JPL PUBLICATION 77-6

Report on Phase I of the Microprocessor Seminar Held at Caltech, October 1976

(NASA-CR-152678) REPORT ON PHASE 1 OF THE MICROPROCESSOR SEMINAR (Jet Propulsion Lab.) 130 p HC A07/MF A01 CSCI 09B

N77-22833

Unclas

G3/60 25143

National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California 91103

JPL PUBLICATION 77-6

Report on Phase I of the Microprocessor Seminar Held at Caltech, October 1976

April 1, 1977

National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California 91103

PREFACE

The work described herein was performed by the Information Systems Divisicn of the Jet Propulsion Laboratory.

ABSTRACT

This report documents a seminar and workshop series on microprocessors and associated large-scale integrated (LSI) circuits attended by NASA center and other government agency representatives. The potential for commonality of device requirements among the agencies was assessed. Candidate processes and mechanisms for qualifying candidate LSI technologies for high reliability applications were reviewed. Specifications for testing and testability were discussed. Various programs and tentative plans of the participating organizations in the development of hi rel LSI are presented.

CHAIRMAN'S INTRODUCTION

Robert E. Covey
Jet Propulsion Laboratory

There are several attributes of microprocessors and associated large-scale integrated (LSI) circuit components that make such components extremely attractive for use in NASA spacecraft designs. The use of LSI should result in more reliable, less costly spacecraft designs with better performance. However, if the use of microprocessors and LSI throughout NASA is to be accomplished in a cost-effective manner, some standardization of microprocessor and LSI components is required to avoid costly duplication of effort in qualifying a microprocessor and associated components for space application.

This conference/workshop is the first step in establishing commonality of requirements for microprocessors and other LSI. As the first of two planned meetings it has the following four objectives:

(1) To establish lines of communications among the interested NASA centers and other agencies.

(2) To assess the potential for commonality of LSI requirements among the agencies.

(3) To review both established and new approaches to LSI qualifications.

(4) To identify those manufacturers who might be interested in producing hi-rel LSI devices.

Your attendance is a demonstration of your interest in these objectives. I hope that our meeting plans and format will be conducive to the level of interaction that must occur if our goals are to be accomplished.

My thanks to the NASA Low Cost Systems Office for providing the funds to sponsor this conference, and to each and every participant upon whom the success of this conference hinges. Thank you for sharing your knowledge and expertise. With your help I am sure it will be a worthwhile and productive meeting.

WELCOMING REMARKS

Fred Felberg
Jet Propulsion Laboratory

Good morning and welcome. For those of you who are here from outside of JPL, let me extend a warm welcome on the part of JPL and Caltech to this workshop. For those of you who are from JPL, I do not think you really need any welcome, but perhaps a reassurance, at least on the part of the Laboratory Director, Dr. Murray, and myself as to the importance of what you are doing may be in order.

I would like to remind you that the workshop is being sponsored by the Low Cost Systems Office of NASA. The Low Cost Systems Office was established some years ago, having the responsibility within NASA for fostering and sponsoring programs which would have the effect of substantially reducing the cost of future space missions, primarily through the avenue of standardization in procedures in operational systems, and in some instances, in actual hardware and software systems. The program is relatively new with respect to our ability to assess its ultimate result, but I think that the promise is high. The area of components, electronic components in particular, has been one in which over the years we have experienced a certain amount of difficulty, and an area which seems to hold promise for providing benefits through standardization, so I think it is natural for the Low Cost Systems Office to be interested in sponsoring exactly this kind of a workshop.

I do not think I have to remind you further about the importance of the challenge that all of you are addressing. It is an extremely complex and difficult problem to try to achieve the combination of reliability, performance and low cost in the kinds of systems where microprocessors are going to be used extensively. The best thinking we can possibly bring to bear on this difficult problem is going to be well worthwhile. I wish you well and hope you have a very enjoyable and productive session.

SEMINAR PROGRAM

WEDNESDAY, OCTOBER 27

8:00 a.m.	Registration
8:35	Announcements - Robert Covey, JPL
8:40	Welcoming Remarks - Fred Felberg, JPL
8:45	NASA HQ Low Cost System Office Program - William McInnis, NASA HQ
9:00	Workshop Background and Objectives - Paul Lecoq, JPL
9:30	Unified Data System - Mike Ebersole, JPL
10:40	Coffee Break
11:00	Assessment of Potential Market for Hi Rel LSI - Ralph Martinez, NELC, San Diego, CA
12:00 p.m.	LSI Standardization - David Haratz, Army Electronics, Ft. Monmouth, NJ
12:30	Lunch
1:30	Nonstandard LSI Approach - Carver Mead, Caltech
2:40	LSI in the Air Force - John DeCaire, AFAL, Wright Patterson Air Force Base, Ohio
3:40	Coffee Break
4:00	Application of Diagnostics and Test Chips for Predicting Reliability of LSI - Joseph Maserjian, JPL
4:25	Summary LSI Requirement Forecasts - Richard Scott, JPL
5:05	Non-User Perspective and Linear Device - Sam Davis, Electronic Engineering Times
5:30-7:00	Social Hour - Athenaeum

THURSDAY, OCTOBER 28

Presentations by Working Group Chairmen outlining issues and topics to be addressed by individual working groups.

8:45 a.m. E. C. Urban

8:50 R. Conklin

8:55 R. Scott

9:05 Meeting of Working Groups:

Working Group A
Chairman: E. C. Urban, NELC
Subject: Commonality of Requirements and Potential for Standardization

Key Issues:

- What commonality of requirements exists between the hi rel LSI users?
- Can some standardization be applied to limit the total number of types of LSI components that must be qualified for hi rel application?
- Can standard specifications satisfy most user needs?

Working Group B
Chairman: Robert Conklin, AFAL
Subject: LSI Qualification Mechanisms
Key Issues:

- What are the implications of various qualification-related parameters to the candidate LSI technologies?
- What are the steps involved in the qualification process and how do they vary as a function of the candidate LSI technologies?

Working Group C
Chairman: Richard Scott, JPL
Subject: Testing Microprocessors and Other LSI
Key Issues:

- How can a "system-on-a-chip" be tested?
- Which potential LSI users are involved in LSI testing?
- Can testability be designed into LSI before manufacture?
- How can tests be specified for LSI?

12:00 p.m. Lunch

1:00 Working Group Meeting continued/preparation of summary

4:45-6:15 Viking Tour at JPL.

FRIDAY, OCTOBER 29

8:45 a.m.	Announcements
	Summary of Working Group Discussions
8:50	Group A Report and Discussion
9:15	Group B Report and Discussion
10:15	Coffee Break
10:45	Group C Report and Discussion
11:40	Summary by Moderator
12:00	Adjournment
12:00-12:30	Planning for Joint User/Manufacturer Workshop (Executive Session of Workshop Organizers/Working Group Chairman only)

CONTENTS

APPENDIXES

NOTE

Editing of this report for the Microprocessor Conference took longer than we expected owing to the large number of people involved, their wide geographic distribution, and the number of series operations required. The general session on the first day of the meeting was tape-recorded and the tapes transcribed subsequent to the meeting. The transcribed tapes were edited by the four editors and submitted to the numerous speakers for their approval and summarization. This report is the collection of these summarizations and certain of the visual material which was presented. Some of the discussion material is presented in its entirety.

Our intention was to reduce the 31 hours (10-3/4 hours of general sessions and three concurrent 6-3/4-hour workshop sessions) of presentations and discussion to a readable report without losing the essence of the total meeting. We hope we have been reasonably successful in producing a useful report. Our thanks to all of the authors, moderators, and session participants, and particularly to Jerry Taylor, the Administrative Chairman, who did so much to pull together both the conference and this report.

Robert Covey
Mike Ebersole
Paul Lecoq
Dick Scott

77-6

SECTION I

NASA HEADQUARTERS LOW COST SYSTEMS OFFICE PROGRAM

William McInnis
National Aeronautics and Space Administration

A. BACKGROUND

Bill McInnis is Manager for Communication and Data Handling in the Low Cost Systems Office in NASA Headquarters. He was formerly at the Kennedy Space Center. As a member of the Special Projects Staff of the Launch Processing Systems Division at KSC he was responsible for a variety of projects such as Space Shuttle, Automatic Landing System, Thunderstorm Predictions Systems and Weather Radar, and more recently, the conceptual in-system design for the Launch Processing System. Prior to joining NASA, he was an inertial guidance engineer for the guidance system on both the Apollo Command Module and the Lunar Module with AC Electronics and Grumman Aircraft. He received his Bachelor of Electrical Engineering from the University of Florida in 1962, was active in IEEE, and is the immediate past section chairman for the Canaveral section at KSC.

B. PRESENTATION

I probably will not succeed in holding the "commercial" to the 60-second break, but I will try to get close to it anyway. There are a couple of remarks that I would like to make to shed a little light on why we are here. As was mentioned, one of the primary activities of the Low Cost Systems Office is trying to develop standard components. Although we have only been at this game for two or three years, I think the record is beginning to show that we are now entering an era in which some of these efforts are starting to pay off. We have, for example, in our stable of standard components, a transponder. It has three versions; a STADAN version, a deep space version, and, for the future, a TDRS version. We have two standard tape recorders, one with a capacity of 10^8 bits, one with a capacity of 10^9 bits. A version of the 10^9 bits tape recorder is being used on the space shuttle. We have a standard computer with a second standard computer of a somewhat larger size on the way. One of the new programs we have just gotten started on is a standard 10^7 bubble-memory storage device. In areas other than communication and data handling we have a standard 0.2-pound thruster, and a standard power initiator, and we have some other things that are starting to bubble or come to fruition in the propulsion area. In the power area we have standard 20-amp batteries. We have had a very long effort now in trying to develop a standard specification for solar cells. Although I suspect the market dynamics in solar cells are considerably different than in LSI, there is some commonality of problems. So we do have a pretty good stable of standard subsystems. I have not mentioned them all by any means; those were just some of the more successful ones.

The payoff is quite real; we estimate that our standard tape recorder program has saved NASA about $26 million; the standard transponder perhaps a little bit more than that now. I had two more customers call me just before I left last week, so it is clear that we are doing things in a way that is working. We would like to think that, growing out of the efforts of you people here, at some future time we will also have a similar stable of standard LSI components.

In our office we have five hardware-oriented panels: Communications and Data Handling, Stabilization Guidance and Control, Auxiliary Propulsion, Electric Power, and Ground Support Equipment. At the present I have three of the five due to departure of some of our staff, so I am spread pretty thin. I hope that is going to be corrected, perhaps by the time I get back.

The way the panels work is that we have a man at NASA Headquarters and a panel which is comprised of representatives from each of the NASA Centers and from other government agencies. The Air Force is specifically represented on each of the panels and we are finding that interface with the Air Force to work quite well.

I would like to mention something that happened last Saturday. I attended the annual meeting for a non-profit organization which was formed a few years back by amateur radio operators to design and build amateur radio satellites. At the annual meeting the plan for the new satellite, the Phase Three Spacecraft, was discussed, and lo and behold one of the things that is causing them concern is their design for an integrated housekeeping unit which employs a microprocessor. The unit works fine. The problem is they can no longer get delivery on it. The design of the microprocessor is unique; you cannot simply unplug one chip and plug in another manufactured chip. They are facing the same kind of problems that we in the government are facing in our designs and in our planning for future spacecraft. I mentioned to the project manager that I was going to be at this workshop this week and he said he wished he could have attended. So I'll get back with him later and let him know if we made any progress. I think perhaps that is a capsule version of the same thing we have been facing. We feel that if we can find some common requirements, settle out on a few microprocessors and other LSI, then maybe we can afford to get them flight-qualified and use them.

SECTION II

WORKSHOP BACKGROUND AND OBJECTIVES

Paul Lecoq
Jet Propulsion Laboratory

JPL has the responsibility to fly planetary missions, and we need qualified LSI to fly them. The purpose of this meeting is to explore the process of qualifying these systems-on-a-chip. We are no longer in the driver's seat, commercial sales are. I would like to challenge all people here to look for any commonality that can be exploited so we do not each have to pay for all the qualification we need. Can we do something about the qualifiability, the reliability, the testability of LSI devices?

We at JPL have faced up to the future challenge just as most of you have. LSI will be required to fly more complex missions. It can also reduce total costs if we do our jobs right because testing and operations as well as design can be less expensive with LSI. The problem is, we cannot do it effectively alone. Many of us here have similar needs, similar hi rel requirements. This commonality in application of LSI is more striking than any differences. Many manufacturers are really interested in talking to us. If we can capitalize on the common requirements our negotiations may go surprisingly well.

Standardization can be a dirty word. It can stifle progress, locking us into obsolete technology. It can also simplify things and reduce costs if we choose the right standards. The standardization of 74/5400 TTL boosted the industry tremendously and did not stifle anything. What it did was to define the prosaic circuit designs and allow engineers to worry bigger, more esoteric problems. It was a de facto standardization. Everyone just recognized TTL was the industry standard so the declaration was easy. We have to do the same with LSI. Recognize what is widely used, then make that usable in hi rel applications.

At the same time we are declaring de facto LSI standards for the near term we will have to look to the future. Defining our future needs now will make the qualification of resulting future technology easier. The sessions at this meeting are not lectures. They have to be interactive discussions. We are trying to open lines of communications and ultimately through these lines to exploit our common needs to get more effectively qualified LSI systems. The point is, I think, microprocessor and LSI are here. They are the neatest things since sliced bread and we have got to find a way to use them effectively. That, in a nut shell, is what it is all about.

The basic job of JPL is to fly missions to outer planets for whatever scientific value we can get out of them and to do that at as low a cost as possible. Our budgets have been shrinking both in actual numbers and the real dollars with inflation, so we are faced with the option of flying more missions and more sophisticated missions with either less money or the same money. We will be trying to do it effectively and to increase the return of data from mission to mission without any net increase in cost.

We have evolved a spacecraft system which requires intelligent subsystems spread about the spacecraft. This allows us to divide the spacecraft into functional blocks that can be handled. We can partition the job in such a way that each job can be handled by one or just a few individuals and that the interconnections of the boxes can also be understood. We have to control the complexity of the spacecraft so we can be more cost-effective than we have been in the past. There are a few problems: LSI is required, but can we really qualify it? Can we really fly at a reasonable cost? Can we afford to treat the LSI that we need like the 54L, the TTL stuff that we have used in the past? That is the big question. It is the same problem you face.

The manufacturers are not particularly interested in talking to us; we can't really influence the way they do things on their line. They can say to us, "We're successful with what we've got; we're not going to let you fool around with it because we're making a profit." Can we really even use these standard LSI devices for flight applications. There are power limitations, reliability, the untestability of some of them or all of them. The basic thing that gets us in trouble though is that we have some very special requirements that are different from everybody else's; they are different from the Air Force, Navy AMSAT, Army, GM. They are all different, right? And they have to be different, right? Well, that is one of the questions we want to talk about today and tomorrow and the next day. Why do they have to be different? And if they do have to be different let us understand the areas where they really need to be.

There was a meeting in Cupertino of manufacturers. They complained about MIL 38510 specs and are asking for a little more cooperation on hi rel parts. It turns out that the manufacturers are not as disinterested in what we are doing as we had originally thought, and that they are working in several areas to produce devices that we can use.

We need to work on the answer to these basic problems that we have outlined this morning. We ought to be learning from each other; we ought to be using each other's output, and we ought to be on the lookout for anything that is going to save us money collectively. It is easier to cooperate when you do not have enough money than it is when you have as much as you might ever want or need. So we are well motivated to cooperate. And again, that is why the conference.

Now the first objective to open communication is probably going to be the easiest to manage. We are all together; we are going to be talking together, working together and discussing our common problems and common needs. I will expect you to search out your counterparts and other agencies and talk to them, take their telephone numbers and maybe call them periodically or visit each other's facility to make sure that whatever commonality we have amongst us we can take advantage of.

The second objective is going to be a little bit harder. We want to work toward the common requirements for LSI. I really feel that there is not all that much difference between the needs of the various agencies. Certainly a tank does not have the power limitations that we have. David Haratz pointed that out to me very explicitly. An airplane does not have the same kind of constraints we have, and certainly a ship does not have the volume limitations that we have. But there are a lot of common threads, the components that run through all these systems that are going to be standardized. The program that the Navy's been sponsoring has attacked the standardization and commonality problem rather well on a module level. The lesson that they have learned at that level can be applied to the lower component level with benefits for all.

We would like to work in one area where there is definite commonality. That is the actual qualification process. We need to see what the areas are of commonality. Do we need "A" level, "B" level, "A+," "B-" level devices? What in the qualification process is applicable to LSI that might be different from the way we have treated previous devices?

The last objective is going to be kind of difficult but it is very important. If each of us goes in with an order for 20 parts or a hundred parts, or even a thousand parts, to our own specs with our own kind of English on them, our own applications and that sort of thing, the manufacturers are not going to be particularly interested in talking to us. But we can come up with some common specs that allow us to go in and say, "I would like a block order of this standard that the Air Force and the Navy and the Army and ERDA and JPL, and Goddard, and all these other people order to with maybe a slash sheet because we need minus 60°C centigrade," God forbid. I think the manufacturers will be much more interested in talking to us and much more interested in doing our work for us without charging us an arm and a leg. Does anyone object to what I have said?

Here is the basic organization. We are going to follow this talk with several off-the-cuff informal discussions of where LSI stands at various areas in the government. Then the sessions chairmen are going to stand up in front of everybody and tell them what their session is all about. We are going to divide off and work together and try to discuss potentially interesting areas. Each of the groups will give an oral report back to the whole body here, and then we are going to write a written report afterwards that is going to try and sum up where we stand as users.

At that manufacturers' conference they said something to the effect that if those who generated the specs would have consulted with the manufacturers, costs would be lower. We are going to have to do that. And we are proposing that a follow-on to this meeting be another meeting, with the manufacturers invited, where we can hash out in great detail what they can do for us, what we can do for them, how we can cooperate and get the best parts that might be available. That is the basis of the conference here.

Group "A", with Dick Urban, is going to talk commonality and standardization. He is from Naval Electronics Center in San Diego. I would like to ask these questions: What commonality exists and can it be exploited? There are going to be areas that are not going to be common, but we ought to understand which are common and which are not. Is there much potential for standardization at the module level? Perhaps. At the parts level? Certainly. Can it save money and will it stifle progress? Certainly standardization can stifle progress if it is supplied slavishly and without considerable thought. Hopefully we can be a little smarter than that. But that is a good point not to forget in the discussion of standardization.

Can we really have specs that are standard and can they apply across the board to more than just one organization? Do we really need the restrictive things in the qualification specs we have now? Can we relax some of those specs? Are all the requirements that we placed on the manufacturers in the past really necessary? Do they really accomplish something? Or are they just there by force of habit. Can we really treat custom LSI? We have a good backlog of data on commercial devices. Millions of units in operations throughout the world. That is good for commercial LSI but what do we do about custom LSI, things we build for ourselves? How about semicustom, the standard logic case? Can we support those kinds of things? Can we qualify the basic gate array and then just wire them up with aluminum deposition to make the devices we need and expect them to have reproducible characteristics. Do we need custom LSI? That is a big question. Can we restrict the total number of parts so that we do not have to qualify the whole world's LSI?

For Group "B". Some more challenges. Can we really qualify these things realistically? They have to be treated differently, but just how differently? Maybe we can take advantage of the fact that we have a new system and come up with new specs that might make our systems a little bit cheaper than they have been in the past. Can qualified standard specs be developed and really used? I know we can develop them easily, but can more than one organization actually use them? What are the impacts of parameters from line to line? What do the differences mean? How do different manufacturers treat things differently? Does packaging make any difference? Can you treat a hybrid package any differently or the same as you would treat a DIP or a flat pack?

These devices that we have now were basically built by taking TTL designs and just transferring them onto a single silicon chip without an awful lot of thought as to how they were going to be tested. There are gates and flip-flops and all kinds of things inside those devices that simply cannot be tested, and there is just no way of getting around that. We have to come to grips with how to qualify something we cannot test. Let's take a little time during the sessions to talk about how we could design them in advance so that we can test them. It would make life easier if we could use fault-tolerant techniques also, if we designed them in such a way to detect the failure and to switch in a backup unit to get total system reliability high. Again, what is appropriate for LSI in particular?

Group "C." More challenges. Can they be tested adequately? Of course the answer is no. What do we do about it? Is the answer no? Is there anybody here that says that LSI devices can be tested adequately? Somebody made a comment a while ago that memories, like amoebae, tend to build up an immunity to whatever particular test they are subjected to. Can testability be designed in? There is a school of thought which says let us not test the heck out of these devices? Let us not grind ourselves into the ground in trying to test something that cannot be tested in the first place. Just have a lot of redundancy so that if one fails, we just switch in the next one and if that fails we switch in the next one. I am not sure what that really means to cost and understandability of the system. I think it deserves some consideration in the meeting. Can good, standard, reliable tests be developed for LSI? I am sure Dick and Bob and Dick Urban are going to have their own questions to start off their own sessions; I would just like to add these things on to continue the discussion.

We are not going to have a bunch of lectures, but general discussions. We would like to have a full, free interchange between everybody. We would like to have everybody get to know each other and get to know where you stand; where you are coming from; where you think you are going. So that we can recognize whatever areas of commonality there are. If somebody has already invented something that does the job that you want to do, do you really have to reinvent it? Each group should take a good set of notes. We are going to have to put out a report on this workshop afterwards and we would like to have something to base that on. We need a good report to the general audience so that we can all have a better discussion.

We have got to decide what we do after this workshop. Have we done everything? Is everybody happy with the situation? Or should we be working together in the future to take advantage of the commonality to save money by helping each other out. What will we take to the manufacturers? Do we take to them the understanding that we have 60 different organizations doing 60 different jobs, to 60 different specs, or do we have one spec or maybe two specs with maybe a few little variations here and there on a theme? Are they going to be interested, or are they not going to be interested? That is the basic question.

We would like to see to it that somewhere in the discussion we talk about testing versus redundancy, testing versus testability. What is the alternative to exhaustive testing of devices? There is a point for using standard regular structures versus random logic. I would defy anybody here to figure out some easy way to tell whether 8080 is really working or not. Maybe in some subsequent generation microprocessor we can work together with the manufacturers to bring out a device that handles things fairly straightforwardly by matrix operations and by a little more regular architecture, so that we can use coding techniques to determine whether it is really working or not. We have done a lot of work in this area, but it has been a long time ago, with the Star computer. Building a machine that is self-testing, maybe that is a good direction to go in the future.

Do we want to qualify 6800's, and 8080's, and 6100's, and 1801's and 1802's and ATMAC's and God only knows what all else? Or could we take the approach of having a microcodable microprocessor that we could code, like a chameleon to change its colors to look like a 6800. If some organization has been using 6800's and likes them, and they feel comfortable with them and they would feel terrible if they had to give them up, maybe we could microcode. I don't know if that is a good idea or not. I think Bob Conklin and John DeCaire will have a few words to say about that.

We would like to apply fault-tolerance techniques and whatever other techniques we can to make these devices as reliable as possible as a system, but there are failure mechanisms that say when one failure occurs in a chip, that chip is probably bad. To build in redundancy on the same chip probably would not be very successful. I do not know. I think it is something interesting to discuss and to look into when we are talking about where we are going in the future.

The last comment really does not need much explanation. I am not going to tell you where we stand, but we are ready to cooperate.

SECTION III

UNIFIED DATA SYSTEM

M. Ebersole
Jet Propulsion Laboratory

A. SUMMARY

Mr. Ebersole is a project engineer responsible for the design and development of advanced digital data systems in JPL's Information Systems Division. He is currently working with the NASA Low Cost Systems Office to develop a microprocessor-based module with NASA-wide application as a standard building block for spacecraft instruments.

Mr. Ebersole began his talk by describing the present situations in NASA relative to the use of microprocessors and other LSI. He stressed that experimenters were extremely interested in using microprocessors in their designs and that NASA must employ microprocessors and other LSI in an organized way in order to avoid very costly duplication of effort. Cooperation with the DoD to qualify microprocessors/LSI for space and missile application was also stressed.

The evolution of spacecraft data handling and control systems was traced through three eras of technology: 1970-1975, 1975-1980, and forecast for 1980 and beyond. The plan for the data systems in the 1980's calls for a network of distributed microcomputers, interconnected by a data bus, to perform the data handling and control function. The characteristics and benefits over previous systems using the distributed microcomputer approach were described.

The Remote Terminal Unit (RTU), a proposed NASA standard microprocessor based module which would be the basic building block of a distributed system, was discussed in great detail. The elements of an RTU are a microprocessor, ROM and RAM, bus interface, and standard I/O components. The use of the RTU for a specific spacecraft mission, the Jupiter Orbiter Probe (JOP), was described. The JOP mission characteristics were used to emphasize some of the key requirements that would impact LSI selection and qualification. Among the important requirements were spacecraft operation for a period of 5 years, low power and volume, mil temperature range, and radiation hardness to 10^6 rad (Si) total dose.

A summary of microprocessor unit (MPU), memory, input/output, and environmental and reliability requirements was presented. The summary indicated that an 8-bit, fixed-word-length MPU with interrupt and DMA capability would satisfy the RTU requirement. The need for ROM, PROM, and RAM-type memory was indicated. A comprehensive family of input/output devices is required including programmable serial and parallel devices and A/D and D/A converters. Special emphasis was placed on the need for a serial bus interface device compatible with the MIL STD 1553A bus.

Mr. Ebersole concluded his talk with a presentation of the development schedule for the RTU. A two-phase development is planned: Phase 1, to be accomplished in FY77, would include microprocessor and associated LSI selection and the electrical design of the RTU. The key milestone is the microprocessor selection, which is scheduled for May 1, 1977. Phase II, which is to be accomplished in FY78, would include the packaging design, fabrication and qualification of a flight model of an RTU.

B. PRESENTATION

I have broadened the scope of my talk to try to indicate the situation in NASA relative to the use of LSI for spacecraft design, rather than just trying to touch on a JPL activity. I should point out that we are just now starting, under the sponsorship of the Low Cost Office, to try to gather together the requirements from the various NASA centers in order to establish some commonality of requirements for LSI. The mechanics of this process are that there will be an inter-center working group established to try to assess the potential for LSI. In order to give you a picture of the trend toward the need for and the use of LSI in the spacecraft design, I am going to discuss a kind of subset of the spacecraft design, i.e., the data handling and control function. And to give you an indication of the trend toward LSI usage, I will trace through some examples of the types of technology that are employed in such systems in the era 1970-1975, and then 1975-1980, the era in which we are trying to employ standards throughout the various NASA centers for such systems. Then I will indicate how such standards should be augmented to include LSI as basic building blocks in such designs as a means of trying to get a handle on the LSI microprocessor question.

The Low Cost Systems Office is currently considering sponsoring development of a Remote Terminal Unit (RTU), a microprocessor-based module that can be used over a broad spectrum of applications throughout NASA spacecraft, both to augment existing designs and to be used as basic building blocks for distributed networks, which is the trend for future spacecraft design. As a specific example of how such building blocks might be used for spacecraft design I have chosen the Jupiter Orbiter Probe mission, which represents a considerable set of rather stringent requirements, both in terms of the reliability and the selection of LSI devices for space applications.

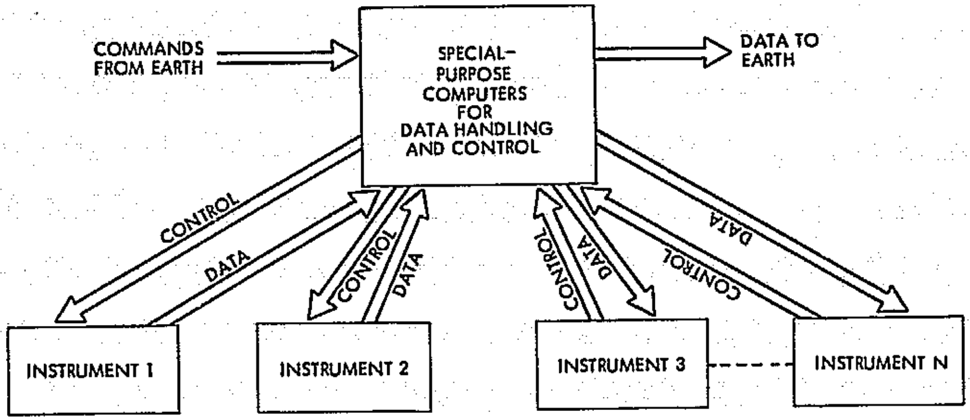
Then I would like to discuss the kinds of requirements we foresee and characteristics we think we need in terms of microprocessor and other LSI for the particular applications under study. I will talk briefly about environmental and quality assurance requirements and the schedule for development of such devices: when we need to make a selection, and when we need to buy. I will discuss who our customers are and preparation for future designs.

Within NASA, the use of microprocessors and other LSI is certainly on the horizon for use in space applications. The time to coordinate the use of LSI for space technology is certainly right. The designers of systems are demanding that they be able to use such devices in their designs and such devices are showing up as integral parts of NASA standard proposals already, both in the design of data systems and as integral microcontrollers as part of star trackers and as special LSI designs in some of the systems that Bill McInnis discussed earlier. So NASA must employ microprocessors and other LSI in an organized and standard way to avoid a very costly duplication of effort. Rather than qualifying a variety of different microprocessor types, we should limit the number of types of devices that we can qualify to meet a broad spectrum of applications. We should cooperate with the DoD to qualify microprocessors and LSI for space and missile application, if in fact common requirements exist. Through our contacts with the Air Force, Navy and Army for other efforts, one being radiation, we have ascertained that in fact such commonality of requirements really does exist and we would like to exploit this as much as we possibly can.

Now I'd like to trace the evolution of NASA spacecraft data handling and control systems from the 1970's through the forecast for the 80's. First, I will describe what the spacecraft data handling and control problem is and the solution to the problem in three different time phases. I will briefly go over the benefits of the use of microprocessors in spacecraft designs.

The spacecraft data handling and control problem is really part of the class of real-time process control problems. A subset of that class is the command and control function. The data handling and control system aboard a spacecraft issues control commands to the various spacecraft subsystems as determined by some mission profile. Commands can be either issued directly from the ground or can be stored on board in some kind of computing device to be issued as a function of time aboard the spacecraft. Also, that particular function is responsible for responding to various spacecraft emergencies. The data handling function is one of acquiring, formating, and coding the telemetry data for subsequent transmission to the ground or for storage on board a spacecraft storage device which, for us, has been a tape recorder that provides the bulk storage media.

The data handling and control systems in the era 1970-1975 (Figure 3-1) are characterized by special-purpose computer designs to perform the data handling and control functions. Those functions accept commands from the ground and issue them to the various instruments or subsystems principally by private wiring to route commands to the various instruments. The special-purpose computers are characterized by the use of small-scale and medium-scale circuit components. The gathering of data and the control of instruments is accomplished by the central device, again by private wiring that exists for the collection of data from the various instruments. The central device in this case is a very busy sort of device with hundreds of wires leading from it to accomplish its various functions.



SPACECRAFT DATA HANDLING AND CONTROL CHARACTERISTICS:

SPECIAL-PURPOSE COMPUTER DESIGN
SMALL-SCALE/MEDIUM-SCALE INTEGRATED (SSI/MSI) CIRCUIT COMPONENTS
CENTRALIZED INSTRUMENT CONTROL AND DATA GATHERING
PRIVATE WIRING FROM COMPUTER TO EACH INSTRUMENT

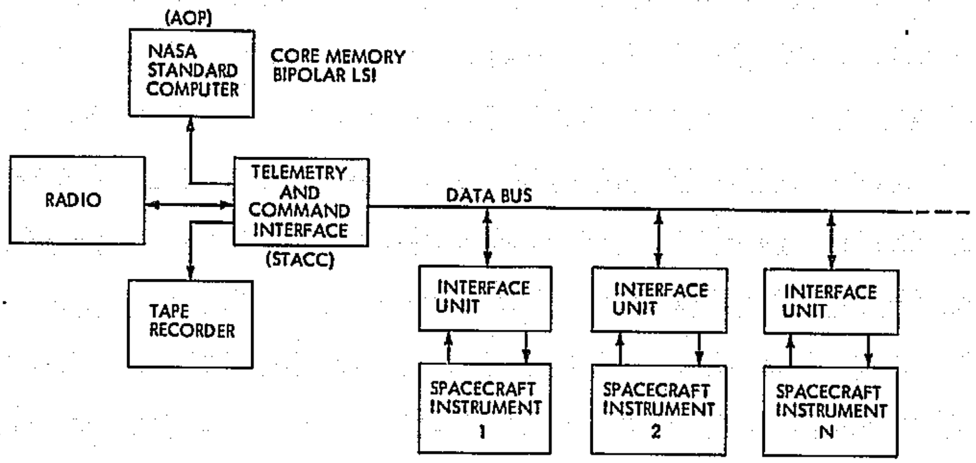
Figure 3-1. Solution to Data Handling and Control Problem (1970-1975)

In the present era, NASA is introducing standard subsystems in order to accomplish the data handling and control function (Figure 3-2). One standard subsystem is the NASA standard computer, the Advanced On-Board Processor (AOP). The AOP design is characterized by the use of core memory and uses special bipolar LSI circuits to implement its logic functions. A data bus is used for intercommunication between subsystem elements. The telemetry and command function is performed by the standard telemetry and command components, which do command decoding and telemetry encoding functions. At present, this particular hardware is out for bid. The technical proposals have been received and are now being evaluated, so the use of LSI as part of this design is not publicly known.

