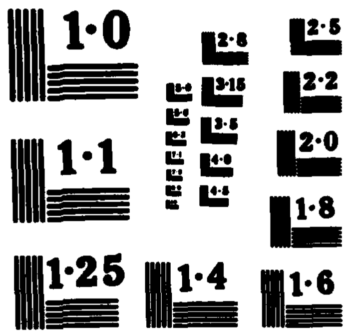
AD-A145 697 PROCEEDINGS PAPERS OF THE AFSC (AIR FORCE SYSTEMS COMMAND) AVIONICS STAND..(U) AERONAUTICAL SYSTEMS DIV WRIGHT-PATTERSON AFB OH DIRECTORATE O.. 1/6

UNCLASSIFIED C A PORUBCANSKY NOV 82 F/G 1/3 NL


1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8
3·2
3·6
4·0

ASD(ENA)-TR-82-5031

(6)

AD-A145 697

2nd AFSC STANDARDIZATION CONFERENCE

COMBINED PARTICIPATION BY:
DOD-ARMY-NAVY-AIR FORCE-NATO

30 NOVEMBER - 2 DECEMBER 1982
TUTORIALS: 29 NOVEMBER 1982

DAYTON CONVENTION CENTER
DAYTON, OHIO

SPONSORED BY:

ADDENDUM
To The
PROCEEDINGS

HOSTED BY

Approved for Public Release Distribution Unlimited

84 09 05 095

NOTICE

This technical report has been reviewed and is approved for publication.

JEFFERY L. PESLER
Vice Chairman
2nd AFSC Standardization Conference

ERWIN C. GANGL
Chief, Avionics Systems Division
Directorate of Avionics Engineering

FOR THE COMMANDER

ROBERT P. LAVOIE, COL, USAF
Director of Avionics Engineering
Deputy for Engineering

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: ASD(ENA)-TR-82-5031, VOLUME X — 2. GOVT ACCESSION NO. A145697	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): Proceedings Papers of the Second AFSC Avionics Standardization Conference	5. TYPE OF REPORT & PERIOD COVERED: Final Report 29 November - 2 December 1982
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Editor: Cynthia A. Porubcansky	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: HQ ASD/ENAS Wright-Patterson AFB OH 45433	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: HQ ASD/ENA Wright-Patterson AFB OH 45433	12. REPORT DATE: November 1982
	13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Same as Above	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

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

N/A

18. SUPPLEMENTARY NOTES

N/A

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

Computer Instruction Set Architecture, Multiplexing, Compilers, Support Software, Data Bus, Rational Standardization, Digital Avionics, System Integration, Stores Interface, Standardization, MIL-STD-1553, MIL-STD-1589 (JOVIAL), MIL-STD-1750, MIL-STD-1760, MIL-STD-1815 (ADA), MIL-STD-1862 (NEBULA).

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

This is a collection of UNCLASSIFIED papers to be distributed to the attendees of the Second AFSC Avionics Standardization Conference at the Convention Center, Dayton, Ohio. The scope of the Conference includes the complete range of DoD approved embedded computer hardware/software and related interface standards as well as standard subsystems used within the Tri-Service community and NATO. The theme of the conference is "Rational Standardization". Lessons learned as well as the pros and cons of standardization are highlighted.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

This is Volume 10

PROCEEDINGS OF THE

2nd AFSC STANDARDIZATION CONFERENCE

30 NOVEMBER - 2 DECEMBER 1982

DAYTON CONVENTION CENTER
DAYTON, OHIO

Sponsored by:

Air Force Systems Command

Hosted by:

Aeronautical Systems Division

FOREWORD

THE UNITED STATES AIR FORCE HAS COMMITTED ITSELF TO "STANDARDIZATION." THE THEME OF THIS YEAR'S CONFERENCE IS "RATIONAL STANDARDIZATION," AND WE HAVE EXPANDED THE SCOPE TO INCLUDE US ARMY, US NAVY AND NATO PERSPECTIVES ON ONGOING DOD INITIATIVES IN THIS IMPORTANT AREA.

WHY DOES THE AIR FORCE SYSTEMS COMMAND SPONSOR THESE CONFERENCES? BECAUSE WE BELIEVE THAT THE COMMUNICATIONS GENERATED BY THESE GET-TOGETHERS IMPROVE THE ACCEPTANCE OF OUR NEW STANDARDS AND FOSTERS EARLIER, SUCCESSFUL IMPLEMENTATION IN NUMEROUS APPLICATIONS. WE WANT ALL PARTIES AFFECTED BY THESE STANDARDS TO KNOW JUST WHAT IS AVAILABLE TO SUPPORT THEM: THE HARDWARE; THE COMPLIANCE TESTING; THE TOOLS NECESSARY TO FACILITATE DESIGN, ETC. WE ALSO BELIEVE THAT FEEDBACK FROM PEOPLE WHO HAVE USED THEM IS ESSENTIAL TO OUR CONTINUED EFFORTS TO IMPROVE OUR STANDARDIZATION PROCESS. WE HOPE TO LEARN FROM OUR SUCCESSES AND OUR FAILURES; BUT FIRST, WE MUST KNOW WHAT THESE ARE AND WE COUNT ON YOU TO TELL US.

AS WE DID IN 1980, WE ARE FOCUSING OUR PRESENTATIONS ON GOVERNMENT AND INDUSTRY EXECUTIVES, MANAGERS, AND ENGINEERS AND OUR GOAL IS TO EDUCATE RATHER THAN PRESENT DETAILED TECHNICAL MATERIAL. WE ARE STRIVING TO PRESENT, IN A SINGLE FORUM, THE TOTAL AFSC STANDARDIZATION PICTURE FROM POLICY TO IMPLEMENTATION. WE HOPE THIS INSIGHT WILL ENABLE ALL OF YOU TO BETTER UNDERSTAND THE "WHY'S AND WHEREFORE'S" OF OUR CURRENT EMPHASIS ON THIS SUBJECT.

MANY THANKS TO A DEDICATED TEAM FROM THE DIRECTORATE OF AVIONICS ENGINEERING FOR ORGANIZING THIS CONFERENCE; FROM THE OUTSTANDING TECHNICAL PROGRAM TO THE UNGLAMOROUS DETAILS NEEDED TO MAKE YOUR VISIT TO DAYTON, OHIO A PLEASANT ONE. THANKS ALSO TO ALL THE MODERATORS, SPEAKERS AND EXHIBITORS WHO RESPONDED IN SUCH A TIMELY MANNER TO ALL OF OUR PLEAS FOR ASSISTANCE.

Robert P. Lavoie

ROBERT P. LAVOIE, COL, USAF
DIRECTOR OF AVIONICS ENGINEERING
DEPUTY FOR ENGINEERING

DEPARTMENT OF THE AIR FORCE
HEADQUARTERS AIR FORCE SYSTEMS COMMAND
ANDREWS AIR FORCE BASE DC 20334

2 8 AUG 1982

REPLY TO ATTN OF: CV

SUBJECT: Second AFSC Standardization Conference

TO: ASD/CC

1. Since the highly successful standardization conference hosted by ASD in 1980, significant technological advancements have occurred. Integration of the standards into weapon systems has become a reality. As a result, we have many "lessons learned" and cost/benefit analyses that should be shared within the tri-service community. Also, this would be a good opportunity to update current and potential "users." Therefore, I endorse the organization of the Second AFSC Standardization Conference.

2. This conference should cover the current accepted standards, results of recent congressional actions, and standards planned for the future. We should provide the latest information on policy, system applications, and lessons learned. The agenda should accommodate both government and industry inputs that criticize as well as support our efforts. Experts from the tri-service arena should be invited to present papers on the various topics. Our AFSC project officer, Maj David Hammond, HQ AFSC/ALR, AUTOVON 858-5731, is prepared to assist.

ROBERT M. BOND, Lt Gen, USAF
Vice Commander

2nd AFSC STANDARDIZATION CONFERENCE

ORGANIZATION COMMITTEE

EXECUTIVE CHAIRMAN
Erwin C. Gangl

EXECUTIVE VICE CHAIRMAN
Jeffery L. Pesler

PROGRAM CHAIRMAN
Jerry L. Duchene

CO-CHAIRMAN
Harold J. Alber

CO-CHAIRMAN
Maj Lee Cheshire

CO-CHAIRMAN
David J. Krile

CO-CHAIRMAN
John Slivinski

EXHIBITS CHAIRMAN
Lt C.W. (Bud) Meyaard

PUBLICATIONS CHAIRMAN
Cindy Porubcansky

SPECIAL ARRANGEMENTS CHAIRMAN
Lt Dennis A. Shoulders

PROTOCOL OFFICER
Capt Francis A. DeCurtis

ADMINISTRATIVE CHAIRMAN
Marie P. Jankovich

TREASURER
Richard H. McBride

CONFERENCE MANAGER
Systems Productivity & Management Corporation

Table of Contents Volume 10

Luncheon Speakers

Executive Session

Table of Contents

Table of Contents

Table of Contents

MIL-STD-1815 Ada High Order Language

Table of Contents

Table of Contents

Advanced Standardized Systems/Subsystems

Standardized Software Development

Authors Index

EXHIBITORS

ARINC RESEARCH
BENDIX
BOMAR INSTRUMENTS
CIRCUIT TECHNOLOGY
CONTROL DATA CORPORATION
DELCO ELECTRONICS
FAIRCHILD CAMERA & INSTRUMENTS
FAIRCHILD SPACE & ELECTRONICS
GARRETT
GENERAL DYNAMICS, FT. WORTH
GENERAL ELECTRIC
GRUMMAN AEROSPACE
HARRIS SEMICONDUCTOR
IEEE
ILC/DATA DEVICE CORPORATION
INTELLIMAC
INTERMETRICS
KAISER
LITTON
LORAL
McDONNELL DOUGLAS ASTRONAUTICS
NORTHROP
RAYCHEM
RAYTHEON
RCA
ROLM
SANDERS ASSOCIATES
SCI SYSTEMS
SEAFAC
SINGER KEARFOTT
SMITH INDUSTRIES
SOFTECH
SPECTRAL SYSTEMS
SPERRY UNIVAC
STC COMPONENTS
SYSTEM PRODUCTIVITY & MANAGEMENT
SYSTEMS RESEARCH LAB
TELEDYNE SYSTEMS
TELEFONAKTIEBOLAGET LM ERICSSON
TEST SYSTEMS
TROMPETER
TRW
UTC/MOSTEK
WESTINGHOUSE DEFENSE

LUNCHEON SPEAKERS

Tuesday Luncheon Keynote Speaker

Major General Marc C. Reynolds

Major General Marc C. Reynolds is Commander of the Air Force Acquisition Logistics Division, and Deputy Chief of Staff for Acquisition Logistics, Air Force Logistics Command, Wright-Patterson Air Force Base, Ohio.

General Reynolds was born in Chamberlain, S.D., on June 2, 1928, and graduated from Chamberlain High School in 1946. He subsequently attended Dakota Wesleyan University and the University of Denver until the outbreak of the Korean War. He holds a Bachelor's Degree in Political Science from the University of Rhode Island and is a graduate of the Air Command and Staff College and the Naval War College.

General Reynolds entered the Air Force as an aviation cadet in January 1951 at Perrin Air Force Base, Texas, and was commissioned upon graduation from pilot training at Vance Air Force Base, Okalahoma, in February 1952. He then attended jet interceptor training at Moody Air Force Base, Georgia, and Tyndall Air Force Base, Florida.

