to as microprocessors:

- Functionally Partitioned Chip Sets -- Separate LSIC chips or parts are provided for CPU, control, register banks, stacks, etc. An example of this type of unit is the Fairchild 9400 series of circuits.

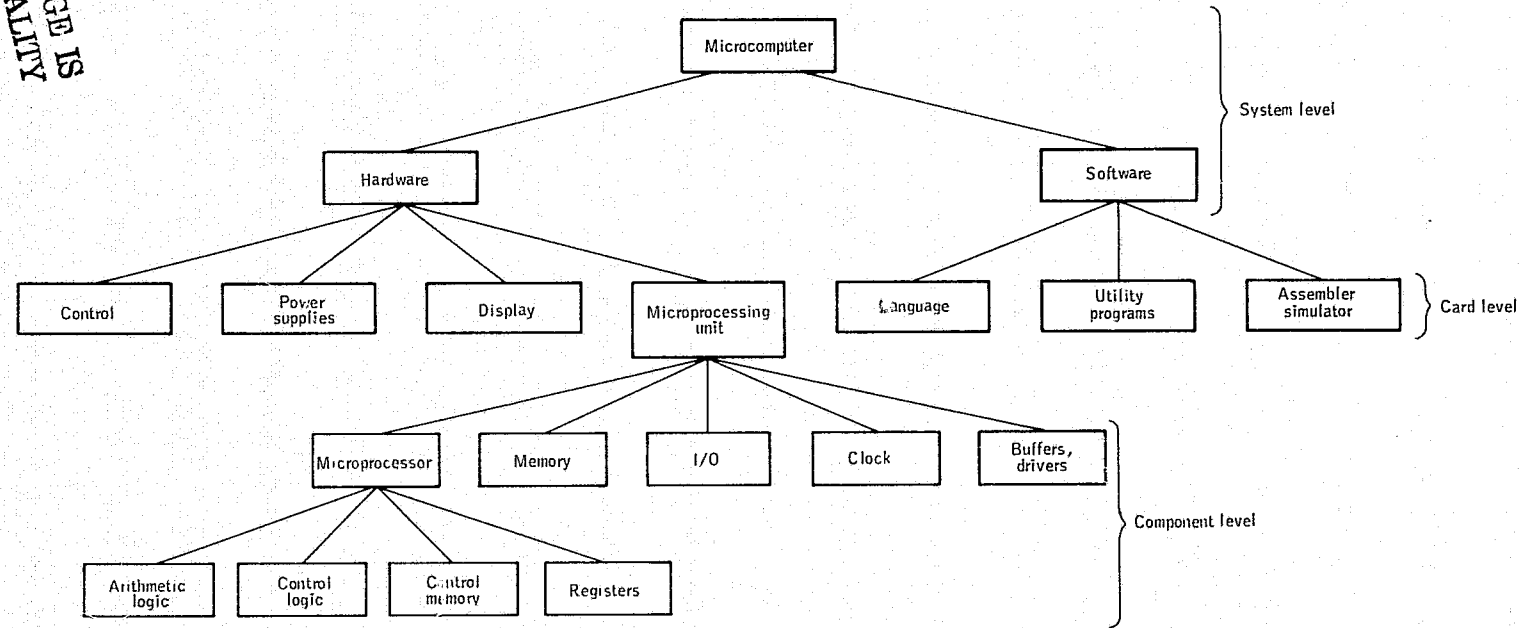


Figure 4. – Integration Levels of a Microcomputer

- Bit Slice Partitioned Chip Sets -- The circuitry for CPU, registers, control, etc., is provided on a single chip or part; the part may provide a 2 to 8 bit "slice". These slices are then interconnected with 10 to 30 standard MSIC devices to form whole-word processors. An example of such an unit is the Intel 3000, which is a 2-bit slice unit that can be used to form a processor of any word length.

- Monolithic Processor -- All circuitry for the entire processor (all bits) is provided in a single part. An example is the Intel 8080, an 8-bit processor.

Microprocessing unit. — The microprocessing unit is in the medium or card level and encompasses the entire group of circuits necessary to render the processing function. It includes, in addition to the microprocessor, the microprogram memory (in microprogrammable devices) or wired memory, the clock, the I/O devices, the buffer and drivers, and the interface circuit elements where and if required. It does not, however, include power supplies and enclosures. This level of sophistication corresponds to the naked minicomputer and is called a bare-bones microcomputer by some manufacturers. If the microprocessor is used as a part of a conventional computer, the main frame memory is not included unless it is an integral part of the operation of the microcomputer subsystem.

The microprocessing unit may consist of a single card or a number of cards, depending on the circuit technology and the physical size of the printed circuit cards used. The Honeywell HDC-301 and Teledyne TDY-52B are typical examples of microprocessing units.

Microcomputer. — This is the highest or system level of sophistication and comprises an entire operating computing system. The claims by some microcomputer manufacturers that a microprocessor represents a "computer

on a chip" and that as such its price is less than $100 is misleading. Such a chip is not a complete computer; it is only the central microprocessor.

To be functional, a complete microcomputer requires additional chips for read/write and read-only memory, I/O decoding, clock generator, address batch, control and interrupt logic, and data buffer, as well as the assembly and testing on a printed circuit board, power supplies, and cabinet. Furthermore, the price of software has to be added. The total price of a complete, usable microcomputer system is one to two orders of magnitude higher than that of the microprocessor chip or even of the microprocessor plus a few memory and I/O chips. Thus a typical microcomputer, at least at present, will not cost $100, but at least five to twenty times as much.

The expressions "microcomputer" and "microcomputer system" are used interchangeably. In the degree of sophistication, the microcomputer system corresponds roughly to the minicomputer system, although there are a number of differences mainly of price and performance between microcomputers and minicomputers. The differences between the two types of small computing systems are rapidly disappearing.

The most complete survey of all forms of microprocessors currently available is Reference 7, "Microprocessor Directory and Microcomputer Systems Directory", found in the November 20, 1975 issue of Electronic Design News Magazine.

Rather than summarize the state of the art in microcomputers and microprocessors as an independent item, these units are discussed in comparison to minicomputers. For the purposes of this report, we can define a minicomputer as a complete package consisting of processor, memory, and I/O circuits, which performs at a moderate level of speed and functionality. This loose definition is sufficient as the primary reason for including minicomputer technology in this technology assessment is to allow a direct comparison with the speeds and functionality of the more important microprocessor and microcomputer technology area. An excellent summary of

available minicomputer is "The Auerback Computer Technology Report, Volume 110, Minicomputer Notebook." This document is available directly from Auerback and is updated on a monthly basis. A general survey and combination of data on minicomputers shows the typical machine to consist of the following characteristics:

- Sixteen-bit word size for internal data paths and memory - In contrast, microprocessors are typically 8-bit machines with a limited number of 4-bit and even fewer 16-bit machines available.

- Memory address space of 64K words - While some microprocessors are capable of addressing memories of this size, the number of internal operations and, therefore, time required to address a single word within the 64K is considerably longer for a microprocessor.

- On the order of 50 to 100 instruction types - In contrast, a microprocessor instruction set generally contains approximately 50 instructions. In addition, the functional capability of a single instruction in a minicomputer is often equivalent to a number of microprocessor instructions, causing an even wider gap between the two.

- Typical short instruction time of 2 microseconds - A short instruction (store, add, etc.) in a typical minicomputer will manipulate a full 16-bit operand in 2 microseconds or less. A microprocessor (or microcomputer) typically requires 6 to 8 microseconds for 8-bit operands. Microprocessor instructions, which operate on 16-bit operands via multiplexing, typically require 20 microseconds for the same short instructions.

- Multiply and divide capability - Virtually all available minicomputers provide a multiply and divide capability that operates in two's complement, signed format; typical times for these (fixed-point) operations are on the order of 10 microseconds. In contrast, current microprocessors seldom provide these more complex instructions

in the basic set. If included, the time for such operations may be on the order of 100 microseconds to 1000 microseconds, depending on whether the answer is to be generated as an 8-, 16-, or 32-bit word.

- Floating-point capability - A hardware floating-point capability is rarely provided in the basic minicomputer configuration provided by a manufacturer, although it is sometimes provided as an add-on item at additional cost. In contrast, the only way this capability can be obtained in a microprocessor is through special design by the user; there are no directly available microprocessors that have vendor-provided floating-point capability.

- Microprogrammed instruction set - Most minicomputers and microprocessors use microprogramming for the basic instruction decoding operations. (Note: There is often confusion between a microprocessor and a microprogrammed machine; the terms refer, respectively, to a type of machine and a specific design/implementation technique for any machine.) A microprocessor is not necessarily a microprogrammed machine. This is discussed in Appendix B.

- Input/output capability - Three types of output operation are of interest here: direct program control, block I/O, and direct memory access (DMA). For direct program control, CPU processing time is required to fetch the I/O instructions and to fetch the data itself. In the case of block I/O, CPU processing time is spent fetching the data only. The I/O instructions are fetched once per block. In contrast, the CPU is involved in DMA only to the extent that it cannot access the memory while the operation is occurring; CPU and I/O accesses are generally interleaved. Minicomputers generally provide both direct I/O and DMA, and sometimes the block form of I/O operation. Microprocessors generally allow only direct I/O, although DMA is provided in a limited number of cases.

- Vendor support - Software packages, documentation, packaged (hardware and software) interface options, and general user assistance is normally provided with a minicomputer. This type of vendor support is not as prevalent for microprocessors, but is available on some of the more widely used units.

- Cost - The original equipment manufacture cost of a minimal (4K, 16-bit words of memory) configuration for a minicomputer is on the order of $2K to $5K. Additional memory, CPU options, etc., can increase this cost considerably. The cost of a minimal microprocessor (no memory provided) is on the order of $50 to $500, depending on whether the processor is provided as parts for assembly by the user or as a complete, assembled PC card. A typical computer may contain 256 to 8000 8-bit words of memory; the cost for these units varies from $500 to $5000 depending upon packaging, control panels provided, etc.

Memory Technology

The main emphasis in memory technology has been towards the development of larger and faster memories. Today's memory devices can be classified into two basic categories:

- Fast and relatively expensive electronically accessed main memory.
- Slow and inexpensive electromechanically accessed peripheral memory.

Because of size and low speed, electromechanically accessed memories have had very limited use in airborne computers (none in real time applications). For example, cassette tapes have been used to store programs or data for navigation and/or maintenance application. This use will continue but will not greatly influence the performance of future avionics; accordingly, data on movable memory devices is not included in the survey.

Currently, two electronically accessed technologies predominate: semiconductor storage and magnetic storage. These technologies are used to implement a number of memory forms. These include:

- Random access memory (fast read-write)
- Read-only memory (ROM)
- Electrically alterable (ROM) (fast read-slow write)
- Block-oriented RAM (serial transfer of data blocks)
- Content-addressable memory

The areas in the memory hierarchy where these memory forms are used and more detail on the devices are given in Appendix E.

In recent years there has been considerable effort in MOS to develop a high-speed, nonvolatile RAM. While this development has not yet resulted in the desired speed, electrically alterable nonvolatile ROM's (MNOS) have been commercially produced. Such memory devices can be very useful in remote vehicles such as satellites, where data alteration through ground intervention is desirable. The development of a high-speed read/write semiconductor RAM that is nonvolatile would be of significant benefit and should be encouraged.

As in semiconductor logic, the fastest semiconductor memories are fabricated in bipolar, the most complex in MOS. Access time of semiconductor memories today range from 20 to 500 nanoseconds and will be at least twice as fast for a given chip complexity in the 1980 time frame. While most semiconductor memories are random access in nature, block-oriented memory chips are currently under development with access times in the microsecond region.

The most common technology found in memory systems today is magnetic storage. The main problem with this technology is the difficulty of interfacing with the semiconductor devices with which they must communicate. Another shortcoming has been the need for electromechanical access when

the memory capacity exceeds 10^7 bits. Recent developments in magnetic bubbles promise to extend the electronically accessed range to 10^9 bits.

The general characteristics of the most important nonmovable storage technologies are ranked with respect to avionics applicability in Table 7. At present, only magnetic and semiconductor technologies are used extensively in available, off-the-shelf memory. Each technology is summarized in Table 8 with respect to five quantifiable parameters. For semiconductor memories, a further breakdown is given between (volatile) RAM and programmable ROM.

Input/Output Technology

Data bussing and multiplexing. – Bussing and multiplexing techniques for interconnecting separate processors within a system and for connecting I/O terminals to processors are becoming popular in all types of real-time control systems. However, this particular area cannot be summarized with "typical" values, for two reasons:

1. The technology is relatively new and has not matured to the point where a single approach is accepted and used within specific applications.

2. The details of a bussing system design will vary considerably depending upon the area of application.

TABLE 7.- CHARACTERISTICS OF MAJOR NONMOVABLE STORAGE TECHNOLOGIES WITH APPLICABILITY OR POTENTIAL APPLICABILITY TO AVIONICS

Storage technology	Speed	Power consumption	Size and weight	Volatility	Reliability	Temperature range
Semiconductor						
Bipolar	VH	H	L	R/W	H	H
MOS	H	L	L	R/W	H	M
C-MOS	H	VL	L	R/W	H	M
SOS/MOS	H	L	L	R/W	H	M
Magnetic						
Core	H	H	M	NO	M	L
Plated wire	H	H	M	NO	M	L

Legend: VH - Very high
H - High
M - Medium
L - Low
VL - Very low
R/W - Read/write

TABLE 8. – CHARACTERISTICS OF MAGNETIC AND SEMICONDUCTOR MEMORY TECHNOLOGIES

Technology	Parameter
Magnetic	
Plated wire	Nonvolatile Access time: 100-1000 nsec Density: 10^3-10^4 bits/cm^2 Subsystem cost[a]: 2.1¢/bit Power consumption: 500 uW/bit operating/0 uW/bit standby Capacity: 4K bits and up
Core	Nonvolatile Access time: 100-10000 nsec Density: 10^3-10^4 bits/cm^2 Subsystem cost[a]: 4.0¢/bit Power consumption: 600 uW/bit operating/0 uW/bit standby Capacity: 4K bits and up
Semiconductor	
Bipolar	Access time: 110-100 nsec Density: 10^4-10^5 bits/cm^2 Subsystem cost[a]: 2.6¢/bit Power consumption: 1000 uW/bit operating/50 uW/bit standby • Read/write (RAM) – volatile – capacity: 1K bits/chip • Read only – nonvolatile – capacity: 8K bits/chip • Programmable read only – nonvolatile – capacity: 4K bits/chip
MOS	Access time: 80-2000 nsec Density: 10^4-10^5 bits/cm^2 Subsystem cost[a]: 1¢/bit Power consumption: 300 uW/bit operating/10 uW/bit standby • Read/write (RAM) – volatile – capacity: 16K bits/chip • Read only – nonvolatile – capacity: 48K bits/chip • Programmable read only – nonvolatile – capacity: 4K bits/chip

[a]Costs per Bit in 1975.

The points of variance between systems are:

- Topology - The physical or geometric pattern used for interconnection

- Control Mechanism for Bus - This may range from a single-point (central) control scheme to systems in which all terminals (processors) may control the bus at different points in time.

- Communication Mechanism - There are a large number of options available regarding coding, modulation, and data transmission techniques.

- Speed - This will vary with the transmission media (twisted pair, coax cable, etc.), distance between terminals and/or processors, number of terminals (drops), etc.

- Data Format - This deals with issues such as bit-serial versus word-parallel data transmission, inclusion of definitive information for various bus management functions, organization of the data into specific messages, etc.

- Error-Control Techniques - This may vary depending upon whether error detection is important, whether bus recovery is desired or necessary upon occurrence, etc.

To indicate the extent of variance possible, two existing bus designs are summarized in Table 9.

With the introduction of LSIC-implemented bus terminals and the use of (wide-band) fiber optic transmission media, it is sufficient to say that in a given application, there is an unlimited range of alternatives in bussing.

The initial general-aviation data bus applications will probably be triple-redundant busses or a dual-ring bus structure with zone disconnects such that the system can perform in spite of open lines or shorts to ground.

TABLE 9. - CHARACTERISTICS OF THE MIL STD 1553A BUS AND ALL APPLICATIONS DIGITAL COMPUTER (AADC) INTRASYSTEM BUS

Characteristics	MIL STD 1553A Bus	AADC Intrasystem Bus
Topology	Linear, all terminals connect to a single global bus, 300 feet maximum length	Linear, 3 feet maximum length
Control mechanism	Centralized with polling of all terminals for message (bus usage) requests	Decentralized, round-robin passing of control
Communication mechanism	Phase modulation, manchester (biphase) coding	Non-return to zero (NRZ), request/ acknowledge
Speed	1-megabit data rate, 2-megabit bandwidth	150 nsec per 32-bit word
Data format	20-bit basic word with 16 bits of data, transmitted bit serially	32-bit data word transmitted in parallel
Error control techniques	Parity over 16-bit data	Parity over 32-bit word

Handling the aircraft, engine, and flight control communications through digital multiplexing to central processors could reduce the number of interconnects and improve system noise immunity by performing analog to digital (A/D) conversion at remote analog sensor locations. A complete survey of bussing and multiplexing is beyond the scope of the report. Reference 8 gives more information.

Some electronic systems and components manufacturers indicate that they expect automotive electronics to develop in an orderly progression of discrete, combinational and central processor-data bus phases over the next ten years. The safety aspects to be perfected before use of a central processor-data bus in general-aviation avionics will undoubtedly require additional time. Therefore, data bus application in general aviation is unlikely prior to 1990.

Interface components and modules.- Because natural parameters such as temperature, displacement, and pressure are analog and because the more advanced methods of data handling and computation are digital, conversion between analog and digital quantities is a fundamental operation in computing and control systems. Thus analog-to-digital (A/D) or digital-to analog (D/A) converters are found at all interfaces between sensors of physical quantities and digital computer and output devices.

Digital I/O function modules, analog I/O function modules, and I/O components may provide some of the same functions performed by the converters. Also, digital transducers, which are another type of I/O module, are discussed in this subsection. More information on these I/O modules and components is included in Appendix F.

Converters: Despite the widespread use of A/D and D/A converters, there is little standardization and there are no product lines that dominate the converter business sufficiently to provide a practical basis for standardization. These difficulties are compounded by the diversity in nomenclature

and error definition. Changing manufacturing technology, development of new semiconductor components, and varied application requirements, when combined with minimal standardization, lead to a growing variety of available converters. Furthermore, as the conversion process varies, similarly specified units may exhibit subtle but important differences in behavior. As a result, choosing the best converter for a specific application requires careful consideration. Converter selection criteria are discussed in detail in Appendix F.

Because A/D converters are essentially interface devices, the basic conversion circuitry must be adapted to suit different applications. Adaptation includes adding registers, buffers, digital clock, and reference voltage; these are frequently provided external to the converter. The exact definition of what is included in the converter module obviously has an important impact on price and makes it difficult to compare competitive models.

Hybrid D/A and A/D converters are being greatly integrated such that units requiring no outboarded devices for operation are becoming more common. Monolithic data converters having 8- or 10-bit resolution are now readily available. However, most converters require one or more associated external components.

Development of complete monolithic converters, both D/A and A/D, having 12-bit resolution is not far off. However, there is now a whole array of 10- and 12-bit monolithic analog and digital building blocks available for making either A/D or D/A conversions. They include successive-approximation registers, quad switches, and precision current sources.

The epoxy-encapsulated variety of modular converters continue to offer the ultimate in performance. For example, conversion time for a 12-bit modular A/D converter is now down to 2 microseconds. Also, complete data acquisition systems have become available in modular form. These units – low-profile packages smaller than a human hand – are multiple-channel systems, containing A/D converters, sample-and-hold circuits, multiplexers, and sometimes even programmable logic.

For pricing comparison, D/A converters may be grouped according to resolution, with performance parameters such as accuracy generally corresponding with resolution. At the lowest end are 4- and 6-bit converters for use in low-accuracy programming applications and in digital communications; these sell typically for about $20 per device. Converters having 8- and 10-bit resolution are essentially similar except for the number of bits, with a relatively small price differential for the larger number of bits. The 12-bit converter is the most accurate of the readily available production types and uses precise analog components, which are nevertheless inexpensive and widely available. It is no coincidence that 12 bits corresponds to the accuracy limit of most analog instruments and control systems. Digital to analog converters in the 8- to 12-bit range typically cost $10 to $20 per bit; a good 12-bit decoder costs about $200.

Converters having 13- to 16-bit accuracies are used in very high-accuracy or wide dynamic-range applications. Their cost can easily be many times that of the 12-bit converter, running $10 to $50 and more per bit. At this resolution level, every additional bit adds substantially to cost. Of course, the user of high-resolution devices must be very concerned with long-term stability, susceptibility to noise, response or settling time, trimming adjustments, and similar factors that must be controlled to ensure that the specified accuracy is actually achieved.

High-speed converters of 15 or more bits represent state-of-the-art performance and are consequently very expensive. Because it is expensive to even test such units, there is skepticism about long-term performance. Where speeds above about 1 MHz are required, prices can soar to many times those of the low-frequency equivalents with the highest speed units costing twenty times more than lower speed units.

Technological advances in data converters have been snowballing for several years particularly for hybrid and monolithic devices. There are now many more of all types from which to choose (Table 10). The biggest factor in comparing digital converter prices is assuring a uniform basis for

TABLE 10. – A SURVEY OF DATA CONVERTERS

Technology	Resolution, binary bits	Digital-to-analog converters			Analog-to-digital converters [a]		
		Settling time	Notes	Approximate small quantity price, dollars [b]	Conversion time	Notes	Approximate small quantity price, dollars [b]
Monolithic	8	85 ns – 1.5 μs	All units require external reference source and operational amplifier	10 approx.	18 μs approx.	All complete	12 approx.
	10	250 ns – 1.5 μs	Some complete; others need reference source and/or operational amplifier	25 – 100	40 μs approx.	All require external clock and comparator	75 approx.
	12	500 ns approx.	All require external reference source and operational amplifier	35 approx.	-	-	-
Hybrid	8	10 μs – 23 μs	All complete	10 – 50	1 μs – 60μs	All complete	60 – 195
	10	23μs approx.	All complete	60 approx.	As low as 2μs	Some complete; others need external register	35 – 110
	12	3μs – 23μs	Most complete; a few need external operational amplifier	30 – 150	8μs – 50μs	Most complete; a few need external reference source and/or register and clock	25 – 275
	16	100μs approx.	All complete	125 – 150	20μs – 50μs	Integrating types only	75 – 250
Modular	8	25 ns – 20 μs	Both current- and voltage-output types	10 – 255	800 ns – 250μs	All complete	60 – 475
	10	25 ns – 5μs	Both current- and voltage-output types	20 – 200	1μs – 300 μs	All complete	80 – 485
	12	50 ns – 20μs	Both current- and voltage-output types	35 – 180	2μs – 350μs	All complete	115 – 585
	14	1μs – 250μs	Both current- and voltage-output types	250 – 850	10μs – 50μs	All complete	200 – 700
	16	750 ns – 250μs	Both current- and voltage-output types	100 – 1,500	5μs – 400μs	All complete	1,400 – 3,000

[a] Principally successive-approximation types.

[b] For commercial-grade products.

comparison. It is easy to make the specification sheet of a low-cost converter appear superficially equivalent to that of a more costly unit; factors relating to stability, noise sensitivity, and other important parameters may be omitted or inadequately defined. Completeness of the converters, with regard to system interfacing, may vary substantially among various choices such that the lowest cost device ultimately becomes the most expensive when all extra interfacing is included. Further, tieing in the interfacing can require costly debugging due to subtle incompatibilities of hardware and problems relating to system layout and wiring. In general, when performance requirements are high and quantities moderate, shopping for price is usually a poor decision.

Because there is no industry-wide standardization, second sourcing can be a problem. This emphasizes the importance of choosing a reliable manufacturer. Sometimes a system can be designed at slight extra cost to accept either of two nonidentical converters with little or no modification.

Digital transducers: Digital transducers are defined as devices in which the parameter to be measured is subdivided into increments by a special quantized transducer, and conversion to digital form is accomplished by determining the number of increments contained in a particular motion or change of state. Digital transducers are comprised of two primary subtypes: 1) devices in which successive increments of position are indistinguishable and must be counted, and 2) devices in which successive incremental positions are coded (numbered) such that the number corresponding to any position can be determined directly.

The absolute encoder has a distinct code for every position and, consequently, has previously been preferred for its inherent reliability. The recent trends toward use of solid-state light-emitting diodes (LED's) and integrated digital circuitry in the encoder package has resulted in an improvement in the reliability of the formerly less popular incremental encoders to the point where they are challenging absolute encoders for angular position and rotational speed measurements.

With an incremental encoder, only one track is needed, in principle, to obtain an indication of position as compared to an absolute encoder with multiple tracks (number of tracks dictated by the resolution required: 8 tracks for 2^8,10 for 2^{10}, etc.). The incremental encoder is considerably less expensive, and the continuing improvements in IC's, solid-state LED's, and electronic processing will serve primarily to increase the application of incremental encoders. Reference 9 discusses a number of other changes affecting digital encoders.

Digital transducers for the primary parameters of temperature, pressure, position, and rate are setting the pace for the development of inexpensive control systems. Hybrid IC's, which appear to make the best and cheapest pressure transducers for the automotive market, are under development by several large semiconductor vendors.

Hybrid IC pressure transducers will include the necessary transducing elements: diaphragm with its constraint and vaccuum reference cell, piezoresistive sensor bridge, signal conditioner with temperature compensation and bridge balance, and signal amplifier. A typical pressure transducer consists of a 165 x 115 mil chip. The pressure diaphram occupies 90 x 65 mils, leaving enough area for temperature-compensating diodes, bridge-balancing resistors, and a zener regulator for the bridge power supply. In its final form, the hybrid transducer consists of a ceramic substrate with laser-trimmed, thick-film resistors, one or two operational amplifier chips, and the transducer chip.

Diaphram dimensions set the transducer's nominal pressure range, which is selected to stress the silicon to less than 15 percent of its rupture point. The diaphram can actually operate double range and will survive triple-range excursions without offset or degradation to the transducer chip.

In another of the many possible applications, IC pressure transducers will meet the needs for electronic fuel injection systems. These systems

must measure pressures with accuracy over a 1 to 15 psi range while operating in ambient temperatures of -40 to 200°F.

Honeywell has had a digital pressure transducer under development that may be applied to automotive as well as aviation uses. The details of its operation are typical of the new IC digital transducers. The pressure sensor developed by Honeywell for application to digital systems is shown in Figure 5, and a block diagram of the transducer is shown in Figure 6.

The transducer consists of a diaphragm cut from a single crystal of silicon (semiconductor material) which has resistors diffused into it. The diaphragm is housed in a small cylinder. One side of the diaphragm is evacuated, the other side is exposed to ambient pressure. Ambient pressure causes the diaphragm to flex, producing a resistance change (ΔR) proportional to strain (which is proportional to pressure).

The resistance change (ΔR) is a function of pressure, resistance, and temperature. Strain can produce resistance changes of up to 10 percent of nominal resistance, and ΔR is approximately proportional to pressure. Most temperature effects are canceled out due to sensor resistor - bridge circuit mechanization.

Signal conditioning circuitry in the sensor provides means for converting ΔR to a digital form, provides sensor temperature information in digital form, and provides constants for characterizing the sensor to provide ΔR corrected for temperature and diaphragm nonlinearities.

The sensor signal output is transmitted to a digital processor that converts the three digital signals to pressure by solving polynomial equations.

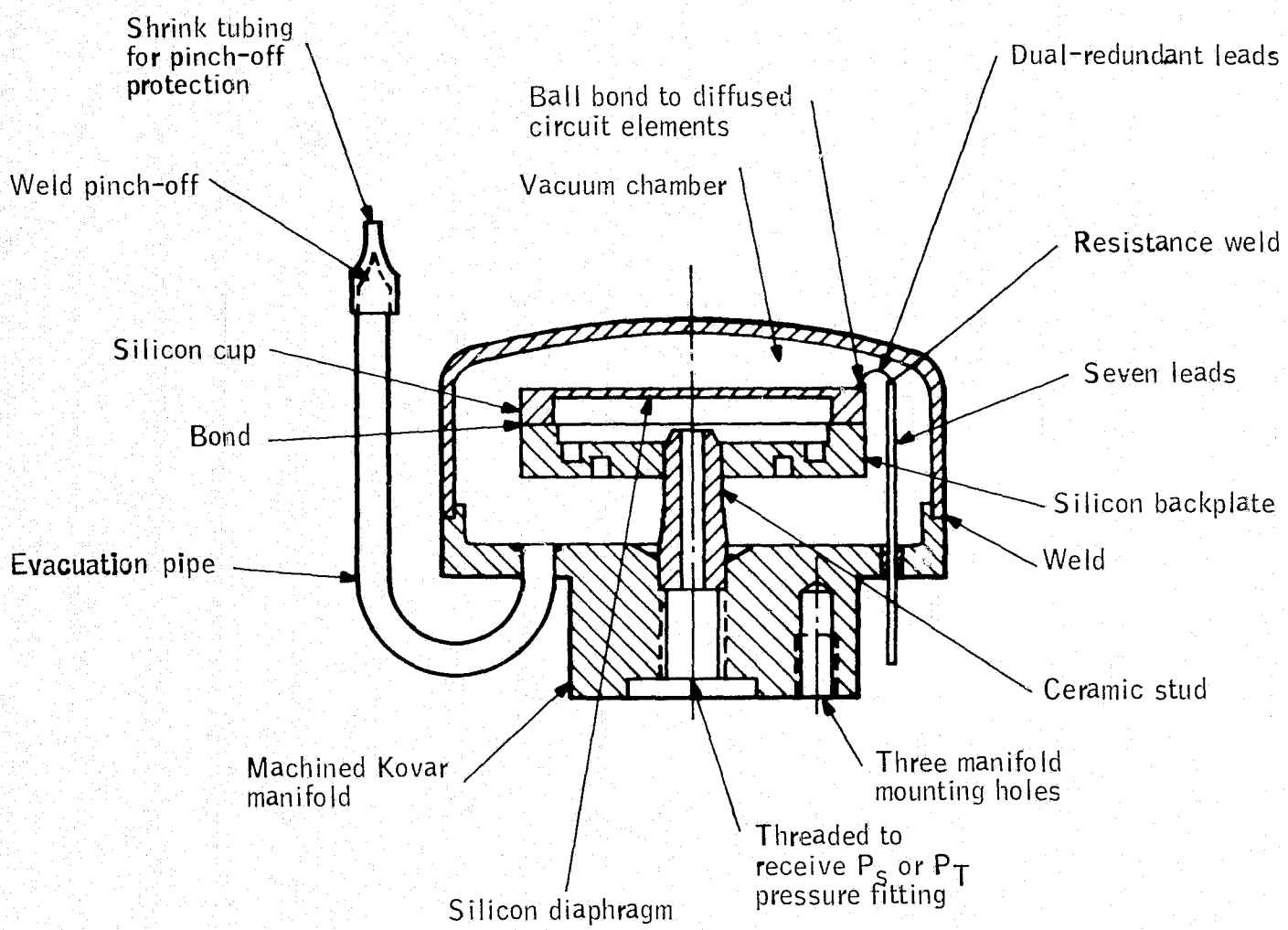


Figure 5. – Honeywell Solid-State Pressure Transducer

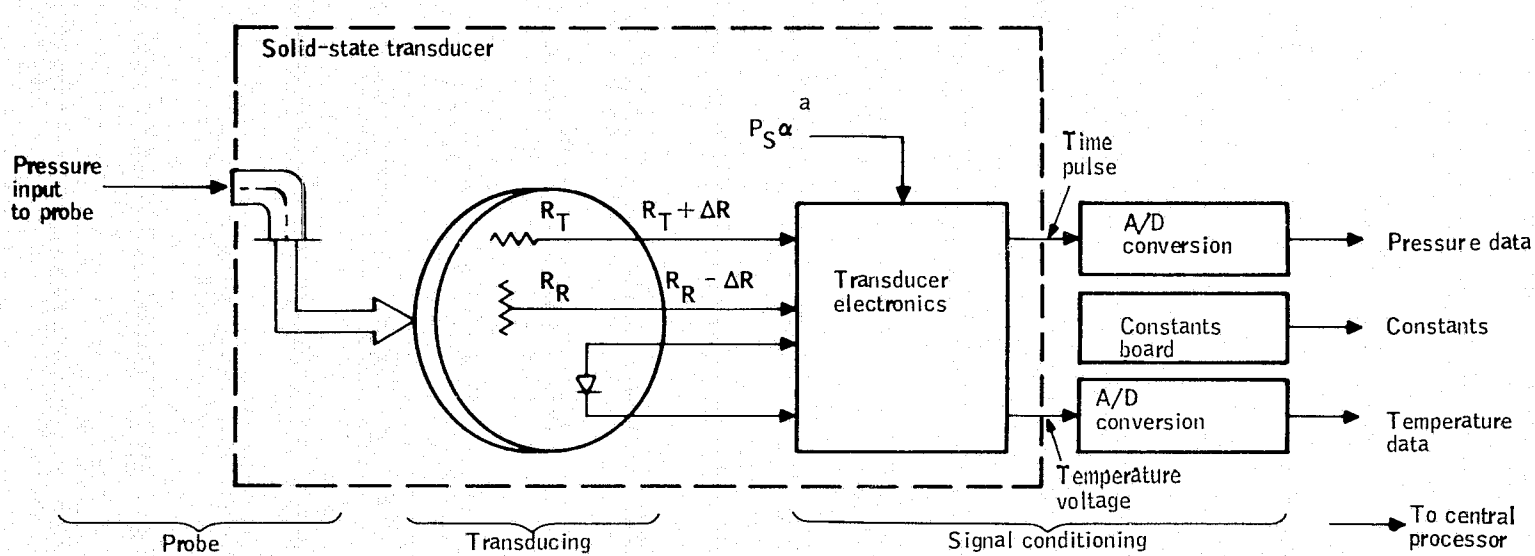


Figure 6. - Honeywell Solid-State Pressure Transducer Block Diagram

COMPUTER SYSTEM ARCHITECTURE TECHNOLOGY

General Considerations

Rapid advances in semiconductor technology have recently spawned the microprocessor -- the general-purpose, stored-program processor based on a small number of LSI circuits (ref. 10). As logic elements, these processors provide unprecedented power and flexibility to designers, and will precipitate many dramatic product developments. From a computer perspective, the current microprocessors are somewhat primitive, but they clearly presage the evolution of much more capable ones.

Thus far, it has been computer architecture which has influenced microprocessors. They first emerged in the form of small, special-purpose calculators and have steadily evolved in the direction of general-purpose computers. Now, however, that relationship is becoming less one-sided; microprocessors are having a growing effect on the organization of larger computer systems. The principal impact is an accelerating trend toward distributed function architectures, in which a larger machine is structured from a multiplicity of smaller ones (ref. 11). Historically, high processor costs have restricted this approach to those few applications that demanded (and could afford) the potential reliability and performance advantages. Because processors are no longer the limiting resource, distributed computers are becoming increasingly feasible on a wider scale.

The techniques for designing distributed computers are as yet in their infancy because the motivation to do so has only recently developed. Perhaps the most significant issues are how the computational load is to be partitioned across the processors, how the resulting processes are to intercommunicate, and what sort of processor interconnection structure is needed to support this communication.

The way in which the processors are attached to each other is largely dictated by the interprocess communication requirements, which in turn reflect the process partitioning and assignment.

Other major issues, which are discussed in detail in Appendix G, are:

- interconnection transfer strategies
- transfer control methods
- transfer path structure
- system topologies
- cost-modularity
- place-modularity
- connection flexibility
- failure-effect
- failure-reconfiguration
- logical complexity
- bottlenecking

It is clear that the three principal distributed processing issues of process partitioning and assignment, interprocess communication, and processor interconnection are intricately related, and must be traded off for specific applications. Some unnecessary and distracting separations of them have traditionally been based on a dichotomy between hardware and software implementations of various functions (ref. 12). Certain aspects of contemporary distributed processing approaches suggest that conventional hardware/software interfaces should be re-evaluated; advances in hardware components and implementation techniques are making it possible to combine flexibility with low cost and high performance.

Distributed computers can be generally categorized according to the way in which the processing is partitioned among the processors. Processors

may function as: special-purpose components of a larger processor, dedicated ancillary processors supporting the main (i.e., central) processor(s), or as multiple main processors. Any or all of the levels in this hierarchy may appear in a particular distributed computer, as illustrated in Figure 7.

Processor Subprocessors

Microprocessors may themselves be special-purpose components of a larger processor, performing functions previously done by hard-wired logic or software, or not done at all.

A function that must occur in all processors is instruction and operand preparation: selecting the next instruction to be executed, forming its effective address, obtaining it from memory, forming the effective addresses of its operands, and accessing them. This may involve dealing with such issues as protection and sharing; address translation; memory and bus contention, arbitration, and allocation; faults; etc. Frequently, these operations could be overlapped with the execution of previously prepared instructions on previously prepared operands. The extent to which this form of pipelining can improve the computer's performance depends on such factors as exactly how it is implemented and locality aspects of the software. Usually the look-ahead mechanism is part of the processor (ref. 13), but may instead be part of the memory (as considered later).

The incorporation of a dedicated processing component to perform certain arithmetic functions within a main processor may offer both modularity and performance advantages (ref. 14). If the functions are optional, placing them in this separate processor may provide a very clean interface to the rest of the machine. Rather lengthy computations (such as byte string or trigonometric operations) can sometimes be carried out by a separate processor in parallel with other functions.

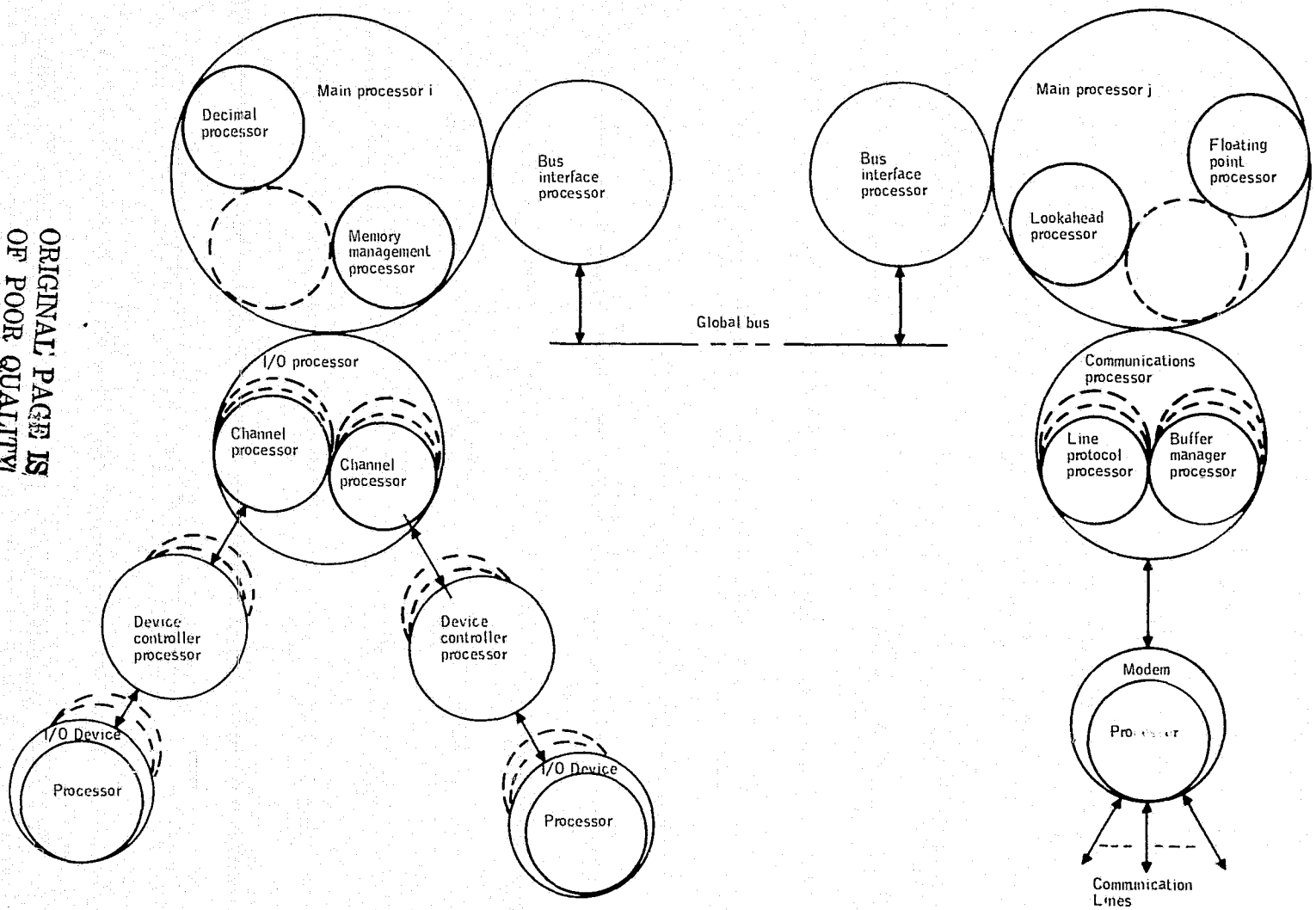


Figure 7. – Examples of Microprocessors in a Distributed Computer

Occasionally very special-purpose tasks need to be performed, such as graphics, signal processing, etc. These may call for either internal processor components as described, or external ancillary processors.

Dedicated Ancillary Processors

Microprocessors can be very effective in the capacity of dedicated function processors supporting the main processor(s). Input/output processors have been a feature of larger computers for some time (ref. 15) and are becoming commonplace on smaller ones as well. Microprocessors bring the same decentralization advantage to I/O processors as to entire computers; the I/O processor can itself be decomposed into individual channels, each controlled by its own microprocessor (ref. 16). Autonomous channels not only have potentially higher bandwidth, their increased intelligence facilitates the performance of functions such as code and format conversions; error and exceptional condition handling; buffer and queue management; device address translation; protection; routing and device selection; seeking, searching, label checking, etc. Microprocessors are also seeing increased use in I/O device controllers to handle complex interface functions (such as for disks [ref. 17]) and to eliminate specialized hard-wired controllers (ref. 18 and 19). At the bottom level of this hierarchy, microprocessors are also appearing in the I/O devices themselves.

A special type of I/O processor worth noting is one which is used for communications applications. Microprocessors are becoming well suited for handling the line protocols, data compression, routing, monitoring, testing, formatting, etc. (ref. 20).

Memory continues to be one of the major cost factors in a computer, so its efficient management is an opportunity for employing microprocessors. Memory management includes address translation, protection, segment/page replacement, garbage collection, etc. It may also include instruction and operand preparation, as discussed earlier. Associating this function with

the memory rather than the processor has potential advantages in certain multiprocessor configurations, i.e., use of the bus connecting the processors with the memories can be reduced from four cycles (instruction address, instruction, operand address, operand) to one (operation code and operand), thus reducing bus contention and raising throughput (ref. 21). The programmable capabilities of a microprocessor are often useful in conventional memory management tasks, but are virtually indispensible for this "distributed fetch" concept. Carrying this idea one step further results in a multicomputer, one of the multiple main processor cases considered next.

Multiple Main Processors

Over the past decade, parallel and associative processors have received considerable attention in the literature, but high hardware costs have restricted actual construction and experimentation. The advent of the microprocessor has revitalized interest in these architectures (ref. 22) as now it is becoming economically feasible to build them. While parallel and associative processors have rather limited application, they are good examples of the trend toward dedicated and special-purpose machines which has been precipitated to a large extent by microprocessors.

The more general-purpose multiprocessor and multicomputer (in which processors are interconnected by I/O channels rather than a common memory [ref. 21)] architectures are also benefiting from decreasing processor costs; it has been difficult to find instances of these organizations having greather than three processors until recently. The necessity of maximizing processor use has led to complex multiprogramming executive software. Even if one could afford multiple processors, using them efficiently usually required so much overhead that system performance rarely improved as expected and often even declined when additional processors were added. Now, when the processor portion of system cost is heavily outweighed by the software and configuration portions, the economic incentive to optimize processor use is diminishing. By employing suitable

intercommunication mechanisms for both hardware and software, distributed architecture can be devised which permit easy incremental changes in processing power using standard, low-cost modules.

One consequence of this modularity is the potential for overcoming the normal price/performance discontinuities between smaller and larger computers. Within a general-performance class of machines (such as microcomputers or minicomputers), an increment in price buys a proportional increment in performance (whether a consequence of technology or marketing factors). However, when the desired increment in performance crosses a boundary between classes of machines (e.g., from minicomputers to "full size" data processing computers), there is generally a large step in price; to get ten times better performance than a particular machine may cost one thousand times the price of that machine. On the other hand, it may be possible to interconnect fifteen copies of that machine at perhaps thirty times its price (allowing for the interconnection cost) to achieve the desired performance. Thus, the availability of very inexpensive though limited-capability microprocessors should be of interest, even to many of those who need higher performance than just one microprocessor can supply.

The microprocessor possesses two important features that gainfully exploit LSI technology. It takes a minimum number of chips to configure a dedicated function. Also, it permits the easy formation of microprocessor networks.

There are many systems now using a single processor/memory approach that are ripe for a multiprocessor application. Such systems include point-of-sale terminals, electronic automobile control and general-aviation avionics.

If the multiprocessor approach is used, unique controllers, sensors, and long wire runs to the CPU for each input could be replaced by a more modular arrangement. The heavy load on a single CPU could be replaced by light loads on multiple CPU's. Software would be simplified in some cases,and shorter runs at the lower data-transfer rates used purely for the exchange of processed data could be substituted.

Current efforts in the direction of multiple main processors include relatively unconventional architectures (ref. 24 and 25), as well as more traditional ones (ref. 26). It should be noted that distributed computers as considered here differ considerably from computer networks, both in perspective (building one larger computer from a number of smaller ones, versus load and resource sharing among a number of individual computers) and in environment (electrically and geographically short interconnections, versus electrically long and geographically long [ref. 27] or short [ref. 28] interconnections).

While microprocessors are expanding the possibilities in computer system architecture, at the same time they are imposing some limitations; the primary limitation being that the processor designs will be determined by the semiconductor industry and will less frequently be subject to the design influence of the computer system architect.

SOFTWARE TECHNOLOGY

The software technology areas relevant to future avionic systems' requirements are modularity, higher-order languages (HOL), software verification, validation, and support, and avionic software libraries. The prime benefit of progress in these areas is reduced software development costs and increased reliability. A summary of software technology is given here, and Appendix H reviews this area in depth.

The concept of software modularity relates to work division and functional software building blocks. Implementing modularity begins with partitioning a system into well defined subtasks. Each subtask results in a program module with well defined inputs and outputs that can be developed, debugged, and validated independently of each other. For example, interrupt routines and input/output service routines are typical executive function modules. Manageable work assignments can then be made for each module.

Software systems are usually defined with a functional hierarchy. High-level program modules can be divided into submodules consistent with this hierarchy; conversely, the submodules at lower levels can be considered as building blocks for the higher levels. The levels extend from the machine object code level to HOL source level. Adherence to modularity can greatly reduce the software development effort, especially at the debug and validation stage. It must be done carefully, however, so as not to introduce functional rigidity in an environment with changing requirements.

Software functional modularity is strongly influenced by the level of the programming language used. In a high-level language, a compiler is used to transform each source statement into one of many machine-level object code instructions. Also, high-level language symbols have more complex attributes than the simple values associated with low-level language symbols. Thus, the use of high-level languages guides program structure, forces discipline, and facilitates modularity.

There are several general requirements used to select a particular HOL. The language should be commonly used to facilitate the training of programmers, transferability of code, and the availability of useful software library functions. The language should be consistent, general, and easily understood; it should also be efficient in terms of writing speed, compile speed, and memory requirements/execution speed of the compiled code.

The specific requirements of avionics include several types of operations and functions that must be represented in the language. These include mathematical functions, conversions, sorting, list processing, data packing and unpacking, low-level I/O, and digital communications control.

Some of the major HOL candidates for avionics applications include FORTRAN, JOVIAL(J3/J73), PL/I, SPL, and CMS-2. Each of these languages has a particular set of advantages and disadvantages in avionics applications. The choice between them should be guided not only by the general language requirements and specific avionics requirements, but also the new concepts that are developing in higher-order languages. These include the relationship between data and program structure, the concept of an extendable language, source-level modularity, and structured programming.

The use of HOL's can greatly facilitate software verification and validation. The procedure for testing software is: select test cases, evaluate execution results, and estimate correctness. The use of HOL's, modularity, and structured programming techniques reduces the quantity of errors and isolates those that remain to a functional area.

An additional path for solving some of the current software problems is to provide hardware support to various software functions. This "bottom-up" approach provides clean interfaces, enforced standards, faster execution, and a lower level of software detail; however, it results in a loss of flexibility and increased size, weight, power and component count. The prime areas for using this approach are executive and communications functions.

An avionics software library can greatly reduce recurring software costs. The software modules must be developed to span the levels from an avionics function such as navigation or air data processing, through subfunctions such as flight control, to the subroutine level such as mathematical or I/O routines. The types of functions that are most useful in library

form are those that are not highly application dependent such as LORAN and inertial navigation functions, which can be modularized for inclusion in an avionics software library.

In practical applications, the cost effectiveness of higher-order languages (HOL's) for production systems has been a subject of controversy for several yeras. It is true that the use of a HOL will reduce programming time, particularly when the programmer is not familiar with the processor being used and when the HOL compiler is efficiently and completely developed. What is sometimes not considered is that in avionics computers, the I/O and I/O control often affect the timing of operations to the point that a HOL compiler for a particular processor must be tailored to fit the specific application of that processor in a computer. To the extent that this is true, it would create problems (or at least design restraints) for an avionics manufacturer who bought a processor and a HOL compiler to use it in constructing his own computer. In general, it seems doubtful that for simple digital applications the HOL will be the better approach; however, as the digital capability (throughput and memory) is increased, the HOL approach will become more attractive and may be desirable, particularly for the manufacturer who buys a processor or computer to put into his hardware. It is also true that a HOL is quite useful in system development when it is desirable to be able to make frequent changes to software programs. There may be situations when it would be advantageous to use the HOL in development and shift to assembly language for programming the final production hardware.

OPERATOR/COMPUTER INTERFACE TECHNOLOGY

Digital techniques are changing military and commercial aircraft crew station layout concepts. The evolution of digital techniques is following the trends in information processing and interactive man/machine interfaces, which are taking place in computer graphic terminals, as well as the results

of intensive experimental psychological studies of the man-machine relationships. The general-aviation cockpit design requirements that are tailored to smaller, cheaper and lighter aircraft appear in direct conflict with the complex advanced developments, but an obvious solution to the problem of making the most efficient use of available cockpit space is the appropriate integration of selectable controls and displays. For example, fuel quantity could be stored but not displayed until requested or until 1/4 full is reached. If desirable, rate of consumption (lbs/hr) could be computed, monitored, and warnings issued if the rate becomes too high.

If one excludes those controls needed for emergency and safety of flight purposes, there remains a vast number of independent avionics control and display functions that are used comparatively little and yet continue to occupy valuable cockpit space. A more efficient use of available space could be achieved by the application of programmable controls and displays, which can be quickly reconfigured to suit the modes and functions desired at some specific time.

Computer Input Technology

Innumerable alternate computer input methods have been developed within the computer industry. Several types that might possibly be useful in digital avionics are discussed in this subsection.

Graphic data entry. – Graphic data entry offers many intriguing possibilities for simple interactive communication with a central computing system. The hardware developed to date is primarily for laboratory and commercial use, and no avionics applications of this technique have been defined. Because graphic data entry is based on use of a relatively powerful central computer and very specialized hardware for data entry and display, application in general aviation does not appear to be likely in the near future.

Hard-copy entry. – The commonly used hard-copy entry methods include punched cards, paper tape, magnetic tape, and magnetic cards, and their continued use for digital avionics appears likely. The reprogramming capability offered by small, magnetically coded cards such as those used with the Hewlett-Packard HP-65 calculators appears to be a technique that may readily be expanded and improved for in-flight general-aviation avionics applications. The low cost and small size of present digital cassettes will undoubtedly be further improved, resulting in widespread application of this form of magnetic tape input for preflight test programs, as well as loading of operating programs, and bulk data for R-Nav, etc.

Voice entry. – The U.S. Air Force has recently announced the development of a voice-input code identifier that enables a computer to recognize spoken digits and certain command words. Several other similar developments have also been reported in the literature (ref. 29). The primary disadvantage of these systems is the relatively high cost associated with the very limited number of recognizable inputs. Also, regional accents create significant problems. The equipment developed to date is complex, needs extensive memory capability, and does not yet appear to be near a production status. Therefore, it does not appear that even high-cost voice-entry devices will be available in any quantity for the 5 to 10 years. Thus, there is little likelihood that such equipment will be available for general-aviation use in the 1980's.

Keyboard entry. – The most common application of keyboard entry to digital systems has been the use of teletype consoles. Most avionics applications do not require the full alphabetic set of characters, and consequently, the number of keys may be reduced. Keyboard configurations may range from a fixed keyboard set to multiple projected legend push-button switches and displays, programmable keyboard and displays, and to fully interactive keyboards.

With the advent of multiple projected legend push-button switches and displays, a new degree of flexibility and design freedom has been achieved. A push-button switch and display can be used for more than one function; i.e., they may be time-and function-shared. These devices, however, have complex designs and require some changes if a word, symbol, or letter of the total repertoire is to be changed. In addition, if multiple legends or annotations to the legends are to be displayed simultaneously, the overall flexibility (multiple functions and modes) would be reduced, as the total number of messages is fixed.

With the programmable keyboard and display, the limitations and problems encountered with multiple projected legend push buttons and displays can be eliminated. The number of different legends that can be displayed on a given push button or display will not be a function of hardware complexity or size, although the number of alphanumeric characters and/or symbols that can be displayed at a given time will be, of course, limited by the push button or display size. The total repertoire, however, becomes a software and computer function. The total number of push-button switches required in a cockpit is at present determined by how many control options the operator requires for the worst-case condition plus N for growth. In the future a programmable keyboard can essentially solve the problems of space limitations and control location priorities for the designer.

An advantage of the programmable keyboard is that if the designer has to add control functions after the total number of switches has been cast in concrete, he may accommodate the growth without a hardware design change. This can be accomplished by a staging technique, where a hierarchy of control formats are preprogrammed into the system by software or hardware. The hierarchy extends from the top level overall control/system interface down to the detailed mode option selection. The hierarchy can be organized at the discretion of the systems engineer and programmed into the computer.

Low cost of ownership is another advantage of the programmable keyboard and display. This is made possible by the reduction in units required for the system (with resultant savings in support costs) and the minimum costs of modifications required through the system's life cycle.

To perform the integrated avionics functions that have been described, a control and display system must provide at least the following functions simultaneously:

- A primary alphanumeric display surface
- A programmable keyboard/pushbutton array of at least 25 keys. The programmable keyboard provides the computer data entry, mode and function selection, and status. This is achieved by using a transparent keyboard overlay on the display surface.

The application of fully interactive keyboards with structured, programmed, decision-free control offers the advantage of consolidation of the large number of independent control panels normally found in aircraft cockpits. This type of multi-function control is particularly beneficial in checklist and routine monitoring, and initializing of aircraft systems. Also, the use of interactive keyboards provides structural dialog with the computation system, ensures that adequate communications are established, and avoids errors or oversights. The principal objection to this format is that it does not permit the speed of reaction required for critical emergency procedures.

Computer Output Technology

This subsection describes those output devices not covered in the display or D/A converter discussions. Analog output devices that can be driven by D/A converters have not been included in this study because they are essentially the same as those used with pure analog systems. It should be noted that servos, particularly for flight control, are critical to the cost and performance and that improvement is highly desirable in both these areas.

Hard-copy output, audio output, and mechanical output devices driven by digital signals, pulse trains, or discrete-pulsed signals will be discussed.

Hard-copy output devices. – Hard-copy output devices are represented by magnetic tape recorders, paper-tape punches, card punches, printers, graphical recorders, and X-Y plotters. In-flight use of such devices has been very limited because of their weight and volume requirements as well as cost. In the past few years however, the development of printers for use with small electronic calculators has resulted in equipment that is suitable for those general-aviation applications requiring hard-copy output (ref. 30).

Typical, larger, commercially available printers may also be applicable and are available from Practical Automation, Inc. They come in a number of sizes (18, 35, 60 and 80 columns), but the 18 column size appears most suitable for general-aviation use. These printers use ordinary paper up to 8-1/2 inches wide, and ribbon or impact sensitive paper rolls. A 7 by 5 matrix produces an ASCII set of 64 characters. The serial input rate is 110 characters per second. The price for the 18 column size is $140 (quantity of 100). Overall dimensions are 3 inches in height by 3-1/2 inches in width by 7 inches in diameter.

Audio outputs. – Auditory displays commonly used for alerting or warning purposes have included whistles, buzzers, bells, horns, chimes, and sirens. These devices are relatively inexpensive and will undoubtedly continue to be used for urgent warnings. The literature on audio outputs deals primarily with problems of signal-to-noise ratio, signal intensity, intermittency, and discriminability. There has been relatively little avionic use of speech as an auditory output; however, audio communication from machine to man has been developed in other applications to the point where it is a viable technology.

On-line interaction with a talking computer requires that the computer system generate speech output. The most simple way to generate speech output is by means of messages assembled by the system out of prerecorded words and phrases. Fully digital systems have been developed that store spoken phrases in the computer system. The recorded speech is filtered, converted from analog to digital form, edited as necessary, and stored in memory. Each phrase is given an identifying code. Speech output involves a reversal of the storage process. If adequate computer memory is allocated for speech storage, the audio output can be of high fidelity, indistinguishable from the voice of the original human speaker. In practice, the economics of computer storage as well as problems of voice matching results in storage of speech in a somewhat degraded form such that the resulting computer voice is still intelligible, but does not have any individualistic human quality.

Research may lead to techniques for storing only words rather than both words and phrases or it may prove more practical to work with still smaller units of speech such as the morpheme or syllable. Eventually, it may be possible to store only a set of programmed rules for computer synthesis of speech. Reference 31 discusses the direction this research is presently taking.

Master Specialties Co. has recently publicized the ARC 9500 audio response calculator, which has a solid-state synthesized voice that announces each entry and the results of every calculation. The eight-function device also has an 8-digit visual display. Each spoken word is digitized and stored permanently in its own ROM. The price of this device is $565. Another talking calculator (Speech Plus) from Telesensory Systems, Inc. ($395) uses a speech system, invented by Professor Forrest Mozer at the University of California at Berkeley, which is implemented in a custom LSI chip, developed by Silicon Systems, Inc., Santa Ana, California. These and other similar developments indicate that simple speech commands or warnings may become practical for general-aviation application in the next few years.

Mechanical output devices. — The mechanical output devices used commercially and for military avionics are also suitable for general-aviation use because they are relatively simple and inexpensive. Solenoid actuators and solenoid-held switches driven with discrete signals are the most common mechanical-output devices. The applications of solenoid-operated devices are extensive, with an extremely wide range of uses, characteristics, and cost. Incremental servo motors include both stepless, high-performance prime movers as well as the more familiar stepper type of incremental servos. The variety of these devices is so great that this study can only make reference to recent technical literature that explores their application trends (ref. 32). Another type of mechanical output device is illustrated by the stick shaker or vibrator, which provides an alerting or warning signal through the sense of touch. The stick shaker has been used in a number of military aircraft as the output device of a system warning of a potential pitch-up condition. There is an obvious application of these devices in any aircraft where a stall warning is necessary, and the necessary sensors (angle of attack, usually) are available.

Computer Display Technology

Aircraft display requirements and display types. -- The primary requirement for any cockpit display is that it must be legible under high and low levels of illumination as well as in darkness and to pilots of all age groups. Illumination levels in direct sunlight can be as high as 10^5 lm/m^2. Increased reliability and ruggedness beyond that typical of incandescent lamps and electromechanical devices is sorely needed.

It is also desirable for displays to have a graphic (line-drawing) capability and be able to image alphanumeric characters and symbols without a large number of wires connecting the display to the drive electronics. The most promising new technique for realizing these features appears to be a matrix-type electronically controlled display. Also, aircraft displays must

be compatible with modern integrated circuits and should operate on low voltage aircraft power. The display should also operate over a minimum temperature range of -20° to +70°C.

Displays may be classified into one of two types; passive displays, which function by modulating ambient illumination, and emissive or active displays, which generate light. The following paragraphs briefly describe promising examples of both types and give their general characteristics; more technical details are supplied in Appendices I through Q.

The introduction of on-board computing systems means that more sophisticated displays can be made available. For example, an on-board computing system can directly indicate through the display the estimated time of arrival at any particular point instead of requiring the pilot to read several dials and perform a calculation. Displays can also present navigational information, data on the mechanical state of the aircraft, and results of any automatic checks carried out by the calculator. It is quite possible that one or more larger displays may be used to perform these functions, reducing the cockpit area taken up by instrumentation.

Contemporary electromechanical instruments are bulky, require large panel areas, and are undesirably heavy. Also, they have relatively slow response, contain moving parts, and may be sensitive to vibration. The only other display technique in common use is the cathode ray tube (CRT); however, CRT's are rather heavy and bulky, require a high voltage, and carry a slight risk of implosion.

Because of cost restraints, it is not practical to develop new display techniques specifically for general aviation. Consequently, technology from other fields must be explored (e.g., military/commercial aviation for high technology and the automotive industry for high-volume, low-cost techniques). This section is devoted to a discussion of the potential application of current and projected display technology to general aviation.

Characteristics of passive displays. -- The advantage of reflective, passive displays is that their contrast is independent of the illumination level, i.e., they modulate the ambient light. This modulation is controlled on individual elements to form the desired image. They have low-power consumption and are ideal for large-area displays. A difficulty with passive displays is material saturation with increasing drive voltages. It is impossible for the passive display to compensate for the necessary short-address period by driving the display harder at higher voltages. To fill this need, some form of either inherent (in the form of high decay-time/rise-time ratio) or on-site memory must be employed.

Passive displays also require some illumination for night viewing and their field of view may be limited. An additional possible disadvantage in airborne application is a degradation of contrast in diffuse illumination, i.e., flying through a cloud, for those displays based on a scattering mechanism.

Characteristics of emissive displays. -- Light-emitting displays (LED) are less critically dependent upon the color, reflectively, and position of surrounding objects than reflective displays and are more easily read in low ambient lighting conditions. However, they must have sufficient luminance to compete with ambient illumination (up to 10^5 lm/m^2), must be visible when the eye is adapted to luminance levels up to 3.4×10^4 lm/sr/m^2, and must have adequate contrast with the immediate surrounding cockpit area.

Several techniques are available to reduce the amount of ambient light reaching the display. Bezels can be used where a limited field of view is acceptable and the space is available. More compact methods restrict the range of incident light on the display through the use of louvered plastic or imbedded mesh filters. Alternately, neutral density, colored, or circularly polarizing filters can be used. The colored filter matches peak spectral emittance of the display. All filters attenuate the incident ambient illumination twice while they attenuate the emitted display light once, resulting in improved contrast.

C-2

One important advantage of the emissive display is its nonsaturable output. The emitted light increases in proportion to the drive voltage up to a relatively high level. However, a means must be provided to reduce the display luminance, as emitted display light which has sufficient brightness under high illumination conditions will be too bright and therefore unreadable at low luminance levels. The disadvantage of the emissive display is the high power consumption that results from the low emission efficiencies.

Cathode ray tubes (CRT) are currently the dominant electronic displays except for pure alphanumerics. They are the standard of performance to which all other displays are compared. The CRT is a vacuum device with a cathode luminescent phosphor that is bombarded by a position-controlled electron beam. Operation is generally either video raster-scan or random access.

In the raster-scan mode, the beam traverses the whole area of the display line-by-line. Display brightness is determined by the beam power, phosphor efficiency, and duty cycle. The random-access mode is used with computer-generated nonvideo signals, typically alpha-numeric or graphic information, which is usually generated with a stroke writing technique.

The disadvantages of the CRT for general-aircraft application are the large ratio of tube-depth to viewing-face-diameter, high voltages required for adequate brightness, associated electronic circuitry, and bulk and power requirements (ref. 33).

The basic advantage of the CRT (ref. 34) is the high peak brightness of greater than three million cd/m^2, which permits high average brightness at TV display rates. The primary concern for television is the low duty cycle of typically 1/500 or 1/1000. In spite of this low duty cycle, there are displays (ref. 30) with luminance levels of 10,200 cd/m^2 with resolution in excess of 40 lines/mm. The CRT can achieve these luminance levels due to very high instantaneous luminance and their long decay time relative to the frame rate.

Another display requirement that is fulfilled by the CRT is the uniformity of both the threshold voltage and output beyond the threshold. Without uniformity there is undesirable mottling of the picture (ref. 35). The CRT meets the uniformity requirement as there are more than 1,000 phosphor particles within the electronic scanning beam; this averaging gives good uniformity of luminance across the face of the display. The redundancy of the display, i.e., multiple elements within the scanning beam, also gives a high degree of reliability.

For a material to be considered for a display it must not only have the capability of high peak brightness but also must have intrinsic memory or persistence. The CRT meets both these requirements. The CRT's major advantages are cost, which is 23 cents per character (Table 11), and the fact that further development is being funded.

TABLE 11. – DISPLAY SUBSYSTEM PROJECTED PRICES [a]

Number of characters	Gas plasma, $	Gas discharge, $	CRT, $	Liquid crystal, $	LED's, $
32	6.05	5.29	3.56		
256	3.75	1.09	1.10		
512	2.33	0.78	0.66	0.40	0.75
1024	1.69	0.62	0.35		
2048	0.97	0.56	0.23		

[a] Original equipment manufacturer price per character including decode, drive and refresh buffer.

A typical aircraft CRT display for an imaging forward-looking infrared sensor has a mean luminance of 171 cd/m^2, with a peak luminance of 680 cd/m^2, and requires 75 watts of power. The display is 11 x 8 cm and has an 800-line-per-picture height resolution. Envelope dimensions are 19x17x28 cm. Typical mean luminance levels for alpha-numeric CRT displays will

run as high as 1710 cd/m^2. The cost of a commercial TV display is $100 and a MIL spec unit will cost between $4000 and $5000.

CRT bulk leads to the desire for flat panel displays. Unfortunately, existing flat panel materials are not bright enough without increased persistence (inherent memory) or creating persistence by use of electronic on-site memory; consequently, this report defines the problems and requirements of matrix addressed displays with intrinsic or on-site memory.

Matrix address displays. — Today the potential for flat panel video technology is limited to liquid crystal, gas-discharge, and possibly electrophoretic devices. Little effort has been devoted to the problem of addressing or signal distribution in multi-element area displays as compared to the development of novel display materials. Currently, however, the problem is primarily that of getting the display message to all points on the screen; the display material itself is becoming a secondary problem.

It is necessary to distinguish between active matrix addressing and passive matrix addressing. Active matrix addressing is defined as containing gain-producing, switching, and/or memory elements at every display cell. A passive matrix, in contrast, is normally composed of two sets of parallel conductors oriented at right angles to each other, with the display medium sandwiched in between. The major problems associated with passive matrix driven displays are: 1) the excessive demands placed on the electrical and optical characteristics of the display material, and 2) the complexity and cost of the external drive electronics. All of these problems can be solved or bypassed by the development of active matrices.

The function of the active matrix is to integrate the electronic functions and to compensate for the deficiencies in the electro-optical characteristics of the chosen display material. Liquid crystal displays, for example, have slow response time while electroluminescent display materials require comparatively high voltages and lack inherent memory. Field effect liquid

crystal panels with their own capacitance and high impedence form their own memory; consequently, only a transistor is required at each display element to gate the initial charge (Figure 8). Electroluminescent displays, on the other hand, in either ac or dc types require a more complex circuitry due to high voltages and significant current levels. A comparison of the circuits in Figures 8 and 9 illustrate this point.

Most flat panel display schemes use a matrix arrangement in which the individual display elements are connected between row and column electrodes (ref. 35). Because all elements of the matrix are permanently interconnected, when a single element is addressed at the intersection of a row and a column, all other elements along the row and column are partially stimulated. Therefore, most elements will be partially excited (cross talk) many more times than directly excited. Each element must have a sharp threshold above the partially excited level to have adequate contrast. In display media not having this inherent sharp threshold, an additional diode is required.

Matrix address arrays are generally operated at a line at a time such that each element in the line can be driven for full line time rather than for just one element time. This provides an increase in excitation time by a factor of the number of elements in a horizontal line, increasing the output brightness of the display as much as 500 times.

The longer excitation time relieves the drive electronics of handling short, high-powered pulses to high-capacitance electrodes. However, line-at-a-time operation at TV rates requires a 15-kHz shift register for the horizontal rows, a video sampler-distributor, two storage registers, and a column driver (see Figures 10 and 11). The video signal for one line of information is fed into the sampler-distributor. Parallel video information is then stored in one register while the other storage register is feeding the column drivers. The storage registers and column drivers may be either analog or digital, but it is difficult to produce a large number of analog cir-

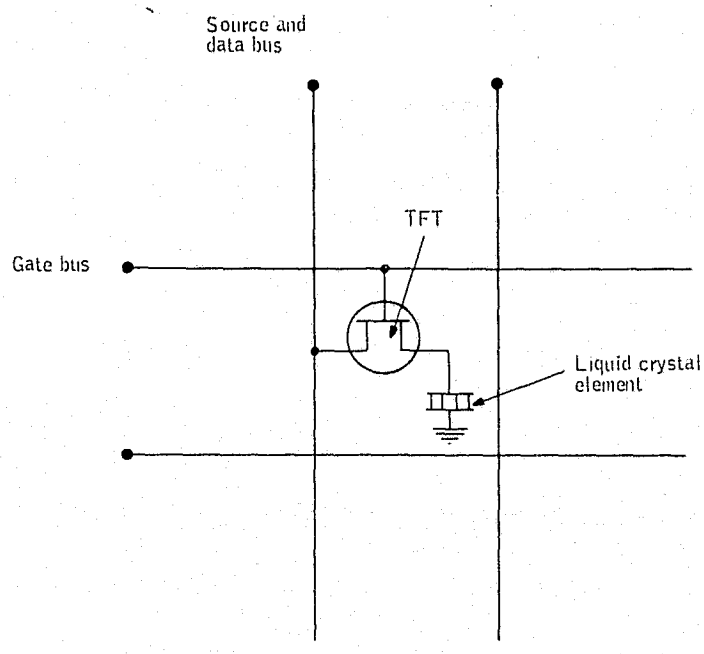


Figure 8. - Design of Elemental Matrix Circuit for the Large-Area Liquid Crystal Display

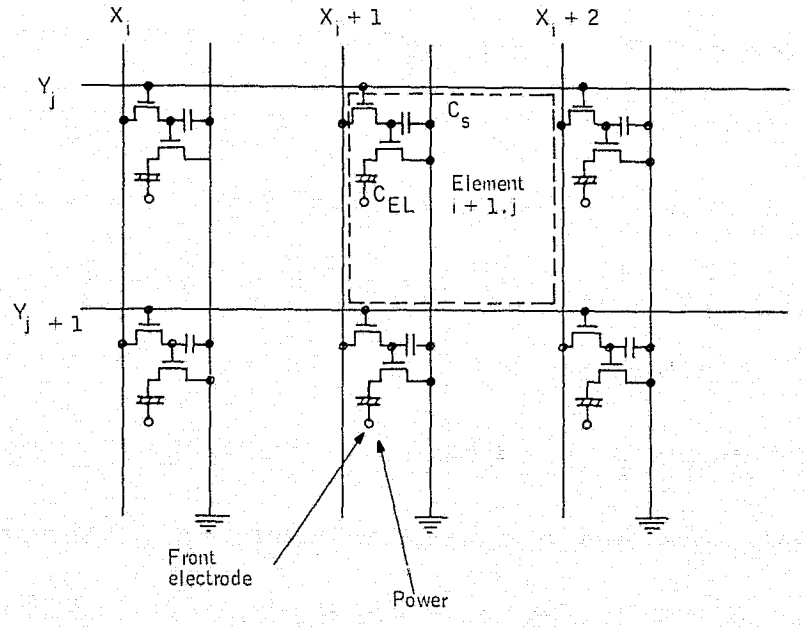


Figure 9. - Design of Elemental Matrix Circuit for the 6 by 6 inch, 20 Lines/Inch EL Display

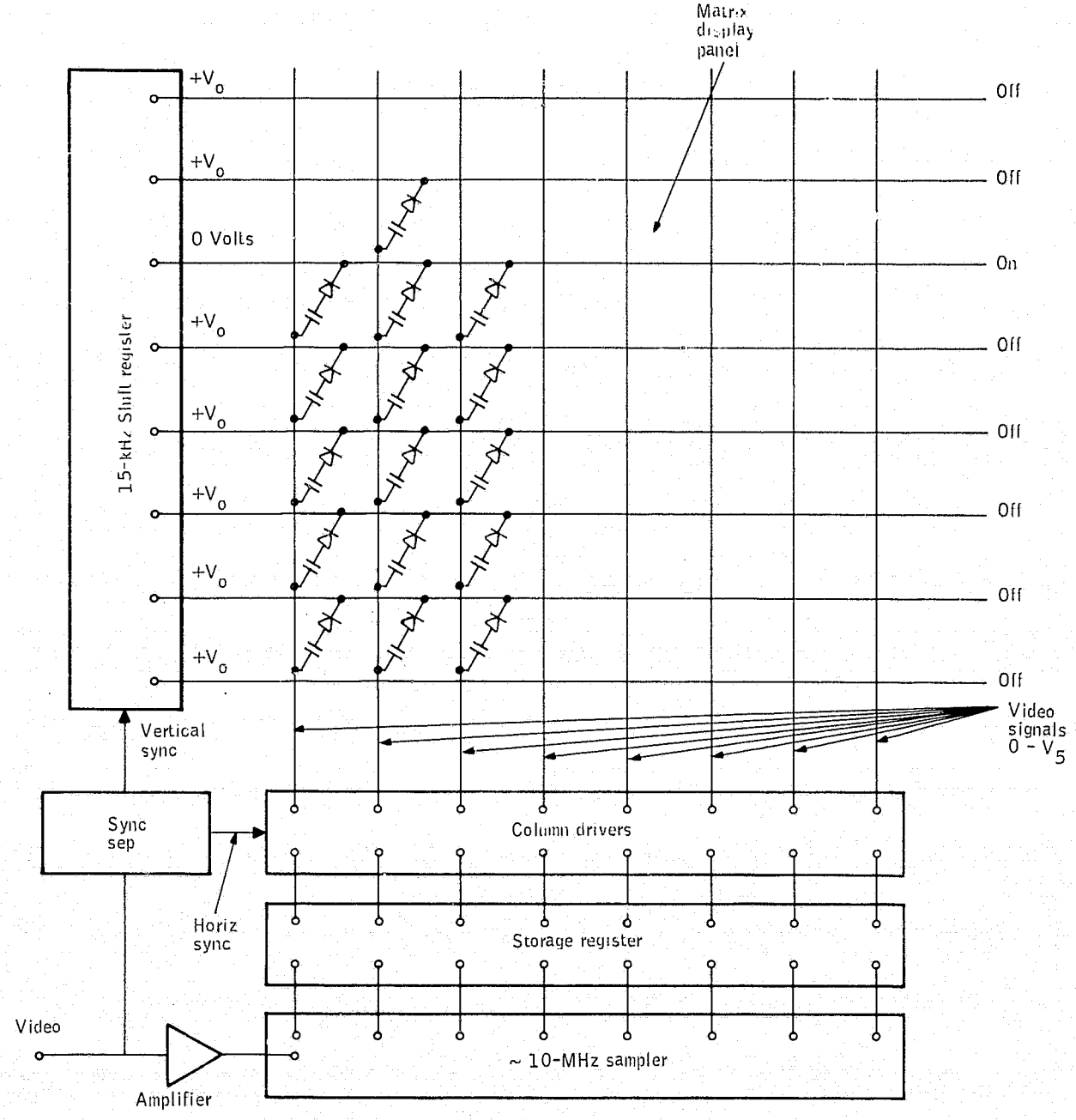


Figure 10. – Block Diagram of Matrix TV Display (Diode shown can be inherent feature of display elements ($V_S \geq V_O$)

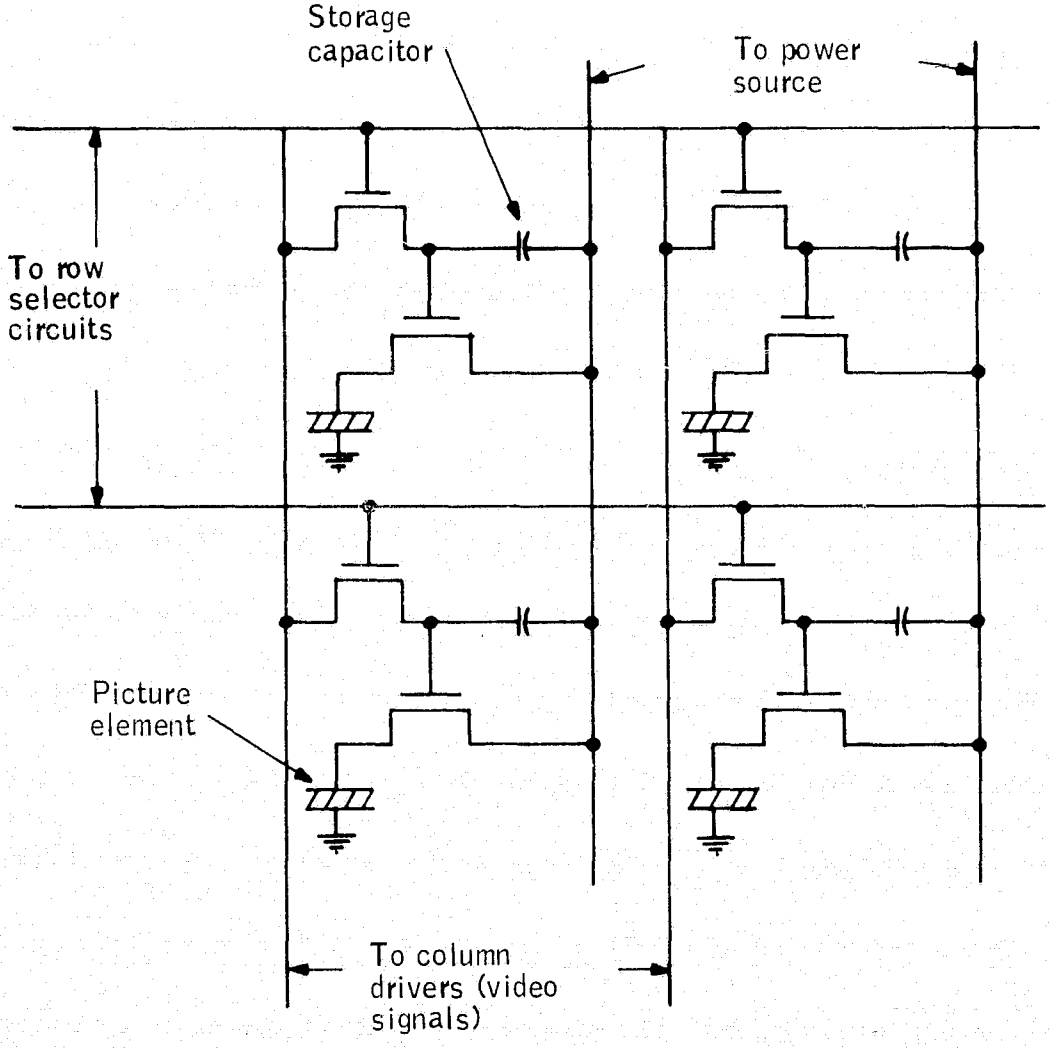


Figure 11. – Circuit Configuration Needed for Frame Storage in Matrix Display

cuits with the same characteristics, thereby avoiding vertical streaks in the picture. It is expected that digital circuits will be used to overcome this difficulty.