At any rate, in the present era we are starting to see LSI devices creep into the designs. The system is still characterized by centralized instrument control and data gathering under the control of either the NASA standard computer or via commands received from the ground that are routed through this STACC equipment.

Question: What are the characteristics of the STACC bus?

The STACC bus is a serial data bus somewhat different than the 1553 bus. It is a dual bus system that uses a supervisory and reply line and Manchester coding techniques like the 1553. The STACC bus is the standard bus for the NASA multi-mission spacecraft.



SPACECRAFT DATA HANDLING AND CONTROL CHARACTERISTICS

USE OF STANDARD BUILDING BLOCK ELEMENTS
CENTRALIZED INSTRUMENT CONTROL AND DATA GATHERING
INTRODUCTION OF BUS FOR INTERCOMMUNICATION BETWEEN SUBSYSTEMS
INTRODUCTION OF SOME LSI

Figure 3-2. NASA Standard Data Handling and Control System for Multimission Spacecraft (1975-1980)

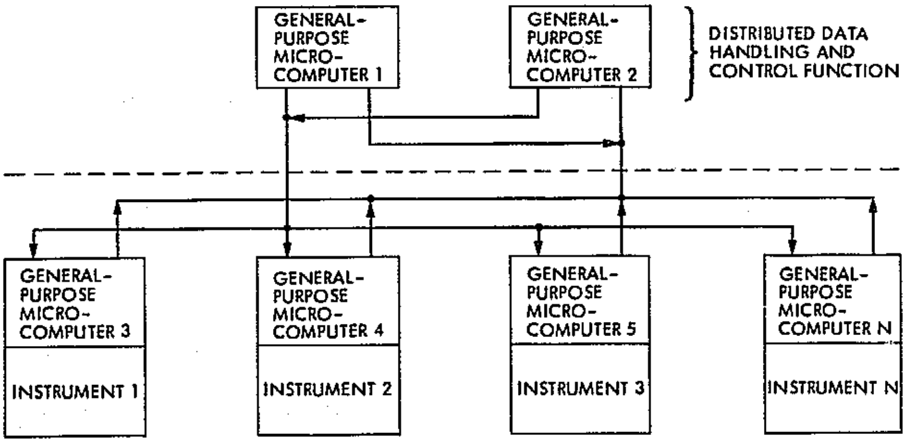
Question: The NASA computer, what is that?

The NASA computer is a general-purpose device designed by Goddard and now being fabricated by IBM. It is characterized by the use of core memories, and it is constructed from some bipolar LSI devices that have been developed by TRW, custom metallized arrays. Another characteristic of this particular standard system is that there are standard interface units which interface the data bus with the spacecraft instrument. In this particular case, these interface units are what might be called "dumb" terminals. They have discrete output and input interfaces, and any computational capabilities, sequencing, or control capabilities are left to the discretion of the instrument or the spacecraft subsystem designer. The other element of this standard system is the tape recorder, which is a NASA standard device also.

The solution to the data handling and control problem for the 1980 and subsequent era (Figure 3-3) is a distributed network of general-purpose microcomputers to perform the data handling and control function. The network is characterized by the use of general-purpose devices rather than special-purpose devices and uses large-scale integrated circuitry technology for the components of the microcomputer. In this particular case, we are advocating decentralized instrument control and data gathering. Each instrumenter uses a microcomputer as an integral part of his particular instrument or subsystem for controlling his own instrument, rather than having it done in a centralized way, and also for gathering and buffering and storing his own data. The interconnection between the modules is via data bus. For this particular concept we are evaluating different types of buses. Although the one standard bus exists for the standard telemetry and command components, we are also evaluating the MIL Standard 1553 bus as a possible bus for use in this application. For the Shuttle Orbiter, a bus similar to the 1553 is used. So, for any given spacecraft launch one might see two or three different kinds of buses used.

There are several attributes of the type of system that we propose for spacecraft launched in the 80's and subsequent, and primary among those is the reduction of mission cost. For future missions, particularly at JPL, we are very constrained dollarwise, and we have to conceive of ways to do a better engineering job in producing spacecraft and the related support functions for less cost. The distributed network is attractive to us in that regard: you can (1) minimize the number and types of components, (2) use standard elements, (3) reduce the subsystem design cycle by the use of a general-purpose microcomputer to replace hard logic wire, and (4) provide a uniform interface with the rest of the spacecraft.

We think that with the use of the bus design we can reduce the spacecraft integration cycle that occurs when we try to assemble spacecraft. The number and types of interfaces will be markedly reduced from the types of interfaces that we now have on any given spacecraft.


GENERAL-PURPOSE MICRO-COMPUTER 1
GENERAL-PURPOSE MICRO-COMPUTER 2
DISTRIBUTED DATA HANDLING AND CONTROL FUNCTION
GENERAL-PURPOSE MICRO-COMPUTER 3
INSTRUMENT 1
GENERAL-PURPOSE MICRO-COMPUTER 4
INSTRUMENT 2
GENERAL-PURPOSE MICRO-COMPUTER 5
INSTRUMENT 3
GENERAL-PURPOSE MICRO-COMPUTER N
INSTRUMENT N

SPACECRAFT DATA HANDLING AND CONTROL CHARACTERISTICS:

GENERAL-PURPOSE MICROCOMPUTER DESIGN
LARGE-SCALE INTEGRATED (LSI) CIRCUIT TECHNOLOGY
DECENTRALIZED INSTRUMENT CONTROL AND DATA GATHERING
DATA BUS CONNECTION BETWEEN DATA HANDLING AND CONTROL FUNCTION AND INSTRUMENTS

Figure 3-3. Distributed Computer Solution to Data Handling and Control Problem (1980 and beyond)

And, of course, the distributed network has the attribute of being adaptable to multiple missions. A great deal of commonality is possible through the use of a general-purpose set of devices that can be reprogrammed for different mission requirements. The increase in reliability by the use of LSI components is attractive. We think we can increase the spacecraft performance capabilities while keeping the cost down. It is essential to us, particularly in the deep space application, to reduce the spacecraft weight and power requirements.

One way NASA, through the Low Cost Office, is trying to get a handle on the LSI situation is to develop what we call a Remote Terminal Unit (RTU), a microprocessor based module. The RTU would have usefulness or application for multiple applications throughout NASA. I will describe a functional block diagram of the RTU and then show an example of its usage on a particular spacecraft design, namely the Jupiter Orbiter Probe.

The RTU (Figure 3-4) is a module which is comprised of a microprocessor read-only and read-write memory, a bus interface (whether the existing STACC bus or a 1553 interface) and some standard I/O components, which would be used to interface the module with the particular subsystem or instrument that it is interfacing with. The RTU performs functions of replacing existing hard wire logic within an instrument and, in this case at the instrument or subsystem level, does local gathering of data, formating and buffering. Also, the RTU provides a uniform interface for an instrument with the rest of the spacecraft.

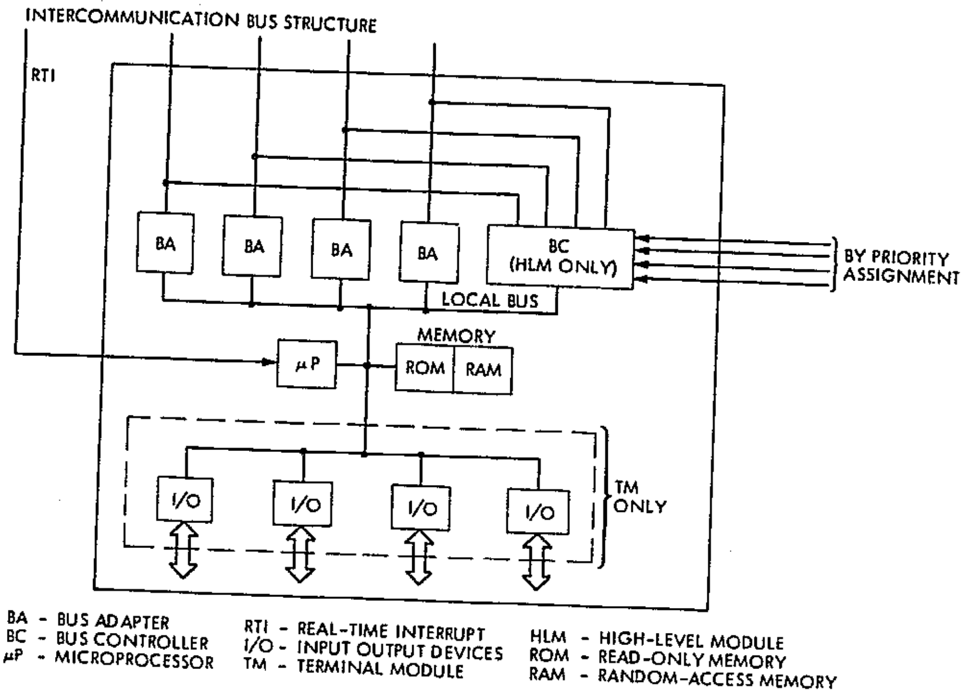


Figure 3-4. Jupiter Orbiter Probe Mission Typical Microcomputer Module

The heart of the module is a microprocessor and its associated ROM and RAM memcries. The interfacing to the instrument is done via the I/O devices. We are trying to standardize not only on the microprocessor and the ROM and RAM, but on a family of I/O devices that can be provided from the family associated with the microprocessor. In some cases, some special-purpose designs might be required to fill out the number and types of devices we need to interface with the instruments or subsystems. We have selected a family already based on our understanding or knowledge of interfacing throughout various spacecraft over the years. This list is being updated as we look at new designs. Our goal is to try to make as much use as we possibly can of existing I/O devices that are provided by the microprocessor manufacturer, but in some cases this will not be possible.

Figure 3-5 shows the block diagram of the spacecraft data handling and control subsystem for the JOP mission. The key points here are that each and every science instrument contains one of these modules. So for any given spacecraft we might have anywhere from 15 to 20 of these modules distributed about the spacecraft. There is one device called the data handling and control subsystem that is responsible for the coordination of movement of data between these devices; it is the subsystem that contains the bus controlling devices. For this particular mission there are two halves of the spacecraft, the despun section and the spun section. This configuration has dictated that a minimum number of wires be allowed to connect one half to the other, and this has almost driven us to the use of a bus design.

The key points I wanted to make here are that in any distributed system the number of devices or remote terminals that we would use would be between 15 and 20, not including the redundant devices we need for the fault tolerance. Distributed intelligence on either side of the slip rings is required, and as a result the number of connections is minimized.

The Jupiter Orbiter Probe mission characteristics were used to give you a feel for the kinds of requirements and restraints the mission levies on the selection of microprocessors and the reliability required. The first characteristic is the mission duration. The spacecraft that is going to go to Jupiter takes three years to get there, followed by nominal orbital mission of 18 months with an option of 6 more months, so there is a possibility of a 2-year orbit after arrival at Jupiter. That means that the data handling and control function has to operate for 5 years unattended. The stated mission reliability requirements stress the importance of delivering the probe into the atmosphere of Jupiter, so the primary reliability requirements say that no single failure will prevent probe delivery and relaying of probe data to earth. Such requirements dictate what we have to do in a fault-tolerance sense in designing the data handling and control system.

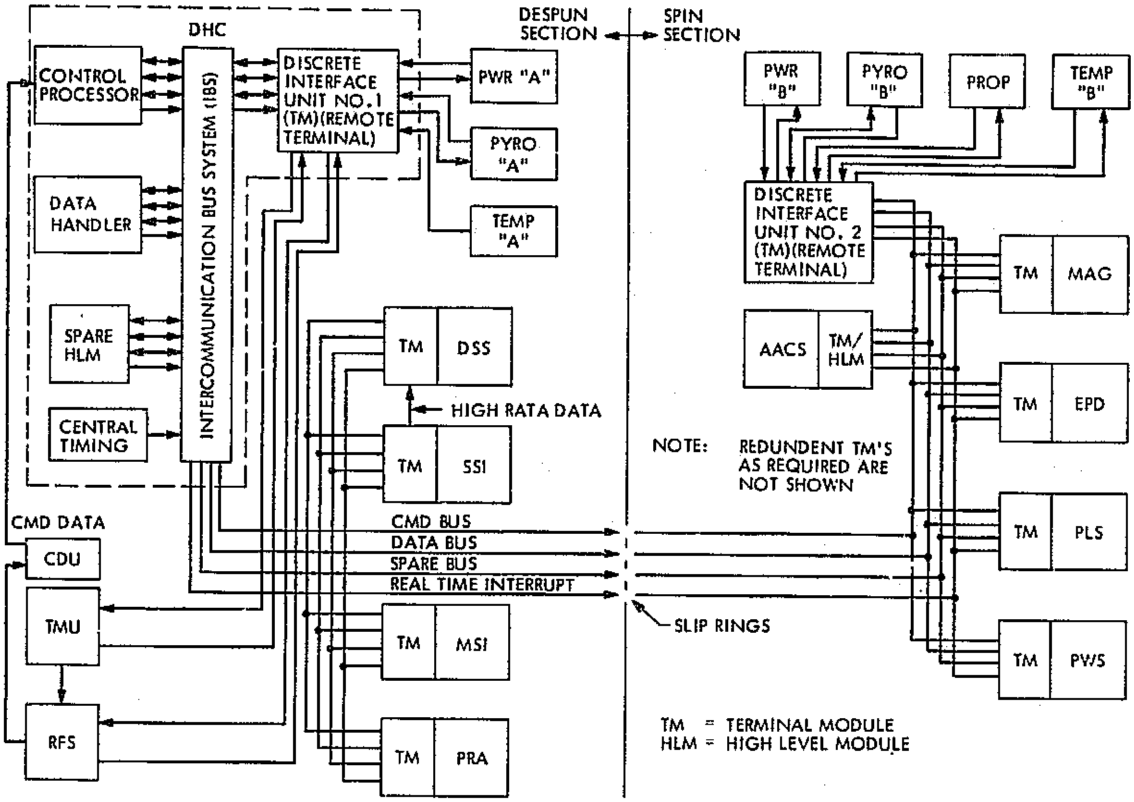


Figure 3-5. Jupiter Orbiter Probe Mission Data Handling and Control Subsystem Functional Block and Interface Diagram

A secondary reliability requirement is that no single failure will cause the permanent loss of science data from more than one scientific experiment, which means that a science experiment may fail on its own but that there should not be any single core function in the spacecraft that would allow the loss of all the data from the science complement or data for more than one experiment.

As far as environmental requirements are concerned, normally we design to a level of -20 to +70°C because of the temperature control capabilities on a spacecraft. We call that range our type approval or TA levels. But in this particular spacecraft, which uses a distributed system with science experiments deployed on appendages, the temperature control problem becomes more severe. When I started looking through the environmental requirements I realized that there are instruments whose requirements are in some cases even greater than the MIL specs. For instance, for instruments out on booms there is a -60 to +80°C temperature range requirement.

Another rather severe constraint on our design is the fact that there is a very severe radiation problem in the Jupiter magnetosphere. How much radiation the spacecraft has to survive in a total dose sense is a function of the orbit and mission profile. Right now we are designing to 150 kilorads or so, but we have been working toward the premise that we ought to be able to design to 10^6 rads total dose to give us enough margin to survive the 2-year orbit.

In any given spacecraft there are certain design requirements levied on the data handling and control function because of constraints in both mass and power. For this particular spacecraft there is 1500 kilograms total mass allowed for the orbiter, of which 15 kilograms is allocated to perform the data handling and control function. Therefore, about a kilogram per module is allocated to perform these distributed processing functions. In the case of power, a 400-watt RTG supplies orbiter power, and the data handling and control function is allocated 25 watts. So these are the kinds of requirements that dictate what sort of modules we must design. As you can see, the key requirements are the radiation tolerance and low volume and power. Also, another requirement is to use standard elements as much as we possibly can in order to keep costs down. This illustrates the kind of reliability problems we face and what kind of selection problem we have to go through and what kind of fault-tolerance measures we have to take in order to meet our mission objectives.

I will try to get down to the specifics of what we think we need as the basic building blocks for these systems. As I indicated earlier, we are trying to develop modules that not only have application for the JPL spacecraft but can be used as NASA standards. We think that if we can build them to meet JPL requirements, which we feel are fairly severe, they can be used for many other NASA applications. This list is a first cut and doesn't represent much coordination with the other NASA centers. We plan to do this in the very near future, by forming a NASA-wide inter-center working team to see if there are some requirements that we haven't factored into our tentative selections.

Table 3-1 presents a list of the microprocessor unit requirements as we see them today. As for word size, we think we can get by with an 8-bit machine, although if a 16-bit machine were available we would sure take it. We have been working in our laboratory to develop a breadboard system of a distributed computer approach using 16-bit minicomputers, and we are skeptical of the software overhead required by an 8-bit machine. We are trying at present to establish some bench mark programs to see if an 8-bit machine will do the job for us. It would appear at first glance that an 8-bit word length would be satisfactory. Addressable memory up to 65K words is required. Some of our central control functions need a significant amount of memory to store sequences and program to conduct the mission profile. As for speed, the add time bench mark looks like around the 5 microsecond mark would be satisfactory. Some applications that we foresee would need a hardware multiply-divide capability, such as the attitude control function. Direct memory access is a key to our concept of moving data from block to block.

Table 3-1. Microprocessor Unit (MPU) Requirements

Microprocessor word size:	8 bits
Addressable memory:	65 kW
Add time (reg. to reg.):	≤5μ sec
Hardware multiply/divide:	Yes
Direct memory access:	Yes
Microprogrammable:	?
Hardware floating point:	?
Indexed and indirect addressing:	Yes
Interrupts:	8
Power:	1/4 w
Architecture:	Fixed word length
Special features of MPU instruction set:	
Desire compatibility with mini-computer instruction set	

Microprogrammability and hardware floating point capabilities are question marks for now. We think probably that as we solicit requirements throughout NASA somebody will need to have these capabilities. The index and an indirect addressing are needed to support our programming efforts. As for interrupts, we try to limit the amount of interrupts needed. We use more of a pooled sort of a system versus an interrupt driven system, but for general-purpose application we think that eight levels of interrupts would be appropriate. As for power, we are talking about a quarter watt for the MPU. We really require something in the CMOS line to meet our power requirements. If we have to deviate from the CMOS technology then we have to pay a pretty severe power penalty to use bipolar, PMOS or some other technology.

Question: Have you considered using the I^2L technology?

We would like to look at I^2L; however, we feel that it is probably not as mature as some of the other technologies but it certainly looks very attractive.

Question: Have you considered using the bit slice?

We have considered some bit slice architectures that might be attractive for our use. For right now we have been trying to identify for the present those microprocessors that we thought had some chance of being fabricated in a hi.rel fashion. We have been looking at things like the RCA 1802 for example. We are trying to take a guess at what kinds of microprocessors might be able to be fabricated to our requirements in the time frame that we are talking about. There are other architectures that might meet our needs, but we are trying to base our selection on what might be available from the commercial market.

Question: What does the question mark opposite the microprogrammable entry in the MPU requirements mean?

The question mark means we do not know yet. For the JPL application for the data handling and control functions we do not need that capability. We are trying to satisfy a more broad spectrum of users and we have not been able to sample all the NASA centers to know whether or not this is a required capability.

Question: Could you tell us more about the desired special feature of having the instruction set compatible with a minicomputer?

We are concerned about the test problem, the support equipment problem, and the associated support equipment that goes with the microprocessor. We are also worried about training people who have skills in both programming a minicomputer and a flight microprocessor, and if we could kill both those birds with one stone then we would be far ahead. I am not sure whether this is a reachable goal but it is a highly desirable one for us to not have to cross-train groups of people to learn how to program two different machines.

Question: Have you looked into software availability for the various microprocessors and made some kind of a decision regarding what you need?

Yes. We have identified what kind of support capabilities we need to support our software development activities. I had intended to show them.

Question: Is there a requirement for higher order language?

Not as we see it for our application. We thought it more important to work the design language problem, and we have in fact developed a design language that is germane to our particular sequencing sort of problem.

Table 3-2 shows a list of memory requirements. As I indicated in the description of our basic module, we need both ROM and RAM. We would also like to consider PROM if it can be made reliable for our usage in terms of blowing fuses and so on. The maximum size needed is as big as we can get for the RAM for storing high volumes of data. It looks like

Table 3-2. Memory Requirements

Type:	RAM, ROM, PROM
Max size:	65 K RAM; 8 K RAM, PROM
Power:	50 MW/1 KW
Speed:	Compatibile with MPU

Input/Output or peripheral device requirements

Programmable parallel I/O interface

Programmable serial I/O interface

Programmable DMA controller

Programmable interrupt controller

Programmable Interval timer

Universal asynchronous receiver/transmitter (UART)

Analog-to-digital converter

Digital-to-analog converter

MIL STD 1553A bus interface

the largest ROM or PROM we might need would be 8K. In any given spacecraft design we try to encourage an instrument designer to put all his control and sequencing programs in ROM or PROM such that these programs are frozen and solidified way before launch and we do not have to have a large volume of support software on the ground to keep track of changing things in flight. This has been a bugaboo in the past for us and a very significant cost driver. The more flexibility you allow on the spacecraft, the more software you need on the ground to keep track of what is going up on the spacecraft. So we like to encourage the use of ROM to try to freeze those programs. As for memory power, low power is required as we are very constrained there. The memory speed must be compatible with the selected MTU.

The I/O devices that we talked about are some of those which would normally be provided as part of the I/O family with a given processor. We do have need for A to D converters and D to A converters. We are also interested in finding LSI devices which implement the MIL Standard

1553A bus. In fact, right now we are working with one LSI device that has been designed by Rockwell to see if it meets our needs. It is a CMOS/SOS device.

Comment from the floor: I hate to tell you you are wrong, but we see the need for more memory than you have indicated.

For the simple devices we have been doing, 8K ROM is required as a minimum. Preliminary estimates that we have done for some of our applications are simplistic maybe compared to yours, but we are using 200 to 600 words for some of our programs. The estimates are based on 16-bit words for our applications so for 8-bit words we are thinking 1 or 2K 8-bit words is the average amount of ROM that any experimenter might need to do his particular function.

The worst environmental requirements I could find were the -60 to +80°C temperature range and the radiation 10^6 rad total dose. As for quality assurance requirements we are looking for something in terms of the qual-conformance test and screening, something equivalent to MIL Standard 883A. Nobody has really determined what MIL Standard quality means in terms of what must be done to LSI, and maybe if you want to address that part of the question we can wait until someone more expert in that speaks. We have some speakers who are going to address that question in a little more depth.

Question: How does the use of the standard module relate to the guidance and control system?

The guidance and control systems that we plan for our future spacecraft will in fact use basic building block modules as integral parts of the guidance and control functions to do some of their computing and also interface with the rest of the spacecraft. Not all of their requirements have been gathered as yet to understand exactly what capabilities might be required. The guidance and control people are right now evaluating the design of the basic module to be used as integral parts of that particular spacecraft function. In some cases, as in the case of JOP design, there is really a multiple hierarchy involved for the distributed system.

The first level in the hierarchy is for the data acquisition and control function. The attitude control function in fact is a hierarchy of its own which may include a high-level control microprocessor module and several others which are associated with particular elements of the attitude control system. For example, a microcomputer is used as an integral microcontroller for the star tracker part of that system. So there might be several microcomputers that are used to perform the attitude control function as well, with the one high-level function performing the interfacing with the data handling and control function. As of now, the multiple hierarchy exists for the JOP example, which uses two or three microcomputers associated with that function.

Question: Could you comment on your experience in using high-level language versus machine language. With the 8080 if you wrote the same program with high-level language instead of machine language, 42% more code would result.

I guess that is another field that we have to look into in more depth. If we are not trying to minimize the number of components and the volume and so on, then you might be able to afford that expense even though hopefully you can generate code faster. To perform the data handling and control function for any of our spacecraft designs we have always been constrained in a rather strict sense in the amount of core that we had available for that function, and we have never had the luxury of generating software in the high-level sense. It has always been an assembly language sort of process for us. We are still thinking that way, but we have developed a design language to help us through the design phase of the software development.

Question: In terms of high-order languages have you evaluated the possible use of the HAL-S language?

We have been encouraged through the Low Cost Office to evaluate the use of that tool for our development, but we have not really done so to any great depth yet.

Comment by Paul Lecoq: I just want to make a general comment. The number of words you use may not be the important criterion. Probably much more important is how you understand the program. On the other hand, high-level language is still in the sphere of what we are trying to do. Right now our concern is how we specify what we want a particular module to do, and we are willing to accept assembly language programs for the present time.

Comment from the floor: I would like to make a comment at this point and argue with you about worrying about new architectures over the next five or six years. Now regarding the two question marks you had on your check list. As you can see, those are probably going to be demanded by some other user. You have really only got about two variables left: speed and word length. In every other respect there is a general-purpose fairly high-level computer, and maybe what you are saying is that the architecture is no longer a problem. We are going to want most of the features including hardware multiply and divide, direct memory access and microprogrammability.

That is a good statement because as soon as you talk about trying to satisfy a broad spectrum of users, which is exactly what we are doing, then all of the listed capabilities will certainly be required.

Comment from the floor: Even in that respect I think maybe what we are coming back to is you might want a common architecture which could be achieved in different technologies to give you lower power or higher speed. You said you had worked with this for about five years time to analyze what architecture would be necessary or common for all of JPL's needs. We ultimately came to the conclusion that you wanted just about everything. Somebody was very interested to hear you say that for your attitude control you were going to need larger word size and maybe more speed, and perhaps the hardware multiply/divide, and then you have got just about the best architecture you can ask for.

I think that is a very good point and something we can discuss during the next couple of days.

Question: Are you looking to the commercial market as a source of a microprocessor or are you planning to design your own?

We would like to find an existing circuit that somebody else has invested in, with all the supporting software and documentation and history and a commercial basis. I don't know if that is achievable or not, given all the requirements that we have identified here, but that is our goal. We do not want to design a microcomputer ourselves, we are in the shopping business here.

Question: Have you identified any commercially available microprocessor that would serve as a basis to meet your requirements?

We have a limited knowledge of what is going on out in the industry, but in talking with the Navy and Air Force and so on, the RCA 1802 keeps popping up.

Question: What about the RCA 1802 radiation hardness?

Some of our contacts say that Sandia Labs is interested in producing a hi rel version of an 1802 that in fact can be hardened or is hardened 10^6 rads. I am not too familiar with the schedule for that activity, exactly what the commitments are and what kind of resources are being expended, etc. Larry Wright can perhaps give a better picture of Sandia's plans.

Comment by Larry Wright: We have a meeting scheduled early next month to discuss this further. Again, this is kind of a hearsay sort of activity that is going on, and one of the reasons for this meeting is to try and find out exactly what all of these other agencies are doing and establish the necessary contact to find out what resources are being spent on that sort of activity and what the schedule is, etc.

Comment from the floor: I can speak for the Air Force. We are driving towards one that meets the 10^6 rad total dose requirement.

Comment from Ralph Martinez: I sort of agree with Paul that, based on what you know now, you can characterize some of the devices. NELC through NRL is doing some rad hard work on the 1802 with the idea of finally coming up with some design rules that would make the 1802 rad hard CMOS. The end result would be to carry that over into a CMOS-SOS process. The report from NRL would be out some time in December on the 1802 characterization. One of the goals incidentally was 10^6 rads total dose.

In our situation our parts selection activities indicated that an evaluation of an 1802 was warranted. We are not sure this is going to be our final selection, but we need to understand how the 1802 works, how it could fit into a design, and whether in fact it could do our JPL job. We did a little homework with the 1802 to try and understand how it would work, to try and establish some software bench marks and see if an 8-bit machine would do our job. We are certainly not locked into the 1802 by any stretch of the imagination. It is kind of a chicken and egg game. You have to poke your nose out and see what is going on and assess the potential of getting these microprocessors to do what needs to be done in the time frame that we are talking about.

Regarding the time frame then, we are proposing that the remote terminal unit be designed and qualified for use for NASA in accordance with the schedule shown. One thing to note: there is a two-phase design involved here. One phase is an LSI selection process resulting in electrical design of a basic module around a microprocessor essentially being completed by the end of fiscal 1977. In FY78 we embark on actually designing and packaging a flight-qualifiable module and actually running qual tests on the basic RTU. The key point in Phase I then is that we have to make our microprocessor and associated LSI selection by essentially May 1 of next year in order to meet our project commitments.

Now there will be two paths that take off from the end of Phase I. One is the completion of the design of the basic RTU, followed by Phase II, where we will actually go through a packaging design, putting it all together. If any hybrids are required, special LSI or what have your, these designs will be accomplished as part of Phase II. In parallel with the Phase II activity will be the parts qualification program. While we are trying to figure out how to package this module, another group will be going through the parts qual program, which is probably a 12-15 month program in itself to actually qualify the components, to qualify the facilities, the products, additional tests and what have you. At any rate, that is the way it looks to us at NASA, and as you can see, our needs are fairly urgent. I do not know how they compare to other agencies, but in order to support our commitments we have to move out. As our first step we are having this conference, and we hope we can learn enough here to allow us to meet these milestones.

Question: Are you planning to use the NASA Standard Parts List?

I guess what would happen is that these selected components would become part of the NASA Standard Parts List. If these parts existed there on the list already, that would be fine, but I guess through Dick Scott's activity we would fold all this microprocessor selection activity into the NASA Standard Parts activity. Also, regarding existing parts, for example, we are looking at existing NASA standards like the standard transponder as a possible source of components. They have A to D converters and some RAM memories that will in fact be put on the list of candidate components. Through various avenues we will be looking for already developed components that we can use for our application.

SECTION IV

ASSESSMENT OF POTENTIAL MARKET FOR HIGH-RELIABILITY LSI

Ralph Martinez
Naval Electronics Laboratory Center

A. SUMMARY

Dr. Ralph Martinez of NELC discussed the Navy's interest in microprocessor technology. Dr. Martinez's principal work areas at NELC encompass microcomputer and microprocessor design and application. Last year he was the Project Engineer on a Navy microcomputer standardization program.

In his presentation Dr. Martinez described a Navy standardization program initiated to take advantage of microprocessor/microcomputer technology. The approach taken was to form a Navy/industry team to define present and future microprocessor trends, identify microprocessor performance requirements, define microprocessor standardization issues/options and make microprocessor standardization recommendations. The goal of the work was to produce a position paper detailing the form and scope of microprocessor/microcomputer standardization for Navy electronic systems.

Dr. Martinez described the detailed steps the team took to develop the position paper. The initial step was to define the current trends in microprocessor technology and identify the requirements for Navy systems. Fifty different Navy systems were reviewed to determine performance requirements and to identify standardization issues and options. Some of the applications reviewed were general-purpose computing, signal processing, dedicated processing, and dedicated controllers. The characteristics that were evaluated for the specific applications were speed, input/output requirements, data rates, environmental requirements and special characteristics.

A summary of processing requirements for Navy systems was presented which is summarized as follows:

Requirement	Comment
Speed	MOS - 2 μsec; bipolar - 100 nsec
Memory	Up to 65K
Data word length	4, 8, 16 bits; 32 via slices
I/O transfer rates	Up to 800K words/sec
HOL	PL/M, SM/PL, LGP only
Floating-point hardware	Non, must use external circuitry

Requirement	Comment
Direct memory access	DMA halts CPU, DMA controllers
Integer multiply/divide	10 μsec on chip, 1-2 μsec off chip
Vectored interrupts	Multilevel, interrupt-control chip
Microprogrammability	Only on bipolar and 3 MOS processors

As part of the study, the team tried to project technology with respect to the applications. Two technologies, CMOS/SOS and I^2L appeared as the dominant technologies, which could cover a wide range of applications, although at present no microprocessors using these technologies exist. For very high-speed applications, Schottky, EFL, and ECL were deemed to be the prominent technologies.

Dr. Martinez next discussed standardization issues relative to microprocessors. The key issues felt to be important in the study were:

(1) The broad spectrum of microprocessor applications.

(2) Proliferation of microprocessors in Navy systems.

(3) Microprocessor part availability over system life span/ second sources.

(4) Rapid advances in LSI technology.

(5) Availability of militarized LSI components.

(6) Procurement/maintenance/logistics.

(7) Microprocessor hardware/software/firmware compatibility.

(8) Microprocessor/microcomputer system interface compatibility.

(9) Microprocessor procurement, design, and documentation guidelines.

(10) Compatibility with existing Navy software.

(11) Total system design and maintenance costs.

(12) Level of standardization.

Levels of standardization were next considered. Among the standardization options considered were (1) fixing the architecture and instruction set of a standard processor, (2) fixing the interface including data bus, (3) standardize on the microcomputer "black-box," (4) standardize on a microcomputer configured from microprogrammable parts (bit slice), (5) a more general type of standard in which a set of microcomputer and support system specifications would be developed, and (6) use of a high-order language as the standard so that implementation of the software would be processor-independent.

The rationale that was used in making the final position paper recommendations was developed. A key point made was that speed, not functionality, was the dividing line between the use of standards and special-purpose devices. Dr. Martinez showed slides that indicated 50 to 200 kips for presently available microprocessors with speed increasing to 200 to 500 kips in the 1977-78 area.

The final position paper recommendations were presented and are summarized as follows:

(1) Standardize on families of 8-bit, 16-bit, and bit slice functional modules.

(2) Standardize immediately on 8-bit CPU as an interim standard.