In July 1952, General Reynolds was assigned pilot duty with the 83rd Fighter-Interceptor Squadron at Hamilton Air Force Base, California, and in September he moved with the squadron to Paine Air Force Base, Washington. In March 1953, he was transferred to the 4th Fighter-Interceptor Squadron at Naha Air Base, Okinawa, where he continued to serve as a fighter-interceptor pilot, flying the F-94B.

His next assignment, in September 1954, was Otis Air Force Base, Mass., where he served with the 437th and 60th Fighter-Interceptor Squadrons as a tactical and training flight commander, flying the F-94C and F-101B, and with the 602d Consolidated Maintenance Squadron as a maintenance officer.

General Reynolds was transferred to Europe in November 1961, assigned to the 10th Tactical Reconnaissance Wing, with duty at RAF Station Bruntingthorpe, England, as a Flight Commander, and later at Toul-Rosieres Air Base, France, as Chief of the Wing Standardization Evaluation Branch.

After Command and Staff College at Maxwell Air Force Base, Alabama, General Reynolds was assigned to the 22d Tactical Reconnaissance Squadron, Mountain Home Air Force Base, Idaho. In November 1966, he moved to the 460th Tactical Reconnaissance Wing at Tan Son Nhut Air Base, Republic of Vietnam, and flew 230 combat missions over North and South Vietnam in RF-4C.

(over)

Following his Southeast Asia tour, he served in Japan as Deputy Chief of the Reconnaissance Division, Headquarters Fifth Air Force, Fuchu Air Station. In April 1970, he moved to Misawa Air Base as Commander of the 16th Tactical Reconnaissance Squadron.

General Reynolds returned to the United States in February 1971, assigned to Shaw Air Force Base, S.C., where he served as Assistant Deputy Commander for Operations in the 363d Tactical Reconnaissance Wing. He attended the Naval War College at Newport, R.I., in 1972-73 and was subsequently assigned to Ogden Air Logistics Center, Hill Air Force Base, Utah, initially as the Director of Distribution and later as Director of Maintenance. In July 1976, he was transferred to McClellan Air Force Base, California, as the Director of Materiel Management, Sacramento Air Logistics Center. In March 1978, he became the Center Vice Commander. He transferred to the Air Force Acquisition Logistics Division in May 1980, where he served as Vice Commander until October 1981, when he assumed his present duties.

General Reynolds is a command pilot with more than 5,200 hours flying time, including 475 combat hours. His military decorations and awards include the Distinguished Service Medal, Legion of Merit, Distinguished Flying Cross, Meritorious Service Medal with one oak leaf cluster, Air Medal with 15 oak leaf clusters, and Air Force Commendation Medal with two oak leaf clusters.

He was promoted to Major General Sept 8, 1980, with date of rank July 1, 1977.

General Reynolds was married to the former Judy Coppage of Falmouth, Mass., who died in February 1982. Their children are Barbara and Scott.

MAJOR GENERAL MARC C. REYNOLDS

COMMANDER

AIR FORCE ACQUISITION LOGISTICS DIVISION

WRIGHT-PATTERSON AIR FORCE BASE, OHIO

LUNCHEON SPEECH TO 2D AFSC STANDARDIZATION CONFERENCE
STOUFFER'S DAYTON PLAZA HOTEL
30 NOVEMBER 1982

"A REFLECTION ON STANDARDIZATION"

It's a pleasure to be here today. As some of you know, I have more than a passing interest in standardization. So the opportunity to be one of your speakers is welcome.

I've looked over the schedule of events for this week. It's quite evident you'll be getting a concentrated dose of standardization. You've already had a day of tutorials which I endorse since it gives all of us a chance who don't specifically work in the area to get at least a baseline of understanding. This morning you heard the keynote speech by General McMullen, who set the stage for this conference. By the time the conference is completed, you'll have covered a variety of service perspectives, as well as the DOD and international level. In addition, you will have covered a session on policy and a number of working sessions specifically addressing the various architectural standards that we in the avionics and armament community deal with daily.

Clearly, the last thing you need from me today is yet another dissertation on the details of standardization. So, rather than preach to the choir, I ask you to step back for the moment and share my perspective on standardization. Simply stated, we're always striving to increase the effectiveness of our combat wings, and I think standardizatiion offers potential in this arena. Reduced cost of operation is certainly a factor; however, combat effectiveness is the first prize.

<u>There are many facets to standardization</u>. There are techical issues, policy issues, budgeting issues, programmatic issues, logistics support issues and, not to be forgotten, cultural blocks. They all come into play.

<u>Then of course there is the basis for standardization</u>. From a military sense, it's rooted most recently from our Vietnam experiences. There is no doubt that the lack of standardization complicated an already complex and stretched logistics pipeline and undoubtedly resulted in reduced sortie rates. Another factor that has become dominate in the standardization scenario, however, more subtle, is that we have an aging inventory of airplanes, an inventory where three-fourths of the force is over 9 years old. By contrast, 30 years ago, only 14% of our fleet was over 9 years old. Further, the aging of the fleet will continue and be compounded. By the early 1990's our force structure will be two-thirds its current size. We'll be keeping some aircraft as long as 40 years. Avionics that not long ago stayed for the life of the airplane now must be changed out every few years. Recent trends verify this and, in my opinion, point directly to an opportunity for those of us who are standardization advocates to excel.

Another factor that points to standardization is our people problem. If we allow systems to profilerate without standard patterns, it is doubtful that our training system can cope. Architectural standards, like those being discussed at this conference, and form, fit, and function standards permit us to build and sustain combat skills faster and more effectively.

Again, however, standardization should be applied when and where it makes sense. That's the reason for the term "rational standardization"--the theme of this conference. One test for a standard is that it be transparent to technology. We need to be able to introduce new technology when needed as soon as we can.

Standardization injects discipline into the acquisition and support process. It's the discipline that's important here. Manufacturers have historically depended upon this discipline in their operations. In the commercial arena, as well as in military markets, companies use their own standards. They expect their suppliers to meet these as well as industry standards in the production of their products. The production line is nothing more than the application of a standardization policy. The whole process is geared to save time, save money, and preclude individual handcrafted parts. Industry approaches standardization for at least two reasons. First, its sameness. Each product is interchangeable with another. Second, it's a guarantee that the product meets a certain property.

The airlines, too, offer an example of rational standardization. They devise standards for their own use. They voluntarily sign up to use them--because they make sense. They standardize as much as possible. About 50% of their equipment meet ARINC standards. The mission and traffic control equipment which includes radios, inertials, and weather radar is standardized in upwards of 90% of their fleet. The rest do not meet standards because each carrier has peculiar needs or finds no compelling economic reason to do so. This is rational standardization at work. In the military environment there is no comparable economic reason for standardization. The closest financial measure is oriented around the annual budget exercises. But, the true measure is combat capability and, as you might imagine, this is sometimes difficult to measure.

Standardization has to be a conscious effort on the part of all those concerned--the user, the acquisition and logistics support communities, plus the aerospace and avionics industry.

We understand that introducing standards is no easy task. The acquisition process is geared around optimal acquisition of individual weapon systems--not always optional acquisition of a force structure. Therefore, it places the aerospace companies in a peculiar position. Going a standard approach exposes the primes to potential loss of follow-on procurements and modification programs. Our policy is to maintain competition--but we recognize that industry performs when the profit incentive works, and the potential for profit is evident. We can ill afford to allow our aerospace industry and its subtier vendors to erode any more.

The requirement as I see it is quite clear. We need the standards like the ones you're working on here at this conference. We also need F^3 standards to facilitate transition of new technologies as they come along, and we shouldn't rest on those laurels. We have to work on future aircraft and the requirements of our current aircraft to operate against future threats. This may necessitate new standards.

The challenge is varied. If we seriously intend to support a combat ready and combat capable force, then we need your support. For those of you from the using commands, it's necessary for you to not only identify your needs but make it abundantly clear the role and need for standards in your operations. You cannot afford aircraft down for long periods because of mod work that otherwise could be shorter had the new equipment been standardized. Software is beginning to emerge as a prime issue in the mod business. A standard higher order language alleviates many problems--but maybe we can do more.

The programmatic area needs more emphasis. Unfortunately, the acquisition process optimizes on single weapon systems. Its emphasis on project management, with its focus on cost, schedule, and performance, tends to shift away from standards. The focus being narrower than the overall Air Force sometimes leads to non-optimum decisions.

For our industry participants, we need your support in making the use of standards a reality. Work with us and show us how to make standardization possible without jeopardizing your markets and ultimately the industrial base. As I've said, we need standardization for combat capability, but we won't get there without your help.

Finally, to all the participants, both government and industry, I personally appreciate your efforts. What you're doing in your day-to-day efforts is being felt and is making our force structure more productive.

I see this conference as a productive effort. Again, thanks for inviting me.

Wednesday Luncheon Keynote Speaker

Dr. Alan M. Lovelace

Effective 1 Sep 82, Dr. Lovelace was named VP, Productivity and Quality Assurance.

Dr. Lovelace joined General Dynamics Corporation as Vice President, Science and Engineering in July 1981. He had served as Acting Administrator of the National Aeronautics and Space Administration since January of 1981.

Dr. Lovelace joined NASA in 1974 as Associate Administrator for the Office of Aeronautics and Space Technology. He was named Deputy Administrator in June 1976 by President Ford.

Since entering federal service with the U.S. Air Force in 1954, he has held many research management positions. He served at the Air Force Materials Laboratory, Wright-Patterson Air Force Base, Ohio, from 1954 through 1972, having been named Director in 1967.

From 1972 to 1973, he served as Director of Science and Technology with the Air Force Systems Command, Andrews AFB, Washington, D.C. From 1973 to 1974, Dr. Lovelace was Principal Deputy Assistant Secretary of the Air Force for Research and Development.

Dr. Lovelace retired as Deputy Administrator of NASA in December 1980, but stayed with the Administration through the first flight of the Space Shuttle Columbia and the appointment of a new Administrator.

Born in St. Petersburg, Florida, in 1929, Dr. Lovelace received Bachelor's, Master's and Doctoral Degrees in Chemistry from the University of Florida. Awards he has received include the Presidential Citizens Medal, the Department of Defense Exceptional Service Medal, the Air Force Decoration for Exceptional Service, the National Civil Service League Career Service Award, and the Office of Aerospace Research Award for Outstanding Contriubitons to Research.

He is a Fellow of the American Institute of Aeronautics and Astronautics and the American Astronautical Society, and is a member of the National Academy of Engineering, Air Force Association, Sigma XI and Phi Beta Kappa.

LUNCHEON SPEECH

2ND AFSC STANDARDIZATION CONFERENCE

"A MAJOR TRI-SERVICE CONTRACTOR'S VIEWPOINT ON MILITARY STANDARDIZATION"

A. M. LOVELACE

1 December 1982

MY REMARKS TODAY WILL EXPLORE STANDARDIZATION FROM THE POINT OF VIEW OF A MAJOR DEFENSE CONTRACTOR, A CONTRACTOR THAT INTERFACES WITH ALL OF THE SERVICES AND WITH THE DEPARTMENT OF DEFENSE. I WILL BE ADDRESSING THIS SUBJECT BY (1) OFFERING SOME OBSERVATIONS ABOUT THE NATURE AND APPLICATION OF STANDARDS, (2) BY OUTLINING THE APPLICATION OF STANDARDS IN THREE EXAMPLES OF GENERAL DYNAMICS' PRODUCTS THAT WERE, OR ARE BEING, DEVELOPED BY DIFFERENT DIVISIONS FOR DIFFERENT DOD CUSTOMERS, AND (3) AFTER TAKING THIS LOOK AT STANDARDS IN PRACTICE, I WILL OFFER SOME GENERAL IDEAS ABOUT THE IMPACT OF STANDARDIZATION ON THE DOD CONTRACTOR COMMUNITY. FINALLY, I WILL CLOSE WITH SOME COMMENTS ON THE RELATIONSHIP BETWEEN STANDARDS AND TECHNOLOGY. MY REMARKS TODAY, OF COURSE, WILL BE DIRECTED AND LIMITED TO THOSE DIGITAL PROCESSING, COMPUTER LANGUAGE, AND INTERFACE STANDARDS OF PARTICULAR INTEREST TO THIS CONFERENCE.