The average brightness of matrix displays developed to date is not as great as desired. Table 12 lists the expected maximum brightness for various display media when operated in a line-at-a-time format. This problem can be eliminated by incorporating an analog memory driver at each picture element location (Figure 11), which would maintain the display element excitation during the whole frame time and increase display brightness accordingly. Moreover, the memory elements must either control an external power source or act as amplifiers, which creates problems of element-to-element uniformity.

TABLE 12.- AVERAGE BRIGHTNESS AND LUMINOUS EFFICIENCY OF DISPLAY MEDIA

	Peak brightness, Cd/m^2	Maximum average brightness at 1/5000 duty factor, Cd/m^2	Efficiency, lm/w
Plasma	7500	10-15	0.5-1
AC EL	500	1-5	5-10
DC EL	3000-5000	25-50	0.5-1
LED	5000	5-10	0.5
Catho-luminescent phosphors	$\gtrsim$ 100 000	$\gtrsim$ 500	100

The need to avoid an unmanageable, large number of connections and to provide integrated electronic circuitry leads to the LSI concept of the entire

display. Monolithic integrated circuits of matrices over 2 to 3 inches may be beyond the present or near-future capability of silicon technologies (ref. 36). However, thin-film technology appears adequate for large-area active matrices and can be integrated with a large variety of display media. Thin-film transistor work is currently being conducted by Westinghouse and Aerojet General to build large active matrices.

It is not clear which technology - thin-film transistors (TFT's) or silicon processes - will eventually produce usable matrix displays. A one-inch square silicon matrix is huge by silicon IC standards; shorting problems, insulator pin holes, metalization, wafer breakage, in-process inspection and testing are the problems encountered with obtaining the needed crystalline perfection over large areas.

The key problem in thin-film technology is the need for 100 percent area yield in all process steps. This is different than the semiconductor circuit manufacture where many chips are formed simultaneously on a single wafer and 100 percent wafer yields are neither needed or expected. The thin-film deposition sequence is long and somewhat complex and subject to operator errors. Not only mask fabrication (for thin-film deposition) but also the alignment during the deposition steps must be perfect. However, mask location changing, deposition material sequencing, thickness monitoring can all be done by automatic equipment, and the process steps are quite fast, therefore, a large output per machine can be expected with reasonable yields as manual operations are eliminated.

Passive display material types. -- Liquid crystal, electrochromic, and electrophoretic, and PLZT are three examples of low-power passive displays. These displays do not emit visible radiation; rather, they control the passage of externally generated light through the display. Structurally, these three displays have similar principal operations but dissimilar properties.

Liquid crystal displays: Liquid crystal displays (LCD's) possess the normal solid and isotropic liquid phases of normal liquids, but have a third phase which occurs between the solid and the isotropic liquid phase. In this intermediate phase, the liquid crystal flows like a fluid but exhibits a crystalline organic state. The molecules are usually long and rod-like in shape and are responsible for the display-related anisotropic properties.

The liquid crystal is confined between two glass plates with their conductive coatings in contact with the material. The electro-optic phenomena can be divided into two groups: 1) those caused by dielectric forces and the so-called field effects, and 2) those induced by a combination of dielectric and conductive forces.

The most promising material and arrangement results in a cell structure called the twisted nematic. Linear polarized light propagating perpendicular to the cell is rotated approximately 90 degrees. Maximum light transmission is obtained by orienting a crossed polarizer and analyzer. The transmitted light decreases when the applied voltage exceeds a certain threshold voltage and the liquid crystal molecules start to change their orientation and the polarization of the display.

The advantages of the LCD's are low voltage and low current, which allows the cell to be driven by CMOS IC's and the fact that their contrast is independent of ambient light level. The disadvantages are slow response time (10-100 msec) and the difficulty of multiplexing. They have a limited temperature range for operation and the viewing angle affects the contrast.

Electrochromic displays (ECD's): An electrochromic material is one in which the color is changed by the application of an electric-field current. One version uses an aqueous solution of a special organic dye contained between a pair of transparent electrodes. This display relies upon the oxidation-reduction reaction of the dye controlled by the application of voltage to the electrodes. The dye is colorless in the oxidized state, but application of about

2 volts across the electrodes causes the reduction reaction to occur, and hence, a dramatic change in color as the dye is reduced. When the polarity of the voltage is reversed, the color compound oxidizes back to the colorless state. Another type of ECD uses a solid, inorganic film as the electrochromic material. A glass substrate with a transparent conducting layer or electrode is coated with a thin film of the material. A layer of insulating material is deposited over the film, followed by a second electrode. The film is colorless in the normal state, but when a voltage is applied, the display appears blue. When the polarity is reversed, the film again becomes colorless.

The advantages of this material are that it operates at 5 volts or less, and its appearance does not change with the viewing angle. The disadvantages are the slow response time (between 20 to 200 msec) and the difficult multiplexing.

Electrophoretic displays: This type of display uses pigment particles of one color that are suspended in a liquid of a different color. The suspension is sandwiched between a pair of electrodes, and at least one of them is transparent. The pigment particles are held in a colloidal suspension and carry a charge. When the electrodes are charged, the pigment particle moves toward the front electrode, where they collect to scatter the ambient light, making the display change color. Reversing the polarity causes the particled to move towards the other electrode, changing the color of the display to that of the suspension fluid.

The advantages of this display are that they operate at low voltages (less than 10 volts) and are legible over a wide range of viewing angles. The disadvantages are that the response time is slow (in 100's of msec) and that the particles may eventually settle out of suspension or may be damaged by shock and vibration.

PLZT: PLZT is a transparent ceramic electro-optical material (Pb, La, Zr, Ti), which exhibits light transmission properties that are voltage dependent. Because the material is in the solid state, it is necessary to use the

optical properties of induced birefringence and of scattering. The birefringent mode is implemented by operating the material such that the applied voltage will vary the polarization of the transmitted light. The effect can be observed through an analyzer sheet as a change in intensity. In the scattering mode, the transmitted light is diverted from the normal propagation direction and scattered into a larger solid angle by the application of the voltage to the material. Within these two modes of operation, the material can be classified according to its memory materials. The material may either have memory or no memories in both of the modes of operation.

The advantage of the material is that it has the potential for intrinsic memory, which would permit displays with higher contrast. However, the disadvantages are high drive voltage (40 volts), cracking under stress, poor contrast, small viewing angles, and a requirement for transverse exitation. This last requirement requires either depositing interdigitized electrodes or cross-slotting the material itself. Both techniques are difficult for matrix cell fabrication.

Emissive displays material types. -- Three examples of emissive displays are light emitting diode (LED), gas plasma devices (GPD), and electroluminescent devices (ELD). These devices emit visible radiation and consequently require more power than the passive display.

Light emitting diode: This device is essentially an electrical transistor. Basic components are the diode leads, crystal chip, and possibly a diffuser lens. Diode chip emitted light is proportional to the current flow. Luminous levels up to 340-1400 cd/m^2 are common without heat sinking and cooling. Voltage levels vary from 1.4 to 4.5 volts depending upon the material. Color of the output of the device is primarily red, but green and yellow are available and a blue output is under development. LED's are high-current, low-voltage devices that are compatible with discrete MOS transistors.

The advantages of LED's are that they can be molded into a large variety of shapes with a large range of optics. They are compatible with the low-voltage supply provided by conventional transistor circuits. They are rugged, have a high reliability with a long life, and are resistant to temperature change. The disadvantages are that the material is relatively expensive and the LED is not readily driven to maximum brilliance by MOS IC's, particularly CMOS IC's, due to the electrical current requirements. Consequently, they are not likely to be used in large matrix driven displays.

Gas-plasma devices (GPD): A plasma panel is a gas-filled device. It is the most advanced class of all flat panel displays. It has two modes of operation: ac and dc. The gas fill is usually neon or other noble gases mixed with other inert gases such as nitrogen. A panel resolution element is defined by a very small gas volume between orthogonally oriented sets of parallel conductors; one is on the front transparent surface, and the other is on the back transparent surface. The visible plasma discharge is observed when a high-voltage, low-current is applied between the electrodes. Some of the many display types are self scanning and require few electrical connections. These displays have been under more intensive study to develop real-time TV displays as they are closest to having all the required characteristics.

The advantages of GPD's are that they have sharp thresholds with a high contrast ratio and that gray scale is possible. The response time is fast enough for line-at-a-time addressing. The disadvantages are that they primarily come in red color and require 170 volts.

Electroluminescent (ELD): These devices are essentially an electrical luminous capacitor consisting of two flat electrodes, one of which is transparent with a space filled with an electroluminescent phosphor. When a low-current, high voltage is applied to the electrodes, a field is created causing the coated areas to emit a luminous cool light. Most displays are single colored, usually green or orange because of the higher efficiencies. The potential may be ac or dc depending upon the structure, but they usually operate in the ac mode.

The advantages of these devices are that they can be driven by TFT's for large-scale matrix displays. They are lightweight, have low current requirements, and produce no heat. They have high reliability and a long life and are not affected by temperature, vibration, or shock. Luminance levels are usually only 15 to 100 cd/m^2, and they come in only one color for a particular display type,but that may be red, green, yellow, blue, or white. The disadvantage is the requirement for high voltage (up to 600 volts).

Display material comparison. -- The basic parameters for comparison of candidate display materials are visibility in the high ambient background illumination of the cockpit, luminance, contrast, life, power requirements, and resolution. A comparison based upon the size and bulk is not made as the materials are all intended for flat panel displays, and it is assumed that panel depth will not be a problem as drive electronics can probably be miniaturized with IC. If the bulk of the electronics becomes a problem for high-voltage displays, it can be mounted in a remote location. Thickness of the panels is expected to be less than one inch.

Table 13 presents display material characteristics, and a detailed discussion of these parameters is presented in Appendices H through P. The pacing requirements for a display material are contrast and luminance. A specific case for these requirements is presented in Table 14. A discussion of these requirements and the deviation from them is presented in Appendix P. Because data on reflectance is not available in the literature, five percent reflectance is assumed for all materials. A display or time-averaged contrast of 10:1 is also assumed as a requirement. The intrinsic material contrast and luminance is equal to the time-averaged contrast and luminance only when the display elements are on continuously (nonmultiplex or memory matrix addressed). However, the contrast luminance is reduced when a character is multiplexed or a matrix is addressed one line at a time without memory.

TABLE 13. - DISPLAY MATERIAL CHARACTERISTICS

Material and type of display	Intrinsic contrast ratios	Present resolution, line/CM	Ultimate resolution, limit, lines/cm	Response times	Voltage and current	Compatible electronics	Temperature range, °C
LCD (Passive)	15:1 to 50:1 possible	40	100	10-20 msec rise time; and 100-300 msec decay time	5V 1 $\mu W/cm^2$	CMOS IC	0 to +70
PLZT (Passive)	100:1 at 90 V	Not reported	Not determined	1 to 100 μsec	30 to 50 V μA	Discrete transistor	Not determined
LED (Emissive)	50:1	20	40	100 nsec	1.4 to 4.5 V 1.5 A/cm^2	Bipolar transistors to interface with MOS drivers	Wide
GPD (Emissive)	50:1 (time averaged)	20	40	1 to 10 μsec	170 V 0.16 A/cm^2	MOS to swing lower voltage of ±25 V on 135 V bias	Wide
EPID (Passive)	20:1 to 40:1	Unknown	Unknown	50 to 100 msec rise time and 100 msec decay time at 50 V	10 V to 30 V Low current	Discrete transistors or PMOS IC's	-15 to +50
ELD (Emissive)	>50:1 (Time averaged)	8	40		200 to 600 V (5 kHz ac) 0.03 A/cm^2	TFT matrix and high-voltage TFT's	Wide
ECD (Passive)	Limited contrast	Unknown	Unknown	100 to 500 msec	< 5 V High current	Bipolar transistor more suitable than MOS	-20 to +70

TABLE 13. – DISPLAY MATERIAL CHARACTERISTICS (Concluded)

Viewing angle and contrast sensitivity	Life and reliability	Color and luminance, cd/m^2	Intrinsic memory	On-site memory	Comments
15° to 20° Angle sensitive	50,000 hr	Not applicable	No	TFT and silicon chips	• Matrix displays under development • Require heaters for temperature control • Ample development funds available from computer and watch industry for multiplex displays
Poor contrast beyond 15[a]	Life not determined Not rugged	Not applicable	Yes	TFT or ferroceramic	• No high-volume user • Seven-segment numerical display available in birefringent mode
Lenses to increase luminance decrease view angle	10^6 hr High reliability and rugged	Red, green, yellow, orange 1000 peak	No	Silicon transistor	• Limited to small displays • Power limits display brightness • Red is poor color for aircraft displays
Wide	> 20,000 hr Rugged and reliable	Red, orange, blue, green 150 (time averaged)	Potential for memory	Yes	• Most advanced flat panel technology • Supported by TV industry • Sharp threshold for good contrast and gray scale being developed
Very wide Not sensitive	10^6 to 10^7 cycles in 3000 hr	Not applicable Not reliable	Yes	TFT	• No application yet
Very wide Not sensitive	20,000 hrs Rugged	Red, green, yellow, blue, white 140 to 340 (time averaged)	Phosphor decay	TFT	• Thin, lightweight, flexible • Matrix display under development
Very wide Not sensitive	10^{+6} to 10^{+7} cycles Unknown	Blue, green Not applicable	Yes with open circuit voltage	Unknown	• Matrix development unlikely • New technology

TABLE 14. — FLAT PANEL DISPLAY MATERIAL INTRINSIC CONTRAST AND LUMINANCE REQUIREMENTS

	Reflective passive displays (a)				Active displays (a)							
	Nonmultiplex or on-site memory matrix address	Multiplex or nonmemory line-at-a-time matrix address, 30 frames/sec			Nonmultiplex or on-site memory matrix address		Multiplex or nonmemory line-at-a-time matrix address, 30 frames/sec					
		16 line	100 line	500 line	Luminance (B) cd/m^2	Contrast	16 line		100 line		500 line	
	Contrast	Contrast	Contrast	Contrast			Luminance, cd/m^2	Contrast	Luminance, cd/m^2	Contrast	Luminance, cd/m^2	Contrast
Luminance and contrast	C	$\frac{CL}{N}$	$\frac{CL}{N}$	$\frac{CL}{N}$	2/100 to 5/100 of	C	$\frac{BL}{N}$	$\frac{CL}{N}$	$\frac{BL}{N}$	$\frac{CL}{N}$	$\frac{BL}{N}$	$\frac{CL}{N}$
White cloud eye adapted display shaded from ambient illumination	10:1	53:1	340:1	1700:1	680 to 1700	10:1	3200 to 9100	53:1	2×10^4 to 5.7×10^4	340:1	10^5 to 2.8×10^5	1700:1
Luminance and contrast	C	$\frac{CL}{N}$	$\frac{CL}{N}$	$\frac{CL}{N}$	$\frac{RE(1-C)}{\pi(\frac{C}{C_R}-1)}$	$>10C$	$\frac{RE(\frac{N}{L}-C)}{\pi(\frac{C}{C_R}-\frac{N}{L})}$	$>10\frac{CL}{N}$	$\frac{RE(\frac{N}{L}-C)}{\pi(\frac{C}{C_R}-\frac{N}{L})}$	$>10\frac{CL}{N}$	$\frac{RE(\frac{N}{L}-C)}{\pi(\frac{C}{C_R}-\frac{N}{L})}$	$>10\frac{CL}{N}$
Ambient sky illumination of display	10:1	53:1	340:1	1700:1	4,460	100:1	2.6×10^4	530:1	1.6×10^5	3400:1	8.2×10^5	17 000:1
Direct sun illumination of display	10:1	53:1	340:1	1700:1	16,000	100:1	9.2×10^4	530:1	6×10^5	3400:1	3×10^6	17 000:1

Key

Symbol	Definition
C	Desired time-averaged contrast ratio, 10:1
C_R	Material intrinsic contrast
L	Number of lines or characters to be sequentially addressed or multiplexed during refresh period
N	Number of refresh periods in eye integration time of 0.1 sec, 3
B	Desired time-averaged luminance of display
R	Display material reflectivity

[a] Contrast ratios and luminance are peak or intrinsic required levels for display material

It should be noted that passive display luminance is a function of reflected ambient illumination. It is assumed that the cover glass reflectance is lower than the display reflectance and, therefore, display contrast remains constant with illumination.

Table 14 presents the contrast and luminance requirements for three conditions of high ambient illumination in the aircraft cockpit. The first condition is when the display is shielded from all ambient illumination, and the required display luminance is established by eye adaption to bright white clouds. The second condition is when the display is shielded from direct sunlight but is illuminated by reflected light from white clouds through the cockpit windows. The last condition of direct sun illumination is the most difficult for active display visibility. This table does not include the use of ambient light suppression techniques for active displays because specific display reflectance data is not available. Therefore, Table 14 presents a worst-case set of requirements and provides a basis for relative comparison of display materials.

Table 13 shows that all passive display materials have adequate intrinsic contract (10:1) for nonmultiplex or memory matrix address displays. All display materials except electrochromic (ECD) and perhaps electrophoretic (EPID) have enough intrinsic contrast (53:1) for use as 16-character line-multiplexed or nonmemory (line-at-a-time address) matrix display. However, there are no materials that have adequate intrinsic contrast when a high resolution (100 to 500 character/line) nonmemory display is required. PLZT may have a future potential of 1000:1 contrast, but it requires a high drive voltage. On-site memory is a better approach to the development of high-resolution passive displays as any material with lower intrinsic contrast but superior electrical characteristics can be used.

Intrinsic contrast requirements for active displays become much higher than for passive displays when the active display is in ambient illumination (Appendix Q). These ratios range from 530:1 to 17,000:1, and none of the materials meet these requirements. On-site nonlinear devices could be used to improve the material excitation threshold (Appendix Q) and contrast; however, on-site memory can be provided as easily and then high contrast would no longer be required. It should be noted that intrinsic luminance requirements for active displays in ambient illumination (Table 14) becomes much larger than the capabilities of the display materials. Even two orders of magnitude reduction in luminance requirements (Table 14) for high resolution displays through the use of ambient light suppression techniques will not help. Consequently, on-site memory is definitely required for both passive and active high-resolution matrix displays.

SECTION 6
PROJECTED LARGE-VOLUME USERS

The technology used in future general-aviation avionics systems will not be defined and developed primarily through the efforts of general aviation; rather, these advanced avionics systems will largely be based upon exploitation of technology developed for other high-volume users. The trend is toward development of technology with the largest user base. It is, therefore, the purpose of this section to define the industries and/or applications that are the driving forces behind this technology and the direction in which those forces are operating.

MICROPROCESSORS, I/O's, AND TRANSDUCERS

Automobile Industry

The automobile industry is potentially one of the largest volume users of microprocessors. The world-wide annual production of passenger cars, trucks, and buses is measured in the tens of millions of units. If several microprocessors are used in each unit, the production volume requirements for microprocessors could exceed one hundred million units with a value in excess of a billion dollars.

The U. S. automotive industry is facing increasingly stringent federal regulations for emissions, fuel economy, and safety. Emission control and fuel economy tend to work against each other as well as against overall engine performance (power, starting ease, smoothness). Industry leaders feel that electronics is the only viable near-term solution to achieving the precise balance required in all these factors. They are particularly interested in microprocessors because they offer the computational power, flexibility, and potential low cost necessary to the following applications:

- Engine control (ignition timing, fuel management)
- Drive train control (transmission shift point, speed control)
- Safety systems (anti-skid braking, collision warning)

Other areas that are being investigated include diagnostic systems (both on the vehicle and at service centers), display management, passenger comfort, and entertainment control. Solid-state electronic systems have already been used in several of these areas (electronic ignitions, fuel injection systems, speed control), but it is hoped that microprocessors can provide increased capability at lower cost.

The main areas of concern to automobile manufacturers are the cost of rugged, high-reliability microprocessors and the availability of good, low-cost transducers. There seems to be no question that currently available microprocessors can meet the speed and performance requirements of these applications; however, there is doubt that the microprocessors, transducers, and all their associated electronics can meet the tough environmental problems found in automobiles, i.e., temperature, humidity, vibration, and dirt. The requirements are stringent; the manufacturers require failure rates less than 1 percent per 50,000 miles because of legislative maintenance liabilities. In terms of electronics, this translates to about 700,000 hours mean time between failures in a 25-component system. The costs of production and screening for these requirements makes the currently available microprocessors and transducers too expensive for general application. Automobile manufacturers are confident, however, that cost and availability problems will be solved as a result of large-volume demand. They need a microprocessor under $10 and a transducer at about $1.50.

The first microprocessors in automobiles may be introduced as early as the 1977 model year. Ford has indicated the possibility of including one for ignition timing in its Mark IV line. Each of the major manufacturers will probably follow by introducing microprocessors in their highest price model lines during the 1977-79 model years. If prices decline enough, the general

spread of microprocessors to all model lines should occur by 1980. By 1982, several microprocessors tied together in a multiprocessor network will probably appear.

In summary, the automobile industry has the need and the large-volume demand necessary to be a major influence in the microprocessor market.

Computer Industry

The general computer industry will, of course, be a major user of microprocessors. Application areas include nearly all phases of the industry, including:

- Peripheral controllers
- Data Communications
- Multiprocessor configurations for mainframes

There is an increasing trend toward replacing the special-purpose, hard-wired peripheral device controllers with microprocessors. The peripherals include card readers, "floppy" discs, cassette tape units, line printers, and CRT display terminals. The use of microprocessors dramatically reduces the total parts count while lowering cost and performing the required function as well as or better than the dedicated special-purpose hardware. This area is already a large and rapidly expanding market for microprocessors. Bit slice microprocessor chips in particular have been designed into a number of such applications since early in 1975.

Data communications is one of the most promising application areas within the computer industry. The typical uses of microprocessors in this field include test equipment, preprocessors, intelligent terminals, data concentrators, message switching, and communication instrument controllers.

The availability, low cost, and functional capabilities of microprocessors has generated interest in multiple microprocessor networks as an alternative architecture to large mainframe CPU's. This approach can increase throughput, reduce resource overhead, and increase fault tolerance in large-scale computer facilities. Maintenance costs and repair time can also be greatly reduced through the reduced parts inventory requirements of this approach. The multiple microprocessor network will affect the entire computer industry from minicomputers to large mainframe CPU's.

Consumer/Commercial Products

Microprocessors are having a tremendous impact on the consumer and commercial products industry. They are being used both to enhance the performance of present-day equipment and in the development of completely new products for offices, stores, households, and entertainment centers. The personal electronic calculator, whose components were the forerunners of today's microprocessors, is an example of how quickly a new product can attain international prominence.

Point-of-sale terminals and supermarket checkouts are already using large volumes of microprocessors. Other areas that are currently using microprocessors include investment houses and the finance industry, banking, business inventory control, and electronic test instruments.

Microprocessor technology is expected to cause the low-cost household computer to emerge as a significant product within the next few years. Hobby kits based on currently available microprocessors have been on the market for about two years. Although the early kits were relatively expensive for the general hobby market, they immediately sparked tremendous enthusiasm. This enthusiasm will become more widespread as prices drop and the variety of household uses becomes known. The home TV set connected to a microprocessor will become an interactive information and

entertainment center as opposed to the passive role it now plays. The home computer will become as widespread as the ubiquitous TV set by the mid 1980's. This will cause an annual demand for microprocessors in the millions of units.

The microprocessor has begun to show up in household appliances as a replacement for electromechanical controllers. This particular application does not require a great deal of speed or accuracy, but will still provide a healthy market for low-end microprocessors. Typical applications include control systems for dishwashers, ranges, clothes washers, microwave ovens, and sewing machines.

In the entertainment segment of the commercial market, the old electromechanical pinball machine is rapidly giving way to a variety of sophisticated games using TV screens and microprocessors. The new electronic games not only provide greater interest and complexity, but sharply reduce the high maintenance costs associated with their electromechanical counterparts.

Highway Departments

Federal and State Highway Departments are faced with ever increasing problems of traffic safety and congestion. In their attempts to solve these problems, they are using automated control systems for intersections and super-highway entry ramps. Until recently, they were forced to choose between minicomputers and special-purpose, hard-wired controllers for these systems. Microprocessors offer significant advantages over both. Compared to minicomputers, microprocessors are a better match in terms of computational power, cost, and size; when compared to hard-wired random logic controllers, they offer much greater flexibility, reduced parts count (at least 60 percent), and reduced power requirements, size and weight.

There are several companies already active in the traffic control business. Multisonics Corporation of San Ramon, California, uses the Intel 8008 as the CPU in its Model 901 intersection control system. Comstar Corporation teamed with TRW Systems, Houston, to provide 1000 microprocessors for traffic control in the city of Baltimore.

With the current and expected future problems of highway safety and traffic congestion, microprocessors will most certainly be used in ever increasing volume. This application may prove to be one of the largest volume users of microprocessors in the next 5 to 10 years.

Military

The Military has shown a rapidly growing interest in designing microprocessors into their defense systems. This interest has generated intense competition from semiconductor firms through defense systems contractors to supply military needs. Estimates of the near-term potential military market for microprocessor devices are in the 8-to-10 million dollars per year range.

The military's interest is sparked by the promise of higher reliability, greater hardware and software flexibility, smaller size, and lower power drain. The Pentagon also sees the microprocessor as a means of lowering defense systems' initial and life-cycle costs. They are constantly pushing for lower maintenance and logistics costs and see the microprocessor as a possible answer. The microprocessor can make it far easier to design built-in-test and diagnostic equipment into their defense systems to troubleshoot potential problems before they arise.

There are several current military projects that involve microprocessors. They include jet engine monitoring and control systems, communications control systems, airborne and manpack navigation equipment, fault detection and diagnostic systems, and torpedo control systems. The Navy's Interim Standard Avionics Digital Computer, which is slated to be used in

all future naval aircraft, is one of the first major military projects to specify microprocessors. Three basic designs are expected: a standard avionics computer, a stand-alone I/O processor, and a combination computer-I/O processor.

The most significant problem areas as far as the military is concerned are finding microprocessors that can meet stringent military specification environmental and speed requirements and cost. Most commercial microprocessors cannot meet these requirements; therefore, many military systems houses are trying to develop their own microprocessors. One concept that is being examined is that of a "standard microprocessor building block." This approach would help to alleviate the problems of small production runs of specialized processors for particular applications.

It is almost certain that by 1980 several military-qualified microprocessors will be in production and available for use in general-aviation avionics. It is also probable that the cost of these units, even though greater than that for commercial-grade microprocessors, will be compatible with general-avionics economics.

Industrial Automation

There are a myriad of applications for microprocessors in industrial equipment and factory controls. They are already being used in factory automation systems, numerical control, data acquisition systems, and in industrial robots. They offer flexible, low-cost computational power that can meet the rugged environmental problems of factories.

The situation within the process control industry can be termed "explosive". Within 1 to 2 years every producer of industrial control equipment will be required to include microprocessors to remain competitive. The availability of microprocessors may encourage newcomers in the industry and will certainly lead to new and more diverse applications.

Current industry surveys indicate a market of about a thousand microcomputer units per year at an average price of one thousand dollars. This is expected to increase to nearly thirty thousand per year by 1984 at an average unit price of $300 for the system.

One area that will develop rapidly as a result of microprocessor technology is industrial robots or programmable manipulators. This application also depends upon other technologies including sensors, actuators, and the development of sophisticated algorithms for pattern recognition and goal-directed behavior by machines. When breakthroughs occur in these technologies, especially in the area of algorithm development, the demand for microprocessors will increase dramatically in this application. Although quantitative estimates of this demand are difficult now, it seams feasible that it will reach the tens of thousands of units by the mid 1980's.

PROJECTED LARGE-VOLUME USERS FOR DISPLAY MATERIALS

The rapidly growing digital timepiece and calculator markets will boost production of the newer displays such as GPD's, LCD's, and LED's. Low cost is an absolute requirement for the consumer market in general, and low-power consumption is a vital requirement for wrist watch and portable calculators. Liquid crystals, which have lower power consumption, will be priced lower than LED's due to batch processing. The GPD requires a higher voltage than LCD and is not a candidate for this high-volume market.

The largest potential market for flat panel displays is the consumer TV. Gas-discharge displays are leading the race for this application, but are a long way from adequate performance at the present time. Almost all flat-panel research is based on the use of matrix array of display devices, which requires the sharp and uniform threshold of gas-discharge devices.

It is expected that there will eventually be a high-volume consumer TV market for flat-plate GPD's as the result of current extensive research and development (Appendix N).

SECTION 7
COMPUTER TECHNOLOGY PROJECTION

The technologies that are of interest based on general-aviation requirements and large-volume users have been covered in the previous sections. This section will address the future of those technologies selected as most significant for future general-aviation avionics advancement in implementation technology, subsystem technology, system technology, and operator/computer interface technology.

SEMICONDUCTOR TECHNOLOGY PROJECTION

Progress in the circuit design and packaging of various technology areas will continue at the current rate at least. Prediction of the specific circuit technology that will be dominant in the 1980's is not within the scope of this study; however, the dynamic advances occurring in the various areas show no signs of approaching a limit. Appendix A discusses probable trends.

COMPONENT AND MODULE TECHNOLOGY PROJECTION

Circuit/systems designers can expect even a wider variety of component and subsystem modules to choose from in the 1980's. For example, the microprocessor offers the designer of inexpensive systems the computational power of a computer functioning as a component. Processor, memory, and IO module technology projections are the major areas covered in the following subsections.

Processor Technology Projections

Large scale integrated circuit (LSIC) technology made the microprocessor possible. Further developments in circuit technology and architecture will product the following general trends in processor technology:

- Greater functional capabilities
- Greater speed
- Lower power requirements
- Increased standardization
- Increased reliability and ruggedness
- Increased flexibility
- Lower costs

The functional capabilities of a microprocessor are measured by the power of its instruction set, ease of addressing, and by its data/instruction word length. Today's microprocessors have relatively simple instruction sets when compared to today's minicomputers. A typical microprocessor has 20 to 60 instructions while many minicomputers have 150 to 200. The short word lengths of today's microprocessors (usually 4 or 8 bits) result in cumbersome addressing when compared to minicomputers. This has already begun to change with the appearance of 16-bit microprocessors with a variety of addressing modes. The application of bit slice microprocessor chips also provides great flexibility. It is anticipated that by the 1980's microprocessors will achieve or exceed the functional capabilities of today's minicomputers. Features such as hardware multiply and divide will be generally available, and some microprocessors will also include floating point hardware and trigonometric functions. Microprogramming techniques may also be applied to provide powerful instruction sets for particular applications. This does not imply that the 4- and 8-bit microprocessors will fade away; for many applications they are perfectly adequate and will definitely see large-volume use.

There are many factors affecting the overall speed of a processor. These include memory access time, instruction execution time, and the power of the instruction set. The most meaningful comparisons of processor speed are made with benchmark tests where the same general function is performed by several processors. Based on this type of comparison, today's microprocessors range from about an order of magnitude slower to the equivalent of today's minicomputers. It is expected that by the 1980's most microprocessors will achieve instruction execution times in the 0.2- to 2-μsec range, which is equivalent to today's minicomputers. There will be a spectrum of available speeds, depending upon the circuit technology used. TTL and ECL will be used in high-performance microprocessors, while the various MOS technologies will be used in low-performance microprocessors.

Microprocessor power consumption is a function of circuit technology and word length. Today's bipolar microprocessors dissipate about 3 to 5 watts, single-polarity MOS microprocessors about 0.4 watt, and CMOS microprocessors about 0.03 watt. These power dissipation levels will gradually decrease over the next few years, and by the 1980's they will be reduced by about a factor of two for a given technology and word length. I^2L technology will see increased use for microprocessors and will produce a range of power consumption as a function of speed.

There is currently very little standardization among microprocessors. This is primarily due to the immaturity of the industry, the explosive growth as a result of the demand, and the lack of a clearly dominant product. Second sourcing of microprocessors and associated components is becoming more common. By the 1980's, the industry will mature, and a few superior products will emerge. Standardization will be necessary to keep the industry growing and will occur in the following vital areas.

- Instruction Set - Manufacturers will develop upward-compatible instruction sets within their own products. Many of these will emulate popular minicomputer instruction sets to take advantage of large software-support bases.

- Higher-Order Language - A few higher-order languages (HOL's) will emerge as industry standards for broad application areas. They may be based on today's FORTRAN, BASIC, COBOL, or PL/1.

- Voltage Levels - There is a wide variation in the required supply voltage levels for today's microprocessors. This will be standardized by circuit technology, with those technologies requiring the fewest levels (such as TTL or I^2L) emerging as leaders.

- Interface Requirements - Interface standardization is urgently needed, but will come slowly. Complete standardization is not expected by the early 1980's; there will, however, be advances towards universal I/O modules for common peripheral devices.

- Packaging - Current microprocessors come in 16, 18, 24, 40, 42, and 48 pin dual-in-line packages. The 40-pin package is the most popular at the present time, but cannot be considered a standard. It is expected that packaging standards will follow standardization in other areas.

Microprocessors are inherently more reliable and rugged than today's minicomputers. This is a direct result of LSIC technology and reduced parts count. One area that does require improvement is the operating temperature range. Most of today's microprocessors will only operate within the range of 0 to 70°C. This range is too narrow for most military and automotive applications. The large-volume demand in these applications will result in microprocessors that meet extended temperature range requirements (-25 to +100°C) in the near future. Full military temperature range (-55 to +125°C) will be available in the early 1980's.

As the size, cost, and performance levels of microprocessors improve, their flexibility will be enhanced. These trends will lead to increased

microprocessor modularity, which is their greatest asset. Microprocessors will form modular system building blocks that can be assembled into complex computing systems. Microprocessor manufacturers will design fully compatible lines of processors with a wide range of capabilities. Modularity will allow low-performance systems to be easily upgraded as requirements change. The problems of standardization, however, must be overcome if microprocessors are to become truly universal computing elements.

The total cost and cost distribution in computer systems has been changing rapidly in recent years. In the early 1970's, minicomputer systems' costs were in the $10 to 20K range with the CPU cost as a major component of that total cost. Current system prices are in the $1 to 5K range for computers based on LSI techniques, with memory taking over as the major cost component. Microprocessors will accentuate this trend until limiting factors appear. As processor prices become insignificant compared to memory, I/O, and peripheral devices, the downward pressure on processor prices will lessen and will shift to the other components of a complete computer system. There will be a tendency to maintain processor price levels while increasing performance and reliability.

Figure 12 shows the history and projection of the price/performance ratio for single-card processors. Included are processors using MSIC and LSIC, as well as current and future 16-bit monolithic microprocessors.

The price/performance ratio was determined for each device as the advertised single-unit quantity price, divided by the number of bits, and multiplied by the device throughput when operating with a flight control instruction mix.

The family of Honeywell processors in this projection is intended for military applications and, consequently, is designed to more stringent environmental requirements than the commercial processors for which data points are included. The figure is not intended to present a comparison of specific proces-

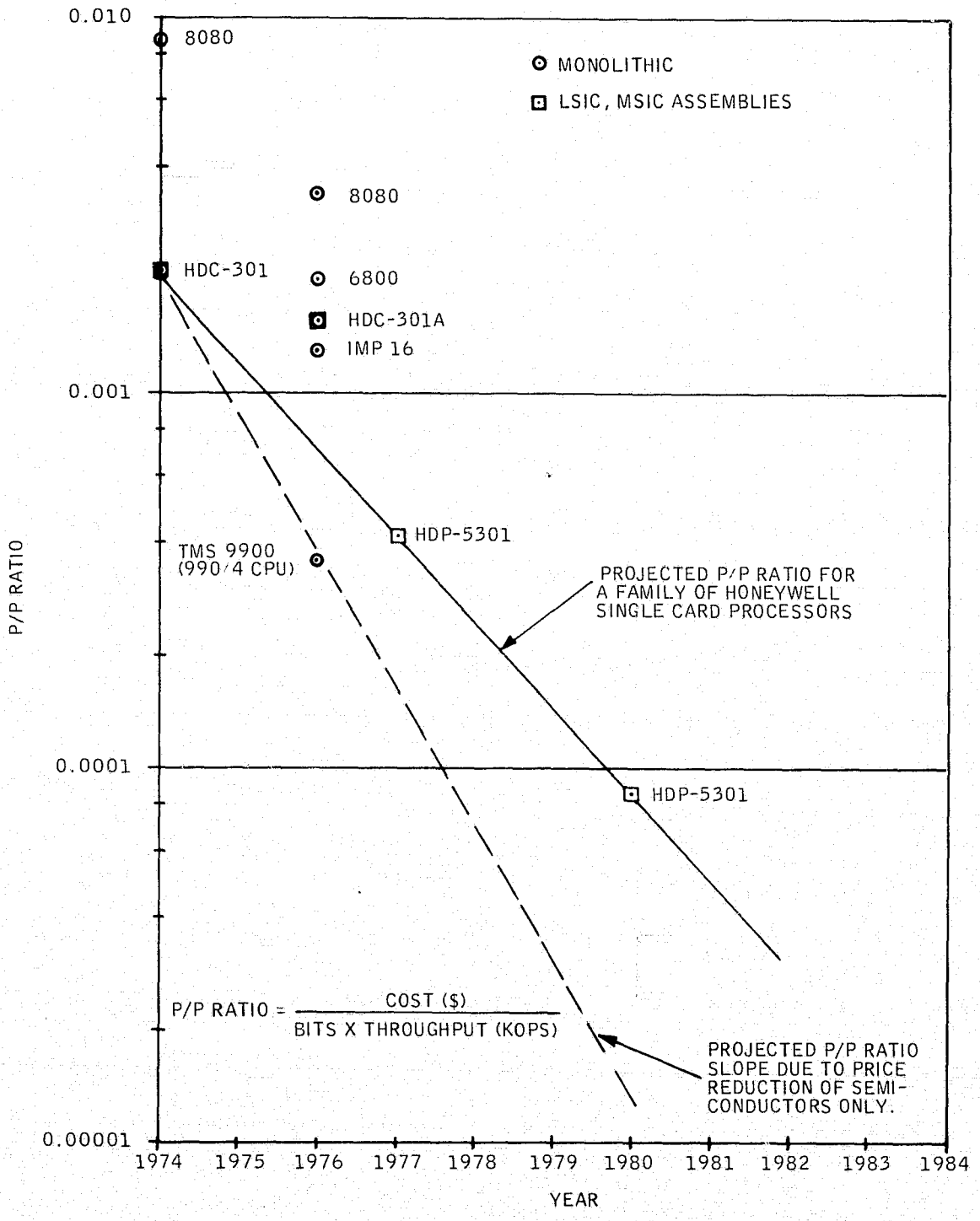


Figure 12. – Projected Price/Performance Ratio for a Family of Honeywell Processors Compared with Past and Present Devices

sors, but only to indicate trends. The dotted line on the figure indicates the trend due to the semiconductor price reductions projected. The total cost of a processor includes PC board,connectors, and other components, as well as assembly labor, which may be increasing rather than falling with the semiconductor prices.

The factors involved in lowering microprocessor prices include the usual pressures of competition and the economics of large-volume production. There are other significant factors that specifically apply to the microprocessor industry. Testing is a major contributor to the total cost of a microprocessor. It is expected that new automated testing techniques will be developed to speed this process, and that the testing devices themselves will be based on microcomputer systems. Assembly labor will decline as LSI techniques give way to very large-scale integration, and computer-aided design techniques will be used in the initial phases of microprocessor development. These techniques will lead to a level of automation where the design, production, and testing of microprocessor will be controlled by microcomputer systems with very little human intervention.

As a result of these factors, an 8-bit CPU should cost less than $20 by the mid 1980's. In commercial grade and large quantities, a 16-bit CPU will cost less than $50, and a complete 16-bit microcomputer system will cost less than $200. Particular microprocessors that become very popular and are used in a variety of applications in large volume may well reach the $1 to 5 range.

Table 15 shows the estimated volume and value of the microcomputer market for the years 1978 and 1982 with a breakdown by word length, execution speed, and application areas.

TABLE 15.— U.S. MICROCOMPUTER MARKET

Microcomputer breakdown	Value, million $		Quantity, million	
	1978	1982	1978	1982
	486.0	1082.0	2.40	5.92
By word length, bits				
≤ 4	120.0	208.0	1.76	3.86
8	185.0	364.0	0.48	1.46
12	11.7	20.4	0.02	0.05
≥ 16	169.0	489.0	0.14	0.56
By execution speed, μsec				
< 0.1	0.6	6.2	<0.01	<0.01
0.1-1.0	90.7	205.7	0.06	0.18
1-10	132.2	347.0	0.15	0.63
10-100	120.5	243.0	0.51	1.63
> 100	142.0	280.0	1.68	3.49
By application				
Business/education	108.0	222.0	0.98	2.10
Communication	71.2	131.0	0.07	0.21
Consumer/auto	27.2	60.2	0.84	2.08
Computer	148.0	374.0	0.37	1.11
Government/military	47.1	124.0	0.02	0.05
Industrial	69.5	120.0	0.07	0.16
Instrumentation/medical	14.2	50.8	0.04	0.21

(Numbers are rounded off)

Memory Technology Projections

This subsection describes the characteristics of memory technologies expected in the 1980 time frame with applicability to avionics.

Bipolar semiconductor. — It is expected that a 4,096-bit bipolar RAM with access times in the 10 nsec region will be available by 1980. An I^2L 4K-bit memory array with 100 nsec access time is expected to be available in sample quantities by 1976. The present power dissipation of 0.5 mW/bit will most likely be cut in half by 1980. It is expected that by 1980 RAM's with 16K-bits per chip will be readily available. It is also expected that non-volatile semiconductor RAM's could be developed of 2 to 4K bits/chip and read/write speeds in the 200-nsec range by 1980.

CMOS semiconductor. — CMOS RAM's with access times as low as 40 nsec and complexities of 1K bits per chip are available now; by 1980 4K RAM RAM's should be available.

MNOS EAROM'S. — MNOS EAROM's of 8K bits per chip are currently under development and extension to 16K bits should be possible by 1980.

Charge-coupled device (CCD). — A 64K-bit charged coupled device (CCD) should be possible by 1980. CCD's operating at 20 MHz and with access times in the 1 to 10 μsec range should also be available for high-speed applications. Because of their serial operation, they will be used mostly for secondary storage.

Magnetic bubble memories. — Magnetic bubble memories have been constructed using single crystal garnet films and amorphous cobalt films. The presently more difficult to process amorphous material promises higher speed and can be put on almost any substrate material.

Bubble technology is very new and the eventual characteristics are difficult to predict. Density projections for magnetic bubbles range from 10^6 to 10^9 bits/in^2 by 1980. Projected access times are in the 2 to 10 μsec range, and transfer rates are estimated at 2 to 5 MHz. Also, a 100K-bit memory has been developed on a 300 by 300 mil area. This technology may become useful for avionics at some point, but it does not appear applicable to general aviation in the 1980-1990 time frame.

Optical memories. -- Optical memories are being researched using both real image and holographic storage. The major problems in this area are concerned with the storage media; a permanent, yet selectively eraseable, storage medium is needed. Techniques which may be viable in the 1980 time frame are given in Table 16 (ref. 37). Two of these approaches are described in detail in Appendix R. Optical memories will be aimed at the secondary and mass memory areas of the memory hierarchy. Optical memories along with bubbles are a threat to existing magnetic technologies for these categories. There does not, however, appear to be any application to general-aviation avionics in the foreseeable future.

TABLE 16. - CHARACTERISTICS OF OPTICAL STORAGE MEDIA

Memory material	Storage density, bits/cm^2	Write/ erase	Read	Problems
Photochromic (alkali halides)	10^6	Optical absorption	Optical absorption	Gradual data loss
Ferroelectric ($BaTiO_3$)	10^4	Ferro-electric	Current sense-destruction	Resolution-lifetime
Ferromagnetic MnBi	10^7	Thermal	Magneto-optic	Low readout efficiency
Thermoplastic	10^5	Thermal	Optical hologram	Resolution-efficiency
Semiconductor capacitors	10^7	Electron beam	Optical	Cathode life beam deflection

Memory technology projections summary. - The characteristics of selected nonmovable memory technologies with current or potential applicability to avionics are qualitatively summarized in Table 17.

TABLE 17. - CHARACTERISTICS OF MAJOR NONMOVABLE STORAGE TECHNOLOGIES WITH APPLICABILITY TO AVIONICS

Storage technology	Speed	Power consumption	Size and weight	Volatility	Reliability	Temperature range
Semiconductor						
Bipolar	VH	H	L	R/WM	H	H
MOS	H	L	L	R/WM	H	M
C-MOS	H	VL	L	R/WM	H	M
SOS/MOS	H	L	L	R/WM	H	M
CCD	L	VL	L	R/WM	M	L
Amorphous	H	M	L	NO	L	M
Magnetic						
Core	H	H	M	NO	M	L
Plated wire	H	H	M	NO	M	L
Bubble	L	L	L	NO	L	L
Optical						
Holographic	VL	VL	H	NO	M	L

Legend: VH - Very high
H - High
M - Medium
L - Low
VL - Very low

The major characteristics for selected on-line bulk memory technologies expected in the 1980 time frame are quantitatively summarized in Table 18.

TABLE 18. – EXPECTED 1980 ON-LINE BULK MEMORY CHARACTERISTICS

	Average access, μsec	Read/write cycle, μsec	Shift/bit rate, MHz	Volatile
Charge-coupled devices	1	N/A	10	Yes
Semiconductor BORAM	10		2	No
Bipolar semiconductor RAM	0.05	0.08	N/A	Yes
MOS semiconductor RAM	0.15	0.3	N/A	Yes
Magnetic bubbles	10	N/A	2	No
Fixed head disk/drum	9×10^3	N/A	0.2	No
Movable head disk	0.2×10^6	N/A	0.08	No
Magnetic core	0.3	0.4	N/A	No
Electron beam	4	N/A	10	No

Table 19 divides the total market for nonmovable storage media into the major technology areas for the year 1978. The total market (captive plus noncaptive) for semiconductor memories will be about $1.3 billion or 459 billion bits in 1978.

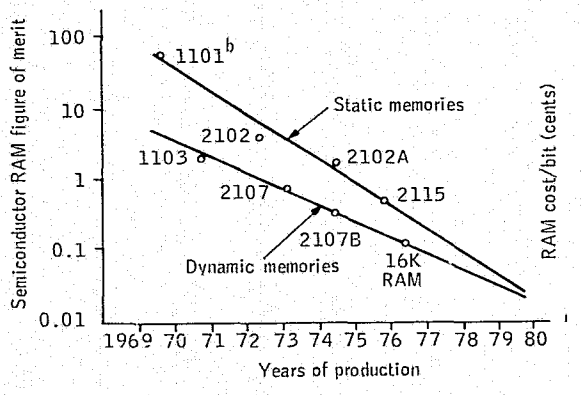
TABLE 19. - MEMORY MARKET SUMMARY

Memory technology	Quantity, billion bits	Value, million $	Component price, c/bit
Semiconductor	459	1338	0.29
Bipolar	106	667	0.63
MOS	279	562	0.20
SOS	13	64	0.52
CCD	61	45	0.08
Magnetic	74	309	0.42
Core	48	136	0.29
Plated wire	16	156	0.94
Bubble	10	17	0.17
Other	4	63	1.53
Total	537	1710	0.32

Among all the memory technologies - core, semiconductor, disk, drum, plated wire, and the more recent memory types — only semiconductors will show significant advances during this decade. These advances are size reduction, capacity increase, speed improvement, reliability improvement, and fabrication cost and price decreases. Presently almost every new computer regardless of size contains semiconductor memory or may have a choice between semiconductor and core memory. In addition to advantages in terms of performance, price and reliability, semiconductor memories offer significant advantages in other parts of the system because of their easier interfacing of logic and memory. In magnetic memory, the incompatibility of the memory signals with the semiconductor logic and the high price of the interface electronics puts economic restrictions on the system organization.

Progress in MOS technology (Figure 13) is best illustrated by plotting the figure of merit of commercial RAM's versus the year of production (Figure 13a). Figure of merit is the product of chip area (mil), dissipation (W), and access time (μsec) divided by the number of bits. Cost/bit curve (b) does not truly reflect the increase in performance with time. For

a. RAM figure of merit (M)[a] versus years of production



a. $M = \frac{\text{Area X Power Dissipation X Access Time}}{\text{Bits}}$

b. The Designations Above Identify a Family of Intel Memory Devices that are Typical of Current Production RAM's.

b. Cost/bit curve

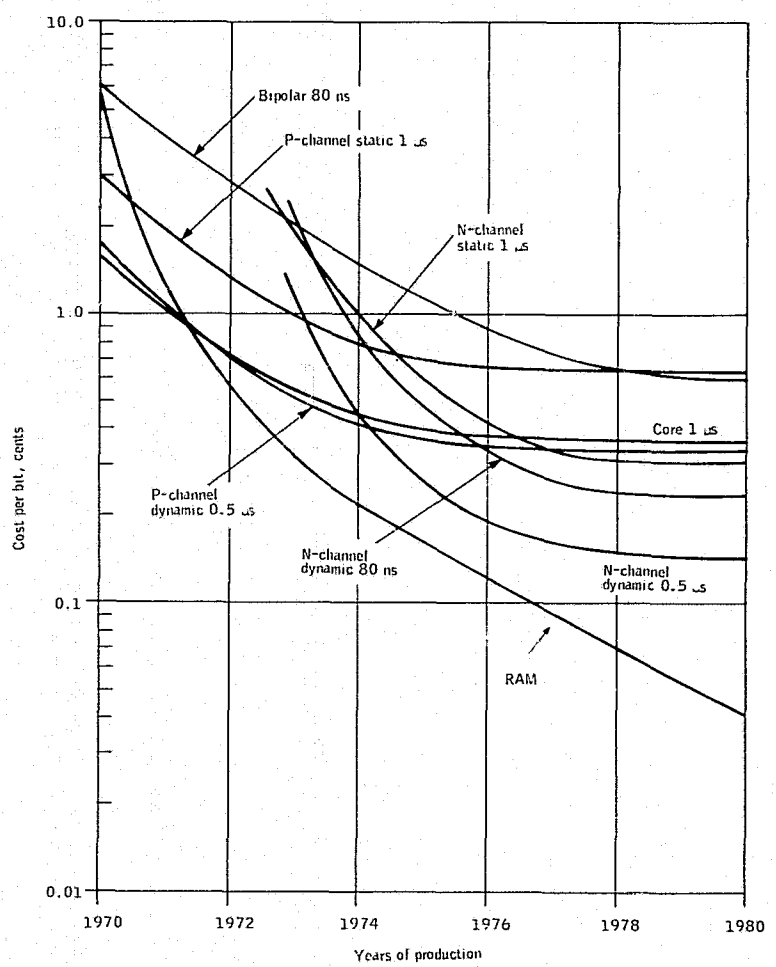


Figure 13. – Graphs Showing Progress in MOS Technology

example, a single bit of storage in 1975 requires far less power and can be accessed much faster than the same bit in 1970.

I/O Module Technology Projection

There has been very little discussion in this study of sensors and servos because this is a computer technology study and is not primarily concerned with peripherals common to analog as well as digital control systems. However, the development of both sensors and servos designed to operate specifically in conjunction with digital systems can be expected to receive an increasing amount of attention. This focus on I/O has already begun as processor costs, performance, and reliability have improved markedly.

It appears that direct digital sensor and servo development will not achieve a breakthrough equivalent to that of the microprocessor in the computational area; data converters are expected to be an important part of digital systems in the 1980's.

Digital to analog (D/A) and analog to digital (A/D) converters are getting faster, smaller, and cheaper. The trend seems to be towards packing more and more electronics into small packages. Instead of just getting an A/D converter in a module, a whole data-acquisition system is packaged as one, including an A/D converter, multiplexer, sample-and-hold amplifier and logic. Often even memory and buffer amplifiers are included. This field is still in its infancy; as it matures, the integration and miniaturization will continue. As shown in Figure 14, by 1980 more than half the data converters sold will be monolithic, about a quarter will be hybrid, and 5 to 10 percent will be encapsulated modular units. Unlike what happened with operational amplifiers, market dominance by monolithics will be considerably slower to develop.

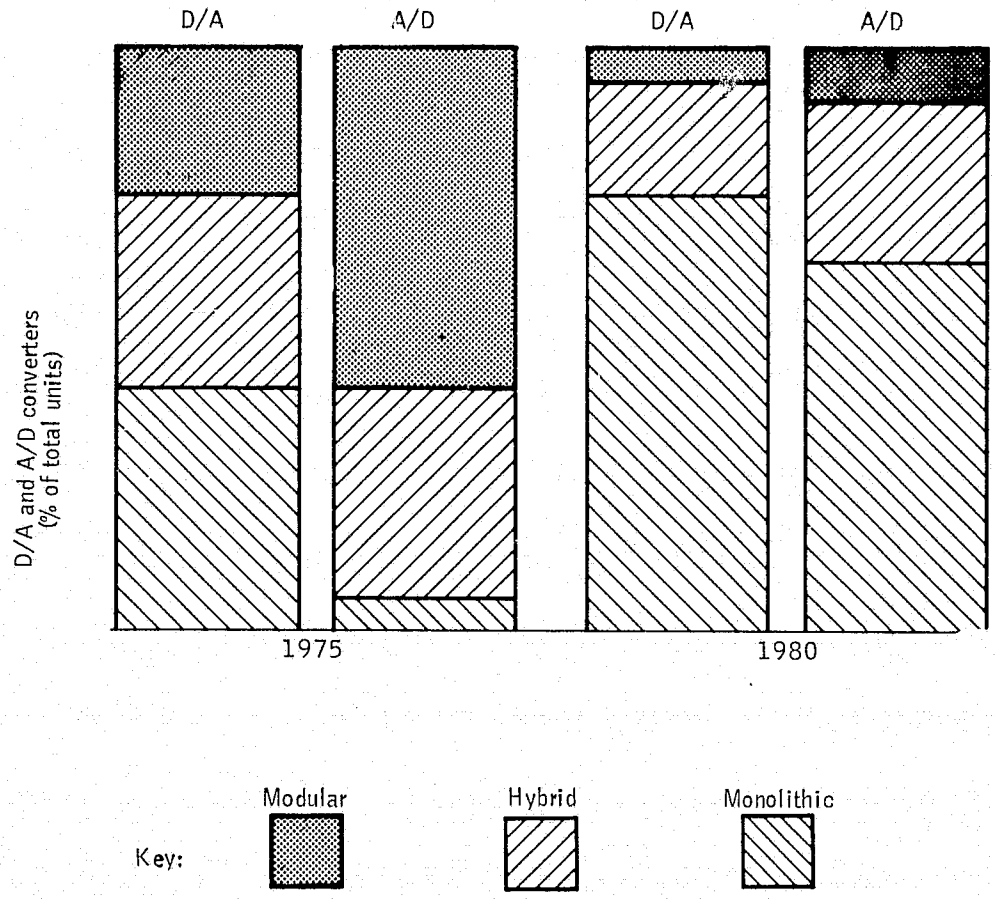


Figure 14. – Potential Growth of Modular, Hybrid, and Monolithic A/D and D/A Converters

Fiber Optics Data Transmission

A development that will have a major impact on computer systems of all sizes in the future is the use of fiber optics technology for data communications. This technology is in its infancy now, but will see rapidly increasing use as its strong inherent advantages over electrical transmission systems are realized.

Fiber optics technology is based upon the propagation of optical energy through small-diameter, flexible-fiber wave guides by multiple internal reflections. Conventional optical fibers are comprised of a transparent core material (usually glass or plastic) with a cladding that has a lower index of refraction. Light that enters the fiber within its acceptance angle will be conducted by multiple internal reflections at this boundary. This effect allows light to be "piped" wherever desired.

A fiber optic data transmission system is composed of a light source that can be modulated (LED's and semiconductor lasers are being used), a fiber optic bundle or cable and its associated connectors and couplers, and a photodetector. With the proper electrical interface, this system can directly replace an electrical wire or cable for data transmission. There are several important advantages in doing this, the most important of which are summarized in the following paragraphs.

Optical fibers have a theoretical bandwidth in the Gigahertz range. Current demonstration systems have shown reliable transmission in the 10 to 20 MHz region, and some laboratory systems have exceeded 100 MHz. This capacity allows extensive use of data multiplexing, which produces major savings in component count, power, size, weight, and cost. Also, the high bandwidth capabilities enhance the reliability and maintainability of data communications systems by permitting more imaginative data multiplexing schemes. This results in fewer cables and simpler connectors and allows redundant data paths.

Optical fibers are made of glass, plastic, or other dielectric materials; as a result, they do not pick up electrical noise. This RFI/EMI noise immunity eliminates the need for extensive shielding and careful cable routing present with conventional electrical cables.

Fiber optic cables show excellent immunity to heat (glass fibers), humidity, corrosive atmospheres, salt spray, and vibration. Performance in various strength tests (tensile, shear, abrasive, crush, etc.) has varied, but is more a function of cable structure and materials than the optical fibers themselves.

Low levels of optical energy are used to transmit information in a fiber optic transmission system; thus, loose or damaged connections cannot spark or ignite combustible vapors or materials.

A common electrical ground is unnecessary for a fiber optics data link. This eliminates the problems of ground loops, voltage offsets, and ground noise pickup. Short circuits and the possibility of spreading damage through circuit loading are also eliminated. This attribute of electrical isolation alone may justify use of the technology for intrasystem communications over short distances.

Due to the nonradiative nature of fiber optic data transmission, there is no crosstalk between data cables. This also results in vastly improved data security as it is very difficult to tap a fiber optic data cable without breaking the link.

Reliability is vital in avionics systems. This is more easily achieved with redundant data paths using fiber optics technology. A short circuit in a branch of a redundant electrical path can destroy the entire link through circuit loading. The normal failure mode in a fiber optic cable is equivalent to an electrical open circuit, which would have virtually no effect on the alternate transmission path.

There are a variety of applications areas where the advantages of fiber optic technology are important. Military and commercial avionics data transmission systems are a logical choice for near-term use of this technology as the required cable lengths are short such that very low-loss optical fibers are not required. The required data rates are also relatively low and easily achievable with current devices.

In one demonstration system, the electrical data cables associated with the navigation and weapons delivery system of an A-7 aircraft were replaced with fiber optics. Results were revealing. The number of cables was reduced from 302 to 13, the total length of cabling was reduced from 4832 to 260 feet, and the total cable and connector weight was reduced from 82 to 3.6 pounds. Also the total cost for connectors and cables was reduced from $7.9K to $1.1K. The performance of the system is excellent. Also, tests were conducted that demonstrated the RFL/EMI noise immunity of fiber optic systems.

The total lack of crosstalk and spark hazard greatly simplifies the cable routing task in aircraft design. This is a significant cost savings in the development of new aircraft systems and in the retrofit of new avionics to existing aircraft. It is clear that military avionics systems will be one of the applications areas for fiber optics. By the 1980's, fiber optic data communications will be common rather than the exception in military avionics systems.

As with any developing technology, there are significant problems limiting the immediate application of fiber optics technology. Most of the problems are associated purely with the level of maturity of the technology and not with any inherent disadvantages. Availability, standardization, and component costs are typical of the problem areas in fiber optics that will be solved as the technology matures; none of the currently known problems seem to be insurmountable. The area of coupling for multi-drop data bus applications is a significant problem area that will take innovation, research, and development to overcome. The problems of ionizing radiation (lumin-

escence and transmissivity degradation) are being attacked through basic materials research. Maintenance tools and techniques must also be developed to install, repair and modify fiber optic systems in the field. Also, to exploit the advantages of fiber optic technology, new approaches to computer system architecture and data bussing must be developed. These techniques must take advantage of the high bandwidth capabilities in optical fibers. The development of these extremely complex techniques and their application to general-aviation requirements doe not appear likely in the 1980's, but may be achieved in the 1990 era.

OPERATOR/COMPUTER INTERFACE TECHNOLOGY PROJECTION

As the computer becomes more of a direct user device, man/machine interfaces becomes critical. Methodologies for analyzing control and display parameters based on the optimal control model of a human operator have been under active development for the past decade (ref. 38). These methodologies appear to serve as useful tools for analyzing the efforts of specific controls and displays on system performance and reliability. The common conclusion offered by most of the recent studies is that a significant performance improvement is provided with integrated and interactive control/display systems. The specific display technologied anticipated to be most used during the next decade are discussed in the following paragraphs.

Prices of the fastest growing technologies — LED's, gas discharge panels and liquid crystals — will decline significantly over the next five years due to increased use and technological advances. These technologies are IC compatible and require less decode and drive circuitry to operate, which also contributes to the declining cost.

Liquid crystals and gas-discharge panels will be used increasingly in portable calculator applications as both types of devices can be powered directly by a CMOS chip and offer better legibility than LED's. The liquid

crystal temperature problems are being solved, making them more competitive with LED's and gas-discharge panels for use in calculators. Also electronic watches are a primary application area for liquid crystals. LED's cannot compete with liquid crystals in this application due to their higher cost, high power requirements, and a need for more decode and drive circuitry. LCD's will be the least expensive display.

The GDP will continue to be used for computer terminal applications and will compete with CRT's for the high-price end of the market. Significant progress has been made in gas-discharge panel fabrication techniques, circuit design, and gray-scale control for the consumer TV market. In addition, IC's for driving the GDP are rapidly improving in capability and are coming down in cost. These factors plus the effort being expended on flat panel TV make it likely that GDP will be used for consumer TV sets.

The market for flat panel alpha-numeric displays is estimated to be 800 million by 1980. To achieve a significant market penetration, active matrix display will require completely integrated drive circuits with the display.

Long-term plans are being made by both industry and government in the area of active matrix displays. For example, Hughes has developed a 1 by 1 inch matrix display with 100 by 100 elements. The display uses liquid crystal for the passive display element and is driven by a monolithic semiconductor chip. (See Appendix I). The ultimate size of the display will not be limited by the individual chip size but by techniques of assembling mod ular arrays. However, it is not expected that silicon-based matrixes will be used for very large area displays; thirty-inch diagonal displays are too great for modularization.

Silicon technology is in a maturing phase and approaching the end of its rapid growth. As this technology begins to saturate, the next breakthrough will come from thin-film technology. Thin-film integrated circuits have a much lower cost than silicone integrated circuits due to the inexpensive substrate and starting materials and the single-step batch processing.

Use of thin-film transistors (TFT) for display drive and address circuitry will be continued. It will have a wide range of application due to its ruggedness, reliability, long life features, and low cost. Current TFT matrix drives are 6 by 6 inches for liquid crystal and electroluminescent panels. Larger panel sizes are expected in the future.

The incandescent indicator display market will not be severely impacted by the new technologies because simple indicator lights will continue to be used extensively in control equipment applications. While planar incandescent displays can be produced, equipment designers are afraid of filament breakage. Therefore, segmented incandescent panel display production is declining and is being replaced by LED's.

The CRT is a mature technology and marginal improvements in performance life and reliability can be expected, but no significant cost reductions are predicted. The primary disadvantages of a CRT are its bulk, lack of memory, and high-voltage requirements. However, there is no serious challenger to the CRT for many types of displays, and it is the most cost-effective solution for a data display in the region of 500 to 1000 characters and above.

COMPUTER SYSTEM ARCHITECTURE TECHNOLOGY PROJECTION

At one time the driving force behind system level technology was the business data processing industry. The goal of development was to obtain maximum capability in a single central processing system. This lead to the development of extremely high-speed processors, and the eventual trend toward centralized multiprocessors. The development of languages and software was also directed to this type of system. This trend has now subsided.

The primary driving force in system architecture is now the so-called small user. This has led to increased emphasis on smaller, more flexible systems, which can be upgraded as requirements change. The current interest in the area of distributed microprocessor systems in one specific example of this trend. Correspondingly, increased effort in the development of software and interconnection technology for modular computer systems can be expected in the next decade.

The role of military avionics should not be overlooked in predicting the driving forces behind system level technology. In many cases military avionics may be directly applicable to general aviation with only minor scaling down of requirements. Because there is a trend in military avionics systems toward distributed processor systems, general-aviation avionics will undoubtedly follow this trend.

The concept of multiprocessor networks is well suited for microprocessors (μPs). First, microproecessors are able to perform many dedicated functions at low cost without the use of supporting electronics or special I/O chips. They can be used as a universal standard component for literally any definable task, particularly data gathering. Second, the microprocessor,

unlike its hard-wired predecessor, is capable of generalized data manipulation, information storage and retrieval, and message communications. These attributes allow two or more microprocessors to be teamed to perform tasks requiring cooperation between dedicated and message-handling functions.

SECTION 8

RISK IDENTIFICATION/TECHNOLOGY DEVELOPMENT RECOMMENDATIONS

As delineated in Section 5, a number of technologies are being developed that can be directly applied to general-aviation requirements. In most cases, the driving forces brought to bear by the large-volume users such as automotive and process control equipment manufacturers will result in continued development in directions most desirable for general aviation, i.e., increased temperature range, improved reliability, and reduced cost. In view of the overwhelming effort of the large-volume users to satisfy many of the same hardware requirements as the general-aviation avionics requirements, it does not appear necessary for NASA to expend additional effort in the semiconductor, circuit, or component development areas, except for display and sensor technology.

In the field of overall system technology, however, it does not seem that other agencies will contribute a great deal toward the improvement of general-aviation avionics. Even though bussing, multiprocessor, and fault-tolerant architecture technologies may be developed for other uses, the effective use of these in producing safer and lower-cost general-aviation systems will require effort specifically oriented toward general-aviation problems.

DISPLAYS

Passive displays. - The most promising passive display material is liquid crystal with on-site memory (see Appendix I), which is being developed by both Hughes Aircraft and Westinghouse. Currently, there appears to be adequate development funding available, a considerable portion of which is aimed at military applications. This should ensure that displays suitable for aircraft use will become available. The risk lies in being able to adapt the military technology to the general-aviation cost and reliability requirements.

The next most promising class of passive display materials are the electrophonetics (EPID's), with the colloidal-sized TiO_2 particles (Appendix K) as the best candidate. As there does not appear to be a specific aircraft-oriented program on this material, there is, even more risk than in the case of LCD's that a product suitable for general aviation will result.

Active displays. - Because of the current effort being expended on GPD's for TV use, there is a high probability that GPD's with performance suitable for general-aviation displays will be available in the 1980's. The risks in involved are the suitability of the physical configuration (size mostly) and the question of how much ambient light suppression and shielding will be required.

LED's will be available in large volume and many types, but because of the relatively high drive currents required for high brilliance, there are some doubts as to the development of the on-site memory necessary for high-resolution matrix displays. There is no doubt as to the availability of suitable alphanumeric displays.

Recommendations. - As indicated, there are several promising techniques for application to general-aviation displays in advanced development. There is risk, however, that none of these techniques will be developed into a form particularly adapted to general aviation unless specifically oriented programs are initiated. Therefore, in planning future general-aviation programs, it is recommended that NASA consider a two-step program to:

1. Define the formats and dynamic performance requirements of a 4-inch multiformat display for high-level general-aviation aircraft.

2. Sponsor the design and manufacture of prototype flat-panel displays to meet the requirements developed in Task I.

OVERALL GENERAL-AVIATION AVIONIC SYSTEM CONFIGURATION(S)

The many hardware, software, and architecture techniques discussed and cited as potentially applicable to general aviation give assurance that the performance of current functions can be improved, that desirable new functions can be added, and that cost can be reduced during the next ten years. There remains, however, the question of how best to combine the technologies and their resultant functions into overall systems that will significantly improve the safety and utility of general-aviation aircraft.

It does not seem reasonable to assume that the improved overall system technology can be adapted from on-going work on either military or commercial aviation. There are many differences, most notably in the pilot qualification and economic areas, that strongly indicate that general-aviation avionics system configuration must be considered as a unique area

of development. According, it is recommended that NASA pursue programs to:

- Determine the functional requirements of improved avionics systems for general-aviation aircraft of various classes
- Conduct design studies to define the characteristics (configuration, cost, weight, etc.) of candidate systems these will meet the established requirements.

REFERENCES

1. Winblade, Roger L.; and Westfall, Judy A.: NASA General Aviation Research Overview. 750500, 1975.

2. Staff of Decision Services Corp.: Analysis of the General Avionics Requirements for the 1980's. NASA CR-137629, 1974.

3. Bender, M. A.; et al: Digital Flight Control Systems for Tactical Flights. AF Tech. Rep. AFFDL-TR-74-69, AF Contract F33615-72-C-1058, July 1974.

4. Borow, M. S. et al: Navy Digital Flight Control System Development. Honeywell Doc. 21857-FR, Contract N62269-72-C-0141, NADC Warminster, PA.

5. Anon: Time Reference Scanning Beam Microwave Landing System. U. S. Proposal for a Non-Visual Precision Approach and Guidance System, U. S. Dept. of Transportation, Federal Aviation Administration.