(3) Standardize on critical support modules for the 8-bit CPU.

(4) Select commercially available 8-bit CPU, support hardware and support software by competitive procurement.

(5) Use the Navy's Standard Electronic Module Program to implement the standardization scheme.

(6) Require second sources for all LSI parts and modules.

(7) Require a full set of hardware/software support systems for each module family.

(8) Select a standard instruction set by the choice of the 8-bit, 16-bit or bit slice CPU's.

(9) Proposed schedule:

8-bit modules	Aug 1976
16-bit modules	Oct 76 - Aug 77
bit-slice modules	Oct 76 - Aug 77

B. PRESENTATION

When we first started this program we had essentially a three-year program defined to come up with a microprocessor/microcomputer family standard. Due to a requirement in one of the Navy offices, our three-year program was compressed to six months. As a result, there was quite a bit of concentrated effort to do this work. One of the things we were interested in with regard to microprocessors is that they form the basis for a modular building block (Table 4-1). This characteristics gives you the capability to do system reconfigurability with a minimum loss of operational readiness and reduces cost with respect to design cycles,

Table 4-1. Navy's Interest in Microprocessor Technology

Modulator building blocks for complex systems
Cost reduction in design, production, and maintenance life cycles
System reconfigurability with minimum loss in operational readiness
Potential hardware and software proliferation
Need to develop systematic approach to take advantage of microprocessor/microcomputer technology

maintenance, etc. However, there is also a bad point in that there is a potential software and hardware proliferation. Thirty odd processors are on the market today, some of which are already getting into advanced development systems in the Navy. The idea was to come up with a systematic approach to define a standardization scheme using the current technology.

One of the first questions to be answered is: Why do you want to standardize in the first place? You can talk on various different subjects here. We are told that the generation and the maintenance of software are among the biggest costs in the military. Once you get the systems in the fleet, other considerations are, how do you do hardware and software support and maintenance, the training of the operators on the system, and the whole bag of logistics (Table 4-2).

So the goal of our program was to come up with a position paper detailing the form and scope of microprocessor/microcomputer standardization for Navy electronics systems. This was primarily a six-month study where we had very little hardware work. We did some hardware work in the background, just to keep current and make sure we had our hands dirty. The approach we took (Table 4-3) was first of all to put together an industry/Navy team composed of eight Navy laboratories, of which three were actually funded and the others had ongoing programs that contributed to the effort. We worked with several industry teams and had a separate standardization study going on at Rockwell-Autonetics independent of what we were doing.

The first thing we were going to do was define what the current trends were in the microprocessor area and identify what the requirements were for Navy systems (Table 4-4). We took a sample of several systems that were in the fleet, ADM, or R/AD laboratory stage. In total, we looked at approximately 50 systems. Based on the trends and the performance requirements, we wanted to identify the types of issues we were faced with and the standardization options, which were the basic ways we could approach the standardization effort in this area. Finally, we wanted to make some recommendations and form a position.

Table 4-2. Standardization: A Tool to Lower Life Cycle Costs

Software generation
Software maintenance
Software support tools
Hardware maintenance
Training
Logistics
Initial procurement

Table 4-3. Approach (Six Month)

Navy/industry team
Define present/future microprocessor trends
Identify microprocessor performance requirements
Define microprocessor standardization issues/options
Make microprocessor standardization recommendations

Some of the things we looked at were system characteristics that were very close to the characteristics of the microprocessors and LSI. We looked at basic functions where we had a general-purpose computer: signal processing (what we call dedicated processing, categorized as data formating), data acquisition and data routing, and controllers (disc controllers, memory controllers, I/O controllers, etc.). We looked at the speed of the subsystem or the processor within the subsystem in terms of instructions per second. We looked at the input-output characteristics, data format, data rates, etc. And we looked at memory, environmental requirements, and special characteristics which included things like software languages, and special hardware such as floating point and multiply/divide (Table 4-4).

Table 4-4. System Processing Requirements

Basic processing function

Speed-executed instructions per second

Input/output characteristics

Memory characteristics

Environment requirements

Special requirements

These areas are also representative of microprocessor characteristics

Figure 4-1 shows system computational function complexity including general-purpose signal processing, dedicated processing, and controllers mapped with respect to instructions per second. This is not a density diagram of the applications that we sampled but tells the general areas the speeds fell into. This is also not to say the Navy does not have any systems which operate about five mega instructions per second. Of the sample, these are the characteristics that we came up with for the instructions per second.

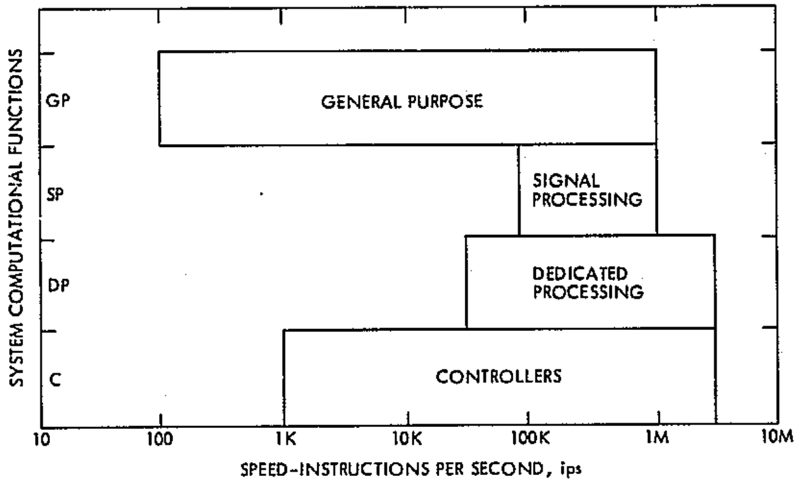


Figure 4-1. Navy System Computational Functions vs Speed

Tables 4-5 and 4-6 are an all-encompassing summary of the systems that we looked at. For all data word lengths, 8, 16, 32 and 24 bits are needed. On the memory, most applications needed up to 65K although in some of the signal processing, dedicated processing, and controllers about 16K was the average required. Something interesting is the special requirements section in that the general-purpose system required floating point arithmetic hardware. There is a big void in LSI in that there is not floating point hardware that supports the LSI microprocessors. There was a big need for multiply/divide hardware for both 8-bit and 16-bit type operations. Interestingly enough, although most of the systems did not have the capability to microprogram, everyone asked for a microprogramming capability.

The components of Figure 4-2 seem to vary from day to day, week to week, and are also a function of whom you talked to last. We concluded that two dominant technologies were capable of encompassing several applications in terms of speed over a wide area of functional requirements, I^2L and CMOS-SOS. The problem is that no processors in those areas exist, at least in a commercial sense. We did see these two technologies to be very similar with wide applicability, if we could get a microprocessor, say a 16-bit, using these technologies. We were also very much interested in the high-speed area, where we felt that Schottky, EFL and ECL were the prominent technologies. Since this slide was made I guess Motorola has dropped the development of militarized EFL processor. But TRW is heavily involved in the triple diffused EFL technology.

I will talk about some of the issues that concern the standardization effort (Table 4-7). One of the things that we quickly saw in the sampling of systems was the broad spectrum of applications. We had everything from low-speed controllers to high-speed signal processors and general-purpose machines, which were in the area of the general-purpose machines like the Navy standard minicomputer, the UYK20 and the UYK-7, which is a 32-bit machine. The other thing is the proliferation of these processors in the Navy systems. In addition to having these parts in several systems, we now have to have a separate hardware and software support system for each part to maintain that part for the system lifetime. So this was also a big problem, not only hardware but also software support system proliferation.

We are concerned about having to obtain parts for systems over an 8-, 10-, or 15-year life span. Some of the manufacturers told us that retaining parts over this period was not possible. We are working around this problem in several different ways, one of which is to do volume purchasing for the system lifetime at initial procurement. Another way is to take the masks and put them in a drawer, file them in the vault and then pull them out in 10 years, and you go out in the streets to see if anyone will build that LSI for you. Not too probable, but this is their approach. And of course you always get the issue of the LSI technology moving so fast that whatever you define today is going to be obsolete next year. We are faced with this problem.

Table 4-5. Summary of Navy System Characteristics

	Class of system			
Parameter	General purpose	Signal processing	Dedicated processing	Controllers
Data word length, bits	8,16,32	16,24	8,16	4,8,16
Maximum memory size, bits	64K by 32	32K by 24	16K by 16	16K by 16
Maximum I/O transfer rates words/sec	250K	500K	800K	200K
Special requirements	Higher order language	Direct memory access	Direct memory access	Direct memory access
	Floating-point arithmetic hardware Non-volatile memory Direct memory access	Multiply/divide hardware Microprogramming	Multiply/divide hardware Microprogramming Vectored interrupts BCD arithmetic	Multiply/divide hardware Microprogramming Small size Low power

(1) Minimum programming language is assembly language.
(2) Temperature requirements included MIL-E-164006 specifications.
(3) Power requirements were not a constraint in most systems.

Table 4-6. Summary of MOS and Bipolar Microprocessor Characteristics

	Microprocessor type			
Parameter	4 bit (MOS)	8 bit (MOS)	16 bit (MOS)	slice (bipolar)
Speed, ips (minimum instruction time)	100K (10 μsec)	625K 1.5 μsec	500K (2 μsec)	10M microcycles, 100 nsec
Data word length, bits	4,8	8,16	8,16	Variable-multiple of 2 or 4 bits
Maximum memory size, bits	4K by 4	65K by 8	65K by 16	Variable
Special characteristics	BCD arithmetic -55 to +125°C	BCD arithmetic Higher order language Vectored interrupts Low power Microprogrammable -55 to +125°C In-circuit emulation	BCD arithmetic Higher order language Vectored interrupts MUL/Div hardware Microprogrammable -55 to + 85° C In-circuit emulation	Vectored interrupts Microprogrammable 2 or 4 bit ALUs -55 to + 125°C

(1) All 4, 8, and 16 bit microprocessors feature assembly language software.

(2) Only a few MOS microprocessors can meet the -55 to +125°C temperature range. However, the majority meet -55 to +85°C.

(3) Some slice microprocessors feature microassembler software.

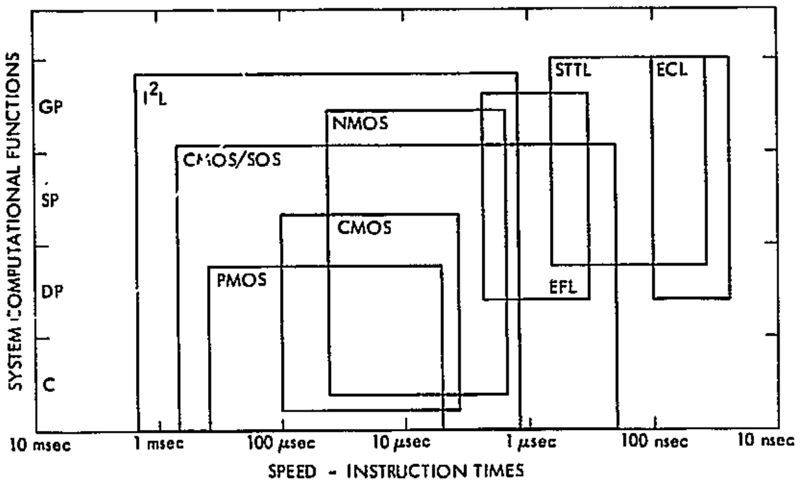


Figure 4-2. Projected Speeds for MOS and Bipolar Technologies

Table 4-7. Microprocessor Standardization Issues

Broad spectrum of microprocessor applications

Proliferation of microprocessors in Navy systems

Microprocessor part availability over system life-span/second sources

Rapid advances in LSI technology

Availability of militarized LSI components

Procurement/maintenance/logistics

Microprocessor hardware/software/firmware compatibility

Microprocessor/microcomputer system interface compatibility

Microprocessor procurement, design, and documentation guidelines

Compatibility with existing Navy software

Total system design and maintenance costs

Level of standardization?

Taking a look at the LSI characteristics, there were very very few militarized components at the time this study was performed in the spring of '75. There were no LSI's, at least no microprocessor that met the full military temperature range. Now since then there are a few that do meet it. All these things serve to give admirals and the like procurement, maintenance, and logistics headaches.

One of the things we were concerned with in choosing the different standards was that we wanted to have all the hardware, software and the firmware play together. One of the things we could have done was recommend that we could get a part from one vendor, an I/O part from another vendor, software from a third vendor. So we not only have standardization at the part level, but also at the microprocessor system level as far as parts availability and software availability are concerned. Another concern was how do these LSI systems interface with current Navy systems: will they handle the current interface capabilities and NTDS type capabilities? And of course there were the procurement and design documentation guidelines, which are probably among the biggest items that will impede a standardization effort in that you have to have, with the standardization recommendations, a scheme of business management for implementing the standards. In fact, what I wanted to do here was quote a paper that was delivered at WESCON this year by Leo Chamberlain of Texas Instruments. What he delivered in the paper was TI's approach to standardization with respect to the 9900 family. What Leo said was, "The final factor in achieving standardization and probably the most important in initializing a standardization program is simply that of management structure to cause standardization to happen." And what he is saying essentially is that with respect to TI that management structure was there and they could dictate what standardization techniques TI engineers would use. He says, "This in turn is probably the greatest impediment to the implementation of standardization within the government. The absence of a single point of authority and the necessary response to the competitive pressures in the total military market make implementation of standardizations more difficult in the government." I think it is true with some of the things I have seen. He also says, "There is no single point of authority that can exercise a standardization program within the government." Within the Navy there are some potential standardization authorities which are the ones that in the past have dictated Navy standards. There is a tactical digital systems office that dictates the UYK-20 standards, and there is now a standard electronics module program which does not dictate standards but makes standards available. I do not know if in the other services there are similar types of organizations. So this was and continues to be a big problem.

The compatibility of the microprocessor standards software with existing Navy software was also an issue. The Navy has invested quite a few dollars in coming up with the UYK-20 standard minicomputer software system, HOL's for that system, loaders, assemblers, etc. One of the issues was that we should capture some of this existing software. It is probably not as good as a PDP-11 software base, but nonetheless it exists and is being used for systems within the Navy.

We are concerned with the total system design and maintenance costs. We really did not care that much about the cost of the parts themselves. Basically, standardization would be low volume compared to the commercial market. We are interested more in the elements which would reduce system design and maintenance costs. One of the things we are interested in is what level of standardization we could approach with this program.

Some of the standardization options that we felt were available are not mutually exclusive; they are all actually interrelated (Table 4-8). One of the things you could do would be to fix the architecture of a standard processor, and fix its instructions set. Essentially when you do this you fix some of the functions of the processor. To a certain extent you also fix how the processor communicates to the external world to a standard interface. So by fixing an architecture, say the CPU registers, the ALU, etc., you also fix the instruction set; and vice versa, if you fix the instruction set you fix the section of the architecture. There are several advantages in standardizing architecture. Given a standard architecture you could possibly come up with different standards and different technologies. One of the big questions is which architecture to use. We do not want to develop a custom processor. There are a lot of similarities which affect existing processors. In fact we did some bench marks, including 16 8-bit'ers and 6 16-bit'ers, and the way the benchmarks came out clustered, there was no one processor that had a significant advantage over the other. Another problem that we as government users face is how do we select one processor over another for general-purpose standardization when they are so close to each other? There is a competitive process there that we normally have to go through.

Another standardization level was to fix the interface. This would include a standard bus and a fixed amount of data lines, address lines, and control lines. The different parts on the different sides of the I/O bus would be variable, so you could put in different CPU's. For example, the problem where a gentlemen is looking for a CPU to replace an existing CPU because they no longer make the first one could possibly be solved by this type of standard. Another standardization level was to come up with a microcomputer box which includes the CPU, the RAM, ROM, and I/O parts configured out of a monolithic CPU. This will be either an 8-bit or a 16-bit microcomputer. One of the problems we saw here was that in a broad spectrum of applications that required several different idiosyncrasies in the applications, a microcomputer black box would not even satisfy 10-20% of the applications.

An interesting thing that was proposed was to be able to use the microcomputer and the corresponding chip set in a "hierarchy of standards" type of arrangement such that you could use the modules or the computer black box in your application. What you do is use the CPU and whatever packaging and I/O configuration that is optimum for your application. This keeps the support systems constant. Another option was to configure the microcomputer black box or modules out of microprogrammable parts to use the bit slice devices. This is interesting also because it allows you to capture existing software bases through

Table 4-8. Standardization Levels

Architecture, instruction set, functions
Interface
Microcomputer (monolithic CPU)
Microcomputer configured from microprogrammable parts (bit slice)
Set of microcomputer and support system specifications
Higher order language (HOL)

emulation. The other thing that it did was in special-purpose applications, where you needed high throughput and special instructions to achieve different functions, you could microprogram these functions. We got a lot of flak in this area because in the past there have been no microprogrammable standards. One of the biggest questions was how do you support a microprogrammable standard when each application has a different microcode? That is a problem for which there are still no actual answers. Although if you look at the history or look at the process that is going on now with the AYK-14 development, which is an emulator of the UYK-20 minicomputer, that basic processor is microprogrammable (with CDC as the prime contractor) and uses 2900 parts. I do not think CDC has any answers as to how the microprogram is going to be supported in different applications.

A more general type level of standardization is to come up with a set of microcomputer and support systems specifications that could be used to procure or classify several types of standards. One of the problems that we saw here is that at present one or two manufacturers would be able to complete this entire specification package, but after 2 to 5 years there would be more - say 8 to 10 standards - and you are back to the original problem of proliferation. The software factions in our study wanted to use a high-order language as the standard, with the idea that you did all the applications programming on an HOL and the implementation of the application was technology- and processor-independent, so the engineer never saw the CPU he was going to use in his application. One of the biggest problems, especially in LSI microprocessors in the areas where we needed a lot of control, bit manipulation, bit shifting, etc., was that some of the high-order languages just did not handle these functions very efficiently. One approach was to come up with microprocessor code generators for high-order languages.

Let me give a little background on the rationale that we used for developing recommendations. Figure 4-3 shows how general-purpose computers were used before LSI. The difference between a standard and

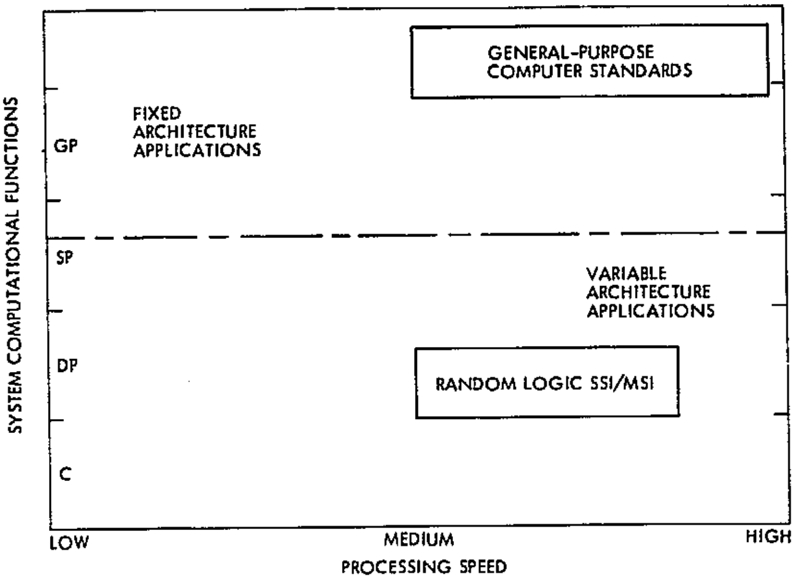

GENERAL-PURPOSE COMPUTER STANDARDS
FIXED ARCHITECTURE APPLICATIONS
VARIABLE ARCHITECTURE APPLICATIONS
RANDOM LOGIC SSI/MSI
SYSTEM COMPUTATIONAL FUNCTIONS
GP
SP
DP
C
LOW
MEDIUM
HIGH
PROCESSING SPEED

Figure 4-3. Computer Applications Before LSI

a random logic implementation is functionality, in that you use general-purpose computers for applications where you do a lot of programming and random logic where you do all control, etc., through SSI/MSI. Another constraint here was that general-purpose computers used for fixed architecture applications were very expensive. It is not economically feasible to use them as controllers for the variable architecture applications. With LSI we contend that this is no longer true. The dividing line between standards and special purpose now is through speed rather than functionality. One of the reasons for this is that you can use a low-speed 8-bit 1802-type processor in a general-purpose application and you can also use it in a controller application. The dividing line that we saw was in terms of speed (Figure 4-4), and this was due to the limitations of the LSI processors on the market, in that most monolithic 8 and 16 bit'ers, depending on the instruction mix, had upper limits of 50 to 200 kips. The speed is increasing with enhanced MOS devices, CMOS devices, and monolithic bipolar devices like the SBP9900. There is an upper limit here that we projected, based on data from the vendors, of 200 to 500 kips in the '77-'78 area. We felt that standardization schemes as we knew them in the past, say for minicomputers, could be applied to the fixed architecture applications where you did use the monolithic CPU. This meant that you had constant support systems, assemblers, cross-assemblers, compilers, and other support software. These devices were now cheap enough that you could use them again as a general-purpose computer or dedicated processor or controller.

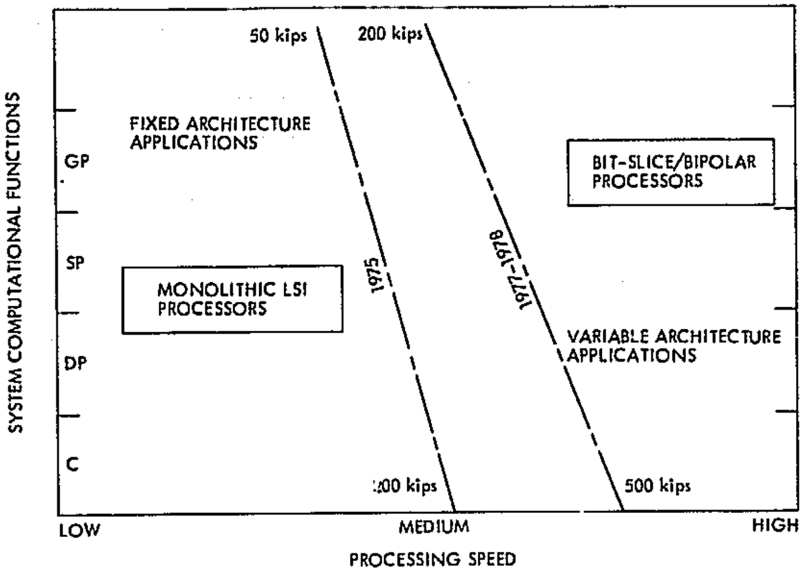


Figure 4-4. Computer Applications After LSI

We saw the area above 200 kips being satisfied by bit slice and bipolar processors (LSI processors) so that you had variable architecture applications where high speed was the prime system requirement. What we tried to do then was look at what the commonality features were between a fixed architecture area and the variable architecture areas. You can name a few of these; possibly one would be an instruction set. You have the same instruction set being executed by a monolithic 8-bit CPU, and a 16-bit monolithic CPU could always emulate that same instruction set in the variable architecture or a sub-set of that instruction set. Instruction set is sort of common between the two areas.

Other features which are common are the memory and I/O sections. There is no reason why you cannot use a programmable I/O device in a monolithic processor arrangement, and then also use the same device in a bit slice and a variable architecture arrangement. Other areas of commonality are the actual modules or the formfit in which these processors were implemented and the support systems for the processors in these areas. Assemblers, compilers, PROM generation systems, etc., could also be used in both the fixed and variable architecture areas. Out of the 50 systems that we looked at in the Navy, 80% had speed requirements over 50 kips. We felt that with one single processor we would not be able to satisfy the majority of these applications and tended to go more toward a family of standards, 8, 16, and bit slice with compatibility, if possible.

Some of the recommendations (Table 4-9) were basically to standardize on 8, 16, and bit slice functional modules, modules that could be configured in the general-purpose applications and in the minimal CPU/ROM type applications. We felt that the 8-bit market had

Table 4-9. Position Paper Recommendations

Standardize on families of 8-bit, 16-bit and bit slice functional modules

Standardize immediately on 8-bit CPU as an interim standard

Standardize on critical support modules for the 8-bit CPU

Select commercially available 8-bit CPU, support hardware, and support software by competitive procurement

Use the Navy's Standard Electronic Module Program to implement the standardization scheme

Require second sources for all LSI parts and modules

Require full set of hardware/software support systems for each module family

Select a standard instruction set by the choice of the 8-bit, 16-bit, or bit-slice CPUs.

Proposed schedule

8-bit modules	Aug 76
16-bit modules	Oct 76 - Aug 77
bit-slice modules	Oct 76 - Aug 77

sufficiently stabilized to enable us to define an 8-bit interim standard, the 16-bit market was very weak, the bit slice market had very few candidates, and the 2900 had just come out. This estimate was made in the spring of '75.

On the 8-bit interim standard, we wanted to have all the modules on the same list that Mike has presented: memory, ROM, RAM, I/O modules, programmable serial and parallel I/O units, multiply/divide. All these to support the 8-bit'er. We felt also that the 8-bit microprocessor should be selected from commercially available devices through a competitive bid process. The idea was not just to procure the 8-bit CPU, but to procure the whole system of 8-bit modules and support software and hardware including the I/O modules. At the time, there must have been two or three vendors who would have actually qualified to satisfy this procurement. Now there must be at least eight or ten. A business system to implement the scheme was recommended: the Navy Standard Electronics Module (SEM) program. The SEM program provided a vehicle to make standards available so engineers and designers could select standards. We also felt that by using the Standard Electronics

Module program we did not have to define a new business system to provide standards. At the time we wanted to require second sourcing on the LSI parts and modules. SEM also has built into it a qualification procedure for the modules that go into SEM and a second source requirement for the parts and modules.

The next recommendation is to require full software and hardware support systems. We also felt that we needed a standard instruction set. We could not go out in a procurement to select a specific one from a commercial device. We decided to let the dictation of a standard instruction set be defined by the selection of the hardware or the total system. On the 8-bit'ers, the approach was to define the 8-bit modules. In March of '76 there was an action by the SEM program which defined an 8080 and 6800 to go on to SEM modules, and in fact these were current developments that had been going on for awhile. On the SEM modules, the existing form factor is a little bit too small for the 8- and 16-bit applications. We recommended a bigger module, which SEM does not have in its inventory, but there is an R/AD program to develop a larger module. The 16-bit processor we felt was not ready yet.

One area of commonality between the 8- and 16-bit area that would be nice would be to have the standard instruction set in both CPU's. You will see this trend in industry now as, for example, TI has proposed to start a 9900 family with an 8-bit CPU which executes the 9900 instruction set. This proposal is based on the I^2L version of the TMS 9900. The bit slice modules stepped up that activity a little bit when the 2900 came out, and in a sense the 2900 family has become an industry-type standard.

I will show you a little bit of what we did on the follow-on procedures to look into the bit slice standards. We wanted to come up with a set of modules after doing some background work, so that we were able to use bit slice processors in several applications. The work was performed by Georgia Tech. We came up with six module types (Table 4-10). One of the modules was a RALU module which had several 2901 on it. Four of them were used as a 16- bit-wide slice with a carry look-ahead on the module with the 16 addressable general-purpose registers. The modules can be cascaded to provide a 64-bit capability. This module essentially is allowed to do computer-type functions. We also had an outrigger-type module which tied things together and was a loose-end type of module. The outrigger module had a priority interrupt circuit, shift and carry look-ahead and some circuitry to allow multiplication using the ALU's. In addition, there was a macro-instruction module with a mapping ROM for the microcode into the micro-sequencer and a micro/macro instruction register for decoding macro instructions. These three modules were basically the type that would be used to do computer-type functions such as emulations.

These three basic modules alone allow you to perform micro-programming-type applications without ALU-type computation requirement. There is a microsequencer module address capability to 64K of microcode. Very few people are using much beyond 1 or 2K now. There was a 16-bit register, so there was no loss in putting in the 16-bits that gave the 64K capability. You could possibly put an HOL in the microcode.

Table 4-10. Bit Slice Modules

RALU module

Selectable word length of 4, 8, 12, and 16 bits with full look-ahead carry

16 addressable general-puspose registers

Cascaded up to 64 bits in 4-bit increments

Condition code flags - C, N, Z, OVF

Performs computer-type operations

Outrigger module

Vectored priority interrupt circuitry

Shift, carry/link, and look-ahead carry logic

Condition code register

Dedicated multiply circuit

Enhances RALU and microprogram control functions

Macroinstruction module

16 bit macroinstruction register

256 × 16 bit high-speed mapping prom

Microprogram sequencer

120 nsec cycle

Next address calculation

64K microcode addressing

Table 4-10. Bit Slice Modules (Continuation 1)

Microprogram memory

- 16 bit microinstruction register - 4 sections
- 1K × 16 microinstruction PROM and memory select
- Microinstruction length horizontally cascaded in 16 bit increments
- Microprogram memory vertically stacked in 1K increments

Timing

- Multiple clocks - microprogrammable
- External event synchronization control
- Conditional branch control
- Master reset control

Also provided was a microprogram memory module that could be cascaded in 1K increments, and a timing module that had the microprogrammable clocks, external event synchronization, etc. These modules are currently in the breadboard stage and being used in applications for mode and control on FFT processors and a floating point processor.

Figure 4-5 shows a module set in a microprogrammed configuration of generalized bus structure with a microsequencer, a microprogrammed memory, a timing module, and the possibility of a macroinstruction module. There is no ALU module, although an ALU module could be in the math function module instruction set.

Figure 4-6 shows a functional diagram of a NOVA emulator using a module set consisting of the six basic types that we came up with. I do have data available on some of these modules. Another thing we did through CDC, San Diego, was an attempt to come up with a technology-independent scheme (Table 4-11) which centered about a standard I/O bus with fixed bus control lines, data lines and address lines (Figure 4-7). This technique would be able to handle CPU's of different technologies, the only difference being the interface from the CPU's to the bus. It would also handle different memories. This technique came out of a program we did using the 1802 processor, in which the application was message-processing and we were able to build up different memory modules, CMOS, TTL and NMOS memory modules and exchange these modules just by exchanging the drivers to the bus. This has also moved into the SEM R/AD area now.