I'VE DEALT WITH PROS AND CONS OF STANDARDS FOR SOME TIME NOW AND HAVE SEEN THE ISSUES FROM A VARIETY OF PERSPECTIVES. CURRENTLY, I AM WORKING TO IMPROVE PRODUCTIVITY AND QUALITY THROUGHOUT THE CORPORATION.

MANY PRODUCTIVITY AND QUALITY IMPROVEMENTS ARE, OF COURSE, TIED TO STANDARDIZATION. STANDARDIZATION IS A PREREQUISITE TO VOLUME PRODUCTION. LET ME CLARIFY, HOWEVER, THAT STANDARDIZATION AND COMMONALITY ARE NOT THE SAME THING.

STANDARDIZATION IS THE RESULT OF A TECHNICAL DECISION THAT PROVIDES A BASIS OR OPPORTUNITY FOR COMMONALITY. COMMONALITY, HOWEVER, IS THE RESULT OF A PROCUREMENT DECISION. IN OUR COMPANY, WE MAKE PROCUREMENT DECISIONS FAVORING COMMONALITY AND THEREBY REALIZE THE LOGICAL OUTGROWTH OF STANDARDIZATION. CONSEQUENTLY, FROM A PRODUCTIVITY AND QUALITY ASSURANCE STANDPOINT, STANDARDIZATION AS A BASIS FOR COMMONALITY IS UNBEATABLE. COMMONALITY ALLOWS VOLUME PRODUCTION WITH ASSOCIATED IMPROVEMENTS IN TOOLING AND AUTOMATED PROCESSES AND LETS US HONE QUALITY CONTROL MEASURES.

PREVIOUSLY, AS GENERAL DYNAMICS VP OF SCIENCE AND ENGINEERING, I SAW THE STANDARDIZATION ISSUE FROM A DIFFERENT PERSPECTIVE. IN OUR BUSINESS, GENERAL DYNAMICS IS ALWAYS FACED WITH TECHNOLOGY ISSUES VERSUS THE ADVANTAGES OF STAYING WITH A KNOWN AND PROVEN APPROACH. OUR ENGINEERS MUST BALANCE THESE ISSUES TO OFFER A PRODUCT THAT MEETS REQUIREMENTS AND IS PRODUCIBLE. I RECOGNIZED THEN THAT EVEN IF COMMONALITY DIDN'T OCCUR, OR PERHAPS WASN'T DESIRABLE, THE STANDARDS DID FREE OUR ENGINEERS FROM REPETITIVE, SIMILAR DESIGN TASKS AND FREED THEIR ENERGIES AND TALENTS TO PURSUE THE NEXT LEVEL OF TECHNICAL ISSUES. WHY DO MULTIPLE DESIGNS TO DO THE SAME FUNCTION?

I HAVE NOT ALWAYS VIEWED THE WORLD FROM GENERAL DYNAMICS CORPORATE HEADQUARTERS. MY EARLIER POSITIONS WITH NASA AND THE USAF SHOWED ME ANOTHER SIDE OF STANDARDS. IN PARTICULAR, I WAS SENSITIZED TO THE BUDGETARY TRADE-OFFS THAT ARE MADE BETWEEN SUPPORTING AND MAINTAINING FIELDED SYSTEMS VERSUS DEVELOPING NEW SYSTEMS. RECOGNITION OF THE COSTS AND LOGISTICS OF OPERATING EXISTING SYSTEMS WAS A PRIME DRIVER IN INITIATING NEW STANDARDS. AND THE RESULT OF THESE INITIATIVES BRINGS US TO THIS CONFERENCE, TO TAKE THE PULSE OF THE STANDARDIZATION EFFORT AND TO ASSESS AND, PERHAPS, INFLUENCE ITS FUTURE.

BASED ON THESE EXPERIENCES AND DIVERSE PERSPECTIVIES, TWO OBSERVATIONS WERE CONSISTENTLY APPARENT. FIRST, A SERVICE GETS THE DEGREE OF STANDARDIZATION THAT IT WANTS. THAT IS, THE PRODUCT IS A SORT OF MIRROR REFLECTING THE ATTITUDE OF THE PROCURING SERVICE TOWARD STANDARDIZATION. RECOGNIZE THAT THE DEFENSE INDUSTRY IS SOPHISTICATED IN ITS ABILITY TO ASSESS THE CUSTOMERS' REAL INTENTIONS WITH RESPECT TO THE CRITICAL PROCUREMENT DRIVERS.

IT UNDERSTANDS THE PROGRAM REQUIREMENTS.

IT DETERMINES CUSTOMER SENSITIVITY TO THE VARIOUS DRIVERS AND STANDARDS.

IT KNOWS THE FUNDING CONSTRAINTS.

OUR SOPHISTICATED DEFENSE INDUSTRY TAILORS THE PRODUCT TO BE THE ONE THE CUSTOMER WILL BUY. THIS PRODUCT INCORPORATES THOSE SOFTWARE, HARDWARE AND PROTOCOL STANDARDS THAT SELL THE PRODUCT.

SECOND, STANDARDS WILL BE APPLIED BY CONTRACTORS WHEN IT IS TO THEIR COMPETITIVE ADVANTAGE TO DO SO. WE WILL RESPOND TO THE "REAL EVALUATION CRITERIA" IN TERMS OF INITIAL PROCUREMENT COST VERSUS LIFE CYCLE COST.

I FOUND IT INSTRUCTIVE TO VIEW HOW STANDARDIZATION HAS BEEN IMPLEMENTED IN GENERAL DYNAMICS PRODUCTS. AS PREVIOUS SPEAKERS HAVE MENTIONED, THE F-16 REPRESENTS A PRODUCT WHICH ALTHOUGH TECHNICALLY CONCEIVED IN THE EARLY SEVENTIES, IS ACHIEVING FULL STANDARDIZATION THROUGH THE AGGRESSIVE ACTIVITIES OF BOTH GENERAL DYNAMICS AND THE AIR FORCE.

ON THE OTHER HAND, THE TOMAHAWK CRUISE MISSILE WAS INITIATED IN A LATER PERIOD ('76-'77) AND THESE SYSTEMS INCORPORATE VERY FEW OF THE COMPUTING AND INTERFACE STANDARDS. THIS IS BECAUSE A CONSCIOUS DECISION WAS MADE BY THE SERVICES AND DOD TO USE OFF-THE-SHELF EQUIPMENT TO REDUCE THE TIME TO FIELD THE SYSTEM. HOWEVER, LATER VERSIONS OF TOMAHAWK, FOR EXAMPLE MRASM, ARE INCORPORATING ALL THE STANDARDS THAT ARE ALSO BEING INCORPORATED IN OUR F-16'S AND AS WITH THE F-16 WE ARE ALSO STANDARDIZING IN OUR SUPPORT EQUIPMENT ARENA FOR OUR LATER MISSILES.

THE M-1 ABRAMS MAIN BATTLE TANK TO SOME MIGHT REPRESENT A STARK CONTRAST TO FIGHTER AIRCRAFT AND CRUISE MISSILES. ALTHOUGH THE M-1 WAS INITIATED IN THE EARLY 70'S AND WITHOUT BENEFIT OF THE MORE MATURE STANDARDS OF TODAY, THE TANK GROWS MORE SOPHISTICATED WITH EACH GENERATION.

IF I WAS TO PLACE YOU IN THE CREW COMPARTMENT OF THE TECHNOLOGY DEMONSTRATOR BEING BUILT TODAY YOU WOULD BE HARD PRESSED TO

DISTINGUISH IT FROM THE COCKPIT OF A MODERN AIRCRAFT. IT IS FORTUNATE THAT AS WE SEE THE SAME TECHNOLOGY SOPHISTICATION EMERGING IN THIS CLASS OF SYSTEMS THAT THE STANDARDS TO SUPPORT THEIR INCLUSION ARE COMING ON LINE. I AM SURE THAT IN ANY MAJOR UPDATE OF THIS CLASS OF VEHICLE, STANDARDS WILL BE CLOSELY EXAMINED BY GENERAL DYNAMICS AND THE ARMY.

RECOGNIZE THAT GENERAL DYNAMICS IS VERY ANXIOUS TO INCORPORATE THE SAME STANDARDS IN FUTURE HEAVY FIGHTING VEHICLES AS WE ARE APPLYING IN THE F-16 AND IN THE NEWER FAMILY OF CURISE MISSILES. WHILE THIS HAS OBVIOUS BENEFITS TO EACH SERVICE, THE DOD AND THE AMERICAN TAXPAYER, IT IS ALSO DIRECTLY BENEFICIAL TO THE CORPORATION. THE SYNERGISM OF COMMON STANDARDS AMONG MULTIPLE PRODUCTS PUTS US IN A VERY STRONG COMPETITIVE POSITION. IT IS EXACTLY THE KIND OF BUSINESS INCENTIVE THE SUPPORTORS OF THIS CONFERENCE ARE STRIVING FOR! AND I SEE MUCH GREATER POSSIBILITIES IN THE FUTURE. VLSI AND VHSIC WITH THEIR SMALL SIZE, LOW POWER AND COOLING REQUIREMENTS WILL ENABLE THIS SYNERGISM TO BE EXTENDED TO OUR SMALLER SIZED TACTICAL MISSILES.

THE IMPACT OF VLSI AND VHSIC COMBINED IN THE STANDARDIZATION INITIATIVES WILL GO BEYOND THIS SYNERGISM. AS I SEE IT, VLSI AND VHSIC WILL ALLOW STANDARDIZATION TO REACH THE HIGH PAYOFF AREA - NAMELY, COMMONALITY. SMALL SIZE, LOW POWER AND COOLING CONSTITUTE A "LEASE COMMON DENOMINATOR" THAT WILL ALLOW THE SAME FUNCTIONS IN DIFFERENT PRODUCTS TO BE ACCOMPLISHED BY COMMON HAR[illegible]

FRANKLY, BECAUSE OF THIS, I FORESEE THAT GENERAL DYNAMICS' COMPETITIVE POSITION WILL CONTINUE TO IMPROVE AS EACH PRODUCT SUPPORTS THE OTHER. AS I MENTIONED, THE DoD AND EACH SERVICE WILL ALL CONTINUE TO BENEFIT.