6. Driscoll, Norman R.: Effects of a Simple Stability Augmentation System on the Performance of Non-Instrument-Qualified Light Aircraft Pilot During Instrument Flight, Langley Research Center, NASA, NASA TN D-3970, May 1967.

7. Anon. Microprocessor Directory and Microcomputer Systems Directory, Electronic Design News, 20 November 1975.

8. Aronson, Richard L.: Line-Sharing Systems for Plant Monitoring and Control. Control Engineering, Jan. 1971.

9. Andreiev, N.: Solid State Optical Designs Set the Encoder Pace. Control Engineering, May 1975.

10. Cushman, Robert H. (Ed.); Microprocessor Directory. EDN Nov. 20, 1974.

11. Johnson, Melvin D.: Microprocessors in Unconventional Architectures. Proc. Asilomar Conf. on Circuits, Systems, and Computers, December 1974.

12. Lampson, Butler W.; A Scheduling Philosophy for Multiprocessing Systems. Communications of the ACM, May 1968.

13. Flores, Ivan: Lookahead Control in the IBM System 370 Model 165 Computer, November 1974.

14. Tiecher, S.; and Hughes, L.: Implementation of a Hardware Floating Point Processor, Computer Design, May 1972.

15. Svigals, J.: IBM7070 Data Processing System, Proc. WJCC, 1959.

16. Staff of the Modular Computer Corporation: The Input/Output Microprocessor. 307-32002-000, 1975.

17. Hancock, R. J.: Microprogrammed Disk Peripheral Control Units, Proc. IEEE COMPCON, 1971.

18. Chung, David H.; and Hay, Don B.: The T-Box: A Low-Cost Software-Based Interface, Electronics, March 13, 1972.

19. Anon, Computer Automation; Distributed I/O Systems User's Manual. 91-53629-00A0, April 1975.

20. Steele, J. M.; and Mattson, R. C.: Architecture of a Universal Communications Processor, Computer Design, Nov. 1973.

21. Deerfield, Allan J.: Architectural Study of a Distributed Fetch Computer, NAECON Record, May 1971.

22. Thurber, K. J.: Introduction to Associative and Parallel Processors, CICS 5299 Class Notes, U. of Minnesota, 1975.

23. Enslow, Philip H. (Ed.): Multiprocessors and Parallel Processors. John Wiley and Sons, 1974.

24. Heart, F. E.; Ornstein, S. M.; Crowther, W. R.; and Barker, W. B., A New Minicomputer/Multiprocessor for the ARPA Network, Proc. NCC, 1973.

25. Jensen, E. Douglas: A Modular Computer System for Real-Time Control, Proc. Symposium on Computer Architecture, January 1975.

26. Wulf, William A.; and C. Gordon Bell; C. MMP--A Multi-Mini-Processor. Proc. FJCC, 1972.

27. Carp, P.: Origin, Development, and Status of the ARPA Network, Proc. IEEE COMPCON, February 1973.

28. Farber, David J.; Feldman, Julian; Heinrich, Frank R.; Hopwood, Marsha D.; Larson, Kenneth C.; Loomis, Donald C., and Rowe, Lawrence A: The Distributed Computing System, Proc. IEEE COMPCON, February 1973.

29. Walker, Donald E.: The SRI Speech Understanding Systems. IEEE Transactions on Acoustics, Speech, and Signal Processing, Oct. 1975.

30. Anon: Printers/Plotters Digital Design, Dec. 1975.

31. Rosenthal; Rabiner; Schafer; Cumminsky; and Flanagan: A Multiline Computer Voice Response System Utilizing ADPCM Coded Speech. IEEE Transactions on Acoustics, Speech, and Signal Processing, Oct. 1974.

32. Bailey, S. J.: Incremental Servos for Digital System Compatibility. Control Engineering, Jan. 1975.

33. Reingold, Irving: Display Devices; A Perspective on Status and Availability. Proceedings of the S. I. D., Vol. 15-2, Second Quarter 1974, pp. 63-74.

34. Van Raalte, J. A.: Matrix TV Displays; System and Circuit Problems. IEEE Transactions on Consumer Electronics, Vol. CE-21, No. 3, Aug. 1975, pp. 213-219.

35. Sobel, Alan: Summary; New Techniques in Video Displays. IEEE Transactions on Consumer Electronics, Vol. CE-21, No. 3, Aug. 1975, pp. 290-297.

36. Brody, T. P.: Large Scale Integration for Display Screens. IEEE Transactions on Consumer Electronics, Vol. CE-21, No. 3, Aug 1975, pp. 260-287.

37. Hodges, D.: Alternative Component Technologies for Advanced Memory Systems, Computer, September 1973, pp. 35-37.

38. Anacker, W.: Superconducting Memories Employing Josephson Devices. Proceedings, National Computer Conference, May 1975, pp. 529-534.

APPENDIX A

CIRCUIT TECHNOLOGY

This appendix summarizes the salient characteristics for representative circuit types listed in Table 6 of the main body of the report.

Epitaxial Collector Structure

The EPI collector device structure is the most common bipolar processing method in use today. It has resulted in a large number of unique circuit forms as shown in Figure A1. The most promising of these for future applications are I^2L and DCT^2L.

The size and low-power advantages of I^2L come directly from shrinking the old direct coupled transistor logic (DCTL) into a single complementary transistor equivalent. As shown in Figure A1, a vertical NPN transistor with multiple collectors operates as an inverter. A lateral PNP transistor serves both as a current source and load. Note that no ohmic resistors are required for either the source or load function. In a typical TTL gate, six to eight area consuming transistors are required.

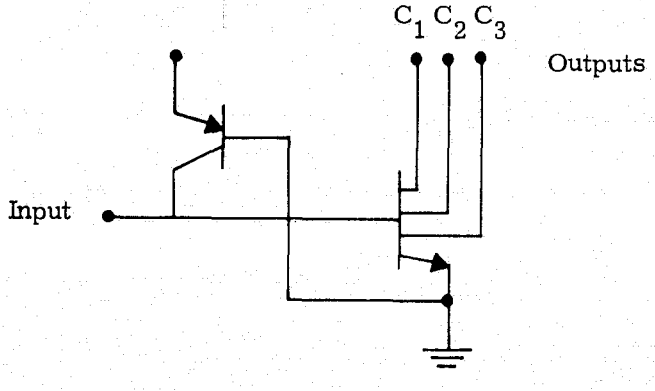


Figure A1. I^2L Gate

Note also that the base of the multicollector NPN transistor is common to the collector of the lateral PNP current source, and that the base of the current source is common with the emitter of the NPN transistor. This means that both transistors can be fitted into the area of a single transistor without requirements for device isolation or intracell wiring. This gives rise to the term-merged transistor logic (MTL), which is another name for I^2L. Other names used for logic types that are basically I^2L are:

- CHL -- Current Hogging Logic
- C^3L -- Complementary Constant Current Logic
- SFL -- Substrate Fed Logic

The essential feature of I^2L logic is the use of a transistor as the current source and load rather than the conventional resistor. All units are powered by these transistors, and the current that reaches each unit is simply the total injector current multiplied by

the α of the PNP, and divided by the number of units associated with the injector. Thus, by altering the current into the injector, a wide range of speed-power operation can be achieved.

I^2L devices fabricated by conventional bipolar processes are presently limited to a delay of 10 to 30 nsec/gate. If the process is optimized for I^2L, it is felt that delays of 1 nsec can be achieved. I^2L devices can be included on the same chip with Schottky TTL, ECL, linear circuitry or other circuit forms because of process similarity. This feature can be very useful in the reduction or elimination of special interface circuit requirements.

The only standard product presently available in I^2L is the SBP0400 by TI. This 4-bit processor element chip contains over 1450 gates and is fully TTL compatible.

Triple Diffusion Structure

The triple diffusion device structure (3D) is a relic in integrated circuit technology that has been recently revived for LSI applications. This structure, introduced in the early 1960's, offered the most direct and simplest means of producing electrically isolated transistors and resistors on a chip. This structure gave way to the epitaxial structure due to the difficulty in controlling the light collector diffusion required. With the availability of ion implantation, this deposition can now be controlled within 5 percent, and this structure is again a viable choice. This process allows both NPN and PNP transistors as well as self-isolated resistors to be fabricated on the same chip.

The simplicity of the process required in 3D allows very large chips to be fabricated with reasonable yields. To date, over 50 LSI designs have been produced with chip sizes as large as 300 by 300 mils. The high density offered by 3D has allowed these chips to contain from 2000 to 20,000 devices per chip.

Presently two circuit forms employ the triple diffused structure, emitter follower logic (EFL), and emitter coupled logic (ECL). Note that ECL can be realized with any of the three bipolar structures. Most of the LSI effort to date has been in EFL.

Basic EFL gates are shown in Figure A2. Note that both PNP and NPN transistors are employed to give wire-AND and wire-OR functions.

EFL presently operates in the 5- to 10-nsec speed range (like TTL), but exhibits power and size advantages over TTL, which is fabricated using the standard EPI structure. While not as dense as I^2L, EFL does not require the tolerance on current levels of I^2L and is thus less sensitive to low-level defects that cause leakage. As a result, much larger chips can be fabricated with good yield using EFL.

Presently under investigation is the combination of EFL and CML (current mode logic) on the same chip. The reason for such a development is that while EFL has more power to perform combinatorial logic, CML performs register logic better than EFL. Such a combination would allow speed and density advantages over the use of a single circuit form.

Oxide Isolation Structures

Bipolar transistors involve bulk phenomena and, therefore, must be isolated from one another when combined on one integrated circuit chip. In the EPI collector structure, this isolation is accomplished by diffusion to form P-N junctions between devices. Because the diffusion areas spread laterally as well as into the wafer, this form of isolation consumes considerable chip area. Thus, in EPI structures, most attempts to increase

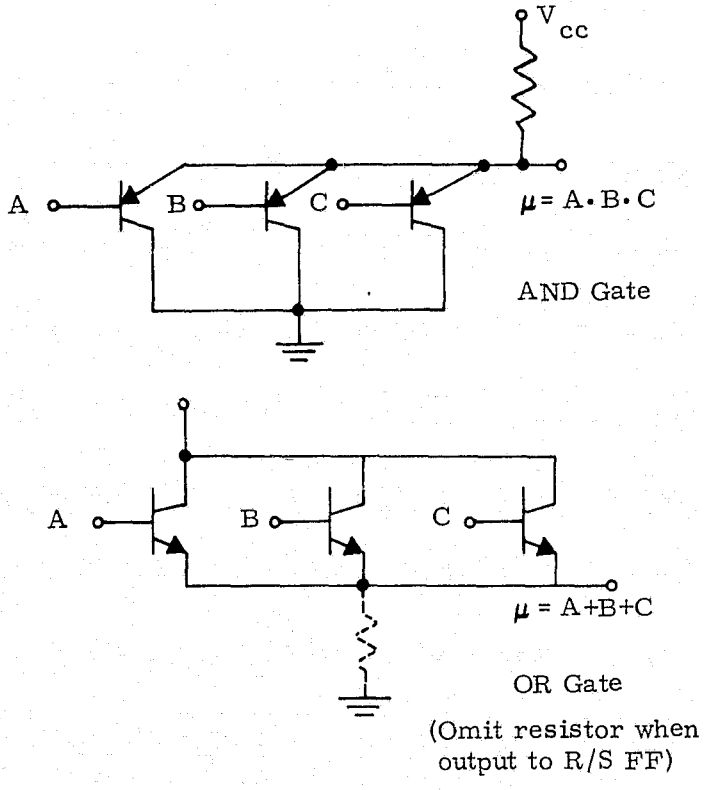


Figure A2. Basic EFL Gates

density involve the use of very thin epitaxial layers and minimization of the size and number of components required for a logic function.

With oxide isolation structures, diffusion isolation is replaced by a form of dielectric isolation where material is removed around each device. This can be accomplished with a much narrower isolation region than results from diffusion. The material is removed by an etch, which does not have as much lateral movement as in diffusion. The use of such an isolation wall eliminates the need for a collector sinker diffusion, thus saving additional area.

Specific names given to this basic structure are isoplanar (Fairchild), VIP (Motorola), and V-ATE (Raytheon). While there are differences in each, the result is the reduction in device geometry through advanced isolation techniques. However, while this structure results in area advantages, the processing steps are more critical, and yield over a given area is not as high as with the standard EPI structure.

While nearly all circuit forms that are fabricated using the standard EPI process can be made with oxide isolation, only T^2L, ECL, and I^2L have employed this option to date. Of these, I^2L has already been discussed, and the other two are commonly known circuit forms, not warranting further elaboration here.

The basic impact of the oxide isolation structure is that it provides better device isolation, both in terms of leakage capacitance and area. Thus, smaller chips with improved speed-power products can be achieved for a given complex function. The process control is more stringent, thus resulting in lower yield over a given area. The

choice of which structure to use will be a function of the degree of process control that evolves and/or the price per function that is tolerable.

Metal Oxide Semiconductor Field Effect Transistors (MOSFET)

The field effect transistor is a majority carrier device where the current flow is modulated by an electric field rather than gate current. Unlike bipolar technology where most device characteristics are a function of diffusion processes, the MOS device is primarily dependent upon the geometry of the gate and its insulation structure. Because the basic phenomena are surface related, surface impurities are much more critical.

The main advantages of MOS over bipolar technology are:

- Smaller device geometry
- Lower power dissipation
- Higher yields due to fewer process steps
- Use of cross-under conduction paths which reduce requirement for multilayer metallization

The main disadvantages of MOS with respect to bipolar are:

- Slower switching speed
- Limited chip-drive capability

At present, there are three basic logic configurations in which MOS can be obtained in low-cost, high-density arrays:

- Static
- Dynamic
- Complementary

Conventional static or d-c circuits employ circuit forms similar to those used in bipolar circuits, and a direct path exists from the power supply to ground through a resistance when the gate has been turned on.

The basic principle of dynamic logic is the elimination, through pulse techniques, of this d-c path between power and ground. This is accomplished by multiphase clocking of the load devices. Dynamic logic has device size, speed, and power advantages over static MOS. The major drawbacks of dynamic logic are the required minimum clocking rate (to preserve data integrity), and the difficulty of generating and distributing high-speed, high-voltage clock pulses. These disadvantages have limited dynamic MOS mainly to memory or register arrays, with random logic primarily realized in static form.

In complementary MOS, both P-channel and N-channel transistors are formed on the monolithic substrate and are arranged such that very low power is dissipated in the logic. This will be discussed in a special section on CMOS.

Of the circuit forms shown in Figure A3 for MOSFET structures, P-MOS is falling from use and will therefore not be discussed in any detail. Very few new designs are expected to employ P-MOS. The main reasons for this decline are the incompatibility with TTL and the emergence of N-MOS.

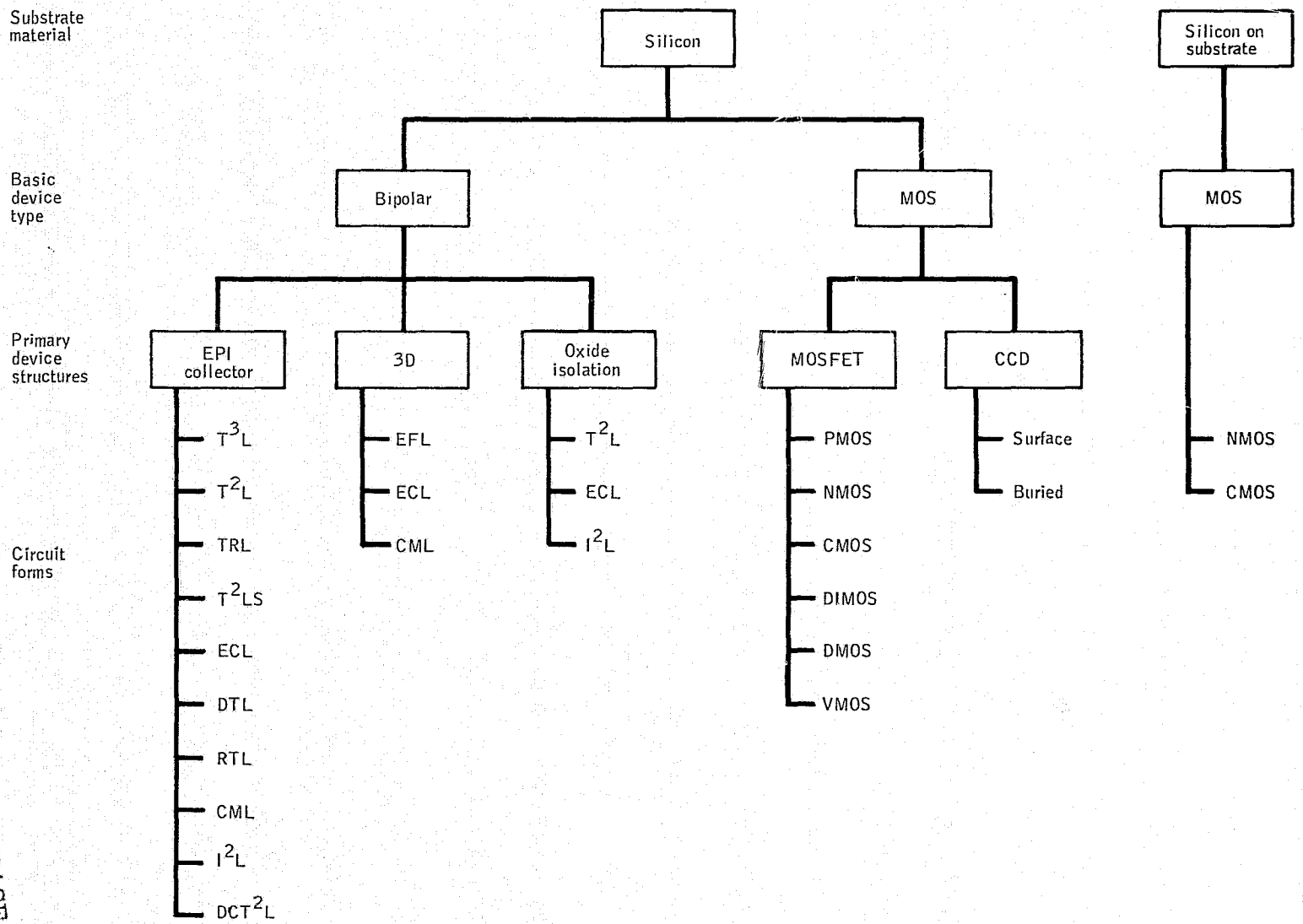


Figure A3. Semiconductor Technologies

N-Channel MOS (NMOS). - N-Channel MOS circuits possess a carrier mobility that is much higher than that available with P-Channel circuits and, therefore, they exhibit higher speed characteristics. This was realized years ago, and attempts were made at that time to process N-Channel devices. Because N-Channel devices are much more susceptible to surface impurities and because process controls were not very good, this process was dropped in favor of PMOS. After years of experience in processing PMOS, the processes became very clean and it was found that NMOS could be economically produced. In recent years, almost all new circuit starts are in NMOS. As an example, in 1974, nine of the 10 producers of the 4K MOS RAM used N-Channel devices.

Most N-Channel devices today employ the self-aligned silicon gate process rather than the area-consuming metal gate process. As shown in Figure 3 of the main body of the report NMOS can operate in the 10 nsec/gate region with a speed/power product in the 10 pj range. With a 6 to 14 mil^2/gate, NMOS is also a high density LSI technology. Because of the process simplicity and widespread experience in this technology, NMOS will see widespread use in LSI developments.

A number of alternate structures have been postulated to further improve the characteristics of NMOS; two of these are DMOS and VMOS.

Double-diffused (MOS (DMOS). - Double-diffused MOS technology yields a high-performance MOS transistor made by bipolar techniques. It uses the time and temperature of diffusion cycles to determine transistor parameters in contrast to the standard MOS process in which characteristics are controlled by geometric tolerances. This process allows very short channels to be achieved with ordinary photolithography and results in a very high speed ($\approx$1 nsec) and high density LSI circuit form. The primary drawback is that the fabrication process is nearly as complex as today's bipolar circuits.

V-Groove MOS (VMOS). - This MOS structure is also borrowed from bipolar technology, and uses a preferentially etched V-shaped notch, like V-ATE. VMOS allows fabrication of high-performance transistors with very small channel lengths and twice the current density of DMOS as well as higher packing density. Another advantage of VMOS over DMOS is simpler processing.

VMOS is capable of very high-speed operation (2 nsec/gate) at very high levels of integration (3 to 7 mil^2/gate) and is competitive with bipolar I^2L. VMOS is most commonly used to fabricate N-Channel devices, but results in a two to one size reduction over the conventional NMOS structure.

Complementary metal oxide semiconductors (CMOS). - The name complementary for this logic form is based on the use of both N-Channel and P-Channel transistors on the same substrate. The principal advantage of CMOS is the very low power consumption (nanowatts/gate) in the quiescent state. As shown in Figure A4, the basic gates consist of a P-Channel device and an N-Channel device between the single power supply and ground. The voltage at the input will enable one of these transistors and will disable the other. Because the input impedance of the transistor pair is extremely high (typically 10^{12} ohms), nearly all the power is consumed during the transition from a logical 0 to a logical 1 when both transistors are momentarily on. Thus, the power dissipation is a direct function of the frequency of operation.

For most other logic forms, the d-c power is so high that the a-c component is not discernible. With CMOS, the opposite is true: the a-c power ($P_{AC} = CV^2F$) is much higher (at 5 pf per load off chip, 1 pfd on chip) than the nanowatts of d-c power. Because of this feature, the power dissipation ranges from nearly zero at low frequency to greater than TTL at a few MHz.

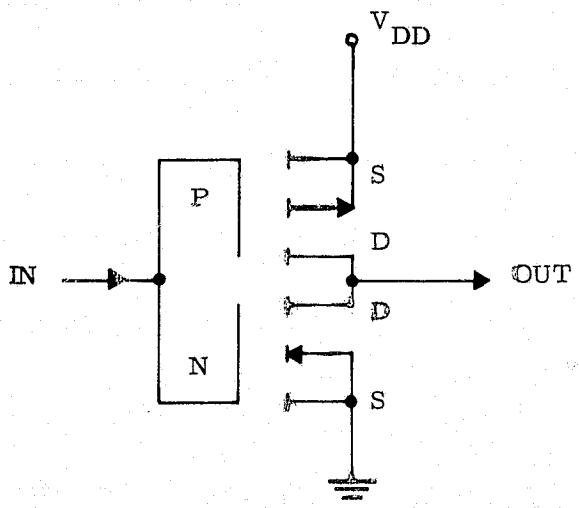


Figure A4. Typical CMOS Inverter

The advantages of CMOS include:

- Low power
- Operation from a single power source
- Logic swings equal to supply voltage
- High noise immunity (45% supply)
- Insensitivity to supply (3 to 15 volts)
- High speed
- High fan-out (typically 50 loads)
- Compatibility with TTL

The disadvantages of CMOS include a loss of packing density caused by three factors. The most significant factor is the need for guard rings around the devices to ensure high field thresholds. The second factor is the larger number of devices required to perform a logic function in CMOS; a CMOS gate requires 2N devices, where N is the number of inputs. A single channel MOS device requires only n + 1. Thirdly, the N-Channel transistors have higher carrier mobility; they have roughly twice the transconductance of the P-Channel transistors with identical geometry. Therefore, to match the characteristics of the pair, the P-Channel device with a given channel length must have approximately twice the channel width of the N-Channel device. Thus, not only are more devices required, but only half of them can be of minimum geometry.

Most CMOS today uses the metal gate structure, with very few silicon gate devices available. If the industry solves the process problems associated with Si gate CMOS, density improvements will result. Presently CMOS chips are roughly twice the area of an equivalent N-Channel Si gate chip.

A form of CMOS that is similar to the isoplanar structure in bipolar is called DI-MOS, or DI-electric isolated MOS. This structure reduces the output capacitance of the CMOS gate to essentially zero, offering improved speed and a-c power advantages over the more common junction isolation.

Appendix A

Charge Coupled Devices (CCD)

One important offshoot of the basic MOS device structure is the Charge Coupled Device (CCD) announced by Bell Telephone Laboratories in 1970. The CCD is a MOS-integrated circuit shift register for analog signals, which is made in the form of a string of MOS capacitors. The initial work was in surface channel devices, where metal electrodes were placed on a silicon dioxide insulation layer over a P-type silicon substrate. No diffusions are required for surface-channel devices. However, because of the noise caused by the trapping of charge at the Si-SO_2 interface, most effort is now in buried layer CCD's.

Buried layer devices are formed by selective diffusion or implantation of N-type silicon into a P-type silicon substrate. In comparison to MOSFET structures, CCD's are very simple to process, and thus result in good yield over a very large area. No isolation is desired between devices, making this a very high density technology.

CCD's operate as shift registers or delay lines, where the mobile charge stored within a semiconductor element is transferred to a similar, adjacent storage element by the external manipulation of voltages. The quantity of the charge can vary widely, depending on the voltage applied and the capacitance of the storage element. Thus CCD's can represent analog as well as digital data.

Many companies are presently developing CCD technology using a number of unique approaches, and government funding is assisting this development. A great number of these devices can be placed on a single chip. For use as digital devices, the complexity is a function of geometry and yield; for analog devices, the chip complexity is more a function of transfer efficiency as analog repeaters are difficult to fabricate. CCD's are useful in three application areas:

Digital memory. - The potential low cost and fast access of CCD serial memories make them candidates for replacing the slow, unreliable electromagnetic disks and drums presently in use. More discussion of these devices will be presented in the memory section.

Imagery. - In imaging, charges are introduced into the device when light from a scene is focused onto the device's surface. As in all semiconductors, the absorption of light quanta creates hole-electron pairs which, under the influence of the potential beneath each storage electrode, are collected as a charge pocket. The quantity of charge stored is proportional to the intensity of the image. In this manner, a spatial charge representation of the scene is stored in the device. By clocking the electrodes, the charge pockets are moved from site to site until they have all been read out.

Applications in imagery for CCD's include:

- Sensors for TV cameras
- Facsimile recording
- Optical character recognition

Signal processing. - The analog properties of CCD's allow this technology to be used in signal processing applications, and efforts are presently underway to realize:

- Analog delay lines
- Reprogrammable filters
- Cross correlators
- Fourier transform processors
- Multiplexers

These devices can eliminate the need for D-A/A-D conversions and can provide cost, size, weight and power advantages over digital approaches in many cases.

The clock frequency control of CCD's allow analog delay lines to provide a delay ranging across several octaves, from microseconds to milliseconds per stage.

By adding a conventional MOS PN junction alongside any desired CCD elements, a tapped delay line can be constructed. Signals can then be pulled out at any specified point along the line, processed, compared to other independent signals, fed back into the delay line, multiplexed, added, subtracted, and manipulated in various manners, while maintaining their analog nature at time intervals that are simply controlled by a clock frequency. Because the transfer of data between CCD elements is not completely loss-less, and because the charge in any element decays with time, the length of the array and the minimum clock frequency must be limited. The analog signals will accumulate the stage-to-stage errors and will result in degraded performance.

Silicon on Substrate (SOS)

All previously discussed IC types have been fabricated on monolithic silicon wafers. As has become obvious, most of the attempts at circuit improvements have been concerned with improved device isolation. With this approach, no bulk silicon is used; instead, a thin film of silicon is grown on an insulating substrate, and silicon islands are formed by selective etching. The transistors are then formed in the standard way except that no guard band diffusions are required. To date, almost all efforts in this technology have employed sapphire as the insulating substrate. Some work was done using Spinel, but it appears to be inferior to sapphire.

The benefits of this approach are the size reduction (see Figure A5) and the lower capacitive characteristics that are achieved by the air isolation between devices. This isolation results in improved speed, lower power, and improved radiation tolerance. A CMOS-SOS chip would be roughly 20 to 30 percent smaller than an equivalent N-MOS Si gate chip.

The main drawback to this approach has been the difficulty in putting single-crystal silicon on the substrate efficiently. This problem now seems under control, as standard MSI circuits are being offered for purchase. The problem stems from the difference between the crystal structure of the substrate and that of the silicon. Distortion in the silicon structure slows the minority carrier mobility to about 300 CM^2/V-sec as opposed to 1500 CM^2/V-sec in bulk silicon. Despite this phenomenon, high speed is achieved due to the reduction in parasitic capacitance.

SOS has been limited thus far to MOS circuit forms, with the highest emphasis on CMOS. The military is very interested in CMOS/SOS because of its inherent resistance to radiation.

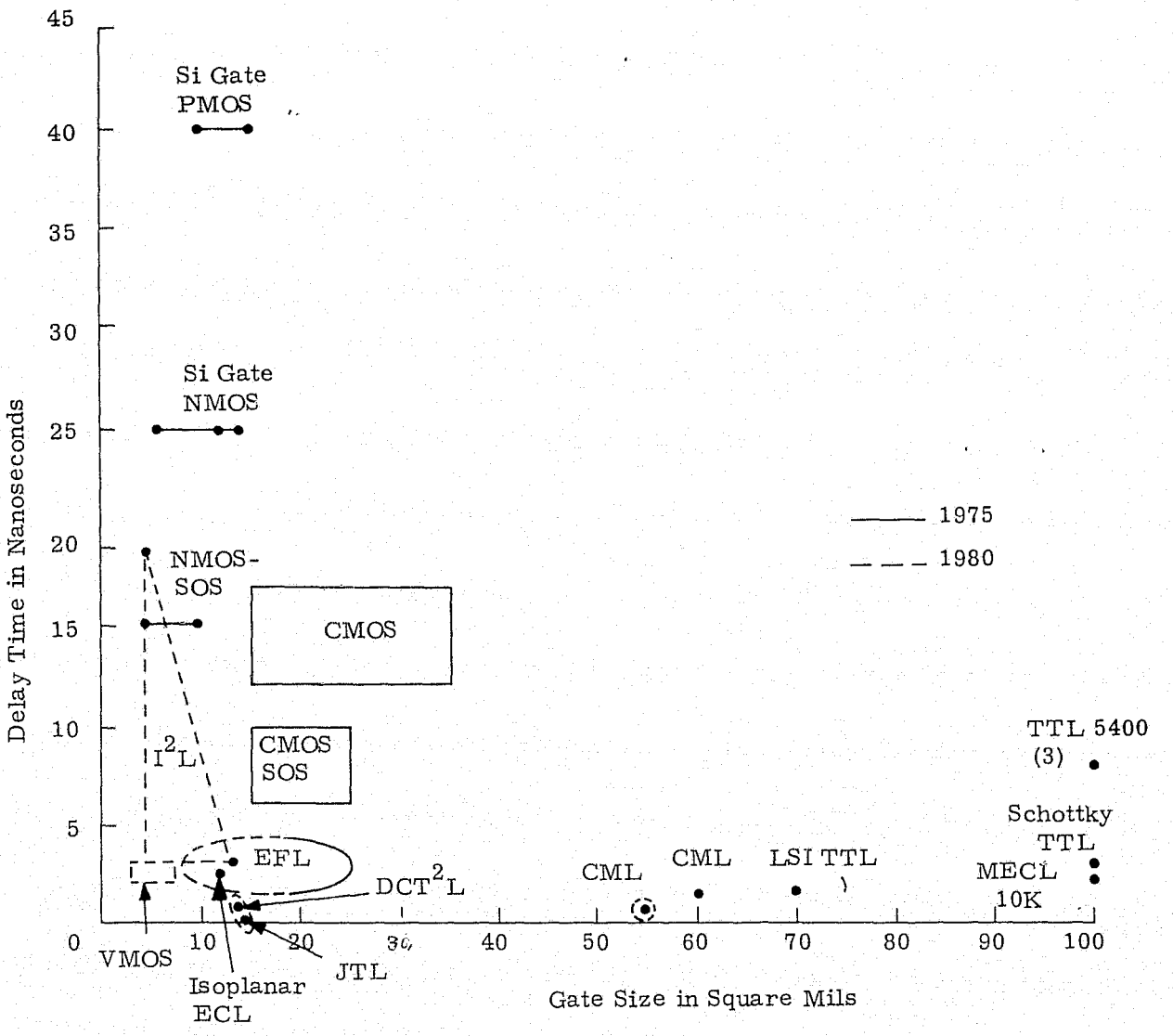


Figure A5. Gate Delay Versus Gate Size

APPENDIX B

ARCHITECTURAL GUIDELINES FOR THE EFFECTIVE USE OF LARGE SCALE INTEGRATED CIRCUITS (LSIC) IN AVIONICS SYSTEMS

Introduction

The advent of LSI has provided system designers with a powerful new tool to meet the expanding requirements of new avionics systems. This tool requires new approaches to system architecture. Although some of the standard criteria for enhancing system performance minimize costs, they are now obsolete. Until recently, the only IC implementation tools available were a number of small-scale integrated circuits, which realized an average of about four logic functions per package. This resulted in cost, size, weight, and power being almost a direct function of the number of gates in the system; thus, gate minimization was emphasized. Because complexity was minimal, the design of standard devices, which were used many times in a system, was relatively easy. These devices, because of their high usage, soon become very inexpensive.

Semiconductor technology now permits putting thousands of logic gates on one monolithic chip. This capability has been hard to use as it is difficult to define commonly used arrays of such complexity, except in the area of memories. This capability has also greatly changed the system implementation considerations. Regular logic, such as memories and registers, can now be achieved with greatly reduced cost, size, and weight per bit. Random logic can be realized in high density custom arrays. Pins and packages become more important than gates.

The spectrum of available logic families has also greatly expanded, covering a wide speed and power range. A number of Metal Oxide Semiconductor (MOS) families are now available, as well as new bipolar types.

Present-day system designers must know all semiconductor technologies available, and the advantages and limitations of each. He should also be cognizant of the relative tradeoffs of standard and LSI implementation on all system requirements. A classic example is provided by the RCA efforts on LIMAC, which showed that computers designed for SSI optimization could not be efficiently implemented in LSI. A new design approach was shown that would result in a 5-to-1 improvement in gate-to-pin ratios, one of the most important design parameters for an LSI system.

A number of system concepts have become more economical due to the availability of LSI:

- Replacement of analog hardware by digital
- Use of semiconductor memories
- Increased use of parallelism
- Increased use of modularity
- Microprogramming
- Replacement of software by hardware
- Increased fault tolerance
- Reduced system interconnections

Each of these concepts is discussed below.

Appendix B

Replacement of Analog by Digital

There is a marked trend towards replacing analog computations with digital because of some of the shortcomings inherent in analog computation, and the arrival of high-density, low-cost digital circuitry. This concept includes replacement of electro-mechanical devices by digital electronics.

The emergence of LSI has accelerated the trend from analog to digital which began a number of years ago. This is due in part to the difficulty of integrating large analog computational blocks. As a rule, analog devices require large values of capacitance, inductors, and precision resistors. These components are much more difficult to achieve with monolithic circuitry than the transistors, diodes, and resistors that form the digital circuits.

Advantages of digital over analog computation include:

- Adaptability to LSI
- Compatibility of the output with digital equipment used in subsequent computations
- Ability to process very low-frequency signals that would require analog devices prohibitively large in size
- Relative insensitivity of digital equipment to environmental conditions
- Increased stability of operation
- Ease of changing performance characteristics
- Ability to time share logic to serve a number of inputs
- Advantages in size, weight, and cost
- Simplified and standardized data bussing and addressing
- Elimination of long settling times
- Ability to hold data history for longer periods of time

Examples of digital computations replacing analog are discussed below to illustrate the above described benefits.

Applied Devices Corporation is developing a LORAN airborne navigation system for the Army. While a conventional LORAN system requires from 0.42 ft^3 to 0.84 ft^3, the digital version is expected to be less than 0.1 ft^3.

Honeywell has designed a digital signal processing module that can be realized on a single LSIC. This device can perform a second-order digital filter computation where the bandwidth and the Q of the filter is determined by values loaded into internal shift registers. Then, one LSIC can be programmed to perform as a number of unique filters.

Thus, the first step in defining an avionics system for LSI implementation should be to replace as much analog computation as possible. Input conversion from analog to digital should take place as soon as possible, and output conversion from digital to analog should be at the last possible stage in most cases. The replacement should be accomplished by a thorough investigation of the function to be performed, rather than just attempting a 1-to-1 substitution of analog blocks by digital.

Appendix B

Use of Semiconductor (LSI) Memories

Until the advent of LSI, using small memories throughout a system was very expensive on a per-bit basis. Magnetic memories, because of the high cost, size and power requirements of the associated electronics on a per-bit basis, were only efficient if the number of words was large in relation to the number of bits. In addition to this restraint, the speed of magnetic memories was not compatible with that of the IC logic.

Before LSI, when, at most, four flip-flops were contained on a chip (in a package), the use of small IC memories was also expensive in terms of size, power, and cost. The increased pin interconnections also posed a reliability problem, thus severely limiting the use of registers.

Memory technology, with its high regularity, universal appeal and low pin-to-bit requirements, is ideally suited to implementation in (ref. B1). As a result, many LSI memory devices are available today as off-the-shelf items. These devices, with their high gate density, high speed, low cost and low power consumption, can provide new system design flexibility.

In addition to the conventional read/write Random Access Memories (RAM), a large number of Read Only Memories (ROM), Content Addressable Memories (CAM), and other structured memories are available to provide even more design flexibility. To date, the largest influence of LSI on avionics system architecture has been the increased use of shift registers, ROM's, CAM's, and conventional RAM memories. This trend should be accelerated in the future. The main drawback to the use of LSI memories (other than ROM's) is the inherent volatility (lack of binary remnance upon interruption of power) of semiconductor devices.

Distribution of small LSI memories throughout a system can be used to:

- Increase computational throughput
- Increase system flexibility
- Reduce system overhead
- Reduce software costs
- Reduce system internal data transfers

Random access memories. - Low-cost RAM chips can be used to implement scratchpad memories in place of the single accumulator architecture, typical of some small computers. The scratchpad memory allows the storage of machine status, privileged data, base addresses, intermediate results, stack pointers, plus a general register store for use by the program. Previously, all of these software aids had to be stored in main memory. The programmer (or system executive) had to manipulate these aids through software, which resulted in high overhead because of many main memory accesses. With microcoding and these scratchpads, this hardware can be made to support many macro instructions, which simplifies the software task as well as minimizing main memory access, thus increasing system speed.

Use of small LSI cache memory can also increase system throughput. Buffering the main memory with a small, very high-speed memory allows repetitively accessed words to be read and written very fast, while minimizing access conflicts to main memory. Thus, the CPU is not strapped to the relatively slow main memory cycle time and can effectively execute near the high-speed memory rate.

The availability of low-cost, high-density RAM's makes multicomputer systems, each with a relatively small main memory, a viable concept.

Read-Only Memories. - Advances in Read-Only Memory (ROM) have already had a profound effect on system architecture. The availability of high-speed, low-cost LSIC

ROM's have enhanced such concepts as microprogramming, table look-up, hard-wired subroutines, and higher-order language interpreters. Microprogram memory using LSI ROM's can be used to realize multiple functions from the same computational unit, making it an efficient special-purpose device through programming. The low cost of LSI ROM's make table look-up of common functions a viable means of saving computational time, program memory and software overhead. Some special-purpose functions, such as coordinate conversion, code conversion, or sine/cosine calculations, can now be obtained in off-the-shelf ROM's.

LSI ROM's have been shown to be a viable implementation of high-speed arithmetic. For example, by using eight Texas Instrument bipolar 256 by 8 ROM chips and three 8-bit adders, a 16-bit product can be obtained in less than 100 nsec (ref. B2). The same 8 bit-by-8 bit multiply can be achieved using MOS chips in 1.0 μsec, while requiring only 8K bits of ROM.

ROM's have also been used to successfully replace random logic, thus decreasing system cost. Using ROM to implement random logic has the following advantages:

- Low development costs as compared to custom-designed random logic LSI
- Short development lead time
- Regular high-density logic with high yields
- Increased flexibility (Universal Logic Module) Reference
- Fewer chip types or die required
- Lower inventory required

By far, the greatest architectural impact of LSI ROM's has been their influence on microprogramming. Because of the inherent speed and the decreased cost of LSI ROM's, microprogramming has become a viable substitute for random logic controllers. (The architectural aspects of microprogramming will be developed in a later section.)

Memory with imbedded logic. - With the reduced cost per gate of LSI, a new family of memory logic devices has been born. These structures are very regular and can be efficiently implemented in LSI while containing the control logic that tailors them to performing special and more complex functions than those available through simple memories. Two good examples are the Content Addressable Memory (CAM), which is presently used to support the cache memory in the IBM System 370, Model 165, and stack memories, which have recently been made available as off-the-shelf items.

Content addressable memories (CAM's): CAM's can increase computational speed by eliminating sequential file searches. Rather than accessing each item in a data set one at a time, as is done in a conventional memory, a CAM can search the entire data set or any designated portion of it simultaneously. It can be used to read, write, or perform equality searches over all words in parallel. Additionally, with the aid of microprogramming, a CAM can perform addition, subtraction, logical operations, and maximum, minimum, or between-limits searches over all memory words in parallel. This drastically reduces the overhead of system control and scheduling, and many other computations.

Stack and queue memories: Another new form of memory with internal logic is the hardware queue and the hardware stack. They are sometimes also termed as first-in, first-out (FIFO) queues and last-in, first-out (LIFO) stacks, respectively. For many years, programmers have implemented these structures in software subroutines at a significant overhead cost. Now, LSI offers hardware implementations that give a 4-to-1 speed advantage for each access to the structure. A major use of FIFO queues is the buffering of data transmission between two devices of different data rates. This allows for an asynchronous operation of both devices at their maximum data rates. Additionally,

these stacks are well suited to reducing executive scheduling and interrupt processing overhead. Burroughs has used stacks and queues to reduce system overhead in many of their machines.

Summary of memory applications. - LSI memory technology has had a great impact on system architecture. LSI has made economical and feasible small distributed memories, RAM, ROM, and memories with internal logic, CAM, queues, and stacks. These memory elements can be used by the system designer to alter machine organization and to increase system speed, flexibility, and throughput, while decreasing overhead and data transfers. The new memory technology forms an integral part of the concepts to be discussed in the next few sections.

Thus, in defining any avionics system, the system designer must be aware of the advantages that can be realized through LSI memories. Basic guidelines for using LSI memories include:

- Use small RAM's to reduce main memory requests (scratchpads, caches, stacks and queues, and registers)
- Use ROM's for microprogramming
- Use ROM's for table look-up
- Use ROM's for hard-wired subroutines
- Use CAM's to eliminate file searches
- Use queues to buffer data transfers
- Use queues and stacks to minimize executive overhead

Increased Use of Parallelism

The attributes of LSI make it suitable for use in systems with a high degree of parallelism. Because LSI can result in very high gate densities, the inclusion of additional hardware to increase computational ability can still result in smaller systems than those presently in use. Prior to LSI, parallelism was a costly concept, and was used only for physically large, high-speed computers or to achieve high reliability.

The concept of parallelism in digital systems means concurrent operations on segments of a single process or simultaneous execution of multiple processes. To achieve concurrent operations on segments of a single process, one can operate on a word-parallel basis rather than a bit-serial basis, or can include unique functional blocks to work on different aspects of a computation. As an example, two adder units in a computer can simultaneously allow an address calculation and an instruction execution. The classic example of parallel operation is the parallel processor, which is suited to problems such as radar data processing, where a single computation is desired on many unique data sets. In this case, a number of identical hardware units, N (where N is equal to the number of data sets), can be used to realize the results nearly N times faster than a sequential machine.

Simultaneous execution of multiple processes is exemplified by the multiprocessor or multicomputer system. In such a system, the tasks to be computed are distributed to the various system resources, on a priority or time-based criterion, for computation. As with the parallel processor, the throughput of the system is the product of the number of processors and their effective computation rate.

Prior to LSI, the use of parallelism was greatly limited. In today's technology, the highest levels of integration can be achieved with relatively slow MOS circuits. In many cases, the speed of MOS is not sufficient to replace existing equipments unless architectural changes are made. Parallelism, as discussed, provides a way to take advantage of MOS and to achieve high computational rates. With parallelism, one can consider

developing hardware units of a very limited capability to achieve high system throughput by parallelism. The advantage is the ability to configure a system from existing modules to more exactly match system requirements. Increased requirements can be met by increased parallelism in small capability increments.

A system designer, when considering parallelism in his system, must first define the optimum basic computational unit and then define how these units are to be configured into a system. The steps that are to be followed include:

Unit

- Consider computation on a word or byte basis rather than a bit basis.
- Determine if portions of basic computations (instructions) can be computed simultaneously and economically by including additional functional blocks.

System

- Factor total task load into independent subtasks.
- Determine if nonindependent subtasks can be made independent on a time basis (predecessor relationships).
- Consider replicating hardware to compute independent and time-wise independent tasks.
- Maximize use of identical hardware units.
- Determine the cost of controlling hardware and task scheduling.

The design of any avionics system using LSI should be accomplished with the concept of parallelism in mind. Basic guidelines for the use of parallelism should include the following considerations:

- Use additional functional blocks to process independent portions of a basic instruction simultaneously to increase unit speed.
- Use multiple copies of the basic unit to increase system speed and to reduce response time.
- Use parallelism to provide incremented growth capability.
- Use parallelism to allow the use of low-speed MOS in high-speed systems.
- Use parallelism to allow fault tolerant operation and increase reliability.
- Use parallel units for problems where identical computations are required on many data sets.
- Use parallel arrangements of low capability devices to increase usage of custom LSI chips.

Appendix B

Increased Use of Modularity

In the past, avionics systems have been characterized by custom designs and standard black-box approaches. Each subsystem used a set of unique functional units and each system application (airplane) used primarily different units. The custom development approach to system realization resulted in very high development costs and long development schedules. The standard black-box approach usually resulted in a capability overkill, which, with standard LSI implementations, entailed a high size, weight and power penalty. Neither of the two approaches provided any flexibility to meet increased capability requirements unless it was built in at the beginning of the program. Due to the diversity of avionics equipment, production and maintenance techniques were only partially learned for any one equipment.

Defining a set of modular building blocks that are expandable, general in purpose, and realized by LSI technology would greatly benefit avionics systems. Such approaches could significantly reduce the cost and response time of new avionics developments.

The introduction of standard microprocessors on a chip such as the Intel 8080 has provided the system designer with a powerful new approach to modularity. Ruggedized versions of such processors could serve as controllers or dedicated preprocessors in avionics subsystems.

LSI favors the concept of modularity due to the ease of including added logic for multipurpose usage and the ability to tailor a general-purpose device through the use of a read-only memory (ROM). The high nonrecurring cost of custom chip developments, coupled with the very low per-gate recurring cost at high production volumes, shows that the more a custom chip is used, the lower the overall system cost will be. The concept of modularity at the micro level allows the designer to minimize the number of unique custom parts, thus reducing the nonrecurring cost of LSI development.

At the macro level, modularity can provide parallel computation and functional growth by adding identical hardware blocks to implement a subsystem. A truly general-purpose module can be used in a number of subsystems within the same avionics systems, as well as in a number of unique avionics systems.

Past attempts at modular approaches to avionics systems have been largely unsuccessful for the following reasons:

- The technology of conventional IC implementations had a large size-weight-power to computational capability ratio; thus, the designs had to be gate-minimized and highly optimized to achieve the speed requirement, while remaining within other physical constraints.
- There was excessive programming overhead of general-purpose (GP) implementations of the special functions. Software overhead, program storage limitations, and program development time favored special-purpose hardware design rather than a specially programmed general-purpose unit.
- Multipurpose modular equipment had low reliability. The added packages and interconnections required to increase the usefulness of a module significantly reduced the overall systems reliability while increasing size and weight.
- Standardization of interfaces was difficult with analog devices and was compounded by themany subsystem black-box contractors.

In the past these problems tended to limit the use of the modularity concept in the design of avionics systems.

Appendix B

With LSI, there is an increased need for modular multi-usage equipments to meet the ever-changing avionics requirements. LSIC technology has provided the following advantages for system modularity:

- A drastic reduction in the size, weight, and power consumption for a given capability; thus, the designer is not gate-limited in developing multipurpose modules.
- Combining cheap ROM and microprogramming efficiently tailors general-purpose devices to special-purpose jobs.
- With LSI, the additional logic required to make a module multipurpose has little impact on system reliability

Use of common modules could be greatly expanded by increasing the commonality of performance and interface specifications between avionics systems. The avionics designer should consider the following guidelines concerning modularity:

- Use existing LSI equipment wherever possible.
- Use more powerful standard equipment than required, if cheaper than custom development.
- Convert general-purpose equipment to special purpose through micro-programming.
- Design custom equipment to be modular.
- Design custom equipment to be applicable to other systems or subsystems.

Microprogramming

Introduction. - From an implementation point of view, microprogramming is a way of removing random control logic and of replacing it with memory-oriented units. This results in a loosely bound grouping of functional units such as registers, ALU's, memories, shifters, and other special logic units. The microprogram or firmware is then used to define the structure of the system and the data paths through these functional units. Thus, the general-purpose architecture can be optimized for special-purpose problems.

This technique has made a significant impact on computer architecture in the past several years due to the availability of LSI ROM's. From a systems architect point of view, there are several reasons to use microprogramming:

- Reduced random logic, resulting in a more structured organization
- Efficient emulation of many special-purpose devices by one general-purpose LSI module
- Simplified software programming through macro instructions
- Simplified diagnostics and maintenance aids

The cost, flexibility, and speed advantages of firmware make it an excellent compromise between software and hardware implementations. The following subsections will discuss firmware in light of these advantages.

Microprogram flexibility. - Flexibility is probably the most important advantage of microprogramming over hardware implementations. The design is not frozen by the existence of hard-wired connections; thus, it is possible to reconfigure by simply overwriting the control memory. Specific advantages of microprogramming include:

- Provides adaptation to new applications without redesign of hardware
- Provides for an increased hardware capability without costly redesigning
- Provides dynamic reconfigurability for increased reliability

Microprogram cost advantage. - LSI has increased the feasibility of microcode implementations by reducing the cost-per-bit of ROM; thus, replacing random control gates and their associated interconnects reduces design and production costs. Because ROM's are approximately six times more efficient in using chip space than are random logic controllers, significant production costs savings can be achieved using microprogramming techniques (ref.B3). Table B1 shows the power and space savings of a ROM implementation (ref. B4).

TABLE B1. ROM SAVINGS IN MICROPROGRAMMED CONTROL

	BITS PER GATE FUNCTION		
	8 Optimistic	12 Realistic	16 Pessimistic
Number of Dips Saved	170	114	85
PC Cards Saved 60 DIPS/Board	3	2	1.5
Power Saved	12.8W	8.5W	6.4W
Volume Saved	150 in^3	100 in^3	75 in^3

Through the use of microprogramming, one hardware device can be used for a number of different applications, which boosts production runs and lowers system costs. This advantage has been discussed in the modularity section and is especially true of the new generation of microprocessors.

Another cost advantage of microprogramming is the reduced cost of producing software. Microcode allows software to be written in a more macro language and allows direct implementation of special functions. Alternately, microcode can provide an emulation capability to allow "old" machine software to execute on "new" machine architectures without revision.

Microprogramming for avionics systems. - Cost, flexibility, and speed make firmware an excellent compromise between a software and hardware implementation for avionics systems. Placing such things as monitoring functions and debugging aids in the microcode significantly reduces operational software costs. Firmware, because it controls the data paths and functional blocks, can be used to implement the housekeeping functions required to support complex operating systems and executives.

Microcode has the advantage of being separated from main memory and protected from unauthorized change. This means that there is no physical way a user can destroy vital algorithms and status, as in the case of software implementations. Thus, it is well suited for implementing system level functions.

- Reduce random logic and expensive custom chip design by using standard LSI ROM's
- Increase flexibility of functional modules
- Define macros that simplify software development tasks

Use of Hardware to Perform Traditional Software Functions

Introduction. - Avionics computing systems are continually being relied upon to perform increasingly complex system functions. This increased complexity has drastically increased the cost of applications software. Estimates of the software-to-hardware cost of a total system range from 4 to 1 through 1 to 1, depending on system complexity. This ratio is constantly widening as LSI technology reduces the cost of hardware (5 to 25 percent per year) while software development costs continue to rise. These considerations, coupled with the increased capability, complexity, and speed requirements of avionic computing systems, have hastened the replacement of traditional software functions by hardware. The new technological developments described now make this transition feasible and economical.

During this transition, a gray area has developed between the gate hardware implementation and the instruction word software. This gray area, which has been termed firmware, has recently been emphasized to enable equipments to operate at the macro instruction level. Thus, hardware and firmware have been used both to replace and to enhance some of the traditional software functions.

This transition of software to hardware is especially significant for avionics systems when volume is a consideration. Software represents a nonrecurring cost that can be made less significant by spreading it over large production runs. In avionics systems, however, the production potential is low, although complex software costs remain high. Additionally, the application software is growing in complexity, which increases the debugging problem and cost exponentially. For example, an avionics guidance and control system with a software package of 12K words complexity has 50 percent of the total system's cost in software. The Space Shuttle guidance and control software package of 64K words represents 80 percent of the per system cost. Thus, the implementation of software in hardware represents a great potential benefit to avionics systems.

There are basically four ways to implement software functions in hardware:

- Random logic
- Table look-up
- Microprogramming
- Dedicated microprocessors

Random logic. - This method of implementation is a true hardware approach as contrasted to the firmware approaches that will be discussed. It is the least adaptable and also the least amenable to LSI, but generally is considered the fastest of the approaches. Except for universal logic modules or programmable logic arrays, this is a custom

approach that requires some volume production to be cost effective if implemented in LSI. Functions that are often implemented in gates are:

- Floating point arithmetic
- Multiply and divide
- Square root
- Fast Fourier transform
- Priority decoding-encoding
- Coordinate transformations

Hard-wired implementations are most economical when the function is used frequently and speed is a critical requirement. For example, a Honeywell computer could execute a square root instruction in 25 to 30 μsec with a software routine; the inclusion of 64 gates of special hardware allowed this computation in 10 μsec.

Table look-up. - A table lock-up implementation is the next closest thing to the hard-wired approach. It has become increasingly attractive to the designer due to the availability of low-cost ROM and electrically alterable ROM. It is economical and amenable to LSI, while providing speeds almost equal to that of hard-wired gate implementations. The use of table look-up has been used for some of the following functions:

- Multiply-divide
- Code conversion
- BCD arithmetic
- Character generation for displays
- Trig functions
- Transducer calibration

At present, a number of these functions are available off the shelf. This is a clear advantage for low production systems as no LSI chip design is required.

A good example of the feasibility of the table look-up implementation is described by Texas Instruments (ref.B1) in their implementation of an 8-x-8-bit multiply. Using a novel hybrid hardware and ROM approach, the multiply can be accomplished in 76 nsec, using only 8K bits of ROM. It is implemented with all standard off-the-shelf MSI and LSI parts, and could be expanded to do 16-x-16-bit multiply using 33K bits of ROM in 150 nsec.

Microprocessors. - Due to LSI technology, a new type of building block has been introduced. Microprocessors are the high-level asynchronous functional boxes, like the Intel 8080 which was discussed in the processor section. They can be considered as modularly parallel functional units that efficiently execute high complexity functions in hardware while not requiring a custom chip. They are now available as off-the-shelf parts, thus providing a library of hardware subroutines to be used by the system designer.

Some functions which could be implemented using microprocessors are:

- Memory pagers
- Vector multiply

- Matrix multiply
- Fast Fourier preprocessing
- Digital filtering
- Output formatting

This implementation can be cost-effective for small production runs or developmental programs of avionics systems. It also has the advantage of being modular, and thus applicable to many avionics systems without having to redesign either the hardware or software.

An application example is communication error processors. This microprocessor would asynchronously monitor and form at the error coding of incoming and outgoing block transfers. If an error occurred, it could initiate a retry. If continuous errors occurred, it could then interrupt the CPU.

Guidelines for the use of hardware to replace software. - For each application, consideration must be given as to which functions to implement and which implementation is best. Basic considerations are:

- Use hard-wired gates where high speed and/or function complexity is required.
- Use table look-up techniques for less complex functions.
- Use microprogramming where recursive or repetitive steps of an algorithm can be identified.
- Use microprogramming where speed is not critical.
- Use microprogramming where compatibility across a family of devices is required.
- Use microprocessors to perform dedicated parallel functions.

Increased Fault Tolerance

Introduction. - Avionics systems have reasonably stringent fault tolerance/reliability requirements. LSI provides many cost-effective means of increasing systems' reliability within size, weight, and power constraints.

While systems grow linearly in complexity, the testing and maintenance problems grow exponentially. Even for ultrareliable systems, there exists a need for hardware aids for testing and fault diagnosis. The reduced cost per gate of LSI can provide, on many levels, aids that simplify these tasks.

Two characteristics of LSI improve fault tolerant systems. The first is the decreased cost per gate of logic. For a given pin-limited partition, there is often a significant amount of silicon area available which effectively gives "gates for free." The second characteristic is the reduced number of pin-outs or physical connections. The predominant mode of failures for electronic devices is contact or connector failures. Using LSI and thereby reducing the number of pins in the total system, increases reliability.

Circuit consideration. - Due to the reduced cost and reduced physical connections, it is now feasible to use added logic to increase the reliability of a logic function. Dual gate redundancy techniques can be efficiently used for fault detection. Triple gate redundancy can be used for fault masking. These techniques, which were previously very costly in terms of size, weight, power, reliability, and dollars, are amenable to LSI due to their regularity.

A separate fault detection circuit can often be included in the chip without any penalty. These circuits (see Figure B1) can continuously monitor for circuit malfunctions and can inhibit faulty outputs. Alternately, it could be used in a testing mode where it would help diagnose a complex LSI circuit during system tests.

Another technique enhanced by LSI is the replacement of random gates by microcoded ROM. This implementation of control logic increases reliability because ROM is more reliable than random gates. Furthermore, the more orderly microprogrammed implementation generally requires fewer interconnects, which also increases reliability of the overall control circuits.

LSI ROM's make error-correcting codes a more feasible fault tolerant technique. Small encoders and decoders for code conversion can be included on the chip at little cost in silicon area or execution speed. These codes can be used internally (i.e., residue code for arithmetic operations) or externally (i.e., hamming code for data transmission).

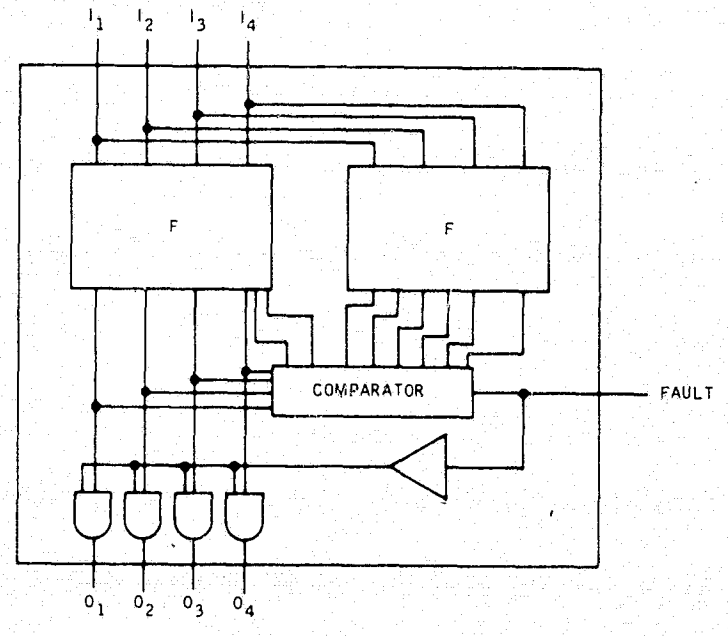


Figure B1. - Example of Fault Detection

Microcode considerations. - Microdiagnostics provide a real aid to built-in test and system production testing. By dynamically performing error checks, fail-safe operations can be achieved by dynamically changing algorithms to bypass hardware components that have failed. Furthermore, microdiagnostics depend on fewer operational hardware circuits and, therefore, can be executed on a smaller, error-free hard core. It has been estimated that the IBM 360/30 reduced its hard core from 50 percent to around 10 percent. This means that more failures can be detected and, more importantly, can be overcome. Because the microcode is directly in control of all the system hardware, microdiagnostics can isolate failures with higher assurance to a smaller unit.

Once diagnosed, microcoding can often provide an alternate execution using slower, but operational hardware. For example, in the IBM 360 85 there is a low-speed multiply algorithm in the microcode that can be used when a failure is detected in the high-speed multiply hardware (ref. B5).

Systems considerations. - On a higher level, LSI savings of size, weight and power, as well as cost, enhance the use of functional redundancy. Where functions can be incorporated on a single chip, dual and triple modular redundancy can be implemented cheaply and easily. By combining stand-by redundancy with microprogrammed fault detection and reconfiguration, an effective fail-safe/fail-soft system can be designed.

Microprocessors provide a new tool that can economically be used to increase system reliability. System configurations with multiple paralleled microprocessors provide a high availability, as well as a graceful degradation. As described before, microprocessors can be dedicated to monitoring for system errors to provide a performance evaluation in real time.

Many subsystem functions that have previously been implemented using analog and electromechanical devices can now be implemented in digital (ref. B6). Replacing these older implementations with LSI has vastly increased the subsystem's reliability. Strapdown or laser gyros are a good example of the increased reliability of electronic implementations over conventional implementations.

Summary. - LSI provides new methods of increasing a system's reliability. LSI's enhancement of modularity and reduced interconnections have also had a significant indirect effect on the system architecture, while increasing its overall reliability. Some general considerations to improve system reliability would include:

- Use ROM's for code conversion on the chip.
- Use microprogramming to organize control and to reduce random gates.
- Use microprocessors to monitor critical functions.
- Use extra silicon area for fault-detection circuits.
- Use modular redundancy within systems to provide fail-safe operation.
- Use functional redundancy (i. e., TMR and voting) to provide reliable modules.
- Use dual redundant gates for fault detection in critical circuits.
- Use triple redundant gates with voting to mask failures.

Appendix B

Reduced System Interconnections

Introduction. - Avionics systems are becoming increasingly complex interactive systems. With the increase of functional block interdependency comes a corresponding increase in communication. The traditional after-the-fact, ad-hoc method of connecting units together is not a satisfactory solution. Considering reliability and packaging, point-to-point wiring in such complex systems cannot be permitted. Thus, for complex system architectures, a new method of bussing is required to reduce the number of connections and the volume of data transfers.

LSI systems provide several new possibilities in system interconnect architectures. While LSI suffers from pin-out limitations, it provides solutions to better minimize system interconnections. Many of these concepts are discussed below, and some additional ideas are presented in the partitioning section.

Chip considerations. - LSI has increased the amount of logic on a chip to a point where whole functions with their supporting control and buffers can be contained on one chip. This reduces the number of package-to-package interconnects to a minimum. Additional gates can be included to form data buffers, which allow communication on a nondedicated (time division multiplexed) system bus.

Multiplexers can be included on the chip to allow the same nondedicated bus to provide data, control signals, or even diagnostic information through the same set of pins. Control signals can be encoded and decoded on the chip to further reduce the required transmissions.

The complexity available on a LSI chip can make serial transmission of data an efficient alternative. For example, consider a LSI chip that requires 350 nsec to operate on 8 bits of data. Parallel data transmission would require 8 input lines on the chip. Serial transmission at a rate of 40 nsec/bit would allow a savings of 7 I/O ports. The chip would require an eight-bit serial in parallel out shift register to feed the holding register. Next, instruction data would be input during the current instruction execution. Thus, an 8-bit shift register could be traded for 7 pins, with a resulting reliability and overall chip partioning advantage.

Using microprogramming reduces the number of main memory instruction transfers. This reduces the contention for main memory bus access, which is often a bottleneck in systems' throughput. In view of the hierarchical instruction execution for parallel processors previously described, this can significantly reduce bus transfers.

LSI, through the enhancement of distributed cache memories, has also reduced the contention for the main memory data access. The localization of data path loops to a single chip has allowed for the concurrent parallel operation of several units without interference. This not only increases the overall system thoughput but also enables the use of nondedicated busses. The latter enhances the system modularity and its attendant advantage - expandability.

Avionics system considerations. - LSI microprocessors, working as asynchronous functional units, efficiently reduce communication between avionics subsystems. By preprocessing sensor data, a minimum amount of critical data must be transmitted to the necessary subsystems. This results in two advantages: 1) an increase in number of sensor devices that can be served by that bus, and 2) a reduction in the computational capability and complexity in the subsystem requiring data from that sensor. A simplified centralized computer can, therefore, service many subsystems without having to interface with each sensor output. Its software complexity is reduced and modularity is enhanced due to the all-digital interfaces and the reduced amount of data flow.

Appendix B

LSI has made feasible a modular interface unit (IU), which can reduce the overall system overhead involved in communicating on a nondedicated bus. These IU's are standard modules that can distribute the control throughout the system, thus increasing the reliability of the overall system. Furthermore, these standard IU's are a high-usage module that can automatically buffer devices of vastly differing characteristics onto one common bus.

In future avionics systems, the number and width of data busses as well as the amount of data transfers must be minimized. Some LSI considerations which could be used by the designer are:

- Use microprocessors at the data entry point, allowing only important nonredundant data to be transmitted.
- Use added logic to encode and decode control-command transmissions.
- Use serial transmission to minimize pins where execution time permits.
- Use nondedicated busses to increase system modularity.
- Use distributed control in bus interface to minimize overhead while increasing reliability.
- Use the multiplexing of pins for input, output, control, and testing.

Summary of Architectural Guidelines

The preceding discussion has shown that many new approaches to defining system architecture are possible when LSI is to be considered in system implementation. It has also been stated that some combination of these approaches is usually necessary to optimize the suitability of a system to LSI. Because many guidelines are presented, the designer must make a judicious selection of those applicable to a given system.

A summary of the architectural guidelines for the efficient use of LSI in avionics systems is presented below (each should be considered in terms of system requirements).

Replace Analog Hardware by Digital

- Convert inputs from analog to digital as early as possible.
- Convert outputs from digital to analog as late as possible.
- Consider new algorithm developments to optimize digital implementation.

Use of Semiconductor (LSI) Memories

- Use small RAM's to reduce main menory requests (scratchpads, caches, stacks, queues, and registers).
- Use ROM's for microprogramming.
- Use ROM's for table look-up.
- Use ROM's for hard-wired subroutines.
- Use CAM's to eliminate file searches.

- Use queues to buffer data transfers.
- Use queues and stacks to minimize executive overhead.

Use of Parallelism

- Use additional functional blocks to simultaneously process independent portions of a basic computation.
- Use multiple copies of basic hardware units to increase system speed and to reduce response time.
- Use parallelism to provide incremental growth capability.
- Use parallelism to allow the use of low-speed MOS in high-speed applications.
- Use parallelism to allow fault tolerant operations and to increase reliability.
- Use parallelism where identical computations are required on multiple data sets.
- Use parallel arrangements of low-capability devices to increase use of custom LSI chips.

Use of Modularity

- Use existing LSI equipment where possible.
- Use standard LSI equipment with excess capability to avoid custom development.
- Convert general-purpose equipment to special-purpose equipment through microprogramming.
- Design each new equipment to exhibit modularity and commonality wherever possible.

Use Microprogramming

- To reduce random logic and custom chip design using standard LSI ROM's.
- To increase flexibility of functional modules.
- To define macros to simplify software development.

Use of Hardware to Perform Functions Traditionally Done in Software

- Use hardware gates to perform special arithmetic functions.
- Use table look-up.
- Use microprogramming for recursive steps of an algorithm.
- Use microprocessors to perform traditional software functions.

Increase Fault Tolerance and Reliability

- Use ROM for code conversion on the chip.
- Use microprogramming to organize control.
- Use microprocessors to monitor critical functions.
- Use extra silicon area on a chip for fault detection circuitry.
- Provide redundant gates on a chip to increase reliability.
- Use functional redundancy to provide reliable modules.
- Use modular redundancy to provide fail-safe operation.

Reduce System Bussing

- Use processors at data entry point; transmit only nonredundant data.
- Transmit data serially where speed permits.
- Use nondedicated busses (time division multiplexing).
- Distribute bus interface control.
- Use same busses for input/output and control.

Following these guidelines will not only allow a more cost-effective implementation using LSI, but also will improve other important system parameters.

REFERENCES

B1. The Semiconductor Memories Course Workbook, William Carr, Texas Instruments, 1972.

B2. K. J. Thurber, R. O. Berg, "Universal Logic Modules Implemented Using LSI Memory Techniques," Proceedings, AFIPS 1971, Fall Joint Computer Conf., Las Vegas, Nevada, November 1971, pp. 177-194.

B3. G. W. Schultz, "MOS LSI Minicomputer Comes of Age," Proceedings, FJCC 1972, pp. 1069-1080.

B4. W. H. Davidow, "General Purpose Microcontrollers, Part I: Economic Considerations," Computer Design, July 1972.

B5. System/360 Model 85 Microdiagnostics, N. Barton, R. McGuire, AFIPS SJCC, 36, 1970.

B6. Roger A. Urban, "An Airborne Associative Array Processor," Proceedings NEC, Volume XXVII, October 9-11, 1972 pp. 318-321.

APPENDIX C

RELIABILITY OF LARGE SCALE INTEGRATED CIRCUITS (LSIC)

Introduction

One of the most widely promised attributes of LSI is that of increased reliability. Industry has moved from the vacuum tube to the transistor to the integrated circuit to increase reliability. There are many reasons to believe that the move to LSI should also be accompanied by increased reliability: systems using LSI will consist of fewer components and will require fewer system interconnections. Thus, unless the chips themselves are more unreliable than the chip reduction factor plus the system wiring improvement, LSI should result in a more reliable system.

Determining the reliability of LSI devices is very difficult, and sufficient field data is not available. The problem is complicated by the number of available technologies and approaches to LSI, and by the different chip sizes, device count, and pin-outs used by each unique chip. Data that is available on LSI devices are taken at different temperatures and employ different pretest screens, further complicating evaluation of reliability.

Determination of failure rates of LSI chips is very costly and time consuming. For example (ref. C1), to ensure a 0.1% 1000 hr failure rate with 90% confidence, 2303 devices would have to operate for 1000 hours (42 days) without a single failure. If one failure was allowed, 3891 devices must be tested. To ensure a failure rate of 0.01%/1000 hr, the number of devices would be 23,000 and 39,000, respectively. The number of devices can be reduced by lowering confidence levels, increasing operating stress, and increasing the time involved; however the problem is formidable for custom LSI chips.

Thus, using custom LSI on low-volume avionics systems is not compatible with standard life tests to determine reliability. The number of devices and the time required to establish the reliability of a device would extend the development cycle and perhaps exceed the total volume of chips required to implement the systems.

Attempts to determine the reliability of a process or a technology would be of more value, yet they are changing at such a rapid rate that by the time reliability assurance is obtained, the processes would have been changed, and hopefully improved. For example, the standard metal gate approach to MOS LSI has been expanded to include silicon gate and ion implantation in a very short period of time.

The importance of technology maturity is illustrated in Figure C1, which shows that the normalized failure rate of a technology decreases by a factor of two after one year in production. After two years, the failure rate has decreased by a factor of nearly 10. This figure was supplied by Hughes Aircraft.

Measured and Estimated Reliability

Table C1 lists reliability data, measured and estimated, for IC's of various technologies and complexities. The first two entries show marked improvement in Motorola MOS over a one-year period. Collins Radio estimates a reliability improvement of 2.5 after 1000 hours of operation. Motorola shows a 20-to-1 reliability improvement with device screening, and a 50-to-1 reliability increase of beam leads over conventional packaging for bipolar SSI parts. For a bipolar 150-gate array, they predict a 10-to-1 improvement with 168-hour burn-in. The same improvement is expected for MOS LSIC's.

Appendix C

Reliability Characterization of Device Alternatives

The device technologies from which the designers have to choose are numerous. From the major divisions of bipolar and MOS, subdivisions such as transistor-transistor logic (TTL), current mode logic (CML) and diode-transistor logic (DTL) can be made for bipolar and P-channel, complementary or N-channel subdivisions can be made for MOS, among others. The latter may be further divided into high voltage, low voltage, thin oxide, ion implanted, silicon gate, etc.

From a reliability standpoint, some failure modes and causes are common to all types, while others are peculiar to a particular family. Overall failure rates reflect this mixing and other influences such that failure modes overlap from type to type and the types are not uniquely separable on the basis of failure rate. Moreover, our interest here is not in any one particular type of failure, but rather in the level of its complexity and reliability as designated by small-scale integration (SSI) or large-scale integration (LSI). Thus, rather than extremely precise estimates, we will need representative failure rates for these general categories to determine reliability of those systems using them.