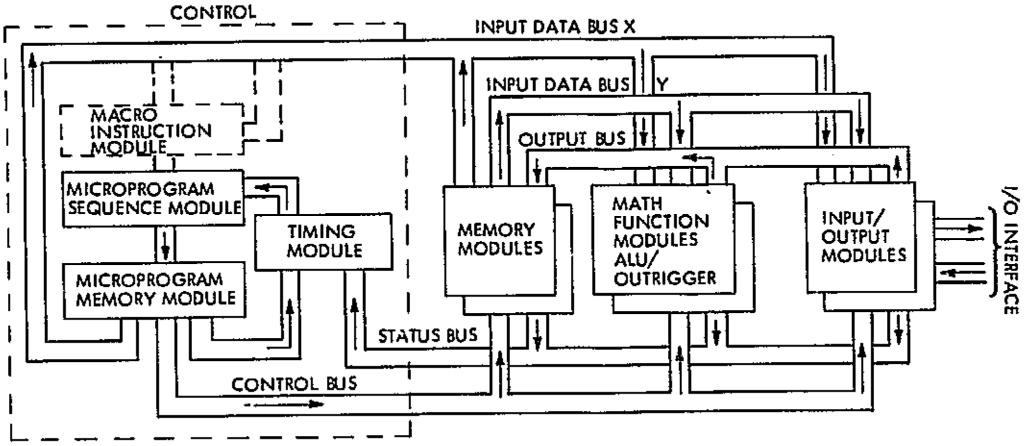


Figure 4-5. Generalized Bus Structure Using Module Set

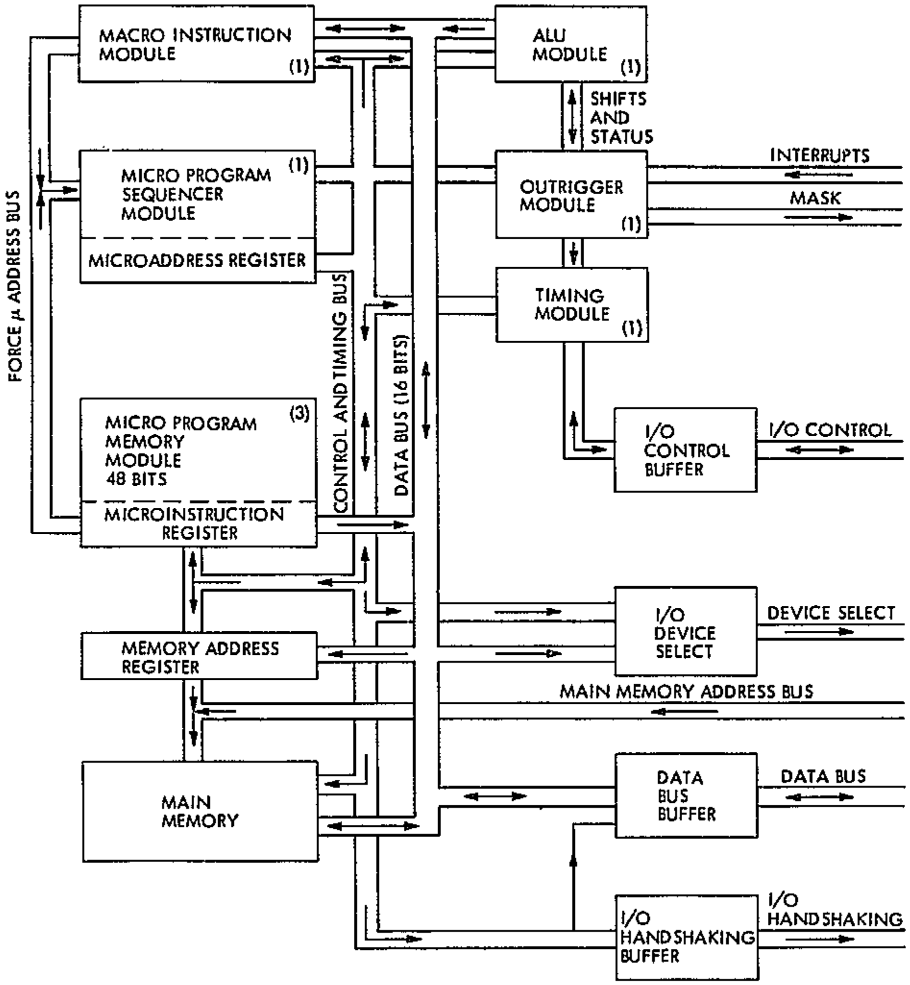


Figure 4-6. Functional Block Diagram of NOVA Emulation

Table 4-11. Technology-Independent Common Functional Modules

Universal bus architecture

Common set of microcomputer building blocks

CPU module

RAM module

ROM module

I/O module

Common system backplane/pinout/power/mechanical

Technology-independent modules

Used for 8-bit, 16-bit, and bit-slice microcomputer designs

Flexibility for multiple applications

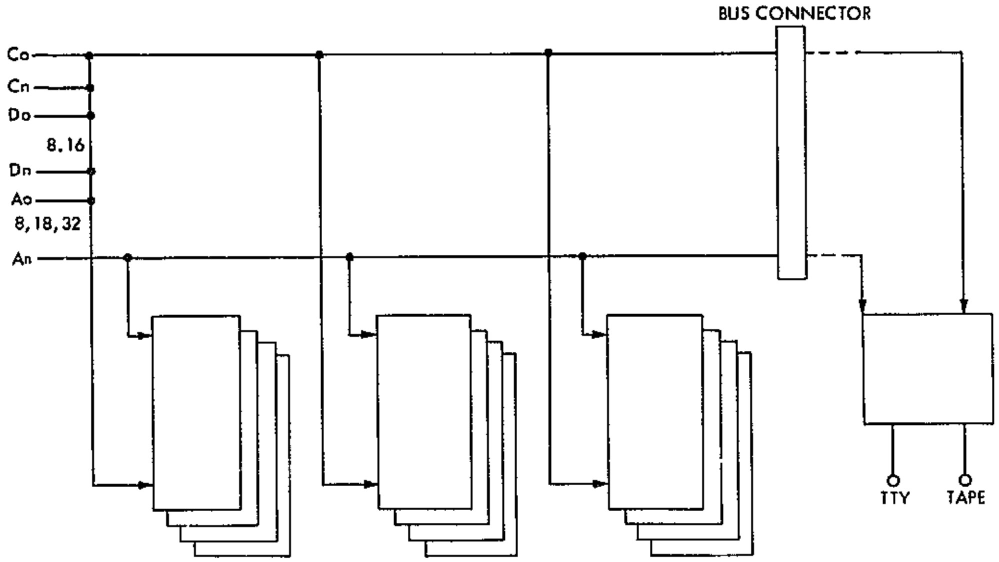


Figure 4-7. Comparison System Architecture

SECTION V

LSI STANDARDIZATION

David Haratz
U.S. Army Electronics Command

A. SUMMARY

David Haratz described this conference as a follow-up to the old triservice Micro Mod Program. The Micro Mod Program was an attempt to standardize the packaging of discretes which would likely be off the market before the conclusion of the program. Mr. Haratz pointed out that standardization is not necessarily always good, not always totally desirable. You have to be very careful as to what it is that you're trying to standardize. When you standardize, it is a give and take situation, something has to give. You must assure that what one gives does not counteract the advantage one attains.

The NASA standard satellite recorder was compared to an Army recorder. The components may be standard but certainly the functions are not. In the NASA recorder program, the recorder is put into orbit, and nobody removes the media because there is nobody there to remove it. On the other hand, in the case of the Army recorder, the Army removed the media memory tertiary program and had a problem with the removable media because they were putting different programs, different kinds of things, on the tape cassette. It used a tertiary store, so the basic function was different. When it comes down to the component level, there is no reason why the bubble chips or even the package itself could not be standardized.

B. PRESENTATION

The worst problem that the Army had on the removable media memory unit was not the 100-g shock but the dirt and filth of the environment itself. In the tertiary store, the Army cannot tolerate an open cassette, a $1.50 kind of cassette that could be bought from 3M. Thus, the Army went to great extremes to close off and capsulize the cassette. To give you an idea of what is left, look at the Gemesco cassette. They removed not just the media, which is ideally the way to do it, but the pitch, roll or the capstan, the motors and the drive. Before they know it, there's nothing left. They may as well replace the whole cassette. And that's the kind of problem they run into and its kind of a dirty air type of problem.

In a satellite, they don't have that kind of problem. But the technology, magnetic bubbles, or whatever, could very well be applicable across the board. So, at that level the standardization is at the component level. What is it they are trying to standardize? There was a great deal of talk here about standardization of form, fit and function; i.e., they take out a module and put in another module with the understanding that the user doesn't care what kind of components are in there so long as the interfaces are the same. Doing this in the Army would pose other problems. Which TM would they use to maintain it, and how many cams would they have to maintain it? The question then comes down to what is it they are trying to standardize?

There is also a question of goodwill in the whole issue of standardization. Goodwill. This was alluded to a few moments ago. One of my functions in my group is to act as a devil's advocate to various programs in the computer area, programs that are getting ready for production or in the latter stage of engineering development. There were certain changes that I advocated. In a particular EWE application recently, there was a proposal by the contractor to do two things: to fill his own machine and to hybridize the parts. Now those kinds of things pose problems. We looked at the characters required by the system and determined that the YUK30, which is basically a bit slice machine, could very well do the job and even though it isn't "standard" it does have an Army/Navy nomenclature. We proposed the YUK30, and found out immediately that functional requirements changed the system all of a sudden. For example, instead of a 15-microsecond multiply, it now had to be 5, so we knew the YUK30 couldn't do it. So you see the question of goodwill comes in there. As for the question of hybridizing, look at the box and I'll exaggerate a little bit for the sake of argument. The box itself, the package or whatever, may weigh 40 pounds, of which the thing we're talking about hybridizing may be 15 ounces or 12 ounces. Who's going to save an ounce or ounce and a half on the whole thing? But nothing is done about the power supply, which accounted for the majority of the weight.

We had a similar standardization problem with thousands of teletypes. The program links very closely with the SEM and QED programs of NELC. We used their parts because we had to demonstrate the reliability of using microprocessor componentry to replace the hard logic of the 74. It was actually a viable approach. We did this within a three-month period with the help of NELC and the QED program, which is basically an 8080 system on SEM modules. We did this in-house and there is now a contract to produce $40 million worth of pieces. The point to be made here is that there are various aspects of these systems and there are many, many teletypes that go out into the field. They are no longer teletypes, they are word processing systems with text editor functions. They contain microprogrammable ROM, which presents a serious handling problem. They go out into the field like radios; that is by the thousands. One of the features is header prompting; the operator is prompted through various headers. There are about 128 headers. One machine, labelled XYZ, may have header one, another machine header two. They'll be in ROM which the operator pulls out to change. The question comes up, is ROM hardware or software? We don't even know that yet so we call it firmware. When you try to support it in the field, we treat it as hardware because each board has a different function, and as we understand hardware, that is what it really is.

These are some of the issues that tie into some level of standardization. The basic issue that I think we are going to have to address to some extent here is, what is it that we are trying to standardize and what are the various levels at which we standardize?

We have been talking here about machine or computer architecture. There are other levels of architecture. For example, the SEM program is aimed at what we call a packaging architecture. Or you can put an architecture on packaging. There is mechanical and formal architecture and interface architecture. Standardizing on the computer architecture is only one phase of the problem. At that level and across the board there are different types of architectures that we are really talking about. I hope in these two days our session will address some of these issues and come up with answers to a couple of questions: At what level are we talking about standardizing? How do we go about doing it? Also, to what extent can the DoD and NASA really dictate to the vendor the degree of standardization?

SECTION VI

NONSTANDARD LSI APPROACH

Carver Mead
Caltech

I would like to offer a rebuttal to all the previous speakers. The form that this will take is to tell you a little about a course we run here to teach people how to design LSI systems, and then to show you some of the projects that people designed who are fresh out of this course. The reason for this is that I believe people think that designing in LSI is harder than it actually is. I want to talk about characterization of projects once they have been designed, about testing of parts once they have been fabricated, and then about reliability, about which I know very . ttle. But I think these topics have a direct impact on system reliability as opposed to component reliability. The conclusion that I want to come to is that it is system reliability that we ought to be thinking about, not component reliability, because we are past the point where there are any components that we can get at.

Designs always start with some people kicking around ideas. Then that idea gets implemented in silicon. The first thing that happens is a layout, that is, a reduction of the function which you are implementing into a geometric design (Figure 6-1). That geometric design is then converted into some language of otherwise machine-understandable form where you can use aids to help you construct a system out of the various building blocks that you design in that way. Figure 6-2 shows a machine translation of that information into a piece of film. In this particular case that piece of film is 125 times the size of the final chip. The next step is to convert that piece of film into a small piece of film, which in this case is 10 times the final chip size.

Thus you have designed an integrated circuit, an integrated circuit of sufficient complexity that in fact it is a system or subsystem in its own right. This 10-times final size thing is put into a piece of machinery at the mask makers and it is repeated several hundred times. Figure 6-3 shows the result. The result of that is a piece of glass with a pattern on it. You will notice about how big each particular circuit is. It is about one-half inch across, and it is repeated across that piece of glass in two dimensions. The form of two-dimensional array of the particular chip that we are talking about and that particular piece of glass is one of a number in the particular process that we use. There are five or six layers which define the function of the integrated circuit. When you are done with all that, as seen in Figure 6-4, those pieces of glass are used in a contact printing process to photographically engrave in each of the five layers, and to define the geometrics of each of those processing steps, and when you are done you end up with a silicon wafer. The wafer has on the surface a number of individual circuits. Each of the circuits nowadays may have as many as 50 thousand transistors on it.

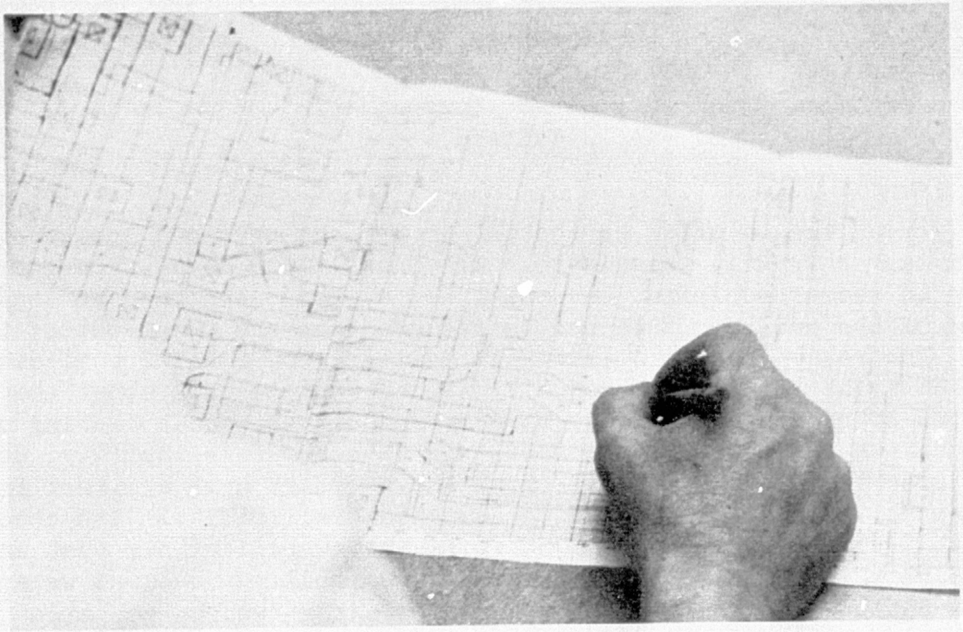


Figure 6-1. LSI Geometric Design Layout

Figure 6-5 shows silicon gate MOS technology. We have students going out of this course that design in probably 10 different technologies today, and the discipline is very similar, so the technology we have chosen is particularly nice because it has a small number of layers and it has other attributes having to do with testability that I will talk about later.

These chips are cut apart along these lines. You will notice the highways between the blocks here looking down on suburbia, these are lines and they are used to cut grooves in the silicon which are then used to break apart these individual circuits which are then stuck into packages - the familiar little bugs which crawl around on 16 or more legs. You will notice a couple of things about this picture. One is that there is a test chip on the wafer which has test patterns on it. These are used to characterize the process. Rather than to characterize the particular system which exists on the neighboring chips, this is probably the only way that real LSI, as time goes on, is going to be qualified for use in high-reliability environments.

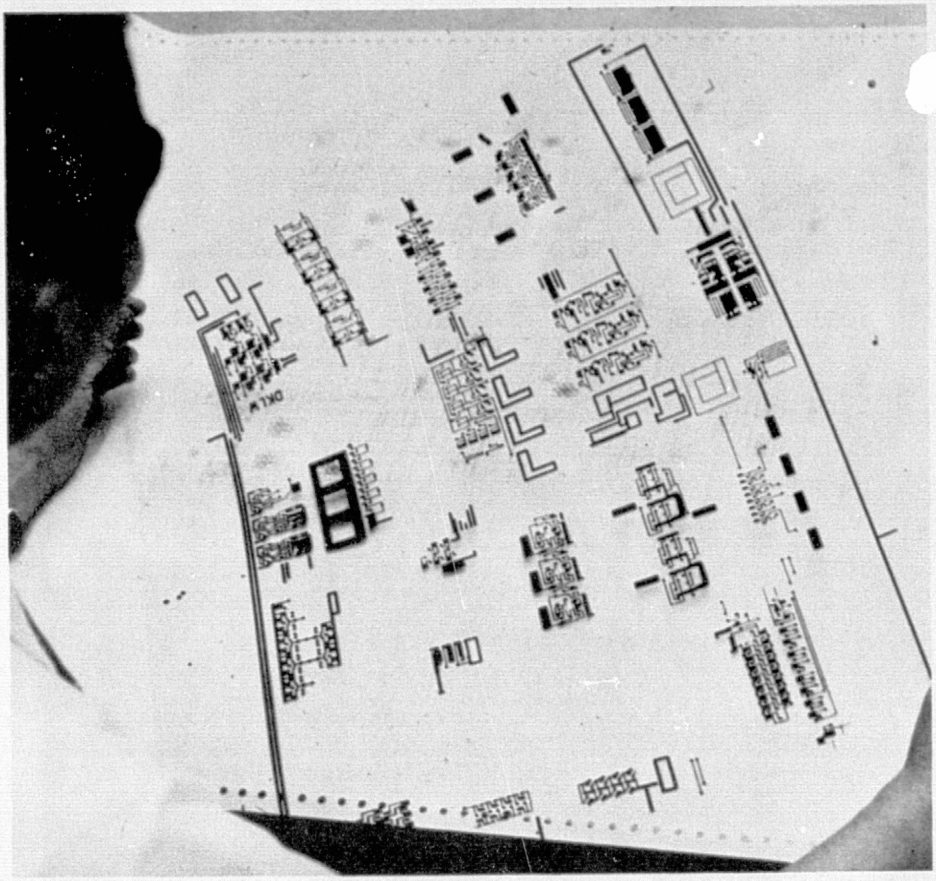


Figure 6-2. LSI Film Process

Finally, the chips are put into a package and bonded. The system is of the order of 50 thousand components and is packaged with as few as 16 or 20 or a few tens of pins. Now as testing people could tell you, there is a thing about testability having to do with being able to get at the internals of the system. That gets more difficult the harder it is to get at the internals, and that is in fact a key problem in LSI. It is impossible to get at all the internal components, and in fact it is impossible to test or even exercise all the internal components, if you count the potential parasitic components which are often present in any technology. That is why it is a problem when you talk to people that require high reliability or require some assurance of high reliability.

Everybody requires high reliability; it is just that some people pay more dearly for its absence than others. When you get one each copy of something you are going to send to the outer planets, it is very serious if 10 years later you find out there was a problem on that particular chip that you had not managed to flush out before you sent the thing off. That is a problem we do not have in the laboratory, where our friendly local computer breaks down and we can go in and replace the chip and it begins to run again. It is embarrassing because

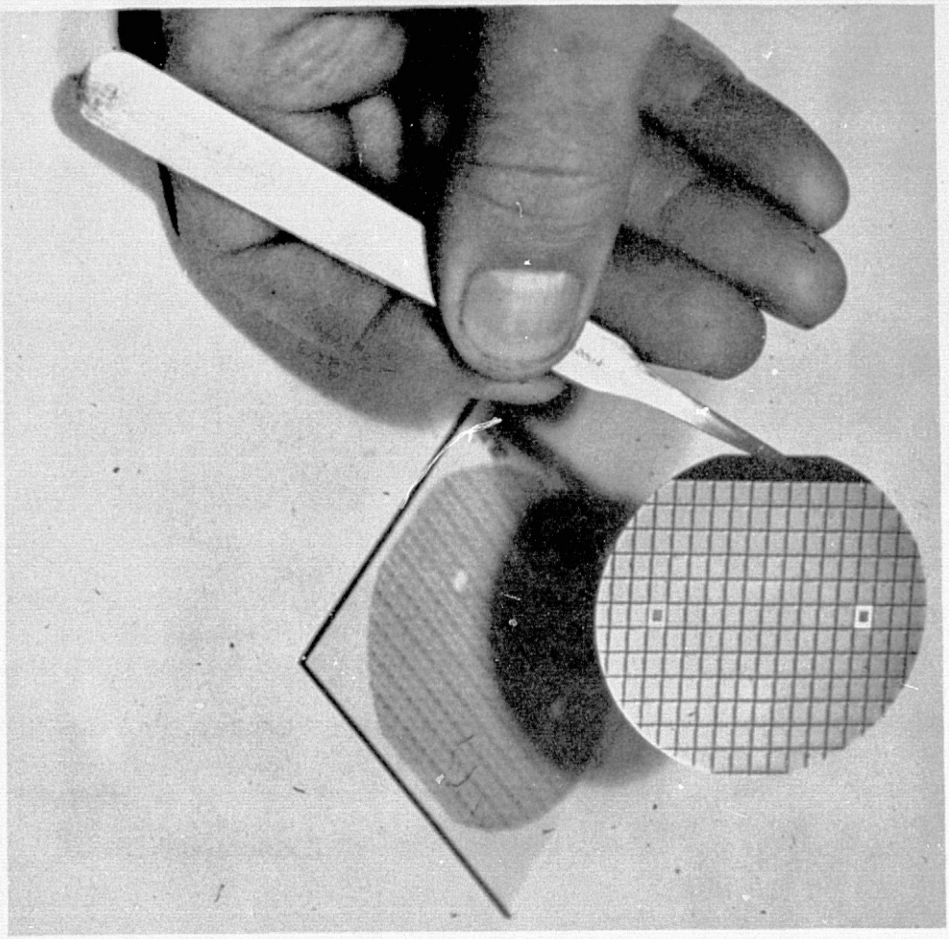


Figure 6-3. LSI Mask

you lost a day's run and you may have lost a day's time while you figure out how to fix it, but you did not lose the mission. On the other hand, things that are monitoring heart patients have some of the same kinds of reliability issues that you have in sending things off in the spacecraft.

Figure 6-6 shows a typical "Project Chip" that was made here at Caltech. The first exercise in the class is to create a small chip of your very own where the design team is one person and what you can design in a month and a half by yourself is the size of the chip that you make. So in fact, the particular integrated circuits that these people are making come in all shapes and sizes and styles. In fact there is a great difference in the style in which things are done by different designers, and if you look at chips from various manufacturers you notice they have the same property. Different people believe in doing things differently and when you look at the chips you see in fact that everybody believes his way is best and for him it may very well be. The typical functions that get created here are sort of MSI scale functions at this level.

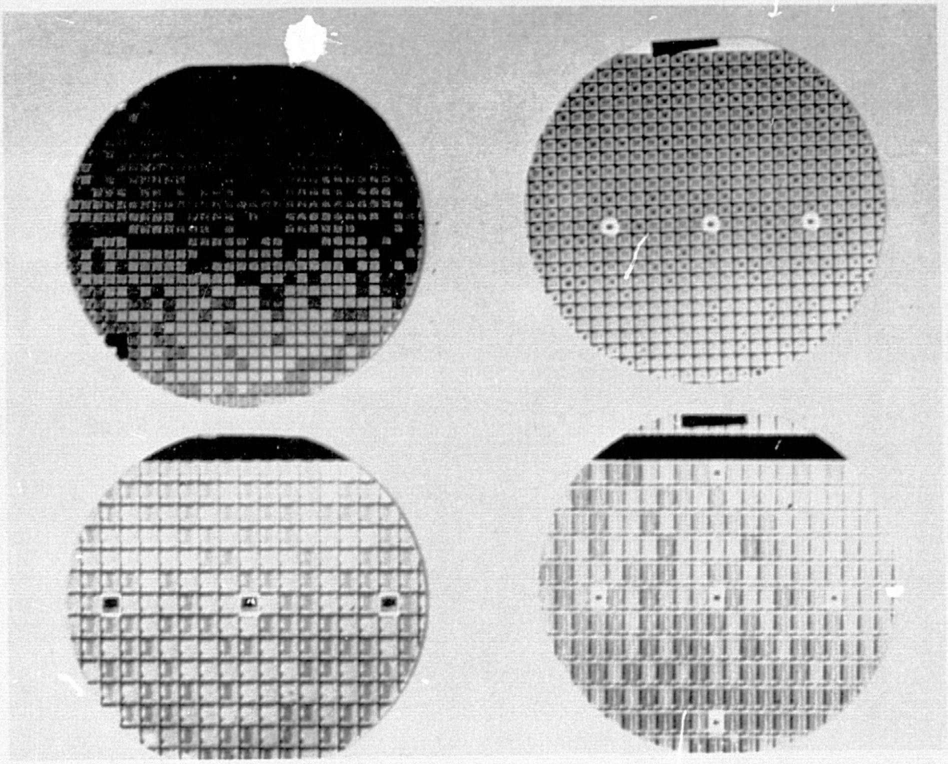


Figure 6-4. Glass-etched Circuits

We do not have a fabrication area ourselves. We get the processing done at a number of places. The quality of processing that one can expect out of high-volume commercial lines is very good, but there are several questions. Given that you have satisfied yourself that someone's line is particularly good, how do you interface with it? How do you get access to a line like that? They are not going to let you come in and take captive that line, because in fact they are running a very high volume of commercial parts through. But because they are running commercial parts, they have dedicated some very smart people to watching the line and being very careful not to get contamination in the line. They are constantly monitoring and watching over it. If they are doing that, that is a great advantage. They are doing just exactly the sorts of things that you would have to spend a lot of time and energy and money doing. It will be very nice to take advantage of that. How do you get into a line like that, and use it, take advantage of it? How do you satisfy yourself that in fact all the words I just said are true at all, or did they foul up half way through and spit on the wafer? You won't know that until you send it off to Mars.

We take full responsibility for a set of masks that we in fact generate and inspect. What we hand the wafer fabrication area is a set of working plates which we have satisfied ourselves are good. What they hand us is a set of wafers that are hot off the end of the line and have not been inspected, tested, or touched by human hands. It turns out

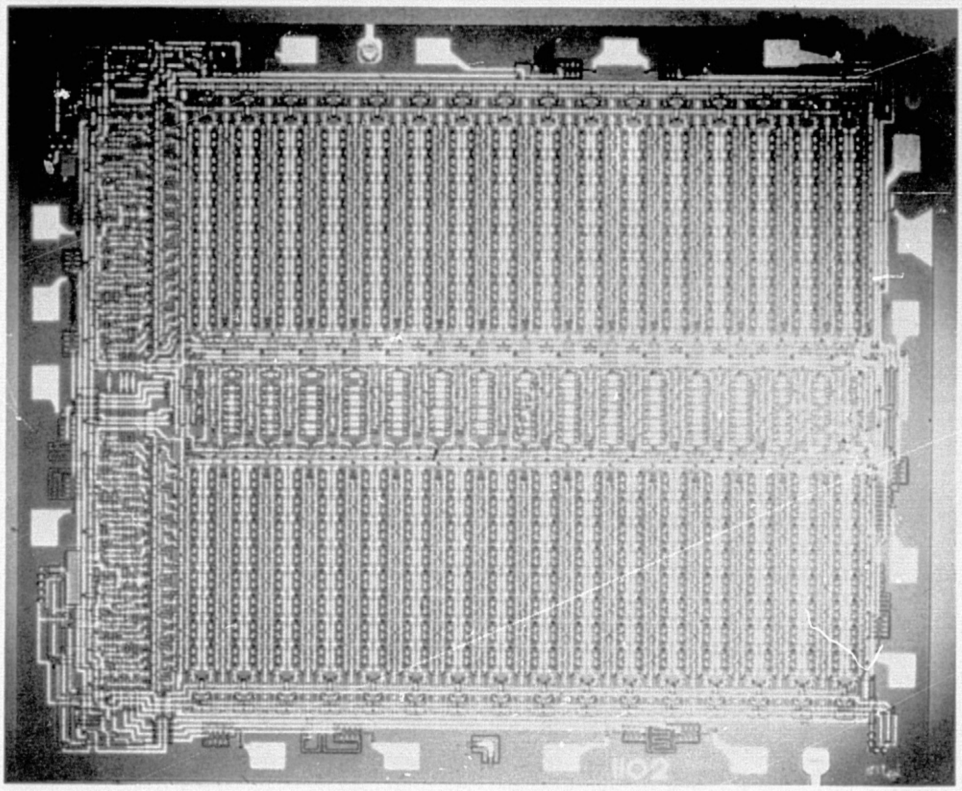


Figure 6-5. Memory Chip: MOS Dynamic RAM

that as near as I know, that is the only way that some one can interact with a fabrication area. As a matter of fact, people within semiconductor companies themselves find it necessary to interact with production fabrication areas in that way, and it is a very clean interface. You must have a mechanism for verifying what they did while your wafers were in their fab area. We will talk about those mechanisms in a little bit. But that is how we do it. And we have been very successful at getting back very high-quality fabrication on some very complex systems with that format.

Figure 6-7 shows part of a larger project. This circuit is a processor. It is a particularly simple processor designed to compute memory addresses for a smart memory. This was a partitioned system, partitioned to put some of the smarts of the processor on the memory card along with the memory chip. We did not build the memory chip, but we built the controller that in fact generated the addresses and the read/write commands and talked to the memory chips themselves and only bothered the rest of the processor when it had an answer. And this was the chip that in fact did that computation. This next chip (Figure 6-8) is a pipeline multiplier chip designed for doing very rapid signal processing multiplications. Most signal processing kinds of applications are sort of what you might call fixed algorithm, variable data kinds of things. What you do is set up an algorithm like a fast Fourier transform or a set of filter functions. You may

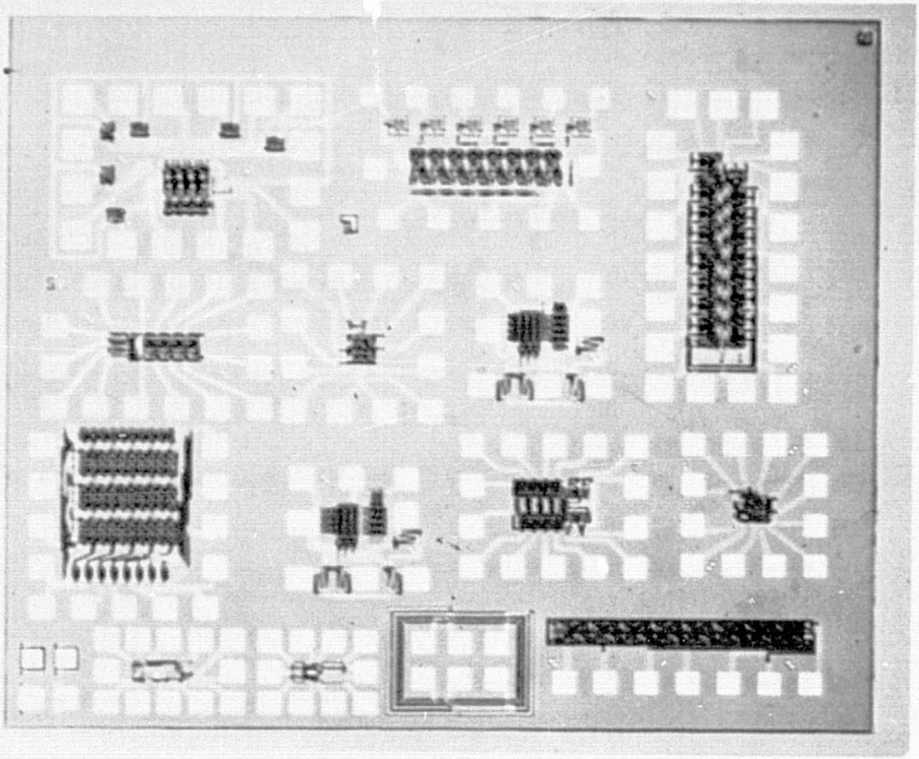


Figure 6-6. Caltech Project Chip

in fact change the coefficients of the filter functions, but you do not change with time the geometry of the filter, the number of poles or the way in which it is organized. This is designed to do a complex multiplication. It is a rather nice thing. Figure 6-9 shows an arithmetic logic unit, with the sort of complexity of the ones found on current microprocessors. It has registers, control, instruction decoding and such, but without the stuff to do memory addressing. The memory addressing was on the chip shown in Figure 6-10. What we have shown was an attempt at sort of distributing the intelligence in a microprocessor.

These are small chips. This chip, for example, had only about 10,000 transistors on it, so it is a small system or a small subsystem. The ones we are building nowadays are quite a lot larger than that. As time goes on, the technology has been advancing at such a rate that we can create systems with approximately twice as many transistors on them this year as we could last year, and achieve the same yield. And that factor of two per year has been going on since 1960 or 1961. So, like I say, there are chips with reasonable yields in excess of 100,000 transistors on them, and you very regularly make things with many tens of thousands. These are becoming systems of reasonable size. When you look at the fundamental limitations on how small you can make things, it appears that there is nothing to prevent us from making our chips with 10 million devices on them. That is not going to happen overnight.

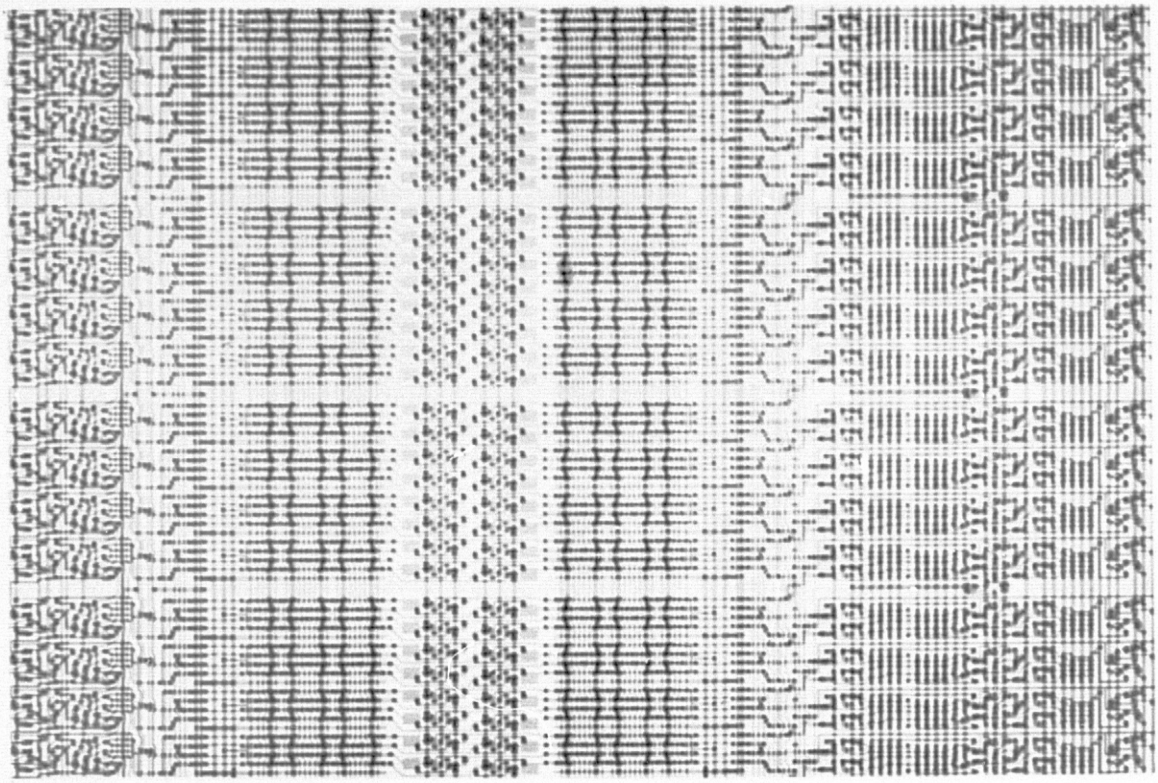


Figure 6-7. Microcircuit Processor

There is a lot of work between here and there but there is less between here and there than there was from 1960 until now. Today, essentially all the individual steps in making things of that size have been accomplished separately. There is no running process today that will produce submicron transistors, but there are processes which will do each operation required to make submicron transistors. And the historical experience has been that all the sort of measurable bad things about integrated circuits, the reliability, the yield, all those things, have not changed very much as the complexity of the devices has increased. In other words, if you take a microprocessor today it costs about the same, has about the same life expectancy, has about the same reliability, etc., as the single logic transistor did when it was first produced.