LET ME COMMENT, HOWEVER, THAT THIS MAY NOT BE TRUE IN ALL FACETS OF THE INDUSTRY. MAJOR SYSTEM HOUSES WILL LIKELY BE IN SIMILAR POSITIONS AS GENERAL DYNAMICS. SUBSYSTEM HOUSES MAY VIEW THIS AS A NEGATIVE TREND. THEY PROBABLY VIEW STANDARDS AS REDUCING THEIR UNIQUENESS AND PROPRIETARY POSITIONS AND, OF COURSE, THE APPLICATION OF COMMON HARDWARE COULD CERTAINLY BE VIEWED BY THEM AS LIMITING OPPORTUNITIES. IN THIS REGARD, I WOULD COMMENT THAT TECHNOLOGY COUPLED WITH STANDARDIZATION WILL (OR AT LEAST SHOULD) HAVE A PROFOUND AFFECT ON THE DEFENSE INDUSTRY. COMPANIES THAT STAY WITH CLASSICAL PACKAGING AND PRODUCT DEFINITIONS WILL LIKELY LOSE MARKET SHARE. THERE WILL BE SOME VERY SUCCESSFUL UPSTARTS! THAT'S THE WAY IT ALWAYS IS WHEN NEW TECHNOLOGY SUCH AS VLSI AND VHSIC START TO BE IMPLEMENTED. COUPLED WITH STANDARDIZATION THIS TECHNOLOGY OFFERS NEW BUSINESS APPROACHES AND OPPORTUNITIES — FOR BOTH SUCCESS AND FAILURES.

WHEN I FIRST STARTED TO PREPARE THIS TALK I PLANNED TO MAKE A PROFOUND AND FEARLESS ASSESSMENT OF THE DoD AND TRISERVICE POSTURE ON STANDARDIZATION. HOWEVER, IT'S NOT FEARLESS AND NOT PROFOUND. MERELY IT'S TO SAY THAT THEIR POSITION AND INDUSTRY'S IS THE SAME - THE BEST PRODUCT AT THE LOWEST COST.

LET ME NOW TAKE JUST A MOMENT TO LOOK AHEAD AND TO MAKE SOME RECOMMENDATIONS. AS YOU WILL RECALL, I MENTIONED EARLIER THAT STANDARDIZATION IN HIGH TECHNOLOGY AREAS GENERALLY COMES UNDER HOT DEBATE AND CLOSE SCRUTINY. STANDARDIZATION SCARES MOST TECHNOLOGISTS SINCE STANDARDIZATION, TO MANY PEOPLE, IMPLIES BEING STAGNANT, WHICH LEADS TO OBSOLENCE, WHICH THE MILITARY CANNOT STAND. AN EXAMPLE OF THIS IS DoD 5000.5X. WHETHER CORRECT OR NOT, THE OPPONENTS OF 5000.5X USED THE POSSIBILITY OF TECHNICAL OBSOLESCENCE AS THE BASIS FOR THEIR POSITION. THIS INEVITABLE DEBATE LEADS TO MY FIRST RECOMMENDED TEST FOR ANY NEW STANDARD. WILL THE PROPOSED STANDARD INHIBIT TECHNOLOGY? TO PASS THIS TEST, STANDARDS SHOULD BE TECHNOLOGY TRANSPARENT. THIS IMPLIES THAT WE SHOULD NOT STANDARDIZE ON PROCESSES OR HARDWARE, BUT SHOULD STANDARDIZE ON INTERFACES AND FUNCTIONS.

ANOTHER VALUE TEST FOR A STANDARD IS THAT STANDARDS SHOULD HAVE A REASONABLY LONG LIFETIME. OTHERWISE, THE TIME INVOLVED IN GOVERNMENT PROCUREMENT, CONTRACTOR IMPLEMENTATION AND SERVICE APPLICATION WILL EFFECTIVELY EITHER VOID THE STANDARD OR INHIBIT ITS EFFECTIVENESS. THE LONGER THE LIFETIME, THE GREATER THE OPPORTUNITY THAT THE STANDARD WILL RESULT IN COMMONALITY OF EQUIPMENT, WHICH IS THE BIG PAYOFF. THIS IS MY SECOND RECOMMENDED TEST. WILL IT HAVE A REASONABLE LIFETIME?

THIS PAGE LEFT BLANK INTENTIONALLY

Along with this test, we must also recognize that, while standards' lifetimes may be reasonable, they certainly won't be forever. Therefore, standardization policies should include procedures for regular review and phase-out. When new standards supersede old standards, the new standards should be downward compatible. My third recommended test is that new standards should be downward compatible with the standard or practice that it is replacing.

Similar to the time dimension are the differences between programs. Programs are each driven by unique schedules, mission requirements, system capabilities, production costs and life cycle costs. Standards must be transparent to these differences otherwise the application will be very limited. This is my fourth recommended test: Does the standard transend program uniqueness?

I mentioned earlier that standards benefit both contractors and DoD. Let me know rephase that into a fifth recommended test: Will the standard be beneficial to both government and industry?

And my sixth, last, and perhaps most important test is a business issue. Does a viable business plan exist to support the standard? Have funds been authorized to implement the standard? Has the military made it a requirement, and will the procuring activity make it a criterion in source selection? Standards development is a technical issue, but implementation is a business issue.

IN CLOSING I WOULD LIKE TO OBSERVE THAT THE SIZE OF THIS AUDIENCE TODAY AND THE OVERALL SCOPE AND SUCCESS OF THIS CONFERENCE IS A MEASURE OF THE INROADS THAT STANDARDIZATION IS MAKING IN THE DEFENSE BUSINESS. TO DATE THESE INROADS HAVE BEEN AT A PROGRAM AND PROJECT LEVEL. IN GENERAL DYNAMICS WE ARE NOW ELEVATING THIS ACTIVITY TO A CORPORATE LEVEL TO THEREBY ACHIEVE THE FULL SYNERGISTIC BENEFITS. IN A NUTSHELL, STANDARDIZATION IS JUST GOOD BUSINESS AT GENERAL DYNAMICS.

THANK YOU.

Thursday Luncheon Keynote Speaker

Charles P. Lecht

Mr. Lecht is President of Lecht Sciences, Inc., a research and think-tank recently established in New York City.

Mr. Lecht is founder and former President/Chairman of the Board of Advanced Computer Techniques Corporation (ACT), a computer softwaree consulting firm.

He holds a B.S. Degree in Mathematics from Seattle University and a M.S. Degree, also in Mathematics, from Purdue. His involvement in the computer field stretches back to 1951, making him an "old-timer" in a very young industry.

Among his earliest professional activities were programming for IBM's Service Bureau and for the MIT community's Lincoln Laboratory/MITRE organizations on a variety of scientific and military simulation projects.

From 1960 to 1962, Mr. Lecht served in the U.S. Army Ordnance Corps, first as Chief of its Programming Division and subsequently of its Mobilization Application Division; Ordnance Industrial Data Agency.

Mr. Lecht came to New York City in 1962, where he founded ACT. In the 17 intervening years, the Company has grown from a one-man show to an international complex employing over 450 persons and deriving more than 50% of its revenues from operations in Europe, Canada and the Middle East as well as the U.S.

In addition to building and presiding over ACT, Mr. Lecht has found time to hold a number of technical posts, author five books and innumerable articles and maintain a heavy schedule of speaking engagements in the U.S. and abroad. In addition to THE WAVES OF CHANGE, his books include three on computer languages and one on project management.

He is a member of the Young Presidents Organization, The Hudson Institute, the Data Processing Management Association, the Association for Computing Machinery and the New York Academy of Sciences.

In 1976, Mr. Lecht was designated by "The Gallagher Presidents' Report" as one of the "10 Best Businessmen in the USA" representing companies with income below $1 billion. Profiles of Mr. Lecht have appeared in the New Yorker and Datamation, among other publications.

If you thought change in computer systems technology was accelerating in 1982, you haven't seen anything yet. By this time next year, January 1984, 1982 will be remembered as a period of relative calm in comparison to what we are to witness in 1983.

So intense will such change be that we will be compelled to alter our most fundamental views of the nature of computer systems and their roles in our lives. New and improved methods of acquiring the powers engendered by computer technology are emerging swiftly and these powers, reapplied to this same technology, are fueling its capacity for change in ever more efficient ways. However unsettling this may be to computer systems users, the net effect of 1983's change should be dramatically positive.

For all the ferocity of its drama, we still have the delicious luxury of exploring this phenomenon and speculating on its meaning in a relatively tranquil state of mind.

While it has been true all along, we've only recently accepted the fact that computer systems impart to their users extraordinary powers in memory, logic and computation unavailable in our natural world. And sewn together like beads into strands and strands into lattices, computer systems have envolved to become at once their own repository of such powers, a part of yet another's, and a portal to still others of lesser or greater consequence. This came about as a result of the synthesis of computer and communications systems technologies in the 1970's. It is as though the once clear ideas that made a computer system so readily distinguishable from a communication system have been swept away in a tidal wave of change. All the substantial attributes of either are found in the other: miles of circuitry, switches, terminals, signal processors, buffers, memories, conversion devices, software, even wave transmitters/receivers. This has caused us to redefine our concept of a computer system to mean a network of devices whose communications veins and arteries may vanish into the microscopic world and extend into the macroscopic world. Since all modern systems are collections of processors, some of which may be located remotely, ad-hoc, run-time system configurations are possible. We send data between devices to be processed and receive the results through massive lattices of wire whose organizational complexities are mind- boggling and whose cross-sectional dimensions vary from micron to meter. This has led, not unreasonably, to a blurring of our perceptions of a specific computer system and its limits, especially with regard to conventional concepts of temporal and spatial dependencies.

As if this were not enough to handle, contemporary systems linking widely dispersed locations are involving an increasing broadcast component. Thus our very concept of systems, hard enough to "domesticate" on terra firma, now extends into a new, heretofore alien, dimension, complete with its own set of barely compassable peculiarities. With its use no longer restricted to mere shipment of data, broadcast is becoming practical as storage medium for massive files humming soundlessly somewhere between the moon and New York City.

Under development in the past few years have been massive computer systems referred to in Bell Labs literature as Integrated Services Digital Networks. The appearance of these Titans underscores our commitment to create ever larger repositories of computer systems power, and to distribute this power in ever more effective ways. I visualize such systems in the form of massive ships suspended in artificial intelligence space. Possessed of their own intelligence and augmented by exogeneous intelligences docked at their ports, they, and the promise of that ISDN technology from which they spring, present a spectacle (and capability) of breathtaking beauty and scale.

As such systems are created in the macroscopic world, so identical systems are aborning in the microscopic world of chips, wherein magnetic field fluctuations invoke device operations and carry messages. In this world, too, are satellities, transmitters, and receivers. The satellite is suspended in a space whose "ether" is silicone, sapphire, metal oxide and exotic, other-world compositions. However flat the world of the silicon chip may seem to the naked eye, some of its elements are separated by distances proportional, within their physical frame of reference, to those separating us from our own space satellites. Peering through a microscope, we are astounded to see a network of cables and devices more complex than their nearest counterparts in, say, some vast, sprawling, tentacular chemical plant. Not only are we surprised that a third dimension appears but that the distances traversed laterally on some chips, however confining their edge sizes may seem to us to be, rival those in whose terms we conduct our lives in the macroscopic world.

The rate of announcement of technological breakthroughs by the world scientific community and the meaningfulness of each such announcement has been nothing short of astounding in the past few years. It has been responsible for the current widespread proliferation of personal computers. But more important still, communications improvements have virtually eliminated the practioner's need to visit a computer, personal or otherwise, to share in its powers; access to them is obtainable virtually everywhere. The once discrete concepts of personal computer and terminal are in the process of synthesizing. This phenomenon, along with communications improvements, provide us with fantastic processing potential requiring very little front-end investment. As every such device fulfills the role of being a functional locus in a strand of companion devices, unified by LAN technology, its usage provides us with the capacity to ignite the strand, energize the lattice of which it is a part, and, if only for an instant, invoke the power of a congress of CRAY's, CYBER's, 3084's and others. The output of this event could vary from the manipulation of a gigabyte database, to forecasting weather, to a few micro-coded real-time computer instructions fed like pablum to a baby micro guidance processor embedded in the belly of a bomb. Electronic mail dispatched by Jupiter and carried by Mercury himself could do no more.