For purposes of discussion, an SSI will be defined as containing from one to 12 gates on a chip (typically five), 14 leads, approximately 40 mils square and using single-layer metalization. A bipolar LSI device will be defined as containing up to 150 gates, up to 60 pins, approximately 150 mils square, and requiring multilayer metalization. A MOS LSI device will be defined as containing 500 gates, 40 pins, and single-layer metalization. The pin reduction reflects partitioning advantages of the added complexity. The chip size will be approximately 150 mils square. A hybrid device shall be defined as containing typically 49 SSI beam lead chips and 60 pin-outs from a 2 by 2 inch ceramic multilayer substrate. The functional nature of the relationship between these characteristics and reliability is typified by the Equation given in the proposed MIL-HDBK-217B or the RADC Reliability Handbook:

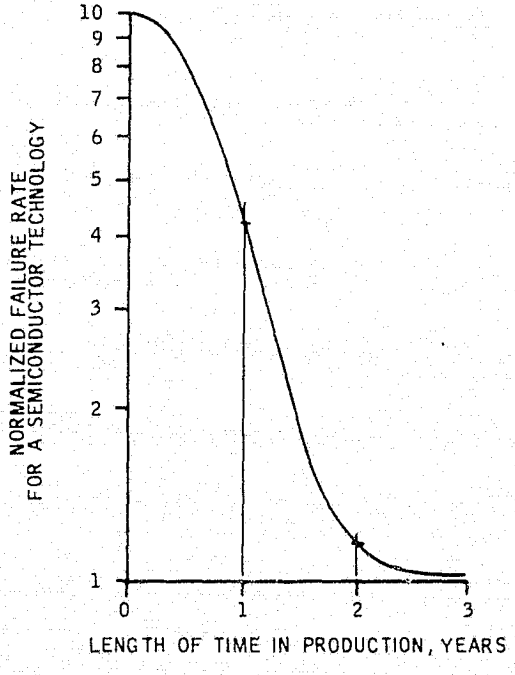


Figure C1. Change for Normalized Failure Rate for a Semiconductor Technology versus Length of Time in Production

TABLE C1. – RELIABILITY DATA

Company	Technology	Temp.	Device Hours	Complexity	%/1000 Hours Failure Rate	Data Source
Motorola	PMOS/LSI	85°C	338.7K	Dual 100-bit shift register	0.91	May 71 Rel. Report. SPD 9928.
Motorola	LSI PMOS-NMOS	125°C	1.5M	LSI memories	0.54	Aug 72 Rel. Report 7235-2.
Motorola	LSI PMOS-NMOS	75°C		Shift registers and logic arrays	0.22*	Aug 72 Rel Report 7235-2.
Honeywell	PMOS LSI	95°C(A)		64x4 RAM 256x12 ROM	0.68*	Honeywell 0972-11188 (9/72).
	PMOS LSI	125°C 50°C		600 Transistors 600 Transistors	0.1-1% 0.02-0.2%	G. L. Schnable, "MOS IC Reliability," IEEE Trans on Rel., Vol. R-21, No. 1 Feb. 72, pp. 12-19.
	MOS			300 Gates	≈ 0.01	R. B. Conn. "Design of Fault-Tolerant, Modular Computer with Dynamic Redundancy," Proc. FJCC 1972, pp. 1057-1067.
Collins	MOS		10^6		(new sys) 0.18 (after 1000 hrs) 0.072*	Working paper 2399, July 1972.
Honeywell	Bipolar MOS Bipolar MOS			LSI 40 pin SSI	0.022* 0.01	Honeywell Aero part failure rates.
National	MOS	45°C			0.001	Questionnaire data.
Motorola	CMOS Bipolar LSI	125°C 125°C(A)	1.2M	150 Gates	0.35 (after burn-in) 0.10*	Rel. Report 7220-1B Personal communication.
Hughes	Bipolar whole wafer			616 Gates	0.292	Questionnaire enclosure data.
Motorola	Bipolar SSI Bipolar SSI			SSI Beam lead SSI SSI with screen Beam lead SSI with screen	0.05 0.001 0.001 0.00005	Beam lead fact book. Beam lead fact book. Beam lead fact book. Beam lead fact book.
Motorola	PMOS MSI-SSI	75°C 125°C	1.89M	2 SSI - 2 MSI	0.105	May 71 Rel. Report SPD 9928.
Motorola	PMOS-NMOS PMOS-NMOS	125°C 75°C	3.5M	SSI SSI	0.12 0.04*	Aug 72 Rel. Report 7235-2. Aug 72 Rel. Report 7235-2.

*Estimated.

(A) Ambient Temperature.

Appendix C

$$\text{Failure Rate} = \lambda(\pi_C \pi_P \pi_E \pi_Q) \text{ for monolithics} \quad \text{(C1)}$$

where

λ = a base failure rate

π_C = a complexity adjustment factor

π_P = adjustment for package type

π_E = an environment adjustment

π_Q = adjustment for quality grade

or, for hybrids,

$$\text{Failure Rate} = (\Sigma \lambda_D + \lambda_S + \lambda_T + \lambda_P)\pi_E \quad \text{(C2)}$$

where

$\Sigma\lambda_D$ = the π_C for discretes using the total active area of all chips

λ_S = substrate failure rate

λ_T = failure rate for thick- or thin-film network or substrate

λ_P = package failure rate

π_E = environmental service factor

When comparing monolithics, π_E and π_Q along with λ, essentially divide out, leaving complexity and package as the main differences.

Package adjustments in both RADC and 217B seem to be influenced by package material and type of die attach rather than number of pins or wire bonds or circumferential length of seal. Both documents consider complexity in terms of gates, but not for very large number of gates or more than one layer of metal.

Various refinements have been made in the models to remedy these shortcomings. There is a circuit size and complexity factor, π_c, given by

$$\pi_c = 0.5 + 0.5\,(A/5000)^{0.8} \quad \text{(C3)}$$

where A is the active area of the chip in square mils. Assuming that the outer 10 mils of a chip contain bonding pads, the preceding definitions give the following factors:

SSI π_c = 0.57

LSI π_c = 2.17

Hybrid π_c = 3.59

The package factor, π_p, is given by

$$\pi_p = 1 + 0.05L \text{ for chip and wire packaging.} \quad \text{(C4)}$$

Assuming beam leads are better by a factor of 4, it yields

$$\pi_p = 1.0 + 0.0125\,(L) \text{ for beam lead chips} \qquad (C5)$$

where L is the number of leads in excess of ten. Adding 1 to π_p in the case of hybrids for the ceramic substrate and interconnections, π_p becomes

SSI π_p = 1.2

LSI π_p = 3.5 bipolar
= 2.5 MOS

Hybrid π_p = 2.62

Taking the product $\pi_p\ \pi_c$ for each type, and then taking the ratio of the result with respect to SSI, shows that (according to our characterization of SSI, LSI and hybrid) bipolar LSI has a failure rate approximately 11 times SSI, MOS LSI has a failure rate 8 times that of SSI, and hybrids have a failure rate 14 times that of SSI. Therefore, on a per-gate basis, the relative reliability of the various implementation levels becomes:

Implementation	Complexity	Rel. Rel./Gate
SSI	5 gates	1
Bipolar LSI	150 gates	2.72
MOS LSI	500 gates	12.50
Hybrid SSI	245 gates	3.5

Systems that are highly digital should exhibit the reliability improvements shown for LSI, provided that the complexity used matches the assumptions and that no attempts have been made to increase reliability through on-chip redundancy for the monolithic LSI approaches.

System Production/Deployment Reliability

Side benefits from progressively reducing part count and complexity are decreased weight and increased freedom for the mechanical designer to improve structural strength and thermal design, which further improves reliability. On one Honeywell SSI design, the desired thermal conditions were not achieved due to weight and volume restrictions. Had LSI been used such that the needed weight and space were available for cooling purposes, reliability could have been increased by approximately 10 percent due solely to the temperature decrease.

The multilayer boards are decreased not only in quantity but also in the number of layers and the number of plated through-holes. The latter produces higher per-board reliability.

The reduction in system complexity achieved when using LSI benefits the production line also, as handling damage is proportional to the number of modules built (given constant fragility).

The testing/logistics questions associated with LSI systems are interesting. As far as factory test of the LSI system is concerned, the same sophistication is needed as for SSI; yet fault isolation to a board in the LSI system should be much easier than in the SSI system. Extensions of this idea might be that boards become line replaceable units

(LRU's) rather than black boxes, and more elementary depots could be established nearer the front lines to improve availability. Cost of ownership for the armed services could be significantly affected.

Basic to all reliability programs is the requirement for a capability in failure analysis. Where possible, it is desirable to have the basic capability at the system house for quick concerted action. A semiconductor vendor is much more receptive to taking corrective action when proof of failure mode and mechanism is presented to him. Unfortunately, LSI failure analysis often requires a sophistication that strains the capability at the systems' house, particularly for degradation-type failures. Due to the lack of standardization of LSI, failure analysis results at one company are often not applicable to the problems of another company.

Standard part approval, although an expensive and time consuming procedure, is an indication of reliability. While the mere listing itself does not guarantee high reliability, the procedure through which the part goes to become standard forces some degree of production maturity. The operating life and environmental testing done during qualification usually uncovers the initial part design/production incipient failure mechanisms. If a problem occurs, the program will not only suffer a costly correction of deficiences in equipment already produced, but will also be faced with a schedule slip to keep from fielding more problem devices. The slip continues until failure analysis/redesign can determine corrective action.

Another advantage of standardization is that the larger the quantity of orders the vendor receives, the more likely that his production lines will continue working without interruption. The number of disruptions is directly proportional to the number of production errors and inversely proportional to reliability.

The observations described are not new; they were basic considerations during the initial organization of the standardization program. The important point is that they are recognized by systems designers today and that LSI is presently operating without their benefit.

Maintenance Cost Analysis

For most avionics systems, maintenance costs over the lifetime of the system can be many times the initial acquisition costs involved. Results of previous studies have in fact indicated that cost ratios can run as high as 10 to 1. For this reason, it becomes increasingly apparent that system implementation decisions should be made in light of their impact upon total life-cycle costs, including both acquisition and maintenance costs.

A maintenance cost analysis will now be performed for several implementation approaches, assuming a simplified but representative maintenance procedure for avionics systems. Three maintenance procedures will be considered in this analysis:

1) Discard-Upon-Failure at the System Level

2) Discard-Upon-Failure at the PC Board Level

3) Repair-Upon-Failure at the PC Board Level by Replacing Subpackages

In the first approach, maintenance is achieved by replacing the total system/subsystem after each failure. This approach is only realistic for less complex systems.

In the second and third maintenance approaches the PC board is considered as the basic repair/replacement element. This is a realistic maintenance procedure for avionic systems of general complexity. In each of these approaches, maintenance costs are predicted by using a simplified maintenance cost model, which is shown in Figure C2. This model has been generalized from AFLC/AFSC Manual No. 800-4,

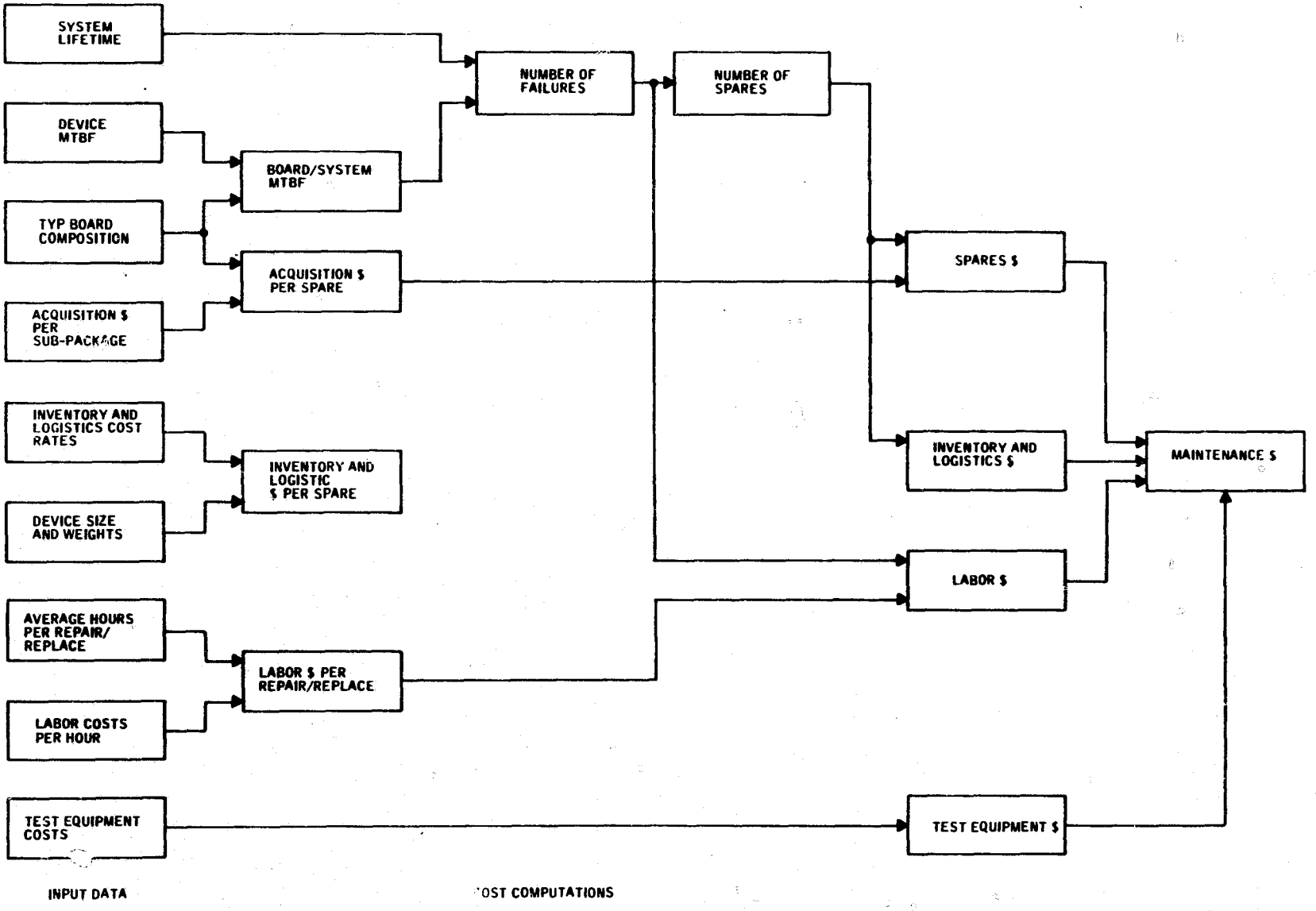


Figure C2. – Simplified Avionics Maintenance Model

entitled "Optimum Repair Level Analysis." In the first of these approaches, maintenance is performed by replacement at the PC-board level when failure occurs. Elements of cost using this approach include:

- Acquisition cost of spares
- Diagnosis and replacement labor costs
- Spares inventory and logistics costs
- Test equipment for a single repair location

With this simplified model for board-level replacement, a number of redundancy cost elements are ignored, including maintenance publications, redundancy of test equipments for multiple repair sites, and the actual cost of acquiring and maintaining the repair facility itself. In so doing, it has been assumed that each of these cost elements are essentially constant for each electronic package implementation approach. Thus, the objective here is to establish the relative relationships of maintenance costs for the various implementation alternatives and not the absolute costs. The maintenance costs predicted in the following analysis will likely be considerably lower than the actual. However, the relative maintenance cost relationships between the various implementation approaches should be accurate.

The final maintenance procedure analyzes cost impacts of repairing at the PC board level. In this case, repair is actually achieved by replacing subpackages on the board and, as such, might be considered as a discard approach at the subpackage level. Typical subpackages might include DIPS, flat-packs, 2-x-2 inch substrates, packaged wafers, and others, depending upon function complexity. The cost elements of this simplified maintenance model are similar to those described.

The maintenance alternatives described are all field level replacement or repair approaches. More complex approaches incorporating repair within subpackages would be a factory-level repair and are not considered meaningful in terms of the average avionic system. Such an approach would be viable only in hybrid implementations using beam-leaded chips.

REFERENCE

C1. Anon: An Integrated Circuit Engineering Report, ICE Corporation, July 1968.

APPENDIX D

EDN's SECOND ANNUAL MICROPROCESSOR DIRECTORY

Because a suitable reproducible was not available, this information was not included here. See Reference 8.

APPENDIX E
MEMORY TECHNOLOGY

The rapid growth of electronic data processing over the last 20 years has been characterized by an almost insatiable appetite for larger and faster memories. Memory systems are now and will continue to be the highest single cost item in the computer hardware structure. Because of these reasons, continued attempts have been made to improve the speed and cost of computer memories. Although these attempts have been fruitful, memory is still the limiting area in many advanced computer system developments. For many computer applications the cost and processing power is largely determined by the speed of the various memories in that system. Any advance in memory is, therefore, of great importance for advancing the performance capabilities of a computer.

Today's memory devices can be classified into two basic categories: fast and expensive electronically accessed main memory and very slow and inexpensive electromechanically accessed peripheral memory. Between these widely diverse technologies is a large memory access gap which has persisted for a long time, even though the boundaries on either side of the gap have moved towards faster access by about an order of magnitude (ref. E1).

The absence of a technology to bridge this gap is not due to a lack of effort. Cryogenics, thin magnetic films, thermoplastics, magneto-optics, charge-coupled devices, magnetic bubbles, and electron beam technologies have, and are now, being investigated to bridge this gap. Some of these show great promise for reducing the access gap in the next five years.

There are a great many technologies that are suitable for fabricating memory elements. This compendium will not attempt to evaluate them all, but will be limited to those considered to be most applicable to future computer systems. This compendium will also largely ignore the many subdivisions of each technology area, and concentrate on the features available within that technology.

Memory technologies can be broken up into three major categories:

- Semiconductor Storage
  - Bipolar Memory
  - MOS Memory
  - Charge-Coupled Device Memory
- Magnetic Storage
  - Ferrite-Core Memory
  - Plated-Wire Memory
  - Movable Memory (Disk-Drum-Tape)
  - Magnetic Bubbles
  - Thin-film Memory
- Optical Storage
  - Laser Memory
  - Electron Beam

These technologies are used to implement a number of memory forms. These include:

- Random Access Memory (RAM) (Fast Read-Write)
- Read Only Memory (ROM)
- Electrically Alterable ROM (Fast Read-Slow Write)
- Block-Oriented RAM (Serial Transfer of Data Blocks)
- Content Addressable Memory (CAM)

Each memory form has an area of application in the hierarchy of computer systems. This is shown in Figure E1 along with typical speed requirements. The main characteristics of the memories are that as you move away from the computer, the memory is increasingly larger, slower, and cheaper.

In deciding which technology should be used, memory storage capacity versus access time is a prime consideration. This is shown in Figure E2 for a number of memory technologies. Note that RAM technology is usually limited to 10^7 bits. This is more a function of economics and the word size required to form an address than a technology limitation. Most all memory applications that require more than 10^7 bits use serial memories, where address is given to a block of data. Figure E3 shows the total time to access and transfer a block of 16K words as a function of the various technologies.

Semiconductor Storage

With the advent of large-scale integration, the use of small-capacity, high-speed memories distributed throughout a system became cost effective. The first applications were in cache memories (RAM) and microprogram store (ROM). Recent advances in the

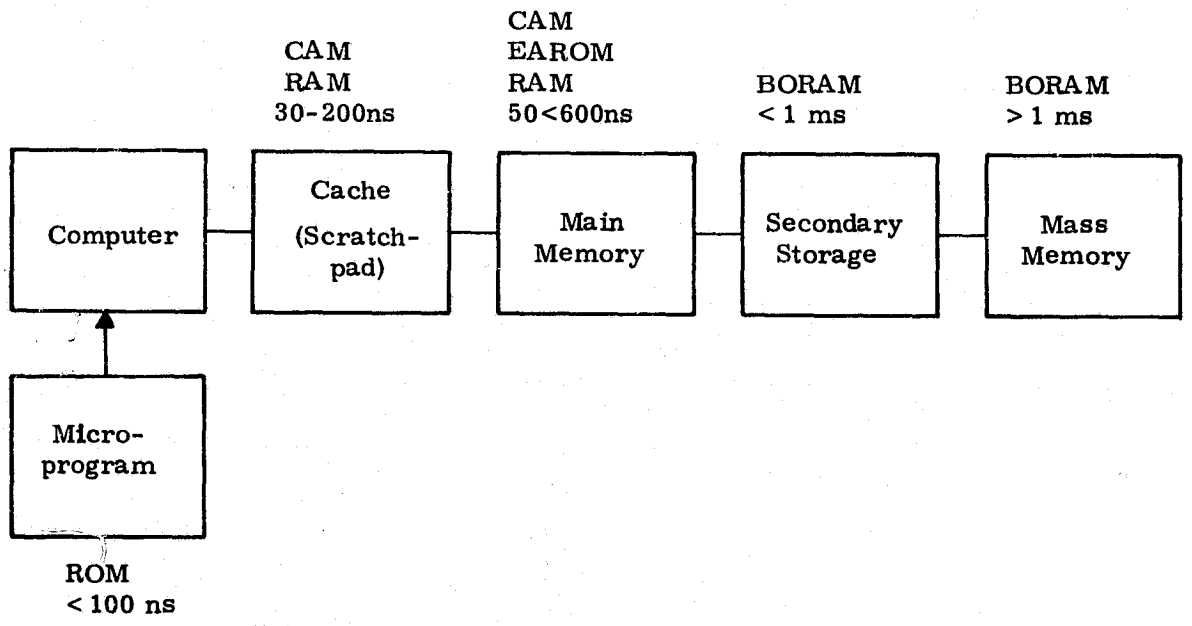


Figure E1. The Computer Memory Hierarchy

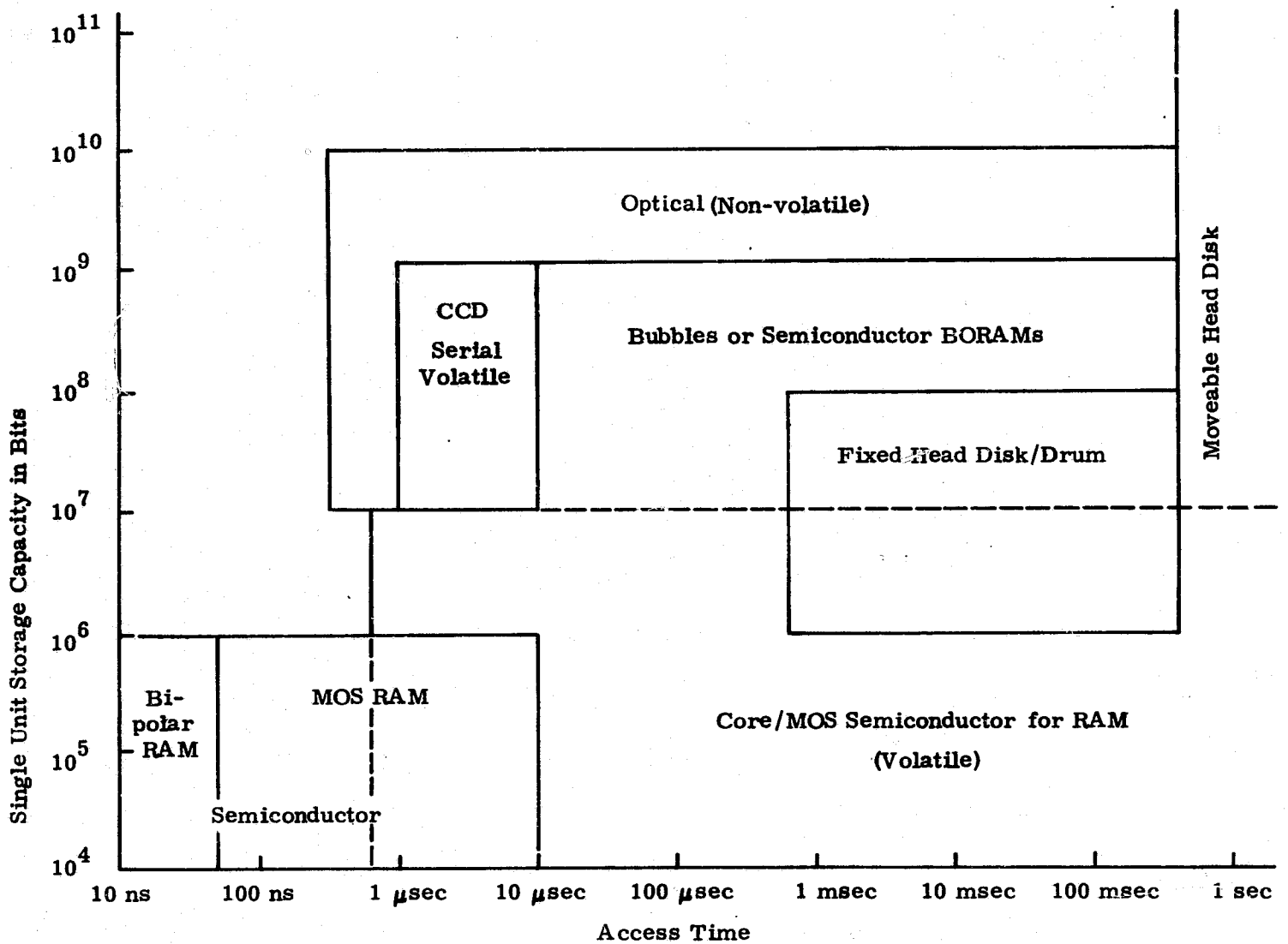


Figure E2. Access Time and Storage Capacity of Various Memory Technologies

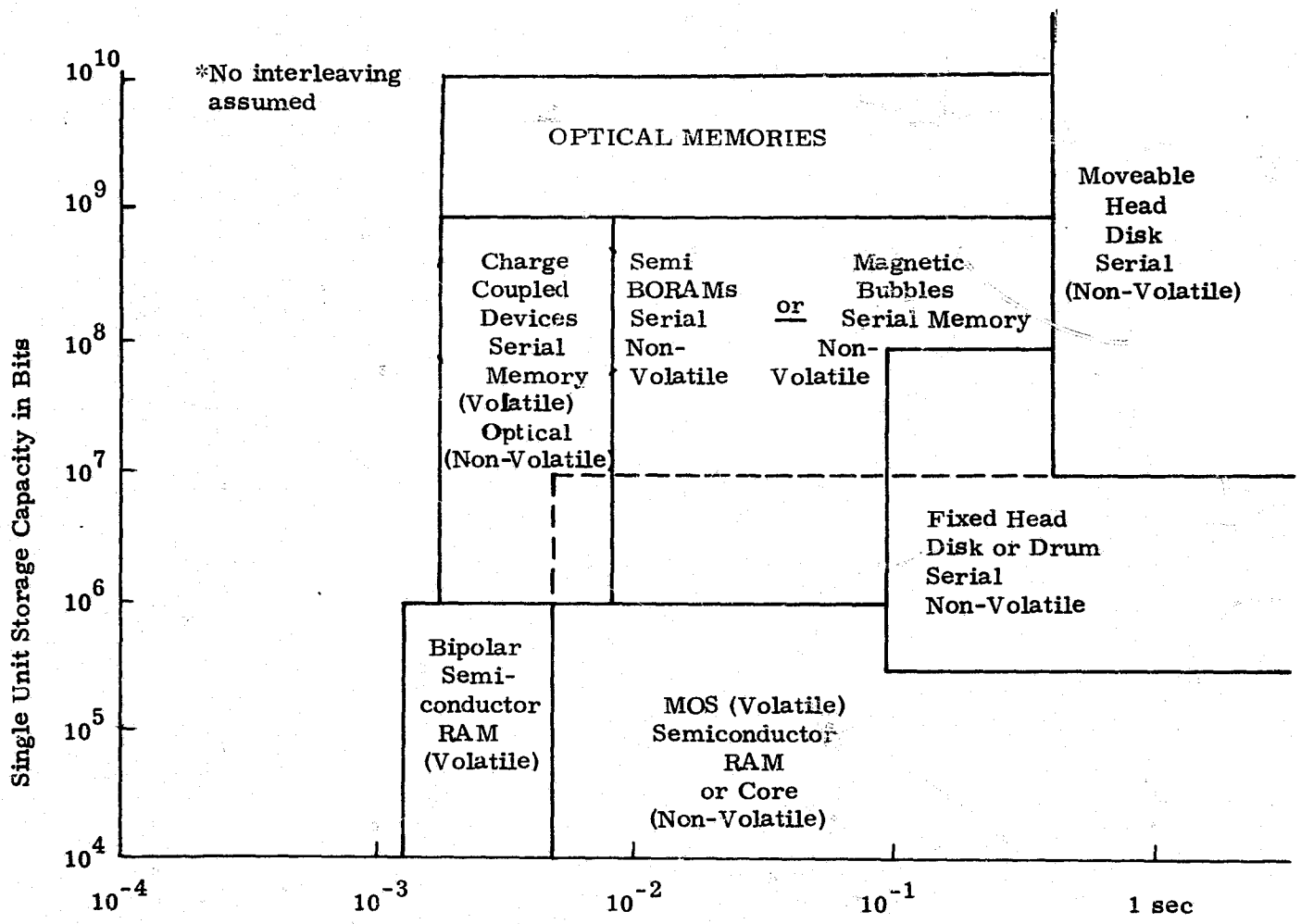


Figure E3. Transfer Time for 16K Block (including access time)

semiconductor art have moved their application into the main memory area. Technologies in the research stage promise to further extend the application of semiconductor memories into the realm of secondary storage.

Unlike magnetic memories, semiconductor memories are fabricated with decode, sense, and drive circuitry on the chip. This results in the cost of a memory system being almost independent of memory size and greatly reduces interface problems.

Semiconductor storage is primarily divided into three technology areas:

- Bipolar for high speed
- MOS for low cost
- Charge-coupled devices for serial memories

Both bipolar and MOS technologies can be used to fabricate random access memories (RAM's) with fast read-write characteristics, or read only memories (ROM's) where data is not alterable. In either technology, read only devices can be programmed at the mask level or by the user in the field. With MOS technologies it is also possible to fabricate EAROM's. An EAROM is a memory that is meant to operate in a read mostly mode, but can be altered when desired. Altering an EAROM is a slow process and is seldom performed on-line.

Approaches are presently under development in MOS, which may result in high-speed read-write RAM's that are nonvolatile. All currently available semiconductor RAM's will lose data if power is interrupted.

Bipolar semiconductor memories.— As in the case of semiconductor logic, bipolar technology offers highest speed in memories. Bipolar memories consume more power and have less device complexity than other semiconductor approaches and can be fabricated using a number of circuit forms to provide some speed/power flexibility. Presently, bipolar RAM's are available with up to 1024 bits per chip and access times in the range of 10 to 125 nsec. Present-day bipolar ROM's are available with up to 16K bits per chip with mask programming and 4K bits in field programmable form. These devices operate in the 15 to 140 nsec access time range and will achieve speed and complexity improvements in the future. The power dissipation at present is about .5 mW/bit.

Because memory systems are very regular, low-cost, high-density memory systems can be configured from bipolar chips using multichip packaging approaches.

Figure E4 shows the lowest cost off-the-shelf semiconductor RAM technologies as a function of access time (ref. E2). This figure verifies that bipolar memories are used only where high speed is required.

In addition to volatile RAM's, MOS technology is used to produce ROM's, EAROM's, and nonvolatile RAM's. MOS ROM's of 32 and 64K bits/chip are feasible with today's technology, and field programmable MOS ROM's are available with capacities up to 8K bits per chip (ref. E3).

Three semiconductor approaches have been used to produce electrically alterable ROM's. Two of these, floating avalanche MOS (FAMOS) and amorphous semiconductors, have not as yet received the emphasis of metal nitride oxide semiconductors (MNOS). An EAROM is a nonvolatile semiconductor memory that can be altered many times. Data alteration differs from a conventional RAM in that an erase is required prior to write. Most EAROM's erase on a block basis and the entire memory must be rewritten to change a single bit. The erase/write cycle is much slower than the read cycle (milliseconds versus nanoseconds), and a fatigue phenomenon limits the number of erase/write cycles to the range of 10^6-10^{10}. The main advantage of EAROM's over RAM's is nonvolatility, over ROM's it is alterability. It is expected that EAROM's will be used for microprogram store, and for program and secure data areas of main memory.

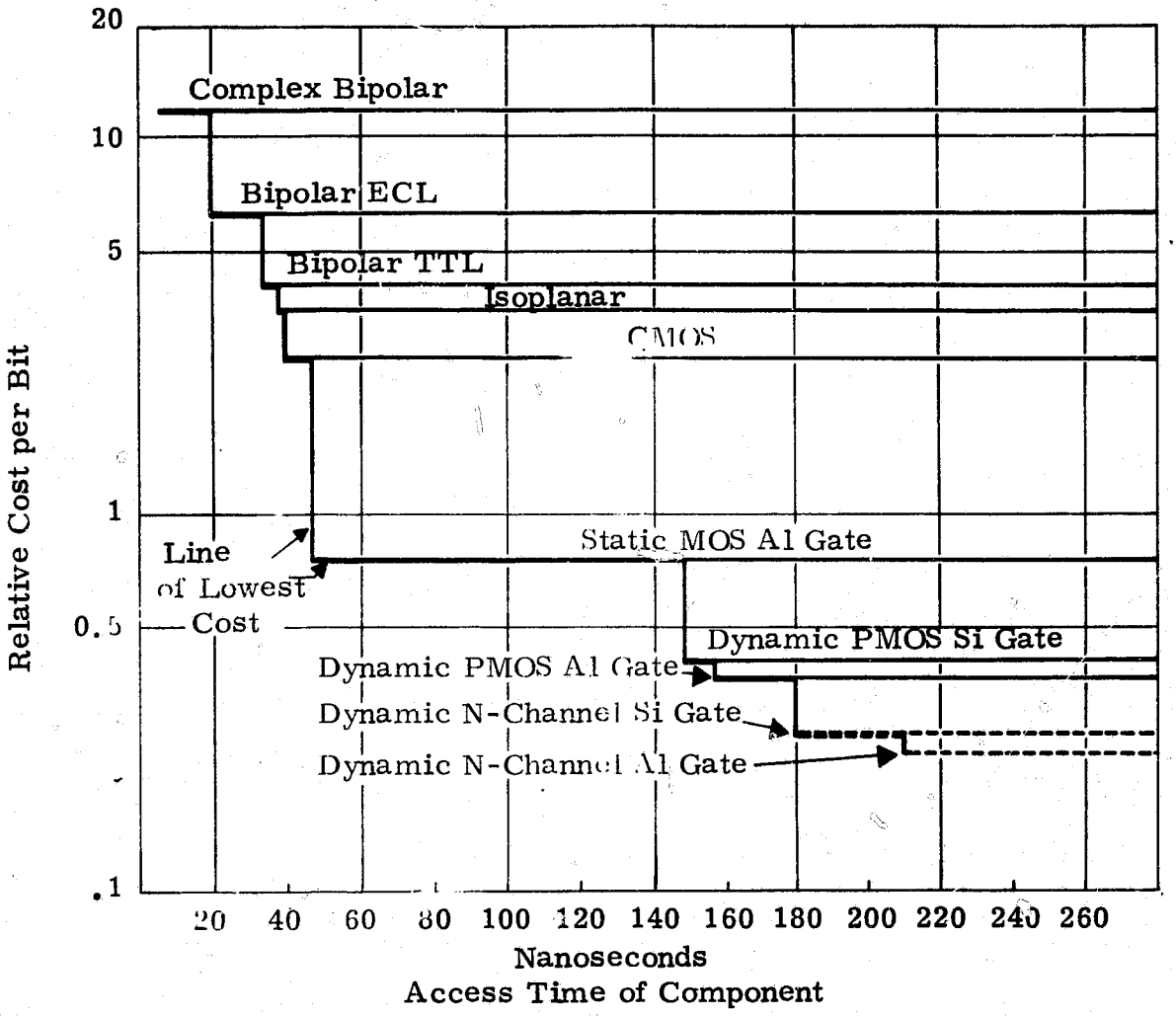


Figure E4. Relative Cost versus Access Time for Semiconductor Memories

The MNOS and FAMOS technologies operate by trapping charge in the gate region of a field effect transistor. In MNOS, the charge is trapped in nitride; in FAMOS, the charge is trapped in the region of a floating gate. Both technologies can potentially operate as nonvolatile RAM's. The government is presently funding efforts in MNOS to achieve nonvolatility with 600 nsec read, 1 μsec write, 1 μsec erase, and 10^{10} erase/write cycles before failure. The main problem is pushing the technology to get high write speed.

It has been postulated that high-speed nonvolatile RAM's can be made by incorporating an EAROM transistor along with a RAM cell. Although this approach would not require a technology advance, it has not yet obtained contract support.

High-speed, nonvolatile, block-oriented RAM chips are also under development using MNOS. These presently have 2K bits/chip with extensions to 4 or 8K possible. They will be used to build large serial memories with access times of 2 to 10 μsec and data rates of 2M bytes/sec. Present block sizes range from 256 to 2048 words. Much of the emphasis on MNOS is a result of the radiation hardness inherent in this technology.

In addition to RAM's, CMOS is being used to realize content addressable memories. A 64-bit chip is presently available that can perform a number of search operations in 200 nsec. The large number of pin-outs required may limit growth in this area.

Charge-coupled device (CCD) memories. — Although a CCD memory can be configured in a number of ways, all are basically serial in nature and, hence, are block oriented rather than word oriented. The CCD is characterized by high-packing density, low-power dissipation, and a structural simplicity that will lead to low cost.

Present-day CCD memory devices operate in a digital manner where charge represents a "one" and lack of charge represents a "zero." Chips are available today with 16K bits of storage. CCD memories are stictly serial and exhibit access times in the 100 to 200 μsec range and data rates of 1 to 10Mbits/sec. Typical power dissipation is 20 μW/bit at 10 MHz and 4 μW/bit at idle speed. Because of transfer inefficiencies, repeaters (sense/inject circuits) are required about every 64 bits. Because the devices are dynamic, a minimum clock rate of 50 to 100 kHz is required to ensure data retention. Refresh rate is a function of operating temperature.

CCD's offer significant speed, power, and reliability advantages over today's disks and drums. Because of the volatility of CCD's, however, semiconductor BORAM's may be required in some high-speed secondary memories, despite their higher cost.

CCD's could also be used to store digital data in analog form. With analog storage, 5 to 13 bits of information could be stored in a single cell and converted to digital, upon readout, through a conventional A/D converter. Process problems are more difficult in this approach, and if data were to be stored for more than seconds, practical analog regeneration circuits must be developed (ref. E4).

Magnetic Storage

For years magnetic storage has been the most common type of memory in use. Magnetic storage has had the cost advantage over semiconductor approaches until just recently, and is nonvolatile and inherently radiation hard. The main problems with magnetic storage have been the difficulty and inefficiency in interfacing with the semiconductor devices with which they must communicate. This factor has almost completely forced magnetic memories into large-capacity applications such as main memory and beyond.

Appendix E

For main memory applications, the ferrite core has been the workhorse of the industry. Where severe environments were encountered, plated wire was used. Secondary storage and mass memory applications have employed disks, drums, and tape units. Thin-film memories have been used extensively by the aerospace and commercial computer manufacturers. The principal characteristics of thin-film memories are given in Table E1. These electromechanical devices are slow and unreliable and will soon be replaced in systems with high-reliability requirements. Thus, most of the present magnetic technologies may not be designed into systems in the 1980's. By far the most promising magnetic technology for future memories is magnetic bubbles.

TABLE E1. CHARACTERISTICS OF THIN-FILM MEMORIES

Parameter	Oligatomic Film Mass Memory	Conventional Film Main Memory
Max bits per module	100 million	1 million
Cost per bit	0.1¢	4¢ to 10¢
Access time	3 μs	0.75 μs
Drive current	30 mA	600 mA
Power dissipation	98W	150W
Volume	2 ft^3	0.75 ft^3
Weight	90 lbs	50 lbs

REFERENCES

E1. Speliotis, D.: Bridging the Memory Access Gap. Proceedings, National Computer Conference, May 1975, pp. 501-508.

E2. Lloyd, R.: RAM Technology: MOS or Bipolar? Electronic Products, 18 June 1973, pp. 53-59.

E3. Torrero, E.: Focus on Semiconductor Memories. Electronic Design, 1 April 1975, pp. 98-107.

E4. Anon: Analog Methods May Yield Denser Computer Memories. Electronic Design, 15 March 1974, p. 25.

APPENDIX F

INTERFACE COMPONENTS AND MODULES

Data Converters

The booming demand for data converters has resulted in creation of a wide variety of types: monolithic or hybrid integrated circuits, potted modules, circuit boards, or rack-mounted instruments. The inherent advantages and disadvantages of each type are so different that no single one is just right for every application. Overall system requirements determine the three basic criteria which influence converter selection: resolution, accuracy, and speed. Other application and budget constraints might include size or form factors, as well as externally required word size and speed.

Digital-to-analog (D/A) converter selection criteria. - A definition of resolution, accuracy, and speed is needed as a basis for discussion of selection criteria. The resolution of a D/A converter is defined as the smallest discernable quantum change in the output for a single step change in the digital input. Specifying D/A converter accuracy requires a basic understanding of the more common terminology and error contributors listed below.

• Nonlinearity or relative accuracy error	The deviation of the actual analog output from an ideal straight line drawn between the end points.
• Differential nonlinearity	Deviation of the output from an ideal least significant bit (LSB) change that results from a LSB change in the digital input code.
• Monotonicity	The analog output remains the same or increases for increasing digital input codes.
• Glitch	A transient spike that usually occurs at major carries.

The speed of a D/A converter is determined by its settling time to a desired accuracy. Most D/A converter manufacturers specify speed for a full-scale range change in digital input codes and for one LSB change in digital input code. Other criteria used to specify D/A converters include: digital input codes, unipolar or bipolar output, voltage or current output ranges, power supply voltages, power consumption and supply rejection, and packaging.

D/A converter types. - The vast majority of D/A converters on the market today use two basic circuits: the weight-resistor network and the R-2R configuration. Both terms relate to the type of resistive ladder network used in the converter, and both types are most conveniently described for the case of a voltage-output D/A; that is, the case where a digital input yields a discrete voltage level at the output.

Weighted-resistor D/A converters: Weighted-resistor D/A converters include a reference voltage source, a set of switches, a set of binary-weighted precision resistors, and an operational amplifier. Each binary bit of the digital input word controls its own switch. If the bit value is a binary ONE, the switch closes; if it is a binary ZERO, the switch stays open. When the switch closes, the reference voltage is impressed across the weighted resistor in series with the switch, and a current flows into a summing bus.

When the currents from all of the legs of the weighted-resistor ladder network are summed in the amplifier, the output voltage of the amplifier is proportional to the total current and, therefore, to the value of the digital input code. In general, weighted-resistor D/A converters are easier to make than R-2R converters.

R-2R D/A converters: The R-2R D/A converters also contain a reference voltage source, a set of switches, and an operational amplifier. However, instead of a set of binary-weighted resistors, they contain two resistors per bit: one in series with the bit switch, and the other, valued at one-half the series resistor, is in the summing line, such that the combination forms a pi network in conjunction with succeeding stages. In the R-2R all resistors and switches must be perfectly matched and must track each other over temperature. The R-2R is best made by monolithic integrated-circuit techniques, and its main advantage is that only two values of resistance are needed.

Multiplying D/A converters: One important type of D/A converter is the multiplying converter, which generates an output that is proportional to the product of the input digital code and a time-varying reference voltage. These converters do not include an internal reference source and can be implemented either with ladder network techniques or with taped autotransformers.

Digital-to-synchro (D/S) conversion: Synchros, one of the more fundamental components of control systems, are extremely versatile, operate well over long distances and in difficult environments, and are capable of very high accuracies when correctly applied. It is not widely appreciated that two inexpensive synchros in a simple two-speed angle transmission system are readily capable of precision to 1 part in 20,000 at a price far lower than that of a comparable shaft encoder system.

Because of their wide use in control systems, conversion between synchro and digital data is an essential operation. Digital computer commands must be translated into analog control signals and, similarly, synchro position data must be digitized for computer processing. To carry out these conversions, a variety of synchro-to-digital (S/D) and digital-to-synchro (D/S) packages is available. Some are simple converter modules; others are complete rack-mounted subsystems including sample-and-hold and multiplexer circuitry.

Consider the simple D/S conversion circuit of Figure F1. Weighted inputs are summed much as in a conventional binary D/A converter; but in this instance the load resistor plays a significant part. By correctly setting the relative value of the load resistor, the ladder network is nonlinearly loaded such that the analog output tends to flatten out with increasing digital input, with the output curve roughly approximating the first 90 degrees of a sine curve. Analogously, if the bits are switched out in reverse order starting with all switch positions on, the output curve resembles the first 90 degrees of a cosine curve. Although the sine and cosine curves are relatively inaccurate, the ratio of sine to cosine is an extremely accurate representation of the tangent of the digital angle. Thus the network of Figure F1 can be taken as the basis for a D/S converter.

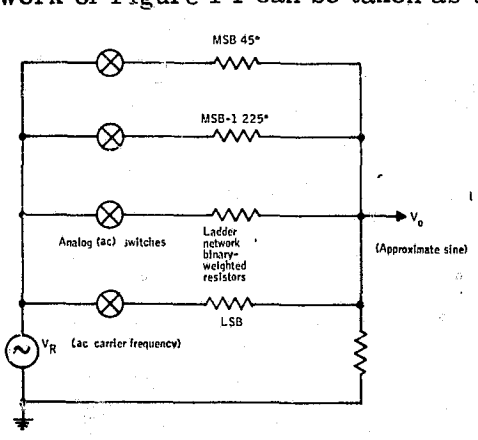


Figure F1. Digital-Sine Conversion Network for 0 to 90 Degrees (A cosine network is identical except that the complement of the binary input must be applied)

D/S converters are readily made to high accuracies such as 13 and 14 bits. The characteristic use of Scott-Tee transformers enables simpler sine and cosine voltages (90 degrees apart) rather than the 120-degree voltage phasing of synchros. By converting between 90- and 120-degree phasing, the Scott-Tee performs a valuable simplification and, at the same time, provides input-output isolation.

Although we have shown a resistance bridge as the major element in the D/S converter, conversion circuits frequently use suitably tapped, wound, magnetic toroids for sine-cosine function generation. These offer high input impedance, very low output impedance, essentially exact output functions, unexcelled long term stability, and very low sensitivity to magnetic noise.

Analog-to-digital (A/D) converter selection criteria. - Again the three basic criteria of resolution, accuracy, and speed are defined with an important fourth criterion, A/D aperture error. The required resolution of an A/D converter is determined by dividing the full-scale amplitude by the smallest desired quantum.

Errors which affect overall A/D converter accuracy are linearity errors. Initial gain and offset errors of most A/D converters are adjustable to zero leaving error components of nonlinearity and drift. Drift errors present in A/D converters are gain and offset errors. Gain drift is more important because some loss of resolution could result if the gain is off by one or more LSB's at full scale. In A/D converters, very low gain drift temperature coefficients are very hard and expensive to achieve. Typically, 5 to 7 ppm/°C represent the top of the line in mass produced A/D converters with many manufacturers offering 10 to 20 ppm/°C for gain drift to maintain competitive prices. This specification, in addition to ±1/2 LSB linearity, contributes more than any other factor except resolution to the cost of an A/D converter.

Analog-to-digital conversion speed is defined by the Nyquist sampling theorem, which states that a minimum of 2 samples per cycle are required to completely recover continuous signals in a noiseless environment. In typical instrumentation systems, noise does exist, and from 5 to 10 samples per cycle are required. For single-channel applications with d-c and very low frequency signals, sample rate is usually a multiple of the power-line frequency. Most integrating A/D converters offer sampling rates at discrete multiples or sub-multiples of 50 or 60 Hz, and provide essentially infinite rejection of these frequencies. The minimum sampling speed required is the number of samples per cycle multiplied by the highest frequency component of the data. For time multiplexed systems, the speed requirement of the A/D converter is dependent on system throughput speed and aperture error.

Aperture error must be examined before a final selection of A/D converter speed can be made. Aperture error can be reduced either with very high speed A/D converters or by using a sample/hold in front of an A/D converter.

A/D converter types. - As with D/A converters, the vast majority of A/D converters on the market are of two basic types: the dual-slope integration A/D converter and the successive approximation A/D converter. Each takes a voltage input and puts out a digital code proportional to the input voltage.

Dual-slope integration A/D converters: Dual-slope integration A/D converters contain an integrator, some control logic, a clock, a comparator, and an output counter. The unknown voltage is fed into the integrator for a predetermined period of time, as measured by the clock. Then a reference input voltage of opposite polarity is switched to the integrator. As the integrator processes the reference voltage, the output of the integrator decreases from the unknown voltage level to zero. The duration of the second period of integration is automatically proportional to the average of the unknown signal level over the predetermined integrating period. A digital counter then measures the reference integration period and provides the digital code at the output. Dual-slope integration A/D converters offer excellent noise rejection and linearity, but are fairly slow.

Appendix F

Successive approximation A/D converters: Successive approximation A/D converters contain a comparator, D/A converter, a shift register, an output register and some control logic. The input voltage is fed to one input of the comparator, while the output of the internal D/A converter goes to the other input. Each bit line in the D/A converter corresponds to a bit position in the output register. When a conversion command is applied, the converter is cleared and the most significant bit (MSB) output of the D/A is fed to the comparator to be tried against the unknown input level. The MSB output is, of course, equivalent to one-half the converter's full-scale range. If the unknown is greater than the MSB, a ONE is inserted in the MSB position in the output register. If the unknown is less than the MSB, a ZERO is inserted. Then the circuit tries the next bit in the D/A. If this bit does not exceed the input, a ONE goes into its position in the output register; if it does, a ZERO is positioned. The procedure continues through to the least significant bit (LSB), at which stage the conversion is completed. The procedure is similar to a chemical balance in which one weight is added at a time. The main advantage of the successive approximation A/D is speed. Accuracy is primarily determined by the accuracy of the internal D/A.

Conversion speed of the successive approximation A/D converter is based on the settling time of the MSB logic ladder network and comparator. In a single rate conversion unit, conversion time would be n times the time required for one bit. Because only one comparator is used and representative hardware is limited to logic and ladder network, the successive approximation technique provides an inexpensive average speed solution.

Sequential conversion: Sequential conversion takes advantage of the fact that the propagation time through a chain of amplifiers is equal to the square root of the number of stages times the individual settling time, as opposed to adding up each stage. By adding a comparator for every binary-weighted network, it is possible to take advantage of this higher speed.

Sequential conversion is used for applications requiring high-speed data conversion. However, because of the increase in the number of comparators and the need to use an amplifier for every weighting network, the cost is considerably more than that of successive approximation. Although several products have been developed using the basic sequential configuration, they have not been as popular as modified versions of this technique which take advantage of gray code conversion to improve speed, or which use a combination of sequential and successive approximation to lower cost.

An all-parallel conversion method, developed for a 100-MHz throughput tunnel diode converter, provides the ultimate in speed but costs the most. In this configuration a comparator is used for each bit generated, and a weighting network or D/A converter increases in number of sections towards the LSB. By using delay lines to time the signal and to store the analog value between comparators, it is possible to operate the converter at a throughput rate of one comparator, but with a total delay of n times the number of comparators. This approach is costly as it requires a D/A converter of successively increasing complexity at each bit-decision comparator, as well as n comparators, and delay line storage for the analog signal, weighting decisions, and binary code outputs.

Synchro-to-digital S/D conversion: To go from D/S to S/D conversion involves much the same use of feedback as D/A and A/D conversion. Typically, instead of a fixed reference, the 2-phase (Scott-Tee transformer) equivalent of the input synchro signal is applied to the two resistance bridges. When a digital input is simultaneously applied, it can be shown that the sum of the two analog bridge outputs corresponds to the sine of the difference between the synchro and the digital input angles. In effect, this sum of two analog voltages serves as a comparator output, indicating whether the digital representation of the input angle is too large or too small. Suitable circuits may be devised, as in conventional A/D converters, to bring the digital angle (output) into correspondence with the synchro input angle.

An important application problem in S/D conversion relates to questions of phase shift and harmonic distortion. Depending on size and frequency, synchros may easily introduce 5- to 10-degree phase shift between the a-c line and the synchro output. Similarly, due to bandwidth limitations, line distortion is nonuniformly attenuated in the synchro for the separate harmonics. Output and input waveforms are different, and the nonlinear properties of magnet iron introduce additional distortion within the synchro itself. Obviously, conversion means which do not take these nonideal factors into consideration must suffer degradation in accuracy; for example, a sample-and-hold circuit may be sampling a harmonic peak.

Digital synchro control transformers -- a special variety of converter -- are often of value in digitally controlled positioning systems. A synchro input is applied from a standard transmitter; simultaneously, a digital command is applied. The output is a single-phase a-c signal proportional to the sine of the difference between the synchro and digital angles. Typically, this output is used as a servomechanism error signal, and is amplified and applied to a motor, which corrects either the synchro or digital input depending on the type of positioning transducer. Synchro differential transmitters with a synchro and a digital input are also available. Output is a set of three-wire synchro voltages that correspond to the angular difference between the two inputs. Some companies offer auxiliary modules for use with synchro conversion systems. Some functions are synchro-to-d-c conversion, d-c-to-synchro conversion, precision modulation and demodulation, and synchro-to-resolver converters. These functions can substantially simplify system design.

The advanced state of development of S/D and D/S conversion equipment, with the well known advantages of synchros and the relatively low cost of high accuracy two-speed synchro systems, makes this approach to digital control a preferred one in many applications.

Many forms of a-c-to-angle solutions are well documented in the literature. Six types, which form the technique's foundations use: (1) RC phase-shift networks, (2) resolver bridge or self-balancing resolver bridge, (3) linear segment function generator, (4) reference voltage generator, (5) harmonic oscillator vector rotation, and (6) digital resolver vector rotation. Not all of these types have been implemented, primarily due to response requirements of the analog components over the temperature range which make compensation such as capacitor trimming necessary. Several types, however, have been produced and have been used satisfactorily in avionics applications.

Miscellaneous A/D converter types: In addition to the two most widely used schemes, there are many other types of A/D converters on the market. These include single-slope integrator, triple-slope integrators, voltage-to-frequency converters, voltage-to-pulse rate converters, parallel-series converters, and straight parallel converters.

For slowly varying signals, either an A/D or V/F converter can produce accurate results. But, as signal frequencies increase, A/D converters cannot follow the signal, and converter digital outputs may have substantial errors. V/F converters usually respond faster than A/D's, but they introduce their own problems -- linearity errors and temperature drifts. The linearity specification defines the accuracy with which the converter will change a voltage into a corresponding frequency, and the temperature coefficients tell you how much the circuit drifts with temperature.

Most manufacturers sell units that are a compromise between good linearity and temperature drift. For example, you can buy a converter that has a linearity error of 0.05% maximum, but has a temperature drift of 100 to 200 parts-per-million (ppm) for every degree change in temperature. Thus, for a 10-C change the frequency might drift by as much as 2 kHz if you use a 1-MHz converter. The linearity of this converter also causes an inaccuracy of 500 Hz even before the drift errors are added in.

On the other hand, you can buy a unit that has a 0.5% linearity but has a 10 to 50 ppm/°C temperature drift. Here repeatability is the key. Linearity error may not be important if you still get the same output frequency five or ten minutes later.

Voltage-to-frequency (V/F) converters have been available in component form for only two or three years. Yet modular versions already are down in price, ranging from about $200 for a unit providing an output frequency as high as 5 MHz to less than $35 for a 10-kHz unit. Linearity error can be as good as ±0.005% to ±0.1%.

The first monolithic voltage-to-frequency converter was recently introduced by the Semiconductor division of Raytheon Co., Mountain View, California. The unit contains a voltage comparator, a one shot, and a precision switched current source. It sells for only $3 each in quantities of 100. The full-scale frequency is resistance-programmable over an operating bandwidth of 10 to 100 kHz, and linearity error ranges from ±0.05% to ±1%, depending on whether an external integrator is used.

Digital Input/Output (I/O) Modules

In most applications it is not enough to acquire the data; you must also send it to the processor. To do this, Analog Devices has developed the Serdex line of industrial data-acquisition and control modules. The Serdex transmitter, Model STX-1003, converts the 4-digit parallel BCD code (that A/D converters can output) into two-wire, compatible, serial ASCII code. This coded output is compatible with either a computer serial link or a data terminal serial link. The receiver module, Model SRX-1005, does the reverse; it converts serial ASCII to 4-digit parallel BCD for possible D/A conversion into a control signal.

The other modules in the Serdex series include a multiplexer pair, SMX-1004 and SMC-1007, which can combine up to 16 process lines onto a two-wire cable, and a clock module, SCL-1006, which can supply all the timing and control signals for all the other units. The clock module also supplies the operating voltages needed by the other units. The transmitter or receiver costs $179, the multiplexer pair $214, and the clock module $65.

Another recently developed system for serial data exchange is DEC's PDM-70 -- a programmable data mover with input/output slots that accept up to seven boards. The boards accept either digital or analog data. The PDM-70 takes these data and converts them into serial ASCII for two-wire transmission. Each of the option boards contains a universal asynchronous receiver-transmitter that does the actual code conversion. The options available include a BCD or binary input board that will accept up to a 32 bit parallel word and reformat it into serial ASCII. The BCD/binary output board does just the reverse. The analog input board can handle up to four differential channels and can convert the analog signals into serial ASCII. The converter boards have programmable gains and full-scale voltage ranges of ±1.999 V, ±199.9 mV, and ±19.99 mV. This board also has a mirror image board -- the analog output option.

There is still another connecting link between the sensor and the computer -- the card and rack system. One, manufactured by Computer Products, has real-time peripheral cards and racks that can be assembled into any type of data-acquisition or control system. The RTP 7400 series of subsystem cards includes such functions as A/D conversion, sample/hold, and low-level multiplexing control cards. Computer input/output interface cards for almost any manufacturer's computer are available from Computer Products. The input/output expander card, which differs for each computer, controls all the cards plugged into the system chassis. Each expander board can handle up to eight peripheral subchassis, which can hold up to 16 circuit boards to perform any desired function.

Appendix F

Analog Input/Output (I/O) Modules

Operational amplifiers. - Mixed linear processing continues to influence monolithic operational amplifiers. With its ion-implantation technology, National Semiconductor has succeeded in producing a line of operational amplifiers that have bipolar outputs and matched junction-FETs at their inputs. National, which calls its new process Bifet, is also using the technique for a line of analog switches and analog multiplexers. The operational amplifiers offer input offsets of 5 pA and 1 mV, and an offset drift of only 3 μV/°C.

Quad operational amplifiers, offering the advantages of low cost and high functional package density, can now provide performance comparable to single units. Input bias currents are down around 500 nA, and output slew rates are slightly greater than 1 V/μs. Frequency performance is also impressive, with a small-signal bandwidths being 1 MHz or more and large-signal bandwidths ranging from 20 to 40 kHz. Moreover, chip designs have been improved such that crossover distortion is no longer a problem when the operational amplifiers must drive a grounded load while operating from a split supply.

Sample and hold (S/H) amplifiers. - To freeze rapidly changing inputs, S/H circuits are often used in front of the A/D converter; however, this does not completely solve the accuracy problem. The S/H circuits introduce other errors that are not always mentioned on manufacturers' data sheets. A pedestal error occurs in S/H circuits each time the circuit switches from the track to the hold state. This stems from the capacitive transfer of residual charge across the turned-off switch onto the holding capacitor. The same switch signal that controls the S/H circuit causes a small voltage spike when the circuit switches to the track mode.

Modular S/H circuits are available from many companies. Most of the larger manufacturers of A/D converter modules, such as Teledyne-Philbrick, Analog Devices, Hybrid Systems, Burr-Brown, and Intech, also have modular S/H circuits that include the holding capacitor. Also, some companies offer S/H modules with pinouts for an external holding capacitor. Burr-Brown, for example, has a hybrid IC S/H circuit that requires only a holding capacitor to function. The voltage droop of this amplifier is a low 0.1 mV/msec, and the capacitor determines the rest of the hold characteristics. This unit, Model SHC-23, costs $45 in singles and is housed in a TO-8 metal can.

Several companies have fabricated pairs of amplifiers with the necessary FET switches as single-chip circuits which they sell as general-purpose S/H amplifiers.

Multiplexers. - To collect many channels of data and to combine them onto a single line, you need either a multiplexer or relay scanner. Of recent advances in multiplexers, the single-chip, 16-channel analog IC dominates; older modular circuits are slowly giving way to this new unit. Relay scanning circuits are advancing too, with better thermal design, isolation, and shielding methods, but scanners will be used less as solid-state multiplexers move in to replace them in all systems except those that involve extremely low-level signals (below 10 mV).

Problems faced when analyzing multiplexer specifications include thermal and voltage offsets, contact resistance, switch resistance, contact bounce, switching time, and level matching.

Solid-state switches should be used to preserve signals buried in noise, especially low-level signals. Noise in semiconductor materials often overrides low-level analog signals. Mercury-wetted reed relays with switched ground shields offer the best solution. Because low-level analog signals are usually of very low frequency, the slowness (100 operations/sec) of mercury-wetted relays should not pose any serious problems.

For low-level transducer signals, signal conditioning at the sensor location usually allows the most accurate data acquisition. If many low-level points are grouped near each other, a low-level signal scanner or multiplexer can minimize the number of amplifiers needed. The signals are first multiplexed and then fed through a single amplifier. In this single-amplifier approach, crosstalk within the multiplexer, bandwidth limiting of the amplifier, settling time of both the amplifier and multiplexer, and other factors can destroy the signals. Other problems, like the need to run a line to the sensor to supply power for the conditioning equipment, must also be considered.

Optical couplers. - In optical couplers, performance seems to be leveling off. Manufacturers are concentrating more on refining technologies. For an imput current of only 1 mA, minimum current transfer ratio can be guaranteed at 100% for a phototransistor coupler, and at around 200% for a photoDarlington coupler. The maximum data rate of logic-gate couplers is typically around 5 MHz, but speeds of up to 10 MHz can be realized with optimum circuit conditions. Guaranteed minimum isolation voltage for a single device in a dual-in-line package can be as high as 5 kV.

Coupler prices should continue to decline for the next several years, dropping eventually to the $1 level for high-performance devices that are now $2 or more. New types of couplers are also beginning to emerge - couplers whose switching threshold can be programmed by means of external resistors, and multichannel couplers intended for analog applications. In the near future, there may be couplers having a phototriac output, or couplers whose input and output remain linearly proportional to each other.

Input/Output Components

Thin-film resistor networks. - Thin-film resistor networks traditionally have been expensive, selling for from $3 to close to $40 each, even for off-the-shelf standard parts. By applying the high-volume automatic processing techniques that are used for IC's, the Resistor Products division of Analog Devices is producing thin-film resistor networks in hermetic packages that in some cases sell for half as much as previous devices, with no degradation in performance.

National Semiconductor has announced a line of thin-film resistor networks packaged in plastic that are intended to compete with standard thick-film networks in ceramic packages. National aims to get prices down to less than 35 cents each in lots of 100,000.

Capacitors. - Ceramic capacitors, both chip and packaged versions, have been the fastest growing segment of the capacitor industry. Prior to 1975, unpackaged ceramic chips were strictly fixed-value devices; now Vitramon Inc. of Bridgeport, Conn., has a line of ceramic chips whose capacitance can be adjusted incrementally without adversely affecting device performance.

Another significant step in ceramic capacitors was the successful transition from a precious-metal system to a base-metal system by USCC/Centralab of Los Angeles. The company is using a nickel alloy to make the internal electrodes and the terminations for its capacitors, permitting price reductions of at least 30%, and even 50% in many instances.

APPENDIX G

COMPUTER INTERCONNECTION STRUCTURES: TAXONOMY, CHARACTERISTICS, AND EXAMPLES

Reprinted by permission from ACM Computing Surveys, Vol. 7, No. 4, Dec. 1975, pp. 197-213.

Computer Interconnection Structures: Taxonomy, Characteristics, and Examples

GEORGE A. ANDERSON
and
E. DOUGLAS JENSEN

Honeywell, Inc., Systems & Research Center, 2600 Ridgway Parkway NE, Minneapolis, Minnesota 55413

This paper presents a taxonomy, or naming scheme, for systems of interconnected computers. It is an attempt to provide an implementation-independent method by which to identify designs, and a common context in which to discuss them. The taxonomy is based on interprocessor message handling and hardware interconnection topology, and distinguishes ten basic multiple-computer architectures. Various relevant attributes are identified and discussed, and examples of actual designs are given for each architecture.

Keywords and Phrases: distributed processing, distributed computers, multiprocessors, multicomputers bus structures, computer networks

CR Categories: 3.81 4.32 6.20

INTRODUCTION

Currently, one of the most active areas in computer architecture is the interconnection of computers to form systems which are called "distributed processors," "distributed-function computers," "computer networks," and similar names. These systems range in organization from two processors sharing a memory to large numbers of relatively independent computers connected over geographically long distances. A discouraging aspect of this activity, however, is the almost total lack of published information describing the rationale for various designs, or comparing the results achieved by various approaches. In part, the authors believe this condition exists because there has been no common context in which such discussion could take place, no set of design issues, no list of system characteristics to be traded off, and, in fact, not even a common nomenclature for system identification. Our paper is an attempt to begin filling this need. In it we present a naming scheme, or taxonomy, for identifying various systems of interconnected computers, and we discuss design decisions and system characteristics which we believe are germane to these architectures.

The authors know of only one other general taxonomy for interconnected computers and that is a brief one (having different dimensions) with few system characteristics and no nomenclature [SIEW74]. Some interconnection topology issues are also considered in [CHEN74] and [THUR72], although these are primarily concerned with the next lower level of the interconnection design—control and communication. One level beneath these are a number of papers dealing with the design of "explicit" switches, such as crossbars [PIPP75] and permutation/sorting networks [THUR 74]. In addition, there is a wide variety of digital

CONTENTS

data communications literature pertaining to queueing, routing, multiplexing, etc. [MART72]; much of this is relevant for certain computer interconnection architectures and implementations.

For the first step toward developing a common nomenclature for system identification we have restricted ourselves in several significant ways. First, we are concerned solely with interconnected hardware units in which "processes" can execute. We use the word process in the conventional sense, and designate the hardware units as Processing Elements, or PEs. By this definition we specifically exclude single-instruction stream, multiple-data stream machines such as ILLIAC IV and PEPE. We further limit ourselves to systems in which any PE can communicate with any other through the system interconnection mechanism.

Our method for identifying the interconnection structure of a system is to isolate the major hardware units involved in the transfer of information between processes in different PEs. We call this transfer a "message transmission," and do not distinguish between instances of *message* such as data blocks, service requests, semaphores, etc. In the interconnection structure itself we distinguish two functional entities—paths and switching elements.

A *path* is the medium by which a message is transferred between the other system elements. Some examples of paths are wires or busses, radio links, common-carrier data-transmission facilities, and memories. The transmission of a message over a path results in no alteration of the message.

A *switching element* is an entity which may be thought of as an "intervening intelligence" between the sender and receiver of a message. A switching element affects the destination of a message in some way—by altering the message (e.g., changing its destination address), by routing it to one of a number of alternative paths, or by both actions. These notions of message, path, and switch are basic to the approach we have taken.

Our taxonomy thus describes configurations of three hardware archetypes: PEs, paths, and switching elements. This small number of types leads to several simplifications which serve on one hand to make system organizational issues clear, but on the other, admittedly obscure the noninterconnection aspects of system design. For example, we do not distinguish between a computer and its interface to the rest of the system—both are part of the PE. Neither do we make a distinction between circuit switching and message-switching—both are accomplished by the switching entity. Perhaps the most significant issue that is not treated is interprocess communication strategies and problems, such as message addressing, deadlock, etc. By these omissions we do not imply any relative importance, but rather, we stress that we have taken a limited step in but one of several important areas. It is our hope that this step will stimulate similar work in such complementary areas as interprocess communication, as well as encourage improvements in our taxonomy.

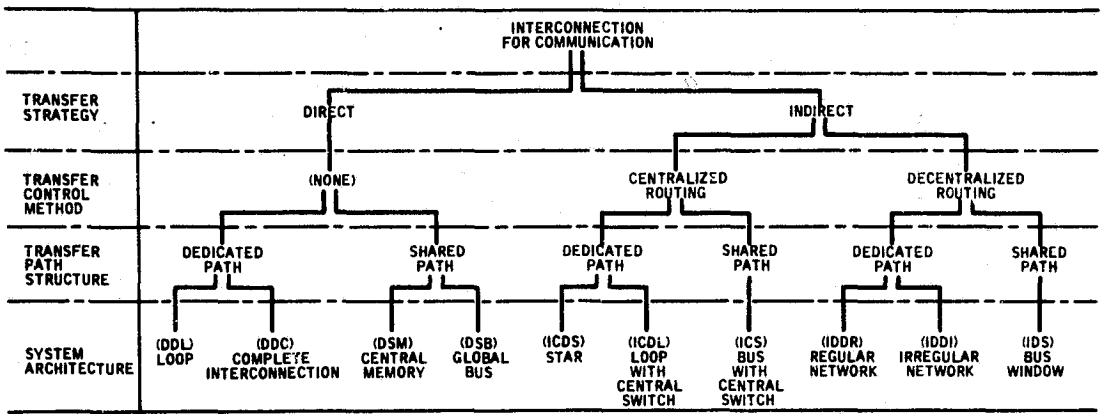


FIGURE 1. The taxonomy.

DESIGN DECISIONS—THE TAXONOMY

An interconnected computer system is the result of a series of design decisions, and the decision space can be considered to be a tree. Our model for this interconnection design process, shown in Figure 1, is a tree of four levels with alternative system architectures represented as leaves. The root of the tree is the decision to interconnect a number of computers for complete intercommunication. Below this are decision levels representing choice of message transfer strategy, the method of controling transfers, and choice of the type of path over which the transfer is to be made. The first two levels are concerned with strategic (policy) issues, and the third and fourth levels with tactical (implementation) issues.

The first strategic choice is between *direct* transmission of messages from source to destination, and *indirect* transmission in which an intervening operation is required. For purposes of the taxonomy, our criterion for distinguishing between these is the existence of one or more switching entities which make decisions for every message. Thus, intervening repeater circuits or storage elements are simply instances of paths, and do not affect the directness of the communication; but an intervenor that alters the message (e.g., address transformation), or an intervenor that routes the message onto one of a number of alternative output paths, is effecting an indirect communication. Another way to make this distinction is to determine whether control information is contained in or sent to the intervenor (e.g., address transformation tables). Decisions made by the sender (for example, which port to transmit on) and decisions made by the receiver (such as whether to accept a given message) do not affect the directness of communication.

If indirect communication is chosen, a further decision concerning the switching method must be made. This is shown at the second level of the tree. The alternatives are *centralization*, in which a single entity switches all messages, and *decentralization*, in which a number of intervenors are used.

The third level involves the choice of *dedicated* or *shared* message transfer paths. We define a shared path as one which is accessible from more than two points. In reality, there are at least three alternatives that may be distinguished: paths that are unidirectional point-to-point; paths that are bidirectional point-to-point; and paths that are bidirectional and visit more than two points. In the first case no contention can occur; but in the second, a rudimentary sharing exists, and hence contention can occur. In the third case, however, contention becomes a major consideration, so we define it as the "shared" path case and classify the other two as "dedicated" connections. We reiterate that the notion of "path" does not imply an implementation, and that both busses and memories can be appropriately used as message transfer paths. It should also be noted that paths which are redundant

for fault tolerance or bandwidth reasons are here considered logically singular.

The final level of the taxonomical tree comprises the leaf nodes representing specific system designs.

Before discussing the characteristics of the system types, we digress here to explain the various attributes which we feel are most significant, and to define our nomenclature. Our emphasis will be on the implementation-independent issues, and on the qualitative characteristics of systems. We avoid quantitative measures such as bandwidths and throughputs because they can only be representative of rapidly changing technologies and therefore must be evaluated within the constraints of a specific application.

For brevity, we will use sequences of capital letters to describe paths down the tree, with lower-case "x"s denoting "unmade" decisions. A direct, dedicated-path system is thus DDx, and so on.

SYSTEM CHARACTERISTICS

Modularity, the ability to make incremental changes in system capability, is a major characteristic to be considered in the design of a computer system. In instances where a specific design is to be configured for a variety of applications, it is often desirable to vary the number of processors according to the computational requirements of the particular problem. This requirement occurs both in homogeneous systems (having only a single processor type) and in nonhomogeneous systems. One measure of system modularity is the incremental cost of adding an element, such as a processor. If this cost is simply that of the element, then the system is indeed modular; but if the addition of the nth processor requires the addition of $n-1$ interconnection paths, then the system is not so modular. At the third level of the tree, some decisions involving this *cost-modularity* measure have already been made. For instance, selection between Direct (Dxx) and Indirect (Ixx) paths involves tradeoffs between the poorer cost-modularity of dedicated paths and the vulnerability of shared paths to bottlenecking.

Another measure of modularity is the degree to which the location and function of the incremental element is restricted. For instance, in a given design there may be particular places where a resource (processor, switch, or path) could be easily added to produce a specific performance increase, and other types of performance increase which are difficult or impossible to obtain in a modular fashion. Again in this case, decisions made in the progression to the third level of the tree have affected modularity. For instance, this *place-modularity* characteristic of Indirect Centralized (ICx) systems is poor with respect to the central switch. Replication of the central switch to achieve an increase in throughput changes the basic architecture to Indirect Decentralized (IDx). A place restriction can also occur in any nonhomogeneous Indirect (Ixx) architecture, since a special-purpose processor which must be added to the system usually cannot occupy a place that must perform a switching function.

Connection flexibility, a characteristic akin to modularity, must be considered for Ixx architectures. In Dxx architectures, the decision to add a processor requires no deliberation on the method of connection; it is fixed by the system type. For architectures allowing indirect communication, there can be alternatives with different costs. For example, in a geographically dispersed system, the cost of adding another processor at the location of an already existing one is significantly affected by whether the incremental processor must have its own paths to the rest of the system, or whether it can share the paths already installed.

Another important design characteristic is the cost of fault tolerance and the method by which a system is reconfigured to mask faults in processors and intercommunication paths. The first measure of goodness here is the effect of a fault. In designs where specific elements are shared (DSx, ICx, IDSx), a single failure of the shared element can completely halt system operation. In other designs the structure is such that failures have less catastrophic results. In addition to this *failure-effect* aspect, it is also necessary to determine the costs of alternative methods

of masking faults to allow operation in a degraded mode. (We maintain that a system operating in the presence of a fault is functioning in a degraded mode, regardless of whether or not the effect is observable using a performance measure.) Design decisions allow this *failure-reconfiguration* measure to range from excellent in systems requiring no overt reconfiguration and having minimal spare hardware, to very poor for those requiring that the entire intercommunication system be made redundant. A reconfiguration may even change the system's basic architecture. For instance, an ICx architecture that experiences a failure of the centralized switching resource may reconfigure to allow decentralized communication (becoming IDx) and thus avoid the cost of replicating all or part of the switch. This is an obvious area in which a hardware/software tradeoff exists, since a dynamic reconfiguration from one taxonomic architecture to another has significant software ramifications.

Inherent performance limitations, and the cost incurred in overcoming them, must also be considered in the choice of an architecture. The problem here is one of *bottlenecks* in resources, due either to a nonuniform flow of communication within a system, or to saturation of a shared resource. At the third level of the tree, DSx, IDSx and ICx architectures can be seen to be limited (i.e., to have poor cost-modularity for increased communications rates), since increasing their performance in areas served by the single shared path or switch requires significant hardware changes.