So this is the kind of thing that happened. You have all been taught in grade school that there are two kinds of chips. There is one that is called the memory and it is very regular in its organization. Figure 6-11 shows the other kind, which is not a memory and is very irregular in its organization. This is an 8080 microprocessor. You can see there are small areas of regularity in it, and the rest of it is spaghetti! This is the way that industry designs parts today. Some people introduce more regularity into the spaghetti than others, but in fact, most of it is pretty much like what you see there.

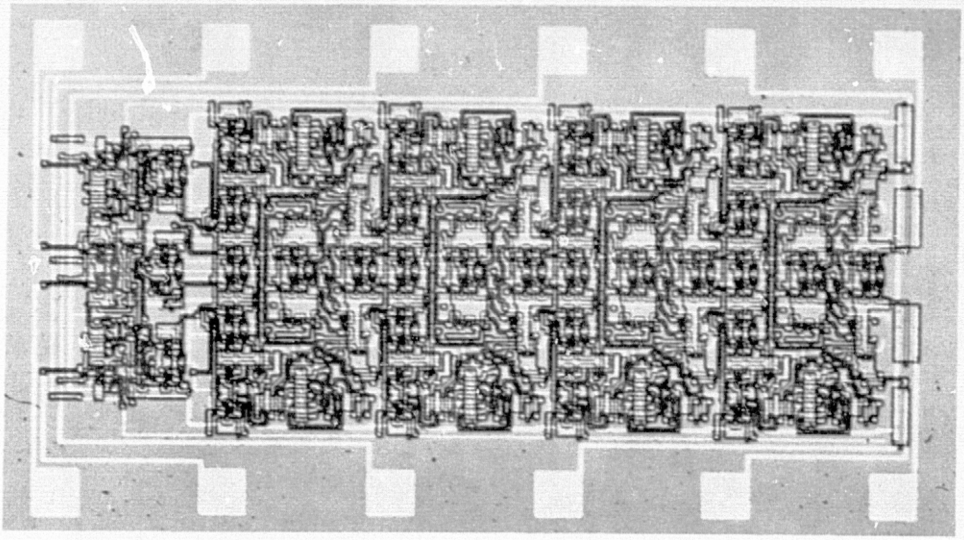


Figure 6-8. Multiplier Chip

I have comments to make about design methodology, and one of the comments is with regard to what I call characterization. By characterization I mean figure out what the part really does once you have designed and built it, but before you are in production. You are just trying to figure out what it does, and it may not do what you thought it was going to do. Testing is different. Testing is taking a known good design (that is one that has been characterized) and assuring that all the unintended features have been put into the instruction manual for the device. Now that sounds like a very easy thing. With AND gates you can put all the desired, all the possible combinations of input on the pins, and in fact because there is no storage state on the device itself, you can exhaustively test every possible combination.

What is the problem in LSI with regard to characterization and testing? Well the problem is as follows: Suppose we have a 4000-bit random access memory, which is the standard today. Everybody knows that memories are simple. This ought to be easy to test on a new tester which runs a test every microsecond. So one ought to be able to test 4000-bit memories in a few milliseconds. Well, yes. In a few milliseconds you can put a bit in each cell of a memory and then read it back and see if you get the same bit out that you put in. And you can do that for a "1" and you can do it for a "0" and you have tested the memory. The only thing that can go wrong with a memory is that it fails to store "1" or a "0". And in fact people do that and they catch all the chips that fail to store "1's" or "0's." But then you notice that many of those chips that you have tested that way do not work when you get them in the system. And so you go looking at what is wrong and after a lot of exercise you find out that the bit may store a "1" if all the surrounding bits are "0." It may store "0" if all the surrounding bits are "0" and it may not store "0" if all the surrounding bits are "1's," or if one particular surrounding bit is a "1." And what may have happened, for example, is that on the mask there was a little fleck of

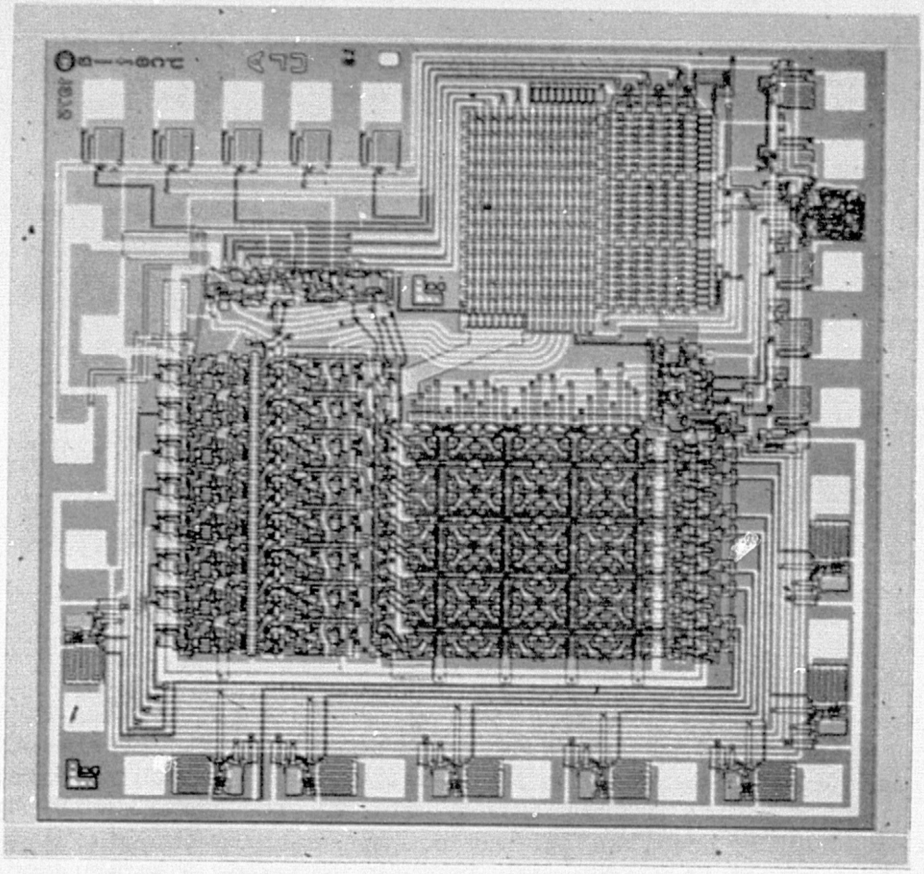


Figure 6-9. Arithmetic Logic Unit

dust right in the place which separated two particular areas of the integrated circuit and they got too close. And when they got too close what happens is they electrically affect each other so the leakage of that particular device is greater than it would have been otherwise. Furthermore, it depends upon the voltage on the wire that is next to it, so that when the voltage on the wire is low the thing acts normally, but when the voltage on the wire is high, there is a punch-through condition and it drains off enough current to cause the thing to not store a bit. If this sounds like imagination, I should point out to you that a substantial fraction of the rejected memories that get thrown in gold scrap up at our friendly semiconductor house suffer not from the simple kinds of failures but from these so-called pattern-sensitive failures. Some kinds of designs suffer from them more than others, but all designs suffer from them to a larger or smaller extent. So you say OK, that is easy, I will exhaustively test the memory. I will put every combination of bit in the memory that you can have and I will see if I get the combination back and if I do, I have tested it. It turns out that takes 2^{4096} operations and, at a megahertz, that is much longer than the age of the universe.

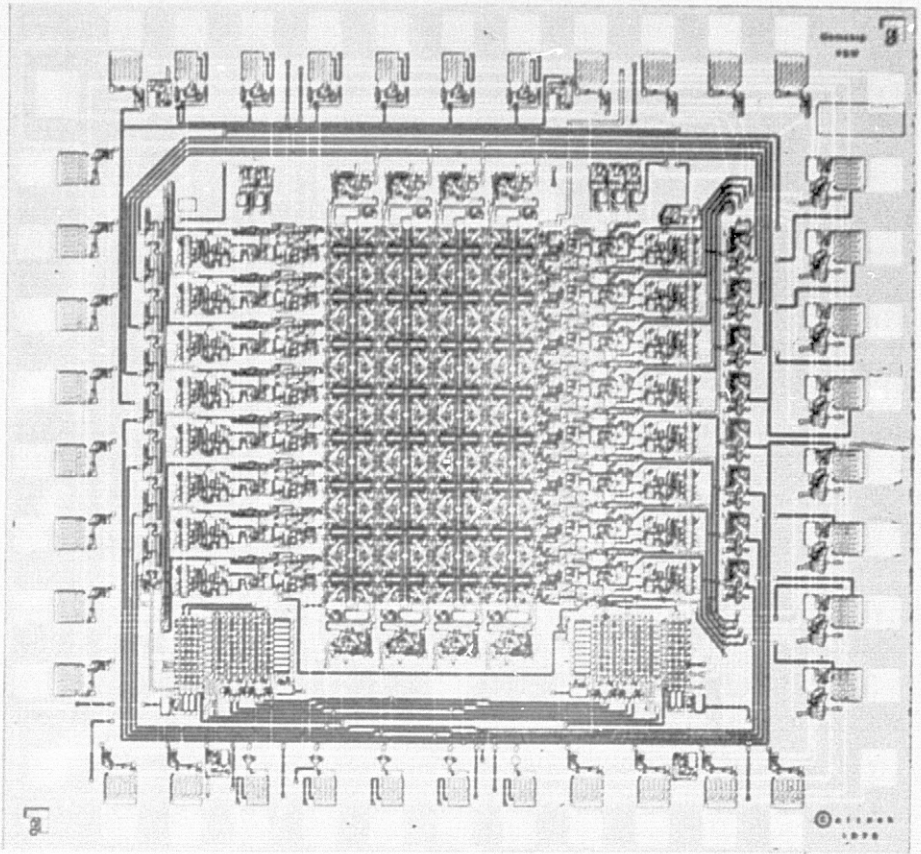


Figure 6-10. Memory Address

A 4000-bit RAM is of the order of complexity of that 8080 chip: about the same number of devices, roughly within the factor of 2 or so. And what it says is that if that device was working in a system, in the entire life of the universe, not just the life of the system, it would not be possible to have exercised that device to all the states that it might assume. Therefore, it is not possible to satisfy yourself that in fact it will do the right thing in all possible states.

Now let's apply that to a microprocessor. How do you test a microprocessor? You test a microprocessor by putting it through a test program. What the test program does is execute each of the instructions on some data and then it checks to see if the resulting data is the right thing. You may decide what the right thing is by comparing it with the resulting data of some other chip that is known to be good. What is known to be good means that you have done that long enough to where you feel good about it. You do not know a thing about that circuit except that it has gone through a sequence of operations of your devising a number of times without being different from the thing you thought it was going to be on that particular sequence of operations. And in fact, that particular sequence of operations is one part in 10^{4000} of the possible states of that machine. So what you have done is

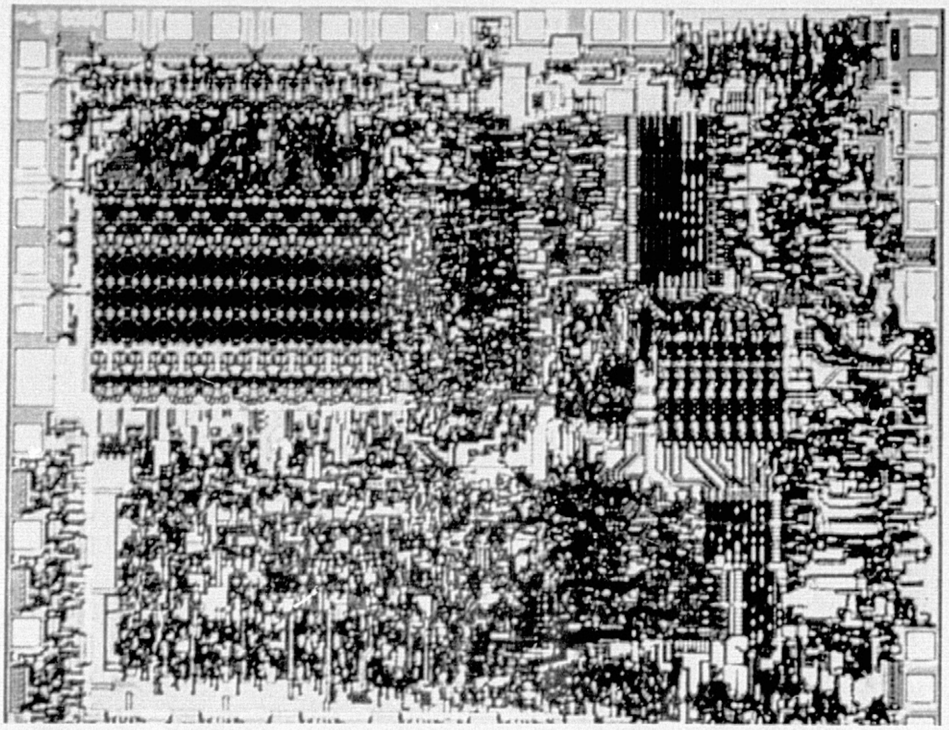


Figure 6-11. 8080 Microprocessor

a sampling, like people do when they are doing quality control on matches. They do not do 100% quality control test on matches or flash bulbs or things like that. What you do is pick one once in a while and test it and if that works you believe that all the others will work and you have not the slightest shred of evidence that that is the case; you just believe it. By the slightest shred of evidence I mean you have an expectation of one part in 10^{4000} that that is the case.

If things were really that bad, of course, it means that life is hopeless, and we might as well all give up and go home, because the minute somebody plugs one of those things in a system, he will exercise states in the machine that we have not been able to exercise in the test program. For example, if you are running a spacecraft, you know that bits that flow through a machine depend upon the particular location of the machine, the particular message which is sent from the ground, the particular attitude of the spacecraft, how bright the sun is that day. There are a lot of real-time inputs which in fact cause data that is different from any possible test program because you do not know everything that is going to happen aboard the spacecraft until it is up there.

When you are monitoring a heart patient, you do not know when the heart patient is going to go into fibrillation. If you knew that, you would not have to monitor the heart patient. The data which flows through the processor depends on whether he is in fibrillation or not because they are digitizing the waves that are coming out of the EKG.

The bits in that waveform and the bits in the data stream are in fact a reflection of something over which you have no control. So there is no way in any of these applications that you can create a program which will exercise the precise states in the machine that will be encountered in real-time application.

For that reason, you know for a fact there will be states in the machine that happen, that are exercised under real-time application that you have not tested. And you must believe, you must cause yourself to believe, that it will still work under those conditions. Now how in the world do we as engineers, who are supposed to be rational people, cause ourselves to be hookwinked when the probability that we have exercised a precise state that the thing will be in is one part in 10^{4000}.

Well we do it by the time-honored method of divide and conquer. We really do test at the component level. And what do we mean by that? We mean that we look at what is inside the chip, the memory for example, and we say, ah ha, that memory is interesting. It is organized into rolls and columns. Now it is possible that there exists ESP between cells on the memory and that one of them can cause the other one to get nervous and forget its contents. That is a possible model for the way memories work. But we as rational engineers are going to discard that model; we are going to believe that they really work with electrons, volts, and amperes and things. We know enough about solid-state physics to know that the distance over which electrical things propagate has an extent which is set by the doping of the semiconductor and that sort of thing. And if there are defects like hair that stretches across a chip you can look into a microscope and see it because it goes from one corner of the chip to the other. By definition it is big enough to go from one corner to another and therefore by definition big enough to see. So we look at the thing carefully to make sure whether someone has dropped a piece of bonding wire down on it which stretched from one end to the other. I understand that there are people who like to shake loose some of the eutectic that holds the die down and shake it on the top of the chip and sort out the nonadjacent things. That is a favorite way of causing failures. But in the absence of things like that we have a model, and our model is that electrical influences or small mask defects that we are not able to see can cause things in a row or a column to affect each other, perhaps between the nearest neighbors, or even the next nearest neighbors. But they introduce a locality into the things which can hurt you, and once you have done that, now you have changed a combinatorial problem which results in numbers like 10^{4000} down to a linear problem. I have to test each memory location with different bits with the things in its row and column and the nearest neighbor. Those things do not run in milliseconds, but in hundreds of milliseconds. And we call those exhaustive because under the particular model that we have in mind of how failures occur, we have satisfied ourselves that we have tested against that class of failures on theoretical grounds, on analytical grounds, not on brute force grounds. And that is the way we have gotten from a probability of one part in 10^{4000} of catching bad chips to one part in 10^{4} of not catching a bad chip.

So we have reduced the probability of missing bad chips in our test by a factor of 10^{4000} by using our analytical capability, by knowing the way semiconductor physics works and the way the chip is designed. And we have done that by building a model, and we all know, having spent time in the scientific world, that building models is a very powerful thing. It allows us to use analytical capability to do this rather than brute force.

Now how does this apply to the 8080 chip that I showed you? Well, it applied to the nearest neighbor things. We know that of pattern-sensitive failures the most likely ones are nearest neighbor failures. We know where every transistor is and who its nearest neighbors are, and we can check somehow that each transistor performs its function when the nearest neighbors are in all of their possible states. Then all we have to do is to know where all the transistors on the chip are, and we have to know where each of the nearest neighbors of each of those transistors are, and we have to find a way of deciding if that transistor is on or off within the context of the function it is performing. And then we have to check if it does the "off" OK and the "on" OK in the presence of one and zero in each of the neighboring bits.

Now unfortunately that test has never been performed. That is a simple model. Memories are tested much more exhaustively than that model. And there is fallout at the end of the test from reasons that would not be caught by a simpler test because people do not test memories for the fun of it. One of the bottlenecks on a semiconductor line is the testing machine. Everything else you do on a wafer is done to three or four hundred chips at the same time in parallel, while testing is done to each chip one at a time. For that reason, it is very expensive. It is a bottleneck and people reduce the testing time to the very smallest amount possible. So they do not overtest; in fact you people would say they undertest greatly. But in fact, they are tested to a model which is more stringent than the one I just applied to the 8080, and I can categorically state that no one has ever done that test on that 8080 to that model. I suspect that it would take some large number of years to construct such a test for the 8080. And the reason is that the 8080 is a very random device; the wiring in it is very random. The state and data information is mixed and crossed over itself in strange ways. It is very difficult to create a bit map of the device, but even once you have the bit map you are not close to a test, which in fact will go in and tell you if each individual transistor is doing its job.

In fact, that particular thing has never been done for that microprocessor, not even close. What I am saying is that the factor of 10^{25} that we consider only right for the semiconductor companies to test their memories to, we have never come anywhere near that with processor chips. Nor will you ever for a processor chip like that.

Now let's make it more interesting. Suppose that I design the processor chip. Your task now, in order to test the thing to the same level of confidence we have in memories, is to make a bit map. That is, to find every transistor or functional element on that device and to find its nearest neighbors. This would only take a couple of years.

It would only take twice as long as having built the chip in the first place. Once you have done that, devised a test that will in fact not only wiggle that transistor in its one or zero state, with the surrounding devices in different states, but will decide if it did its job when you did that. My guess is that it is physically impossible to construct such a test for the 8080. In other words, the number of operational codes in the 8080 are decoded in such a way that it is almost certainly impossible to take and independently wiggle nearest neighbors in many situations. In fact, I would be willing to bet you 10^{22} to 1 that it is not possible to construct the test that I just described. It almost certainly would be easier to design a new 8080 than it would be to figure out how to test this one. That I can state with great certainty.

We are in the stage of building a processor which has a great deal more power and speed than the 8080. It is a 16-bit device with a lot of capability. It does not look like the 8080. In fact everyone in "Silicon Valley" to whom I show it says that looks like a memory. Well, it is not, it is a processor. It is a processor with a much richer instruction set than the 8080 and much greater speed. I do not think I have to tell you that it is much easier to construct a test sequence that does the thing which I just described for this machine than it was for the 8080. The reason is that this machine was designed in a very regular and structured way and when you design a machine with that philosophy you create a thing where you do know where the nearest neighbors are and the combinatorics problems are all local combinatorics, not global combinatorics any more, and you can therefore go in and do the local kinds of things we talked about much more easily and with more certainty than you can with a random design. So testability and characterization are two of the reasons for doing structured designs of this sort. It also turns out that when you are careful in doing a structured design, you can get a great deal more function in a great deal less space and make it run a great deal faster, but that is a by-product. Even if that were not so, I would urge everyone to go about designing things this way because if you do not, you simply have no hope of ever being able to test it against a reasonable model for failure.

Suppose you have designed things in this way. When your design engineers deliver the mask, they also deliver a program which will test every element in the processor to the model I just described. It is possible to do that. It means that in fact the engineer may have to put additional things in the processor to allow him to wiggle the bits independently; e.g., there may be a flip-flop that sets the processor into a test mode, and it may be that in test mode you have the state information. The internal state information that normally you can get at may be accessible through the pins, where normally it would be the data coming in and not what went through the pins. But in this special mode you get the internal secret state information that you cannot get in any other way. I am not talking about everything checking itself on a local scale. I am talking about building in a certain amount of accessibility to state information so that you can go and see it and then run another cycle, and then go look at it again and therefore get a window on places in the machine that you cannot stick probes down but you need probes there to test it. But I can tell you that although we do not have a perfectly good algorithm for doing that, there are many things that help greatly that do not occupy more than 10 or 20% of the area of the chip. So it is possible to do much with a small amount of overhead provided you design it in rather than do it as an afterthought.

The next thing I want to talk about is supposing you have designed your chip that way, which is the only way you are ever going to have any confidence that your chips are reliable or that they are in fact operating. Since you have no reason otherwise to believe you have exercised even a small subset of the possible states on board the system and you cannot get at the internal points inside, then you must do the divide and conquer thing in order to have any confidence whatsoever that you have exercised a reasonable fraction of the state space that you are going to be looking at against some model.

Suppose you have done that, and suppose you have convinced yourself that there is a test which will show, under a certain class of models of failures, the fact that the thing is working. Now what do you do? Well, it turns out that even such a test is a very poor window to decide the long-term reliability properties of the underlying process. In other words, looking at an 8080 or our machine or anything else through the particular 40 pins or whatever were provided very often gives you no idea at all as to how to extrapolate what the thing does under radiation or under bias temperature stress, or whatever. Except if you bias temperature stress the particular part until it fails. Then you say, yes, I really did cook it at 250° for 5000 hours, and it is only after that that it failed and therefore it would have survived the trip to Mars all right if I had used that one. If you can test to see that the device works and that it works with adequate margins, how do you satisfy yourself that latent failure modes are not built into the oxides and the metallization? Well, the way I propose to do this is through the use of test patterns. These test patterns are sufficient to characterize rather completely against a very, very wide class of models of how you can make silicon devices fail. It is adequate to characterize very, very completely whether the process by which that particular wafer was made would result in parts which are reliable against a given set of stresses, given that the design was correct, and that there are not any latent mask defects. And this can in fact be tested to destruction in a lot of ways. You can get at a lot of the parameters which couldn't be gotten out through the pins of an 8080, for example. And you can satisfy yourself that the individual elements which went into your design are in and of themselves reliable on that particular wafer - on the exact wafer that the 8080's or whatever you are going to ship in the spacecraft are from - because in fact this chip is an insert, a test chip, in the sea of processor chips.

So, what you have here is a window into the process by which the device is made. And what you have in the design methodology is a way of convincing yourself that the design is correct, and what you have in the testing methodology, which includes some design also, is a way of satisfying yourself that the particular chip is an instance of the chip that you designed originally. And I believe that this is the only possible way, by using those three interrelated pieces of information, that you are ever going to be able to convince yourself that systems made on silicon chips with many tens of thousands of components on them and potentially with millions of components on them are operating at all, let alone reliably.

SECTION VII

LSI IN THE AIR FORCE

John DeCaire
Air Force Avionics Laboratory

A. SUMMARY

Mr. DeCaire, whose position within the Air Force is R&D oriented, is trying to get LSI into real systems, but there is a time lag of 8-10 years between new technology and systems in the field. He feels that the real technology drive is dollars. As the supply of money decreases there is more and more emphasis on cooperation within the Air Force and with other agencies.

He feels that standardization is a management approach, a way of doing business. Done properly, standardization is a good way of doing business in the Air Force. The basic principle in the Air Force is to use available LSI and pay for the qualification. If standard devices are not available and custom devices are needed, the development has to be paid for. The justification for custom LSI is the trading of development cost for reduced production cost. In large volume this is an easy trade. With small volume the trade is not so easy.

Mr. DeCaire feels that programmability of software is a driving factor. It is the hidden 90% of the iceberg. It must be addressed in the future. The short-term solution for AF microprocessors will be the 8-bit monolithic devices available now. For more capability and speed, they are investigating bit slice architectures. They have an ongoing program for the development of a very capable 8-bit slice General Processing Unit (GPU) or CMOS SOS. The Avionics and Materials Labs would like to cooperate with other centers on the characterization and utilization of this device and its related family. The Avionics Lab has surveyed potential uses and projected future needs in defining the GPU family. It appears to be a good candidate for standardization.

Radiation hardening is one of the more important factors in Mr. DeCaire's programs. Other important factors are availability, surveyability, reliability and maintainability. The various Air Force programs appear to be well-funded and well-conceived, and Mr. DeCaire is interested in interacting with other centers.

B. PRESENTATION

I see that I was billed on the program as speaking for LSI and the Air Force. Since we are being taped I am going to qualify my remarks right away by saying I do not pretend to speak for anything more than the organizations of which I have charge. I will say though that those two organizations are the mainstream elements in the Air Force's approach to large-scale integrated circuits. I guess I would be remiss if I did not at least say in a public relations sense that the Avionics Laboratory and the microelectronics branch of which I am a part are where the whole business of integrated circuits started. At least so I'm told; that was before my time. But it is interesting to at least look at something from an historical point of view. In the early fifties and sixties, when we were starting integrated circuits in the Air Force, Texas Instruments, Fairchild, Motorola, General Electric, Westinghouse, RCA, etc., were crawling all over themselves trying to get our money. They still are, but the people who now crawl after our money are in the equipment group of those companies, not the semiconductor group.

The equipment groups have the same problems with the semiconductor groups, even within the company, that we do. Their only advantage is that they get the parts at cost instead of a negotiated price. Somebody commented this morning about what is changed. Well, profit and loss in the sheer volume in the semiconductor business have changed entirely. And of course that has forced a change in our policy of development. The point that I want to make is that dollars are now the universal control. I'm sure it has not always been so. Company profits in the semiconductor business are driving and that means a difficult design, a long, expensive design, clever designers and sheer volume of manufacturing. The assembly line syndrome, if you like. It seems to be inherent in our American way of doing things that the assembly line mass productive scheme is regarded as the eventual way to low cost items.

Somebody mentioned this morning that the automobile industry is going to take care of qualification. There are two ways to do it. One is to build it in, the other is to test it in. And if I understand the way the automobile manufacturers are going about it, they are testing it in, and then working back from testing to design and building a dedicated piece. If we want to go with our own dedicated designs, then we can do it the same way. But I think you are going to find the products the automobile manufacturers are using are dedicated to their application.

I want to comment on one other thing. This is primarily a workshop of users and I represent the R&D environment. I do not know if the Air Force is unusual in that respect; I understand from talking that it is. We are very well partitioned in a distributed structure in the Air Force. Research is in one area, R&D in another, engineering development in another and, finally, systems production in still another house. As in any distributed system, it is only as good as the interfacing and communications that you can establish.

Is anyone familiar with the book by Thomas Martin called, "Malice in Thunderland," about bureaucracy and bureaucratic management? Sir Thomas said that the government organizations tend to be somewhat paranoid; they tend to be a cooperative force bound together by internal hostility. Paul said this morning that the spirit of cooperation is rampant among us and that goes back again to the dollar control. There is nothing like shrinking dollars and competition to force cooperation in doing something. The Air Force LSI strategy use is undergoing a dramatic overhaul at the present time. We recognize we are no longer in the driver's seat with the mainstream semiconductor industry, at least from an R&D sense. I guess the fundamental aspect of this strategy is that we are going to use dollars intelligently. By that I mean we are going to cooperate among organization, reestablish a smooth interface and do what I call total program planning. What does that mean? Well, closing interfaces between the basic development, manufacturing qualification assurance and systems requirement people, if you will. I might mention that in partitioning organizations some things always fall through the cracks. Somehow the responsibility for them tends to get left out. Qualification seems to be one of the areas, at least in the Air Force, that has been left out. Assurance, basic fundamental development, and manufacturing have an organizational responsibility; the qualification has dropped out of the middle and I guess that is what we are all here for. David Haratz defines standardization as volume utilization. That is the way of the semiconductor business, and if it is in volume use, it is a de facto standard.

The last thing I want to talk about is management discipline. Standardization is a way of doing business. And the way of doing business is strictly management. I advocate that approach.

The Air Force LSI strategy is selective investment, selecting very carefully where we have to depart from the semiconductor industry. With users, if you want to use mainstream semiconductor pieces, we are all for it. It is not my responsibility and I suggest you establish management discipline if you are going to try to use them. If you want some special pieces, we are going to try to do that and we are going to use the dollar control variable; we are going to just pay for the whole process. That is the only one we have found in which we can get any attention.

As in R&D, this is the only way in which I have been able to get a reasonably good definition of system requirements (Table 7-1). I call it the "ilities" list. It is, of course, all the motherhood terms, but frankly, it is the only one that, in a time sense, seems to us to be reasonable. You as users want infinity in the first item, zero in the second, zero in the third, infinity in the fourth, zero in the fifth. I am not sure what you want in the sixth one - modifiability, programability, adaptability, affordability.

Question: You interface at the system environment level; is that a correct statement?

In purchase requests, I define generic requirements usually as functional requirements. I interface by paying suppliers to design and develop circuits. How do I relate to systems people in the Air Force? Basically by telling them I have a feeling for their selfish interest and how much I am going to deliver to them. But the point I wanted to make is that you people as users have a complex range of platforms and missions. And you really do have a legitimate purpose in trying to structure what your priorities are among those various "ilities." And I call it a decision surface if you want a decision envelope. But they do differ by what platform class you are trying to address. And LSI has the advantage in that it tends to move in the right direction for a large number of those things.

Table 7-1. System Requirements

Parameter	Platforms						
	Ground		Aircraft		Missiles		Satellites
	Fixed	Mobile	Tactical	Strategic	Tactical	Strategic	
1. Performance capability							
2. Powerability							
3. Physical suitability							
Configuration							
Size/weight							
4. Reliability							
Availability							
5. Maintainability							
Survivability							
6. Modifiability							
Programmability							
Adaptability							

Table 7-1. System Requirements
(Continuation 1)

Parameter		Platforms						
		Ground		Aircraft		Missiles		Satellites
		Fixed	Mobile	Tactical	Strategic	Tactical	Strategic	
7. Affordability								
Producibility	Development							
	Production							
	Support							
8. Deliverability (schedule)								
9. Credibility								
Qualifiability	Technical risk							
Assurability	Cost risk							
Sourceability	Schedule risk							
Testability								
10. Conformability (acceptability)								
Visibility								
Saleability								
Politicability								

There were some words said this morning that I want to comment on. We heartily endorse what Carver Mead was saying. We have come to the fundamental realization that if we are going to move ahead in LSI, we have to overhaul the design methodology of integrated circuits. We are moving to do that characterization of testability through regularized structure design. We will touch some more on that later.

I want to make a comment on availability, survivability, reliability maintainability. Someone mentioned fault tolerance on the chip. We have been addressing that. You are led quite naturally to that kind of a consideration.

In the programmability area I said that there is nothing like fundamental decision. If you can decide on what your design is and you want minimum life cycle costs and minimum production costs, we can pick it up by trading a little bit of development costs for production and life cycle costs. We can do custom LSI and we can do it well. But you have to decide, and you cannot change your design rules at the last minute. So I say, programmability is a form of decision avoidance as well as flexibility in growth. I throw that challenge out to you as users. In microprocessors, etc. (Figure 7-1), software is the lower 9/10 of the iceberg; from my point of view that seems to be a real problem.