So what does this all have to do with standards? A lot. Choosing standards which serve the intentions of governmental directives in today's swiftly changing technological climate is a more demanding task than it has ever been. For these directives were cast at a time when our technology was very different than I've just described.

There are so many hardware and software standards with which our government must be concerned. And the concern is one of legitimacy, nay, duty. One cannot help but empathize with those who must ultimately make the choice; the probability of error is increasing.

While contemplated product life cycle durations of computer systems haven't changed too much, the rates at which compelling alternatives arrive have. This causes perfect reasonable life cycle plans to suffer premature obsolescence with ever-shortening regularity. These alternatives are new systems with such price performance improvements that conversion brcomes worthwhile; if conversion is required at all. Today's systems managers trained, say, in the 1960's and 1970's and shell shocked from the trauma of having lived through but a few conversions, are appalled at the thought of doing so every few years. Yet, with burgeoning requirements and limited budgets, those in position to take advantage of the new alternatives are compelled to do so. Thus, for example, those users who acquired Honeywell DPS/5 systems only a few yrars ago may find it cost effectve to enter the conversion process needed to upgrade to a DPS-8/70 even though the DPS/5 was forcast to last, say, 5 to 10 years. This may even be the case despite the fact that the users work load is not burgeoning, so improved in price performance is the new system. The phenomenom of upgrading while down-costing is not new in our industry, but its increasing appearance signals radical change. It further reveals that we are inexorably drifting toward an ever faster stream of product announcements. Remaining operational at the terminus of this stream will involve hooking into a massive ISDN and hanging on for life.

No one has achieved the capability to upgrade and down-cost better than IBM without much conversion at all. Especially if the user is willing to buy more. The stream of this kind of IBM hardware flows ever more swiftly to the marketplace as IBM wisely readies its own massive ISDN.

Fancy lease to purchase plans; purchase, rental, even trade-in and other financial plans abound. Thus, if not propelled by technological change, economic incentives alone may trigger a systems change.

"When the stream is flowing swiftly, tubing becomes proportionately hazardous. The wise tuber avoids white water and sticks to the center. Not that thrills aren't to be had in the diversionary whirlpools of peripheral tributaries. It's just safer." This advice was given me by a wizened old tuber on Arizona's Salt River. "It's hardly worth it to express your opinion," he said. "The river expresses it for you."

Back to choosing standards in today's technological climate. As the tuber must learn to respect the river's flow, our standards choices are being affected by equally powerful overriding forces.

So very serious persons like those attending the USAF standardization conference in Dayton this week meet to discuss their plans to cope with these forces. The Dayton meeting addresses DOD's needs.

I assume Dayton was chosen to underscore the somber nature of DOD's problems; Miami or New Orleans appearing to jovial. Bent upon reviewing, amending, proposing and adopting standards which meet the various intentions of DOD directives; this group faces an awesome task.

INTERLUDE

In my mind's eye I conjur the mischievious image of a secret cult meeting in the most laid back place possible; the Grey Temple of Dayton. The high priests of JOVIAL (Jule's Own Version of the International Algorithmic Language) venerate ALGOL while linguists, metaphysicians, and MENSA members sing cantos to complexity. And ALGOL begat JOVIAL which begat JOVIAL II which begat JOVIAL 3 which begat JOVIAL 3B and JOVIAL 73. JOVIAL 3B begat JOVIAL 3B2 which with JOVIAL 73's decendent, Jovial 73/I, begat JOVIAL 73(not to be confused with JOVIAL 73/I's progenator JOVIAL 73.) ADA novitiates swarm about. Before ordintion they must convert at least one COBOL-caholic to SHORTHAND, and obscure dialect known only to K. Gibbsian scribblers. The holy ark is opened and when ALGOL speaks, everyone listens. He makes his ten statements and declarations as the cult mumbles I/O protocols and code for STRECH and LARK. The ceremony ends as an ADA novitiate is confirmed.

END OF INTERLUDE

Back to standards. Reviewing yesterday's choices made at a time when things were moving far more slowly, we can conclude that todays usefulness of these is questionable. The FIP I/O standard is a case in point. Discarded by most manufacturers in planning new technology, its utility seems limited to gaining approval to bid on government contracts and sustaining the PC industry. And what about JOVIAL?

When DOD Directive 5000.31 specified JOVIAL as the cross-compiler for USAF embedded weapons applications, who could have faulted it's intentions; to minimize assembly language programming, minimize maintenance, and discourage proliferation of new higher order languages.

Did JOVIAL do this? It doesn't seem so to me. And if my perception is correct, what should we learn from it?

No one could argue with the underlying reasons for wanting standards like JOVIAL, or what JOVIAL was supposed to be. And not all JOVIAL events missed the mark; J3B's role in the F-16 and B-1 programs has been significant.

When JOVIAL was chosen, no one, in or out of the government seemed to have a clue that swift and enduring change in computer technology was to take place. Crippled at the onset as a language wanting of definiton, its subsequent history tells a tale of heroic struggle, by implementers and users alike, to conform to the USAF decision. Evolving to be an encyclopedic language, intended to be all things to all people, its complexity masks its usefulness. So does its size. Except in erudite examples in some scientific journals, its usage is a dead giveaway to the Russians that an embedded weapons system is involved; nobody else uses it.

And now ADA! I predict it will suffer the same fate. Overly complex, over sized, it's another software development dream that will course the river of accelerated innovation out of its mainstream.

Prudence would have sugested FORTRAN's immediate descendants, or PASCAL-like dialects like MODULA II. Everyone with an Apple, IBM, etc. personal computer, who isn't solely playing games with it, knows something like it. Scientists programming the big stuff were brought up on it. And while usage of a dialect like MODULA II might require a bit more assembly language programming than we'd like, savings in other aspects of the intention of DOD Directive 5000.31 would more than compensate for this extravagance. With versions already available on just about every micro-to-macro processor, the well of available talent would be virtually unending. Chips running MODULA II could be manufactured and supplied to virtually every computer systems company for inclusion in their offerings, all using the same language repertoire and syntax. Easy to implement, this standard could be promulgated with euphoric ease. The net effect on productivity in the short term would be stupendous; no new higher order languages would be promulgated, maintenance would be a breeze.

It's quite daring to question the utility of JOVIAL, not to speak of ADA, these days. So much money and time has gone into their creation and usuage that in doing so, I fear retaliation from America's DOD; planes, bombs and all. And I don't pretend to know all that went into JOVIAL's and ADA's selection decisions. Nor the politics of these decisions.

What I do know is that our technology has undergone dramatic change since AF Directive 5000.31 and our standards selection methodology hasn't. Nor does it appear that 5000.31 intentions have been reviewed in light of this change.

With fifth generation processor capability providing scientists with picosecond (range) computational speeds and gigabyte memories we've entered a new dimension where the languages we use are less important than we think. Truly mobile software becomes possible when differences in operating speeds occur in the picosecond world. When redundant multiprocessors are commonplace, maintenance is conducted remotely, replacement is cheaper than repair, and back-up is always available through network communications, a score of earlier premises for standardization dissappear.

We need the most modern technology possible in a world where its manufacture is widespread. In order to get it, we'll have to abandon preconceived notions of computer systems software and hardware and get on with it.

EXECUTIVE SESSION

SESSION CHAIRMAN : Jeffery L. Peeler
ASD/ENAS

MODERATOR : Col. R. P. Lavoie
ASD/ENA

OPENING REMARKS
TO THE
2ND AFSC STANDARDIZATION CONFERENCE
DAYTON, OHIO

LIEUTENANT GENERAL THOMAS H. MCMULLEN
COMMANDER, AERONAUTICAL SYSTEMS DIVISION
WRIGHT-PATTERSON AFB, OHIO
30 NOVEMBER 1982

ON BEHALF OF THIS CONFERENCE SPONSOR, THE AIR FORCE SYSTEMS COMMAND, AND ALL OF US IN THE AERONAUTICAL SYSTEMS DIVISION WHO ARE YOUR HOSTS, IT GIVES ME GREAT PLEASURE TO WELCOME YOU TO THE 2ND AFSC STANDARDIZATION CONFERENCE.

FOR SEVERAL YEARS NOW, WE'VE BEEN WORKING WITH OUR SISTER SERVICES, OUR NATO ALLIES, AND INDUSTRY TO REALIZE THE BENEFITS OF STANDARDIZATION. I'M HAPPY TO SEE THAT KEY PEOPLE FROM EACH OF THESE PARTNERS IN STANDARDIZATION ARE HERE WITH US TODAY.

WE'VE TRIED VERY HARD TO TARGET THIS CONFERENCE TO YOU, THE EXECUTIVES, THE MANAGERS, THE SYSTEMS INTEGRATORS AND THE ENGINEERS, BOTH IN GOVERNMENT AND INDUSTRY. WE WILL PRESENT A LITTLE BIT OF EVERYTHING: FROM EXECUTIVE OVERVIEWS TO DETAILED VISIBILITY INTO OUR NEW STANDARDIZATION INITIATIVES.

THIS MORNING WE WILL PROVIDE SOME INSIGHT INTO THE CURRENT DOD VIEWS ON STANDARDIZATION; AND WE WILL ADDRESS STANDARDIZATION FROM THE PERSPECTIVES OF ALL THREE SERVICES' AS WELL AS A NEW PERSPECTIVE FROM NATO. FOR THE NEXT TWO DAYS, THE CONFERENCE WILL CONCENTRATE ON THE HARDWARE AND SOFTWARE AVAILABLE TO SUPPORT STANDARDIZATION INITIATIVES WE HAVE ALREADY IMPLEMENTED. THIS GIVES US THE OPPORTUNITY TO TELL YOU ABOUT THE BENEFITS WE EXPECT FROM THESE INITIATIVES, THE KEY EFFORTS SUPPORTING THEM AND THE LESSONS LEARNED FROM ACTUAL PROGRAM APPLICATIONS.

THIS SYMPOSIUM AFFORDS YOU THE OPPORTUNITY TO LEARN ABOUT OUR CURRENT STANDARDS, TO SEE WHAT IS BEING DONE TO SUPPORT THEM, AND TO VISIT THE EXHIBITS TO SEE SOME ACTUAL HARDWARE RESULTING FROM THESE STANDARDS. IN JUST THREE DAYS, YOU CAN GATHER A LOT OF INFORMATION HERE WHICH MIGHT OTHERWISE TAKE A LOT OF TIME AND EFFORT TO COLLECT. I HOPE YOU'LL TAKE ADVANTAGE OF IT--ESPECIALLY THOSE OF YOU FROM WRIGHT-PATTERSON. ENCOURAGE YOUR BOSSES TO COME TO THE EXHIBITS ON WEDNESDAY NIGHT. I'M SURE THAT EACH ONE OF THE EXHIBITORS WOULD WELCOME YOUR INTEREST AND YOUR QUESTIONS.