The nature and number of decisions that must be made to effect communications within a system are an important consideration. We call this attribute *logical complexity* and use the term to refer to the totality of decisions made during communications, whether made by source and destination processes, or by switching entities. Logical complexity is a characteristic that is significantly affected by the architecture, but its major effect is on software cost. From the standpoint of the architect, this makes logical complexity an almost unquantifiable element in tradeoffs, and the best that can be done is to make relative rankings for systems under consideration. In Ixx systems, the method by which the switching information is communicated is a major logical complexity issue. A "chicken and egg" puzzle pertains; the switching information that must be communicated (from somewhere in the system) to the switching resource comprises a message, but messages cannot be sent unless information for switching them exists in the switching resource. The magnitude of problems arising in the communication of switching information depends both on the system's type and on its operating environment. ICx systems are better in this regard than IDx systems; the more dynamic the processing environment, the more complicated the problem.

In addition to the characteristics just listed, which are largely determined by strategic decisions, there are a number of characteristics that are the result of implementation decisions. Among these are the physical dispersibility of the system, compatibility with commercial communication paths, message transfer delay between sender and receiver, and the cost of the interconnection paths. These, together with the strategy-dependent characteristics, are detailed in the following sections describing the architectural alternatives.

SYSTEM DESIGN TYPES

In the following paragraphs we discuss the significant features of each of the system species in our taxonomy. Our first attempts at these descriptions were made from a completely implementation-independent viewpoint, a perspective which we found untenable unless important design issues were to be omitted. Because of this, the discussions represent a compromise and certain observations are made both from a strict taxonomical viewpoint as well as from research and experience with actual designs. We also identify particular implementations of each interconnection type.

DDL—Loop

Loop architectures (Figure 2) have evolved from the data communications environment, and consist of a number of individual

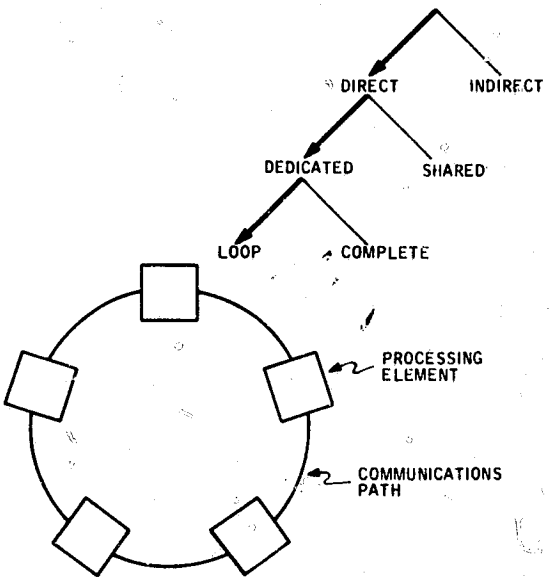


FIGURE 2. DDL (Loop).

processing elements (PEs), each of which is connected to two neighboring processing elements. The traffic in a loop could, in principle, flow both directions. In practice, the complexity of bidirectional traffic has constrained all the loops (of which the authors are aware) to only unidirectional traffic. In a unidirectional loop, one neighbor of a PE can be regarded as the source neighbor and the other as the destination neighbor. A given PE receives messages only from its source neighbor and sends messages only to its destination neighbor. Messages circulate around the loop from source to destination with intermediate PEs acting as relay or buffer units. DDL systems may allow one [FARM69] or more [REAM75] messages to circulate simultaneously, and messages are of either fixed or variable length [WEST72]. Some systems which have been referred to in the literature as loops contain a centralized switching function, and thus appear as ICDL in our taxonomy. Other systems are coupled loops with decentralized control, which we classify as IDDI.

Both the cost-modularity and the place-modularity of DDL systems are very good. An additional PE can be inserted anywhere in the loop with the addition of a single communication path, and the flow of messages is not significantly affected by its presence. The failure-effect and failure-reconfiguration characteristics of DDL systems are poor, however. A single failure in a path or a PE interface causes intercommunication to stop (at least between PEs separated by the failed resource). If reconfiguration to mask the fault is necessary, there must be a fully redundant path structure and some type of bypass switching in the PE interfaces. Reconfiguration from DDL to another structure is not an obvious option either, since the paths are unidirectional and the interfaces are relatively simple. The logical complexity of communications in a DDL system is low; a PE must only relay messages, originate messages and transmit them to a single destination, recognize messages destined for itself, and strip off messages according to the discipline. The bandwith of the single loop is, of course, a potential bottleneck as communication rates increase. In addition, some loop disciplines have the weakness that a single user, possibly with malicious intentions, can saturate the entire available bandwidth.

DDL architectures that have been proposed or implemented have almost all used bit-serial data links as the communication paths between PEs. This, together with the delay involved in relaying the messages, has resulted in significant increases in message transit times around the loop. In general, these systems have been designed for applications where reliability and performance constraints were not stringent. The primary goal of most designs has been the interconnection of geographically dispersed minicomputer systems to allow file and resource sharing. Thus, reconfiguration after failure has not been performed, nor has the message delay-time been a problem.

The best-known example of a DDL computer system is the Distributed Computer System at the University of California, Irvine [FARB72]. This system originally developed out of an interest in data communications rather than from a concern with distributed computing, although that emphasis was reversed early in the history of the project. The Distributed Computer System currently consists of five minicomputers and a number of peripheral devices looped around the Irvine campus. The loop (or

"ring") is bit-serial and operates at a data rate of 2.3 Mbs. A multiplicity of variable-length messages can circulate simultaneously. Fault tolerance is provided by a redundant loop and bypass switches.

The already good place-modularity inherent in loops has been enhanced in the Distributed Computer System by the incorporation of "soft" or "associative" addressing of messages. Rather than sending a message to a physical processor, it is sent to a logical process; the "Ring Interface" corresponding to the processor in which the destination process currently resides recognizes the address and accepts the messages. This allows communication to be independent of the number of processors in the system, and of process/processor assignments. This idea has also been incorporated into other distributed architectures.

A loop version of the CAMAC data multiplexing system has been defined for both bit-serial and byte-serial transfers [AEC73]. Originally designed for nuclear laboratory instrumentation, the CAMAC loop is appearing in distributed computing schemes, although its protocol is not well suited for such use.

DDC—Complete Interconnection

The DDC architecture is perhaps the conceptually simplest design type in the taxonomy. In it (Figure 3), each processor is connected by a dedicated path to every other processor in the system, and messages between processors are transferred only on the path connecting them. The source processor must choose the path to the destination processor from the alternative paths available, and all processors must be equipped to handle incoming messages on a multiplicity of paths.

The most significant characteristic of DDC systems is their poor cost-modularity. The addition of the nth processor to a DDC system requires not only the addition of n-1 paths between it and the other processors, but also, all processors in the system must have facilities for accepting the incremental PE as a data source. Thus, their interfaces must have at least M-1 ports, where M is the maximum size of the system. Alternatively, it must be possible for all processors to accept extra connections when the number of PEs must be increased beyond the number of available ports. Place-modularity of DDC systems is good, as are failure-effect and failure-reconfiguration characteristics. The DDC architecture is one which can be easily degraded in the event of a failure without changing its interconnection class—a failed processor, or one of the two processors terminating a failed path, can simply be disconnected from the system. In addition, reconfiguration to an Ixx system could be used in event of a failed path if the software cost and increased message transit time incurred were acceptable. DDC systems have no obvious bottlenecks, and their logical complexity is relatively low. It should be noted, though, that the architecture forces a location-addressing policy on interprocess communication, since switching within the processors and message relaying activities would put a design into the ICDx or IDDx categories.

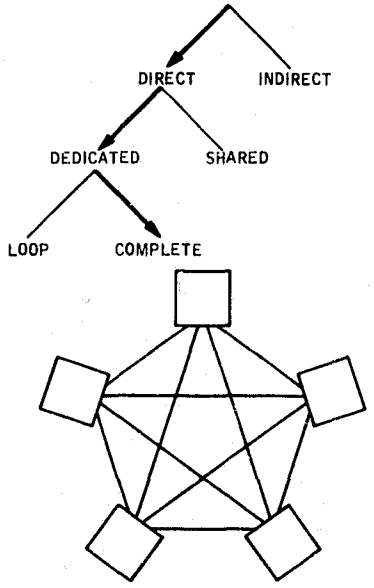


FIGURE 3. DDC (Complete interconnection).

DDC systems may be geographically either localized or dispersed, although there are few examples of either case. The best-known instance of a localized DDC architecture is a fully connected version of the IBM Attached Support Processor System [IBM],

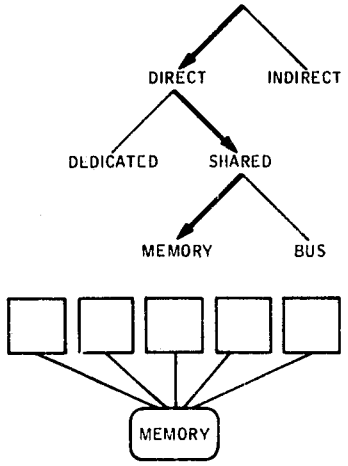


FIGURE 4. DSM (Multiprocessor).

in which up to four System/360 or /370 computers may be linked through I/O channel couplers.

Virtually all extant examples of geographically dispersed DDC systems are small ($\leq$3 PEs), and appear to be ad hoc interconnections of formerly existing computer installations, as exemplified by the fully connected configuration of the MERIT system [BECH72]. MERIT consists of two IBM 360/67s and a CDC 6500 located on three separate Michigan University campuses, and connected by common carrier lines. The MERIT system design does have the potential for less fully connected configurations (such as IDDI) to reduce communications costs.

DSM—Multiprocessor

Certainly the most common way to interconnect computer systems is the DSM or multiprocessor architecture (Figure 4), in which two or more processors communicate by leaving messages for one another in a commonly-accessible memory. The key characteristic of DSM architectures is that the memory is, or can be, used as a path rather than solely as storage.

The place-modularity of DSM systems is very good; it is possible to add processors arbitrarily (since the processors are not topologically distinguished), and it is also possible to increase the in-transit message capacity of the path simply by increasing the size of the memory. The cost-modularity of DSM systems depends almost completely on the path structure by which the processors access the memory system. If each processor is provided with a direct path, then cost-modularity can be poor, since an incremental processor can possibly bring the total to greater than the number of available memory ports. Alternatively, if the memory is accessed through a single bus with a suitable allocation mechanism, cost-modularity can be very good. A DSM system is quite vulnerable to a bottleneck in which the memory's bandwidth becomes a restriction on communication rates. Cost-modularity is poorer in this case, as it is expensive to increase bandwidth of the memory or the access path. Logical complexity of DSM systems is quite low. The failure-effect and failure-reconfiguration characteristics of DSM systems are good in the case of processor failures, but poor in the event of failure of the central memory unit (or of a shared access bus). There is also a software failure-effect problem because processors normally have unrestricted access to the central memory, thus faulty or malicious software can prevent or damage message transactions to which it is not a party.

Almost every implementation of a DSM system has occurred because the designer(s) wished the memory to be shared as a storage place for programs and data—use of the memory as a communication path has almost been a side effect. In implementing this multipurpose sharing, it has been found that the systems' performance has increased more slowly as the number of processors increased and, in general, systems consisting of more than about four processors have not been cost-effective. The reason for this has been the extreme contention for memory bandwidth when the (functionally) single memory must serve for all purposes. The bandwidth required for communications alone is, however, unlikely to cause bottlenecking in a memory solely dedicated to this function.

An example of a contemporary multiprocessor is the Carnegie-Mellon C.mmp [WULF72], which allows up to 16 processors to share up to 16 memory modules through a crossbar switch. Currently, five PDP-11/20

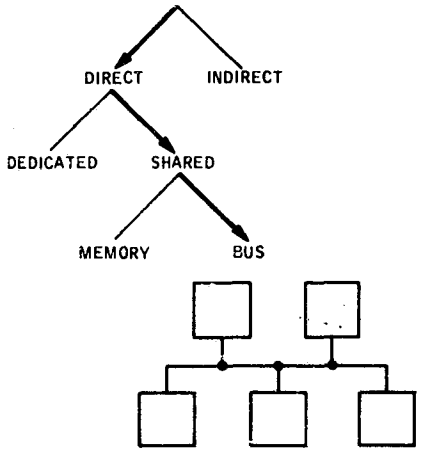


FIGURE 5. DSB (Global bus).

processors are operating with negligible interference.

The surveys of Miller, et al. [MILL70], and Enslow [ENSL74] include a wide variety of both commercial and aerospace DSM machines.

DSB—Global Bus

The DSB architecture, shown in Figure 5, comprises a number of processing elements interconnected by a common, or global, bus. Access to this bus is shared among the processors by some allocation scheme, and messages are sent directly from the source PE onto the bus, to be recognized and accepted by the proper destination(s).

Both the cost- and the place-modularity of DSB systems are good with respect to the PEs. Depending on the choice of bus allocation scheme, it can be possible to add a processor to the system in any position with little or no effect on the other PEs. Cost- and place-modularity of the communications path are poor, however. It is not possible to increase the bandwidth easily as needed, nor is it often possible to increase performance only where needed. Rather, to increase performance it is usually necessary to change the implementation of the entire bus or to replicate it, alternatives which have significant design impact on the bus interfaces of all PEs in the system. Similarly, the failure-effect and failure-reconfiguration characteristics of DSB systems are very good with respect to the processors, and poor with respect to the bus. For PE failures, the DSB architecture requires no overt hardware reconfiguration activity to continue operation as a DSB system. Given reasonable care in the design of the bus interfaces, processor failures will have little effect on system operation. Failures of the bus, however, are inevitably catastrophic, and replication is required if the DSB architecture is to be retained after configuration. The global bus is, of course, a potential bandwidth bottleneck.

Much of the current interest in DSB systems has occurred in the aerospace environment, where both serial and parallel paths are being used. In these applications, the place- and cost-modularity characteristics of DSB systems are particularly advantageous, and allow flexibility in configuring systems for specific applications, including both homogeneous and nonhomogeneous instances of the same basic system. The failure-reconfiguration characteristics of the DSB architecture are useful for these applications, too, because most aerospace applications are of a real-time nature and, therefore, excessive reconfiguration delays must be avoided. Serial bussing for the communication path seems to be the more popular approach, primarily because the system applications tend to have high costs associated with physical wiring, a need for physical dispersibility over electrically long distances, and relatively low data rates. Replication of the communications path is the predominant technique for mitigating both the bandwidth restriction and the fault vulnerability of the shared bus.

Most of the aerospace DSB-like architectures have been hybrids with respect to our taxonomy [SYMS72], [ANDE73]. However, the military services are beginning to establish data multiplexing standards [USAF73] that will require the use of DSB approaches in future aerospace systems (although these initial standards leave something to be desired, from a distributed computing standpoint).

A pure DSB architecture is typified by one utilizing word-wide busses [JENS75], and a philosophical derivative of it (using a bit-

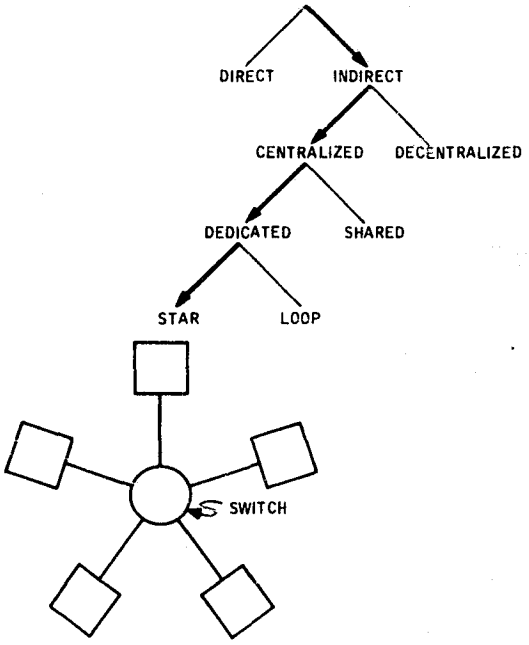


FIGURE 6. ICDS (Star).

serial bus), which has been constructed by the authors for the US Navy (for which there are as yet no public references).

DSB is also a popular line-sharing discipline in industrial and laboratory automation applications [ARON71], as exemplified by the CAMAC [COST72] and IEEE standard 488-1975 (originally Hewlett-Packard [KNOB 75]) data-multiplexing systems.

ICDS—Star

ICDS systems (Figure 6) consist of a central switching resource to which a number of processors are connected, each by a functionally single, bidirectional path. Messages are exchanged among the PEs using the central switch as an intermediary; it is the apparent destination and source for all messages. The function of the switching resource is usually seen as "insulating" the processes running on a given PE from physical knowledge of the system, and protecting them from each other.

The ICDS architecture has most features in common with the DSx architectures because both have shared message transfer facilities. Its cost- and place-modularity are good with respect to the PEs, and poor with respect to the central resource. Similarly, failure-effect and failure-reconfiguration characteristics are good for the PEs, and poor with respect to the switch. Bottlenecking in the switch is a potential problem. The connection flexibility of ICDS systems is poor because incremental PEs must always be provided with individual paths to the central switch. The logical complexity of ICDS systems is moderate. Sufficient information (e.g., routing tables) must be provided within the switching resource to allow communications to take place, but the fact that there is only a single copy of this (usually dynamic) information simplifies its handling during reconfiguration. The poor failure-effect and failure-reconfiguration characteristics of the central resource are extended to one of the PEs in the system if switching information is maintained outside the resource (i.e., where the switch uses address translation to accomplish message routing, and the mapping registers are under control of one of the PEs). ICDS architectures are quite common. One example is IBM's Network/440 [MCKA70], in which remote System/360 user nodes are connected over leased lines to a 360/91 central controler.

ICDL—Loop with Central Switch

In a manner analogous to Direct, Dedicated path (DDx) systems, the direct connections required for ICDx architectures can be implemented either in the ICDS star pattern or in a loop, which we call ICDL (Figure 7). In such a system, messages are placed on the loop by senders, removed for an address-mapping operation by a central switching element, then replaced on the loop properly addressed to their intended destination.

ICDL systems share characteristics with both the ICDS and the DDL organizations. Their failure characteristics are those of DDL with respect to the data paths, and those of ICDS with respect to the central resource. Connection flexibility is improved over the star in that the incremental PE need be connected only to its physically nearest neighbors, not to a possibly remote control element. As in DDL systems, bottlenecks and vulnerability to malicious users

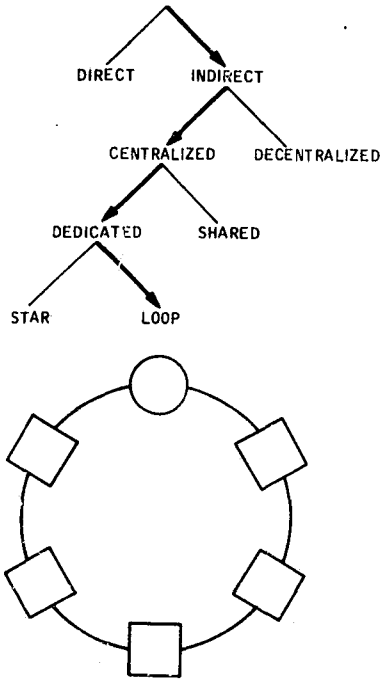


FIGURE 7. ICDL (Loop with central switch).

are potential problems, but since these are risks inherent also in the centralized switching of the ICx approach, they do not necessarily weaken the system over ICDS. The logical complexity of ICDL systems is moderate, though slightly increased over ICDS due to the additional demands the loop discipline places on the PE interfaces.

An example of the ICDL architecture is SPIDER, an experimental data communications system interconnecting eleven computers at Bell Laboratories in Murray Hill, N. J. [FRAS75]. In this system, multiple fixed-size data messages circulate over 1.544 Mbs common carrier lines; currently there are three loops, but only one is in use. The central switch for all loops is a minicomputer. Any computer in the loop (except the central switch) can be switched to receive-only in case of failure.

ICS—Bus with Central Switch

The ICS architecture shown in Figure 8 is functionally equivalent to ICDS, with the major exception that the processors are not individually connected to the switching resource but, instead, share a path by which to access it. Thus, when a PE wishes to transmit a message, it must first acquire the bus, then transmit the message to the switch. From the switch, the message is retransmitted over the (functionally) same bus to its proper destination. (This retransmission is the characteristic by which ICS systems can be distinguished from DSB organizations and from DSM systems using a single bus to memory.)

As might be expected, the characteristics of the ICS architecture are similar to those of ICDS systems. By the failure-effect measure it is poorer, since the access path is no longer replicated once for each processor. The existence of the shared path to the switch need not contribute significantly to bottlenecking, however, since it is quite feasible to balance its performance with that of the switch. Then, as long as the two saturate at the same time, or the switch saturates first, the bottlenecking risk is not increased over the ICDS organization. The cost-modularity of the ICS system is influenced positively by the fact that the incremental processor need only be connected to the shared bus, not directly to the (possibly remote) switch. As mentioned in the discussion of direct, shared bus (DSB) systems, certain bus designs make this connection cost very low.

The low popularity of ICS systems may be inferred from the fact that there is no com-

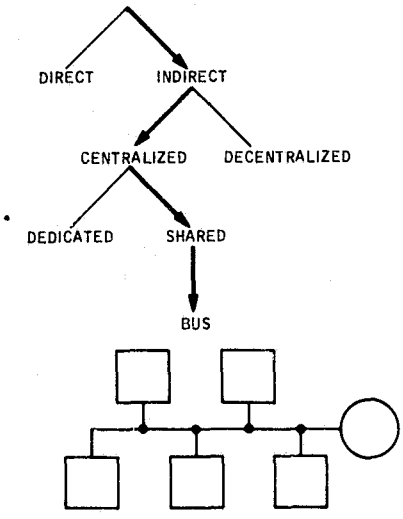


FIGURE 8. ICS (Bus with central switch).

mon name for them. The designs that do exist, however, use processors as the switching resource, with processors dispersed over geographically short, but electrically long distances. Although nothing intrinsic to the architecture prevents the use of hardware (such as a crossbar) for the switch, the speed implications on the bus, if balance is to be achieved, probably preclude its use as a practical matter. As mentioned previously, the shared bus is not normally available from commercial carriers, so ICS systems are restricted geographically.

An ICS system has been designed by Hughes for the US Navy [ROWA74]. The centralized switching resource is a special-purpose processor called the Network Manager, which interfaces with a number of functional nodes over one or more shared busses. In addition to handling interprocessor communication, the Network Manager also provides a variety of other executive services for the nodes. Because the system is physically localized, the bus(ses) to the Network Manager are bit-parallel.

The ALOHA system [ABRA73] is a unique ICS computer network which uses a radio "bus." Hardwired and incoherent light links are also accommodated. A 24K-baud full duplex radio channel connects the central IBM 360/65 to remote terminals, which include a number of minicomputers.

IDDR—Regular Network

The IDDR organization (Figure 9) comprises a number of PEs interconnected with dedicated paths and having identical neighbor relationships (except, perhaps, at the boundaries). In the figure, each PE has "left," "right," "above," and "below" neighbors, although other geometries have been proposed. Messages are routed through the network from source to destination with each intervening PE determining which of its alternative neighbors should be the next recipient of the message. (It may be noted that DDL is a special IDDR case in which each PE has two neighbors and there is no switching decision to be made.)

The modularity and failure characteristics of IDDR systems are significantly and negatively affected by the requirement for absolute regularity. It is not possible to add only a single PE or path; rather, the size of the increment depends on the number of PEs in the system, the number of neighbors per PE, and the interconnection pattern. Thus, both cost-modularity and place-modularity are extremely poor. There is no connection flexibility. Logical complexity of the design is moderate; although many switches exist, the regularity of connection simplifies routing. The failure-effect measure of IDDR systems is moderate-to-good, depending on the method by which message routing is performed. A single PE or path failure does not stop communication entirely in any case, nor need it stop any messages to which the failed element is not a party. Failure-reconfiguration of IDDR systems is exceedingly poor, with 100% sparing of PEs and paths being required if the basic structure is to be unaltered after reconfiguration. If reconfiguration to an irregular structure (IDDI) is acceptable, however, this shortcoming is eliminated.

The elegance of IDDR structures has caused significant academic interest but, to the authors' knowledge, their practical

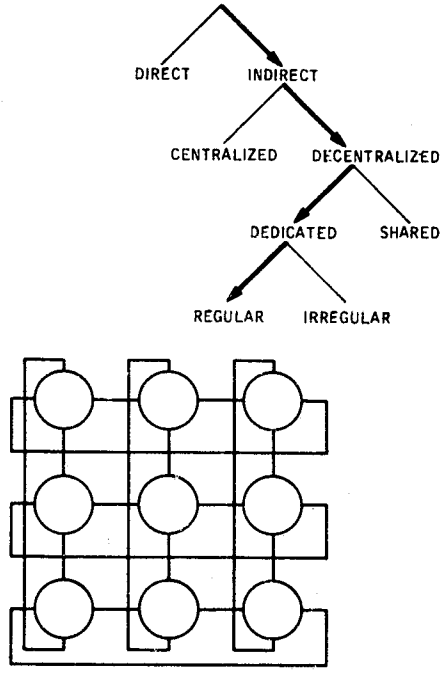


FIGURE 9. IDDR (Regular network).

difficulties have prevented actual implementations. Paper designs have usually been for four-neighbor (rectangular) systems, with either busses or memories proposed as paths. One nonrectangular IDDR system is TREE, designed at the US Navy Postgraduate School [GOOD73]. This machine comprises a number of PEs connected as a tree. A PE may communicate with its superior or any of its subordinates in the hierarchy; because there is only one path between any two nodes in a tree, the logical complexity of the message routing is not severe. TREE was envisioned primarily as being physically centralized, but could instead be implemented with common carrier lines for greater dispersal.

IDDI—Irregular Networks

The distinction between IDDR and IDDI (Figure 10) systems is simply that consistent neighbor relationships are not required for IDDI. Thus, a given PE may have from one to an arbitrary number of neighbors with which it communicates. Many of the system characteristics vary with the degree of interconnection regularity. Place-modularity of IDDI systems tends to be extremely good, with both processors and paths added as, and where needed. Cost-modularity is similarly good, with incremental PEs being architecturally required to have only one (or two in loop-oriented approaches) connections to the rest of the system. Connection flexibility is another advantage, with connections allowed to any PE in the system. The more irregular the interconnection pattern, the more the system's failure-effect and failure-reconfiguration characteristics can be enhanced by providing a multiplicity of potential paths between PEs. Likewise, irregularity improves the extent to which reconfiguration can be performed without departing from the IDDI category. The logical complexity of IDDI systems is usually very high; at each switch, routing decisions must be based on knowledge of the overall system topology. Due to the good place-modularity, bottlenecks are not a likely problem.

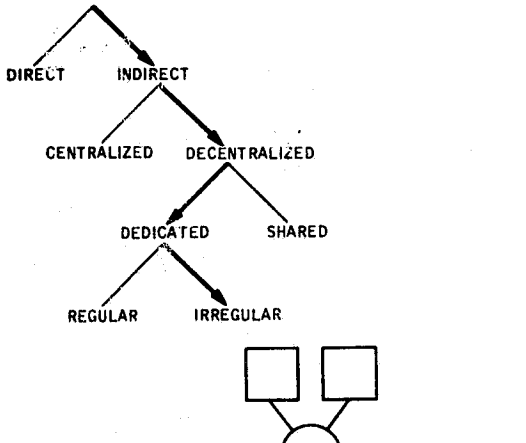


FIGURE 10. IDDI (Irregular network).

The dominant current application of IDDI interconnection is to geographically dispersed computer networks. In such systems the paths are supplied by a common carrier and the switching is done by processors dedicated to that function. The (significant) cost due to logical complexity, and represented by the switch processors, is incurred to minimize the number of high-cost interconnection paths required. Routing algorithms are usually inelegant, especially for the less regular systems, and ad hoc solutions are apparent. It is the authors' feeling that hardware switching facilities are ruled out by the complexity of the decisions to be made, and that the current practice of using processors for this function must necessarily continue.

Most systems commonly called "computer networks" are IDDI structures. These are discussed in the surveys by Schneider [SCHN73], Abramson [ABRA73a], and Rustin [RUST72], and in the bibliography by Blanc, et al. [BLAN73].

A more regular sort of IDDI system is represented by the coupled loops of Pierce [PIER72], [COKE72], [KROP72], intended for telecommunications. A number of independently controled DDL "local" loops are coupled together, directly and/or by "trunk" loops (which may themselves be

DDL). A variety of alternate, redundant, and bypass-switched loops is proposed to provide failure protection.

IDS—Bus Window

In IDS architectures, an example of which is shown in Figure 11, access to the switching resources is via a path shared by multiple PEs. Switching is performed by more than one resource, and messages may be retransmitted onto the path from which they were received, or onto another.

The modularity characteristics of the IDS architecture are similar to those of the IDDI structure. The failure-effect and failure-reconfiguration characteristics are poorer, however, because multiple PEs and switches can be affected by the failure of a single path. Also, systems of this type are not easily dispersible due to the shared busses.

Digital Equipment Corporation manufactures a device called the DA11-F Unibus® Window to facilitate the implementation of IDS architectures [FITZ73], [DEC75]. Similar mechanisms have been designed at Carnegie-Mellon University for their Computer Module System [FULL73], and at BBN for their PLURIBUS IMP [HEAR73] (which we classify as a hybrid, as discussed in the Conclusion of this paper). All of these units provide a bidirectional path between the busses of two minicomputers. Blocks or segments in the source-address space are translated to the destination-address space. A number of these interfaces may be used to construct hierarchies of processing elements. Simulation and programming experience by users of all three mechanisms indicate that the logical complexity of this approach grows rapidly as the number of translation levels increases, and as the translation binding becomes more dynamic. This type of interconnection is also subject to deadlock [BELL73], [CHEN74], unless designed and used with great care.

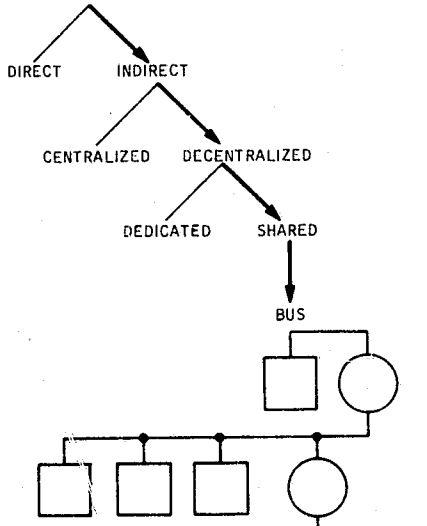


FIGURE 11. IDS (Bus window)

FUTURE DIRECTIONS

Current activity in distributed architectures is undeniably more on paper than in hardware, but there do seem to be trends emerging in designs for several application areas.

DDL organizations seem to predominate in designs where their excellent modularity and compatibility with common-carrier data paths can be used to advantage, and where their poor failure characteristics and long message transit-times are not a problem. Typically, the DDL designs are used to connect multiple minicomputer systems, dispersed over electrically long distances in university, research laboratory, or industrial automation environments. The individual PEs are usable both as stand-alone systems and with other PEs for resource (file, peripheral) and load sharing.

The DSB organization with serial bussing is becoming the dominant architecture in real-time control environments. These applications typically have a number of iterative, sampled-data control loop functions loosely connected by a requirement for information exchange and centralized monitoring. The environment occurs in both the conventional process-control type application and in aerospace applications such as integrated avionics processing. In such systems, the good place-modularity for processors in DSB organizations is particularly

valuable. DSB is chosen over DDL because simple redundancy of the global bus allows fast and almost automatic reconfiguration after failures, and because of the reduced message transit-time.

In environments where sophisticated interprocess communication is a requirement, the ICx organizations are a common choice. Compared to IDx, the lower logical complexity of centralized switching is believed to outweigh its failure-effect and failure-reconfiguration disadvantages. There is a great variety of system designs, particularly in the ICDS category, with varying levels of performance in the central switch area. It is the feeling of the authors that as understanding of interprocess communication matures, movement toward IDx structures may occur.

The fourth "dominant" system is IDDI, which is uniquely suited to applications requiring interconnection of relatively large computers over geographically long distances. In such systems, the high interconnection cost necessitates that only large computers be involved. Furthermore, significant processing power should be dedicated to the switching function in order to minimize those interconnection costs. At this time, such applications have used almost exclusively ad hoc solutions to controling the complex IDDI structures, with the majority of the efforts going toward more fundamental problems of intercommunications protocols and experiments in load sharing. It is hoped that future work in such areas as routing algorithms and recovery techniques for IDDI structures will complement interprocess communication research currently being conducted with ICDx structures, to the ultimate benefit of both.

In our opinion, the remaining five system types each have one or more significant weaknesses not sufficiently compensated for by strengths that are useful in real-world applications. DDC systems have poor modularity and high interconnection costs, to the extent that large systems are impractical. DSM architectures, from an intercommunications viewpoint, have little to offer over DDL or DSB structures except an ability to communicate large messages quickly by use of pointers. (They will undoubtedly continue to exist for their other advantages, which are not related to communications.) IDDR and IDS organizations both incur significant logical complexity with few compensating advantages. Further, IDDR has the poorest modularity and failure characteristics of any of the organizations.

CONCLUSIONS

The notion of a taxonomy requires both that all species be clearly identified, and that the methods used to make distinctions be clear and unambiguous. Our discussion here has failed to achieve either of these goals completely, but has, we hope, been a worthwhile beginning. All species (systems) are not clearly identified; there are many hybrids using combinations of our leaf-node architectures. The PRIME system at the University of California, Berkeley [QUAT72] is physically a combination of DSM (multiprocessor) and ICDS (star), although logically it is purely ICDS since all interprocessor communication goes over the External Access Network, rather than through the shared memory. Honeywell's DP/M system for the US Air Force [ANDE73], with its global and local busses, uses two levels of DSB. The BBN PLURIBUS IMP for the ARPANET [HEAR73] is structurally an amalgam of DSM (although by convention there is no interprocessor communication using the local memories), IDS (for accessing shared memory), and ICS (the Pseudo-Interrupt Device).

Neither are all distinctions crystal clear; we have classified systems on the basis of "dominant" features, and to some degree even on our understanding of the intent, as well as on prima facie evidence. Our goals in attempting to construct a taxonomy will be satisfied, however, if two effects occur. First, we hope to stimulate a refinement of this approach or the development of a better one. Secondly, we hope to have contributed to a common ground for discussion in the meantime.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the valuable comments and criticisms of the Editors and the referees of COMPUTING SURVEYS. We are also indebted to many colleagues who informally refereed this paper. Special thanks go to Kenneth J. Thurber for his suggestions regarding the figures, which substantially improved the readability of the paper.

REFERENCES

[ABRA73] ABRAMSON, NORMAN, "The ALOHA System," *Computer-communication networks*, Prentice-Hall, Englewood Cliffs, N. J., 1973. (ICS)

[ABRA73a] ABRAMSON, NORMAN; AND KUO, FRANKLIN F. (Eds.), *Computer-communication networks*, Prentice-Hall, Englewood Cliffs, N. J., 1973. (Survey)

[AEC73] AEC/NIM AND ESONE, CAMAC *Serial System Organization*, National Tech. Information Service, TID-26488, December 1973; *Addendum and Errata*, May 1975. (DDL)

[ANDE73] ANDERSON, GEORGE A., "Interconnecting a distributed processor system for avionics," *Proc. Symposium on Computer Architecture*, December 1973, IEEE, N. Y., 1973. (Hybrid)

[ARON71] ARONSON, RICHARD L., "Line-sharing systems for plant monitoring and control," *Control Engineering*, January 1971. (Survey)

[BECH72] BECHER, WILLIAM D.; AND AUPPERLE, ERIC M. "The communications computer hardware of the MERIT Computer Network," *IEEE Trans. Communications*, June 1972. (DDC)

[BLAN73] BLANC, ROBERT P.; COTTON, IRA W.; PYKE, THOMAS N. JR.; AND WATKINS, SHIRLEY W. *Annotated bibliography of the literature on resource sharing computer networks*, National Technical Information Service, COM-73-50750, September 1973. (Bibliography)

[CHEN74] CHEN, ROBERT C., *Bus communication systems*, University Microfilms Dissertation 74-20493, 1974. (Control and Communication)

[COKE72] COKER, C. H., "An experimental interconnection of computers through a loop transmission system," *Bell System Tech. J.*, July/August 1972. (IDDI)

[COST72] COSTRELL, LOUIS, CAMAC—*a modular instrumentation system for data handling; revised description and specification*, National Technical Information Service, TID-25875, July 1972. (DSB)

[DEC75] DIGITAL EQUIPMENT CORPORATION, *PDP-11 peripherals handbook*, 1975. (IDS)

[ENSL74] ENSLOW, PHILLIP H. (ED.), *Multiprocessors and parallel processors*, John Wiley and Sons, New York, 1974. (Survey)

[FARB72] FARBER, DAVID J.; AND LARSON, KENNETH C. "The system architecture of the distributed computer system—the communications system," *Proc. Symposium on Computer-Communications Networks and Teletraffic*, April 1972, Polytechnic Press, Brooklyn, New York, 1972. (DDL)

[FARM69] FARMER, W. O.; AND NEWHALL, E. E. "An experimental distributed switching system to handle bursty computer traffic," *Proc. ACM Symposium on Problems in the Optimization of Data Communications*, October 1969, ACM, New York, 1969. (DDL)

[FITZ73] FITZGERALD, BRIAN, "Standard interfaces promote new minicomputer networks," *Electronics*, September 13, 1973. (IDS)

[FRAS75] FRASER, A. G., "SPIDER—An experimental data communications system," *Proc. Internat. Conf. on Communications*, June 1974, IEEE, N. Y., 1974. (ICDL)

[FULL73] FULLER, SAMUEL H.; SIEWIOREK, DANIEL P.; AND SWAN, RICHARD J. "Computer modules: an architecture for large digital modules," *Proc. Symposium on Computer Architecture*, December 1973, IEEE, N. Y., 1973. (IDS)

[GOOD73] GOODWIN, RICHARD J., *A design for distributed-control multiple-processor computer system*, National Technical Information Service, AD-772 883, December 1973. (IDDR)

[HEAR73] HEART, F. E.; ORNSTEIN, S. M.; CROWTHER, W. R.; AND BARKER, W. B. "A new minicomputer/multiprocessor for the ARPA Network," *Proc. AFIPS 1973 National Computer Conf.*, AFIPS Press, Montvale, N. J., 1973. (Hybrid)

[IBM] IBM, *IBM System/360 and System/370 Attached Support Processor Version 3 Asymmetrical Multiprocessor System: general information manual*, GH20-1173. (DDC)

[JENS75] JENSEN, E. DOUGLAS, "A distributed function computer for real-time control," *Proc. Symposium on Computer Architecture*, January 1975, IEEE, N. Y., 1975. (DSB)

[KNOB75] KNOBLOCK, DARYL E.; LOUGHRY, DONALD C.; AND VISSERS, CHRIS A. "Insight into interfacing," *IEEE Spectrum*, May 1975. (DSB)

[KROP72] KROPFL, W. J., "An experimental data block switching system," *Bell System Tech. J.*, July/August 1972. (IDDI)

[MART72] MARTIN, JAMES, *Systems analysis for data transmission*, Prentice-Hall, Englewood Cliffs, N. J., 1972. (Digital Data Communication)

[MCKA70] MCKAY, DOUGLAS B.; AND KARP, DONALD P. "IBM Computer Network/440," *Computer Networks—Proc.*

APPENDIX H
SOFTWARE TECHNOLOGY

This appendix deals with the concepts of modularity, higher-order language (HOL), and software verification, validation, and support. The first sections deal with a bottom-up approach. Latter sections deal with a top-down approach. Each of these concepts represents an essential consideration in the design of an effective and reliable software system. The major thrust of this study is an avionics application. Concepts are discussed, however, from a more general point of view whenever possible. Honeywell recognizes that avionic software has the most demanding standards of reliability. Additionally, in some environments avionic software may have demanding real-time, event-oriented requirements. Generally, however, the software runs in a very fixed iterative environment without demanding real-time events driving the system.

DEFINITION OF THE BASIC TERMS

Modularity is a term for a programming philosophy where the total system is organized by partitions. A well-defined partitioning of a system will ensure system modularity. This partitioning begins with identifying the project tasks. Each task results in a separate and distinct program module. Each module should have well-defined inputs and outputs, and should have clean interfaces with the other modules of the system. Modules can be independently debugged and validated. System errors and deficiencies should be traced to individual modules to limit the scope of error analysis.

A higher-order language (HOL) is a computer language with relatively powerful primitives oriented to specific applications. The traditional level of programming, especially in avionics, has been at the assembly-language level. This approach usually requires that the primitives of the assembler used by the programmer map nearly one-to-one to the primitives of the hardware implementation of the machine. An HOL relative to this basis, then, includes languages such as FORTRAN, PL/I, CMS-2, etc. However, each of the contemporary HOLs differs in the degree of being a true HOL. This is because of structural properties, as well as the requirement that an individual HOL be compared to a specific, intended range of applications. The avionics applications are such a range. An avionics HOL offers primitives oriented to the functions natural in system design and implementation.

Verification and validation of systems involve testing the system to ensure that it performs as specified. Often this process is associated with exercising the implemented system. Rigorous tests of programs are not feasible because of the large numbers of combinations. Hence, verification and validation involve analyses of system behavior to arrive at meaningful testing situations.

RELATION OF MODULARITY, HOL, AND VERIFICATION AND EVALUATION

The sections below separately discuss modularity, language, and verification. However, the three areas are very closely related. This dependence is evident in later sections. Modularity relates to language structurally. Modularity relates to verification and validation because of the structural influences on the debugging and testing problem. Finally, language relates to verification and validation because of the support level and because of the level of the program source description in the implemented system.

DISCUSSION APPROACH

Wherever possible, the study will be approached as follows. In each of the sections, existing methods will be described and/or evaluated. Then the state of the art will be discussed and recommendations made on procedures which appear useful in the avionics environment will be made.

MODULARIZATION

The major concepts of modularization discussed are:

- Work division
- Functional building blocks

Also discussed is the problem of modularity and its relationship to HOLs.

Work Division and Assignments

It is traditional practice that when software systems are designed and implemented, the staffing of the project is aligned with the system structure. Modularization is thus at least necessary for giving pieces of the system to individuals or groups for implementation. For extremely large software systems, this practice is necessary to manage the total project. In smaller systems, the primary reason is to divide the labor. Thus, in smaller systems, modules may not be well defined. Considerations in the larger systems tend to force a somewhat manageable division of work.

The end product of the initial design phase of an avionic system is the functional specification. The functional specification describes what the system will do, in terms of expected inputs, desired outputs, and a top-down definition of the chosen algorithms. Often, a functional specification will break the system down into various functional modules, along with elaborate descriptions on how these modules will fit together to fulfill the system's

function. Ideally, this organization would be functional, but occasionally a functional specification will be limited to certain subsets of the system environment. When this happens, the interfaces toward the outside system can become arbitrary, weakening the functionality of the specification.

Under any definition of modularity, intermodule communication is extremely important. If a division of labor exists in the system implementation, then interpersonal relationships are also involved. Modularization decisions often produce modules which share data with other modules, thus producing complex and subtle dependencies between modules. The usual method of managing a project is to precisely define the limitations of modules and the acceptable methods of information transfer between modules. However, more often than not, especially in larger systems, the initial specifications of a system do not remain intact throughout the development period. This usually results in the rules for module-to-module interface being compromised and, thus, less precise.

Almost all software systems use a functional hierarchy concept; for example, avionic systems have monitors or executives. In such cases, a kernel within the system supports the basic functions upon which the remainder of the system is built. The fact that such organizations are considered a hierarchal basis relates to the idea that an executive, etc., is usually considered to be "closer" to the hardware and, hence, a lower level module in the hierarchy. Within given levels of a system element, certain hierarchies may also exist. Few contemporary software modules embedding major system elements define such elements homogeneously. Accordingly, avionic systems are structured layers comprising a hierarchy. However, the relationships between layers or elements are often indefinite. Different modularity criteria may have been applied in one module than in another. Larger systems can have the intended hierarchy so completely fragmented that no hierarchy or ordering can be recognized. This is especially true of systems subjected to a large number of design alterations, or having been frequently changed and patched during the debugging phases.

Functional Building Blocks

As mentioned above, a criterion of modularization in many specifications is based on functionality. Each component or module is associated with specific functional duties within the total system. At the lowest level, such specific functional operations can be considered primitives to the system above them. To the extent that the components of the system exist at various levels, it is possible to consider the modules or primitives of lower levels as building blocks. This concept of using modules as building blocks can exist both at the language-source level (for example, in a higher-order language), and at the object level (machine code).

A method of system decomposition directly dependent on a source code convention is a property of a particular source language. The modularity implications of using an HOL and vice versa are discussed later. However, the notion of identifying specialized sections of source-language code can be considered as modularization as it involves using such source-code sections as building blocks. This usage can be based on some functional or decompositional convention. The usage may simply be to decrease source verbosity to obtain a more compact program source while maintaining access to all features of the particular HOL. Examples of such source-level building blocks are macros, and addressable data and procedure blocks.

A macro is a parameterized prototype for a string of source characters. The capability for defining and using macros exists in many of the contemporary assemblers in commercial software. When used along with assembly language coding, the macro provides the capability for a programmer to invoke extended code sequences through simple statements asking for macro expansions. The ability to do this is equivalent to having a source module (the macro prototype) which performs a given function (produces the extended code sequence).

An addressable data or procedure block is an element of an HOL which can be retrieved from a library of such blocks. An example of this in a commercial application might be the existence of a standard library of COBOL data divisions pertinent to a class of applications. Obviously, this type of source-level modularity is closely related to the syntactical structure of a particular HOL. In a block-oriented language such as ALGOL or PL/1, procedures and structures global to the outer block constitute building blocks as if the source existed in a block containing the program. Such a concept can be used as a basis for specialized applications where the specialized program has a set of global procedures or structures available.

Object-level modularity is the traditional approach to breaking down an avionic system. An object code is that code upon which the physical machine operates. Modularity at this level results either from program development directly at the object level via very low-level languages, or from a mapping directly from source-level modularity, such as FORTRAN. The functional decomposition imposed on the avionics system is carried down to a partitioning of the object program along the same lines. Certain common functions need not exist in parallel if there is the concept of a library of object modules.

Library object modules imply an off-the-shelf capability for these software modules. After all new source codes have been compiled to object code, a binding process is invoked which references the library as necessary to build a completely bound system. However, the notion of binding actually implies a dual-level program construction. The first is the construction of object modules; the second is the binding of the object modules to arrive at the total system. Because this process implies that the module definition exists at the machine level, the organization aspect of the total system construction is, at best, primitive.

Modularity also exists in avionics systems in the form of executive functions. Program modules at object time have available a set of functions within the executive portion of the system. In this respect, any service modules in the executive end available to the supported program modules are available to these modules as library functions. Examples of this type of feature would be interrupt, routines, input/output service routines, or control exchanging functions. Because these facilities are external to the source module produced by the programmer, these facilities correspond to object-level library functions.

Problems

In a highly modularized system exhibiting hierarchy, the actions of a specific module at a specific functional location within the total system must be carried out via manipulation of its local data structures and via demands to other modules. In this respect, then, a structure imposed by a particular modularization scheme also imposes a range of functional operations upon individual modules within the structure. The problem often occurring in systems subject to evolutionary development is that the available operations are insufficient to accomplish the desired task. Patching or violating the overall functional structure create a system in which modules are not clearly distinguishable at either the source or object level.

Because modules depend on the execution of other modules, changes in one module will affect the execution of others. This problem is difficult because the interface definitions may be too imprecise to identify dependencies. Usually this happens when the dependence is too loose. In effect, this means modules share these data structures, and changes in one module can have subtle, but disastrous effects on other modules.

To summarize, modularization can introduce rigidity into the functional structure of a system and create interdependencies between system elements which make execution unreliable. Proper modularization can minimize these two effects. A partitioning of the system must recognize both current and potential requirements. Module interfaces must be made explicit and data structures should exist as entities local to specific modules.

Relationship of Modularity to HOL

As discussed above, modularity is the partitioning of an avionics system into various functional pieces or elements. Furthermore, this partitioning usually allows individual modules to reflect both functional and structural designs. This implies that the modularization actually embodies the evolution of the system conceptualization or design. As an avionics system is implemented, it is documented and described in terms of the basic modules of the system. Depending on the level of the description and the modularization, modules may be considered functional primitives of the system.

Because modularization relates closely to the system concept it is important to consider how modularity relates with the use of higher-order languages (HOL). This involves how the concept of modularity relates to language structure, and the concept of a library in terms of language and modularity. (An actual description and evaluation of most HOLs is presented later.)

Most HOLs consist of a set of statement prototypes, which represent language primitives. In addition, HOLs contain conventions dictating program and data structures. For example, several HOLs embed the concept of a subprogram, in terms of a function, a subroutine, or a procedure. Within such subprogram units, provision exists for internal and external data associations, in addition to a capability of formal parameterization. Other HOLs imply a block-oriented structure, either instead of, or in addition to the subprogram concept. These structures allow locality of data structures to individual blocks in order that programs can be structured as hierarchies of modules.

The structural capabilities of a chosen HOL influence, to a large degree, the modularization considerations of systems implemented in the HOL. The capabilities of an individual HOL in the areas of data structure and association to local program elements as well as interface conventions between program elements help define the eventual modularity of a system. In considering potential HOLs for the avionics application, the prime consideration must be how well the modularity natural within the candidate HOLs fits the avionics system requirements. On the other hand, once an HOL is established as the language for an avionics application, the final design and implementation should exploit the capabilities of the HOL. Modularity based on language constructions is essential to ensure an efficient and comprehensible system.

A library consists of a set of modules compatible with a given language and functional system basis. In the avionics application, a library would consist of functional modules with interfaces compatible with the individual language employed in the system implementation. This modularity can exist either as structural entities implying object time modularity or as source structures implying source time modularity. In either case, a library is characterized by functions which can be concisely and precisely defined.

Using a library increases economies in implementation and development. Assuming that an HOL and library are properly matched, the actual source which must be provided for a new avionics mission can be minimized because of the use of library modules. This provides benefits in terms of reduced costs and increased reliability due both to tested library modules and fewer new sources.

The library concept does have its disadvantages. Perhaps the most serious is that library modules must be very precisely defined. In a specialized application such as avionics, it is essential that module interaction

with the total system be completely predictable. Thus, the library concept has the disadvantage of a less exacting functional specification and documentation. To the extent that library modules represent natural constructions within a particular HOL, this problem can be simplified. For example, if a particular HOL provides clean module-to-module interfaces with established communication protocols, then modules from a library could be specified in terms of such a protocol. This would eliminate the need to document those details embedded in the HOL.

The Notion of Optimal Modularity

Many of the concepts noted in the recent literature represent a new conceptual viewpoint as to one functional and physical structure of programs. This material is applicable to more general systems than avionics systems, but certainly applies to them as well.

Optimal modularity is considered in terms of vertical and horizontal modularity, in terms of code and data structure dependencies, and in terms of the concept of a useful abstraction.

The concept of vertical or horizontal modularity relates to the ordering of a hierarchy of modules into a tree of dependencies. The depth of the tree represents the degree to which the hierarchy is vertical, while the width of the tree represents the degree to which the hierarchy is horizontal. If we assume that a system consists of P primitives and is organized into N modules at a single level, then Figure H1 represents a plot of the structural complexity of the system. Here we assume structural complexity to be a function only of the number of primitives, N, in a given module.

As shown in Figure H1, considering the entire system to be one module of P primitives is equivalent to considering the system to be P modules, each with one primitive. If we assume that the total system complexity results from N modules as primitives of the global system and from P/N primitives within individual modules, then the total system complexity, TSC, is

$$TSC = F(N + P/N).$$

If the function F behaves monotone increasing with its argument, then the total complexity would be minimized for $N = \sqrt{P}$. Such models do not apply to more realistic problems but even then the behavior of the TSC function must be considered. In general, the payoff in the modularization of a system is significant only if the degree of modularity is not to either of the extremes of vertical or horizontal modularity.

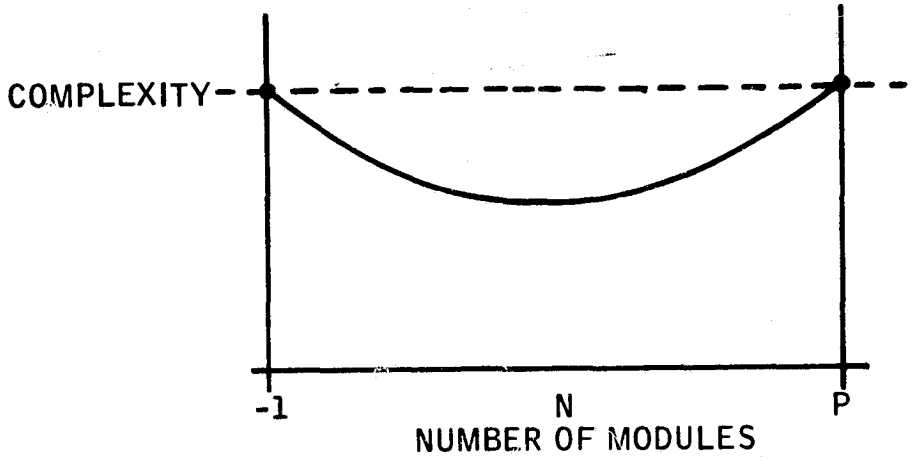


Figure H1. Complexity versus Modularity

A module can consist of a combination of code and data structures. Depending on the criteria used for module defintions, the code of a module will interact with data structures local or global to the module. Data structures which are local to a module are those accessible only from within the module. Data structures not local will be considered global. As we have discussed above, the degree of global data referencing in a modular system is a function of the formality of the interface conventions between modules. In a rigorously interfaced modularized system, for example, the only global data structure references allowed might be via a formal parameter mechanism.

Recent papers by Parnas (Ref. H1) and Haney (Ref. H2) show the importance of the relationship between the code and data structures of modularized systems. Work reported in these papers indicates that intermodule dependencies, because of shared structures, contribute to system complexity.

Parnas deals with the criteria of module selection and concludes that good modularity must be oriented about the data structures of the system. Individual modules should reference only local data structures. All data structures of the total system should ideally be hidden within specific modules. Such a criterion would imply that the modularity of the system would not necessarily follow the lines of functional decomposition.

Haney deals with the statistical dependencies between modules in the presence of corrections being made in individual modules. His analyses show how changes in a system can cause ripples to propagate throughout the system. The key to a well-modularized system is to minimize the ripple and to make modules as independent as possible. This relates to the criterion proposed by Parnas, as hiding data structures within modules minimizes intermodule dependencies.

Liskov (Ref. H3) introduces a design methodology for software systems. This is the concept of a "useful abstraction", and demonstrates how such constructions can be embedded as modules in a hierarchal system. Accordingly, this methodology represents a criterion for modularity. The criterion consists of a system structured as a hierarchy of partitions, each level depending on the useful abstractions (modules) of the next lowest level, with the total connections between modules minimized.

Notice that what Liskov proposes does not necessarily coincide with the criterion of Parnas. The useful abstraction of Liskov may, for example, be a global data structure. However, all schemes emphasize the idea of minimum connection. Selecting modules should be based on a scheme which allows each module to be independent and which formalizes intermodule connections.

Source Level Modularity

The concept of source level modularity has been mentioned above as a convenient way to economize in producing a program source. Also, the relationship between modularization and HOL was related to the formal binding of program and data structures. The issues of modularization and HOL arise because of the need to transform a specification for program behavior into a working avionics system. Modularization is a tool for breaking up a very complex system into pieces which can be handled more easily. Parnas (Ref. H1) has suggested that the modularity used in system construction need not exist at the final implementation of the system. This section discusses how modularity might be maintained at the source level, and outlines advantages and disadvantages of various levels of source definition.

Modularity can exist only at a source level if a suitable HOL is used. Because compilation of the HOL source must result in the integrated and deployed state of the system, the HOL must include those constructions necessary to describe the desired application system. Among the features most desirable are facilities in HOL for exploiting architectural characteristics of the hardware. However, in recent years there has been a growing interest in developing machines designed to execute specific HOLs, both indirectly and directly.

A second requirement to facilitate modularity at source levels is for a reliable implementation of the HOL compilation tools. Errors resulting during the compilation or assembly of the system are as damaging as those contained directly in the source code of the system. Furthermore, if the use of source-level modularity with HOLs is to provide optimized (in some sense) object systems, the resulting object code will be very difficult to test independent of the structural organization of the HOL. This is due to the difficulties of exhaustive testing programs.

Avionics systems have been subjected to very rigid standards to ensure system reliability. It is common practice to perform binary comparisons of software modules to verify correct avionics programs. As the debugging process produces various patches to fix errors, these patches are cemented into the module because of the binary-verify. Needless to say, this produces software modules which are not cleanly constructed. The problem with this type of approach is that system reliability and validity are tested and maintained at the object level. In the case of source-level modularity, changes during the debugging phase must be based on the functional behavior of the system and verified on the same basis. The result of source-level maintenance is a total system which retains more of its original functional characteristics.

HIGHER-ORDER LANGUAGES

This section discusses the attributes and capabilities of HOLs.

Discussion of HOLs and Avionics

A programming language represents a computer program design in a form capable of being converted into machine language. The language is used to describe the data and variables which will be used during the execution of the program, to define the procedures which will act on the data and to control the intermediate translation and loading activities which transform the source code into the machine code.

There are many modes of communication which could be used to program a computer. Console switches, patch boards and two dimensional CRTs are examples of methods used for special purposes. However, for general purpose programming, sequential input devices employing a fixed character set are the normal mode. Sequences of these characters specify the design which is to be communicated to the machine.

Two distinctive aspects characterize a programming language: syntax and semantics. A language is usually considered to consists of a character set and a set of grammatical rules which define the syntax of valid statements in the language, if the syntax rules apply. A programming language

also has associated with it a set of semantics or meanings associated with valid syntactic statements. For example, the FORTRAN statement GOTO 234 obeys the FORTRAN language syntax rules, but to be valid semantically there must be some other statement with the label "234".

A third important aspect using programming languages is the pragmatic implementation of the language compilers. These characteristics are related to the convenience and naturalness of using the programming language. Pragmatic characteristics are dependent as much on the implementation of the programming language compiler as they are on the syntactic and semantic definitions. These characteristics can also be subjective in that they are colored by the user's preferences and habits and can be hard to qualify. Yet, pragmatic characteristics may play as important a part in determining whether a programming language can be used successfully in a specific application as the technical syntactic and semantic characteritics of the language.

Honeywell's approach to programming language for avionics is twofold. First, we will discuss some constraints placed on the use of programming languages in avionics applications and list general language characteristics required. These requirements can then be used as a filter to eliminated nonresponsive programming languages. Some technical characteristics of the most eligible languages will then be tabulated to show similarities and differences.

Programming Language Levels

One gross way of separating programming languages into classes is to assign a notion of "level" to them. A low-level language implies a direct-source code to object-code translation, and forces the programmer to work at the level of the machine instructions of the computer which he is using. A Higher-Order Language (HOL) allows the programmer to communicate more in terms of the natural language of the application area.

Low-level programming languages are machine oriented and usually have a very simple syntax. The following are examples of low-level languages:

- Binary or octal - The programmer directly codes the object-program bit patterns. There are no constraints or formalisms. This is the most primitive form of programming, but is still used even with high-level languages for the presetting of data in tables.

- Assembly language - Some constraints are imposed. Symbolic operations and addresses and minimal syntactic and semantic

checks are provided to help the programmer. In simple assembly languages, one source statement translates to one object machine instruction. However, macro- and meta-assemblers can be extended to be very machine independent and produce one source statement to many object statement translations.

- Special Purpose Languages and Processors - There are low-level languages which are oriented toward special application areas. They are low level in that there is a very direct correspondence between the source code and the object code. Semantically they may be as powerful as some higher-level languages. Examples of such languages are string- and list-processing languages. Symmetric List Processor (SLIP), for example, is FORTRAN-based and uses the syntax of the FORTRAN CALL statement. SLIP can be considered a low-level language in that only a very restricted set of FORTRAN statement types need be written by the user. Another example is Bell Telephone Laboratory Low Level Linked List Language (L^6). In terms of capability, L^6 is about equivalent to SLIP, but has its own special syntax.

High Level Languages

Following are some of the characteristics which distinguish high-level languages from low-level languages:

- In a high-level language, a compiler is involved in translating the source code into the object code. In low-level languages, the translation usually only involves a simple replacement of symbols for their numeric codes.

- In high-level languages, the symbols used represent and have associated with them the attributes of more complex objects. In low-level languages, symbols usually only refer to simple values, or to addresses holding the attributes of the complex objects. For example, in a high-level language, the fact that a symbol represents a matrix variable can be directly communicated to the language compiler, whereas in a low-level language only the address is communicated and the programmer must mentally (or through comment cards) keep track of the fact that he is working with a matrix.

- In high-level languages, one source statement will usually be translated into many object code instructions.

- In high-level languages, extensive transformations may be performed on the source code to optimize the object code. For a low-level language, the translator usually only performs direct one-for-one translations.

- Programming in high-level languages forces program structure and discipline. What can and what cannot be done is controlled more rigidly in a high-level language. As a result, programming errors and programming costs are reduced. However, if the control is so rigid that the programmer has to use tricks to get around a language restriction to get his job done, or if not enough control is left to the programmer to allow him to generate a sufficiently efficient code, the high-level language can become a great enough burden that it becomes more cost-effective to use a low-level language.

General Language Requirements

Before getting into details about the specific requirements placed on programming languages used for avionics applications, we can list a few general characteristics which are not technical in nature, but which impact the relative success of programming languages.

Some of these characteristics and their relative importance may have to be evaluated subjectively. For example, relative ease of modifying is a characteristic influenced as much by the compiler and file support system for the language as by the language itself. Furthermore, if two users have two different types of changes to be made, the ease of making them may vary for the same language and support system. Also, one language may be specified such that relatively few user-initiated changes have to be made as compared to a second language. Therefore, the relative importance of the language characteristics defining ease-of-making-modifications varies.

With the above caution in mind, some general programming language characteristics are listed below which are important to avionics applications:

- Commonality of Use -- The extent to which a programming language is to be used is a very important characteristic. For example, FORTRAN is used very widely, and, therefore, retraining programmers for a FORTRAN project is minimized. Also, there is a large data base of programs which may be usable in the project. Commonality of use can compensate for many other shortcomings that a language may have.

- Consistency -- Applying the same language rules for all similar source-code cases is very important. Otherwise, the programmer has to be more concerned with the consequences of exceptions than in getting his problem solved. For example, in standard FORTRAN an array index may only be of the form c*v+c, where c is a constant and v is an integer variable whereas any expression which evaluates to an integer could be accommodated.

- Generality -- The language should be sufficiently general to be applicable to a wide range of avionics programming problems. This characteristic must usually be traded for compiler efficiency and for development costs. If the language is not general enough, dialects and extensions have to be used, thereby violating the commonality-of-use characteristic. If, on the other hand, the language is too general, compilation efficiency may be degraded to an extent that it is no longer practical to use the language. If a switch to another language is required, the commonality-of-use characteristic is again violated.

- Efficiency balance -- A programming language can be evaluated for efficiency in at least four respects: writing speed; reading convenience; compile speed; and execution speed of the compiled code. Writing speed refers to the relative speed with which the programmer can write the code. If a language forces the use of long key words and identifiers, the preparation of the code can be slow. Reading convenience refers to the ease with which the original or another programmer can extract the meaning of the program from the source code. Compile speed refers to the relative speed of converting the source code to the object code. Compile speed is most directly related to the implementation of the compiler for the language, but may also be influenced by language characteristics which force the compile algorithms to be slow. The execution speed of the compiled code is again more a function of the compiler implementation than the language. However, there may be restrictions in the language which make optimization impossible to perform, or at least sufficiently difficult so that they are not cost-effective to include in the compiler.

- Ease of creating re-usable modules -- If a frequently used function can be programmed and parameterized so that it can later be re-used rather than reprogrammed, software development costs can be significantly reduced. The ability to re-use previously programmed modules and the desire to do so depends strongly on the support software systems on which the compiler is implemented, on the compatibility of the compilers (if a number of installations are used), and on the political and pricing characteristics surrounding a development effort. The language has to be able to support modularity and have good parameterization and parameter passing capabilities.

- Transferability of code -- The ability to transfer a program written for one machine to another is an important characteristic for avionics applications. Even though the programming language may be machine independent, programmers must often use data and program structures in the source code which they know will result in efficient structures in the object code. If this is done extensively and without discipline, the resulting source code may be tied very strongly to one object machine.

- Ease of making modifications -- The relative ease of making changes to the source code may be important. But this characteristic is also associated more directly with the compiler support system than with the language. The source code structure and macro preprocessing capabilities of the language do affect the relative ease of making modifications.

- Comprehensibility of code -- A language with an easily understood code is also important. The program source represents the final authority for the problem specification. Insofar as this source provides direct documentation, problems concerning the communication of programs can be removed.

Avionics Environment Implications

Since a programming language is to communicate a program design, it is important to match the language with what is communicated, within what context it has to be communicated, who communicates it, and to whom it is communicated.

What is communicated -- In the avionics environment, a programming language represents primarily the operational software which will be executing on the avionics computer during mission time. Following are examples of the types of operations and functions which must be represented in the programming language:

- Mathematical functions
  - Real number operations (floating point and fixed)
  - Trigonometric functions
  - Coordinate transformations
  - Vector and matrix operations (these can often be simplified)
  - Integration functions
- Conversion, sorting, list processing

- Data packing and unpacking (bit picking and packing in particular)
- Executive, low-level I/O and digital communications control

Note that the following types of functions are specifically excluded:

- Support software related functions - compilations, file operations formatted input/output
- Complex number operations (perhaps these functions will be used more extensively in the future).

In what context are designs communicated -- To reduce software development costs during deployment, the tools and languages must be specified so as to be optimally matched to the tasks during the various phases of system deployment. But, before this can be done, we must identify and analyze the activities during a deployment which contain software development activities.

A set of deployment phases was postulated with the intention that these would represent typical sequences of steps which will be traversed before an avionic data processing system is fully deployed.

The phases in the deployment are shown in Figure H2. Each will be briefly described and discussed regarding its software implication.

- Systems Requirements Definition - The required functions and major constraints on the avionics system are specified.

  At this stage in the system design only rough data processing requirements are formulated. No approach need be specified. It is assumed that no software tools or languages are used in this phase since the requirements are expected to be in English or mathematical equations.

- System Feasibility Study - The requirements are studied and system solutions postulated. High-level system analysis tools are used to evaluate the alternatives to determine the feasibility of the approaches.

  It is assumed that high-level simulation languages and other system analysis tools are available to the developers.

- Detailed System Functional Design - During this phase of developing the deployed avionics system, detailed solutions are developed for the data processor configuration and software. Designs are also developed for the environment simulator, which will be later used to verify the system and for the interfaces to the non-data processing components of the avionics system.

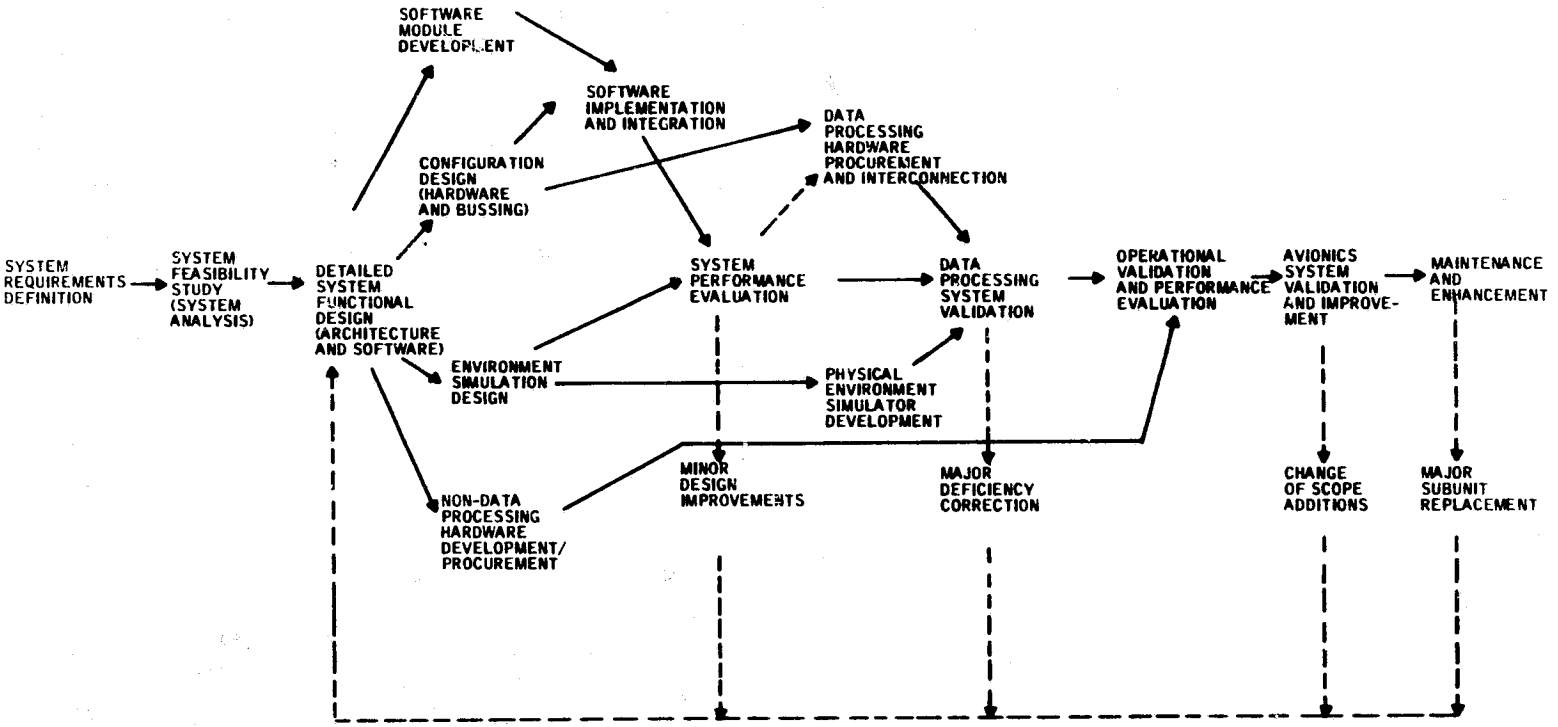


Figure H2. Anticipated Deployment Phases

The development of the configuration and software module designs will involve iterating through a sequence of solution refinements. At the start of the detailed design only a rough, high-level picture of how the system operates is known. Through the sequence of refinements, the developers will eventually arrive at a configuration and software design in which the data sets are partitioned, the processing functions allocated, and rough data rates known.

The tools which will be required include high-level system simulators, processor simulators, code generators for the processors, and tools to extract performance and timing characteristics from code samples. Ideally, only high-level languages should be required in the detailed design; but some low-level code generation will have to be used, at least during the development of timing characteristics.

- Software Module Development -- More than likely each deployed system will have at least some unique functions. Specific subsystems for which this may be true include display processing, weapon control, and electronic warfare. The software modules required to implement these unique functions can be programmed in a high- or low-level language. If no future use of the modules is seen beyond the system for which they are being implemented, generality and concurrency can be sacrificed for lower development costs of the modules.

- Configuration Design - This phase of the deployment of the system involves the selection, specification and design of the specific busses and processing element components which will be interconnected to realize the data processing system.

  It is assumed that code generators for the selected high-level language will be available for all processing elements, and that code generators or at least macro-type capabilities will be available in the support software systems so that new functions can be adequately programmed and integrated into the rest of the system. Adding a new hardware functional unit implies that there must be a parallel effort in software development to guarantee programmability and compatibility with existing system communication conventions and procedures.

- Software Implementation and Integration - The modules developed for the specific system and the modules from the library are integrated according to the specification of the system configuration. Included in this integration process is the functional testing of the modules at the various levels of integration.

Implicit in this phase of software development is the presence of at least a high-level interpreter and some convenient means of specifying integrated module tests. As more and more modules are brought into the test cycle, the need for automatic integration tools and a simulated executive become more pronounced. The function of the integration tools at this stage is to eliminate unnecessary computations and, therefore, make the testing more efficient. The simulated executive is required when the integration of the modules arrives at a point where actual executive functions are required in the simulated system to realistically test the integrated modules.

- Environment Simulation Design - The purpose of this phase is to develop environment simulators which will be used in non-real-time and in real-time simulations of the avionics system. These simulators are viewed in our context primarily with the intent of validating the data processing system, and in determining realistic performance characteristics of the system.

  Initially, a software simulator of the environment is required, but this simulator need not run in real time. This simulator is used to drive high-level simulations of the avionics system, and is used to simulate external responses during functional software module tests.

  Later in the deployment cycle, a real-time environment simulator is required to drive the prototype system at real-time speeds.

- System Performance Evaluation - The software modules have been integrated and tested functionally. Now realistic physical processor implementation characteristics must be applied in a simulated environment in order to get an initial validation of the system performance.

  During the detailed functional design of the system, performance assumptions had to be made to arrive at a first design. This phase represents the first opportunity to verify that those assumptions were in the right ball park. If they were not, corrective redesigns are still reasonable, since the system has not yet been committed to hardware. It is expected, though, that the system developers will become proficient in their design of later systems. Eventually the results of this preliminary system performance evaluation should indicate the need for only minor design improvements.

This phase requires further use of the compilers, integration tools, and executive functions. Also, the simulators of both the environment and the processing elements and performance measurement tools will be required. The developer controlling the performance evaluation runs will need to know only external physical process characteristics.

- Data Processing Hardware Procurement and Interconnection - This phase of the deployment of a system includes the procurement of a prototype of the designed hardware and the interconnection of the components. No software tasks are included in this phase.

- Data Processing System Validation - The data processing system hardware has been implemented and the software for the data processing system developed and validated. This phase brings these two together and performs a validation under what appears to the data processing system as real-environment conditions. Any critical interactions not accurately reflected in the simulators or functions which were not properly specified (but perhaps accurately implemented as specified) will be caught at this phase of the deployment. (Note it is assumed that deficiencies uncovered here are those of system designers, and those resulting from inaccuracies in the software simulators.)

  During this phase of the deployment, the developed software will be handled extensively but will not be altered functionally. Performance monitoring will be done during the validation, implying the need for software hooks.