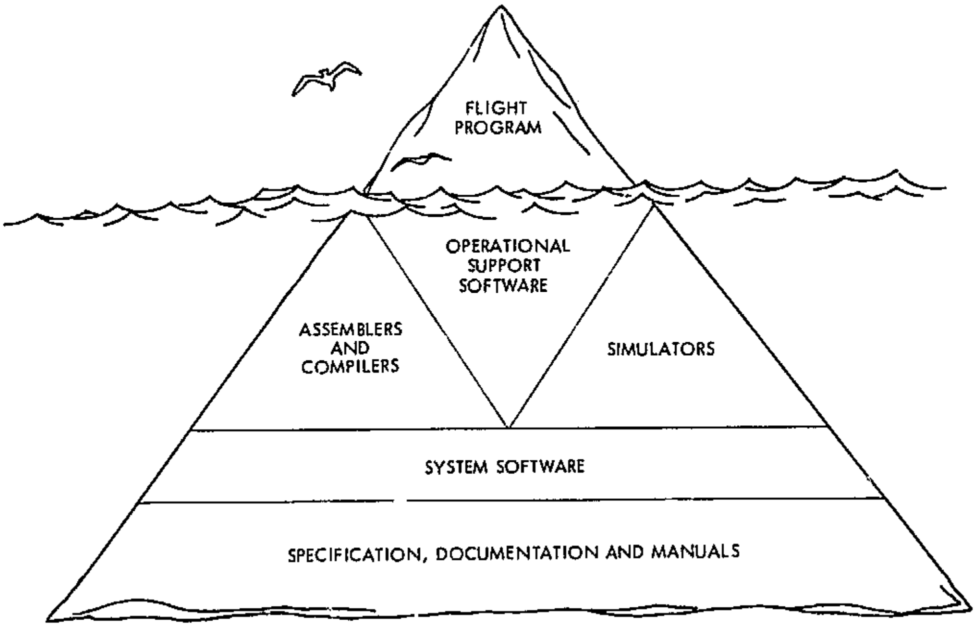

FLIGHT PROGRAM
OPERATIONAL SUPPORT SOFTWARE
ASSEMBLERS AND COMPILERS
SIMULATORS
SYSTEM SOFTWARE
SPECIFICATION, DOCUMENTATION AND MANUALS

Figure 7-1. The Software Iceberg

In the curve shown in Figure 7-2, efficiency of hardware use is plotted versus costs. If you look at hardware and you want to use it very inefficiently, it costs you a lot, both in size and weight, and it tends to be rather a universal abstract term. So is software since you try to drive it to use the hardware to its maximum extent. You have heard the comment: the last 10% of performance is 90% of the costs (Figure 7-3). Costs rise very rapidly in that area.

What are trends that go with LSI? I have not seen much of the structured approach or much progress being made in the software area. There is an old line that designers design for the applause of designers and programmers program for the applause of programmers; that is pretty inherent. I think the hardware people are getting ahead of the game.

Question: Will you be getting ahead of the game if you are not taking software into account?

Amen! But I am a hardware person. About five years ago I was in charge of a memory technology group and I was also in charge of a logic technology group. Now I am in charge of a processor technology group. That reflects reorganization within the Air Force Technology Development and reflects the fact that LSI is here; systems on a chip are here; logic and memory operating together are fundamental to things. It is easier for systems people to understand integrated circuit technology than for integrated circuit technologists to understand systems and we have operated from that latter approach. So we are trying to learn systems language and also some knowledge of systems things.

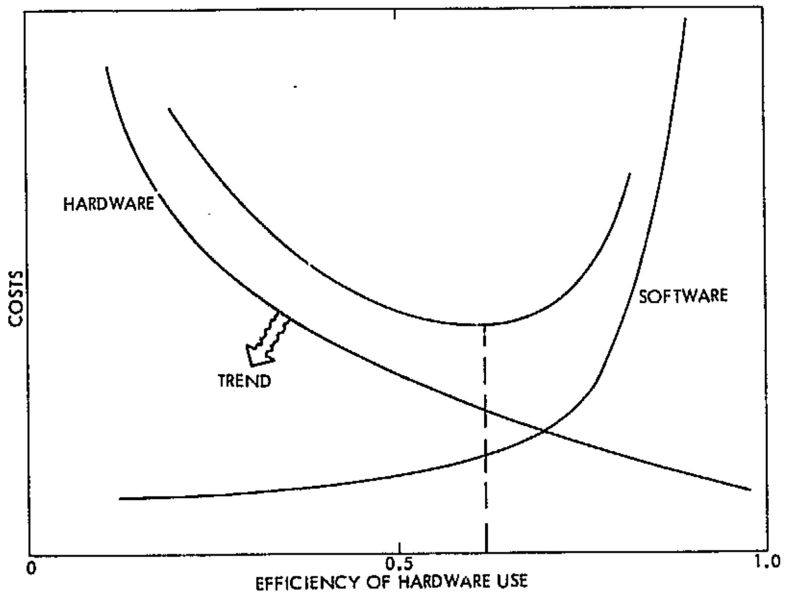


Figure 7-2. Hardware Efficiency vs Costs: Programmability

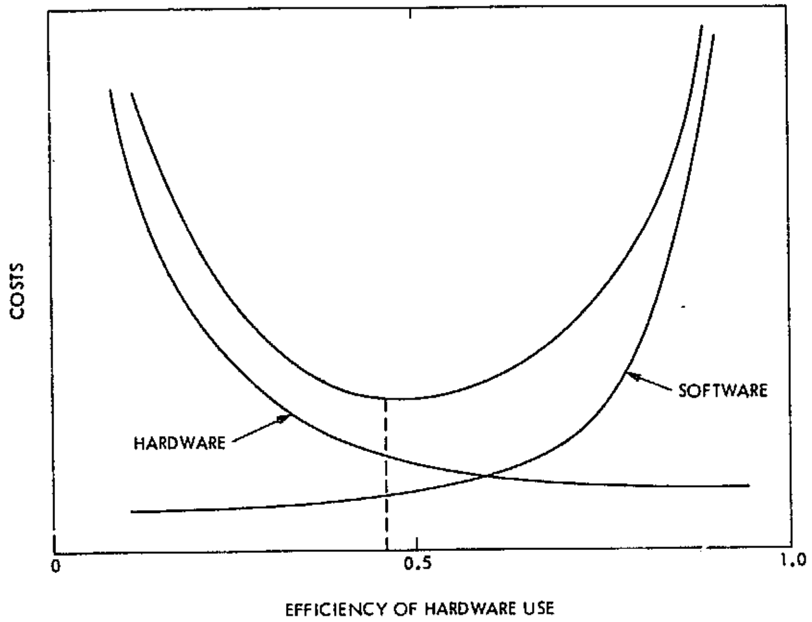


Figure 7-3. Hardware Efficiency vs Costs: Software/Hardware Interaction

The computer regime is the one I would say is the dollar controller. I do not speak for the computer side of the Air Force, but they are concentrating their approach at the moment into bit slice regime, trying to use bit slice for defining a standard avionics instruction set (Figure 7-4). They are trying to define a standard high-order language, and there are major activities going on.

In the 8-bit areas or the fixed instruction monolithic approach, frankly, from a technology point of view, we watch industry. Industry has a way of letting competition establish what are standards, and if you users are clever you will make a management decision as several companies have done. Thou shalt use this one unless you can show why it does not satisfy your requirements and then you can use that one. Several companies have established use of one or two or three and established a management control point that dictates the use of them and that makes some sense. It is not a legislative standard but it is an adversary approach where you are forced to justify the use of something different from the recommended one. And I think that is a good approach in that area.

There is one aspect of microprocessor development where we are down in that regime and that is in the radiation-hard area. There is no real commercial driving interest, but in our programs there is.

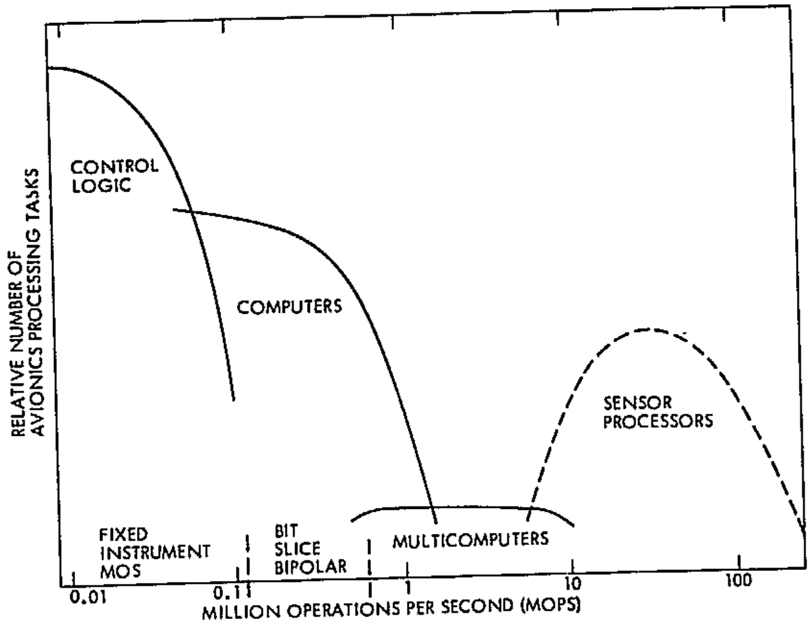


Figure 7-4. Microprocessors: Avionics Applications

As LSI gets bigger, about 20,000 gates or so (maybe that is too low a level, it should be up higher as Carver says), pretty soon you are going to be at a point where even human designers cannot design devices (Figure 7-5). So you are going to have basically a standard cell approach or structured approach to it. Does anyone want to make any comments on that?

Question: Are you talking about efficiency as the important parameter? Perhaps you should take the total system cost. I would think efficiency would be one of the least important factors and testability one of the most important.

Okay. The point I want to make is you can go to a TI or a semiconductor vendor and he will give you the argument that the reason he cannot produce those chips down there or does not want to, is because they are inefficient with regard to silicon real estate. Which means the size of the chip for a function is up, which means his yield is down, which means his costs are up and it is more costly and he cannot turn the high profit. Of course we in the military, at least as I view you users, see somebody buying a commercial chip for $3 and you want it for $3. If you were willing to pay $300 for the kind of volume you have, you could get it fairly easily, but nobody wants to do that.

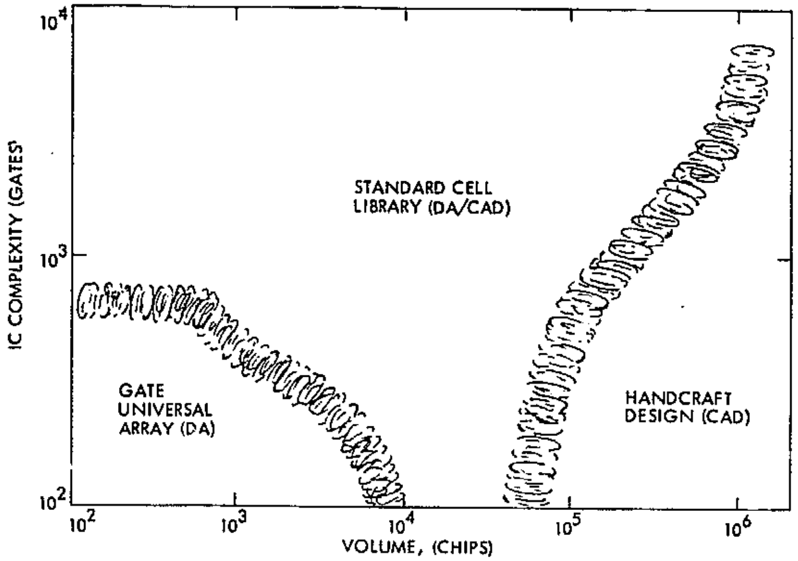


Figure 7-5. LSI Approaches

You made a clever point this morning, which was that you need to choose chips which are qualifiable. I guess from my point of view in interfacing with Air Force designers, they choose only chips which are qualified, not qualifiable. I think there is a real-time constant difference therein in where systems go. By the time we go through RDT&E, there is about an eight-year lag. Add two years for development and the integrated circuits you have in a system just going into production are almost 8 to 10 years old. So that is the lag time I have to work with.

<u>Question</u>: Do hi rel manufacturers do large volume?

No, they do not need a large volume. That is the point!

<u>Question</u>: How do they make dollars?

They need dollars! That is the point!

<u>Question</u>: A complex circuit design will have nonrecurring charges.

That is right. You have to absorb the nonrecurring charges. That is the point I made. That is the decision on your part. You have to trade somewhere. LSI trades development costs for production costs. That is precisely what it does. You have to be willing to pay somewhere. This is policy. I am telling you we are following an LSI development, and frankly the policy I mentioned that we are doing, we are going to pay for. We are going to pay for the design, we are going to pay for the production, the second sourcing, and the characterization. We are going to sink the development and hopefully be very selective as to where we have to depart from the mainstream industry for our parts.

I guess my observation is that in the military system area you are faced with small volumes for a large number of your LSI chips. How many satellites are you going to build? You are faced with two choices. You either use something that is designed especially for your use, that you paid to have designed, or you suffer what is required using the industry chips. You can test what you want and be very selective. You seem to be very dissatisfied with that approach. I would not argue that there is a fundamental development, a science to be done there, but you are finding, or at least I keep hearing, that they cannot produce them for you. We are trying to evolve the most cost-effective way we can think of to come up with those kinds of chips for you.

The Air Force LSI policy is to use commercial items wherever possible, wherever it makes sense, and we will try to drive you to do that. We are trying to get very smart, and we are working with the systems program offices in an advisory capacity to tell them you do not need to build this special thing, you can use that one. But there are always those cases where you need special applications, and that is what I am talking about. Standardization is something we heartily advocate; as a matter of fact, I think in a standard module sense LSI is a very good way to establish standardization. That is the whole basis of the industry.

For example, take the I/O area. If you find that, of the commercially available chips that you want to use, it takes 20 chips to do your I/O and that is too much real estate for you to pay in your system, and if you want a custom chip design and you are only going to build three satellites that use a thousand chips, then you had better look at that approach to doing your circuit.

Question: Looking at the systems viewpoint, maybe we are only going to launch three satellites, but if we can segment the satellite into such groupings that we use the same thing in 15 parts of that satellite doing different jobs, then we have one and one half orders of magnitude less parts that we have to qualify.

That is true. I will make an observation on that. In the Air Force there is a whole discipline called communication sensors. There is another whole discipline called navigation sensors, there is another one called radar sensors, and another one called electronic warfare sensors. Frankly, they are clubs. They have grown up with 20 years of history and designing very legitimately. LSI can cross all of those lines. The jargon is different; the functions, the algorithms and the basic functions are all the same, but that is a management problem. That is the whole business of standardization - management discipline - a way of doing business. There is no substitute for that, and I do not know how you address that. Yes, you can drive design to maximize commonality; we strongly advocate that. But there are cases in which you cannot do that.

Question: Do you feel that the Air Force is bound by the use of MIL 38510. You seem to advocate the opposite?

No. We would tend to pay for the MIL 38510 qualification, but that is not my ball game. I think the qualification techniques could be overhauled somewhat. I think they tend to cost us more than they should. But that is a personal opinion. That is an area where perhaps we have standardization as discipline, if you want to call it that, and discipline can be overvigorous at times. I look at Qualification Procedures 38510 and I do not know whether to kiss you people for being very disciplined in your approach to doing things or to kick you for not being innovative enough. It is a very fine line. I think there is room for some improvement in the qualification we pay for.

This is just a general program plan which shows there are two aspects to LSI if you want to call it that. One is design development and the other is manufacturing, and the Air Force has responsibility for development at the Avionics Laboratory and manufacturing at the Air Force Materials Laboratory, and since I wear both hats, the Air Force has found it very convenient to establish close liaison between those groups, I might say that we are trying to close the loop with quality assurance, which is at Rome Air Development Center. From the Air Force Materials Laboratory we have a memorandum of agreement with the Rome Air Development Center to work together and cooperate on getting our chips characterized and qualified as we manage to pay for the pilot production quantities.

An approach we have outlined for a radiation-hard microprocessor is shown in Figure 7-6. That is one area in the industry performance compatible regime that we have decided to do something to support since we cannot find anyone commercially doing radiation hardening. So we have laid out a development approach. It is a cooperative between the Avionics Laboratory and the Air Force Materials Laboratory with regard to specific sponsorship, and Rome Air Development Center is also involved. Radiation-hardened CMOS-SOS is an Air Force Avionics Laboratory program. It is based around a gate universal array, and it is investigating basic radiation-hardened oxide that goes by the name of dry, wet, and hybrid. If anyone wants details, we can take that up separately.

MMT is Manufacturing Methods Technology. You see a CMOS-SOS and it is not radiation hard but you see three circuit types: (1) general processing unit, which is an 8-bit slice unit, (2) read only memory, and (3) gate universal arıay, which is slightly different from the one in the top, but the same generic type, but an improved design. That is in CMOS-SOS circuit and establishes the manufacturing pilot production of generalized CMOS-SOS. You see the design rule ›xchanged here. The manufacturing methods designs have been made desig.ı rules compatible with the radiation hardened process.

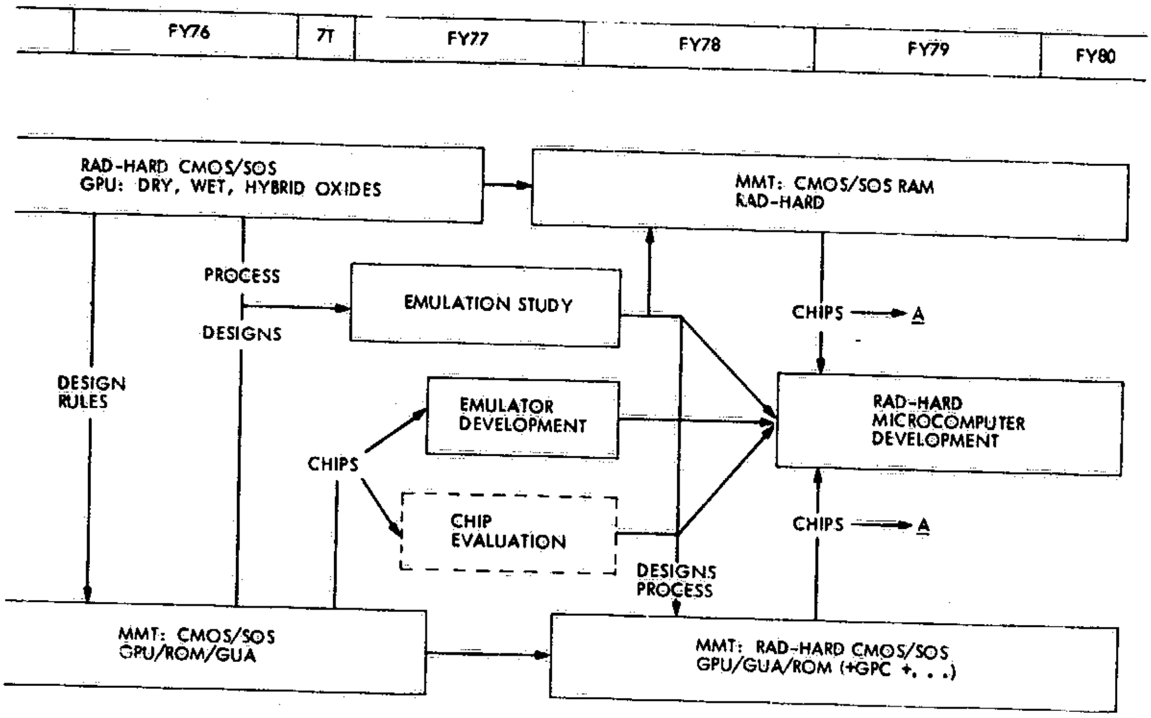


Figure 7-6. Air Force Radiation-Hardened Microcomputer Development

Here we have another unique feature. We own the designs and the mask sets for these chips. We define, we can get both process and the chip designs. As I said, the GP uses a slice approach. We are doing an emulation study and we still have a target vehicle to pick. We are assessing that now, to examine the chip designs and how they would operate in an emulation mode. We are looking at taking the output of the chips and establishing an emulator development to really characterize how the chips functionally perform. Hopefully, RCA is the manufacturer of these things, and I think if anyone wants to characterize we can act as an intermediary for getting your chips to characterize, at least to a limited degree. The CPU is an 8-bit slice; that is, the fundamental CPU element has 16 GP registers 8 bits wide and 300 nanoseconds add time. It is a fairly high performance thing. I think we estimated that if we tried to emulate the 8080 we would probably end up on the order of 5 times faster for the same kind of function. The "gate universal array" is what we intend to use for all the various interfacing chip designs. Out of the emulation study and the emulator development we would hope to get the basic interfacing designs, the controller designs. You see two programs starting; one kind of an independent mode manufacturing methods technology (MMT) for radiation hardening. RAM is the remainder of the chip complement needed and that will be done in the radiation-hardening process that will be starting up late this fiscal year.

SECTION VIII

APPLICATION OF DIAGNOSTICS AND TEST CHIPS FOR PREDICTING RELIABILITY OF LSI*

J. Maserjian
Jet Propulsion Laboratory

A. INTRODUCTION

Conventional methods of qualifying and screening IC parts are extremely difficult to apply to LSI. There is mounting pressure to incorporate LSI into new system designs, but there currently exists no high-rel qualification procedure. Even if we were able to establish qualified lines in the near future, there remains the serious problem of how to adequately test complex LSI parts for reliability assurance.

An approach being attempted under a NASA program involves extensive use of diagnostic measurements, both chemical and electrical, applied to test wafers or chips off selected wafer-processing lines. The central idea is to obtain sufficient diagnostic data for reliability prediction without placing heavy controls and qualification procedures on industry. At the same time, it is intended that the information assist the manufacturer in upgrading his process and improving his yields. In this manner, it is possible to deal with established commercial LSI lines which may otherwise be inaccessible to a high-rel government market. This year it is planned to start applying these techniques on lines of industrial participants in a joint USAF-NASA radiation-hardening program. The following sections outline the approach and give some examples of our previous results.

B. APPROACH

Figure 8-1 illustrates the general features of the approach. Chemical diagnostic methods are applied to special test chips removed at selected points of the process. Electrical diagnostics are implemented on test chips from a single wafer diverted from a standard wafer lot through a special set of test-wafer masks. These masks are independently designed to incorporate special test structures that can be periodically updated. A natural extension of this procedure would incorporate the most useful test structures on special test chips to be included on the LSI masks, allowing data to be obtained from the same wafer as the LSI chips. Implementation of this latter procedure would be best carried out after optimizing the method through the test-wafer results. Clearly, this step would be greatly facilitated if the manufacturer also derives useful information from the data.

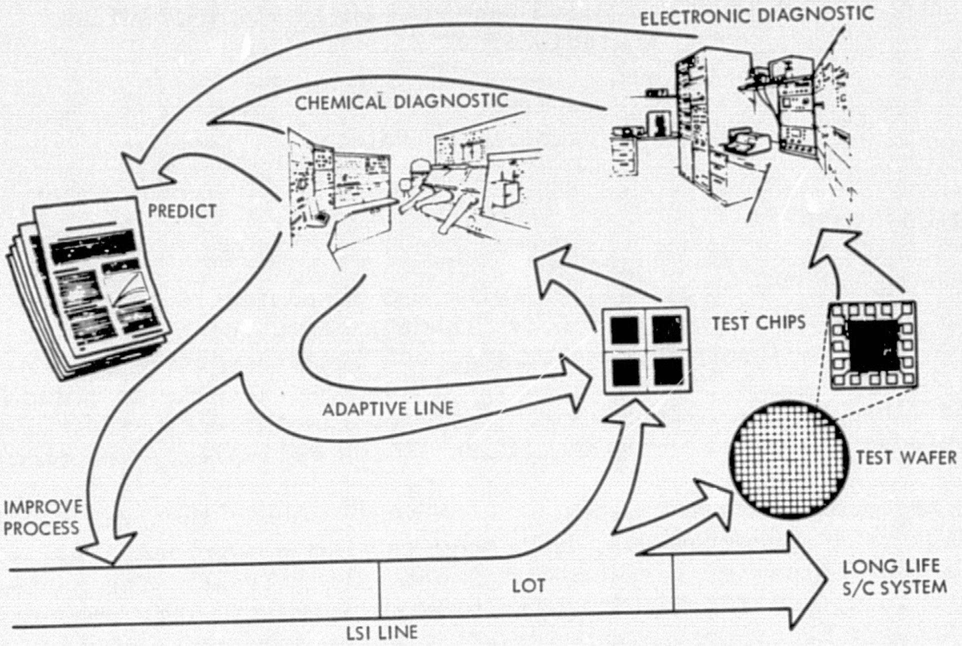


Figure 8-1. Relationship of Diagnostics to Reliability Prediction

The diagnostics provide a comprehensive set of electrical and chemical information that can be interpreted in terms of reliability. That is, with sufficient information and cross-correlations of different kinds of data, we greatly enhance our opportunity to relate correctly these observations to basic life-limiting mechanisms. Even though the understanding of failure mechanisms, such as charge buildup during irradiation, has steadily improved in recent years, only rarely have there been reported instances in which a sufficient breadth of data has been used to adequately characterize a process in these terms. A successful start in this direction was described last year by Sandia (Reference 1).

An adaptive line at JPL allows in-depth investigations of certain processing effects or anomalies that would be difficult to carry out through an industrial line. Although a relatively modest facility (see Figure 8-2), it is equipped to handle complete CMOS wafer processing under fairly rigorous control. Most of the previous investigations at JPL on basic mechanisms have been carried out through this line.

C. ELECTRICAL DIAGNOSTICS

Electrical diagnostics are carried out in an automated test facility to permit rapid turnaround of reduced data. After initiating the specific measurement, a minicomputer automatically controls the test sequence and reduces the data. The computer is interfaced to command digital power supplies and relays (for switching between test samples

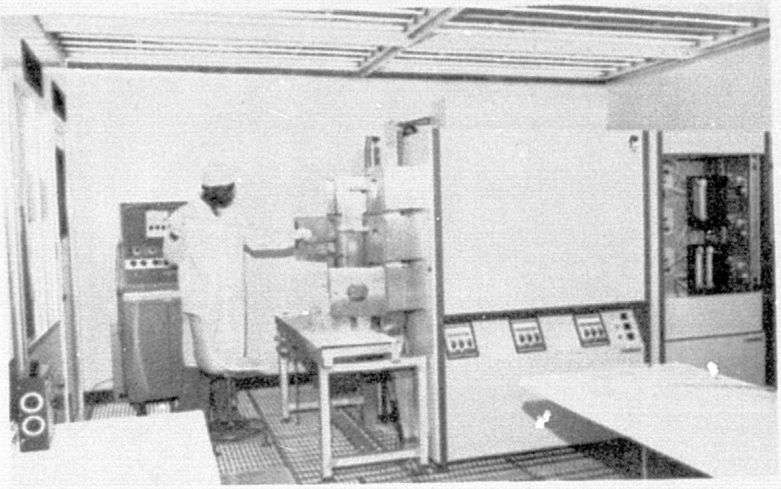


Figure 8-2. Furnace Area of JPL Wafer-Processing Facility

and different test operations), and to read different instruments. Measurements of voltage, capacitance, charge, and current are read from integrating digital voltmeters for greater precision. Two wafer probe stations include both hot (300°C) and cold (LN_2) temperature stages. A station for packaged test chips is being added.

Electrical measurements include both physical and device parameters. Physical parameters include surface-state density, fixed charge, surface doping, carrier lifetime, mobile ion concentrations, breakdown defect density, oxide trap properties, sheet resistance, contact resistance, etc. Device parameters include basic transistor and simple cell parameters. These measurements will also include appropriate stress such as bias, temperature, current, radiation, and changes in ambient gas in order to accelerate instabilities or known failure mechanisms. Measurements have been performed in the past on experimental test wafers such as illustrated in Figure 8-3. Test wafers with step-and-repeat test chips will be used in future designs.

The application of electrical diagnostics to reliability prediction can best be illustrated by two examples. In the case of oxide breakdown in MOS structures, we have established new test procedures to properly take into account the time-dependence associated with breakdown (Reference 2). The phenomenon responsible for this time-dependent breakdown has been shown (Reference 3) to be the gradual emission of ions such as sodium, normally trapped at the metal/oxide interface, and transported to the silicon interface where they form clusters (Reference 4). Figure 8-4 illustrates this time-dependence of oxide breakdown probability for a wafer that was deliberately contaminated to enhance the effect. The same form of time dependence was observed with low contamination but with correspondingly lower breakdown probabilities. The important point to be noted is that the time delay associated with this mechanism would go undetected with the usual voltage-ramp procedure of measuring

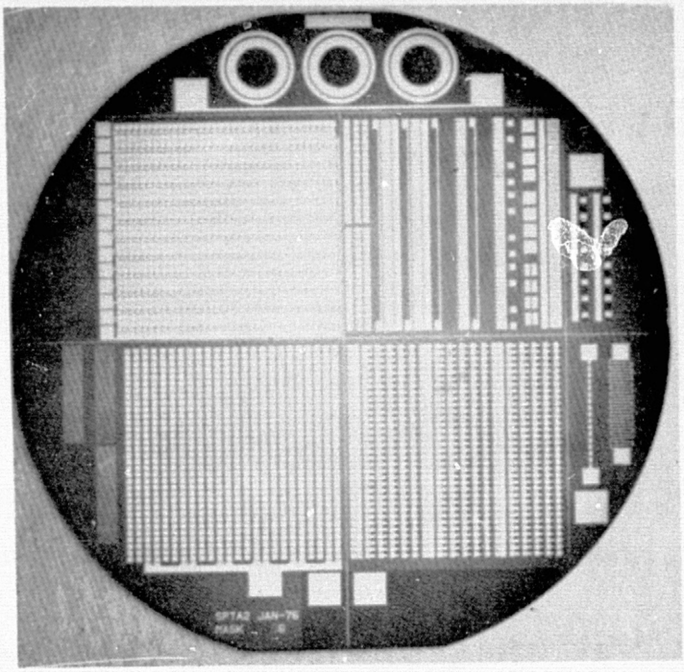


Figure 8-3. Experimental CMOS Test Wafer

distributions of breakdown voltages. On the basis of these findings, we have introduced a procedure of simultaneously testing an array of MOS capacitors with a sufficient field-temperature-time stress to detect the density of defects contributing to time-dependent breakdown (Reference 2). For example, results obtained over a range of oxide thicknesses show a surprising exponential decrease in this defect density with decreasing oxide thickness.

The radiation hardness of LSI is another major reliability concern. It has become clear in recent years that hole trapping in SiO_2 is the dominant radiation effect in MOS devices. To obtain more direct information on this mechanism, we have recently developed means of directly measuring the hole-trapping parameters by avalanche injection of holes into the oxide from the silicon interface (Reference 5). Figure 8-5 shows an example of the measured response of voltage shift and its derivative with respect to the number of injected holes per unit area. The solid line is the theoretical fit to the derivative which gives the desired parameters--the trap density N_T and the trap cross-section σ. From these parameters we can predict the radiation effect due to hole trapping (Reference 6). Figures 8-6 and 8-7 show the agreement of actual radiation-induced shifts with predicted values using the independently measured hole-trapping parameters. The scatter in Figure 6 is due to wafer-to-wafer variations we have consistently seen in N_T. Such variations emphasize the desirability of using test chips from the same LSI wafer.

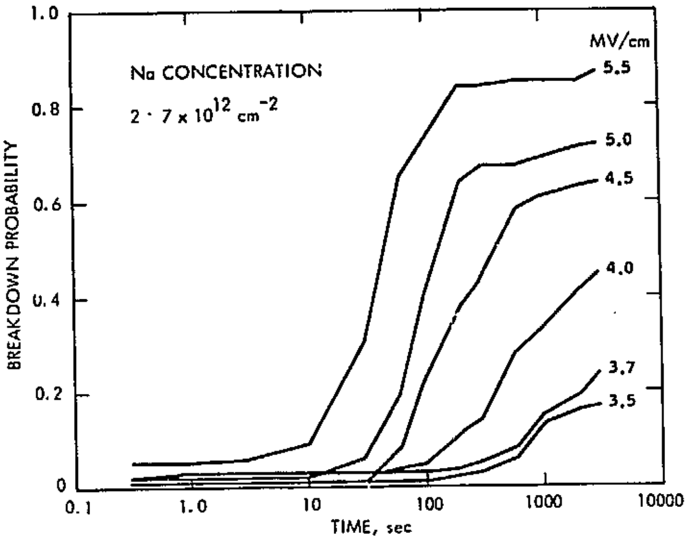


Figure 8-4. Time-Dependent Oxide Breakdown. Probability Represents Fraction of 200 Test Capacitors Undergoing Breakdown. Applied Oxide Field is Shown

D. CHEMICAL DIAGNOSTICS

The chemical diagnostics facility (see Figure 8-8) is based on a number of interrelated surface analytical techniques. These methods have been implemented in a system which permits rigorous control of sample preparation. The facility is controlled by a minicomputer which enables digital operation, automated sample analysis, and sophisticated real-time data reduction. The key method emphasized in this facility is high-resolution X-ray photoelectron spectroscopy (ESCA, or XPS) since it gives chemical state information with minimum sample perturbation. Supplementary analytical information is given by secondary ion mass spectroscopy (SIMS), which provides depth-profile information and extreme sensitivity for compositional analysis. Pictorial capability and high spatial resolution are provided by a scanning Auger microprobe.