NOW, TO THE THEME OF THIS YEAR'S CONFERENCE -- "RATIONAL STANDARDIZATION." WE ALL KNOW THAT EVERY STANDARDIZATION ACTIVITY IS AN EXERCISE IN COMPROMISE: TO GAIN SOME BENEFITS WE ARE WILLING TO SACRIFICE OTHERS. THE TRADE-OFFS WE HAVE TO MAKE ARE NOT ALWAYS INTUITIVELY OBVIOUS--NOR ARE THEY ALWAYS CREDIBLY QUANTIFIABLE. OUR ULTIMATE OBJECTIVE IS INCREASED EFFECTIVENESS IN BOTH COMBAT AND PEACETIME TRAINING--AND GREATER AVAILABILITY TO EACH WEAPON SYSTEM IN OUR FORCE STRUCTURE. WE INTEND THE TERM "RATIONAL" TO BE USED SYNONYMOUSLY WITH THE PHRASE "COMMON SENSE." WE RECOGNIZE THAT OUR RECOMMENDATIONS OR DECISIONS IN THIS AREA HAVE TO BE BASED ON ANALYSIS--SO THAT IS WHAT I'D LIKE TO DISCUSS WITH YOU TODAY-- "THE ELEMENTS OF THE THOUGHT PROCESS THAT SHOULD GUIDE OUR STANDARDIZATION DECISIONS."

OUR FIRST ELEMENT IS "RETURN ON INVESTMENT." WE MUST BE ABLE TO IDENTIFY WHAT THE REAL BENEFITS ARE. WE LOOK FOR FINANCIAL, TECHNICAL AND OPERATIONAL PAY-OFFS--AND, IF WE DO IT RIGHT, WE CAN GET A COMBINATION OF ALL THREE. BUT THE "PAY-OFF" EQUATION HAS TWO SIDES TO IT--BENEFIT AND COST. IT IS SOMETIMES EASY TO CONFUSE THE TWO--AND ONE MAN'S BENEFIT CAN EVEN BE ANOTHER'S COST. WE ARGUE THESE ISSUES OUT AND TRY OUR BEST TO QUANTIFY ALL ASPECTS. THAT ISN'T AS EASY AS IT SEEMS WHEN YOU MUST "ASSUME" WHAT THE MARKET FOR THE STANDARD IS, AND "PROJECT" WHAT THE LONGER TERM BENEFITS MIGHT BE. THE JOB IS A LOT EASIER WHEN YOU KNOW WITH SOME DEGREE OF CERTAINTY HOW MANY SYSTEMS WILL BE INVOLVED.

THE SECOND ELEMENT IS VERY CLOSELY RELATED TO PAY-OFF--IN FACT, WE REALLY DON'T KNOW HOW TO SEPARATE THEM. THE SECOND ELEMENT IS, "THE APPROPRIATE LEVEL OF STANDARDIZATION." OUR CHOICES USUALLY RANGE FROM STANDARDIZATION OF PIECE PARTS, OR EQUIPMENT, TO STANDARDIZATION OF EQUIPMENT INTERFACES. THE KEY ISSUES HERE INVOLVE LOGISTICS COSTS, REALIZABLE TECHNICAL PERFORMANCE AND, OF COURSE, THE VERY REAL THREAT TO PROGRESS ACCOMPANYING A FREEZE OF TECHNOLOGY INFUSION. IN THE RECENT PAST, WE BELIEVED THAT USING STANDARD PIECE PARTS WOULD BENEFIT US A LOT, BUT THE ECONOMIC HALF LIFE OF DIGITAL MICROELECTRONIC COMPONENTS IS SO SHORT THAT WE'VE HAD TO RESTRUCTURE OUR THINKING IN THIS AREA. WE ALSO HAVE TO BE VERY CAREFUL WHEN WE PICK AN EQUIPMENT OR "BLACK BOX" STANDARD WHERE THE TECHNOLOGY IS MOVING VERY RAPIDLY. OUR BEST BETS FOR HARDWARE LEVEL STANDARDS SEEM TO BE WHERE NEITHER OUR REQUIREMENTS <u>NOR</u> THE TECHNOLOGY ARE CHANGING TOO RAPIDLY

ONCE WE HAVE SOME AGREEMENT ON THE FIRST TWO ELEMENTS OF STANDARDIZATION, WE FACE A DECISION ON THE THIRD--THAT IS-- THE "FORUM WE USE TO DEVELOP AND MATURE THE STANDARD." WE KNOW THAT BROAD ACCEPTANCE OF A STANDARD REQUIRES MAXIMUM PARTICIPATION OF ALL THOSE AGENCIES THAT WILL BE AFFECTED BY IT. WE TRY VERY HARD TO ESTABLISH BOTH THE ENVIRONMENT AND THE OPPORTUNITY FOR ALL INTERESTED PARTIES TO PARTICIPATE IN THE TECHNICAL DEVELOPMENT PROCESS. HERE TOO, WE ARE FACED WITH COMPROMISE AND MUST CONSIDER SACRIFICING TECHNICAL PERFORMANCE OR PERFECTION FOR ENGINEERING PRACTICALITY. WE TRY TO USE OPEN FORUMS AND ENCOURAGE INDUSTRY

TO JOIN "USER GROUPS" FOR EVERY MAJOR STANDARD WE CONSIDER. SO FAR, IT SEEMS TO BE WORKING AND IT'S THE BEST APPROACH WE'VE BEEN ABLE TO COME UP WITH.

AFTER WE'VE IDENTIFIED THE PAY-OFFS, SETTLED ON THE APPROPRIATE LEVEL OF STANDARDIZATION, AND OBTAINED AS BROAD A CONSENSUS AS POSSIBLE ON THE TECHNICAL CONTENT OF THE STANDARD -- WHAT'S LEFT? WELL, A NUMBER OF KEY QUESTIONS ASSOCIATED WITH "RATIONAL STANDARDIZATION" STILL HAVE TO BE ADDRESSED, LIKE:

--HOW ARE WE GOING TO IDENTIFY WHAT WE MUST DO TO MAKE IT A MATURE STANDARD? OR,

--HOW ARE WE GOING TO BUY IT SUCH THAT IT WILL ACHIEVE ALL OF OUR FORECASTED BENEFITS?

THIRD - HOW ARE WE GOING TO SUPPORT IT, ONCE WE'VE BOUGHT IT?

AND FINALLY -- HOW ARE WE GOING TO INSURE THAT THE "STANDARD" STAYS VIABLE OVER TIME?

THE FIRST QUESTION - HOW TO TAKE A STANDARD FROM A PAPER CONCEPT TO A STANDARD THAT IS VIABLE, WELL-UNDERSTOOD, AND EASILY IMPLEMENTED IS NOT ONE WE CAN ANSWER OFF THE TOP OF THE HEAD. WE HAVE TO ANALYZE IT, TRY IT, FIX IT, DEVELOP SOME APPLICATION GUIDES BASED ON THIS EXPERIENCE, AS WELL AS SOME FORM OF COMPLIANCE OR VERTIFICATION TESTING -- ALL OF THIS IN TIME TO MEET THE FIRST APPLICATIONS FORECAST FOR THE STANDARD. UNFORTUNATELY, FUNDING TO DEVELOP A STANDARD IS DIFFICULT TO GET; BUT THE MONEY TO DEVELOP THE "THINGS NEEDED TO INSURE ITS MATURATION AND EVENTUAL SUCCESS"

IS ALMOST IMPOSSIBLE TO GET. NOT EVERYONE IN THE BUDGET PROCESS UNDERSTANDS WHAT IT TAKES TO PRODUCE A "MATURE STANDARD."

QUESTIONS TWO AND THREE ARE SO CLOSELY RELATED THAT I HESITATE TO SEPARATE THEM LET ME TRY. WE MUST SETTLE THE QUESTION OF HOW WE INTENDED TO SUPPORT THE STANDARD VERY EARLY IN ITS CONCEPTUAL PHASE AND TWEAK THAT DECISION AS WE GO ALONG. THE SUPPORT REQUIRED WILL SURELY CONSTRAIN SOME OF THE TECHNICAL PERFORMANCE EXPECTED OF THE STANDARD--AND SHOULD BE A FACTOR IN THE BUSINESS STRATEGY DECISION OF "HOW WE BUY IT." FOR EXAMPLE, WE HAVEN'T ALWAYS RECOGNIZED HOW SENSITIVE AND CLOSELY RELATED THE SPECIFICS OF THE HOW-TO-BUY FORM, FIT, FUNCTION (OR F^3) [illegible] ARE TO THE LOGISTICS SUPPORT CONCEPT WHICH ENSUES. SO THAT IS THE ESSENCE OF QUESTION TWO; NAMELY, "HOW DO WE BUY THE STANDARD IN A MANNER THAT PRESERVES THE FORECASTED BENEFITS?" MANY OF THE COST BENEFITS WE FORECAST FROM A DECISION TO BUY A STANDARD CAN BE COMPLETELY WIPED OUT BY A DUMB PROCUREMENT STRATEGY. WE HAVE TO BALANCE COMPETITION, ECONOMIC ORDER QUANTITIES, AND AN OFTEN UNCERTAIN TOTAL MARKET. WE MUST COME UP WITH A CONTRACTING APPROACH THAT YIELDS COST BENEFITS TO US AND REASONABLE PROFITS TO INDUSTRY. A POOR DECISION HERE CAN HAVE A LONG TERM IMPACT ON ELEMENTS OF OUR FORCE EFFECTIVENESS AS WELL AS OUR INDUSTRIAL BASE.

GIVEN THAT WE MANAGE TO PROPERLY ANSWER THE FIRST THREE QUESTIONS AND THEREBY ACHIEVE AN EFFECTIVE, VIABLE STANDARD, WE STILL HAVE TO

ADDRESS THE FOURTH QUESTION: HOW WE KEEP A STANDARD VIABLE OVER TIME. A NUMBER OF THINGS ARE FIXED AT A POINT IN TIME WHEN A STANDARD IS ADOPTED BUT PASSAGE OF TIME BRINGS CONTINUING CHANGES. THE THREAT GROWS, OUR REQUIREMENTS CHANGE, THE TECHNOLOGY EVOLVES -- HOW DO WE DEAL WITH THIS IN THE "FIXED STANDARDS" ARENA? WELL--HERE AGAIN, WE MUST COMPROMISE. WE NEED TO BALANCE THE FIXED SOLUTION A STANDARD REPRESENTS WITH THE CHANGES OCCURRING IN THE WORLD AROUND IT. WE NEED TO "UPDATE" WHEN IT'S SMART AND CHANGE TO ANOTHER STANDARD WHEN THAT MAKES SENSE. THIS - IN TURN - MEANS THAT BOTH SYSTEMS COMMAND AND LOGISTICS COMMAND MUST SHARE A LONG TERM INVOLVEMENT IN BOTH PRESERVING AND ADAPTING THESE STANDARDS TO CHANGE. THE POTENTIAL IMPACTS OF UNADAPTABLE, INFLEXIBLE STANDARDS ON THE SERVICES AND INDUSTRY ARE TOO GREAT TO DO ANYTHING LESS.

WELL--RATIONAL STANDARDIZATION IS A LOT EASIER TO SAY THAN TO ACHIEVE. YET I AM GRATIFIED AT THE PROGRESS WE HAVE MADE. LET ME ILLUMINATE WITH SOME RATIONAL STANDARDIZATION EFFORTS NOW UNDERWAY.

OUR CURRENT TRIAD OF DIGITAL AVIONICS STANDARDS ARE EXAMPLES OF RATIONAL INTERFACE STANDARDIZATION. THEY HAVE NOT, TO DATE, INHIBITED THE TRANSITION OF TECHNOLOGY INTO WEAPON SYSTEMS. IN FACT, WE THINK THE PRESENCE OF STANDARDS HAS ACTED AS A CATALYST TO ACCELERATE TECHNOLOGY TRANSITION AND INSURE COST-EFFECTIVE AVIONIC SYSTEMS.