- Avionics System Validation and Improvement - During this phase, the entire avionics system is validated. It is assumed that neither the data processing hardware nor software will require much attention at this stage (other than possibly monitoring).

  Explicit software tools required at this phase of the deployment are the software hooks needed to monitor system performance and support software to analyze the results of this monitoring.

- Maintenance and Enhancement - This assumes the normal operational mode of the avionics system containing the data processor. Diagnostic tests are required for two purposes: to verify that the components within the system are functioning properly; and to diagnose and isolate a detected error. The latter requirement can be further subdivided into the diagnosis of errors to initiate real-time recovery procedures, and the diagnosis of errors during non-operational periods to repair the system.

Who communicates the design -- The tools and languages required for use in the deployment of an avionics data processing system must be designed so that information flows between the support system and the user as freely as possible. This can only occur if the tools and languages are matched to the background and functions of the developers.

Table H1 lists a postulated range of developers, their functions, their backgrounds and the kind of tools and languages which they will probably use as they develop a specific avionics system. Though the list is not necessarily complete, it already shows a requirement for a wide range of language levels and tools which the language must drive.

An objective in the specification of the support software tools should be to exploit commonality in the development of the tools and languages when feasible. Flexibility and level ranges must also be associated with the languages so that the developer with a rich theoretical background can communicate with the tools at a high level, and at the same time the developer who has a less complicated function to communicate is not forced into learning symbol definitions and concepts not required to perform his function.

Summary of information communication: what; by whom; to whom -- The characteristics of the avionics system to be deployed and the characteristics of the processors being used are the basic factors determining what information has to be communicated to successfully implement the avionics system. How the information is communicated depends implicitly on who generates the information, on who receives it and on what is being communicated.

The deployment characteristics presented previously were reviewed and some of the more important items identified which had to be communicated. For each of these items it was then postulated who would communicate the item, who would receive the communicated information, and, finally, how the communication is accomplished. The results of this exercise are tabulated in Table H2. The contents of this table are not claimed to be comprehensive as to what has to be communicated, nor complete as to how it is communicated. Because of the subjective nature of communication, the entries in the table may very well be controversial. However, the table does present at least a basis for discussing communication and language requirements during deployment.

Programming Language Candidates

This section discusses some of the major programming languages which are contenders for avionics applications. By discussing structure and capabilities, an adequate technical comparison can be made. Possible extensions of programming language capabilities which are within the state of the art but which are not yet incorporated into the languages discussed here will be covered in the next section.

TABLE H1. POSTULATED FUNCTIONS AND BACKGROUNDS OF DEVELOPERS

Deployed System Developer Category	Functions	Background	Tools Used	Software Language Level Used
• Avionics system designer	• Analyze requirements • Postulate solutions • Develop high level models • Specify system constraints	Avionics systems experience	• Manuals from related systems • Problem projections	None or very high
• System architect/ analyst	• Synthesize specific solution • Performance projection • Sensitivity analysis • Define architecture • Specify software/hardware	• Technology capability • Computer architecture experience • Software experience • Analytic capability • Synthesis capability	• High-level languages (system description) • High-level simulator • Analysis packages • Executive control languages	High
• Program developer - program designer - program coder	• Develop module code • Integrate code • Optimize code locally • Develop verification tests • Develop diagnostic tests	• Experience with development language • Theoretical background in assigned area (high school to PhD)	• Language processors • Optimizers • Low-level simulators • Debug and verification tools • Integration tools • Execution control language	High to low
• Performance analyst	• Develop simulation data and statistics • Develop benchmark data and statistics	• Experience with simulators, performance instrumentation, statistical packages	• Code timers • Instrumented simulators • Hardware instrumentation equipment • Execution control language	High
• Operator	• Manual procedure executors	Special avionics areas • pilot • other flight crew • maintenance supervisor No knowledge of internals of system	• External reference manuals • Simulators (external)	None

TABLE H2. COMMUNICATIONS DURING SOFTWARE DEPLOYMENT

Deployment Phase	What Is Communicated	By	To Whom	How Communicated
System feasibility study	High-level simulation models Algebraic system models	Avionics system designers	Analysis tools	• Existing algebraic languages • Simulation language • Algebraic manipulator
			Other designers	• Block diagrams • Algebraic models • English-language descriptions
Detailed system functional design	Problem constraint specification Partial solutions Hardware and software module external specification Data flow descriptions Module external characteristics	Data processing system designers	Analysis tools	• Algebraic language • Simulation language • High-level functional specification language • Procedural language
			Other designers	• English function descriptions • System equations
			High-level interaction system	• High-level function specification language • Interactive solution refinement language
Software module development	Software module implementations based on previously defined library of modules Module classfication: • algebraic functions • decision, display, command and control • Executive, I/O, data compression • New functional modules	Module programmer	• System designers • Compiler • System integrators	• Procedural language • Low-level language • High-level language • Command-oriented, procedural with minimal algebraic capabilities • Command-oriented, procedural and low-level language
Architecture design	Unique module hardware design	Hardware system designers	Logic designers	• Logic design diagrams • Logic design equations • Register transfer language
	Hardware module interconnections	Hardware system designers	Hardware system builders	• Block diagrams (2-D interactive language) • Interconnect language

TABLE H2. COMMUNICATIONS DURING SOFTWARE DEPLOYMENT (concluded)

Deployment Phase	What Is Communicated	By	To Whom	How Communicated
Software implementation and integration	During build tasks: • Interaction of functional modules • Interrelation of data • Interconnection of processors • Implementation decision control	Data processing system designers	• System integrators • Other designers	• High-level interactive language (rule and event driven)
	During verification tasks: • Monitor point control • Consistency checks (programmable) • Test drivers (programmable • Status display	Software verification programmers	Test monitor and simulator	• High-level language (test and debug oriented) • Low-level language
Data processing hardware procurement and interconnection	Acceptance tests for new hardware	Hardware-oriented programmer	Hardware test monitor	• High-level language oriented toward automatic testing
	Diagnostic development for new hardware and interconnects • GO/NOGO operational tests • Comprehensive fault detection/isolation diagnostics Diagnostics for dynamic recovery	Hardware-oriented programmer	Diagnostic-oriented compiler	• High-level language oriented toward diagnostics • Low-level language
System performance evaluation	Monitor point control Statistics gathering control (programmable) Data analysis (programmable) Data/result display (programmable) Simulated external environment control	System designer	• Simulators • Test monitors • System integrators	High-level language • Event-driven • Algebraic
Data processing system validation	Monitor point control Statistics gathering	Validation team	• Hardware monitors • System integrators	High-level language modifications
Maintenance and enhancement	Error isolation procedure selection (non-programmable)	Maintenance technician	Test monitor and controller	• Function buttons • Special language
	Programmable diagnostics: • Modify diagnostics • Add new diagnostics			• Special language • High-level language oriented toward diagnostics

A brief genealogy of the major general-purpose, high-level languages is shown in Figure H3. The languages within boxes will be discussed in detail. The number shown below the languages is the year in which the language was introduced. The actual dates may be contested but they do give an indication of relative maturity and timing. Each of the high-level languages will be discussed briefly first.

The Navy's new language (CMS-3) currently under development for Advanced Avionic/All Application Digital Computer (AADC) and Houston Aerospace Language (HAL) were not considered in this study. CMS-3 is just in the definition stage, and HAL, although implemented for the IBM4π and used on the space shuttle program, has no user history to support claims. Jovial J73 was only included because it is an extension of a lot of experience and effort on Jovial J3 and PL/I. We have tended to be conservative in the language area due to the following factors:

1) It is arbitrarily easy to define a language, but arbitrarily hard to define a "good" language

2) Benefits of language can be argued, but the proof of goodness is in the use

3) Languages need time to be refined and tested to prove their effectiveness

4) Knuth's study on programming habits showed that simple assignment statements are what is mostly used

5) Many of the newer languages, CMS-3 for example, do not support 4) without a lot of external clutter

6) Avionics problems appear constrained and well-defined enough that a language may be selected from the set available today that should be adequate

FORTRAN -- FORmula TRANslator (FORTRAN) is an algebraic, procedure-oriented, scientific, formula-translation language. The language was first developed by IBM for the 704 series of computers and gained rapid recognition. An improved version, FORTRAN IV, was introduced in 1962 and this version has been adopted as the ANSI 1966 standard FORTRAN. Since then, it has become the most widely used general-purpose, algebraic programming language. A revision of the 1966 FORTRAN standard is now being processed by the American National Standards Institute.

The greatest benefit of FORTRAN is its wide acceptance. Because its use is so common, there are many programmers who have a very good working knowledge of the language and programs written in FORTRAN can have a relatively high degree of transferability. The language has a fair number of

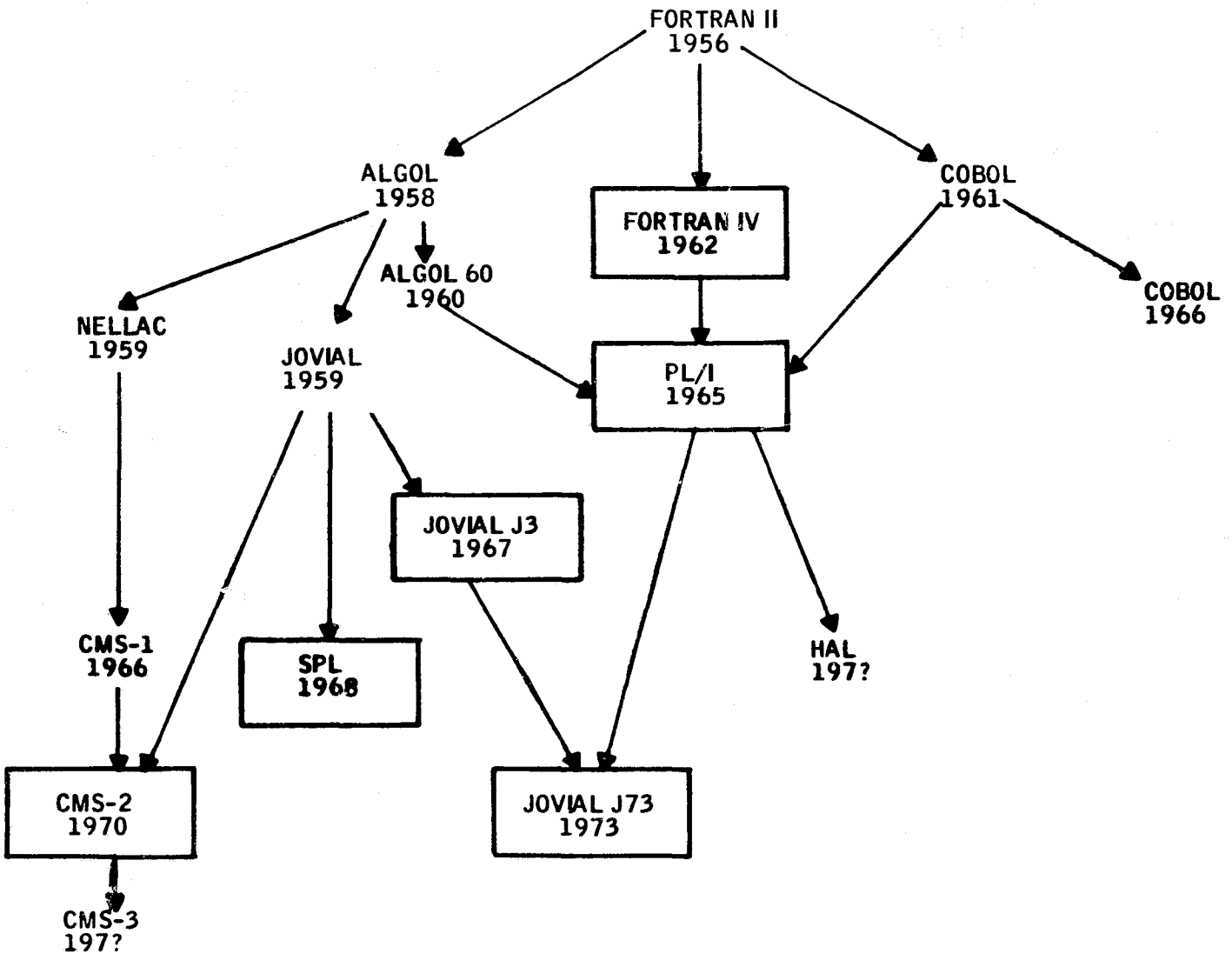


Figure H3. Major Programming Language Genealogy

inconsistencies, some of which will be corrected in the new standard. Its greatest short-coming for avionics use is its inability to control for a fine enough granularity, and its lack of controlled fixed-point operations. FORTRAN is also weak in the definition and control of data structures.

COBOL -- COmmon Business Oriented Language (COBOL) is a procedural language for business programming applications. It is very English-like and wordy compared to FORTRAN, and is not a convenient language for programming algebraic computations. COBOL is definitely not a candidate language for programming aircraft avionics computers.

However, COBOL is mentioned because it is one of ANSI standard languages and has influenced the PL/I language. The method and extent of structuring data in COBOL is COBOL's strongest technical advantage for use in avionics.

ALGOL -- ALGOrithmic Language (ALGOL) is an algebraic, procedure-oriented language which has served as a publications reference language for algebraic algorithms and as an international standard programming language. The language was specified and improved in the late 1950's. ALGOL 58 was the first version which was defined and ALGOL 60 is the final standard which still holds today.

In the early 1960's, ALGOL was used relatively extensively and many dialects, some of which were for aerospace applications, were developed. The use of ALGOL seems to be declining because of improved capabilities in other languages and because of a lack of interest by computer manufacturers in supporting the language. Also, ALGOL programs tend to perform less cost-effectively than FORTRAN programs. An updated and radically different version of the language, ALGOL 68, has been defined and is being worked on at universities, but has not received much acceptance as yet.

The greatest advantage of ALGOL was that it was the first language to be very rigorously defined. ALGOL has very powerful and convenient procedure-specifying capabilities. It does not have a standardized I/O specification and it is very deficient in controlling precision and in packing data as required for avionics applications.

NELIAC -- Navy Electronics Laboratory International Algebraic Compiler Language (NELIAC) is an algebraic, procedure-oriented language and is an example of a successful ALGOL 58 dialect developed for a special application area. NELIAC has been used successfully and documented, but has not been widely accepted. NELIAC also is missing important capabilities which other, more modern languages have.

PL/I -- Programming Language/One (PL/I) is a general-purpose, procedure-oriented programming language designed by IBM to replace ALGOL, FORTRAN and COBOL. The language is very large and capable, but its size

and complexity has restricted the implementation of compilers to relatively few large-scale computer families. There are many large computers which do not have PL/I compilers.

Many subsets and dialects of PL/I have been defined and used. There do not appear to be any subsets of the language oriented specifically for avionics applications.

JOVIAL -- Jules' Own Version of the International Algebraic Language (JOVIAL) is a general-purpose, algebraic, procedure-oriented programming language. It was initially defined as a dialect of the ALGOL language and consequently contains many of the capabilities and characteristics of ALGOL. There are many dialects of JOVIAL; the two which we are most interested in are JOVIAL J3 and JOVIAL J73.

The JOVIAL J3 language was developed for the Air Force primarily as a command-and-control programming language. It has been adopted as a standard Air Force programming language.

JOVIAL J3 has many of the features required for avionics application programming tasks. It is weak though in the specification and control of data structures, and does not provide for adequate load-time control and linking. The latter characteristics tends to make it expensive to use.

The JOVIAL J73 language has been specified and is being proposed as a standard Air Force programming language. JOVIAL J73 uses 60 characters, as in PL/I, and contains a modern set of capabilities. The data structure defining and control capability are particularly appropriate for avionics programming.

JOVIAL J73 is still in the definition phase; no compilers have been built for it. It is expected, though, that it will be a widely accepted language.

SPL -- Space Programming Language (SPL) is an algebraic, procedure-oriented programming language for developing software for computers used in space applications. SPL is basically derived for JOVIAL and, therefore, is structurally very similar to JOVIAL.

SPL is a very capable language technically, and satisfies most of the requirements for avionics applications. The acceptance of SPL for space programming applications has been very limited thus far.

CMS -- Compiler Monitor System (CMS-1 and CMS-2) computer programming systems used by the Navy primarily for the development of tactical data systems. CMS-2 is the improved and currently used programming system. CMS-2 is also the name of the high-level programming language used in the CMS-2 programming system. The language is procedure-oriented, and is derived from features contained in the languages ALGOL, FORTRAN and JOVIAL.

The CMS-2 language has many of the technical characteristics required for avionics applications. One of its unique properties is that even arrays are passed into procedures in a call-by-value mode. A problem with the language is that its acceptance has been restricted to the Navy.

Technical Characteristics Summary -- This section presents some of the key technical characteristics of programming languages and compares some representative languages according to these characteristics. If a programming language is to be chosen or developed for a particular avionics application or range of applications, these technical characteristics must be evaluated and weighted according to the requirements. These technical characteristics must then be added to other non-technical factors which affect the choice.

The technical characteristics of FORTRAN, JOVIAL J3, PL/I, SPL, CMS-2 and JOVIAL J73 are shown tabulated in Table H3. In addition, one column compares the characteristics of a representative assembly language with macro capabilities. For SPL, the J6 dialect of the language is described. Most of the languages have many alternates for specific statements; only some of these are included. Also, some languages have capabilities which are not included in the table.

The languages are described in terms of the following four major headings: syntactic characteristics; compiler and loader directives; executable statements; and subprogram characteristics.

The syntactic characteristics of a language basically describe the input format and structure of the source code. The lexical properties describe the minute detail of the language: its character set; separators and punctuation; identifier formats; source code representation of constants; and literals and formats of comments. The source code structure aspect of the syntactic characteristics describes more the large syntactic units: how programs are formed; whether block structure is used; and how the scope of identifier definitions is defined.

The compiler and loader directives contained in a language describe to the compiler the characteristics of variables and programs and also control the execution of the compilation and loading. The directives are only discussed here in a very general sense since they are very dependent on the compiler and loader implementations.

Declarations are a specific form of directive to the compiler. Declarations basically describe to the compiler the properties of the variables and data structures which are used in the generated object code and the execution properties of the code itself when these properties cannot be deduced by the compiler. Of specific interest are the various methods of declaring variables and the specific data types which are handled by the languages. The relative capability the language has for making data structure declarations is also important in avionics applications.

TABLE H3. TECHNICAL CHARACTERISTICS OF HOLs

Technical Characteristics \ Language	FORTRAN	JOVIAL J3	PL/1	SPL	CMS-2	JOVIAL J73	Representative Assembly Language
Syntactic characteristics							
- Lexical properties							
• Character set	48	48	60	48	48	60	48
• Input format	72 column cards, continuation cards, free field, blanks ignored	Continuous line; free field, blanks are separators	Continuous line; free field; blanks are separators	Continuous line; free field; blanks are separators	Continuous line; free field; blanks are separators	Continuous line; free field, blanks are separators	Fixed field; one instruction per line
• Reserved words	No	Yes	No	No	No	Yes	Yes
• Separators, punctuation	Blanks are not separators	Blanks are separators; "$" for statement separation	Blanks are separators; ";" for statement separation	Blanks are separators; no statement separators	Blanks are separators; "$" for statement separation	Blanks are separators; ";" is statement separation	Blanks are separators
• Identifiers (first character is alphabetic)	1-6 alphanumeric	1 character names reserved; 2 and up alphanumerics with prime (') for separation	1 and up alphanumerics with underscore (_) for separation	1 and up alphanumeric	1 and up alphanumeric	1 character names reserved; 2 and up alphanumerics with ($) or (') for separation	Same as FORTRAN
• Constants and literals							
- Numeric							
• Integer	Decimal number string; character constant	Decimal, octal, hexadecimal number string	Decimal, binary string	Decimal, binary, octal, hexadecimal string	Decimal string	Decimal string, bit pattern status	Decimal, octal string
• Fixed point	Not provided	Decimal with point, scale	Decimal with point, exponent	Decimal with point, exponent	Decimal with point, exponent	Decimal with point, exponent	Not provided
• Floating point	Decimal with point, exponent	Same as fixed	Same as fixed	Same as fixed	Same as fixed	Same as fixed	Same as FORTRAN
• Logical							
- Boolean	.TRUE..FALSE.	1,0	'1'B, '0'B	'TRUE', 'FALSE' or 'ON', 'OFF'	1,0	Integers 0,1	Integers 0,1
- Status	Not provided	Symbols defined as V(RAINY), V(CLOUDY), V(SUNNY)	Not provided	Sympols defined as 'RAINY', 'CLOUDY', 'SUNNY'	Symbols defined as 'RAINY', 'CLOUDY', 'SUNNY'	Symbols defined as V(RAINY), V(CLOUDY), V(SUNNY); treated as integers	Symbol equivalency
• Literal strings	Hollerith strings	Hollerith strings	Single or repeated	Strings allowed	Called Hollerith constants	Called character constant	Character constant
- Character	6HSTRING. Restricted use, machine-dependent code.	6H(STRING) or transmission strings 6T(STRING)	Strings allowed 'STRING' or (1)'STRING'	'STRING'	H(STRING)	'STRING'	
- Bit	Not provided	Bit pattern, octal or hexadecimal constant 0(77)	Binary bit string '111111'B	Binary octal or B'111111' O'77' X'3F'	Not provided	Bit pattern 1B'111111' 3B'77' 4B'3F'	Octal, hexadecimal constant

TABLE H3. TECHNICAL CHARACTERISTICS OF HOLs (CONTINUED)

Technical Characteristics \ Language	FORTRAN	JOVIAL J3	PL/I	SPL	CMS-2	JOVIAL J73	Representative Assembly Language
• Comments	Separate lines with 'c' in column 1	Quoted strings; legal where blanks are legal "COMMENT"	Bracketed string where blank is legal /*COMMENT */	Quoted strings; "COMMENT"	Quoted string where blank is legal; COMMENT statement	Quoted strings where blank is legal; "SOME COMMENT", semicolon terminates comment	Asterick in Column 1
- Source code structure							
• Code division	Division not enforced. Recommended - data declarations before executable statement. Each subroutine, function compiled separately.	Recommended but not enforced; data declarations before use.	Division not enforced	Data declarations before use	Data designs and procedures; data defined before procedures	Natural division: procedure declarations; data declarations; statements	None enforced
• Block structure	None below subroutine levels; subroutines, functions treated as code blocks.	Extensive for procedures; limited for data.	Very extensive; used primarily for control of scope of identifier definitions.	Extensive for procedures; limited for data.	None below procedure level	Very extensive as in PL/I	None
• Static scope	Data declarations hold for entire subroutine, function or main program	3 absolute scopes: COMPOOL is global scope; main, and procedure.	Data and procedure declarations hold for inner blocks; external scope for address linking.	Statement labels have program as scope; data names have local and block structured scopes.	System declarations hold for procedure scopes.	4 absolute scopes: COMPOOL, external, main, procedure declarations hold for inner blocks; scope further specified by block structure	
Compiler and loader directives							
- Directives							
• High level	Data structure equivalence; no listing control; no low-level code insertion; not extensible.	Macro string substitution; data structure OVERLAYing; low-level code with DIRECT; COMPOOL and procedure library calls.	Macro preprocessor with extensive compile time capability; concurrent processing control.	Macro string substitution; low-level code insertion; concurrent processing control.	Macro string substitution; compile time variables.	Macro string substitution; data structure OVERLAYing; conditional compilation; optimization directives; low-level code with DIRECT.	Macro instruction: sequence substitution
• Loader directives	Implicit in COMMON statement; EXTERNAL subprogram references.	Not provided	External definitions	No provided	External definitions and references for linking loader	Extensive external definitions and references for linking loader	Control sections similar to COMMON. external references for linking loader.

TABLE H3. TECHNICAL CHARACTERISTICS OF HOLs (CONTINUED)

Technical Characteristic \ Language	FORTRAN	JOVIAL J3	PL/I	SPL	CMS-2	JOVIAL J73	Representative Assembly Language
• Compile time variables	None	None	Integer, character string (in preprocessor); can be used in compile time evaluations	Declared variables can be made constants	Identifiers defined by EQUALS directive	No, but has compile time formulas	Yes; can be used in assembly time expressions
• Load, time variables	COMMON groups EXTERNAL subroutine names (ENTRY names)	None	Externalized address variables	None	System procedures, EXTERNAL definitions	Externalized (DEF) variables and names	Externalized address variables
- Declarations							
• Program structure declarations	SUBROUTINE ... END; FUNCTION ... END; PROGRAM ... END data declaration and executable statements may be intermingled.	PROGRAM...START ... TERM $ PROC...BEGIN... END $ CLOSE...BEGIN... END $ Variable names must be declared before they are used.	PROCEDURE... END Data declarations should be defined before being used.	START...TERM PROC...EXIT CLOSE...EXIT Variables must be declared before used.	System, system data, design, system procedure, local data design, procedure, and function declarations are allowed. Example: SYSTEM...END-SYSTEM	COMPOOL declarations; program declarations; procedures can be defined within procedures.	PROGRAM ... END Entry definitions
• Variable declarations methods							
- Implicit	Yes; first letter in identifier determines type; I, J, K, L, M are integer.	MODE directive specifies default declaration mode	Similar to FORTRAN; partially declared variables have default values set to remaining attributes	Same as JOVIAL J3	MODE directive specifies default declaration mode	None	None
- Contextual	Integer labels (in columns 1 to 5)	Labels; loop variables	Labels; loop variables	Labels and contextual items	Labels	Statement names; loop variables	Labels
- Explicit	INTEGER, REAL etc., type declarations.	In item declarations (I=interger, A=fixed, F=floating)	In variable declarations (FIXED, FLOATING, etc.)	In variable declarations (FIXED, FLOATING, INTEGER)	In variable declarations (I, A, F, and B = Boolean, H=Hollerith, S = status)	In variable declarations (U=unsigned fixed and integer, S=signal fixed and integer, F=floating, C=character)	Symbol definitions
- Multi-variable declaration formats	Many variables per attribute; one attribute per statement.	One variable per statement; all attributes in statement; no factoring.	Many variables per statement; all attributes in statement; factoring allowed.	Like in PL/I	Many variables per statement; all have same attributes.	Many variables per statement; all have same attributes.	One declaration per line

TABLE H3. TECHNICAL CHARACTERISTICS OF HOLs (CONTINUED)

Technical Characteristics \ Language	FORTRAN	JOVIAL J3	PL/I	SPL	CMS-2	JOVIAL J73	Representative Assembly Language
• Simple variable declarations							
- Data types allowed							
• Integer	X	X	Use fixed	X	X	X	Only addresses are used; programmer must associate address with data type held in addressed location
• Fixed point		X	X	X	X	X	
• Floating point	X	X	X	X	X	X	
• Logical (Boolean)	X	X	X	X	X	Equivalent	
• Status		X	No	X	X	X	
• Complex	X	No	X	No	No	No	
• Double precision	X	Equivalent	Equivalent	Equivalent	Equivalent	Equivalent	
• Label	No	No	X	No	No	No	
• Vector	No	No	X	X	No	No	
• Matrix	No	No		X	No	No	
• Alphanumeric	Yes - machine dependent	X	X	X	X	Character	
• Bit strings	No	Equivalent	X	X	Equivalent-tables	Equivalent - integers, tables	
• Pointer	No	No	X	No	No	X	
- Presetting of variables	Yes, (DATA statement)	Yes, in declaration	Yes, INITIAL attribute	Yes, with PRESET attribute	Yes, in declaration or DATA statement	Yes, but restricted to class of variables; similar to FORTRAN	Yes
- Allocation modes							
• At load time	All variables allocated required space	Default	Default	Default	Default	Default	Default
• At procedure entry	No	Yes	Variables with AUTOMATIC attributes	For local variables in recursive and re-entrant procedures	No, variables are permanent until explicitly changed	Can be controlled with environmental specifier	No
• Permanence	Variables are permanent until explicitly changed	Similar to FORTRAN	Controlled	Not controlled	Not controlled	Controlled; permanence can be keyed to outer level procedures; default is similar to FORTRAN	Not controlled

TABLE H3. TECHNICAL CHARACTERISTICS OF HOLs (CONTINUED)

Technical Characteristics \ Language	FORTRAN	JOVIAL J3	PL/1	SPL	CMS-2	JOVIAL J73	Representative Assembly Language
• Data structure declarations							
- Arrays							
• Maximum dimensions	3	Any number	Any number	Any number	3 (special table type)	Any number	1
• Subscript expressions	Restricted: Integer Constant + Constant	Any integer valued expression	Any expression; converted to integer	Any expression; converted to integer		Any expression, converted to integer	Used in address expression
• Adjustable dimension	Can be passed into subroutine as arguments	Not provided	Provided but fixed at allocation time	Can be passed into subroutine as arguments		Controlled through allocation increments	No
- Tables	Not provided		Not provided				Not provided
• Organization		Provided		Provided	3 dimensions plus fields, words	Any dimension plus	
• Control (serial or parallel)		Serial or parallel		Serial and parallel	Vertical (serial) and horizontal (parallel)	Serial and Tight	
• Packing options		No packing, dense, medium		Same as JOVIAL J3	NONE, MEDIUM, DENSE	Same as JOVIAL J3; medium is system dependent	
• Allocation control		Static		Static or automatic	Static	Flexible allocation increments	
- Hierarchical structures	Not provided	Not provided	Very flexible; variables, arrays may be elements	Less flexible than PL/1.	Tables, fields, words	Compound structures provided; referencing less flexible than PL/1	Not provided
• Structure equivalencing	Variables and arrays with EQUIVALENCE		Arbitrary depth		SUB-TABLES allow overlays	Flexible overlay control of tables and items	
- Allocation control for structures	Arrays allocated at compile and load time	Structures allocated at compile time	Default: Allocated at compile and load time; AUTOMATIC structures allocated at procedure entry; programmer has control.	Structures allocated at compile and load time	Structures allocated at compile and load time	Structures allocated according to permanence level of name; very flexible control.	Not provided
• Switch declarations	Assigned GOTO	Switch statement provides a more convenient label switch than FORTRAN	Label vectors	Switch declaration; more convenient than FORTRAN	Label switches; procedure switches; very flexible.	SWITCH statement provides a little more flexibility than FORTRAN-computed GOTO	Indexed jump instructions

TABLE H3. TECHNICAL CHARACTERISTICS OF HOLs (CONTINUED)

Technical Characteristics \ Language	FORTRAN	JOVIAL J3	PL/1	SPL	CMS-2	JOVIAL J73	Representative Assembly Language
• Executable statements							
- Computational expressions							
• Arithmetic (scalar)							
- Operators	+-*/ absolute remainder	Similar to FORTRAN	Similar to FORTRAN	Similar to FORTRAN	Similar to FORTRAN	Similar to FORTRAN	Unique machine instructions must be coded
- Mixed expressions	Not allowed	Allowed	Allowed	Allowed	Allowed	Allowed	
- Scaling	Not provided	Automatic conversion to floating point	Automatic conversion to floating point	Automatic or controlled scaling	Automatic or controlled scaling	Automatic and controlled scaling; very flexible evaluation control	
• Arithmetic (array) operations	Not provided	Not provided	Simple element-by-element operations provided	Vector and matrix operations provided	Not provided	Not provided	
• Relational operations	.LT., .LE., .EQ., .GE., .GT. (= complete set)	Complete set	Complete set	Complete set	LT, GT, LTEQ, GTEQ, EQ, NOT equal	Complete set	
• Boolean operations	.AND., .OR., .NOT., Not provided	AND, OR, NOT	AND, OR, NOT	AND, OR, NOT, EQUIV	AND, OR, COMplement	AND, OR, EQV, XOR, NOT	
• String operation	Not provided	Bit and byte extraction operations are provided	Boolean bit vector operating concatenation; bit and byte extraction	Boolean bit vector operations bit and byte extraction left and right shift	Bit and CHARacter extraction	Concatenation	
• Format	Infix	Infix	Infix	Infix	Infix	Infix with evaluation control	
• Execution rules	Normal precedence; left to right; reordering allowed	Normal precedence; strict left to right		Normal precedence; strict left to right	Normal precedence	Normal precedence	
- Environment altering statements							
• Assignment statements	Variable = expression	Variable = expression	Variable = expression	Variable = expression	SET variable TO expression	Variable = formula	Unique machine instructions must be coded
- Simple variables	Conversion automatic	Conversion automatic	Conversion automatic	Conversion automatic	Conversion automatic or controlled	Conversion automatic or controlled	
- Structured variables	Not provided	Not provided	Arrays, hierarchical structures	Arrays	Not provided	Not provided	
- Multi-variable	Not provided	Not provided	Allowed	Allowed	Multi-word assignments	Multi-variable, multi-formula	
- Exchange statement	Not provided	Scalars only	Not provided	Scalars and arrays	SWAP...AND...	Scalars only	

TABLE H3. TECHNICAL CHARACTERISTICS OF HOLs (CONTINUED)

Technical Characteristics \ Language	FORTRAN	JOVIAL J3	PL/I	SPL	CMS-2	JOVIAL J73	Representative Assembly Language
• Control statements							
- Unconditional jump	GOTO label	GOTO label, close name, program name	GOTO label	GOTO label, close name	GOTO label	GOTO statement, label	Jump instructions
- Selective jump	Computed GOTO	Index switch list	Label array	Index switch array or switch-list	GOTO switch index	SWITCH statement	Indexed jumps
- Jump to label or switch variable	Assigned GOTO	GOTO switch variable	GOTO label variable	GOTO switch (index)	Not provided	Not provided	Jump to register specified address
- Scope of jumps	To labels within subroutine program	To local or outer scope	To local or outer scope	Anywhere in the program	To labels within program or procedures	To local or outer scope	Anywhere in program
• Conditional statement	Arithmetic IF:	Basic IF:	Basic IF:				
- ae = arithmetic expression	IF(ae) L_1, L_2, L_3;	IF le $ s $	IF le THEN s ;	IF le ss	IF le THEN ss $	IF le ; cs ;	Conditional jump instructions
- le = logical expression	Logical IF:	Alternative IF:	IF le THEN s ELSE s ;	IF le THEN s ELSE s END;	IF DATA FOUND THEN ss $	IF le ; cs ELSE s ;	
- L = label	IF (ls) ss	IFEITH le $ s $ ORIF le $ s $ ... ORIF le $ s $		Alternative IF: IF le THEN s ... ORIF le THEN s . . . ORIF le THEN s ELSE S END	Validity decision: IF table VALID THEN ss $		
- s = statement							
- ss = simple statement							
- cs = compound statement							
• Looping control statements							
- Basic control clause/statement v = variable f = first value l = last value incr = increment	DO L v = f, l, incr	FOR v=f, l, incr$	DO WHILE le; DO = TO l BY incr	WHILE le UNTIL le FOR v=f BY incr UNTIL l	VARY v FROM f THRU l BY incr $	FOR v (WHILE le) FOR v (ae BY ae UNTIL v=ae)	Conditional jump instructions
- Termination	Statement with label L	First statement after FOR is repeated; can be compound	END statement	END statement	END statement	First statement after FOR is repeated; can be compound	
- Nesting	Any level	Any level via compound statements	Any level; END's match control	Any level; END's match control	Any level; END's match control	Any level	No restriction
- Parallel loop variables	Not provided	With added FOR statements	Not provided	Provided with ALSO; FOR a = ... ALSO vb = ... END	Provided with AND extension to VARY statement	With added FOR statements	No restriction

TABLE H3. TECHNICAL CHARACTERISTICS OF HOLs (CONCLUDED)

Technical Characteristics \ Language	FORTRAN	JOVIAL J3	PL/I	SPL	CMS-2	JOVIAL J73	Representative Assembly Language
Subprogram characteristics							
- Macros (open subprogram)	Statement function; restricted use	Not provided	Provided through preprocessor	Not provided but closed statements are nearly equivalent	Provided through EQUALS and MEANS substitution scheme	Provided through "DEFINE" substitution scheme	Provided with macro definition capability
- Closed subprograms							
• Restricted scope	Statement function is equivalent	Closed statement; no parameters; only usable in program where it is defined.	Procedures can be used in context of block structure such that their scope is restricted to a limited number of blocks in a program	Closed statement; same as JOVIAL J3.	Not provided	Same as PL/I	Not provided
• Independent scope	Subroutines and FUNCTIONS	Procedures	Procedures	Procedures	Procedures	Procedures and functions	Independently assembled subroutine
- Calling format	CALL name ($r_i, \ldots r_n$)	Name ($i_1, \ldots i_m = v_1, \ldots, v_n$)	CALL name ($r_1, \ldots r_n$)	Name ($i_1, \ldots i_m = v_1, \ldots, v_n$)	Name INPUT $i, \ldots$ OUTPUT $v_1 \ldots$ EXIT e_1 $; Name USING index INVALID statement INPUT $i_1 \ldots$ OUTPUT $r_1 \ldots$ $	Name ($i_1, \ldots,$ i_n: $v_1, \ldots, v_n$)	No restriction
- Entry mode	Simple entry only	Simple entry only	Can be declared as recursive	Can be declared as recursive	Recursive capability not specified	Can be declared as recursive	No restriction
- Parameter passing	Call by reference	Call by value is normal mode; call by reference for arrays.	Call by reference	Call by value is normal mode; call by reference for arrays.	Call by value only; call by reference can be achieved but with inconvenience.	Call by value is normal mode; call by reference for arrays.	Arbitrary
- Local variable scope	All names are local to the subprogram in which they are defined	Names defined in procedures are local	INTERNAL names are local	LOCAL names have restricted scope	Local names are restricted to procedure in which they are defined	Names defined in procedure are local	Local to entire assembly
- Global variable use	Variables in COMMON are globally available in subprograms where COMMON name is defined	Main program variables available to subprograms. COMPOOL variables available to all.	Variables from outer blocks are accessible to inner block procedure. EXTERNAL variables are accessible to all procedures where the variable is declared external.	Same as JOVIAL J3	COMPOOL and system data designs are globally defined; external variables are accessible to independently compiled procedures.	COMPOOL variables available to all procedures; outer scope variables available to inner scope.	Not explicitly provided
- Memory allocation	Permanently allocated	Permanently allocated	Automatic allocation on entry and permanent allocation optional for procedures; controlled allocation available.	Permanently allocated	Permanently allocated at load time	Very flexible allocation control with "permanence" attribute	Permanently allocated

Executable statements handled by the languages are presented in terms of their capabilities for evaluating computational expressions, and in terms of the statements altering the environment. The latter category is further subdivided in terms of assignment statements which change values held in variables, in terms of control statements which determine the execution sequencing, in terms of conditional statements whose execution is data dependent, and in terms of looping control statements which control the repetitious execution of code segments. The control statements, conditional statements and looping-control statements are environment-altering statements in that they control the program execution pointer, one of the state variables in an executing program.

One important class of executable statements has been omitted; the input and output statements. In avionic systems these are very system dependent and would be the most likely functions to be programmed in a low-level language. Also, most of the input/output statements in high-level languages are oriented toward character handling I/O peripherals. This type of I/O has little use in operational avionic systems.

The subprogram characteristics of a programming language were made a separate category because of the importance of modularity in avionics applications. All of the described languages have good subprogram definition and call capabilities. The main differences are in the parameter passing modes and in the treatment of the use of local and global variables.

HOLs versus Assembly Languages -- The main advantage of assembly languages is the fact that the programmer may access the details of the machine's structure and, thus, may write codes slightly more efficient than those generated by a compiler. However, assembly language programming tends to be less structural, less understandable, more machine-dependent, and less communicative of programmer intent, etc., than a high-order language.

Probably the best approach now available is to code the programs in an HOL (avoiding as much as possible any assembly-language coding), compile the code, and test it. If the compiled code does not run as fast or takes more space than desired, alter the HOL source to tune it up. If necessary, the programmer may have to go into the assembly language on some routines but this should be kept to a minimum.

The disadvantages of assembly language compared to the advantages of HOLs cause us to feel that, with the nominal loss of efficiency attributable to modern compilation, assembly-language coding should be avoided as much as possible.

New Concepts in Higher-Order Languages

The preceding sections have discussed the current state of the various languages applicable to avionic applications. However, the problems encountered in developing and implementing avionic systems are not unlike those found in the more general software areas. Accordingly, we now will address some more global issues concerning the relationship of HOLs to the general software dilemma. In the following sections we address some of the more controversial and recent concepts.

Data/Program Structure Relationships -- A higher-order language definition implies a set of conceptual constructions to be used in designing and programming an avionic software system. These constructions form the basis for the data objects and the program code for manipulating these data objects. The role of an HOL in designing and implementing avionics systems is much like that of a mathematical notation in solving analytic problems. This point is illustrated by the notation introduced by Iverson, which has become a popular and powerful interactive language called APL. In this case, the organization aspects of APL, along with the availability of operators to deal into the complex data structures, make APL a powerful language for mathematical problems.

In the avionics environment, the construction of software programs involves basic operations and structures. An HOL which facilitates the representation of specialized constructs within avionics systems is an important tool. Additional benefits of such an HOL would be realized in the areas of modularity and program validity, as discussed above. Finally, another advantage of such an HOL would be that it would provide a conceptual basis for thinking about, designing, and implementing the system. The importance of the tool in this respect is relatively new and controversial (Ref. H4).

The concepts of an HOL have been used in most commercial systems, both in the construction of user programs and in the operating system. An example of an operating system in an HOL is the MULTICS system (Ref. H5). Such systems based on an HOL are designed using data structures as provided by the host HOL. The disadvantage now of this approach is its inefficiency. The machine code produced by translation or emulation from an HOL is seldom optimal. However, the advantage of easier management and control of large systems provided by an HOL will offer a necessary future compromise.

Extendable Languages -- An HOL which is extendable provides the capability for incorporating specialized primitives in a natural way into the HOL. An HOL is not extendable by virtue of its modularization. The key concept of extending an HOL is an ability to modify or specialize existing primitives and features, to serve a specific purpose. For example, ALGOL or PL/I can be extended in system-provided procedures because any procedure can be redefined globally to arbitrary blocks of code.

A specific system usually evolves from certain basic primitive functions. The role of an HOL in providing several such basic primitives was discussed above. However, a general-purpose HOL cannot effectively provide all things to all applications. By providing for extendability, an HOL allows those primitives specific to an application to be incorporated as necessary. Accordingly, application can be developed with all of the advantages of a complete consistent HOL. The orientation of the HOL to the application resulting from its extension provides for a convenient framework of work assignments during the implementation of the avionics system.

An HOL designed for individual application is often not suitable for other uses. Such an HOL contains a set of primitives which lend themselves very well to the intended use. These primitives are formulated and described in a notation natural for the application. Thus, using this HOL for a differing application is difficult. Accordingly, different languages with different standards have come into existence. Many differing HOLs have thus been specialized for individual applications, resulting in a suitable HOL for each area.

In the avionics environment, the range of applications is far less diverse than the general commercial field. However, differing thrusts within the avionics environment will require an HOL adaptable to differing orientations. The extensibility of an HOL can be an advantage, while the HOL itself remains to provide a system standard. This is accomplished by basing the system on constructions inherent in the HOL and ensuring that specialized applications are proper extensions from this base.

Source Level Modularity -- This section explores the concept of source-level modularity as it relates to HOLs. Source-level modularity implies that the modularity requirement need not be carried to the object or machine level. The deployed system will not require the degree of modularity that would exist with all required modularity within the deployed object system. The advantage of an HOL is that the potential organizational and management considerations of modularity need not exist within the deployed system. The disadvantages are increased costs for system generation, and a technique which is displeasing to a machine-oriented programmer. Three aspects of source level modularity make it an attractive approach to such high-requirement applications as avionics: the ability for system integration in source before compilation; the possibilities for conditional inclusion of source; and the possibilities for optimization based on source-level aids.

In the previous sections, the notion of a total system description was compared with the data and program structures of an HOL. There, the advantage of an HOL was in its ability to form a basis for the total structure of the system. To a lesser degree, organization based on an HOL consisting of modules can be designed and implemented at the source level. This is what happens, for example, during the compilation and loading (linking)

phases of FORTRAN-object program generation. However, the load-link philosophy implies that the modularity at the source level will be retained at the object level as well.

The notion of a program construction consists of creating subsystems by combining smaller modules. In the process of constructing a subsystem, the individuality of the combined modules may be lost. This results in a more efficient or compact subsystem. Because the basis for the design of a system is a source-level modularity, the blending of modules into a homogeneous object package does not lessen the original advantage. An additional advantage is that the homogeneous object package can be optimized as desired.

Conditional compilation is a feature within the source level of a language which exists in contemporary assembly languages. It allows including source characters, conditional on certain flags or relationships, in the source at compile time. The advantage is that source characters may be placed into a program and removed by simply altering a conditional variable. In applications involving changing configurations, the differing software configurations can be established by simply specifying the appropriate constants. Debugging constructions can be included in the source in the form of tracing tools and software redundancy. Once a program has been satisfactorily debugged, these constructions can be removed via the conditional exclusion of source characters.

The third aspect of source-level modularity within an HOL is that of optimization. At the source level, module behavior, interfaces, and data dependencies are more readily available. With this type of information, the problem of meaningful optimization is more easily dealt with. Information can be exploited to perform the optimization processes and then dropped during the translation of the source program to object code. The advantage of source-level modularity here is that the resulting object code is free from redundancy and information constructions which apply only to program construction.

Source-level modularity implies that the construction of the object system is automated via the HOL compilation. This fact implies that all manipulation of the system code by a programmer must occur within the framework of the source code. In the avionics environment, this will represent a significant change in practice. However, we should emphasize that the primary function of a good HOL as discussed here is to provide a structural basis for the system. Within the structural basis, individual modules can be designed and implemented. Program generation will produce an object code. With sufficient debugging constructions, errors in object behavior can be related to the appropriate source constructions and the repairs made at the source level.

Structured Programming -- Structured programming increases program reliability. As described by Liskov (Ref. 3), structured programming is characterized by two rules. The first is that structured programs are developed from the top down, in levels. Each level is a collection of components or subsystems. The components of a level are themselves structured programs, which consist of components at yet a lower level. The decomposition of components ultimately leads to the basic primitives of the language used. A second rule of structured programs dictates the way in which components can be combined into control structures. Only the following control structures are permitted: concatenation; selection of the next statement based on the testing of a condition; and iteration. The significant structure not allowed is a transfer of control via the goto statement.

Structured programming limits program control structures to those which lend themselves to proofs of correctness. The process of proving correctness consists of examining each level of the program and asserting that the level meets its specifications, if the components from lower levels do. This form of inductive reasoning down the levels of the program eventually leads to components which are basic primitives of the implementation language. These primitives meet specifications to the extent that the implementation language does what it is supposed to do.

The proof process depends in two ways on the control structures allowed in structured programming. First, the allowable structures ensure that control enters and leaves an individual structure in a unique fashion. Only a single entry and exit is permitted. Control always leaves one structure and "falls into" the next. This limits the reasoning to the flow of control as it pertains to the component being analyzed. The second dependency is that each of the three control structures is associated with a well-known rule of inference. Concatenation is associated with linear reasoning, iteration is associated with induction, and conditional selection is associated with analysis-by-cases. These rules allow each of the components of the system to be easily understood.

The discussion to this point has examined structured programming as it facilitates analytic proofs of program correctness. Because of the restrictions imposed by the three allowed control structures, certain application programs may be much less than optimally implemented. It would seem that a compromise situation exists. A system designed with structured programming (hence attempting to minimize control structures not one of the above three), would have greater reliability. This requires that designers and programmers accept the above discipline.

SOFTWARE VERIFICATION AND VALIDATION

There are three main steps in testing software:

1) Selecting test cases
2) Evaluating their execution results
3) Estimating correctness (inductive leap correctness assuming test cases were good choices and executed properly).

While this testing process is well known, the complexity and inexactness of software precludes a good validation procedure. Such new techniques as program provers are not yet feasible since they are only applicable to very small programs and are impractical. The only practical way now to have confidence in the test cases is proper construction. This means modularly structured programming with interfaces. However, the current state of the art is such that no established procedures exist for ensuring that system developments result in well-built systems. Nevertheless, it is reasonable to assume that such current research areas as structured programming and modularity analysis will produce the needed procedures.

BOTTOM-UP CONSIDERATIONS

Introduction

An alternative way to solve portions of the software dillemma is to provide hardware support to various software functions. This approach has some advantages:

- May enforce clean interfaces
- Enforces standards
- Provides a faster execution
- Provides a lower level of detail to software

There are a number of disadvantages to this approach:

- Loss of flexibility
- Implementation difficulty in implementing special functions
- Increased size, weight, power, gates, etc.

Some of the areas in which hardware may be used to support software are described below.

Hardware/Software Interface Primitives

One of the key software areas that can be implemented in hardware or assisted by hardware is that of executive functions (Ref. H6).

Typical functions that one might support are described below. These functions have been taken from Reference H6, and range from I/O support to memory hierarchy support.

Basically, any multiprocessor or multicomputer system can benefit from hardware supporting executive functions. Typically, the most common and useful piece of hardware is the associative memory (AM) or its derivations. The AM can provide several orders of magnitude speed-up in several functions, thereby making a significant impact on system throughput. Associative logic is regular, thereby providing a high-density, powerful system. The time taken to search tables and lists, and the required indexing and link manipulation hardware and software are not required when a search memory is available. This cuts down overall system complexity, overhead, and execution time. The executive programming cost is also reduced and the programmer does not have to worry about the impact of the selected data structure on his program's execution speed.

Hardware associative memory aids can be applied to executive functions in the following areas:

- Virtual memory -- Translation tables, replacement algorithm hardware (see paper by D. Jensen, COMPCON 72), etc.
- Interrupt processing -- Priority interrupts, processing of interrupts of same level of priority (see paper by D. Jensen, 1970 FJCC), etc.
- Resource allocation -- Determination of available resources, determination of resources in test (see paper by D.C. Gunderson, WESCON 1966), etc.
- Scheduling -- Determination of next job, assist to deadline (variable priority) scheduling (see paper by Berg and Thurber, NEC 1972, Thurber and Jack, COMPCON 73) etc.
- Monitoring -- General statistic collection, high-speed resource monitoring, etc.
- Tables -- Any functions which require multikeyed searches over lists of parameters, status, or capability can be easily supported.

- Mappings/other arbitrary functions -- Any functions which require a dynamic translation mapping, a logical or physical binding, or a translation of any sort can be readily supported.
- Relocation -- Functions which deal with the difference between a logical and physical location, such as relocating a program, can be supported.
- Memory maps -- Even if a virtual memory is not used, memory maps must be kept. Easy access to I/O buffer location and usage, entry points to code blocks, allocation and deallocation of memory space, etc., information is readily provided.
- Protection -- Dynamic protection schemes.
- Sharing -- Sharing of files and data.
- I/O -- Chained I/O can be simplified and new requests can conceivably be chained to existing lists without a large real-time penalty, since a list would not have to be searched to insert into the middle of an I/O chain.
- Semaphores -- Semaphores can be supported as a two-way index; i.e., one operation can clear the semaphore list of all processes waiting on that single event, additionally, the correct fields of processes waiting on multiple events can be easily cleared.
- Time Outs -- Time-out manipulation is relatively easy so that one can afford to use a fast clock routine and keep close track of time to run, etc.
- Reentrant code -- Data pointers are easy to construct and maintain.

The advantage of implementing these functions in hardware is that they provide an efficient interface between the hardware and applications software. Also, the software functions which were replaced can be thoroughly checked out, as can the reliability of the associated hardware block.

The other main area of inquiry is the I/O. Since I/O is very well defined, its functions may be put into hardware, thus giving the attendant advantages of fast execution and reliability.

Architectural Functions

There are a number of functions that could be implemented in hardware which would support the system functions and ease system implementation. A detailed list of these functions was constructed (Ref. H6). These functions are described as follows:

- Enqueue/dequeue: These would be useful, particularly for an operating system built around conventional system balance concepts in past operating systems.

- Procedure call: (Stack or equivalent) Would be useful for reentrant and recursive procedures, especially if the programming language is CMS-2 or PL-1.

- Wait for: (a or b, a and b) Allows tasks to synchronize their activities.

- Hardware protection: For example, the MULTICS software gatekeeper can cost milliseconds to permit ring boundary traversal.

- Tag interpretation: Evaluate tags in context of operation, for example, an operand may be the name of a process or a pointer to the actual operand (e.g., B5000 EVAL operator, R-2 machine).

- Test and set: To allow multiple processors and/or processes to use a common code together (READ LOCK instruction in B6500, test and set in UNIVAC 1108).

- Dynamic linking: Static link resolution prior to allocation and execution of a program is a big OS task. Hardware support for dynamic linking (e.g., MULTICS) would be advantageous.

- Process creation: This primitive is fundamental and should have a very clean interface to the rest of the OS.

- Load and unload primitives: There must be mechanisms for loading (portions of) a program into a PE and for transferring new data from TM into RAMM; these mechanisms can be viewed as primitives. It would then be possible for the executive to send a LOAD instruction to a PE, which would cause the loading of a kernel page. (This implies that the processor is in a "ready to receive" state when it is not executing.) Similarly, an UNLOAD instruction could be executed by the PE to transfer data from TM to RAMM.

- Wakeup and block: The first changes the execution state of a process to logically ready, the latter to logically block.

- Request and release: Primitives suggested by Weiderman for resource synchronization.

The main use of these functions is to implement communication schemes in multiple-processing element systems. More detailed descriptions of some of the communication functions typically implemented in current systems are given below.

The Concept of Semaphore -- Semaphores allow processes to synchronize themselves. Basically, semaphores allow processes to act relatively independently of their environment. The primitives used by Dijkstra are P(sem) and V(sem), where sem is a system semaphore. Sem consists of a variable and a list. Sem may be a binary variable (0 or 1) or, more generally, an integer variable. Assume sem is an integer variable. Then, the synchronization primitives appear as given below:

-P(sem): Decrement by 1, sem. Then if sem < 0 put the process on the blocked process list for sem.

+V(sem): Increment by 1, sem. If sem ≥ 0 remove a process from the blocked process list.

One notes that a semaphore must be accessed by only one process at a time; if two operations are invoked on a semaphore, they are executed in an undetermined order so that they should be logically indistinguishable and the semaphore operation must be completed before it can be interrupted.

Processes are potentially blocked on a P(sem) operation and reactivated only when a V(sem) operation occurs. V(sem) removes some form of roadblock, which allows a process to continue its execution on some programmer specified condition.

IBM OS/360 Primitives -- IBM provides four basic macroinstruction synchronization primitives: ENQ; DEQ; WAIT; and POST. ENQ and DEQ are analogs of P and V, while WAIT and POST are analogs of block and wakeup instructions. (See Lampson, 1969 FJCC.)

ENQ(DEQ) operates on an eight-character parameter string (alphanumeric) called a resource. Each resource has a list of ENQed objects which represents both blocked processes waiting to use the resource and processes which are using the resource. A process can specify whether it wants shared or exclusive use on a conditional or unconditional request, thus ENQ is more general than P.

The status of the request is as follows:

1) If grantable, process is queued on the last and remains logically active
2) Unconditional and not grantable, process is queued and becomes logically blocked
3) Conditional and not grantable, process is told that the resource is unavailable

ENQ and DEQ are in some sense more restrictive than P and V because they must be paired, i.e., every ENQ on a resource must be followed by a DEQ and no DEQ can occur (in execution) before an ENQ on the resource. Therefore, a resource must represent a single specific item such as a tape drive, whereas Dijkstra's semaphore can represent a pool of tape drives (in the integer variable case). Also, the count associated with sem provides a means of counting the number of resources in the pool (if the counter is positive), or the number of processes waiting for the resource (if the counter is negative). It is advantageous to handle pools of resources (such as tape drives) in a pooled fashion, rather than as individual items because they are all basically the same.

WAIT and POST are analogs to block and wakeup. WAIT operates on an event control block (ECB) and blocks (logically) a process, unless the ECB has previously been POSTED. POST awakens a process associated with an ECB if it has been blocked. (If no process is waiting, POST sets a flag in the ECB.) WAIT can be generalized to allow a process to wait for n out of m possible events ($n \leq m$), but the major drawback is that it is impossible for more than one process to wait for a single event. Basically, WAIT and POST should be used only in cases in which Process A signals its completion to Process B.

Hansen's Primitives for Process Synchronization -- Some primitives for synchronization have been aimed at message and data passing. The philosophy is an interprocess message-handling capability and will help solve synchronization problems. A similar concept is discussed concerning the data and control interfaces supported by RAMM of AADC. As an example of such primitives, consider those developed by P. Brinch Hansen in connection with the RC 4000 multiprogramming system of A/S Regnecentralen, Copenhagen, Denmark.

Hansen's four primitives have the form:

1) Send message (receiver, message, buffer)
2) Wait message (sender, message, buffer)
3) Send answer (result, answer, buffer)
4) Wait answer (result, answer, buffer)

Send message 1) obtains a fixed length message buffer from a buffer pool, 2) places message into the buffer, 3) and queues the buffer in the message queue of the specified receiver. The sending process is informed of the location of the buffer so it can wait for an answer. The message buffer also contains the sender's identity. Should the intended receiver of the message no longer be in the system, the sender is sent an answer by the system. If the receiver of a message is waiting, its state is changed from blocked (logically) to ready.

The wait message suspends the calling process until it receives a message in its queue. At that time, the name of the sender and message are copied into the receiving processes data area, the buffer is taken off the queue and its location returned to the calling process for answer transmission. The wait answer is analogous to the wait message, except that, on receipt of the answer, the buffer is released to the buffer pool.

The send answer copies the answer into the buffer where the message was received and places it into the sender's queue. The sender's execution state goes to ready if it was waiting for an answer.

These primitives impose very rigid conventions for process communication. This may enable hardware support to be built. In Honeywell's system, the RAMM SFE can readily support such concepts. The rigid conventions also make things easier to analyze and understand. However, the primitives may not be economical, requiring the use of other primitives also.

Other Primitives -- MULTICS uses an exchange of messages in a common "mailbox" whose identity is known to the user processes by convention. The notify primitive returns the first message to be put in the mailbox, if any (otherwise it blocks the process).

GECOS (GE 600 series) uses a generalized WAIT and POST concept. The shared objects are called events and a process can request notification when an event occurs by using a notify primitive. The event occurs when a process issues a CAUSE primitive to the event. The CAUSE then activates all processes waiting on the event.

The use of precedence fields to pass control and data boundaries between PMs in the MEC of AADC (Ref. H6) is another example of a synchronization concept. In this case, RAMM creates the I/O Communication buffers for messages and I/O. Also, the process [POST (process name) after (time)] manipulation facilities support process synchronization as detailed in Reference H6.

AVIONIC SOFTWARE LIBRARY

The objectives of this subsection are two-fold. First, an investigation of the feasibility of applying the modular library concept to a multifunctional avionics problem will be made. Secondly, a study of the benefits to be gained by the application of such a library will be made. These investigations will be made such that the functional divisions inherent in the avionics subsystems considered are preserved.

The approach used during these evaluations will begin with the selection of several avionics functions for consideration. Based on these selected functions, assessments of the applicability of the modular library concept will be made. These assessments will be extended from those considered to include a wide range of avionics functions. In addition, the library concept will be considered at different levels within the selected functions.

Summary and Conclusions

A library of avionics software appears to be both feasible and beneficial. In the development of this library, it must be recognized that significant differences exist in the software for given subsystems as a function of implementation and/or application. Consequently, the library must be developed at several different levels ranging from software for an entire function, key modules, and subroutines. The benefit of the avionics software library is that recurring software costs can be reduced.

A summary of avionics software that should be included in the library is shown in Figure H4. It is recommended that the software for functions such as LORAN, air data, OMEGA, and certain navigation subroutines can be standardized. Changes as a function of subsystem mechanization will be minimal. Development of library software for INS, display, and flight control is recommended at a subfunction level. A number of INS mechanizations are possible. However, there is a significant degree of software similarity between all of them. A library of key software modules should be developed with standardized interfaces. Many display functions are relatively independent of the aircraft or avionics systems. Consequently, software modules to drive integrated displays should be developed. Different equation subroutines are required to support digital flight control.

The algorithm for a Kalman filter is then independent of the application; however, the size of the matrices and the input data is unique to each application. Consequently, matrix operations will best facilitate the software development for a Kalman filter.

It is recommended that these software modules be developed at the source language level so long as the data processing hardware is not standardized. If the hardware is standardized, then an assembly language library is a reasonable alternative. In either case, it is recommended that the development of the library be preceded by the development of an easily understood Engligh language and mathematical description of each avionics function.

Benefits from this library will result from reuse of a given module across avionic subsystem applications and reuse of the same subroutine across functions within a given avionic system.

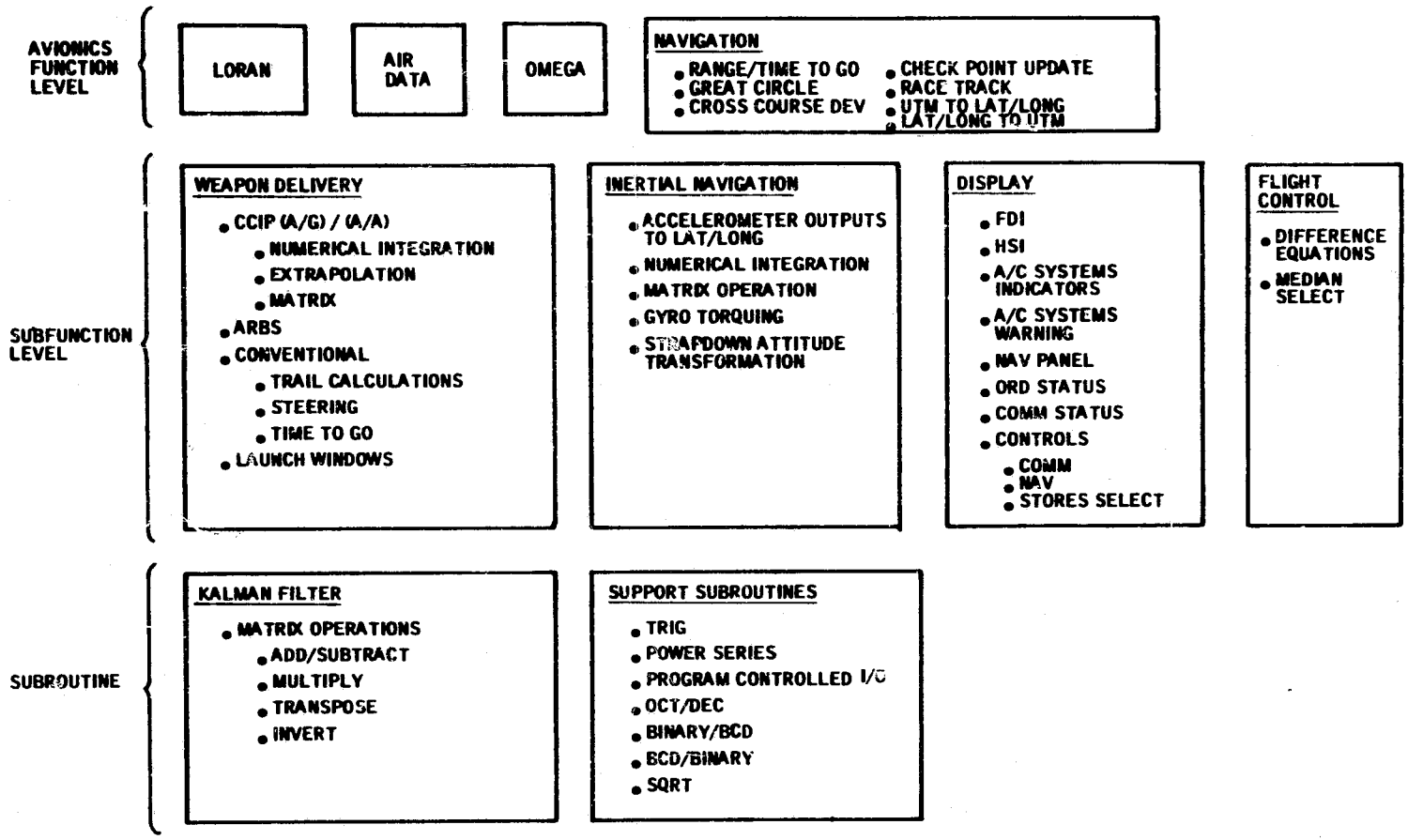


Figure H4. Software Summary

Selection of Avionic Subsystems for Consideration

The selection of avionic components to be considered for use in a modular library must be based on properties of the components which affect that modularity. The amount of software which applies from one mission to the next is restricted by the changes in the software that occur from one mission to the next.

The items which are considered most important in this respect include computational precision requirements, data constant values, routine parameters, variations in algorithm implementations, and variation in module communication requirements. Each selected function must be evaluated in terms of each of these items to obtain a true assessment of the applicability of the module concept to that function.

In obtaining a true evaluation of the feasibility of the modular library concept, the avionics subsystems considered must be representative of the entire range of potential applications. In making subsystem selections, therefore, it is necessary to choose those which are likely to require recoding as a function of mission and/or subsystem mechanization as well as those which are not. The range of reusability is expected to extend from those modules which are completely reusable (except for data constants* to those that may be highly application dependent and may not be reusable at all.

Characteristic of the first type of reuseability mentioned is a LORAN type navigation system. Its implementation is likely to change little from one mission to the next. Kalman filter data processing, on the other hand, is dependent on the application to the extent that gathering the input quantities into the state variables matrices is new and unique each time a new system is defined. The computational sequence required to operate on these matrices may remain fairly standard over a number of applications. Somewhere in between these two problems is an inertial navigation system of one type or another. Although the computations required depend on the particular implementation of the navigation system (e.g., platform or strapdown, etc.), it is expected that certain functions which are common to all navigation systems can be modularized. Such common functions might include coordinate transformation, or range calculations.

Since these avionic subsystems seem to illustrate the range required, the major emphasis of the investigations will be placed on

- LORAN systems
- Inertial navigation systems
- Kalman filter routines

Secondary emphasis will be given to air data, flight control, and display.

Software Library Feasibility Evaluations

Typical computational sequences for the three avionics functions to be considered are given in Figures H5, H6, and H7. For the inertial navigation subsystem, the sequence is given for each of several different implementations of the subsystem. Using these as a base, the application of the modularity concept can now be evaluated for each subsystem. The evaluations will be made with respect to the criteria mentioned earlier. A summary of these evaluations is presented as Table H4.

LORAN -- Of the three avionics subsystems considered, LORAN navigation provided the best opportunity for modularization. LORAN is a hyperbolic navigation system that transmits pulses at precise intervals. One station in the network serves as the master and the other two as slaves. The master station transmits a pulse train which is retransmitted by each slave at known time delays after receiving the master signal. The time-of-arrival of the three pulse trains is measured by a LORAN receiver and provides the basis for the calculations. First, expected time differences, based on present aircraft position, are computed. These are then compared to measure time differences to obtain errors in the computed position. Errors are then added algebraically to the computed present position to yield LORAN latitude and longitude (Ref. H7).

Although different LORAN networks may transmit with different delay constants, all require the same computational precision. 24-28 bits is typical to maintain the inherent accuracy of the LORAN calculation. Since LORAN calculations are performed typically at rates of 10 to 25 times each second, the position change between each calculation is small so a large dynamic range is not required on input data values. Since most present avionics computers have word lengths of 16 bits or less, double-precision calculations will also be a standard requirement for a LORAN subsystem.

Approximately 500 memory locations are required by LORAN for constant values. These include coordinate data on the LORAN transmitter network, coding delay required by each LORAN slave station, and various correction data to compensate for geometric, magnetic, and environmental factors. If modularity is handled at the source level, this data can be provided at compile time via a "block data program" in FORTRAN, a "COMPOOL" in JOVIAL, or a corresponding mechanism in another higher order language.

Since all LORAN mechanizations are based on the same concepts, there is justification that the computational implementation of the LORAN software can be standardized. Minor differences in algorithms exist in order to exploit the characteristics of one particular environment or the other; however, benefits obtained are small.

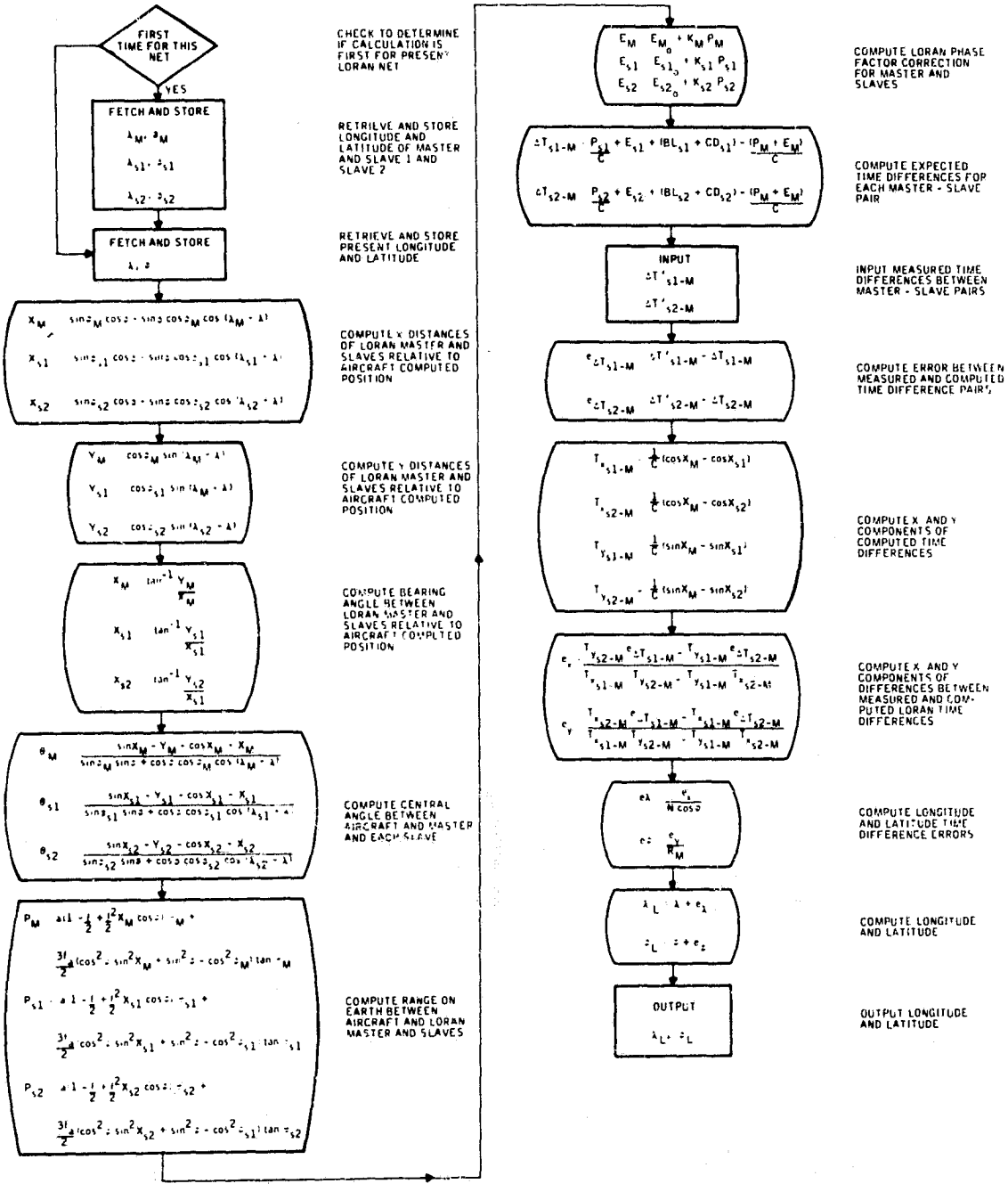


Figure H5. Computational Sequence for LORAN Navigation

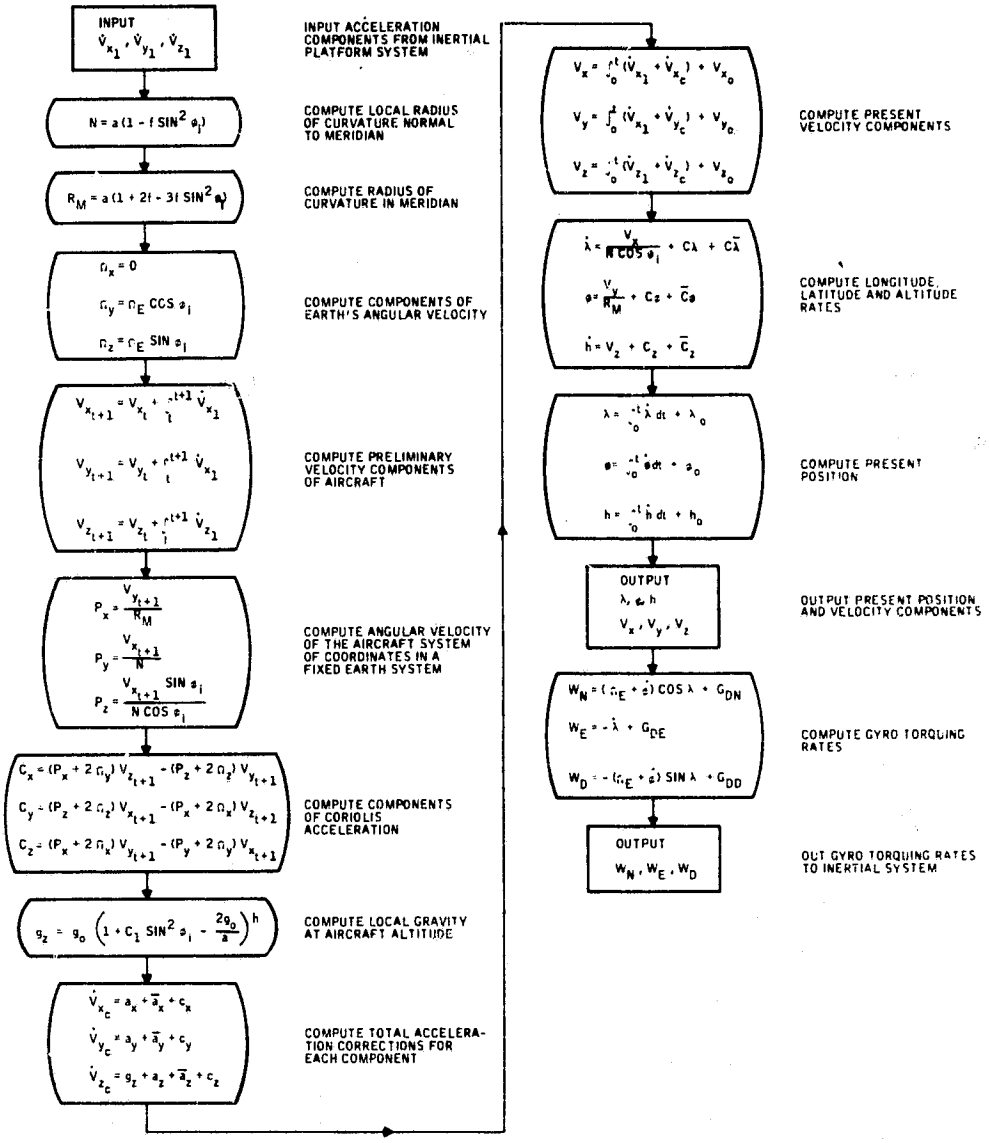


Figure H6. Computation Sequence for Platform Inertial Navigation

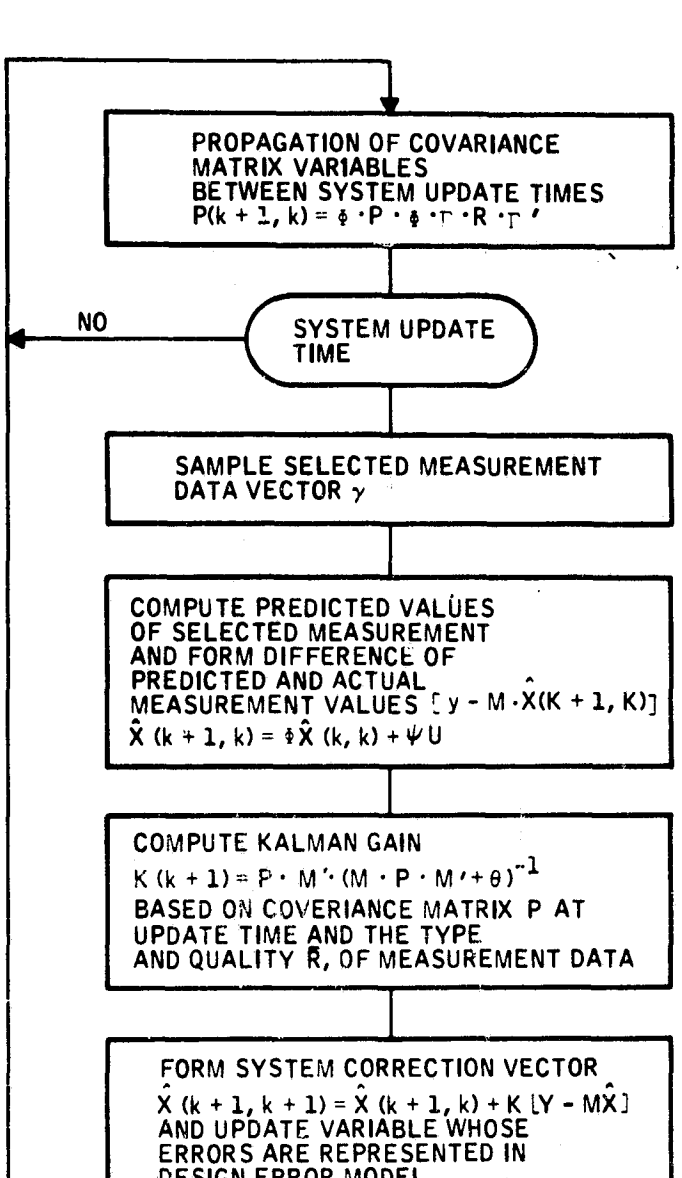


Figure H7. Computational Sequence for Discrete Kalman Filter

TABLE H4. MODULARIZATION SUMMARY

	Computational Precision	Constant Data	Parameters and Arguments	Algorithmic Variations	Program Controlled I/O	General Ease of Modularity
LORAN Navigation	• 24-28 bit precision typical for all LORAN mechanizations • Double precision required for most current avionics computers	• Transmitter coordinates • Coding delays • Environment propagation correction • Earth mag decl. correction	• Angular values • Coordinate data	LORAN mechanization all based on same concept so algorithm implementation can be standardized	• Input time differences from LORAN receiver • Output latitude/longitude values to other systems	Easy
Inertial Navigation	• Accuracy required is function of implementation. Differences may vary an order of magnitude or more.	• Gyro drift rates • Accelerometer biases • Initial position conditions	• Number of steps in numerical integrations	Algorithm to be used depends on implementation • Platform • Strapdown • Free azimuth • Polar and system alignment modes	• Sensor inputs from accelerometers • Gyro torquing signals • Latitude/longitude display outputs	Moderate
Kalman Filter Process	• Precision requirements change as a function of parameter values. • Wide dynamic range required to preserve contribution from all components. • Scaling required	• Initial conditions to Kalman matrices • Initial error estimates	• Order of matrices • Matrix starting locations • Precision requirement	Algorithm is independent of characteristics of data being analyzed	• Display output • Kalman error estimates	Difficult

Figure 20. Modularization Summary

One common method involves a table hookup procedure in part of the calculations. This requires a large amount of core for storage of the table values. A closed-form version is available which performs the calculation required with sufficient accuracy and reduced core requirements; however, computational time requirements are high. A compromise of these two methods is an iterative method which appears satisfactory from both the storage and speed requirements as well as accuracy. Even if these algorithm differences were critical to the successful performance of an avionics mission, multiple LORAN modules can exist in the library.