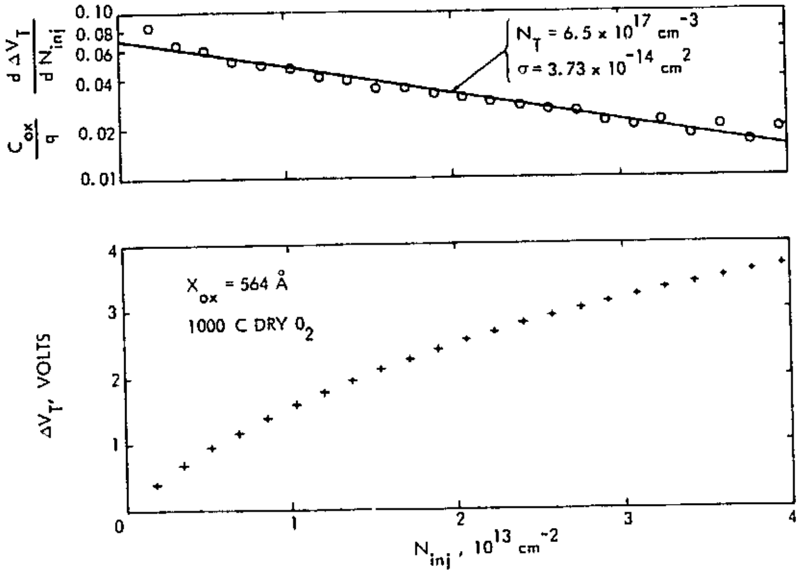


Figure 8-5. Avalanche Injection of Holes into SiO_2. Determination of Hole-Trapping Parameters, N_T, σ

The application of chemical diagnostics to LSI processing can be summarized as analytic or experimental. Analytic tasks emphasize the determination of elemental compositions and distributions, while experimental tasks are concerned with the chemical state of particular elements which determine the characteristics of the device. This latter topic is concerned with understanding the cause-and-effect relationship between the process and the resulting device.

Figure 8-9 shows an MGRS failure. This failure mode was observed in hermetically sealed parts which had small amounts of water vapor within the package. Failure occurred after extended operation (parts passed screening tests) and was accelerated at low temperatures and retarded at high temperatures. Dendrite-like filaments of gold grew between adjacent metallization strips despite the presence of a passivating glass layer. XPS analysis of the glass surface showed the presence of extremely low levels of iodine present as an etchant residue. It could not be detected by electron (Auger electron spectroscopy) or ion (SIMS) methods because of electron-stimulated desorption in the vacuum environment of the required spectrometers. This detected iodide was a critical component of the failure system because it enabled gold transport, which ordinarily could not have occurred in a water overlayer (Reference 7).

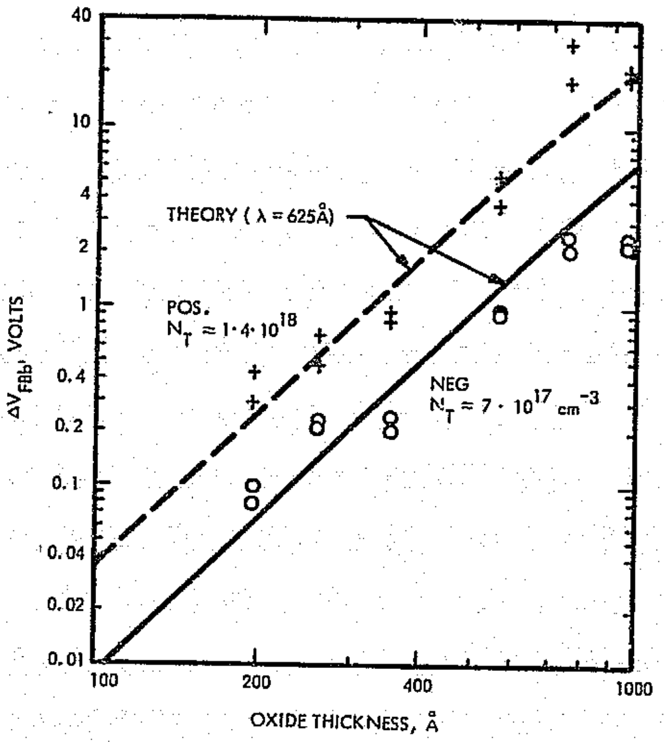


Figure 8-6. Flatband Shift ΔV_{FBb} Corrected for Surface States After 600 Krad of Co^{60}. Theoretical Curves Use Parameters From Avalanche Measurements

Figure 8-10 illustrates experiments which studied the chemical environment of the mobile ion Na^+ in silicon oxide (Reference 8). It was found that the sodium in SiO_2/Si could be detected by XPS when present at a level of 10^{11} atoms/cm^2. However, several unique chemical states of sodium were observed (at least five) which appeared to be a function of the physical location of the sodium. By using an "in situ" bias-temperature stress, two different ionized states could be located at each interface (vacuum/SiO_2 and SiO_2/Si) and one corresponding to a neutral immobile species. The different states at the Si interface may be related to the clustering effect involved in oxide breakdown.

Another illustration of the desirability of determining elemental compositions is given in Figure 8-11. Chemical contamination was causing severe problems in the quality of junctions. This set of scans gives an elemental analysis of a sample on which a boron-rich oxide had been grown during diffusion. The lower scans (center and lower) show the presence of iron in the boron-rich oxide, but virtually no copper. The upper scan gives the spectrum of the substrate after the oxide has been stripped away. Now, copper is the major component and iron is negligible. These experiments demonstrate segregation of iron into the boron oxide, while copper remains an interface contaminant.

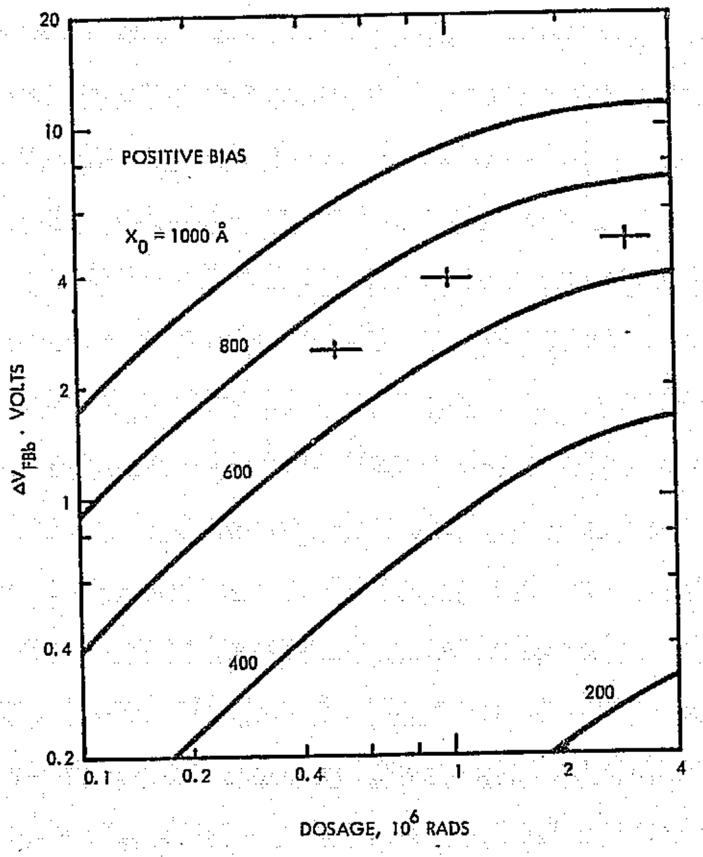


Figure 8-7. Flatband Shift ΔV_{FBb} vs Radiation Dose. Solid Curves are Theoretical Using Parameters From Avalanche Measurements. Crosses are Data From 700 A Oxide MOS Capacitors

Knowledge of elemental composition is not adequate for successful chemical diagnosis, as is illustrated in Figure 8-12. Here, two silicon samples have been etched for similar times (oxide removal) with the only difference being that the sample giving rise to the upper trace was etched in an HF/H_2O mixture while the lower trace involved an $NH_4F/HF/H_2O$ mixture. The copper 2p XPS spectra diagrammed in the figure clearly show the different states of copper induced in the silicon substrate. Subsequent work has distinguished chemical states of copper which are electrically active, as well as those that are not.

Chemical diagnostics provide unusual information when great care is taken to minimize the destruction or modification of the sample. One such case is the effect of organic contamination. Standard cleaning procedures are found to modify only the organic layers normally present, and carbon remains at the silicon interface in bound chemical states

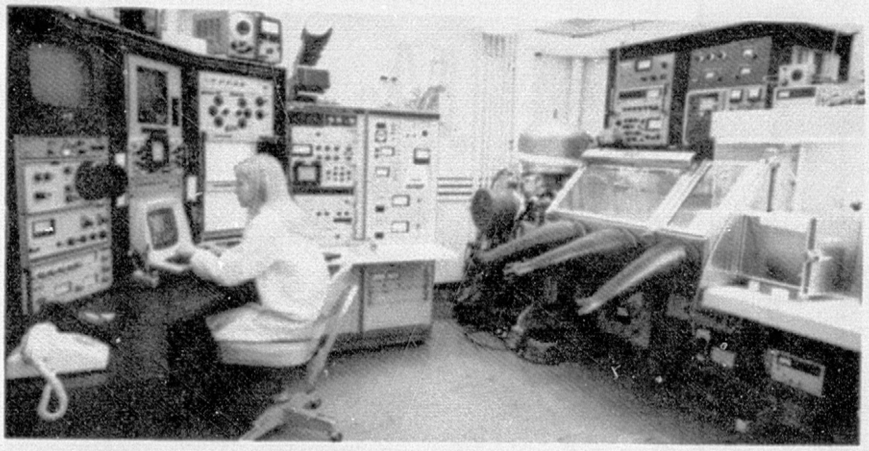


Figure 8-8. Chemical Diagnostics Facility

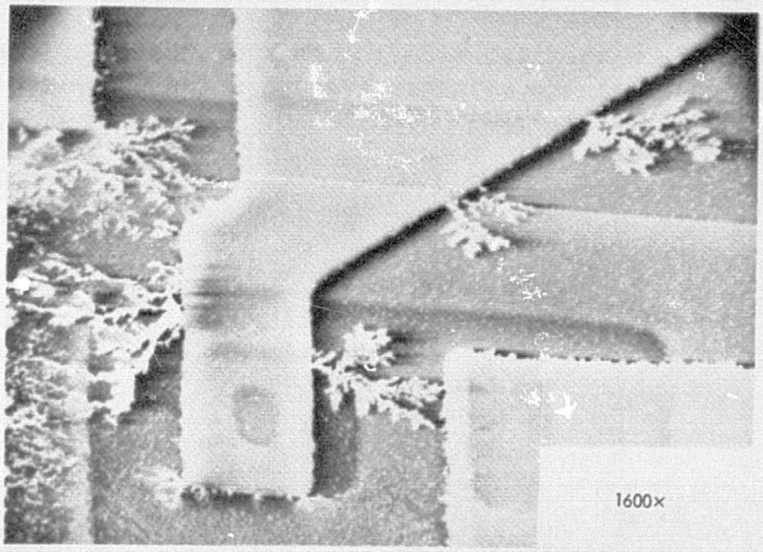


Figure 8-9. SEM Photo of Migrating Gold-Resistive Shorts (MGRS)

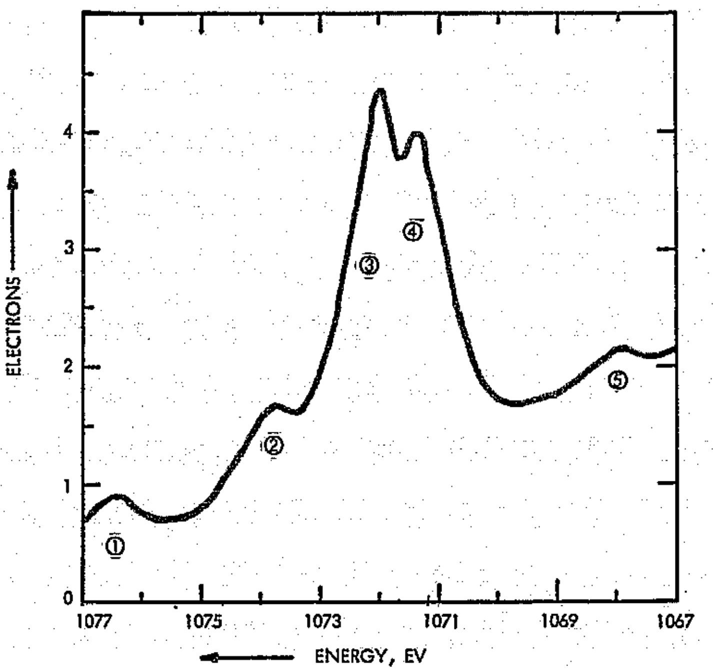


Figure 8-10. Composite XPS Spectrum of Various Sodium Species Observed in Thin Oxides Grown on Silicon, Sodium 1s Region

after oxidation. Such chemical modifications of the interface due to organic as well as other contaminants are under intensive study as part of the diagnostics program.

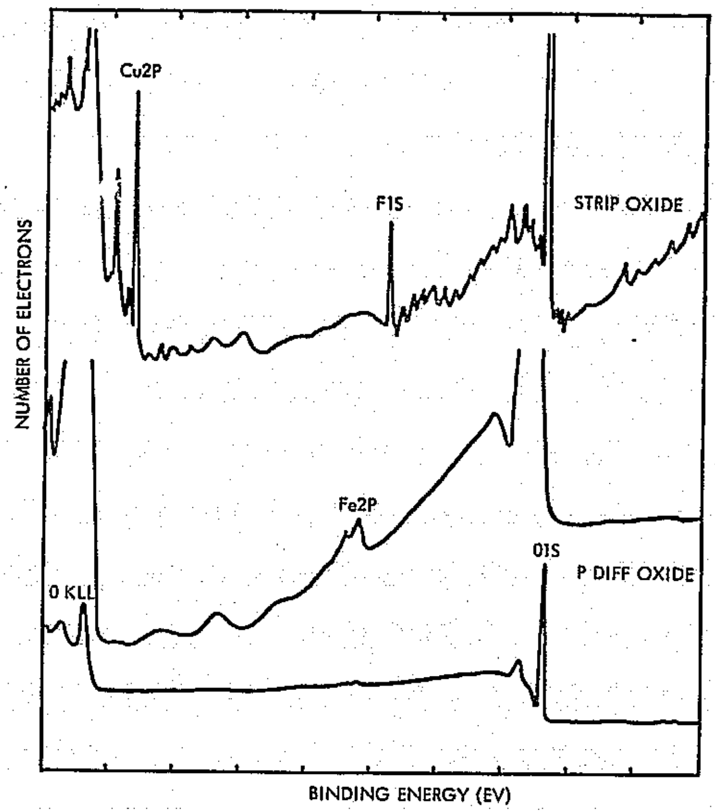


Figure 8-11. Wide XPS Scan After Boron Diffusion. Lower Two Traces of Oxide Surface, With Center Trace Plotted on Expanded Vertical Scale. Upper Trace of Silicon Interface After Stripping Oxide With HF

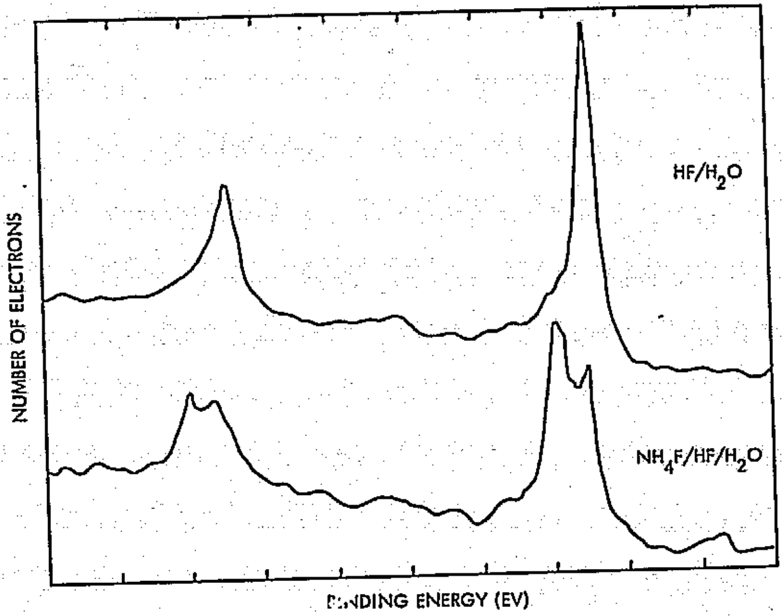


Figure 8-12. Copper 2p XPS Spectra of Silicon Wafers Permitted to Equilibrate With the Etchant Solutions Noted

REFERENCES

1. Derbenwick, G. F., and Gregory, B. L., IEEE Trans. Nucl. Sci. NS-22, 251 (1975).

2. Li, S. P., and Maserjian, J., IEEE Trans. Elect. Dev. ED-23, 525 (1976).

3. Li, S. P., Bates, E. T., and Maserjian, J., Solid-State Elect. 19, 235 (1976).

4. DiStefano, T. H., J. Appl. Phys. 44, 527 (1973).

5. Maserjian, J., and Bakowski, M., in preparation.

6. Viswanathan, C. R., and Maserjian, J., "Model for Thickness Dependence of Radiation Effect in MOS Structures," 1976 IEEE Conf. Nucl. and Space Radiation Effects, San Diego, CA.

7. Grunthaner, F. J., Griswold, T. W., and Clendening, P. J., Proc. 13th IEEE Rel. Phys. Symposium, pp. 99-106 (1975).

8. Grunthaner, F. J., and Maserjian, J., Proc. 13th Annual IEEE Reliability Physics Symposium, pp. 15-25 (1975).

SECTION IX

NON-USER PERSPECTIVE

Sam Davis
Electronic Engineering Times

A. SUMMARY

Mr. Sam Davis of the Electronic Engineering Times addressed the workshop general session and presented the manufacturers' views relative to procuring LSI to the MIL-STD-38510 specification. A summary of his talk follows:

Mr. Davis had attended a conference in Cupertino, California, of semiconductor manufacturers the week prior to the users' conference at Caltech. He described some of the concerns and issues discussed by the manufacturers relative to producing parts to the MIL-STD-38510 specification, and he quoted the manufacturers as saying that they were having extreme difficulty in producing components to the 38510 specification. This difficulty was encountered with relatively simple SSI parts, and the manufacturers expect an even more difficult situation with LSI parts as complex as microprocessors. Some typical problem parts were given as examples.

Mr. Davis's conclusion was that the manufacturers recognized the necessity to be able to produce hi rel parts to 38510 but that a great deal of work needed to be done by the user community to develop 38510 to specify reasonably achievable requirements. He further felt that the hi rel LSI situation was not hopeless and that face-to-face communication between the users and manufacturers could resolve the differences.

B. PRESENTATION

One thing that I found interesting is that in the space of less than a week there were independently arranged users' conferences here and in Cupertino, up north, where the semiconductor manufacturers got together to discuss their problems in being able to build parts to 38510. They started out thinking linear only, but when they got into it they decided to attack the whole LSI spectrum pretty much the way you are doing it, but they are doing it from a manufacturers' viewpoint. I feel lucky, because I do not have a vested interest in either one and I can think very openly. I have nothing to gain or lose either way, but I think it might be wise to give you an idea of some of the things they were talking about in parts that are not even LSI that they cannot make. A typical example: Floyd Bonny who is the general manager of National Semiconductor was talking about a part called an LM111. It is a dual comparator that they have made for years, and he made a comment - I cannct quote him exactly, but he said we invented the part but we can't build a 38510 part. Jack Gifford, who is vice president for marketing and engineering at Intersil and is also trying to make the part, says we have a zero yield on it and cannot make it to the 38510 spec. Here are two of the major suppliers of a rather simple-minded part that apparently, and I have to put this in quotation marks, "is unrealistically specified in 38510." I hope you were not the ones who generated the specification.

As another typical example, the 741 op amp that has been out for five years, in order to meet the 38510 specs, has been redesigned three times. They have put thousands of dollars into doing this and they have come up with all sorts of gimmicks and test boxes to check it out and they find they are just barely meeting the spec. The manufacturers are looking at it from their own selfish viewpoint in that they are interested in one thing and that is profit. In order to maximize profits you get the highest yield you can and they want to do whatever they can to that spec to get a higher yield. You also have to recognize another fact; i.e., if they can get a higher yield, your cost is going to be less too. The question is, can you use the component? Now, one guy says the spec is unrealistic and the other guy says that's exactly what I want, and there may be a guy in the middle who says, gee, I don't need a one pico amp offset current.

These are some of the problems. You have to take what people say with a grain of salt. For a typical example, Gifford claims a 9% yield on a 741 op amp. It cost him $8.50 and a 9% yield to come up with a 741 op amp that you could buy a commercial equivalent of for less than $1, depending on where you buy it.

I can show you a chart where Gifford claims that it costs him about $8.50 when he gets through testing and everything else with a 9% yield. He also claims that if you can get the yield up to 27% (20-30%) the price goes down about $2. Their cost, not selling price. The other thing he said, and here again there are vested interests that have their own way of looking at things, was that they are going to have to invest something between $17-$30 million of capital investment to be able to handle some of these linear parts, and for the parts he was talking about he is projecting a total demand of something like 3 million parts for '77.

<u>Question</u>: Who was represented at the Cupertino conference?

The whole group within the industry. There were members there from Intel, Intersil, National Semiconductor, AMD, Siliconix, and others. Some of what the manufacturers are saying has to be true, some of what they are saying is not necessarily 100% true, but it is at least 1/2 true. Some of the specs were possibly not generated as well as they could be, depending on your viewpoint. I think you have to avoid the extremes under any circumstances. I personally think if you are not careful you will have a kind of a Ralph Nadarism or consumerism projecting that particular attitude into a generation of specs. Not that some of that should not be there. The problem that you always run into is how far do you go and where do you stop. You can specify the best part in the world, but if nobody can make it it isn't going to do you a damn bit of good.

Yields to manufacturers are dollars, and the thing that can happen in '77 is that those systems that were designed a couple of years ago that use these bread and butter, everyday parts might have difficulty getting parts ordered to MIL-STD-38510 if the manufacturers cannot get decent yields. First of all, you may not be able to get the parts. The LM111 apparently does not even exist any more in the 38510. There is probably some reason for this. I talked to the test manager at Intersil and he told me that that is a dual comparator if I remember correctly, and he said that the spec is written as if it were an op amp. I would have to say that if it is really done that way, it does not make any sense at all, although I have not looked into it specifically. I am assuming what he is telling me is true. But, taking that as another example, they have had to build literally dozens of test boxes just to try to qualify that part. And you have to think about it in another economic sense: you want a single part in the bucketful of parts produced in conformity with 38510. They have a certain number of production test systems to check out their parts as they come from the production line. Anytime someone comes by with something that does not command their profit interests, that slows down the rest of their lines that they are cranking out by the millions; that starts affecting them, and they could conceivably get to the point where they may not want that kind of business. The only one who may want to do it is the smaller shop that does not have a broad line. But even they probably do not want to get involved with 38510 because of the other problems involved in it.

I think the thing that Bonny and Gifford and the other people I talked to objected to is the fact that apparently (and I do not know this for a fact) no one had them review the specs when they were done to ask them if they could make the parts. Now, to me that sounds logical, to have manufacturers review the specs, but I guess there are personalities involved in any kind of an organization, bureaucracy, or whatever, and if I were writing a spec I would want to see if someone could make it. It would not really take a whole lot to ask a manufacturer if he could make the part.

Now if I could just project this from the op amp stage to LSI. You know, if in the op amps you are measuring voltages and currents and that sort of thing, and if they are analog measurements there are plenty of instruments around and there is some software for the test equipment. But now project that over to the LSI. We are talking large memories and you are talking microprocessors and now you are not building test boxes that are necessarily hardware test boxes; they are software test boxes, they are software retainers, and as I understand from talking to the people who make the software (the LSI testers), you almost have to know the application that that device is going to be used in to really test it to any confidence level. Now how are you going to specify that in a LSI 38510? Are you going to give the guy a list of 100 test programs and say, well, whatever is closest to that application test it that way, or do you want to wait the 10^{22} years while someone checks that thing exhaustively.

Question: Could you elaborate on the 38510 part as it related to the op amp?

There are some aspects of the tests that are either inconsistent or even contradictory. He said there was an inconsistency between whatever the slew rate of that particular op amp was and its capability to drive a certain capacity load. Mathematically it did not make sense and what they had to do to pass the spec was to go to the margin of each one and just barely get through. They could not get through the test spec.

Question: Did the manufacturers discuss providing different grades of parts to 38510?

Yes, that is another thing that was proposed: dash numbers on the slash sheets for different grades of parts. Why should a guy pay for a $15 part when for his application a $2 part might work?

Question: What solution would you recommend in working with the manufacturers to procure parts to 38510?

Well, I really think the people are basically reasonable and that all you have to do is to get them together. They all know each other's problems. It takes face-to-face confrontation or communication, and I do not think the problem is that hard to resolve.

Comment: Even that is no good if they do not know their product.

I cannot defend what they are doing because I do not know. I think this whole system ought to be a closed-loop system. If you want stability you must have some sort of a closed-loop system. Unfortunately, the users have gone off this way and RADC and DSC have gone off another and the manufacturer has gone his way. I really think you have to close the loop and get everybody together, and you may not get exactly what you want but you will get something that is as close as you can achieve, and that is important. The manufacturers realize it. They told me the way you are going to get something done is by the user. The manufacturers realize that they cannot force a change. It is only the user who can force a change either in the spec or the manufacture or something else.

SECTION X

WORKING GROUP A: COMMONALITY OF REQUIREMENTS AND POTENTIAL FOR STANDARDIZATION

Chairman: Richard Urban
Naval Electronics Laboratory Center

A. KEY ISSUES

Key issues addressed were:

(1) What commonality of requirements exists between the hi rel LSI users?

(2) Can some standardization be applied to limit the total number of types of LSI components which must be qualified for hi rel applications?

(3) Can standard specifications satisfy most user needs?

B. PROCEEDINGS

1. Definition of Levels of Standardization

The first order of business was to define standardization and what levels of standardization should be considered by the group. Dr. Ralph Martinez of NELC had done considerable work in producing a Navy position paper on LSI and presented a list of standardization levels as follows:

(1) Architecture, instruction set, functions

(2) Interface

(3) Microcomputer (monolithic CPU)

(4) Microcomputer configured from microprogrammable parts (bit slice)

(5) Set of microcomputer and support system specifications

(6) Higher order language (HOL).

The list presented by Dr. Martinez was considered to be an adequate list to be addressed by the group.

2. Presentation of Microprocessor and Associated LSI Requirements

Representatives from NASA (JPL), the Air Force (AFAL), and the Navy (NELC), each presented a list of microprocessor and associated LSI requirements. These requirements are considered to be "long term" requirements (those which must be satisfied by 1980).

(1) NELC microprocessor/LSI requirements (R. Martinez)

(a) Functional requirements:

Architecture: 8 bit, 16 bit, bit slice
Microprocessor support modules:

CPU support
ROM/RAM memory
UART/USRT
Programmable ROM (PROM)
Erasable PROM (EROM)
Programmable parallel I/O interface
Programmable interval timer
Programmable interrupt controller
Programmable DMA controller
Hardware multiply/divide unit
Floating point unit
Fast Fourier transform unit

(b) Speed/environment/power requirements:

Speed: Fixed architecture for $\leq$ 200 kops
Variable architecture for 200 $\leq$ kops $\leq$ 500

Temp: Two categories:
1. -55 to 125°C
2. -40 to 85°C

Power: Need low-power technology (CMOS, CMOS/SOS, I^2L)

Radiation: Radiation-hardened $\geq 10^6$ rad (Si)

Reliability: MIL 38510

(2) Air Force Microprocessor/LSI Requirements (J. Decaire)

(a) Functional requirements
Architecture: bit slice desirable

(b) Speed/environment/power requirements

Temp: Full mil range

Power: New low-power technology (prefer CMOS/SOS)

Radiation: Need hard technology for both total dose and transient radiation environments

Reliability: MIL 38510

(3) NASA microprocessor and LSI requirements (D. Rennels, JPL)

(a) Microprocessor capability

Approximately 8080 capability for near-term applications.

Require multiple DMA capability

Byte slice architecture desirable for emulation of mini or other microcomputer

Op rate of emulated machine $\leq$ 200 kops

Need capability to add special function units like hardware multiply/divide, floating point, etc.

(b) Environment/reliability

Need low-power technology (CMOS, CMOS/SOS, I^2L).

Need radiation-hardened for some limited applications (e.g., Jupiter missions).

Mil temperature range.

Need reliability for five-year space missions.

3. Generation of Strawman Specification for Standard Interagency Microprocessor and Associated LSI

From the requirements shown by the various representatives there are significant areas of commonality between the Air Force, Navy, and NASA for microprocessor and associated LSI requirements. Because all agencies had both near-term (1976 through 1980) and long-term (1980 and beyond) requirements for microprocessors, two strawman specifications were generated; one reflecting near-term needs, and one reflecting long-term needs. In either case the specifications represented a consensus of the agencies represented. The specifications are given in Tables 10-1 and 10-2.

Table 10-1. Strawman Specification for Microprocessor and Associated LSI for Near-Term (Present Through 1980) Application

A. Microprocessor and associated LSI requirements definition

Development---------based on commercially available microprocessor

Microprocessor word size----------------------------------8 bits

Speed --≤200 kops

Support modules --

CPU support

ROM/RAM memory

UART/USRT

Programmable ROM (PROM)

Erasable PROM (EPROM)

Programmable parallel I/O interface

Programmable interval timer

Programmable interrupt controller

Programmable DMA controller

Multiply/divide unit

B. Environmental requirements

Temperature: -55 to +125°C

Radiation: radiation-hardened ≥10^6 rad (Si)

C. Product assurance:

MIL 38510

D. Other

Need low power for space application.

Second sourcing for all modules.

Table 10-2. Strawman Specification for Microprocessor and Associated LSI for Long-Term (1980 →) Application

A. Microprocessor and associated LSI requirements definition

Development---------based on commercial microprocessor government development

Microprocessor word size-----------------8 bit slice architecture

Speed --$\leq$200 kops $\leq$ 500

Support modules ----------------------------------

CPU support (timing, buffers, etc.)

ROM/RAM memory

UART/USRT

Programmable ROM (PROM)

Erasable PROM (EPROM)

Programmable parallel I/O interface

Programmable interval timer

Programmable interrupt controller

Programmable DMA controller

Multiply/divide unit

Floating point unit

Fast Fourier transform unit

Mil standard 1553A serial bus interface

B. Environmental requirements

Temperature: -55°C

Radiation: radiation-hardened $\geq 10^6$ rad (Si)

Table 10-2. Strawman Specification for Microprocessor and Associated LSI for Long-Term (1980 →) Application (Continuation 1)

C. Product assurance:

MIL·38510

Consideration for "built-in" testability

D. Other

Low power equivalent to CMOS technology for space application.

4. Summary of Applicable Microprocessor/LSI Developments Under Way

a. General Processor Unit. The development of a general processor unit (GPU) sponsored by the AFAL and described by John Decaire stirred considerable interest in the group. The GPU is an LSI device to be implemented in the CMOS/SOS process which is designed to be used as a bulding block in the implementation of an arithmetic logic unit and register file of a general-purpose digital microcomputer. Some of the salient features of the GPU are:

(1) Performs basic parallel arithmetic, logic and storage functions.

(2) 8-bit slice architecture.

(3) Contains data paths, control, and status information to implement multiply, divide, and floating point algorithms from microcommand inputs.

(4) CMOS technology: silicon on sapphire process.

(5) Microprogrammable for emulation of other target micro or minicomputer.

It was the consensus of the group that the GPU development could satisfy a significant percentage of the user requirements defined in the long-term strawman specification. John Decaire took the action to make information relative to the GPU available to interested parties. The schedule for GPU development is given in the Figure 7-6.

b. NELC Position Paper Recommendations on Standardization. A significant effort was undertaken by the Navy to provide recommendations relative to standardization of microprocessors/microcomputers for Navy systems. The product of the Navy study is a position paper whose recommendations are summarized as follows:

(1) Standardize on families of 8-bit, 16-bit and bit slice functional modules.

(2) Standardize immediately on 8-bit CPU as an interim standard.

(3) Standardize on critical support modules for the 8-bit CPU.

(4) Select commercially available 8-bit CPU, support hardware, and support software by competitive procurement.

(5) Use the Navy's Standard Electronic Module Program to implement the standardization scheme.

(6) Require second sources for all LSI parts and modules.

(7) Require full set of hardware/software support systems for each module family.

(8) Select a standard instruction set by the choice of the 8-bit, 16-bit or bit slice CPU's.

(9) Proposed schedule:

8-bit modules	Aug 1976
16-bit modules	Oct 1976 - Aug 1977
Bit slice modules	Oct 1976 - Aug 1977

Most of the Navy recommendations were considered by the group to have universal application to the other agencies represented.

c. Air Force (Rome Air Development Center) Microprocessor Qualification Program. RADC has a significant comprehensive reliability program under way to perform product evaluation, electrical characterization, and reliability assurance on a significant number of microprocessors, memories, and support chips. The RADC reliability program technical activities include the following:

(1) Product evaluation (P): ambient gas analysis, functional layout, physical properties, manufacturing processes.

(2) Electrical characterization (E): fault detection, fault isolation, performance validation (electrical) mil-spec preparation.

(3) Reliability assurance (R): life testing, screening and qualification, failure rate prediction, special analytical techniques.

A list of microprocessors, memories and support chips to be included in the plan is given in Table 10-3.