MIL-STD-1553, THE MULTIPLEX DATA BUS STANDARD, IS THE MOST MATURE OF THE GROUP AND PROVIDES THE BEST INSIGHT INTO THE EFFECTIVE USE OF STANDARDIZATION. BECAUSE "1553" IS AN INTERFACE STANDARD, IT IS INHERENTLY TECHNOLOGY INDEPENDENT. DATA BUS INTERFACES MAY BE DESIGNED USING DISCRETE-SEMICONDUCTOR, INTEGRATED CIRCUIT, MICROPROCESSOR, OR EVEN VHSIC TECHNOLOGY. IF DESIGNED TO THE STANDARD, THEY ALL WORK TOGETHER. THE FIRST "1553" BUS INTERFACE DESIGNS EMPLOYED A MIX OF DISCRETE COMPONENTS AND MEDIUM SCALE INTEGRATED CIRCUITS. BUT AS THE STANDARD MATURED AND USAGE INCREASED, INTERFACE CIRCUIT VENDORS RESPONDED TO THE MARKET WITH THICK FILM HYBRIDS AND LSI CIRCUITS--AND V-LSI CIRCUITS WILL SOON BE INTRODUCED. A CONTINUOUS STREAM OF NEW TECHNOLOGY DESIGNS HAS BEEN MADE POSSIBLE BY THE PRESENCE OF THIS FIRM INTERFACE STANDARD. WITHOUT IT, MANY VENDORS WOULD NOT HAVE MADE THE FINANCIAL INVESTMENT NECESSARY TO IMPROVE THE TECHNOLOGY.

MIL-STD-1750, THE COMPUTER-ARCHITECTURE-INSTRUCTION-SET STANDARD, IS A RELATIVE NEWCOMER TO THE STANDARDS SCENE. IT TOO REPRESENTS AN INTERFACE, BUT IN THIS CASE IT IS THE INTERFACE BETWEEN THE AVIONICS COMPUTER HARDWARE AND THE SUPPORT SOFTWARE (THE COMPILER, ASSEMBLER, LINKER, LOADER, ETC.). EARLY ON IN THE DEVELOPMENT OF "1750," WE GOT LOTS OF PRESSURE FROM SOME SOURCES TO DEFINE A STANDARD PIECE OF AVIONICS COMPUTER HARDWARE, AND THUS HELP ATTACK THE SPARES ELEMENT OF THE SUPPORTABILITY PROBLEM.

WE CAREFULLY THOUGHT ABOUT IT, BUT ULTIMATELY REJECTED IT BECAUSE IT WOULD INHIBIT TECHNOLOGY TRANSITION IN AVIONICS COMPUTERS--INHIBIT IT BEYOND THE PAY-OFF WE COULD SEE. INTERESTINGLY ENOUGH, THE U.K. IS ADOPTING "1750" AS A DEFENSE STANDARD JUST AS THEY DID "1553." WHILE "1750" DOES NOT ADDRESS THE HARDWARE ASPECT OF SUPPORTABILITY, IT DOES DEAL WITH THE SOFTWARE ASPECT. USE OF "1750" PERMITS STANDARD SUPPORT SOFTWARE TOOLS TO BE EMPLOYED IN THE DEVELOPMENT AND MAINTENANCE OF AVIONICS SOFTWARE. AN AIRCRAFT USING "1750" BASED PROCESSORS IN ITS HUD, INERTIAL SYSTEM, STORES MANAGEMENT SET, RADAR, AND SO FORTH, AND AS THE MAIN INTEGRATING COMPUTER, NEEDS BASICALLY ONE SET OF SUPPORT SOFTWARE TOOLS, AND ONE TEAM OF SOFTWARE ENGINEERS TO DEVELOP AND MAINTAIN THE AVIONICS SOFTWARE. THIS SUPPORTABILITY SIMPLICATION IS ACCOMPLISHED WITHOUT INHIBITING TECHNOLOGY SINCE IT MAKES NO DIFFERENCE WHAT TYPE TECHNOLOGY IS USED IN A "1750" COMPUTER--ONLY THE "SOFTWARE INTERFACE" MUST BE OBSERVED. FURTHER, RECENT ACQUISITION PROGRAMS, SUCH AS THE F-111 UPDATE, HAVE SHOWN THAT THE STANDARD HAS FOSTERED INCREASED COMPETITION WITH A CONSEQUENT REDUCTION IN ACQUISITION COSTS.

MIL-STD-1589, THE JOVIAL HIGH ORDER LANGUAGE, IS THE THIRD MEMBER IN THE TRIAD OF AVIONICS STANDARDS. IT DEFINES A STATE-OF-THE-ART LANGUAGE WHICH ENCOURAGES USE OF "MODERN" PROGRAMMING TECHNOLOGY--SUCH AS STRUCTURED PROGRAMMING--AND MODULAR, TOP-DOWN DEVELOPMENT IN AVIONICS. MIL-STD-1589 COMPLETES THE INTERFACE BETWEEN THE AVIONICS COMPUTER PROGRAMMERS, THE SUPPORT SOFTWARE AND THE MIL-STD-1750 COMPUTER. SOFTWARE DEVELOPMENT IS OFTEN ON THE

CRITICAL PATH FOR NEW WEAPON SYSTEMS, AND THESE TWO STANDARDS TOGETHER WILL PROVIDE THE ABILITY TO BEGIN DETAILED SOFTWARE DESIGN AND DEVELOPMENT INDEPENDENT OF HARDWARE AVAILABILITY.

IN CONJUNCTION WITH MIL-STD-1750, 1589 PROVIDES THE STABILITY IN COMPUTER LANGUAGE AND HARDWARE INTERFACE NEEDED TO STIMULATE INVESTMENT IN COMPLEX SUPPORT SOFTWARE SYSTEMS. THESE SYSTEMS, SUCH AS OPTIMIZED COMPILERS AND "INTEGRATED SOFTWARE DEVELOPMENT ENVIRONMENTS," WILL PROVIDE DESIGNERS WITH EASY TO USE, PROVEN "TOOLS," AND MANAGERS WITH INCREASED VISIBILITY AND CONTROL. THIS EXAMPLE OF "TECHNOLOGY INFUSION," RESULTING FROM USE OF THE STANDARDS, CAN ULTIMATELY CONTRIBUTE TO THE SOLUTION OF PROBLEMS SUCH AS SOFTWARE RELIABILITY, CODE PRODUCTIVITY AND OTHER DIFFICULT SOFTWARE PROBLEMS WE ALL FIND ON OUR PLATES.

I THINK THE F-16 MULTI-NATIONAL STAGED IMPROVEMENT PROGRAM (MSIP) IS A GOOD ILLUSTRATION OF DIGITAL AVIONICS STANDARDS BEING IMPLEMENTED ACCORDING TO PLAN. ALL OF THE NEW COMPUTERS WILL BE "1750A" TYPES, PROGRAMMED IN THE SAME HOL - "1589" AND WILL USE THE SAME SOFTWARE SUPPORT PACKAGE. THE ENTIRE SYSTEM WILL BE INTEGRATED USING THE "1553" DATA BUS. THIS WILL BE THE FIRST TIME A MAJOR WEAPON SYSTEM INTEGRATOR HAS ASSEMBLED ALL THESE AS INTENDED--AND GENERAL DYNAMICS IS DOING IT BECAUSE IT MAKES SENSE.

NOW, THE AIR FORCE'S HARDWARE STANDARDS ARE NOT QUITE SO TRANSPARENT TO EASY TECHNOLOGY INSERTION. SOME OF THESE, LIKE THE F^3 INS, COULD RESPOND TO COMPETITION AND TECHNOLOGY INSERTION IF WE HAD AN APPROPRIATE BUSINESS STRATEGY AND LOGISTIC SUPPORT PHILOSOPHY THAT WOULD NOT PENALIZE ALL BUT THE ORIGINAL WINNING CONTRACTOR. THE COMBINED ALTITUDE RADAR ALTIMETER AND THE STANDARD CENTRAL AIR DATA COMPUTER ARE LESS COMPLEX SUBSYSTEMS AND DIRECTED PRIMARILY AT A LARGE RETROFIT MARKET. THEY ARE, THEMSELVES, TECHNOLOGY INSERTION EFFORTS IN AN AREA WHERE CHANGE HAS BEEN SLOW TO COME, AND THE NEED FOR FUTURE CHANGE UNCERTAIN. WE MAY BE ABLE TO PERFORM THESE FUNCTIONS ANOTHER WAY IN THE NEXT GENERATION AIRCRAFT.

WE THINK OUR DECISION TO FOCUS OUR STANDARDIZATION EFFORTS PRIMARILY ON THE INTERFACES RATHER THAN ON HARDWARE, HAS PROVEN TO BE A GOOD ONE. OUR DIGITAL AVIONICS STANDARDS, CREATED TO ASSURE THE POSSIBILITY OF CONTINUOUS TECHNOLOGY INFUSION, HAVE NOT, TO DATE, INHIBITED THAT PROCESS. AS A MATTER OF FACT, THESE STANDARDS HAVE HELPED ESTABLISH AN ENVIRONMENT WHICH EXPEDITES THE TECHNOLOGY TRANSITION PROCESS. ON THE OTHER HAND, OUR HARDWARE STANDARDS HAVE BEEN FOCUSED PRIMARILY ON CURING EXISTING LOGISTICS SUPPORT PROBLEMS, AND HAVE BEEN DIRECTED AT LESS COMPLEX SUBSYSTEMS IN THE "COMMON" AVIONICS AREA. WE DID THIS BECAUSE IT IS POSSIBLE TO USE

THEM ACROSS A NUMBER OF WEAPON SYSTEMS. THE "NEED" FOR TECHNOLOGY INFUSION INTO THESE BLACK BOXES IS LOW AND IS OUTWEIGHTED BY THE IMMEDIATE LOGISTICS SUPPORT ADVANTAGES AND LOWER TOTAL LIFE CYCLE COSTS. EACH WAS SELECTED AS A CANDIDATE BECAUSE OF THE VERY HIGH SUPPORT COSTS OF THE CURRENT SUBSYSTEMS THEY WILL REPLACE.

IN CONCLUSION, I BELIEVE OUR TRACK RECORD HAS BEEN GOOD SO FAR, BUT THAT DOESN'T MEAN WE CAN REST ON OUR LAURELS. WE SPONSOR THESE CONFERENCES SO THAT ALL OF YOU WHO ARE INVOLVED IN THE STANDARDIZATION EFFORT CAN CALIBRATE AND TRACK US. YOU NEED TO KNOW WHERE WE ARE COMING FROM AND WHERE WE ARE HEADED. WE, ON THE OTHER HAND, HAVE A GREAT NEED OF YOUR FEEDBACK AND CRITIQUE.

TODAY'S SESSIONS SHOULD SHOW YOU THAT THE DOD'S EMPHASIS IN STANDARDIZATION IS INCREASING AND THE CROSS TALK AMONG THE SERVICES IS EXPANDING. YOU CAN EXPECT MORE JOINT STANDARDIZATION PROGRAMS IN THE FUTURE. WE'VE ESTABLISHED A NUMBER OF COMMUNICATION CHANNELS FOR YOU TO BE HEARD. IF SOME QUESTIONS ARE STILL UNANSWERED AFTER THE NEXT THREE DAYS OF DISCUSSION, THEN GET IN TOUCH WITH MY STANDARDIZATION TEAM AT ASD: MY CONSCIENCE -- THE DEPUTY FOR AVIONICS CONTROL; AND MY STRONG TECHNICAL EXPERTISE-- THE DEPUTY FOR ENGINEERING. IF WE ANSWER JUST ONE FUNDAMENTAL QUESTION FOR EACH OF YOU, WE'LL LOOK UPON OUR INVESTMENT IN THIS CONFERENCE AS HAVING AMPLE RETURN.