Inputs and outputs to the LORAN algorithm are limited and consist of time difference inputs at the algorithm iteration rate and outputs of latitude and longitude. LORAN is considered to be a good candidate for modularization.

Inertial Navigation -- Inertial navigation has a few characteristics which make it less easily modularized than LORAN navigation. None the less, inertial navigation is still considered a likely candidate for incorporation into a module library.

Inertial navigation computation determines current position, velocity, and gyro torquing rates using acceleration components supplied by the inertial system. These computations include the following steps:

- Measure vehicle nongravitational acceleration components
- Compute acceleration due to gravity and total vehicle acceleration
- Integrate acceleration to obtain current latitude and longitude and velocity components
- Compute gyro torquing rates

There are various implementations of inertial navigation of which free azimuth, strapdown, electrostatic gyro (ESG), and stationary platform are only a few. Each of these systems requires its own unique computational sequence. Accuracy requirements also depend on the implementation. An inertial navigation system based on ESGs requires an order of magnitude greater accuracy than most other implementations. This represents a potential problem which must not be overlooked.

For inertial navigation, some of the constant quantities to be used include gyroscope drift rates, accelerometer bias, and initial latitude and longitude inputs. For inertial navigation, these constant values are a function of the individual aircraft involved and of the position on earth where the mission is to take place.

The number of arguments required to implement the inertial navigation package is small. Arguments include accuracy requirements and the number of numerical integration steps required along the way.

The most serious problems encountered when trying to implement inertial navigation software in a modular fashion will be due to difference in algorithms required by different hardware systems. As an example of this, consider the three systems shown in Figure H8 (Ref. H8). Figure H8 shows a system known as a "North-East-down" system. Accelerometers are used to measure acceleration along two axes which are held normal to each other and normal to a nominal vertical axis. These are corrected for bias errors, Coriolis acceleration, and integrated to obtain the angular velocities with respect to these coordinates. By appropriate trigonometric operations, these can be converted to a vector with three components which are used as gyro torquing inputs to re-align the platform. Figure 21b shows a system known as "free azimuth." This is similar to the North-East-down system except that no torquing is applied along the vertical axis. Thus, the azimuth gimbal will have zero angular velocity about the vertical axis. In the North-East-down system the inertial coordinate system is kept close to the geographic coordinate system. This is not true with a free azimuth system, and it is therefore necessary to apply coordinate transformations to the accelerometer inputs and to the torquing outputs. The free azimuth system requires more computation but has distinct advantages of accuracy when flying near the geographic poles. Near the poles the rotations around the vertical axis can become large. The required torquing signals force the gyros to be able to respond to a large dynamic range of inputs. Thus, the North-East-down system has accuracy problems near the poles. Since, in the free azimuth system, no attempt is made to torque the system about the vertical axis, it does not have the same problem, but it requires more computations.

Comparing Figure H8a with Figure H8b, the computational requirements for the "North-East-down" system and the "free azimuth" system are nearly the same. The exceptions are that the free azimuth system requires some coordinate transformations at both input and the output. These observations will be important to the conclusions made with respect to the modularity of inertial navigation sfotware.

A third example is a "non-rotating platform" inertial navigation system (Figure H8c). In this system the platform does not rotate with respect to an inertial system. This is accomplished via three gyros placed on a fourth gimbal in the system. Accelerometers, also mounted on the fourth gimbal, provide inputs of movement. Transformations are again required, and torquing is required with respect to three orthogonal axis. This system, however, requires a three-dimensional coordinate transformation, and the computations required break down in the polar regions.

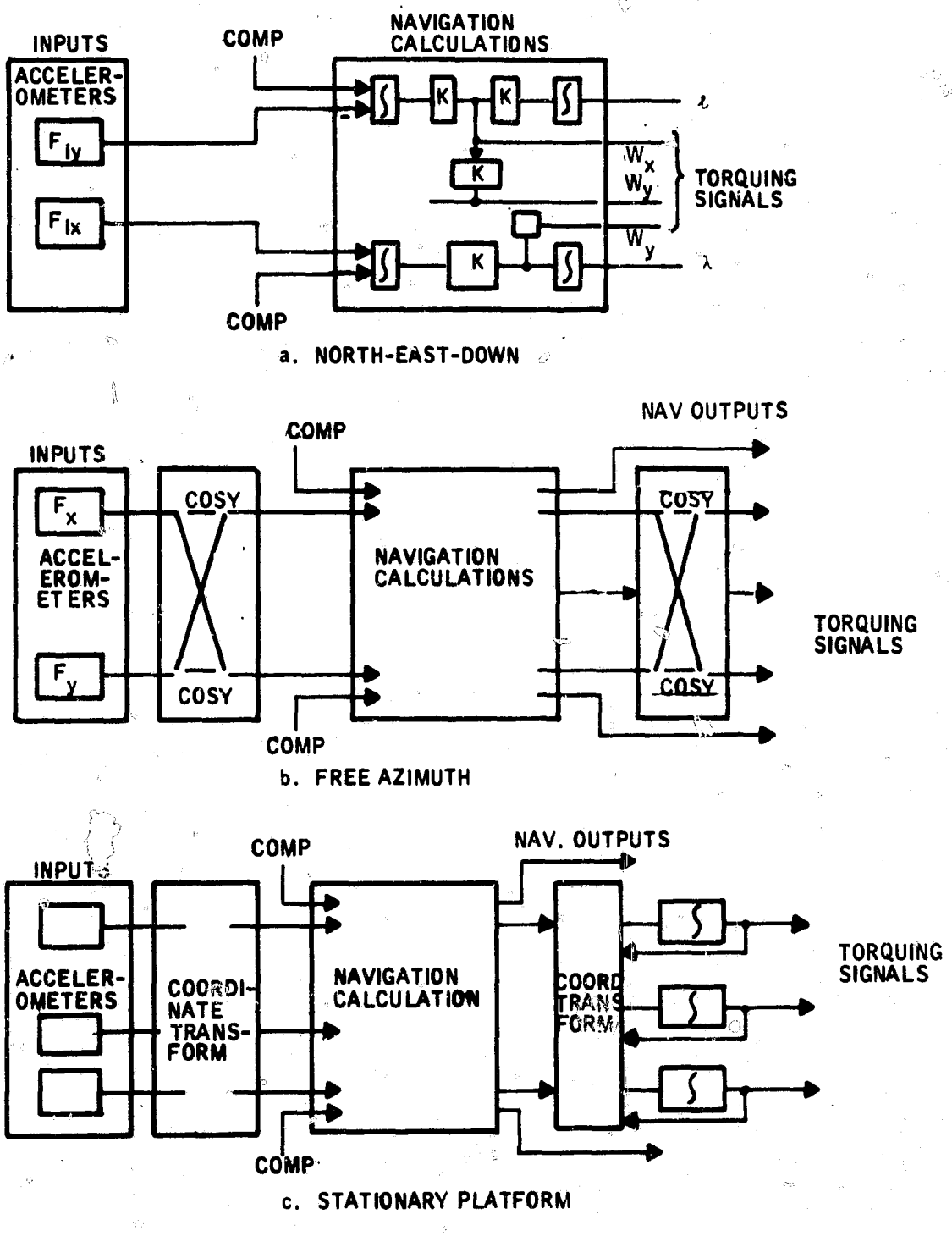


Figure H8. System Functional Block Diagrams

From this discussion, algorithmic variations required by different implementations of inertial navigation will restrict the ease of modularity experienced with LORAN navigation. Notice, however, the computational similarity between the "non-rotating platform" system (Figure H8) and the two described earlier. The bulk of the calculations are the same, but there must be coordinate transformation of the inputs similar to the free azimuth system (Figure H8b). In addition there must be integration at the outputs to obtain the torquing signals required to maintain the system heading.

The range of input and output values used for inertial navigation should be easily handled. Program I/O includes the accelerometer readings and the torquing signals for the various gyros. Latitude/longitude coordinates will probably be required for pilot display. The quantity of I/O and the dynamic ranges required will be a function of the particular implementation used.

Kalman Filter -- Future aircraft navigation problems are expected to involve the use of Kalman filter techniques for refining navigation data obtained via other sources (such as LORAN or inertial methods). Kalman filter techniques are based on the error models of the other navigation systems. These error models take into account discrepencies in computed position among the various techniques and predict a best estimate of the true position. When this expected value of the measurement is compared to values actually measured, the differences can be used to correct the estimate of the aircraft's position and velocity in the next time increment. A sequence of such measurements, separated in time, together with an accurate mathematical model of the system and environment will eventually produce estimates with sufficient precision to permit maneuvers to be made with confidence.

Figure H9 shows a block diagram of a Kalman filter implementation of a complex navigation system. The Kalman filter outputs are fed back to the navigation sensor at various points in the system and are used to form the differences between expected values and actual measured values. Kalman filtering techniques are complex matrix operations involving state variable vectors. The way the sums and differences of various signals are produced is highly dependent on application (Ref. H9).

This description of the Kalman filter characteristics indicates that Kalman filter computation is not a good candidate for modularization of the functional level. Since Kalman techniques are used to improve accuracy and refine values obtained by other methods, the combinations of other system signals into the state variable matrices is a new and different problem with each new flight system. However, once the input values are gathered, the matrix operations required to reduce the data may be fairly standard. The feedback of the Kalman outputs back into the system is again highly dependent on application. Kalman filter processing requirements do necessitate a number of matrix operations. These include:

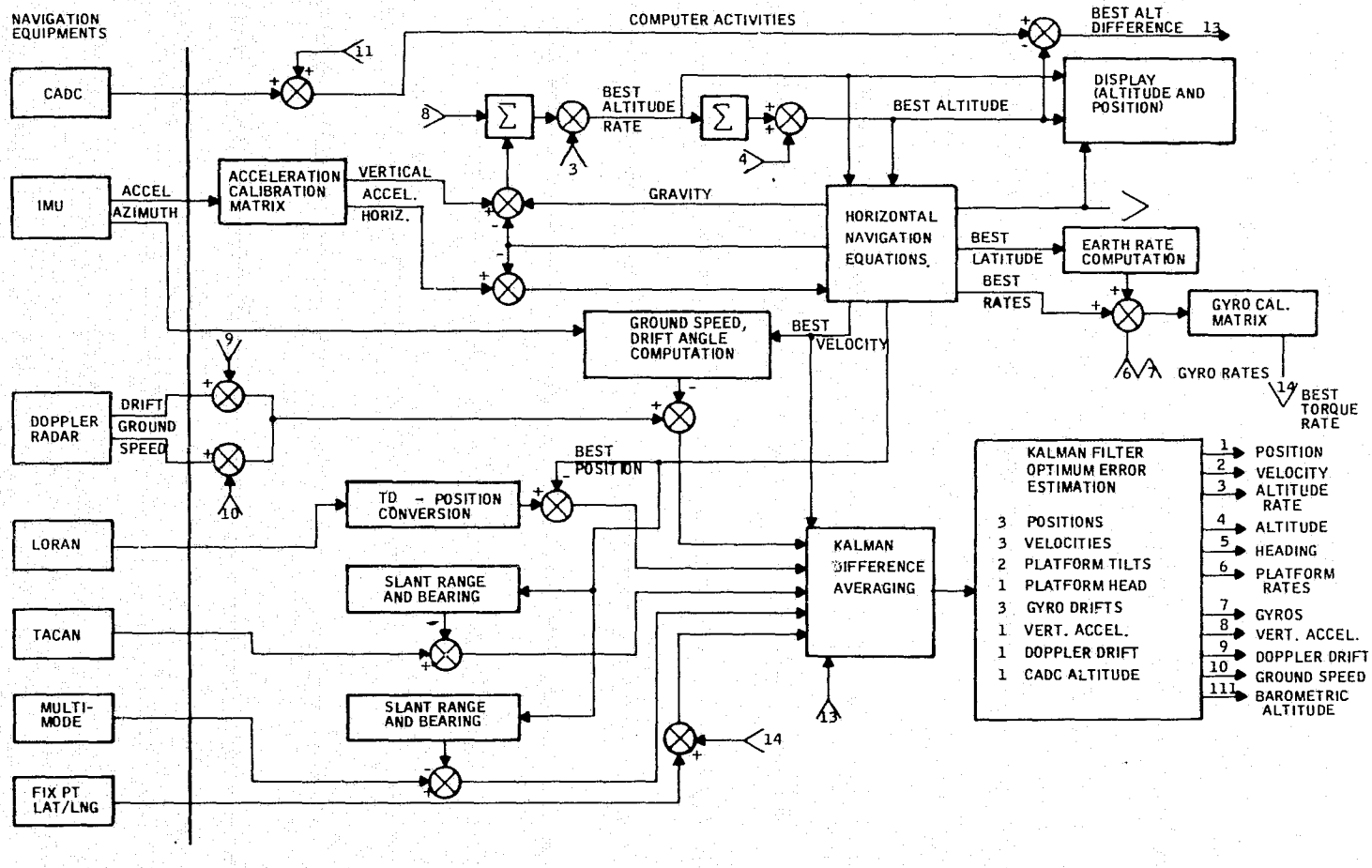


Figure H9. C-5A Kalman Navigation Mechanization

- Matrix addition and subtraction
- Matrix multiplication
- Transpose multiplication
- Matrix inversion
- Scalar multiplication

Any attempt at modularizing the Kalman filter requirements should begin with implementation of these matrix operations. This would form a firm basis for coding the rest of the Kalman algorithm. If this is done, arguments of the matrix processing routines should include starting addresses of matrices involved and their dimension. These matrix operations should then become part of the compile time library which form the lowest level of the subroutine hierarchy as shown in Figure H4.

By looking at Table H4, one can see that the properties of the Kalman filter are such that they become disadvantages to modularization as compared to the advantages for LORAN and inertial navigation.

Digital Flight Control -- The implementation of digital flight control is based on the solution of difference equations in place of analog filters. The number, sequence, order, and data coefficients for the difference equations will vary for each flight control system. Consequently, properly parameterised difference equation subroutines are required. A standard, reusable library module is not feasible. However, an English language and block diagram description of the typical computational sequence and digital filters that are required for the stability augmentation and pilot relief loops is needed.

Air Data -- The calculations required for this function are relatively standard. This junction is a prime candidate for a standard library module. Differences exist depending on (1) whether supersonic flight is possible, (2) output requirements to other subsystems, and (3) error correction terms. These differences are slight and can be handled by allowing for data values unique to each application to be used. Power series expansion subroutines are useful for this function.

REFERENCES

H1. Parnas, D. L., "On the Criteria to Be Used in Decomposing Systems into Modules." CACM 15(12), December 1972, pp. 1053-1058.

H2. Haney, F. M., "Module connection analysis - A tool for scheduling software debugging activities," Proc AFIPS FJCC, 1972.

H3. Liskov, B. H., "A Design Methodology for Reliable Software Systems," Proc AFIPS FJCC, 1972.

H4. Dykstra, E. W., "The Humble Programmer," CACM 15(10), October 1972, pp. 859-866.

H5. Organick, E. I., The Multics System: An Examination of Its Structure. The MIT Press, Cambridge, 1972.

H6. Thurber, K. J., et al., "Operating System/AADC Preliminary Functional Specification," Final Report, Navy Contract N62269-72-C-0051, October 1972.

H7. Study of Information Management Aspects of Integrated Avionics - Volume I, B.H. List, Texas Instruments Inc., Technical Report No. AFAL-TR-69-166, Volume I, July 1969.

H8. Broxmeyer, Charles, Inertial Navigation Systems, McGraw-Hill, New York, 1964.

H9. "Theory and Applications of Kalman Filtering," AGARDograph No. 139, edited by C. T. Leandes, February 1970, AD 704306.

APPENDIX I
LIGHT-EMITTING DIODES

Introduction

In the past decade there has been a rapid improvement in the efficiency and reliability of visible-light-emitting diodes (LED), the major problems limiting the performance of these devices have been mastered, and as a result large-scale manufacturing of diodes suitable for a variety of applications has become possible. $GaAs_{1-x}P_x$ diodes that emit red light are the most widely used, but newer materials, such as GaP for green emission, $In_{1-x}Ga_xP$ for yellow, and GaN for blue are being developed.

Luminescence is the emission of optical radiation ultraviolet, visible, or infrared light; i.e., luminescence is a direct result of the energy released during electronic transition within a material. Luminescence arises from a two-step process in which electrons and holes are generated in concentrations greater than those statistically permitted at thermal equilibrium, and then a significant fraction of these carriers recombine. Radiative recombination occurs when the energy of the recombining holes and electrons generate protons. The recombination process is nearly independent of the source of excess carriers, but is very strongly characteristic of the physical and electrical properties of the material.

Figures of Merit

A measure of performance of a LED must include the relative response of the eye at the wave length of interest. The two important figures of merit for LED's are quantum efficiency and luminence. Quantum efficiency is the ratio of the number of photons produced by the number of electrons passing through the diode. The internal quantum efficiency is calculated in the p-n junction of the LED, while the external quantum efficiency, is calculated in the interior of the diode. The external quantum efficiency is always less than the internal quantum efficiency due to the optical losses in the semiconductor. Average external quantum efficiency values fall between 0.1 and 7.0 percent at room temperature, while internal quantum efficiencies can exceed 50 percent under optimum conditions. Table I1 (ref. I1) gives the quantum efficiency as well as the luminous power efficiency of some recent LED's.

Because LED's emit radiation, they are more visible in dim ambient light and less visible in bright ambient illumination. Experiments at NADC show that LED's are not bright enough for military aircraft cockpit application. Also, the red LED color is not the best for aircraft use as it is a universal danger signal and is in the region of poor eye sensitivity especially for older people (ref I1). However, LED's are evidently being used to some degree in commercial aircraft and are probably only visible under bright ambient light with some shading.

LED's are high-current, low-voltage devices that are compatible with discrete MOS transistors. However, they are not readily driven by MOS IC's, particularly CMOS IC's due to the electrical current requirement. Therefore, large LED displays will not be cost-competitive with other display materials (LCD, electrochromic, or electrophoretic) that can use less costly low-voltage, low-current TFT's or SOS IC's to build the elements of an active matrix.

The fast time response of the LED makes it easy to matrix address or multiplex a display; however, display luminance will be limited unless memory of some sort is provided for each diode. The upper limit of the number of lines that can be multiplexed (or

TABLE I1. – PERFORMANCE CHARACTERISTICS OF DIFFERENT TYPES OF LED's

LED	Color	Peak emission wave length	Quantum efficiency		Luminous power efficiency	
			Research	Commercial performance	Research result	Commercial performance
$GaAs_{0.6}P_{0.4}$	Red	6490	5.0×10^{-3}	2.0×10^{-3}	0.38	0.15
$GaAs_{0.35}P_{0.65}$	Red-orange	6320	4.0×10^{-3}		0.76	
LPE GaP:Zn.O	Red	6925	1.5×10^{-2}	$0.5\text{-}2.0 \times 10^{-2}$	3.0	0.1-0.4
VPE GaP:N (Zn diffusion)	Green	5700	1.0×10^{-3}	$0.3\text{-}0.8 \times 10^{-3}$	0.6	0.1809.48
LPE GaP:N (grown junction)	Green	5700	3.0×10^{-3}	$0.5\text{-}2.0 \times 10^{-4}$	1.8	0.03-0.02
$GaAs_{0.15}P_{0.85}$:N	Yellow	5890	0.8×10	$0.5\text{-}0.8 \times 10^{-3}$	0.36	0.23-0.36
VPE GaP:N ($N > 10^{20}/cm^2$	Yellow	5900	1.0×10		0.45	

addressed) is the maximum permissible current through each diode divided by the average current required for the corresponding desired average brightness (ref. I1).

For line-at-a-time address, the required peak luminance B_R with a desired time-averaged brightness (B) is:

$$B_R = \frac{B \times L}{N} \tag{I1}$$

where

L is the number of display lines, and N is the number of frames that the eye integrates in 0.1 second.

Peak luminance of up to 6800 cd/m^2 (ref. I2) is possible with heat sinking and cooling, but 340-1400 cd/m^2 is more common. Application of LED's is presently limited to small numerical instrument readouts of one line, but alpha-numeric vector-graphic displays of up to 10 cm are technically feasible (ref. I2). Table I2 shows the time-averaged luminance levels for 16, 100, and 500 line displays, all at 30 frames/second TV rates.

TABLE I2. – MULTIPLEX OR MATRIX ADDRESS DISPLAY WITH LUMINANCE AT 30 FRAME/SECOND RATES

Line-at-a-time address, number of lines	Time-averaged luminance, cd/m^2	Required contrast ratio
1-	340-1400	10:1
16	63-261	53:1
100	10-42	333:1
500 (TV formate)	2.-8.4	1700:1

These displays must have a luminance of 680 to 1700 cd/m^2 when the vision is eye adapted to a cloud luminance of 3.4×10^4 cd/m^2. It can be seen that the luminance for displays with over 100 lines will have poor legibility under high ambient illuminance levels unless memory is provided between refresh periods to increase the average brightness. No references have been found in the literature of efforts to develop on-site memory.

Another basic problem with line-at-a-time display address is the steady-state residual luminance of the LED in its off state. The time-averaged contrast is the time-averaged brightness of the display divided by the residual luminance of the LED.

The required contrast ratio (C_R) between the peak luminance and the steady-state residual luminance is the desired time-averaged contrast (C) times the number of lines (L) to be addressed, divided by the number of frames (N) integrated by the eye. Required contrast ratios are included in Table I2 for a time-averaged contrast of 10:1 at 60 frames/second.

$$C_R = \frac{C}{N} L \tag{I2}$$

where

C_R is the required cell contrast, and C is the desired time-averaged contrast, 10:1.

It can be seen that contrast ratios requirements for a large-scale LED matrix would be severe even if the efficiency and luminance could be dramatically increased. Development of on-site memory appears to be the only way to develop matrix LED arrays for aircraft use.

Rise and decay times for LED's are in the 10 to 1000 nsec range with small temperature variation (ref. I1); therefore, temperature sensitivity can be neglected. Also, light output of the LED increases at a rate of approximately 1%/°C between 0° and 50°C, which is not important for most displays.

The cost of LED's are estimated at $0.50 to $2.50 per digit with a system cost of $1.00 to $3.50. Test results show that diodes can operate continuously for 50 000 hours although the light efficiency is low – 0.1 to 0.5 lm/W (ref. I1), compared to CRT's of 100 lm/W (ref. I3). Typical power dissipation for a segment in a calculator display is 0.67 mW with an integral lens (increased brightness but reduced viewing angle) or 3 mW without a lens. A 10 by 10 alpha-numeric display with 256 (nine-segment) characters would require 7 watts for just the display.

Monolithic LED Arrays

Progress is being made in the fabrication of monolithic matrix addressed arrays (ref. I4) of LED's in GaP using selective liquid-phase epitaxy (LPE) for 5 by 7 arrays. These arrays are fabricated by GaP LPE processes on a single wafer with interconnected LED elements. Voltages of the dot matrix elements are compatible with silicon-integrated circuits. This permits logic chips to be hybridized to the substrate of the GaP wafer, thus reducing the requirements for lead fan out and creating a natural interface with data-handling systems.

One of the problems with monolithic displays is optical isolation between adjacent junction areas. Light spreading in GaP material would cause a large part of the semiconductor wafer to light up the outside junction area without optical isolation. It is claimed, however, that efficient monolithic arrays in indirect GaP semiconductor can be produced by taking advantage of the air-isolated integrated circuit technology.

The monolithic display structure is preferable for small diodes with high-packing densities; the single chip approach should be used for large-area displays with low-packing densities limited for 50 elements/inch. However, changes in technology, material, cost, and applications are too rapid at the present time to permit a realistic appraisal of the transition point between the two structures.

Conclusions

Advantages of LED's are long operation life and low-voltage requirements. Disadvantages are low-element density (50 elements/inch) high-power consumption, and low brightness without on-site memory. Present LED displays are probably limited to 10 cm by 10 cm, 256 character alpha-numeric displays with line-at-a-time address. The unknowns are cell isolation ratios, potential increase of efficiency, maximum packing density, and future development of on-site or intrinsic memory.

Appendix I

REFERENCES

I1. Goodman, L. A.: The Relative Merits of LEDs and LCDs, Proceedings of the SID, Vol. 16/1, First Quarter 1975, pp. 8-20.

I2. Reingold, I.: Display Devices; A Perspective on Status and Availability, Proceedings of the SID, Vol. 15/2, Second Quarter, 1924, pp. 63-74.

I3. Van Raalte, J. A.: Matrix TV Displays, System and Circuit Problems, IEEE Transactions on Consumer Electronics, Vol. CE/21, No. 3, August 1975, pp. 213-219.

I4. Kravitz, L. C.; et. al.: Monolithic LED Arrays by Elective Liquid-Phase Epitaxy in GaP. Proceedings of the IEEE, Vol. 61, No. 7, July 1973, p. 891.

APPENDIX J
LIQUID CRYSTALS DISPLAY (LCD)

Introduction

The advantages of liquid crystal passive displays are many – low power, low cost, low voltage, and flexibility in size. The reflective type of liquid crystal display (LCD) requires little power to operate compared to other types of displays such as LED's and low-discharge tubes. Because of the low cost of the liquid crystal material and the simplicity of the display cell structure, the cost of liquid display is very attractive. With operating voltages in the range of 1 to 6 volts, LCD's can be driven off a MOS chip or other low voltage electronics. This is an important cost advantage for a matrix display with many active elements.

Reflective dynamic scattering displays require a specular mirror-back electrode (ref. J1). This creates some viewing problems particularly under conditions of diffuse illumination (flying through clouds), and contrast wash may occur. However, this type of display material will probably not be used for aircraft application as twisted nematic material has much less power consumption with a faster response.

A diffuse reflector can be used with the twisted nematic display because twisted nematic does not induce light scattering. Consequently, an observer can see the voltage-induced change in light transmission without the chance of a reflective glare from the dynamic scattering LCD. The twisted nematic display is also visible under diffuse illumination. It should be noted that display contrast will fall off with viewing angle unless a drive voltage of greater than 20 volts is used. This higher voltage, however, may negate the advantage of driving the LCD with CMOS IC's and of the ease of fabrication of large matrix displays.

Liquid crystal properties are temperature dependent and the prospects of finding new materials with low operating temperatures and wide temperature ranges is not encouraging. The only practical solution is to incorporate a heating element in the liquid crystal display. Manufacturers are currently quoting lifetimes of 50 000 hours as typical for LCD display with ac drive voltages (ref. J2). The disadvantage of dc operation is that the electrochemical process degrades the crystals and shortens their life; the disadvantage of ac drive is that a capacitive coupling may partially turn on adjacent elements during the multiplexing or matrix address. Another disadvantage of some LCD materials is that they have a slow rise time of 10 μsec and long decay time of 100 to 300 μsec. However, on-site memory not only solves the problem of the slow rise time but increases display brightness.

LCD technology is relatively new and costs should continue to decrease as the manufacturers gain more experience. Present costs for a 3-1/2 digit watch display 0.2 inch high is $5.00 in production quantities (ref. J1). System costs for instrument applications are estimated at $0.85 to $2.75 per digit.

Properties

Liquid crystal states. – Liquid crystals differ from ordinary liquids by their retention in the liquid state of certain types of crystalline order (ref. J3). The individual liquid crystal molecules have a somewhat cigar-shaped structure. An ordinary liquid does not have an ordered arrangement of molecules and is therefore referred to as isotropic, i.e., the crystalline properties are the same along all optical axes. Liquid

crystals, however, have a nonrandom, ordered arrangement of the molecules at certain temperatures. These arrangements are called the liquid crystal phases and are referred to as anistropic states (crystalline properties vary along different axes). These aligned liquid crystal molecules can polarize, rotate the plane of polarization, scatter or diffract incident light, and produce the desired display effects.

There are three liquid crystal phases: nematic, cholesteric, and smectic (Figure J1). The molecules of the nematic liquid crystals are arranged with their axis parallel, but not in layer form as in the case of the smectic liquid crystals. Nematic molecules are free to slide past each other but remain essentially parallel within a given region. Cholestric liquid crystals also consist of layers of crystals with parallel axes; however, the axes of the various layers are oriented in different directions within the plane of the layers. These liquid crystal phases exist only over a limited range of temperatures. Below this temperature range, the liquid crystal material may become solid without crystalline properties and above it, the material may cease to be a crystalline liquid and may become an ordinary liquid with isotropic properties.

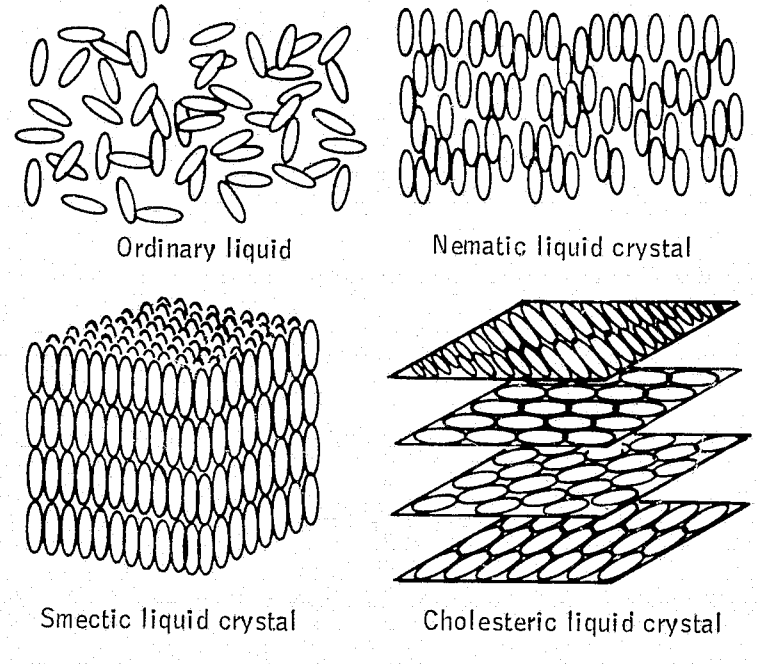


Figure J1. Characteristic Crystal Structures for the Three Liquid Crystal Phases versus an Ordinary Liquid

Liquid crystal materials. – Commercially available liquid crystal materials must be purified to remove all traces of water, oxygen, and unreacted components (ref. J4). Most common nematic liquid crystal materials are either MBDA, EBBA, or a mixture of both. (see Table J1).

Appendix J

TABLE J1. – COMMON LIQUID CRYSTAL SYSTEMS

Type	Temperature	Comment
MBBA (Schiff base)	Freeze point at 20°C, Nematic at 20°C to 41°C	Disadvantages: Hydrolyze in presence of moisture, acids, and bases. Could decompose during cell assembly. Decomposes in ultraviolet light and sunlight. Yellowish color.
EBBA (Schiff base)	Nematic at 38°C to 60°C	
Mixture MBBA and EBBA	Freeze as low as -10°C, Nematic at 10°C to 50°C	Advantages: Both EBBA and MBBA have fairly good scattering properties and easy to synthesize.
Trans-Stilbens C-N band of Schiff replaced by C-C band	Nematic at room temperature	More resistance to hydrolyzation. Colorless.

Liquid crystal effects. – Liquid crystal's most promising effects for display application are summarized in Tables J2 and J3 (ref. J5 and J6). Figures J2 and J3 illustrate the basic modes of operation and configurations.

At the present time the field effect (dielectric forces) liquid crystal displays are the most popular (ref. J1) for matrix panels due to its low power requirements. Of the types of liquid crystals listed in Tables J2 and J3, only dynamic scattering, (dielectric and conduction forces), twisted nematic (field effect), and the unknown Hughes Aircraft proprietary field effect material will be discussed in this report

Surface alignment of liquid crystal molecules. – Only two distinct types of alignment of the molecular nematic liquid crystal axis near the surface of an electrode face are possible (ref. J7): homogeneous and homeotropic. Homogenous alignment of the liquid crystal molecules occurs when the molecules are parallel to the surface. This type of alignment is also referred to as parallel homeotropic orientation or optical normal homeotropic orientation. Homeotropic alignment occurs when the liquid crystal molecules are perpendicular to the surface. Homeotropic alignment is also referred to as basal or perpendicular homeotropic orientation or dipolar molecular axis or director perpendicular to the surface of the electrode.

The on or off states of a liquid crystal phase must represent either uniform optical clarity (one of the homeotropic states) or scattering (isotropic phase). Of all the physical interactions of liquid crystals, the interactions of the molecular with the surface are least understood. However, permanence of molecule orientations near the surface are necessary for long life time of the cell.

Perpendicular orientation of the molecule to the surface is obtained with high purity materials and an exceptionally clean substrate. The surface is considered to be indifferent to the liquid, and minimum interaction occurs when the molecules are perpendicular to the surface. The presence of an interphasal material (on the surface) or surfactant acts as an orientation layer to produce the parallel homeotropic orientation; the surface active agent must interact specifically with the surface and the nematic liquid. These

TABLE J2. — LIQUID CRYSTAL CHARACTERISTICS

E/O Effect	Field Effect and Transmittance, Maximum	Cell Structure	Mechanism		Electronic Drive	Response Time
			Off State	On State		
Dynamic Scattering	No 100%	Negative dielectric anistrophy between electrodes.	Transparent. Cell is dark if light source is not visible.	Scattering by turbulence induced by charge flow. Milky white appearance.	0.10 $\mu A/cm^2$ -20V(17)	1 μsec at 250°C and 10^5 V/cm. Normal 10 μsec rise time, 100-300 msec decay time
Parallel Homeotropic	Yes 25%	Positive dielectric anisotrophic. Liquid crystal between parallel homeotropic electrodes and crossed polarizers.	Homeogenous orientation. Plane polarized light is converted to elliptical by bulk of parallel homeotropic liquid crystal. Analyzer passes plane polarized light to observer.	Perpendicular homeotropic orientation produces quasi-isotropic scattering. Depolarized light through analyzer produces dark cell.		
Guest-Host Interaction	Yes 50%	Positive dielectric anistrophic. Liquid crystal with 1% concentration dichroic dye between parallel homeotropic electrodes and parallel polarizers.	Dye molecule axis aligns with nematic liquid crystal axis in parallel homeotropic orientation. E light vector parallel to dye molecule.	Field converts nematic and dye orientation to perpendicular homeotropic, perpendicular to E light vector. Color absorption is tuned by voltage.		
Voltage-Controlled optical activity or twisted nematic	Yes 50%	Positive dielectric anistropic. Liquid crystal between crossed rubbed (parallel homeotropic) electrodes and crossed or parallel polarizers.	Parallel homeotropic nematic molecules turn 90° between electrodes. Polarized light will follow molecules and twist 90°. Cell is bright with crossed polarizers and dark with parallel polarizers.	Liquid crystal becomes isotropic and scatters light. Nematic molecules rotate to assume perpendicular homeotropic orientation away from electrodes. Cell is bright with crossed polarizers and dark with parallel polarizers.	• 3V - 5V threshold • 7-8V maximum • 1μ amp/cm^2 current • Good match to complementary MOS • 10 V Pulses	• 20 msec rise time and 50 msec decay time with 5V dc drive. • At 20°C and 8V, 250 msec on and 130 msec off.
Fast turn off or Chevron mode	No	Same as Dynamic Scattering	Transparent	Turbulent scattering induced by 600-Hz voltage. Milky white appearance.	Same as Dynamic Scattering	Turn off 5 msec
Deformation of Vertically Aligned Phases (DAP Effect) or tuned birefringence	Yes	Negative dielectric anisotropic. Liquid crystal between special perpendicular homeotropic-textured surface electrodes and crossed polarizers.	Molecules are in perpendicular homeotropic alignment. Polarized light is passed by cell uneffected and absorbed by analyzer.	Molecules are in parallel homeotropic orientation and field-induced birefringence alters polarization to elliptical or 90° rotation. Some or all of the light is passed by analyzer.	• 4 - 6V or 7 - 8V • 1 - 5 kHz • 5V threshold • 1 μamp/cm^2	

TABLE J3. — LIQUID CRYSTAL CHARACTERISTICS

E/O Effect	Grey Scale Resolution	Cost	Viewing Angle	Contrast	Life Time	Comment
Dynamic Scattering	10 shades above 100 V 25 lines/in.	Lowest	Highest	25 : 1 to 50: 1	Low with dc	• Liquid crystal deteriorates with time due to dc current flow • Subject to cross talk with ac drive • Polarizers are not needed • Used in 90% of liquid crystal displays
Parallel Homeotropic					High	
Guest-Host Interaction						Color display. Density varies upon field and type of dye determines color.
Voltage-Controlled Optical Activity or Twisted Nematic	20 lines/in.	Highest	25 : 1 contrast decreases to 4 : 1 at 40° for reflective and 85° for transmissive display. Contract decreases from 40 : 1 to 30 : 1 at 30° off from normal.	40 : 1 10 : 1 at 25V 5 : 1 at 10V	High	• Prime candidate for matrix displays with thin-film vacuum deposition transistor technology • 120 x 120 matrix cells have been fabricated with TFT on-site memory
Fast Turn-off or Chevron Mode				Not as good as Dynamic Scattering		Oscillating domains do not scatter light as strongly as intense turbulence
Deformation of Vertically Aligned Phases (DAP Effect) or Tuned Birefringence		Higher	Lowest	1000 : 1	High	• Matrix addressed cells with 100 x 100 elements have been built without cross talk. • Generation of color hues is possible by birefringent effect. • High purity material. • Low cross talk.

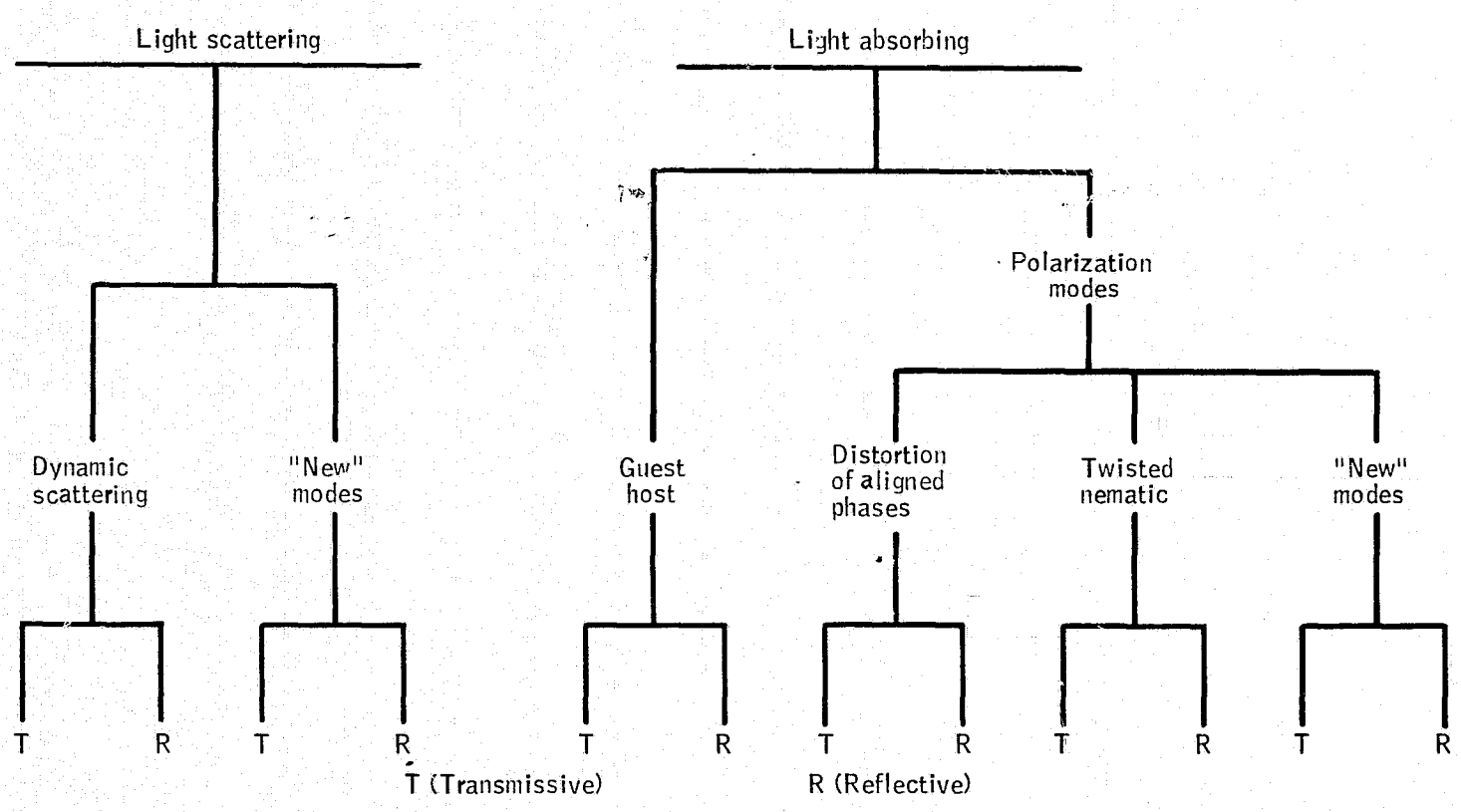


Figure J2. The Two Basic Types of Liquid Crystal Displays and Their Basic Modes of Operation

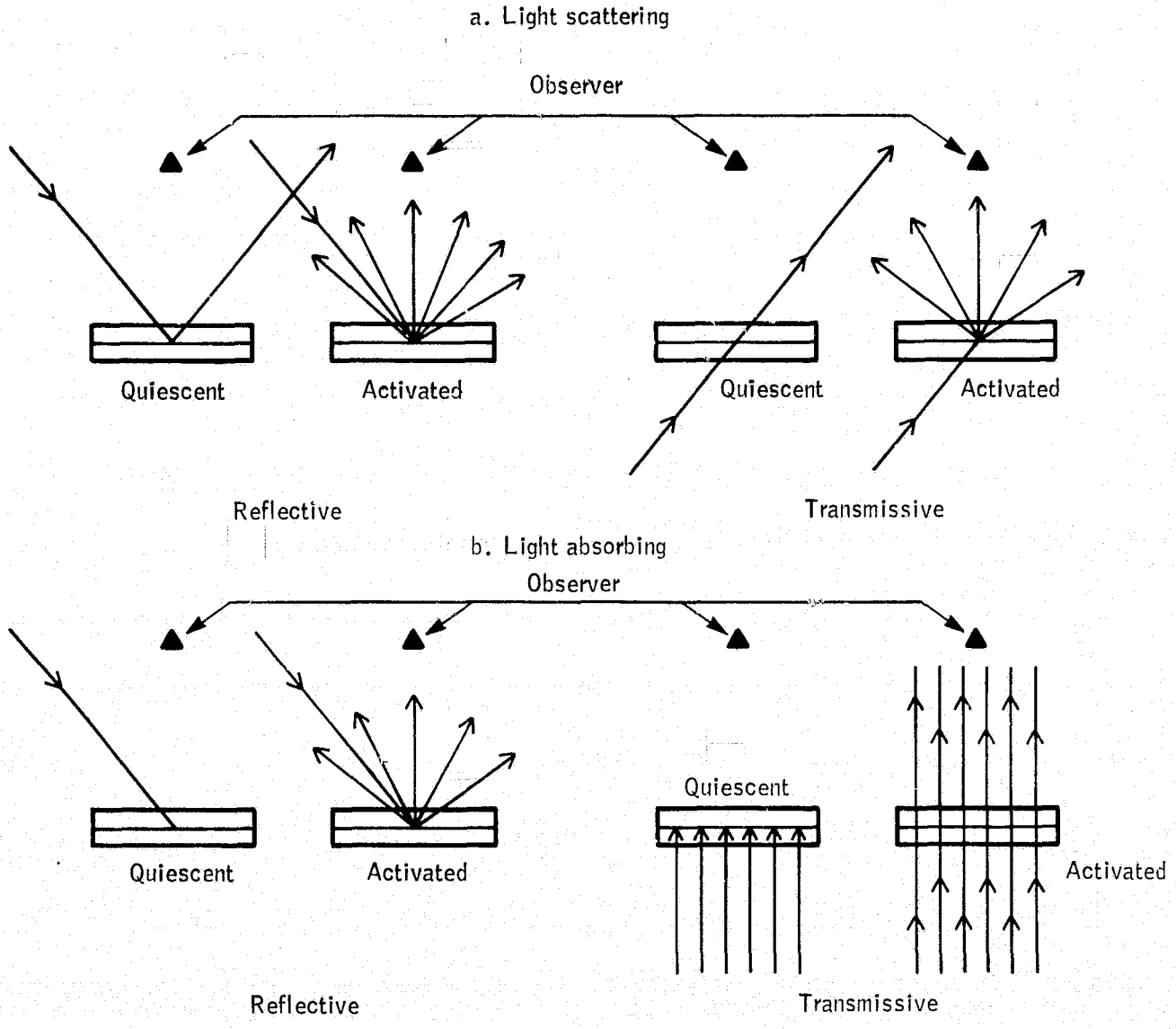


Figure J3. Light Paths for the Two Basic Liquid Crystal Displays

interfaces have been generated by a variety of means including chemical cleaning or etching, rubbing, mechanical surface scribing or deformation, or the deposit of organic or inorganic thin films (ref. J7).

Positive and negative dielectric anisotropy. — The existence of electric dipoles in the crystalline molecular structure is necessary if electro-optic effects are going to occur. The direction of the electric dipole moment determines whether a crystal molecule will have a positive or negative dielectric anisotropy, a property that controls the electro-optic behavior of the crystalline dipolar molecule. For positive anisotropy, the component of the electric dipole moment parallel to the molecular axis is greater than that perpendicular to it. For negative anisotropy, the component of the dipole moment perpendicular to the molecular axes is greater. Thus, an applied electric field will tend to align the positive anisotropy molecular (optical) axis to be parallel with the field. However, the molecular axis of a negative anisotropy crystal will tend to rotate perpendicularly to the applied field. These alignments, produced by applied electric fields, are necessary to produce desired display effects.

Types of Liquid Crystals

Dynamic Scattering. — Dynamic scattering of liquid crystals occurs only with nematic liquid crystals that have a negative dielectric anisotropy. Dynamic scattering cells consist of a few drops of nematic liquid crystals between two parallel glass plates with transparent electric conductors. Reflective displays use a highly reflecting rear electrode; however, if the display is to control transmitted light, then both the rear and front electrodes should be transparent.

In the unactivated state, the liquid crystal layer is transparent as the dipolar molecular axes of the liquid crystal molecules align themselves in the same direction. If the light source is not visible in the display, all of the inactivated areas will appear black. Whenever an electric field is applied, the molecules attempt to align themselves perpendicular to the field. However, due to impurities in the mixture, charged states exist, and the fluid begins to flow. This flowing produces the hydrodynamic effect that disrupts the alignment of the molecules and in turn causes a scattering of the light because of spatial variation in the index of refraction. This scattering of light is mostly in the forward direction.

One disadvantage of the dynamic scattering reflective cell is that a mirror-like surface is required and if a bright source of light is seen as a reflection, the display can be washed out. Another disadvantage of any dynamic scattering display is the necessity of the current flow and power to create the turbulence and scattering. The threshold voltages for dynamic scattering is about 5 to 8 volts with minimal contrast ratios of 5:1. However, the contrast ratio improves with voltage to 50:1 until the saturation point is reached (Table J3).

Twisted nematic. — The twisted nematic liquid crystal effect is most promising material for the matrix display application and is available commercially (ref. J8). The twisted nematic cell structure is achieved by rubbing the front and back electrodes (parallel homeotropic texture) at right angles to one another using a liquid crystal with positive anisotropy. If polarized white light entering one face of the cell (Figure J4) is parallel to the rubbing direction, it will follow the twist around, emerging parallel to the rubbing direction to the second face. Thus, if the cell is placed between crossed polarizers, the cell area is bright; if it is placed between parallel polarizers, the cell area is dark. When the field is applied above its threshold level (Figure J5), the liquid crystal bipolar molecules will rotate from the parallel to the perpendicular homeotropic orientation. Therefore, the nematic liquid crystal molecules are rearranged such that their orientation no longer has a continuous twist of 90 degrees. Hence, light now passes

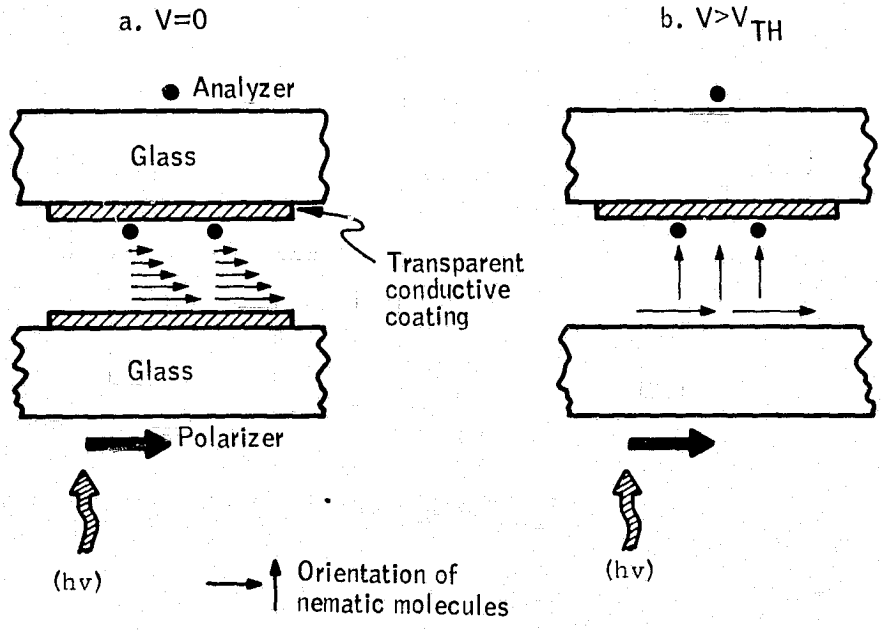


Figure J4. Side View of Twisted Nematic Effect for (a) V = 0 and (b) V > V_{TH}

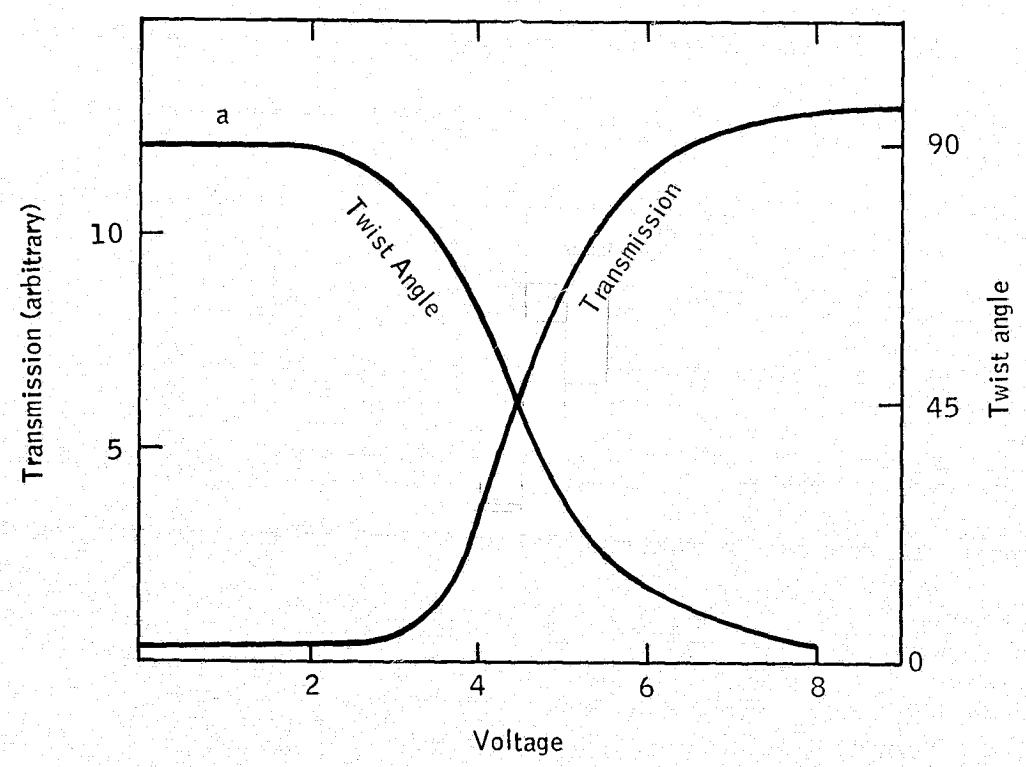


Figure J5. Curve (a) Shows Rotation Angle of Linearly Polarized Light versus Voltage for a Nematic Liquid Crystal at Room Temperature and 1 kHz; Curve (b) Depicts the Transmission versus Voltage Response with Parallel Polarizers. The light source and detector are colinear and pointing normal to the cell.

through the nematic liquid crystals without rotating the vibration plane of polarization. The cell becomes dark (near zero transmission) between crossed polarizers and becomes bright when the cell is placed between the parallel polarizers.

Hughes Aircraft liquid crystal matrix display development. — Hughes Aircraft is presently developing at least two LCD's. One is a multimode alpha-numeric display being developed for NADC, and the other is a pictorial display with a TV-compatible matrix one-line-at-a-time address. Hughes Aircraft decided several years ago that LCD was the best approach for the high ambient light environment of an aircraft cockpit as neither LED's or gas plasma displays were thought to have sufficient brightness.

The multimode alpha-numeric display has 30 to 40 lines of information with 40 to 60 characters per line. Each character is one quarter inch high with 14 segments and one dot in a starburst pattern. The display provides a readout of weapons, fuel, and engine status and is intended to replace the warning lights. This type of display has been checked out under high ambient flood lighting, and both NADC and Hughes Aircraft agree that the display visibility is satisfactory. The segments of all characters are wired in parallel, and individual characters are lit up by encoded digital words. Each character has a logic circuit to decode the input binary data and memory latch such that the display character stays on between refresh periods.

The TV-compatible pictorial matrix display uses a field effect proprietary liquid crystal material. It uses silicon on sapphire [SOS chip as a substrate with MOSFET (transistors)] with ion implementation to form the source and drain region (ref. J9). The matrix array has a transistor and capacitor at each X-Y bus bar intersection to latch up the memory after the display element is addressed. The time constant of the on-site capacitor and the liquid crystal resistance is longer than the 1/30 second frame rate (refresh rate) of the display. Consequently, no power is required to keep the display element turned on, i.e., field effect LCD consumption is less than 20 watts at 5 to 10 volts.

The yield of the present one-by-one inch display wafers is 2 or 3 defects out of 10,000 elements (capacitors or transistors). The liquid crystal thickness is one-quarter to one-half mils to obtain the required time constant. Present cell resolution is 100 elements per inch, and plans are for resolution of 256 elements per inch next year. Hughes is presently assembling four, one-inch elements into a quad assembly to get a 200 by 200 line resolution. Later this year plans are to go to a three-inch wafer for a larger display.

A potential problem is the cross talk between the contiguous edges of the wafers forming the quad display. The two outside edges of each quad contain the matrix address electronics. Also, the usable display temperature range is limited to 40°C. The display is heated to permit operation at low temperature; however, very little power is required to heat the thin display.

Predictions at Hughes Aircraft are that the X-Y address matrix LCD's will be used in aircraft for radar presentations and proximity warning devices. They expect that the prices will be compatible with TV displays in 5 to 7 years.

The LCD is front (wedge) lighted as common with aircraft instrumentation. The present display contrast is 15 to 1 and it decreases when off from normal viewing incidence. Contrast improvement of from 30 to 1 is expected with new techniques to increase the reflectivity of the surface of the liquid crystal material in its off state.

Appendix J

REFERENCES

J1. Goodman, L.A.: The Relative Merits of LEDs and LCDs. Proceedings of the SID, Vol. CE-21, No. 3, August 1975, pp. 213-219.

J2. Anon: Readouts are Plugable. Electronics, September 4, 1975, pp. 145.

J3. Gurtler, R. W.: Liquid Crystal Displays. IEEE Spectrum, November 1972, pp. 24-31.

J4. Anon: Liquid Crystal Displays Bring Flat TV Closer. Electronics, March 29, 1971, pp. 106-107.

J5. Castelano, J. A. : How Heat is Off Liquid Crystals Can Show Their Colors Everywhere. Electronics, July 5, 1970, pp. 64-70.

J6. Tults, J.: A Facsimile Printer Utilizing an Array of Liquid Crystal Cells. Proceedings of the SID, Vol. 12, 1971, pp. 199-203.

J7. Kahn, F. J.: Surface-Produced Alignment of Liquid Crystal. Proceedings of IEEE, Vol. 16, No. 7, July 1973.

J8. Williamson, Dr. J. H.: Engineering Liquid Crystal Display Devices. Electronic Engineering, June 1975, pp. 46-48.

J9. Brody, T. P.: Large Scale Integration for Display Screens, IEEE Transactions on Consumer Electronics, Vol. CE-21, No. 3, August 1975, pp. 260-287.

APPENDIX K
ELECTROCHROMIC DISPLAYS (ECD)

Electrochromic displays make use of color changes on the surface of a chemical electrode. They have the potential for the most contrast and the least contrast angle sensitivity of any passive display (except possibly electrophoretics). However, this material is in an early stage of development and much more research is needed before it can be used for displays.

One report (ref. K1) claims that the ECD's advantage of high contrast is offset by short life and slow response. The report also states that although the drive voltage is low, each write and erase operation requires the passage of a charge through the display material. Thus the power requirement is high at video rates as it varies with the writing rate. However, a more technical report (ref. K2) states that the threshold voltage is low, on the order of 1 volt. This report differentiates between electrochromic (EC) and electrochemichromic (ECC) systems. The details of the various electrical and electro-optical properties, and chemical reactions are beyond the scope of this report and the reader is referred to the literature.

It is important, however, that a reference is made to an ECC system that has 10:1 contrast and an electro-optical response time of 1 to 10 milliseconds. This is suitable for matrix line-at-a-time address with on-site memory. Further reference is made to a potential intrinsic memory or persistence which would then require low power. However, there are crosstalk leakage problems, and this limits matrix size to about 7 lines in the reflection mode unless integrated TFT circuits are used. This application of TFT has not yet been explored, and further research is needed to determine its suitability for multiplex or matrix address displays.

REFERENCES

K1. Sobel, A.: Summary: New Techniques in Video Displays, IEEE Transactions on Consumer Electronics, Vol. CE-21, No. 3, August 1975.

K2. Change, I. F.: Performance Characteristics of Electro-Chromic Displays, Proceedings of the SID, Vol. 16/3, Third Quarter 1975, pp. 168-177.

APPENDIX L
ELECTROPHORETIC

Introduction

An EPID panel uses the electrophoretic migration of charged pigment articles in a suspension. The suspension composed of pigment particles is sandwiched between a pair of transparent electrodes with a layer of thickness of 25 to 100 micrometers. An applied dc field moves the particles toward one or the other of the electrodes, depending on the polarity of the charged particles. The reflective color of the suspension layer viewed through the transparent electrode changes as a result of the migration of the particles. Take as an example white pigment particles that are positively charged in a black suspending liquid. As the white pigment particles move toward the charged transparent electrode, that portion of the panel becomes white in reflective color. On the other hand, when the other transparent electrodes are charged with an opposite polarity, the white pigment particles move to the opposite electrode and are hidden behind the black suspending liquid.

Many color combinations are possible, and the EPID panel has a memory function. This is the result of pigment particles remaining on the surface after removal of the applied voltage due to the Van de Walls theory of attractive forces between pigment particles and electrode. This memory function serves to simplify the driving circuits and reduce the power consumption.

Applications and Development

One panel (ref. L1) constructed by Matsushita Electrical Industrial has a maximum contrast ratio of 20 to 1 with rise and fall times of 20 to 10 milliseconds. Voltages required to achieve these contrast ratios depend on the type and thickness of suspension level; but for contrast levels of 20 to 1, voltages of approximately 50 volts are required at a frequency of one thousand cycles. This company has developed an experimental digital clock using a black and white panel with a contrast ratio 40 to 1 at 75-volt pulse operation. The electrical power dissipation is about 200 $\mu w/cm^2$ at a field density of 100 V/100 μm. This referenced article does not indicate potential dot density for a matrix display; however, due to the slow rise time and decay time, on-site memory is required for a matrix display.

Another suspension display material (ref. L2) was produced by Marks Polarized Corporation and is referred to as Herapathite dipole suspension. These cells are similar to the type already described. An ac voltage at a minimum frequency of 50 Hz is required, but 3 kHz is preferred. The operating voltage is typically 35 volts peak to peak per mil of fluid path length. The cell is used in the reflective mode, and less than 1 percent of the ambient light is reflected through the suspension fluid in the offstate; about 40 percent of the light is reflected back to the viewer from the energized areas. This yields a contrast ratio of 40 to 1.

Response time of this material was tested (ref. L3) and found to be about 50 μsec 50 volts peak to peak. Significant settling was also observed in the material (VARAD200 and 300) over a 2.5 day period, and some discoloration was observed. Further development is needed to perfect the material.

A recent electrophoretic display (ref. L4) developed by Philips Laboratory consists of colloidal size particles pigment (Figure L1). The particles are T_i, O_2 and have better light-scattering properties than any other display material tested and have

excellent contrast over very wide ranges of viewing angles (see Figures L2 and L3). It is the only display that approaches the scattering properties of paper and ink. However, a d c pulse of 30 volts is required, and switching time is 100 msec. Power consumption is less than 20 $\mu V/cm^2$ of area.

Principal degradation of this display is agglomeration of the particles with nonuniform covering of electrode surface. The maximum lifetime is now only 10^6 cyles, but progress is being made. If a nonlinear device is added to the cell to improve the threshold, then small line-at-a-time address matrix displays are possible. On-site memory can improve the contrast such that large-area matrix displays are possible.

Conclusions

The passive electrophoretic display materials require very little power ($\mu W/cm2$) and have the very real advantage of excellent contrast over wide viewing angle. However, these displays have problems with the suspended particles sticking to the electrode surface, deterioration of the charging characteristics of the particles, irreversible electrochemical reation of the suspension with the electrodes,electrolytic decomposition, precipitation, slow response time, and drive requirements of 25 to 30 volts.

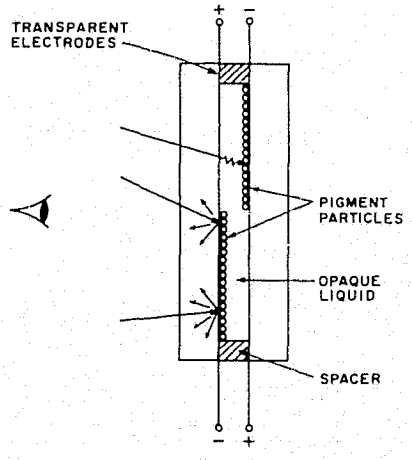


Figure L1. Schematic of EPID Cell

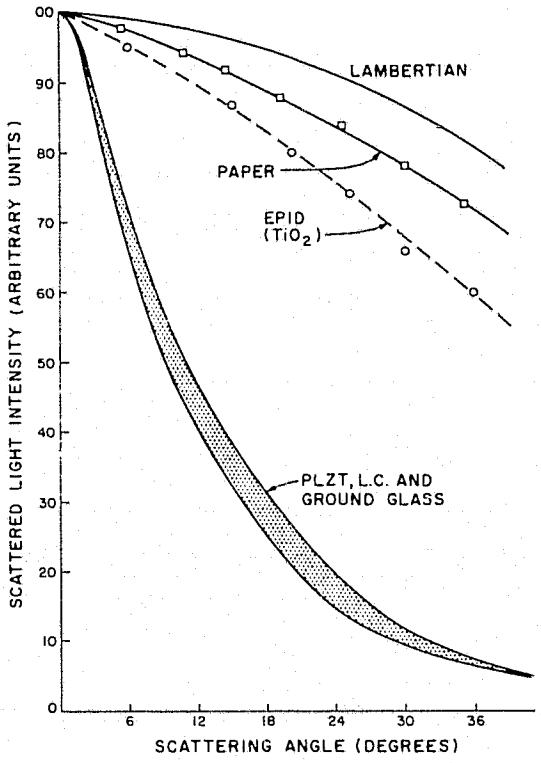


Figure L2. Scattering Profiles Comparing the EPID Device with Several Other Passive Display Media

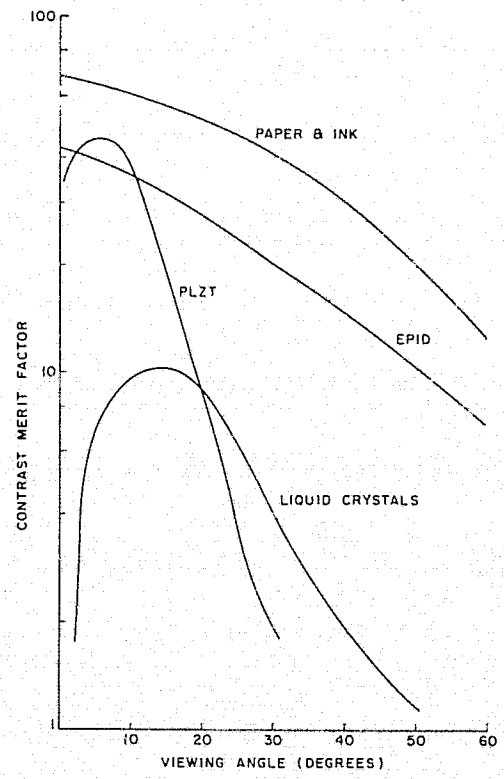


Figure L3. The Contrast Merit Factor for the EPID Device and Several Other Passive Display Media

Appendix L

REFERENCES

L1. Ota, J., et al; Electrophoretic Image Display (EPID) Panel, Osaka, Japan, Proceedings of the IEEE, July 1973.

L2. Marks, A. M.; Electroptical Characteristics of Dipole Suspensions, Applied Optics, Volume 8, No. 7, July 1969.

L3. Bonne, V.; Electro-optical Properties of Herapathete Suspension II, Memo to J. C. Myers, June 1974.

L4. Dalisa, A. L.; Recent Progress in Electrophoretic Displays, 1974 SID International Symposium, May 1974, pp. 88-89.

APPENDIX M
PLZT SYSTEMS

Introduction

Generally, the PLZT electro-optic material may be classified into one of three passive operational groups: memory, linear, or quadratic. Memory materials exhibit the conventional ferro-electric P versus E hysteresis loop in which a continuous range of remanent polarization (polarization state with no external electric field) can be achieved. The grains or crystallites of these polycrystalline ceramic materials are randomly orientated, but the application of polarizing electric fields causes ferro-electric domain growth and orientation in specifically allowed directions. Consequently, polarization occurs, and a birefringence condition results, the magnitude of which is a function of the remanent polarization. When thermally depoled, PLZT ceramic materials exhibit zero effective birefringence. However, in the electrically depoled state, a net birefringence may exist.

Memory materials with high coercive fields (typically those with tetragonal crystal structure) exhibit a linear electro-optic effect (see Figure M1). However at zero-field conditions, the linear materials possess a net birefringence that must be accounted for in operation.

The quadratic PLZT compositions exhibit a very narrow hysteresis loop (Figure M1). Below saturation, the electric birefringence of these "slim-loop" materials is a quadratic (Kerr effect) function of the electric field. At zero field, cubic crystal symmetry exists, and the materials are optically isotropic, exhibiting zero birefringence.

Extensive theoretical and experimental work has been done on optically active piezoelectric material (PLZT) at Honeywell and elsewhere. At Honeywell, basic parameters such as birefringence, switching speed, and transmission characteristics have been studied for several materials manufactured in-house. Cockpit displays have been built at Honeywell, ranging from two-color landing aid displays (PAFAM) to bar-type displays, and reticle-type displays.

Application of PLZT for a display suggests that it be used in a memory mode to have as high an optical transmission duty cycle as possible. The memory mode can be obtained from memory-type PLZT materials or from on-site electrical memory. On-site electrical memory requires a fairly complex, though feasible, deposition of microcircuit thin-film transistors on the PLZT substrate. A system with a resolution of a few hundred lines per inch and a transmission factor of 1/16 appears possible, but intrinsic on-site memory is required for adequate contrast. At present, the optical transmission qualities of nonmemory PLZT are better than the memory types, principally in transparency and reduced scattering characteristics.

Required drive voltages for PLZT on the order of 30 to 50 volt tend to restrict the application of microelectronics. Alternately, a dc bias of 30 volts plus a reduce drive of ±3 volts can be applied to reduce the drive voltage amplitude (ref. M1).

Improved PLZT materials are available, but more effort is needed to further reduce scattering and to increase transparency.