C. CONCLUSIONS, RECOMMENDATIONS AND FOLLOW-UP ACTIONS

The conclusions reached by Group A and recommendations and follow-up actions to be accomplished prior to the next user/manufacturer workshop were summarized and presented to the workshop general meeting by the group chairman, Richard Urban. The conclusions and recommendations are as follows:

1. Conclusions

(1) Commonality exists between the Navy, Army, Air Force and NASA in the need for high reliability, radiation-hardened microprocessors.

(2) The commonality translates into the need for an 8-bit, slice, radiation-hardened (CMOS/SOS) microprocessor for the 1980 time frame.

(3) Standardization is a viable approach to reduce costs and provide leverage to microprocessor manufacturers.

(4) Benefits can be achieved through subsystem interface standardization (data busses, etc.).

(5) There are no microprocessors (now or near future) which will satisfy radiation-hardened requirements without a large investment.

(6) There may be potential for intergovernmental microprocessor standardization in other than radiation environments.

(7) The Air Force CMOS/SOS development efforts are consistent with intergovernmental standardization requirements.

2. Recommendations

(1) Pursue the development of microprocessor, capitalizing on the existing Air Force CMOS/SOS work to date. The desired characteristics of such a microprocessor are:

8-bit slice architecture

Microprogrammability

Table 10-3. Rome Air Development Center Reliability Plan for LSI

Type	Variations	Technology	Initial Avail FY (otr)	Requirements
Microprocessors				
8080A	3	NMOS	76	P*,E*,R
6800	2	NMOS	76	P*,E*,R
2900	3	LST^{2}L	77 (1)	P,E,R
3000	2	ST^{2}L	77 (1)	P,E,R
1802	2	CMOS	77 (2)	P,E,R
Z80	1	NMOS	77 (2)	P
5701	2	ST^{2}L	77 (2)	P
6100	2	CMOS	77 (3)	P,E,R
9900	1	I^{2}L	77 (4)	P,E,R
UDAM	1	NMOS	78 (1)	P,E
PPS series	2	PMOS	78 (1)	P,E
Hard	1	CMOS/SOS	78 (2)	P,E,R
10800	1	ECL	78 (3)	P,E
	23			
Memories				
4K dynamic	8	NMOS-I^{2}L	77 (1)	P,E,R
4K static	8	NMOS-I^{2}L	77 (4)	P,E,R
16K	5	NMOS	77 (4)	P,E,R
PROM	10	NiCr,AIM, UV	77 (1)	P,E,R
ROM	8	T^{2}L	77 (1)	P
BAROM	3	NMOS	77 (1)	P,E,R
Specials	4	CCD, bubbles	77 (3)	P,E
	46			
Support Chips				
In/out	26	Compatible with	77 (1)	P,E,R
Clock gen.	26	microprocessor	77 (1)	P,E,R
Memory access	26	family	77 (1)	P,E,R
Trans/rec	26		77 (1)	P,E,R
	104			

*Work completed

CMOS technology, SOS process

Software compatibility (emulation)

Multiple data busses (serial, parallel)

LSI device/subsystem reliability and testability for hi rel applications

Environmental mil temperature range and radiation hardened

Availability of special function units.

(2) Continue dialogue between agencies at the working level (workshop participants) to better understand requirements and particulars of individual agency plans relative to microprocessor application.

(3) Develop a mechanism for establishing the potential hi rel LSI market prior to the user/manufacturer workshop.

D. FINDINGS

Dick Urban reported on the conclusions and recommendations reached by Working Group A: Commonality of Requirements and Potential for Standardization. The major conclusion was that some commonality does exist between the services' representatives in the high-reliability radiation-hard microprocessor area, where we focused most of our discussion. That commonality translated into the need for an 8-bit slice microprocessor that is rad hard. It was also felt that the CMOS/SOS offers the greatest potential for meeting those requirements. And we were looking for that requirement to be satisfied in the 1980 time frame.

We also concluded that standardization is a viable approach to reduce costs and to provide leverage. What we mean by leverage is that if we all get together and voice our requirements to industry, they will see a greater potential for them to develop technology in the area that will satisfy some of our requirements. How great that leverage is, we don't know, but it is better than we have now.

We looked at where we could achieve the best benefits from standardization, and that seemed to be at the subsystem interface; that is, within a box versus external to a box in a 1553A standard where we could integrate common functions so everyone would know the interface requirements for their different subsystems. We looked at what we could do today, if anything. What microprocessors were available today that could satisfy our needs or what microprocessors were available today into which we could pump a little money in order to meet our needs. We concluded that there really isn't anything available, that trying to rad-harden an existing microprocessor would take a lot of dollars and it wouldn't satisfy our time frame needs.

Also, in other than the high-reliability, rad-hard environmental requirement, there may be some potential for intergovernmental standardization on current microprocessors in a ground support type of area where the Air Force, the ground-based NASA systems, and the Navy have some low-power, low-performance requirements.

John DeCaire presented some of his efforts in CMOS/SOS on the first day of the conference. The people in the working group thought that work was fairly consistent with the type of requirements that they were looking for in the 1980 time frame.

The working group people spent a lot of time talking about the 1802 and were under the impression that Sandia was going to rad-harden the 1802's in the near future; however, they didn't see that as a near-term solution. They didn't think Sandia was going to be that fast. If they are, it should certainly be discussed further. The consensus was that they would like to capitalize on existing programs, if that's possible.

The recommendations that came out of the working group were that we should pursue the development of a CMOS/SOS rad-hard 8-bit slice. We should capitalize on the current Air Force CMOS/SOS work. That work will go on no matter what we decide here. If we can capitalize on that, it will be to our advantage. We may want to dump some money into that effort and get the results a little bit sooner if the Air Force will provide that support. Also, we figured that standardization issues are very controversial. We had a lot of stimulating discussions. Many people with wide and varied backgrounds had different ideas which should probably be followed up. It appears that this conference has gone a long way toward establishing lines of communication which will promote such a follow-up.

SECTION XI

WORKING GROUP B: LSI QUALIFICATION MECHANISMS

R. Conklin
Air Force Avionics Laboratory

P. Lecoq
Jet Propulsion Laboratory

This group was charged with discussing qualification mechanisms suitable for LSI. It was to investigate the classical qualification process to determine the changes needed to accommodate LSI and indeed to determine whether they were appropriate at all. The following questions were posed:

(1) Can LSI devices really be qualified? What has to be different for LSI?

(2) Can standard qualification specifications be developed and used?

(3) What are the impacts of various parameters on qualification (technology, line, MFGR, packaging)?

(4) Can quality and testability be built into an LSI device instead of being evaluated after delivery?

(5) What qualification mechanisms are appropriate for custom LSI?

Group B included people from a broad background; most had considerable experience in hi rel parts or systems. Even so, there was some difficulty in defining qualification and characterizing the process as it applies to LSI. We agreed that LSI, particularly microprocessors, are too complex to test completely, but that adequate tests and an extension of 38510 should be sufficient for most hi rel applications. Major Ehrenfried of Rome Air Development Center reported after the group meeting that RADC is actively working on slash sheets for several microprocessors. There was considerable discussion on how manufacturing processes would be qualified without the government inspection and control of the process itself. The group recognized the need for a much more complete market survey before we can expect the manufacturers to spend money trying to sell to that market.

A. QUALIFICATION

Qualification is a means by which one prejudges a manufacturer on his ability to deliver a part to meet a predetermined set of specifications:

(1) Specifications must be as complete as possible.

(2) Specifications must be derived from real applications and needs.

(3) Specifications must derive from worst-case analysis.

(4) Parts must be testable.

Attachment 1 summarizes the group's attempt to define the necessary steps to qualification of LSI. Each of the seven basic steps addresses one aspect of the process. In general, the group agreed that qualification specifications will have to include more complex tests, but there is some potential for relaxing many of the parts-related requirements. We felt that manufacturers could contribute significantly to the qualification process by providing at least the following information:

(1) A reasonable description of the parametric performance of the parts to work within a system. This should include ac and dc measurements, I/O voltages and load specifications, speed, etc., over the temperature and voltage ranges allowed.

(2) Construction and packaging information.

(3) Extensive data to define accelerated life test criteria.

(4) Test-equipment-independent test patternization.

All of the above information must be obtained from more than one source to assure that the data is valid and does not tend to favor one manufacturer's product at the expense of others.

B. VISIBILITY INTO THE PRODUCTION LINE PROCESSES

Qualifying the line of an LSI manufacturer presents new problems. More emphasis will be required on inspecting and testing the end product of a line as it becomes more difficult to inspect and control the process itself. Even so, a description and understanding of the process and its control must be a prerequisite to any qualification program. Manufacturers will be reluctant to release proprietary processes and parameters. They will also object to any control or interference with their commercially productive line.

Although determining the quality of an inaccessible line is not completely realizable, several mechanisms expose potential failure mechanisms rather well. Visual inspections can indicate basic flaws, misalignments and other first-order problems, but are time consuming. Chemical tests reveal information about device quality. Sample destructive testing can indicate wafer lot quality. Special test chips included on wafers can make these tests relatively straightforward.

C. TESTING LSI

The visual, chemical, and sample destruct tests should verify that there are no stuck nodes; the speed versus temperature characteristics must be demonstrated over the full range. Pattern and instruction sensitivity must be detected. Care must be taken to differentiate between a designed pattern sensitivity and bad parts.

Test specifications must derive from real requirements, but will be iterated as more is learned about the system being qualified. Complete tests of these devices are out of the question, but comprehensive tests are possible, although expensive. The basic 8080 test, which will probably not suffice for hi rel devices, took about six man-years to develop. If the manufacturer must perform these extensive tests, it could make devices too expensive. Government agencies might be better off running the tests themselves. Test fallout devices might be useful in non-hi-rel applications, helping to make user testing cost-effective.

D. LSI AS SYSTEMS

There are some fundamental differences between LSI devices now and transistors when they were first qualified. LSI devices cannot be fully tested. They are really systems-on-a-chip and must be treated as such. Another fundamental difference is that the military users are no longer the controlling influence. The tremendous commercial market is directing the evolution of designs in LSI. Fewer LSI parts will be used than MSI and SSI, so for economic reasons, smaller lot samples should be required.

Each LSI device tends to be used as a component of a larger system. For example, one normally uses an Intel clock for an Intel microprocessor. The group felt that the parts that make up a family should be qualified as a family, not as separate, unrelated components. We recommend that a family be specified by a single slash sheet with dash numbers for the separate members. We felt that the ability to interact with other family members was a more important parameter than basic parts parameters. Memories can be excluded from this recommendation since they tend to be so widely used. We all felt that 38510 is still a valid qualification mechanism for LSI, but that it would have to become more like systems specifications and less like classical parts specifications. Bus parameters and other system level items must be included.

E. SOFTWARE QUALIFICATION

The instruction set of a microprocessor and other similar "software" aspects must be included in the specification. A high order language, if available for the microprocessor being qualified, should be specified as tightly as the basic instruction set. Although it would not be appropriate for 38510, we recognized that software can be less reliable than the hardware. Qualification of reliable software would be a fertile field to investigate.

F. POTENTIAL USE OF LSI

No one from Rome Air Development Center, the prime center involved in MIL qualification, was able to attend this discussion. We all agreed to the necessity of working with them in any program. A much more thorough user survey must be made to determine basic requirements and usage levels. Perhaps the results of some RACC analysis can be the basis of such a survey.

G. HOW LSI FITS INTO THE QUALIFICATION PROCESS

1. Characterization

This step calls for learning about all important parameters of the LSI device. This is considerably more complex for LSI devices than the SSI and MSI we are used to. All parameters should be defined in terms of "min/max" rather than "typical." An understanding should be gained of the functional capability, instruction set, speed, and temperature range. This characterization should come primarily from the manufacturer and his specifications.

2. Specification

This step is particularly critical for LSI qualification since the LSI will require far more extensive specifications than MSI. Recognizing that a microprocessor is a system rather than a mere part will call for more system level specifications and less reliance on gate parameter tésts. 38510 specifications should derive from manufacturer's specifications as well as application requirements. In cases of multiple source qualification, care must be taken to identify characteristics which differ between sources. Only those characteristics which are common to all sources should be included in the specifications.

The following areas should be adequately covered in the specifications:

(1) Design criteria and constraints.

(2) Design rules.

(3) Process control requirements. Qualification of line and special means for evaluating product quality at the line level.

(4) Product assurance systems requirements.

(5) Performance requirements.

(6) Software. The instruction set is definitely a part of a microprocessor and should be included in the specification. Some means is required to qualify even applications software.

3. Environmental

(1) Operating temperature range

(2) Vibration

(3) Thermal cycles

4. Operating Life

(1) Life tests should be run on parts running as a system.

(2) There will only be a small number of parts qualified, so sample size will be small.

5. Testing

Test requirements should be specified independent of the test machine. The requirements will have to be adapted ir qualification test equipment is different from that the manufacturer uses. Test sequences for hi rel parts are generally the same regardless of the target application. It might be possible to specify applications-oriented test sequences as dash number specifications for the part. This could shorten test cycles and increase yield, since all components would not have to pass all tests. Differences might include device speed, temperature limits, ability to perform specific tasks, etc. Manufacturers would prefer one test rather than several, but it might be an effective approach and deserves further discussion.

6. Screening

This step specifies the performance test requirement and lot sample size.

7. Visual Inspection

This step allows a window into the production process revealing some potential flaws in devices which could not be detected without full line inspection. There was some discussion on the need for visual inspection since it requires so much time to inspect large chips. The feeling of the group was that it snould be done.

SECTION XII

WORKING GROUP C: TESTING MICROPROCESSORS AND OTHER LSI

W. Richard Scott
Jet Propulsion Laboratory

This working group opened with the chairman restating the key issues of the working group:

(1) How can a "system-on-a-chip" be tested?

(2) Can testability be designed into LSI?

(3) Which potential LSI users are presently performing LSI testing?

(4) How can tests be specified for LSI?

Several of these questions had been touched upon, if not thoroughly discussed, in Wednesday's general session. For instance, questions 1 and 2 were the subject of Prof. Carver Mead's discussion. It is Prof. Mead's position that the only conceivable approach to exhaustively testing a complex system-on-a-chip is through "structured design" of the chip. This technique, Prof. Mead stated, offers the only means by which we could reduce the time required to exhaustively test a device of the complexity of the 8080 (estimated to be of the order of 10^{22} years) to an economical manageable task.

Major Charles Ehrenfried of RADC discussed the Microcircuit Reliability Program that he is managing for the Air Force. This is a three-year, 90-man-year program concentrating on the evaluation, characterization and reliability assurance aspects of 23 microprocessors, 46 memories, and 104 peripheral devices.

The product evaluation task looks at ambient gas analysis, functional layout, physical properties and manufacturing process. The electrical characterization consists of fault isolation, fault detection, electrical performance validation, and preparation of MIL-M-38510 slash sheets. Under the reliability assurance task, they will be conducting life tests, screening and qualification tests, predicting failure rates, and developing any necessary special analytical techniques.

While the value or applicability of the RADC approach was challenged by some, no one had a better alternative approach to offer. Perhaps the RADC approach is not the best; however, it is a best shot at doing something right now, attacking immediate problems.

Glen Case, a working group panel member from Sandia, presented a recommendation somewhat similar to Carver Mead's. Glen's approach is one of redesigning the devices and additionally pinning-out certain circuits or functional elements not otherwise accessible such that more complete testing is made possible. To accomplish this, Sandia developed an in-house process capability and, using a standard cell approach, set out to design its own devices. Ideally, Sandia would develop the technology, develop device masks, and turn them over to a commercial manufacturer to be fabricated.

Glen discussed the computer-aided design programs in use at Sandia. These include logic minimization and simulation programs, circuit analysis codes, automatic mask layout programs, fault analysis and test sequence generation programs, error checking routines, and interactive graphics software and mask analysis codes for use in designing logic cells, test devices, and special custom circuits. Glen further discussed the Sandia logic simulation (SALOGS), which provides a logical function test to assure proper circuit operation and the absence of race conditions.

John Shea of ICE Engineering Corp, another working group panel member, reported to the working group concerning the semiconductor manufacturers' meeting held in Cupertino, Calif., earlier in the week. Most of the semiconductor manufacturers from "Silicon Valley" were represented. The main concern voiced at that meeting was the unrealistic MIL-M-38510 specification requirements for testing of linear devices. After having processed certain linears per the MIL-M-38510 requirements, the yield on final electricals was less than 10%. Concurrent with their concern for yields is the fact that after having the manufacturers qualify product to 38510 and subsequently bid procurement requisitions to 38510, the DoD is not requiring its contractors to purchase parts to the mil spec but rather allows procurement of commercial devices. John suggested that unless we take a firm position for MIL-M-38510 across the board, the mil parts may never be available. Further, he stated that when we do purchase to 38510, we must enforce all the requirements and better train our source inspection to assure that all requirements are satisfied. John feels that if the customer's top management meet with the manufacturer's top management to discuss the overall program plans and théreby improve the vendor's total visibility, the whole procurement process will be much improved.

After considerable discussion it was acknowledged by the working group as a whole that the current generation of LSI devices has not been designed to be and indeed is not exhaustively testable. This presents two problems: a short-range problem of how we adequately test the presently available product for reliability, and, a long-range problem of how we design testability into the next generation product. Because this working group met for only one day it was immediately decided to concentrate on the short-range problem and deal with the long-range problem at another time. This is not to say that the attempted solution should await the solution of the short-range problem. They are both serious problems and must be worked simultaneously.

The principal costs of MPU testing were identified as test equipment capital expenditures and the cost of preparing the software test programs. Because of concern over the magnitude of these two cost drivers, two suggestions were made. First, an alternative not to be overlooked is thorough MPU testing by the MPU manufacturer starting at wafer level, with close surveillance and monitoring by the user/

customer. However, in light of the problems often experienced in the past with such an approach, a second alternative, one favored by all working group members, is the formation of a government test equipment users group. Because NASA, Navy and RADC all have Tektronics 3260 test systems, it was felt that a significant savings could accrue to the government by the immediate establishment of a Government 3260 Users Group. The group would exchange program tapes and detailed program plans to help avoid needless duplication, and even make available machine time, especially when dealing with common problems. Such a users group could not only save considerable resources but allow the various users to move ahead to solve a broader range of testing-related LSI problems.

It was also concluded by the working group that, unless we bring this hands-on approach to LSI testing back in-house, we will be left helplessly behind the fast-moving LSI technology.

Software reliability is another area which appears to have been completely overlooked at the parts testing level. What is it and who is or should be responsible for it are just two questions which must be addressed. It was the recommendation of this working group that a governmental group be formed to study this problem area and to assure that it is well-covered in the second conference we sponsor.

Another area which received considerable discussion was failure analysis. The question which needs attention is what do people really need and want from LSI failure analysis? Can we expect to apply the SSI and MSI failure analysis techniques to LSI? Whether or not, it was agreed that before one can solve the failure analysis problem, he must first solve the LSI test problem.

APPENDIX A

SUMMARY LSI REQUIREMENTS FORECAST

In preparation for this conference a questionnaire was provided to interested attendees. This questionnaire and a summary of the responses are reproduced in Attachments 1 and 2.

ATTACHMENT 1

MICROPROCESSOR AND ASSOCIATED LSI SPACECRAFT APPLICATIONS AND REQUIREMENTS FORECAST

1. Microprocessor and Associated LSI Application Forecast

Please indicate the spacecraft instrument, device, or system for which the use of a microprocessor and associated LSI is planned or under consideration.

Instrument or Device Using Microprocessor and Associated LSI	Host NASA Project	Quantity Required

____

2. Microprocessor and Associated LSI Requirements Definition

Please indicate the microprocessor and associated LSI characteristics that would most nearly meet your requirements for the applications listed above.

Microprocessor Unit (MPU) Characteristics:

Microprocessor Word Size ____________________ bits
Addressable Memory ____________________ K words
Add Time (reg. to reg.) ____________________ μ sec
Hardware Multiply/Divide. . . . Req'd ____________ Not Req'd ____________
Direct Memory Access. . . . . . . Req'd ____________ Not Req'd ____________
Microprogrammable . . . . Req'd ____________ Not Req'd ____________
Hardware Floating Point . . . . Req'd ____________ Not Req'd ____________
Indexed and Indirect
Addressing. . . . . Req'd ____________ Not Req'd ____________
Interrupt Req'd ____________ (No. Channels) Not Req'd ____________
Power ____________ W
Architecture (Byte slice, fixed word length, etc.) ____________
____

Special features of MPU instruction set ______________________________

____

Other characteristics ____

____

Memory Characteristics:

	Random Access (RAM)	Read-Only (ROM)	Programmable ROM (PROM)	Erasable PROM (EPROM)
Type				
Max Size (K Words)				
Memory Increment (K Words)				
Power (W)				
Read Access Time (μ sec)				
Read/Write Cycle Time (μ sec)				

Other characteristics: (Please comment if applicable)

Input/Output or Peripheral Device Requirements:

	Req'd	Not Req'd
Programmable parallel I/O interface	_____	________
Programmable serial I/O interface	_____	________
Programmable DMA controller	_____	________
Programmable interrupt controller	_____	________
Programmable interval timer	_____	________
Universal Asychronous receiver/ transmitter (UART)	_____	________
Analog to Digital Converter	_____	________
Digital to Analog Converter	_____	________
Other: (please list other required I/O devices and/or peripheral devices if applicable		

Environmental Requirements:

Temperature Range (^{o}C) __________ to __________
Radiation (K rad {Si} __________ K rad (Si)
Other ______________________________________

Product Assurance Requirements:

Support Requirements:

Support Hardware:

Prototyping system . . . . Req'd________ Not Req'd________
Time share computer service. . . Req'd________ Not Req'd________
Batch computer systems . . . . . Req'd________ Not Req'd________
Other________________________________

Support Software:

Resident assembler on prototyping system. .Req'd______ Not Req'd______
Resident software aids including. . . . . .Req'd______ Not Req'd______
editor, debugger, diagnostic, and
utility programs
Resident higher order language (HOL). . . .Req'd______ Not Req'd______
on prototyping system
Cross-assembler . . . . .Req'd______ Not Req'd______
Software Simulator. . . . . .Req'd______ Not Req'd______
Other______________________________________

3. Preferred Microprocessor and other LSI Devices

Please indicate the Microprocessor and other LSI devices which are your present preference for the applications identified in 1) above.

4. Preferred LSI Packaging

For those devices not available as standard commercial devices, please indicate your preference for hybrid packaging vs. custom LSI packaging.

ATTACHMENT 2

MICROPROCESSOR AND LSI A.'PLICATIONS

	Military	Space	Commercial
Guidance/Nav./Control	6	7	1
Data Processing	1	2	-
Telemetry/Comm.	1	2	-
Scientific/Instrument/ R&D	-	13	2
Test Equipment	1	1	-

Note: Dedicated data processing not included in data processing.

MICROPROCESSOR AND ASSOCIATED LSI REQUIREMENTS

Word Size:	8 bits - 9	12 bits -2	16 bits - 9	32 bits - 1
Addressable Memory:	>64K - 3	64K - 5	∠ 64K - 7	
Reqd. Memory:	≤ 64K - 4 ≤ 8K - 3	≤ 32K - 1 ≤ 4K - 3	≤ 16K - 3	
Add Time:	≫ 10 μs - 1 ≤ 2 μs - 7	≤ 10 μs - 1 ≤ 1 μs - 1	≤ 5 μ - 2 ≤ .5 μs - 2	

	Reqd.	Not Reqd.
Hdwr. Multi./Div.	12	4
DMA	14	2
u Programmable	10	7
Hdwr. Floating Point	11	7
Indexed/Indirect Addressing	11	2
Interrupt	10	1

No Channels	≤ 32C - 1 ≤ 4C - 4	≤ 16C - 4 1C - 1	≤ 8C - 2
Power	≤ 25W - 2 ≤ 2W - 1	≤ 10W - 1 ≤ 1W - 1	≤ 5W - 3 ≤ ½W - 1
Architecture	Byte Slice - 6	Fixed Word Length - 9	

MEMORY CHARACTERISTICS

	RAM	ROM	PROM	E PROM
TYPE	13	8	10	7
TECH				
CMOS	1	2		
NMOS	1	1		
Bipolar	1	1		
MAX SIZE				
≤ 1K	2	1	2	2
≤ 4K	2	2	2	1
≤ 8K	2	3	1	
≤ 32K	2	2	2	2
≤ 64K	3			
INCREMENT				
≤ 1K	3	3	3	3
≤ 4K	4	4	2	1
≤ 8K	1			1
≤ 16K	1			
POWER (ea. Increment)				
≤ 10 mW	1	1		
≤ 0.5W	2	2	2	2
> 1W	1		1	2
ACCESS TIME				
≤ .5 µS	6	2	3	1
≤ 1 µS	1	2	1	1
Greater	1		1	
R/W CYCLE				
≤ .5 µS	3	2	1	1
≤ 1 µS	3	1		1
Greater	1		1	

INPUT/OUTPUT OR PERIPHERAL DEVICE REQUIREMENTS

	Reqd.	Not Reqd.
Programmable parallel I/O interface	13	2
Programmable serial I/O interface	12	3
Programmable DMA controller	12	3
Programmable interrupt controller	12	3
Programmable interval time	12	4
Universal Asychronous receiver/ transmitter (UART)	12	4
Analog to Digital Converter	13	2
Digital to Analog Converter	10	4

Other:

Standard Hardware

Multi./Div. Unit

FFT

MIL STD 1553A I/F Bus

SUPPORT REQUIREMENTS

	Reqd.	Not Reqd.
Support Hardware:		
Prototyping system	14	0
Time share computer service	3	10
Batch computer systems	4	8
Support Software:		
Resident assembler on prototyping system	12	2
Resident software aids, including editor, debugger, diagnostic & utility programs	14	2
Resident higher order language (HOL) on prototyping system	4	8
Cross-assembler	10	4
Software Simulator	5	9

ENVIRONMENTAL REQUIREMENTS

TEMPERATURE

-55°C to 125°C	5	0°C to 100°C	1
-40/-55°C to 80/85°C	2	20°C to 75°C	1
-20°C to 52/75°C	4		

RADIATION

$>10^6$K rad (Si)	1	Unknown/Unspecified	5
10^3 to 10^5K rad (Si)	3	N/A	3
25 to 30K rad (Si)	2		

OTHER

Vibration

MIL-E-16400

PRODUCT ASSURANCE

883 Class A/Equiv.	4	Commercial	1
883 Class B/Equiv.	2	Unspecified	7

PREFERRED MICROPROCESSOR

1802	6	2901	3
PPS 8	3	6502	1
8008A	1	Z80	1
6800	4	M10800	1
9900	2	ATMAC	1
8080A	5	6100	1
LSI11	1		

OTHER DEVICES

6800 Support Chips

8080 Support Chips

1802 Support Chips

2901 Support Chips

Custom Devices

MEMORIES

INTEL 5101

2102

MMI 5306

MWS 5501D

PREFERRED LSI PACKAGING

Limited response indicating no definite preference

APPENDIX B

CONFERENCE ATTENDEES
NON-JPL ATTENDEES

Case, Glenn R.
Sandia Laboratories
Box 5800
Albuquerque, New Mexico 87115
505) 264-2145

Conklin, Robert E. AFAL-DHE
Wright Patterson AFB
Ohio 45433
513) 255-4557

Davis, Sam
Electronic Engineering Times
22543 Ventura Blvd.
Woodland Hills, CA 91364
213) 347-3814

Decaire, John A. DHE
Air Force Avionics Laboratory
Wright Patterson AFB
Ohio 45433
513) 255-2562

Deve, Bernd
Air Force Weapons Laboratory ELP
Kirtland Air Force Base
Albuquerque New Mexico 87117
505) 264-0316

Ehrenfried, Maj. Charles E
Rome Air Development Center RBR
Griffiss Air Force Base, N.Y.
13441
315) 330-4921 Auto Von 587-4921

Fitzwilson, Roger L.
Questron Corp
101 Cont. Boulevard #855
El Segundo, CA 90245
213) 640-0901

Gaudiano, Saverio
NASA - L. B. Johnson SC
Houston, Texas 77058
ED-6
713) 483-4061

Goldsmith, T.C. - Code 724
NASA - Goddard SFC
Greenbelt, Maryland 20771
301) 982-5374

Hamiter, Leon
NASA - Marshall Space Flight
Center EC-43
Huntsville, Alabama 35812
205) 453-4572

Haratz, David
U.S. Army Electronics Command
NL-BP
Ft. Monmouth, New Jersey
201) 544-4525

Kiernan, Gerard F.
Goddard SFC - Sperry Support
Services
Code 310.1
Greenbelt, Maryland 20771
301) 982-6633 or 4838

King, Everett E.
Naval Research Laboratory
Code 5216
Washington, DC 20375
202) 767-2429

Langley, Frank J.
Raytheon Missile Systems Div.
Hartwell Road
Bedford, Massachusetts 01730
617) 274-7100 X2306

Ledbetter, Capt. James D.
Air Force Weapons Laboratory ELP
Kirtland Air Force Base
Albuquerque, New Mexico 87117
505) 264-0316

Martinez, Ralph
Naval Electronics Laboratory
Center
Code 4300
271 Catalina Blvd.
San Diego, California 92120
714) 225-2752

Matsumoto, Tak
NASA - Ames Research Center
244-7
Moffett Field, CA
405) 965-6570

McInnis, William E.
Los Cost Systems Office, Code KC
NASA
Washington, DC 20456
202) 755-3040

Malloy, H.
Intel/JC 133-13
3065 Bowers
Santa Clara, CA
408) 966-0585

Mosier, Denton B.
Electronic News
845 S. Figueroa St.
Los Angeles, CA 90017
213) 624-1981

Platteter, Dale
US Naval Weapons Support Center
Bldg. 2917 - Code 7024
Crane, Indiana 47522
812) 854-1854 or X1808

Pritchard, Gary K.
Air Force Avionics Laboratory - DHE
Wright-Patterson AFB
Ohio 45433
513) 255-3802

Shea, John D.
ICE - Suite 260
N. Winchester
Santa Clara, CA
408) 248-1843

Urban, E.C.
Naval Electronics Laboratory
Center
Code 4300
San Diego, CA 92152
714) 225-2752

Yearwood, Hoyle
NASA - Marshall Space Flight
Center EG-02
Huntsville, Alabama 35812
205) 453-5469

JPL ATTENDEES
(4800 Oak Grove Drive, Pasadena CA 91103)

Ashlock, John
JPL 238-625
213) 354-4519

Brown, Barbara D.
JPL 158-142
213) 354-7207

Bryden, James
JPL 198-223
213) 354-5489

Byers, Robert A.
JPL 180-805
213) 354-3732

Caplette, Russell
JPL 198-136
213) 354-4124

Case, Richard K.
JPL 233-307
213) 354-4569

Coffey, Lyndall D.
JPL T1180
213) 354-2918

Covey, Robert E.
JPL 158-205
213) 354-5590

Dodge, Stuart
JPL T1180
213) 354-5003

Ebersole, Mike
JPL 114-122
213) 354-4595

Felberg, Fred
JPL 180-501
213) 354-4241

Greenberg, Edward
JPL 114-122
213) 354-7257

Griswold, Thomas W.
JPL 198-229
213) 354-4145

Grunthaner, Frank J.
JPL 198-229
213) 354-5564

Hess, Larry M.
JPL T1180
213) 354-5527

Jenkins, George
JPL 161-228
213) 354-4217

Kavanaugh, Hubert M.
JPL 233-307
213) 354-7770

Lau, Gary K
JPL 233-307
213) 354-6480

Lecoq, Paul E.
JPL 114-122
213) 354-5517

Lee, Steven Y.
JPL 161-228
213) 354-6387

Lord, Donald D.
JPL 198-136
213) 354-4117

Mallen, Walter
JPL T1180
213) 354-3070

Maserjian, Joseph
JPL 198-229
213) 354-3801

Mead, Carver A.
116-81
California Institute of Technology
2036 Pasadena Glen Road
Pasadena, CA 91125
213) 705-1807

Miller, Emmett L.
JPL 158-205
213) 354-6026

Momberger, Paul
JPL T1180
213) 354-3510

Moore, Charles
JPL 198-136
213) 354-4126

Nagano, Satoshi
JPL 198-220
213) 354-6382

Newton, John W.
JPL 238-420
213) 354-6766

Nixon, Robert
JPL 114-118
213) 354-4268

Norton, Harry N.
JPL 233-307
213) 354-6732

Olson, Robert N.
JPL 233-307
213) 354-6306

Parker, Paul
JPL 126-300
213) 354-2069

Pinck, Melvyn
JPL 130-117
213) 354-3439

Ross, Darrell L.
JPL 198-220
213) 354-4049

Scott, Richard W.
JPL T1180
213) 354-5750

Shumka, Alex
JPL 158-205
213) 354-5059

Small, John
JPL 180-403
213) 354-3880

Smith, Ted
JPL 198-112A
213) 354-4822

Springer, L. R.
JPL 156-142
213) 354-4512

Tam, Man K.
JPL 161-228
213) 354-6288

Trigonis, Art C.
JPL 158-205
213) 354-6167

Tyree, Vance C.
JPL 156-142
213) 354-4263

Whitney, William M.
JPL 198-229
213) 354-4410

Wright, Larry W.
JPL 158-205
213) 354-5603

NASA-JPL-Coml., L.A., Calif