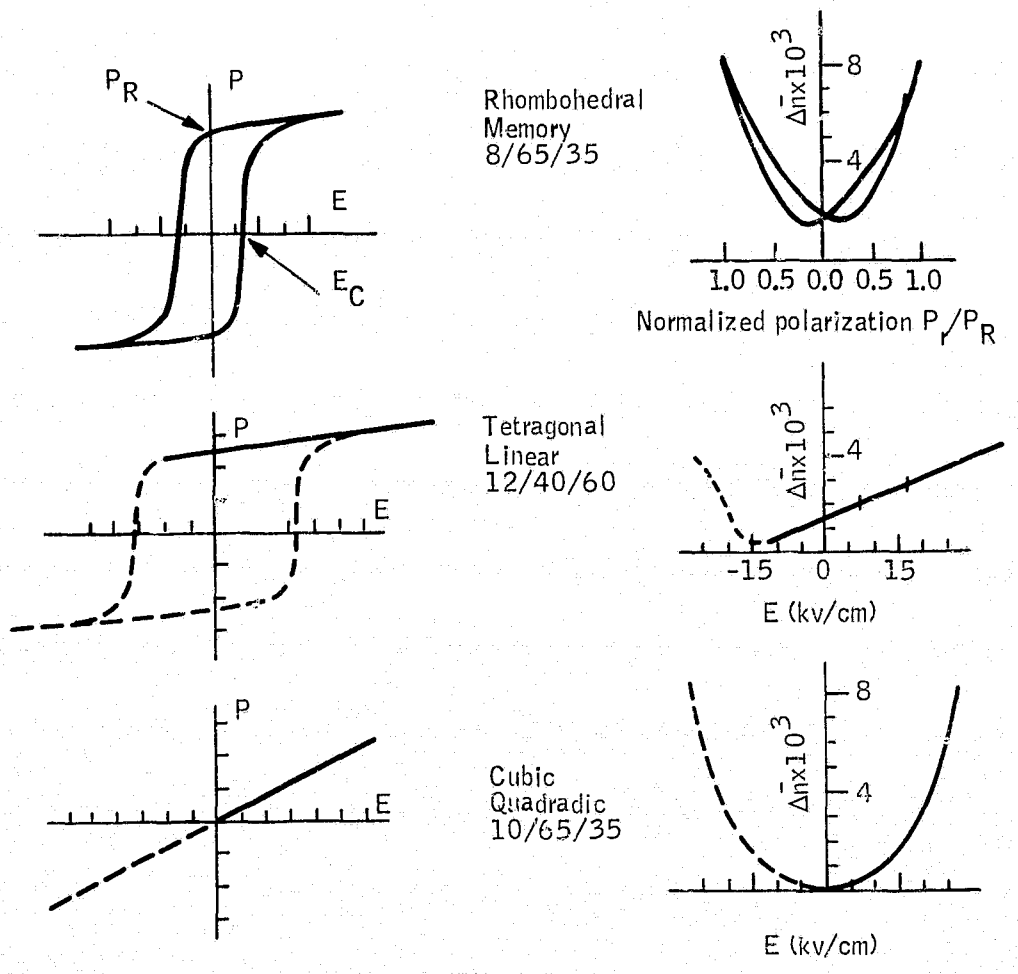


Figure M1. Hysteresis Loops and Electro-optic Birefringence Characteristics for Memory, Linear, and Quadratic Materials in the PLZT System; P scale = 10 $\mu c/cm^2$/div., E Scale = 5 KV/cm/div.

Appendix M
Modes of Operation

The transparent ceramic PLZT (from the constituents elements (Pb, La, Zr, Ti) exhibits light transmission properties that are voltage dependent. Without lanthinum, the materials is not transparent, but exhibits conventional piezoelectric characteristics of dimensional change with applied voltage (PZT). To use the material in a solid-state display, it is necessary to use the optical properties of induced birefringence and scattering.

The active optical effects of the scattering mode are achieved by applying an electric field transverse to the direction of the transmitted light. The birefringent mode is implemented by operating the material such that an applied voltage will vary the polarization of the transmitted light. The effect can be observed through an analyzer sheet as a change in intensity. In the scattering mode, transmitted light is diverted from its normal propagation direction and is scattered into a larger solid angle by the application of voltage to the material. Within these two modes of operation, the material can be classified according to its memory properties. The PLZT wafer may have memory or no memory properties in either the birefringent or scattering modes. The birefringent and scattering modes of operation can be applied to either transmissive or reflective passive displays.

Drive Requirements

Electrical requirements for driving PLZT elements are determined mainly by the required voltage gradient and the desired switching speed. The material impedance is largely capacitive. The capacitance of a bar 7 mils wide by 1 inch long and 5 mils thick is 12 pF for a dielectric constant of 4000. Dissipation factors (tan δ) vary from 1 percent for memory materials (EO 70, -80, -120) to 7 percent for new memory materials (EO-90).

Typically 10 V/mil are required to switch the materials at very low frequencies. At switching times of a few microseconds, the required voltage gradient increases by a factor of four (ref. M2). Based on data for EO-90 from reference M2, the required drive voltage for a given switching time is shown in Figure M2. The pacing parameter is the thickness of the PLZT element as this determines the voltage gradient for a given applied voltage. It can be seen from the figure that for a switching time of 40 μsec, a 2 by 2 mil element requires about 40-volts drive; on the other hand, a 5 by 5 mil element could not be switched much faster than 600 μsec for the same voltage.

Erasure or switch-off can be accomplished in one of two ways: a reverse voltage can be applied to restore the birefringence to its initial value or the material can be heated to its curie point temperature to restore it to the original state. Heating or thermal erase has been used on bar-type aircraft cockpit displays made at Honeywell. The response time for thermal erase for such configurations is approximately 1/2 second, and it could conceivably be reduced to 0.1 second for smaller elements. Application of the thermal erase technique is limited as the material must be cooled back to its operating temperature after being heated to the curie point. In a high-density, multi-element matrix, the heat generated in localized elements may be more difficult to remove and may cause some "smearing" in viewing quality. One method of thermal erasing is to apply an ac field of approximately 100 Hz and to traverse the hysteresis loop a few hundred times.

A voltage-pulse erase may be the only high-speed method of switching "off" in memory materials having a more "square"hysteresis loop. However, there is the limitation that temperature effects will alter the geometry of the hysteresis loop such that there could be a drift or migration of the switching values of birefringence. Such changes could introduce a reduction of contrast. A compensation for temperature effects is to provide thermal stabilization in the matrix by operating it at a controlled temperature above the ambient.

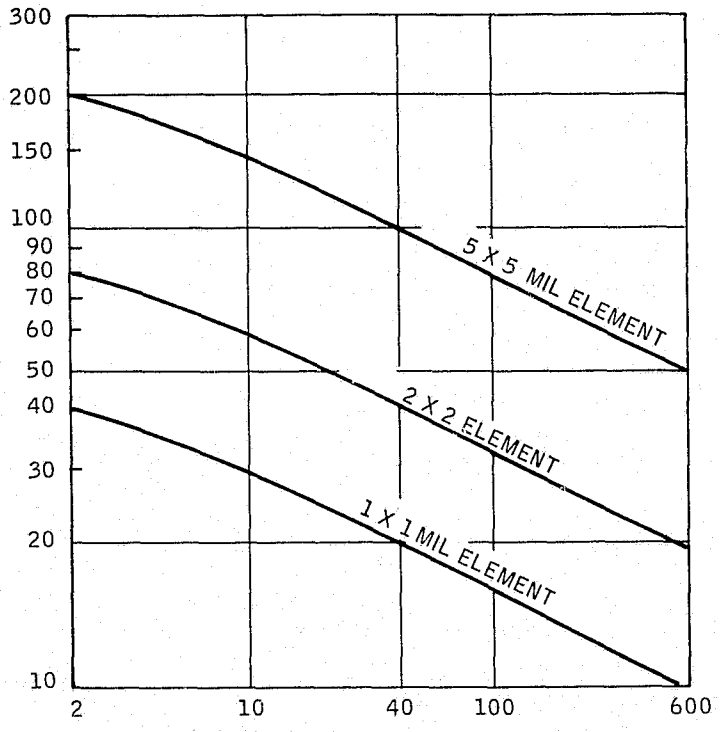


Figure M2. Required PLZT Driving Voltage as a Function of Switching Time for Element Sizes of 1x1, 2x2 and 5x5 x 10^{-3} Inches

Electrode Configurations

A number of approaches towards electrode configuration have been made to provide the transverse field required for birefringent action in PLZT. Surface electrodes have been used, but their inefficiency in establishing a uniform transverse field has led to a slotting and applying electrodes to the sides of the slots. Further, to provide for x, y addressable elements in the matrix that are compatible with memory or nonmemory materials, methods have been worked out using slots on both the front and back surface of the PLZT wafer.

The double-slotted wafers could have a 1 mil wide by 8 mil deep slots, in which the front set of parallel slots are at right angles to the rear set. In addition, the depths of the front and back slots overlap to allow for bringing through deposited "finger" electrodes, as well as bar electrodes. A model of this grid is shown in Figure M3. This approach to a slotted wafer results in a rather fragile and costly structure, but it is possible by proper mounting to use it in airborne applications.

PLZT Scattering Characteristics

A study (ref. M3) has been recently made of the scattering characteristics of several display materials including PLZT. Scattering predictions were based on a convolution-scattering model. The predictions were in agreement with experimental data of PLZT material.

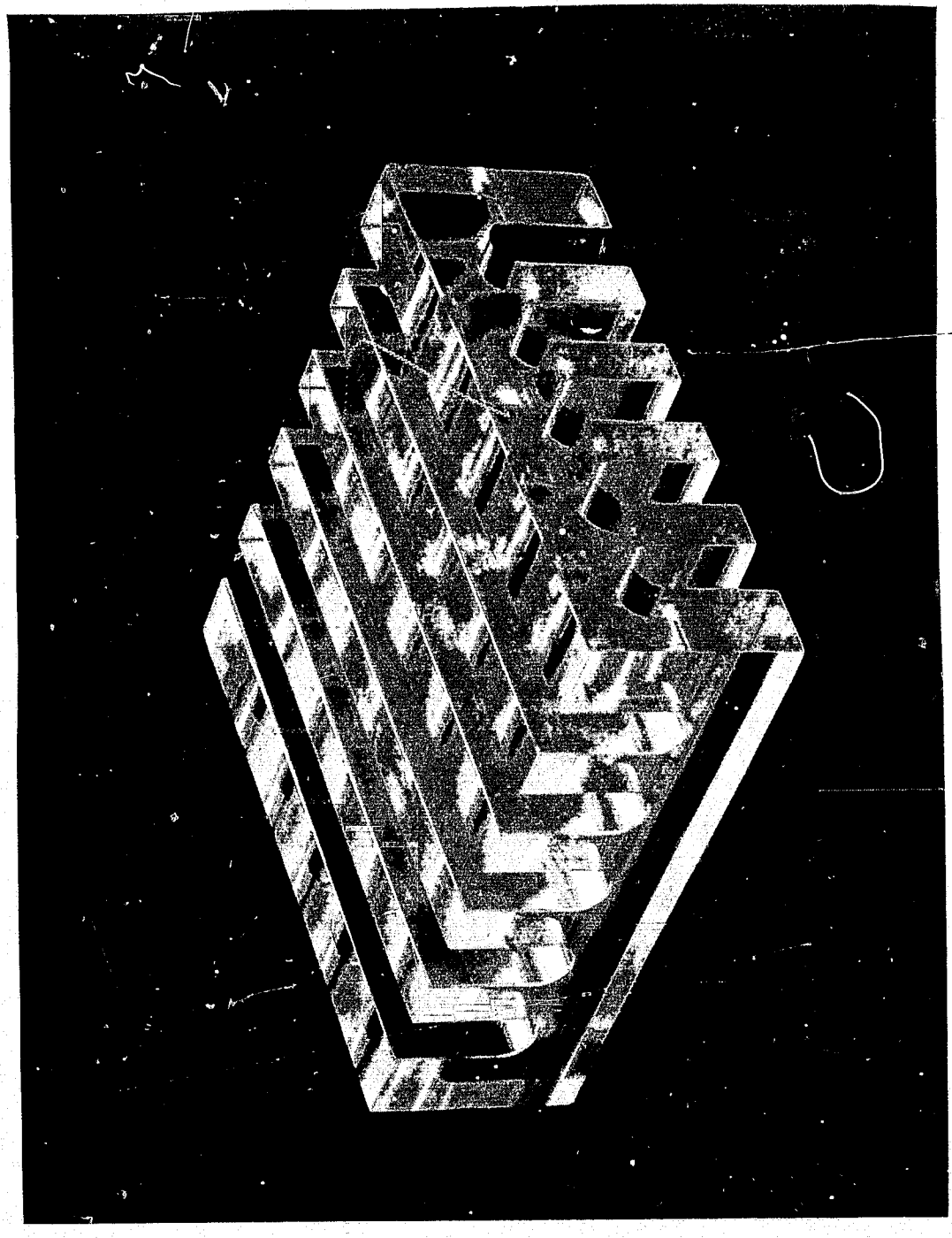


Figure M3. - Plastic Model of Slotted PLZT Cell

The results of the study show that two active media, PLZT and LC, have similar scattering profiles (on states) that are below that for ground glass and far below that for white paper (Figure M4). The relatively narrow forward scattering of these media, however, limits their effectiveness in wide-angle direct-view display applications (Figure M5). The comparison of their contrast ratios show that PLZT and LC undergo a contrast reversal (Figure M6) at a viewing angle near the optical axis.

In addition, electrically poled PLZT has a poor contrast ratio due to the poor transparency of the off state. A large reduction in the transparency of PLZT samples were shown when the material was switched from the thermally depoled state to the electrically poled state. It is evident that a strong requirement for an acceptable contrast ratio, in an electro-optic scattering media is a highly transparent off state. Consequently, the electrically driven PLZT material may have poor contrast for an aircraft display.

Conclusions

The disadvantages of the PLZT material are high drive voltages, cracking under stress, poor contrast with electrical depoling, small viewing angles, thermal sensitivity, a complex technology, and need for transverse electrical drive fields. Drive voltages of some materials can be as low as 40 volts (40 volt bias, ±3 volt drive) to as high as 300 volts. The PLZT material (ref. M4) shows an extreme sensitivity of the structure to electrical and mechanical stresses. Careful and thorough annealing is required to put the material in an isotropic condition at room temperature and ordering stresses have been found to generate mechanical as well as optical anisotropy. The need for a transverse drive field requires either the depositing of interdigitated electrodes or cross slotting of the PLZT material itself (Figure M5).

Interdigitated electrodes are required on both sides of the PLZT material to allow matrix element address. However, this may introduce half-select and grey-scale problems. Cross slotting of PLZT slabs and electrode deposition may not have these problems, but cutting will be difficult on 2- to 3-mil centers. The material is already susceptible to failure under stress, and cutting will probably result in a very fragile unacceptable structure.

In conclusion, although PLZT has the capability for intrinsic memory, there are ptoblems with both material and electroding techniques. Moreover, as no articles in recent literature have been found which suggest use of PLZT for display application, it appears that no one has found a solution to the problems.

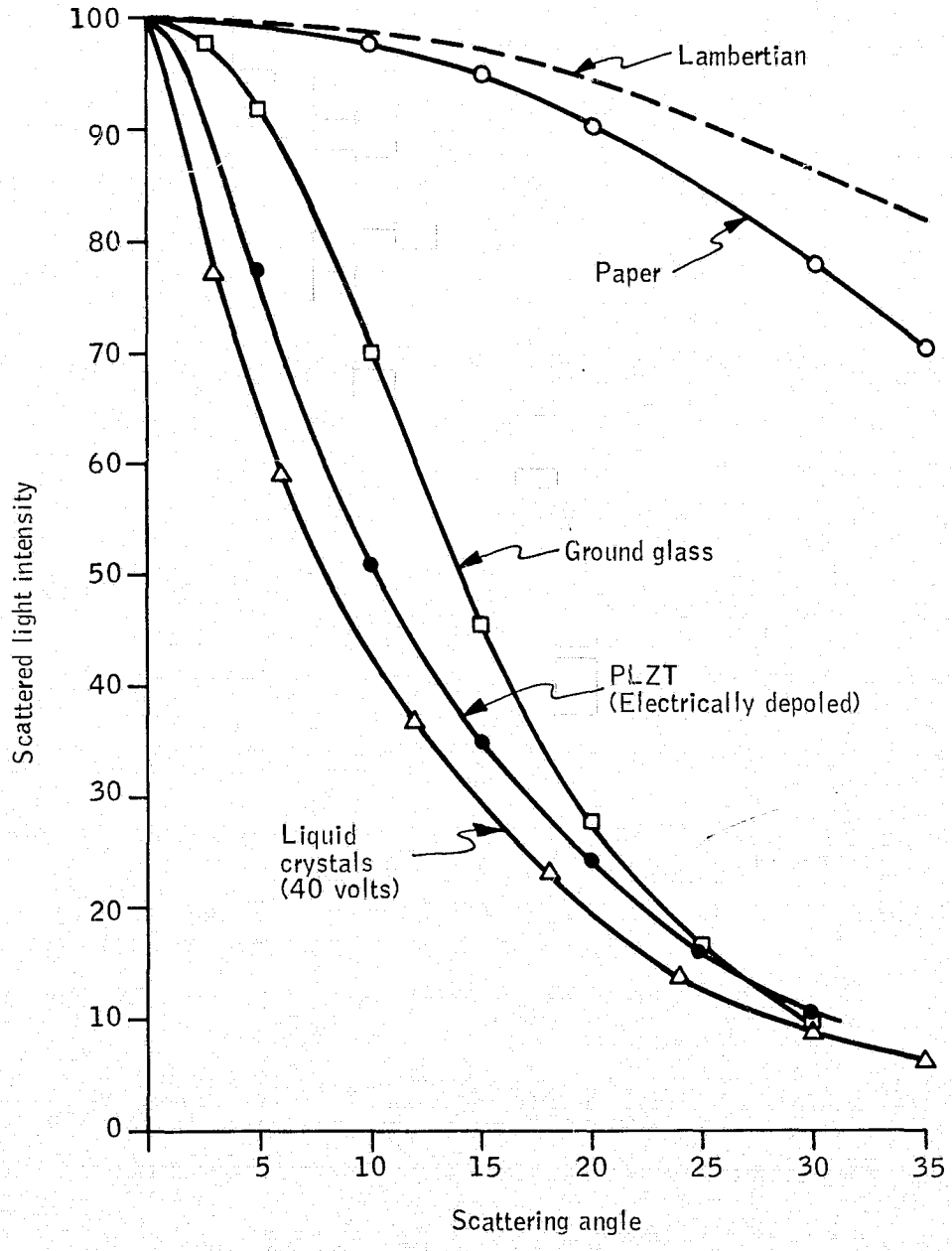


Figure M4. The Scattering Profiles of Various Display Media

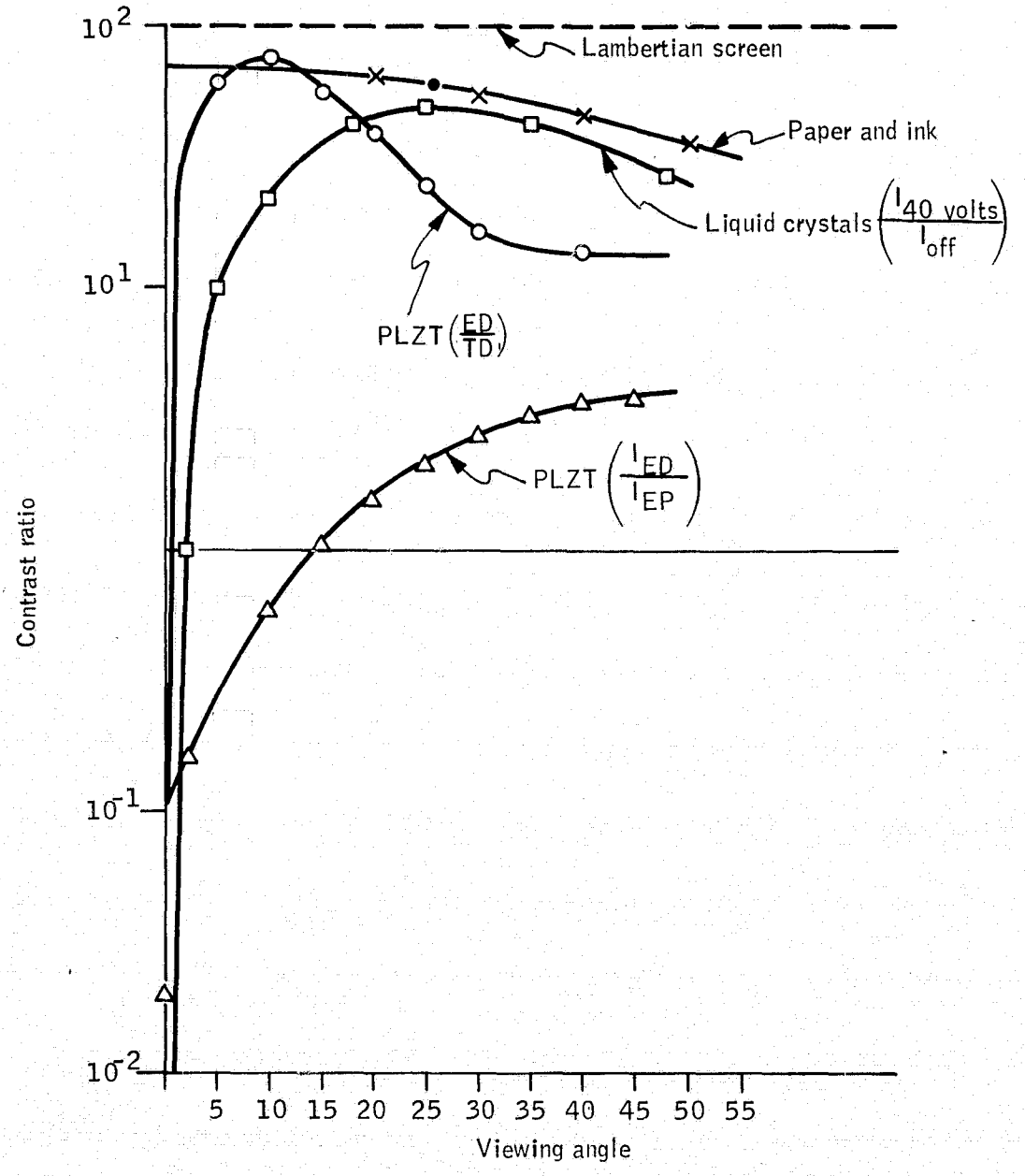


Figure M5. The Contrast Ratio of Several Display Media. (The on and off states are assumed to vary similarly so that the contrast ratio remains an arbitrarily chosen 100.)

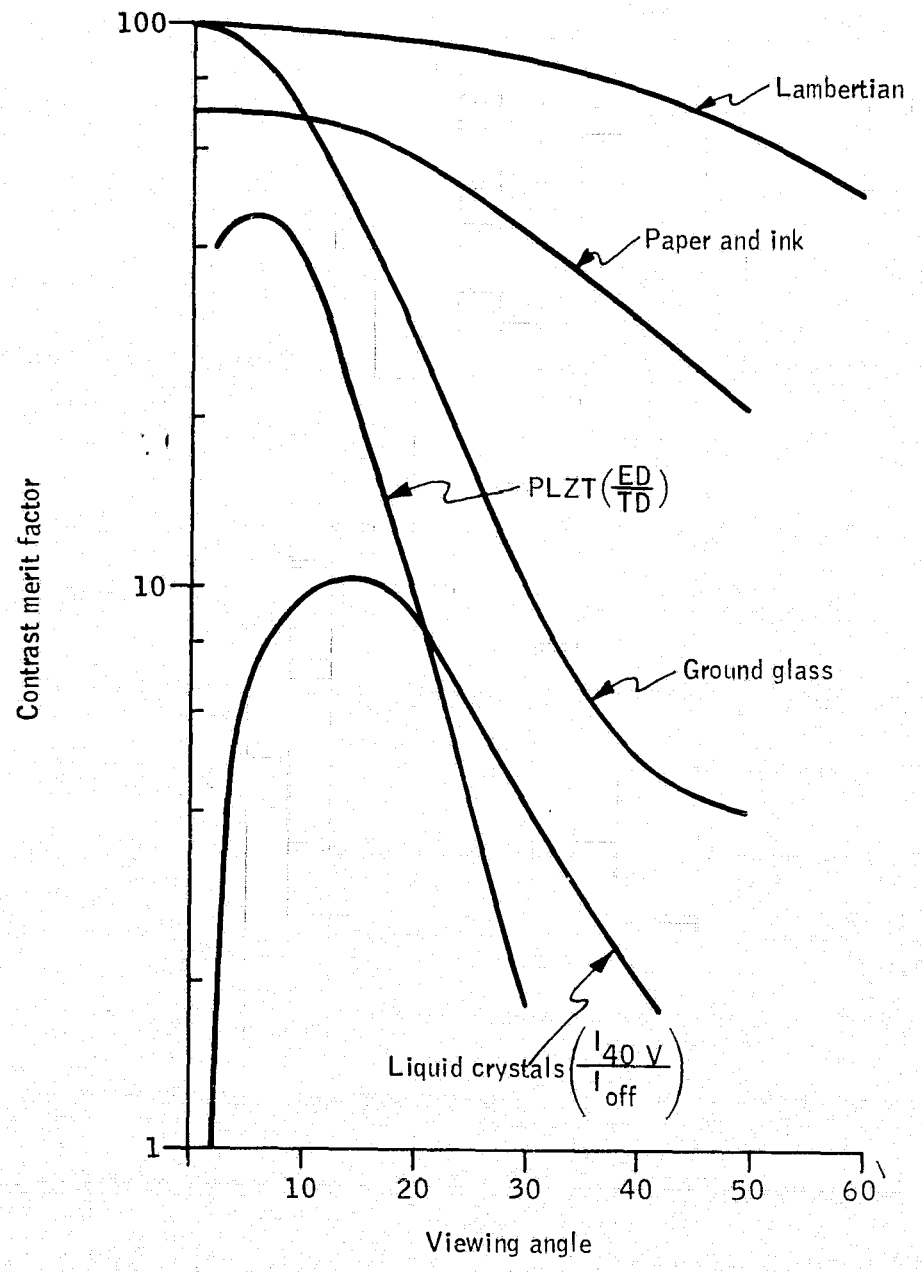


Figure M6. The Contrast Merit Factor is the Product of Contrast Ratio and the Scattering Light Intensity (normalized to unity on axis for each medium)

Appendix M

REFERENCES

M1. Bonne, U.: IC Compatible EO-Ceramic Display, Honeywell Patent Disclosure 857: 4-35, January 13, 1972.

M2. Bonne, U.; and Hicks, J.: Properties and Limitations of EO-Ceramics as Display Media II, Honeywell Memorandum HR-R- 1-243-1-32,

M3. Dolisa, A. L.: Convolution Scattering Model for Ferroelectric Ceramics and Other Display Media. Proceedings of the IEEE, Vol. 61, No. 7, July 1973, pp. 981-991.

M4. Meitzler, A. H. et. al.: Polymorphism and Penferroelectricity in PLZT Ceramics. Proceedings of the IEEE, Vol. 61, No. 7, July 1973, pp. 959-966.

APPENDIX N
GAS PLASMA DISPLAYS

Introduction

The gas-discharge panel has gained acceptance as a computer terminal display device. Its advantages are (ref. N1): high brightness, high contrast, inherent memory, direct digital addressing, and selective write-erase with see-through capabilities. Also, the gas panel is capable of displaying up to 300 characters. Its disadvanage is that it is too expensive to compete with the CRT (ref. N2).

A gas-discharge display panel consists of a pair of glass plates that are separated by a space containing the working gas mixture. Row and column electrodes are deposited on the inner surfaces of the glass and then all overcoated with a thin transparent dielectric, which acts to retain the wall charges. A sustaining ac voltage of peak value is maintained across the panel, although the voltage by itself is insufficient to ignite the discharge of the panel. Discharge occurs if a selected cell, which is defined by intersection of the electrode pair, is momentarily addressed by a higher voltage. The cell then breaks into a self-quenching discharge that repeats itself (sustains) at a lower drive voltage. The discharge is visible as a continuous glow in the neighborhood of the intersecting pair because of the high repetition rate (50 kHz). Erasure of the written spot occurs by temporarily applying an erase pulse of sufficient magnitude and polarity; thus, the sustaining voltage is reduced to below the threshold.

Current Display Technology

One display fabricated by IBM (ref. N1) uses a doughnut-shaped electrode that provides a port for viewing the gas discharge. This display has a packing density of 50 elements per inch, requires 340 to 480 volts to operate, and experiences no unwanted ignitions of neighboring cells.

National Cash Register Company has reported development (ref. N3) of a capacity coupled plasma line display that has a capacity for 16 segmented numerical digits. Thick-film techniques are evidently used for the electrodes, and neon gas is used to fill the panel. Projected life is over 20 000 hours, brightness uniformity of ±7 percent over the panel, contrast ratios are greater than 50:1,and power dissipation is 100 mw/0.7 inch digit (ref N2).

Burroughs has developed a technique (ref. N4) using low-voltage MOS elements to drive the high-voltage display. The approach is to regulate the voltage on the electrodes of the plasma display to about 50 volts short of a potential necessary to fire the tube. The MOS device can swing the anodes up 25 volts and a cathode down 25 volts to provide the firing ionization voltage; however, a cathode bias potential of -135 volts is still required. This technique requires one MOS element diode and two resistors per segment of a digit. Therefore, a minimum number of segments would be equal to the sum of the horizontal and vertical electrodes for an HMD.

One of the advantages of the display developed by Burroughs (ref. N5) is the self-scanning feature, which alleviates the interconnection problem for addressing each of the matrix elements. Normally, a 512 by 512 element GPD panel would require 1024 connections to the panel for addressing and driving circuits. However, this has been reduced by the self-scanning feature of the Burroughs panel to about seven drivers by using two cavities within one panel (ref. N6). One cavity is for the display and one is for

scanning. The scanning cavities in the panel have a glow discharge that shifts down each column, creating a glow scan. This creates, in effect, a shift register that is an integral part of the panel. The glow discharge is established on the back side of the reset cathode and is transferred sequentially down the entire length of the panel at the rear of each cathode. The glow scan is hardly visible from the front of the display. However, a refresh memory is needed for each frame as is the case for the CRT display.

The Owens-Illinois panel, on the other hand, uses a more complex multiplexed addressing such that only 18 lines are used instead of 1024 connections for a 512 by 512 panel while retaining the inherent memory feature.

Luminance Display

Peak luminance (B_R) of current matrix displays is 25 700 cd/m^2 (ref. N7). Table N1 gives the scale for the time-averaged luminance (B) of matrix displays without memory from $B_R = BXL/N$, where N = 3.

TABLE N1. MULTIPLEX OR MATRIX ADDRESS DISPLAY LUMINANCE

Line-at-a-time address, No. of lines (L)	Times averaged luminance, cd/m^2
1	25 700 (peak brightness)
16	4800
100	771
500	153
Eye adaption vision	3.4×10^4 (clouds)

Efforts are now underway to reduce high-voltage drive requirements (170-200 vdc) as high-voltage IC's are more vulnerable to breakdown. Experiments are also being conducted to find different phosphor materials for colors other than orange.

TV Flat Panel Displays

Most of the work done to develop flat panel TV is with gas-discharge devices. Efficiencies currently are at 0.5 lm/W (ref. N2), and the limit to illuminance is power dissipation and life. Gas-discharge devices have a sharp threshold for good contrast, while modulation voltages are low enough to be compatible with IC's. Response time fast enough for line-at-a-time addressing.

Gas discharge panels are of two types; dc and ac. Alternating current (ac) panels have memory (bistable - on or off) and stay on over the entire frame or refresh rate while dc panels do not stay on. Direct current (dc) panels are preferred for TV application due to simpler circuitry for gray-scale control and vertical scan. Table N2 (ref. N8) presents a summary of fifteen experimental TV flat panel displays announced to date with their performance. It should be noted that no panel has yet exceeded 342 cd/m^2 (100 ft-L) luminance, except for a single-cell, green, mercury vapor with positive column geometry, which must be operated at 80°C.

Two major problems have been reported (ref. N9) that have prevented the development of gas-discharge matrix cells with memory for higher luminance. First, the low impedance and negative resistance characteristics of gas discharge require high-value limiting resistors

and cause oscillations. Secondly, the write, erase, and sustain operating margins require close control of cell electrical characteristic uniformity. A report (ref. N10) has been published describing a gas discharge panel with graphite cathodes that provides memory and high luminance without the complication of individual megohm resistors for each cell. The object is to limit the current flow and to maintain dc voltage to sustain display luminance. Luminance of 3000 cd/m^2 is claimed.

Further progress (ref. N11) is being made on color displays, reduction of drive electronics cost, reductions of the number of panel connections, and reduction of control voltages to within the capability of MOS IC's. Progress is also being made in gas-discharge display technology and should be monitored for application to aircraft displays.

Conclusions

It is predicted (ref. N5) that the resolution limit for GPD's is about 40 lines per cm. Current resolution is about 20 lines/cm (Table N2). GPD's require greater drive complexity than LCD's, and the electronics is more expensive. However, if LSI's can be modified to higher voltages or if GPD operating voltages can be reduced, then the cost of the driving circuits could be reduced.

Although power consumption is low, high drive voltages of 170 volts are a distinct disadvantage. MOS elements might be used with the Burroughs low-voltage technique (ref. N4), but mounting these elements presents a mechanical problem. On-site or inherent memory must be used for displays over 100 lines as the luminance is not high enough for aircraft application with line-at-a-time address. Most GPD's lack grey scale, but it is being developed, and grey scale is of no concern for an alpha-numeric display.

Costs of present GPD's are too high for a 500-line matrix; the cost per digit is \$0.75 to \$1.90, and system cost is \$1.50 to \$3.00 per digit. Assuming the cost per matrix element is equal to one segment, 1/9 of \$0.75 or \$0.08/element yields a display cost of \$21 000.

This technology should be followed as progress is being made to reduce cost, drive complexity, and to improve luminance.

Appendix N

TABLE N2. — CHARACTERISTICS OF CURRENT GPD

Company	Panel type, region of discharge used	Color	Reported luminance (foot-lamberts) duty factor (percent)	Calculated luminance at 0.600 or 0.2 duty factor	Luminous efficiency (lumens/watt)	Contrast ratio	Video storage, intensity modulation	Panel size rows x columns
Philips	dc, negative glow visible emission	Orange	60 ft-L 1.1%	11 ft-L	0.2 ℓm/W	10:1	Analog, continuous duty factor	2 in. x 5 in. 40 x 100
Zenith	dc, negative glow visible emission	Orange	8 ft-L 0.2%	8 ft-L	0.1 ℓm/W	40:1	Analog, continuous current	6.3 in. x 2.4 in. 212 x 80
Bell Labs	dc, negative glow visible emission	Orange	25 ft-L, 0.33%	15 ft-L	0.1 ℓm/W	30:1	Analog, continuous current	8.1 in. x 3.4 in. 222 x 77
Japan NHK Broadcasting Co.	dc, negative glow visible emission	Orange	15 ft-L, NR	NR	NR	25:1	NR	NR
Bell Labs	dc, negative glow visible emission and phosphor excited by low-energy electrons	White	11 ft-L, 1%	2.2 ft-L	0.09 ℓ m/W	NR	Analog, continuous current	6.6 in. x 0.42 in. 110 x 7
Philips	dc, positive column UV excites phosphors	Red, Blue, Green	NR; 2.5%	NR	0.8 ℓm/W	NR	NR	0.040 in. diam, 16 x 16
Toshiba	dc, constricted glow visible emission	Orange	60 ft-L, 0.35%	0.6 ft-L at 4×10^{-4} T	0.4 ℓm/W	20:1	Digital, 16 discrete current levels	2.9 in. x 3.9 in. 48 x 65
Mitsubishi	ac, negative glow visible emission	Orange	NR	NA	NR	NR	Digital, 12 discrete duty factor levels	NR 128 x 128
Sony	dc, negative glow visible emission	Orange	25 ft-L 0.24%	21 ft-L	0.5 ℓm/W	40:1	Digital, 32 discrete levels using duty factor and current modulation	4.2 in. x 5.5 in. 212 x 282
Hitachi	dc, negative glow UV phosphor excitation	Red, Blue, Green	5 ft-L white, 0.38%	2.5 ft-l white	0.05 ℓ m/W	8:1	Digital, 64 discrete levels using duty factor and current modulation	4.7 in. x 6.3 in. 120 x 160
NHK	dc, negative glow UV phosphor excitation	Red, Blue, Green	5 ft-L white 0.8%	1.25 ft-L white	0.07 ℓm/W	30:1	Analog, continuous duty factor and current modulation	5 in. x 6.4 in. 127 x 160
GT&E	ac, negative glow visible emission	Orange	25 ft-L, nearly 100%	NA	NR*	NR	Digital, 64 discrete duty factor levels	8.5 in. x 8.5 in. 512 x 512
GT&E	dc, positive column UV excites phosphors	Red, Blue, Green	19 ft-L green, 0.15%	25 ft-L green	1.2 ℓm/W	Data is for single cells, no panels reported		
Zenith	dc, positive column UV excites phosphors	Red, Blue, Green	360 ft-L green spot luminance, 0.18%	400 ft-L green spot luminance	3.4 ℓm/W	Data is for single cells, no panels reported		
Bell Labs	ac, negative glow visible emission	Orange	NR, 100%	NA	NR*	NR	Digital; ordered dither	512 x 512

NA = Not Applicable

NR = Not Reported

*AC "Digivue" panels typically have a luminous efficiency of several tenths of a lumen/watt (19).

Appendix N

REFERENCES

N1. Welber, B. et. al.: A Gas Display Panel with Quasi-Axial Electrode Configurations, 1972, SID Digest of Technical Papers, p. 32.

N2. Weston, G. F.: Plasma Panel Displays. Journal of Physics, E.: Scientific Instruments, 1975, Vol. 8, pp. 981-991.

N3. Coleman, W. E., et. al.: An-All-Thick-Film A. C. Plasma Line Display. Proceedings IEEE, Nov. 1972.

N4. Lee, J. Y. et. al.: MOS Chip Plus Level-Shifting Circuit Drives Gas-Discharge Display. Electronics, March 1973.

N5. Stepke, E. T.: Display Terminals: The Callenge of the GPDs Electro-Optical System Design, October 1972.

N6. Sobel, Alan: Summary; New Techniques in Video Displays. IEEE Transactions on Consumer Electronics, Vol. CE-21, No. 3, August 1975, pp. 290-297.

N7. Van Roalte, J. A.: Matrix T.V. Displays; System and Circuit Problems. IEEE Transactions on Consumer Electronics, Vol. CE-21, No. 3, August 1975, pp. 213-219.

N8. Chodil, Gerald: Gas Discharge Plasma for Flat-Panel. IEEE Transactions on Consumer Electronics, Vol. CE-21, No. 3, August 1975, pp. 221-235.

N9. Holtz, G. E.: Post Gas Discharge Display with Memory, 1972, SID Digest of Technical Papers, p. 36.

N10. Smith, James: Experimental Storage Display Panels Using DC Gas Discharges without Resistors. IEEE Transactions on Electron Devices, Vol. ED-22, No. 9, September 1975. pp. 642-649.

N11. Sobell, Allan: Introduction to Special Joint Issue on Display Devices and Systems. IEEE Transactions on Electron Devices, September 1975, pp. 629-633.

APPENDIX O

ELECTROLUMINESCENT DISPLAYS (ELD)

Electroluminescent (EL) matrix displays are under development, and resolutions of up to 12 line pairs/cm^{-2} for areas of 15 by 15 cm have been reported (refs. O1 and O2). Vacuum-deposited thin-film transistors provide the required nonlinearity for good threshold and a contrast of 50:1 (ref. O3). Matrix address is line at a time with an ac drive of 80 volts (rms) at 8 kHz. Consequently, the capacitance of the ac driven phosphor between two electrodes cannot be used as a storage capacitor; an extra capacitor and associated switch (transistor) is required for on-site memory to provide frame-period storage at each display element.

Maximum cell luminance, with a virtual 100 percent duty cycle, is 140 cd/m^2 Power consumption is one watt under typical alphanumeric display conditions. Progress is being made toward developing a gray-scale capability, and the display is expected to operate at TV video rates.

Direct current EL phosphors do not require on-site memory and can be driven at lower voltages but have a short lifetime. Alternating EL driven phosphors have a life of over 20 000 hours with luminance levels of 340 cd/m^2 (ref. O4), but the driving voltage becomes high (250 V rms).

REFERENCES

O1. Brody, T. P.: Large Scale Integration for Display Screens. IEEE Transactions on Consumer Electronics, Vol. CE-21, No. 3, August 1975, pp. 260-288.

O2. Kramer, G.: Thin-Film-Transistor Switching Matrix for Flat-Panel Displays. Proceedings of the SID, Vol. 16/3, Third Quarter, 1975.

O3. Brody, T. P.: A Six by Six Inch 20-Line per Inch Electro-Luminescent Display Panel. Proceedings of the SID, Vol. 16/3, Third Quarter 1975, pp. 158-167.

O4. Sobel, A.: Summary: New Techniques in Video Displays. IEEE Transactions on Consumer Electronics, Vol. CE-21, No. 3, August 1975.

APPENDIX P

THIN FILM VERSUS SILICON TECHNOLOGY

One promising approach to the development of an integrated matrix flat panel display has recently emerged from a maturing thin-film device technology. Thin-film technology (TFT) has now advanced to the state where thin-film integrated circuits (TFICs) are deposited on the same insulating substrate in one vacuum pump-down. Thus, integration of the drive electronics around the edge of the flat-paneled display with the X-Y electrodes and on-site memory is possible. TFICs offer the advances of low cost, unique packaging, and systems applications for the development of an aircraft display.

The present integrated circuit (IC) technology uses monolithic silicon crystal integrated circuits (SIC) formed within the surface of a single block of silicon. The associated passive elements are either formed within the silicon surface or are deposited over the semiconductor substrate by thin-film techniques. The TFT approach is to replace the crystalline semiconductor by an inert insulating substrate. All components, including active and passive elements, are deposited upon a substrate by thin-film techniques. These thin films are prepared entirely by sequential thermal evaporation through shadow masks of the respective metal, semiconductor, and dielectric materials on the insulating substrate.

TFTs (matrix on-site memory) deposited on insulating substrates have many advantages. It makes interfacing with other electronic and system components easier. Also, this technique does not suffer from any of the limitations of high voltages or high power and can be used for gas panel matrix displays requiring high or low voltage LCDs. Costly chip and wire bonding is eliminated and reliability is improved over silicon technology, and TFTs have a higher yield over larger areas than silicon chips. Also, localized defects in silicon substrates will result in defective transistors; this is not the case with TFTs. TFICs have a much lower cost than SICs due to the inexpensive substrate and starting materials and the single-step batch processing. Silicon technology is a maturing technology, approaching the end of its rapid growth. As this technology begins to saturate, the next big breakthrough in cost reduction will come from thin-film technology, which has the potential for a factor of 10^2 to 10^3 lower cost. Also, development of TFTs was abandoned in the late 1960's due to persistent stability problems. These stability problems have been solved, however, and the solution has been confirmed by research with experimental data (refs. P1 and P2).

Fabrication of liquid crystal (LC) cells with thin-film deposition of TFT, storage capacitance, and X-Y electrodes has already been accomplished. For example, a laboratory thin-film twisted nematic LC transmissive cell developed by Westinghouse has a 20 line/inch resolution (ref. P3).

The basic addressing scheme of the LC thin-film matrix display is shown in Figure P1. Each matrix element of the LC panel is composed of a TFT, which is integrated with a suitable storage capacitor (C_{11}, etc.). The combination allows rapid address and temporary capacitive storage of charge sufficient to activate the slower responding LC. The TFT is used only as a simple on-off switch and the required performance is modest.

Matrix address is at a line-at-a-time rate. The row generators (Figure P1, R_1, etc.) turn on the TFTs along a given row. A serial-in-parallel-out shift register containing one horizontal line of video information is connected to the column generators (C_1, etc). The charge, which represents a video pulse level, is then stored in capacitor C_{11} through the conducting TFT from column C_1. When the next row, R_2, is addressed, the formally addressed TFTs turn off and trap the charge on the capacitor. There it

Appendix P

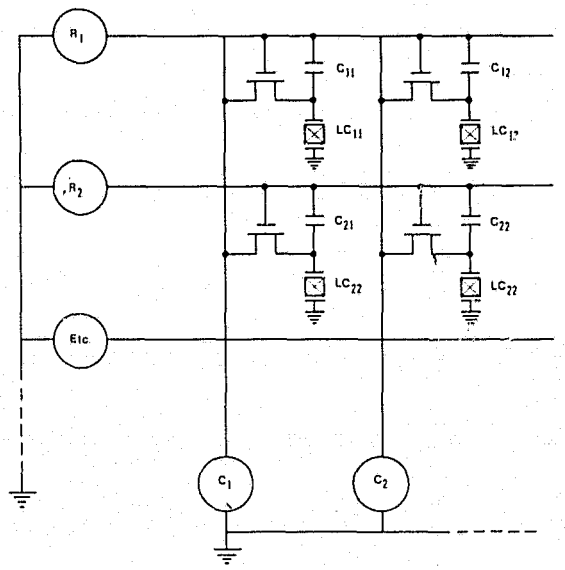


Figure P1. Line-at-a-time Addressing Scheme

stays until the next addressing occurs at one frame time later. The charges of video information on columns C_1 through C_2 come from the storage of the serial-in-parallel-out shift register. The polarity of the video signals are reversed for each frame period.

The TFT, X-Y/electrodes, and capacitors are deposited on a LC glass face plate in one vacuum pump-down. This is achieved in twelve deposition steps through the use of a variable multi-aperture mask and four materials (ref. P3). It is expected that thin-film depositions on a liquid crystal will reduce the number of wires and connections needed for matrix address. The drive electronics must be mounted along the edges of the panel to allow operation of a transmissive display and must be integrated with a termination of the X-Y electrodes.

Appendix P

REFERENCES

P1. Cramer, G.: Thin Film Field Effect Transistors. International Microelectric Symposium, Oct. 22, 1973.

P2. Sadasiv, G.; et al.: Thin Film Circuits for Scanning Image-Sensor Arrays. IEEE Transaction, ED15, 1968, p. 215.

P3. Fisher, A.: Design of a Liquid Crystal Color TV Panel. IEEE Conference on Display Service, C, 11, Oct. 12, 1972.

APPENDIX Q
LUMINANCE AND CONTRAST REQUIREMENTS

Luminance of emissive displays is reduced if the active elements are not on during the 0.1 second integration time of the eye. Most display materials are quoted for their peak luminance unless it is the time-averaged luminance of the display that is measured. It is necessary to know the refresh rate of a display when it is multiplexed or matrix addressed to establish the relationship between the required peak luminance (B_R) and desired time-averaged luminance (B).

$$B_R = \frac{B \times L}{N} \qquad (Q1)$$

where

N = number of refresh periods during 0.1 second integration time of the eye.

L = number of characters to be sequentially multiplexed or the number of lines of a matrix display being driven one line at a time.

The required peak luminance must be 170 times the time-average of brightness for a 500-line matrix address display with a 30 TV frame rate. It is apparent that a display with intrinsic or on-site memory would greatly improve display luminance as the driven element would be on during the entire refresh period.

The luminance of a display (instantaneous threshold) must be at least 1/100 of the luminance level of the surroundings (to which the eye is adapted) to be visible. However, this data was from test of a black square with an angular subtend against a white background, and a smaller ratio of 2/100 or 5/100 should probably be used for a/c displays Therefore, an aircraft display should have a luminance of from 680 to 1700 cd/m^2 to be visible with the eye adapted to the luminance levels of white clouds (i.e., 34 000 cd/m^2).

The intrinsic contrast between on and off elements of a display are degraded when that display is multiplexed. The relationship of the required display intrinsic contrast ratio (C_R) to the desired time-averaged contrast ratio (C) is

$$C_R = \frac{C \times L}{N} \qquad (Q2)$$

The intrinsic contrast between cells is measured when the display element is on over the entire integration time of the eye. The time-averaged contrast is measured while the display is being multiplexed or matrix addressed and the display cell is not on continuously. If a time-averaged contrast of 10:1 is required, then an intrinsic contrast of 1700:1 is required for a 500-line matrix display with a 30 frame refresh rate (N = 3.0). Because few display materials have this high of an intrinsic contrast, on-site memory is needed for matrix addressed passive displays.

This contrast expression (Equation Q2) is valid for a passive reflective display under any conditions of ambient illumination as the luminance ratio is the same as the contrast ratio. However, the contrast of an active display is degraded by any ambient illumination on the display due to direct sun illuminance or ambient illuminance from the sky. The time-averaged contrast (C) of an active display is also

$$C = C_R \frac{N}{L} \qquad (Q3)$$

or

$$C = \frac{B_{Hi}}{B_{Lo}} \frac{N}{L} \tag{Q3a}$$

where B_{Hi} is the peak cell brightness during the on time, and B_{Lo} is the average cell brightness during the off time.

The time-averaged contrast ratio changes when the luminance due to the effective display reflectance (R) with an incident illumination (E) is added to the intrinsic cell luminance levels B_{Hi} and B_{Lo}:

$$C = \frac{(B_{Hi} + R\,E/\pi)}{(B_{Lo} + R\,E/\pi)} \frac{N}{L} \tag{Q4}$$

or

$$C = \frac{(B_{Hi} + R\,E/\pi)}{(B_{Hi}/C_R + R\,E/\pi)} \frac{N}{L} \tag{Q4a}$$

It can be seen that the time-averaged contrast is reduced by any illuminance on the display. The required (peak) display luminance (B_{Hi}) during the cell on time is found from the above equation to be

$$B_{Hi} = \frac{RE(N/L - C)}{\pi\,(C/CR - N/L)} \tag{Q5}$$

An example of increased display luminance requirements is given here for explanation. Direct sun illuminance is about 100 000 lm / m^2. Ambient illuminance (E_C) on the display from white cloud luminance (B_C) over a solid angle (Ω) of 1.6 steradians (estimate of solid angle of white, bright clouds through cockpit windows) is approximated as

$$E_C = \frac{\Omega\,B_C}{2} \tag{Q6}$$

$$E_C = \frac{1.6 \times 34\,000\ cd/m^2}{2}$$

$$E_C = 28\,000\ lm / m^2$$

A nonmultiplex display (N/L = 1.0) in direct sunlight with an intrinsic contrast ratio (C_R) of 100:1, a desired display contrast (C) of 10 with a cell reflectance of 0.05 would require an intrinsic peak luminance (B_{Hi}) of

$$B_{Hi} = \frac{RE\,(1 - C)}{\pi\,(C/C_R - 1)} \tag{Q7}$$

In substituting for the case of direct sun illumination,

$$B_{Hi} = \frac{0.05 \times 10^5}{\pi} \frac{(1 - 10)}{(10/100 - 1)}$$

B_{Hi} = 16,000 cd/m^2 and in cloud ambient illumination, B_{Hi} = 4456 cd/m^2.

It should be noted that as the intrinsic contrast (C_R) approaches the required contrast (C) or when C/C_R becomes unity, the display luminance requirement becomes infinite. It can be concluded that the intrinsic contrast of a nonmultiplexed display should be roughly ten times the desired display contrast to avoid unreasonable display brightness requirements in an aircraft.

The display luminance and intrinsic contrast ratio requirements for a matrix line-at-a-time display becomes severe in the ambient illumination of an aircraft cockpit. The required peak display intrinsic luminance is

$$B_{Hi} = \frac{RE\ (N/L - C)}{\pi\ (C/C_R - N/L)} \qquad (Q8)$$

The numerator is negative as N/L is less than C for matrix displays. Therefore, C/C_R must be less than N/L to have a positive (real) solution for the display peak luminance. If C/C_R is to be 10 times less than N/L to minimize the peak luminance requirements, then the intrinsic contrast (C_R) must be 17 000:1. This is for the matrix display case for C = 10, L = 500 lines, and N = 3.

$$C_R > 10\ \frac{CL}{N} \qquad (Q9)$$

$$C_R > \frac{10 \times 10 \times 500}{3}$$

$$C_R > 16\ 660{:}1$$

This high contrast is expected because matrix address cell elements are on for a short time, making residual luminance of the "off" cells significant. Therefore, it is critical that there be very little drive leakage to the off cells, i.e., the off cells residual luminance must be 17 000 times less than the peak luminance of the on cell. The peak display luminance requirement for this case in direct sunlight is

$$B_{Hi} = \frac{0.05 \times 10^5\ (3/500 - 10)}{\pi\ (10/17\ 000 - 3/500)} \qquad (Q10)$$

$$B_{Hi} = 3.0 \times 10^6\ cd/m^2 \text{ for direct sun luminance}$$

$$B_{Hi} = 8.2 \times 10^6\ cd/m^2 \text{ for ambient sky luminance}$$

These luminance levels are very high compared to what is available for flat panel materials. The best solution for matrix displays in an aircraft is to develop on-site intrinsic or inherent memory capability; it can be concluded that development of on-site memory is a firm requirement for active matrix addressed displays in an aircraft.

This analysis does not account for the improvement of display contrast and reduction of display brightness through the use of ambient light suppression face plates, louvers, or filters because these devices will probably not reduce the brightness or contrast requirements of a nonmemory active display to a reasonable level. These devices, however, will permit some active nonmultiplex displays or a matrix display with on-site memory to function in ambient illumination. Operation in direct sun may be possible only for a passive display in the nonmultiplex mode.

Appendix Q

REFERENCES

Q1. Semple, C. A.: Analysis of Human Factors Data for Electronic Flight Display Systems. AFFDL-TR-70-174, April 1971, pp. 266-267.

APPENDIX R

OPTICAL MEMORY PROJECTION

Optical memories are being researched using both real image and holographic storage. The major problems in this area are concerned with the storage media: the major need is a permanent, yet selectively erasable, storage medium. Techniques that may be viable in the 1980 time frame are given Table R1. (reference R1). Two of these approaches will be discussed in more detail. Optical memories will be aimed at the secondary and mass memory areas of the memory hierarchy and ,along with bubbles, constitute a threat to existing magnetic technologies.

Table R1. Characteristics of Optical Storage Media

Memory Material	Storage Density Bits/CM^2	Write/ Erase	Read	Problems
Photochromic (Alkali Halides)	10^6	Optical Absorption	Optical Absorption	Gradual data loss
Ferroelectric ($BaTiO_3$)	10^4	Ferro-electric	Current Sense-Destruction	Resolution-lifetime
Ferromagnetic MnBi	10^7	Thermal	Magneto-Optic	Low readout efficiency
Thermoplastic	10^5	Thermal	Optical Hologram	Resolution-efficiency
Semiconductor Capacitors	10^7	Electron Beam	Optical	Cathode life beam deflection

Electron Beam Addressed Memory (EBAM)

An EBAM memory is shown in Figure R1. A beam of electrons is generated by an electron lens onto a storage target, where the beam causes local physical changes. The bit size corresponds to the size of the beam at the target. Addressing is achieved by deflecting the beam to different areas of the target plane. Either random access or block access can be achieved, depending on how the target is scanned. Electrostatic means can be used to focus and deflect the beam.

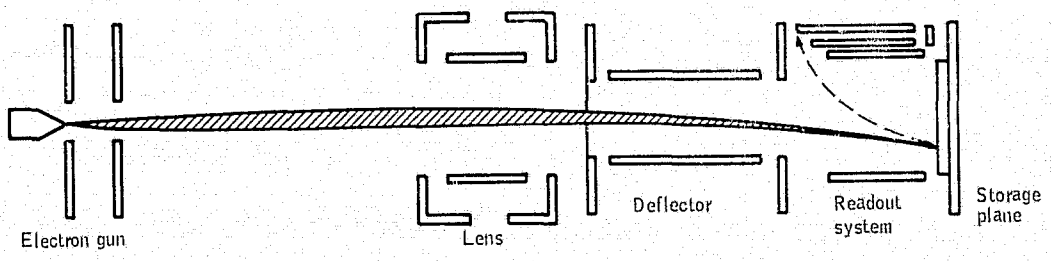


Figure R1. Major Components of EBAM Tube

One method stores data by depositing charge generated by the beam in the oxide of the MOS capacitors located on the target place. The charge state of the oxide is subsequently probed by the same beam during readout. No structure is necessary in the plane of the MOS target, thus fabrication and beam-target registration are simplified. Fatigue effects caused by prolonged write-erase operation have been noted. These effects are thought to be related to radiation-damage effects in the bulk silicon (reference R2). EBAM's are useful for storage systems in the 10^7 to 10^{10} bit range and can provide access in 0.5 to 3 μsec. Write speeds of 1 megabit/sec and read speeds of 10M bits/sec can be achieved. Because MOS capacitors are used, the data must be refreshed. Volatility can be ensured with a flood beam to restore lost charge. Costs of 0.02¢ to 0.05¢/bit are forecast with densities of 2-4x10^6 bits/inch^2 (reference R3, R4).

General Electric's BEAMOS (Beam Addressed Metal Oxide Semiconductor) Memory uses unstructured MOS chips and an electron lens to read, write, or erase. The memory can operate without power for over a month, has an access time of 30 μsec, and a transfer rate of 10 Mbits/sec (reference R5).

Problems that must be overcome in this technology include:

- Beam forming and deflection
- Improving cathode life
- Improving storage medium
- Reducing fatigue effects

High-density modules will be fabricated with the main limitations on size being the size of the storage cell and the amount of deflection possible.

The dynamic nature of this technology, and the flood beam necessary to achieve non-volatility, limits the use of this high-capacity technology to on-line applications rather than archival storage. The speed of operation qualifies it for secondary storage use.

Laser Memories

Two basic types of read/write laser memories are under development today: one stores a hologram of the data, and the second type stores the actual data in a bit serial fashion.

Holographic Memories. — The storage media most commonly used are photographic film and thermoplastic materials. A block diagram of a laser holographic memory without moving parts is shown in Figure R2 (reference R6). Such a memory with a capacity of 10^8 bits and access times of 1 to 5 μsec should be feasible in 1980 if a suitable storage medium can be developed. While the theoretical bit density limit is on the order of 6x10^8 bits/in^2, the practical limit may be more like 6 x 10^7 bits/in^2. Experiments have already produced systems with data rates of 50 Mbit/channel, thus high speed can be expected from this technology (reference R7). The elements shown in the figure perform the following functions:

- Laser - provides an intense monochromatic light source.
- Modulator - a crystal whose temperature and bias voltage is very precisely controlled to effect a transparent-to-opaque state change and thus modulate the laser beam.
- X-Y Deflector - a crystal whose index of refraction is varied by appropriate voltage and temperature bias to deflect light.
- Deflector Optics - used to direct and condense light.

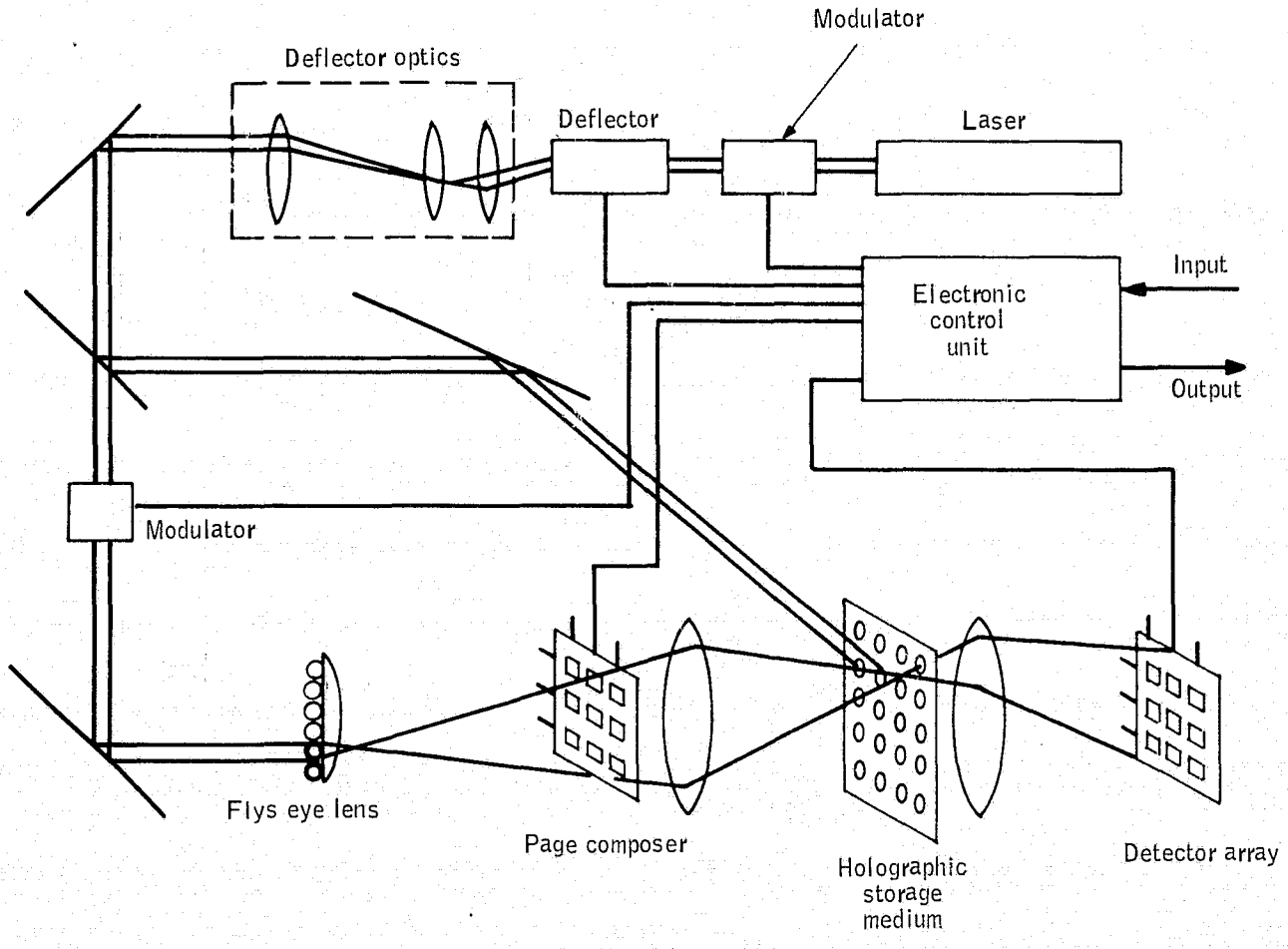


Figure R2. Holographic Memory without Moving Parts

- Page Composer - an array of light valves upon which information is composed before entry into the storage medium.
- Storage Medium - the medium upon which the digital information is stored.
- Detector Array - a photosensitive readout transducer.
- Control Unit - provides overall system control.

The storage process requires that data bits first be entered and stored in a page composer, which is an array of x-y addressed light valves or shutters electrically or optically operated according to the required bit positions and states, as specified by the control unit. This transforms the data bits into a two-dimensional digital pattern or page having light and dark spots.

The page composer is placed near a lens; when the page composer and lens are illuminated with the laser beam, a pattern, which is the Fourier transform of the bit pattern array produced by the page composer, is formed at the plane of the holographic storage medium. A reference beam, obtained by splitting the original laser beam, is made to be coincident with the Fourier transform pattern. The resulting interference pattern is recorded in the storage medium in the form of a hologram. Because individual bits are not uniquely recorded in a hologram, the system is quite tolerant of such things as dust or surface impurities on the recording medium.

Holograms are stored in other locations on the storage medium by moving the incident beam with the beam deflector. The beam deflector system is followed by an optical system that positions the beam on any element in an x-y array of small fly's-eye lenses. Movement of the beam from one lens to another changes the position of the recorded hologram in the x-y plane. Likewise, the reference beam tracks the signal beam such that both address the same spot in the storage medium.

Data are read out by addressing a hologram with only the reference beam or else another beam incident at the same angle as the reference beam. The reference beam is diffracted by the hologram, and an image of the page composer data is projected through a field lens onto a photodetector array having the same relative dimensions as the page composer. Each bit originally stored in the page composer is incident on a photodetector in the array, and the data are converted back to an electric signal. Each photodetector is connected to a transistor flip-flop, which is set by the photodetector signal and read out by the controller (reference R8).

Figure R3 shows a holographic memory that employs a moving tape. The operation of this system is much the same as the preceding approach except that a one-dimensional hologram is stored on a movable medium (in this case a thermoplastic tape). While this approach suffers an access time penalty over the fixed medium approach, it allows storage of 2×10^8 bits per linear foot of 1/4 inch wide tape. The maximum storage capacity and access time is, thus, a function of the tape length. A 50-foot tape could contain 10^{10} bits.

Serial Memories. – The two-dimensional holographic memory stores or records up to 10^5 bits at a time; the single-dimension hologram allows about 300 bits to be stored and read at once. The serial laser memory stores a single bit at a time. Thus, the read/write speed of the holographic approaches are much higher than the serial laser memory.

Serial laser memory development has been aimed at both ROM and RAM applications. The main difference between this approach and the holographic approach is the use of low power - thus low cost - lasers and the elimination of complex optical parts.

As with the holographic approach, serial memories can be constructed using a fixed or moving media and have about the same storage density and capacity. The main drawbacks of the approach are the speed reduction due to serial operation, and the dependence on media uniformity due to the single location of each data bit. The main advantages are cost, size, and reliability.

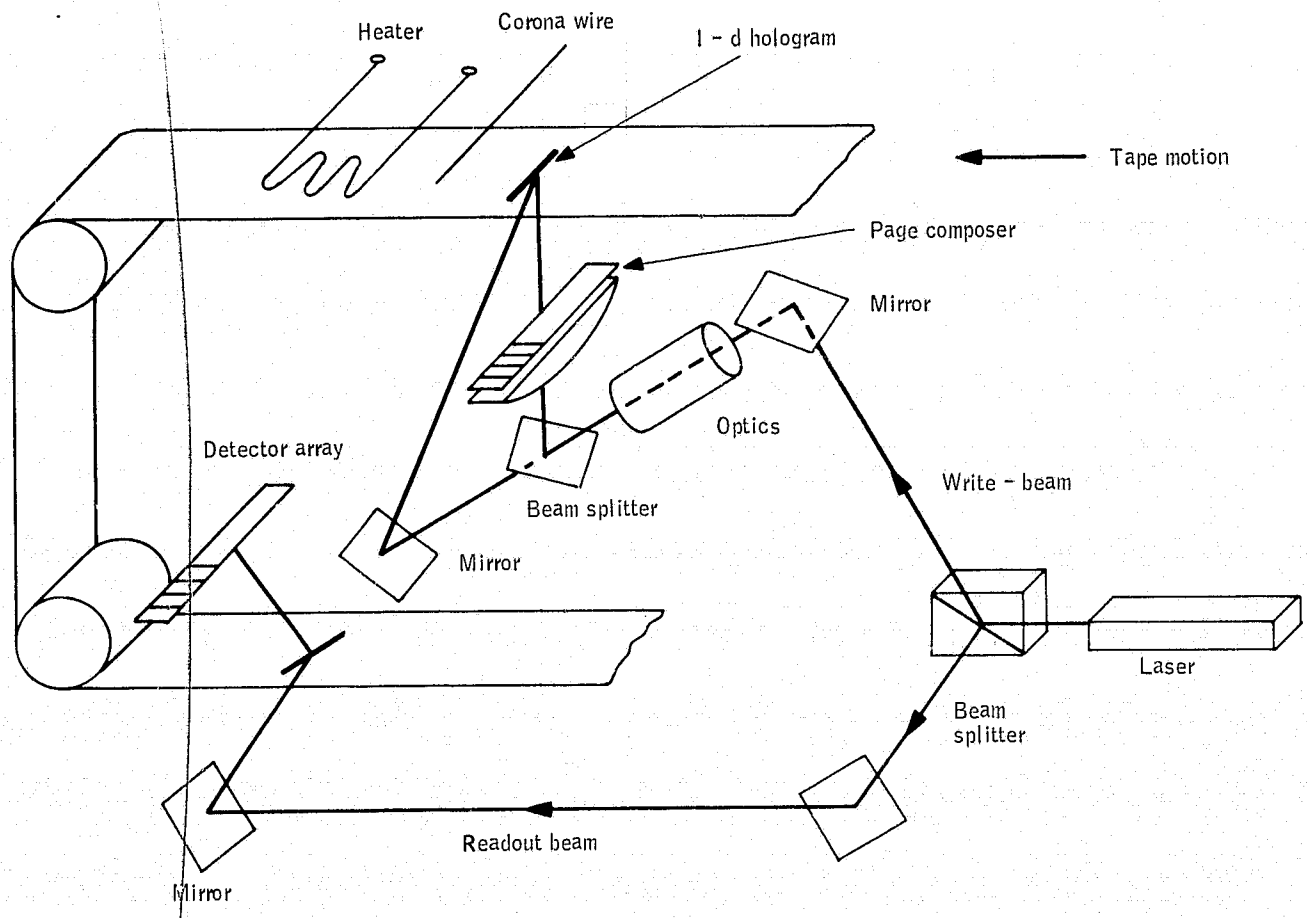


Figure R3. Holographic Memory with Moving Tape

Problem Areas. – Technical difficulties are currently encountered in all subelements of laser memory systems and some are discussed briefly below.

Laser power of up to 10 watts may be required for the holographic approaches, and current prices would place these devices at around 20 thousand dollars, thus affecting cost advantages of the system. Lasers also require elaborate cooling apparatus if they are the high-power variety (reference R8).

The most serious roadblock to development of a read/write laser memory is that of the storage medium. If a suitable medium is not found, R/W laser memories may not be feasible. The use of thermoplastics, for example, requires thermal deformation of the film, and fatigue problems have been shown after only a few thousand erase/write cycles.

The storage capacity of a fixed medium holographic memory is also limited by the size of the optics required. Capacities beyond 10^8 bits will most likely require precision lenses over 1/2 meter in diameter. This will adversely affect the cost, size, and weight parameters of the memory system.

The limited lifetime of the laser and the elaborate optics required also present potential reliability problems (reference R9).

Conclusions. –

- Laser memories are limited to 10^8 to 10^9 bits, if fast access (1-5 μsec) is to achieved.
- Read/write laser memories depend upon development of a suitable storage medium.
- Reliability and cost of high-powered lasers may limit the use of holographic laser memories even if a suitable medium is found.
- EBAMS appear to be a better choice.

Josephson Memory Devices

Experimental superconducting Josephson devices have been shown to operate at very high speeds and have ultimate density limits that exceed present day photolithographic capabilities.

Figure R4 shows a practical NDRO memory cell for a bit-organized RAM. Persistent circulating currents ($I_W/2$) in clockwise or counterclockwise directions represent the data stored as a logic "one" or "zero", respectively. When an external current I_W is applied,

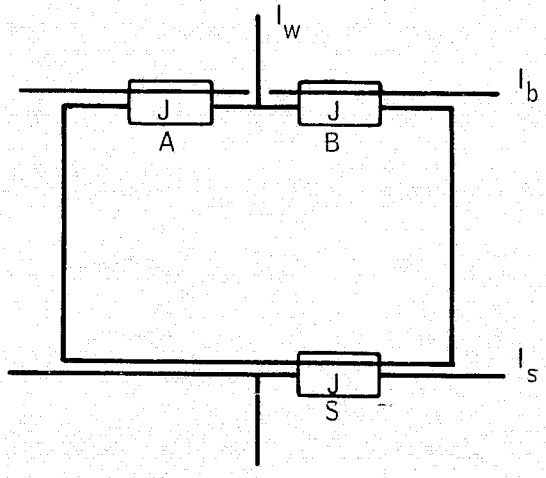


Figure R4. NDRO Random Access Memory Cell

it splits equally into both branches and is superimposed on the existing circulating current. Thus, I_w will flow through one branch and no current will flow through the other. As long as the loop remains superconducting, the original current is restored when the external I_w is switched off. The direction of circulating current can thus be detected by applying external currents I_w and I_s. I_s causes devices to switch only when the control current exceeds $I_w/2$. Data is written by coincident word current I_w and bit current I_b, which rerouted the work current to the alternate branch.

With line-width limits of two microns, these cells have been fabricated in 1.4 mil^2 and have switched in less than 100 psec. It has been shown that 5×10^8 read operations do not disturb the data contents.

DRO memories can be fabricated using a single device per cell and even higher bit densities than the NDRO cells.

Like semiconductor memories, Josephson devices allow the fabrication of decoders, drivers, and sense circuitry of the same technology and on the same substrate as the memory cells. Thus, with these devices, large-scale integration is feasible.

The main obstacle to the Josephson device development is the low reproducability of the extremely thin and defect-free oxide tunnel barriers. Encouraging improvements are being made however.

With present day photolithographic limits, a 4K bit NDRO RAM with a 2.5 nsec cycle time could be fabricated with a bit density of 2×10^5 bits/in^2; a 16K bit DRO cell could be fabricated with 20 to 30 nsec cycle time and 6.5×10^5 bits/in^2. These estimates assumed that the technology was developed and ready for manufacture. If line-width resolution could be improved as a result of uv, e-beam, or x-ray exposure techniques, density and speed would improve (reference R10).

Appendix R

R1. Hodges, D.; Alternative Component Technologies for Advanced Memory Systems, Computer, September 1973, pp. 35-37.

R2. Cohen, M. and Moore, J.; Physics of the MOS Electron-beam Memory, Journal of Applied Physics, December 1974, pp. 5335-5348

R3. Kelly, J.; The Development of an Experimental Electron Beam Addressed Memory Module, Computer, February 1975, pp. 32-42.

R4. Martin, R. and Franekl, H.; Electronic Disks in the 1980s, Computer, February 1975, pp. 24-30.

R5. Hughes, W., et al.; BEAMOS - A New Electronic Digital Memory, Proceedings, National Computer Conference, May 1975, pp. 541-548.

R6. Butter, C.D. and Lee, T.C.; Thermoplastic Holographic Recording of Binary Patterns in PLZT Line Composer, IEEE Transactions on Computers, April 1975, pp. 402-406.

R7. Roberts, H., et al.; High Speed Holographic Digital Recorder, IEEE Transactions on Applied Optics, April 1974, pp. 841-856.

R8. Williams, J., Stevens, J.; Research Guidelines for Laser Memories, System Development Corporation document TM-HU-201/000/00, 5 June 1975.

R9. Little, A.D., Inc.; Technology and Cost Forecasts for Electronic Data Processing Hardware, Software, and Data Communications Networks, Contract No. F19628-74-C-0093, 15 February 1974.

R10. Anacker, W.; Superconducting Memories Employing Josephson Devices, Proceedings, National Computer Conference, May 1975, pp. 529-534.