THANK YOU.

DOD PERSPECTIVE ON STANDARDIZATION

Mr. WILLIAM A. LONG

DEPUTY UNDERSECRETARY
FOR
ACQUISITION MANAGEMENT

Let me first tell you how delighted I am to be here to address this group on the DoD perspective. It is a real delight to me as an OSD representative to see the enthusiasm for the standardization program, both within the Department of Defense and the contractor community that the crowd here evidences.

This is the second AFSC conference on standardization. It evidences the growing support for, as General McMullen properly puts it, "Rational Standardization". The concept is here to stay, what I'd like to do as a starting point is to spend a few minutes talking about the role of standardization within the planning and policy process of OSD and I want to do that by talking about three basic points. First of all, the Acquisition Improvement Program of which most of you have some familiarity; particularly Initiative 21 which is the standardization initiative. This, as many of you know, calls for the Department, overall, to use standard operation and support systems to the maximum practical extent and I emphasize the word practical because as we know the standardization process in theory and in practice is not an end in itself but rather a tool to be used to improve the quality and the cost effectiveness of the weapon systems and support systems that we acquire. That is the ultimate goal. I also want to focus a little bit on how the standardization efforts throughout the Department have a very positiveeffect on other elements of our Acquisition Improvement Program and our attempts to rapidly implement that program. Finally, I would like to address briefly some recent changes in the Department of Defense Standardization Program both as to substance and emphasis.

The first point, the Acquisition Improvement Program and the role of standardization within that program. A little bit of background, as many of you know when Messrs. Wineberger and Carlucci came into office in January of last year, they recognized that even though the Department of Defense acquisition process system and people are probably the finest in the world, we still had to take effective steps to do a better job throughout the acquisition process. We had a mandate as the polls reflected to rearm America, to commence a defense buildup, to reverse a course of the prior few years. But in order to maintain the mandate, if you will, the balance in favor of the defense buildup, which was a fragile balance and always will be, we had to do the best possible job in making effective use of the taxpayer dollar. So, given the recognition of the need to improve the acquisition process, the question is how to go about it.

One of the early and easy decisions that Carlucci and Wineberger made was that we would not as a Department engage in a new in-depth study of the acquisition process. It has been studies to death. The Commission on Government Procurement in the early 70's, the Defense Science Board Task Force of the mid-70's, and on and on and on. What they really decided to do, which I think in retrospect was a very wise and effective decision, was to put together a steering rgoup or a task group to look at the studies of the past generation, draw out of those studies the principal practices and phjilosophies of acquisition that made sense, tie them together in a package and set about implementing them.

This all came to fruition in March, 1981 when the task group presented to Mr. Carlucci a report entitled "Improving Defense Acquisition Systems and Reducing systems Cost". the steering group which had reviewed literally hundreds of recommendations brought up within the Department and through substantial industry participation, the steering group synthesized this body of suggestions diwb ti thirty-one elements. One of these recognized the need to consolidate requirements, to develop single standard items either interface or in more detail depending upon how the standardization p ogram itself were to evolve. It recognized that there was tremendous payoff and benefit here if we do the right thing in a smart way. The thirty-one recommendations resulted in Mr. Carlucci's famous (or infamous) April 30, 1981 memo called the "DoD Acquisition Improvement Program". By way of a footnote, as many of you know, there are 32-elements to the program now; the thirty second one on competition was added several moths later. But in that April 30th memo again the notion of the contribution that effective standardization can make in reducing overall acquisition cost was recognized and is embodies, embraced in Recommendation or Initiative Number 21 in that memo. I hasten to point out that there is no prioritization in the numbering system of that memo. It is not to be inferred that because standardization or the standardization philosophy is in Number 21 that the precedeing 20 are more important. There is no prioritization in the arrangement of the elements within that memorandum.

In that recommendation or in that initiative, Mr. Carlucci directed that standard subsystems and related support equipment be identified and developed to meet projected weapon system needs, with the view to achieving significant benefits in the areas of reducing acquisition cost, reducing develpment time, reducing the maintenance, supply and operating costs and saving substantial time and money by using previously tested, reliable and proved equipment. Now let me emphasize those four points because they really reappear and they will throughout this conference as well as throughout the entire program of standardization.

The four points; reducing or lowering acquisition costs; reducing development time; cutting the maintenance, supply and operating costs; and using previously tested, reliable and proved equipment. There are lots of different way to say that but those are, I guess, the four points that really drive the program.

As I indicated earlier and as General McMullen indicated, standardization is not again in and of its self, but its a program that can lead to positive benefits in these four areas. Now one could expand that and put susbsets ther, but I think those really are the fundamental underpinnings of what we are trying to accompish. Now, progress toward implementing the Carlucci Initiative Number 21 is, I think we have to acknowledge, satisfactory. It is slower than we would like but it is coming along. General McMullen indicated or identified several programs; we're always impatient, there are others that I could mention; he mentioned 1750; there is the multi-role radar, the Joint Tactical Information Distribution Systems which they call JTIDS, the next generation IFF system; these may not be quite as far along as some of the programs that General McMullen identified, but they are major programs that have significant involvement in the standardization process or major programs in which the standardization process is playing a significant role. So we are coming along. I guess we're always impatient adn we ought to be impatient, we like things to move along faster, but Rome wasn't built in a day and we're not going to change in a day, or a week, or a month the basic way we do business.

We move it along slowly doing the best job we can to make certain our decisions and our implementation programs are effective and in essence, that we do the right thing.

In all of these systems, the ones the general mentioned, the ones that I mentioned and others that are too numerous to mention, we really are recognizing and beginning to see the tangible benefits in cost, time supportability and, of course, readiness; the four tiems I mentioned earlier perhaps stated in slightly different words. As a part of our DoD Standardization effort, we set up a cross services or tri-services DoD panel to examine ways that contractors and program managers can better identify and develop and use standardized systems and subsystems. Some of the results of this panel have now come into our defense material standardization and specifications board and are being evaluated. As time goes on, you will see new ideas coming out of the field, to the community which you represent, in a large part, the people who really make it happen. You'll see these ideas evolve out of the panels and the boards; we'll dialogue it with the entire community and come up with new and better and more effective ways to keep the program moving both philosophically, from a policy standpoint, and practically, from a program implementation standpoint.

Let me move to the second point now: how standardization can help the acquisition process in other ways. One of the examples that is most indicative of the broad role of standardization is the area of capital investment. As you know, we have a variety of initiatives designed to stimulate capital investment within the defense industry for a whole variety of reasons stemming from or starting with program instability and running the gamut to a hundred other contributing factors. We have seen a defense industrial base in the last 10 or 15 years substantially under-capitalized. New investment just hasn't flowed into the defense industry the way many of us would like it to have flowed in order to keep a strong industrial base. And I'm not being critical of the industry, I think the system of which we are all a part has really inhibited investment.

One of the things that standardization does is play a significant role in reversing that trend and stimulating investment. As we move into areas of multipurpose support equipment, for example, this would permit the contracting community to engage in longer production runs and this is a very natural stimulus attractive for capital investment if the contractor knows he is going to build a thousand widgets over three years rather than 250 widgets this year and maybe none next year; then he's really in a position to make the investments in producibility enhancing equipment and other kinds of equipment to just do a better more efficient job. that's one of the cross fertilization benefits of standardization and there are others. Running throughout the entire Acquisition Improvement Program you will find various initiatives directed to readiness and support. There are at least half a dozen that bear directly on readiness and support. Standardization is a program that can, independent of the six specific initiatives, enhance readiness support. Our standards bear parts-related support equipment, training devices and related technical data, Flowing out of the standardization program can go a long way to improving readiness and support.

Effective competition can be enhanced when properly planned and executed. Standardization encourages technical improvements, and in view of many of us, will permit greater competition in development and in production. There are naysayers, of course who dispute this and say that standardization stifles innovation and stifles competition, this, of course is neither the desired result nor the necessary result, and, as General McMullen pointed out, if properly ... standardization can enhance innovation or technology transition and can and ... in face enhance competition.

Training is simplified as i mentioned since the need for knowledge and skills are reduced at least to the extent that there is a cross-utilization of common equipment.

Some of the changes in direction and emphasis: the third point I wanted to mention briefly within OSD abd tge policy activities throughout the services. The Defense Material Standardization and Specifications Board which I mentioned earlier is a board that was a bit dormant for a number of years until about mid-1981. It is made up of, as many of you know, flagrank or equivalent civilian representatives from all of the services, from the Defense Logistics Agency, from Research and Engineerig and from the logistics arm of MRA and L. The board's responsibilities include developing defense standardization guidance, which I'll come back to in a moment, recommending through the Acquisition Management Office to Dr. DeLauer cost effective standardization programs reviewing progress on the standardization programs. The board itself is now considering how it can better work with a couple of service panels, one is the Joint Services review Committee on avionics, which the folks here at Wright-Patterson are very familiar with and the Joint Logistics Commanders Panel on ground support equipment. There is a natural fix between the Board, the Review Committee and the Panel not that the activities of the Review Committee and the Panel are going to be diminished but rather there will be an enhanced opportunity to speed along the process by the interaction between the Board and these two groups.

The Board is also looking at a very important matter, a better way to ahndle the budgeting and funding aspects of defense wide standardization programs. Specifically, can we do a better job of supporting the programs manager's upfront needs for standardized material as a prt of ne systems equipment developments. That's a problem that we've got to face up front and I want to come back to that because it seems that that is one of the big challenges for this conference.

I know that you have a lot of activity related to specific programs, but I think while the standardization community is collected here, it is entirely appropriate to look at some of the basic underpinnings to the overall program as well as the specific elements and funding it seems to me is one of those general issues that out to be addressed. Within the department and through the auspices of the Standardization and Specifications Board as well as others, there has been a substantially increased emphasis overall in the standardization process. The services and their implementing organizations are following this lead or maybe they're leading and we're the followers but it's plain to see throughout the military departments that the commitment to rational standardization and achieving the benefits of a successful standardization program is really coming to the full.

The Air Force for example, has increased the level of authority and responsibility fo its standardization office in the Secretariate, it now reports directly to the Deputy Assistant Secretary for Logistics, Lloyd Mosemann. ... the services are looking for ways to enhance the stature, if you will, of the standardization activities. I mentioned earlier the guidance ...

Standardization Board and Dr. Delauer's office issued the 1982 Defense Standardization Program Guidance. Many of you perhaps have seen that and read it in detail, but let me just mention briefly some of the things that it addressed.

The development and implementation of standardization plans for identified high payoff product and technology areas. Areas such as VHSIC, fiber optics and embedded computer resources. The reduction in the proliferation of material to satisfy similar generic kinds of operational requirements. Optimizing the use of commercial products which meet military requirements in lieu of products designed specifically for DoD use.

The '83 guidance is well into preparation and should be out shortly. I would guess it would be out earlier in '83 than the '82 guidance was. At least I hope this one come out before March of '83. Again, we are going to be focusing on ways in whioch the standardization program, rationally applied, can reduce acquisition life cycle cost, reduce the acquisition risk, reduce lead times and improve readiness and supportability of our defense equipment. The challenges that we, as a Department and we, as the standardization community, face are geat. We're looking particularly at standardization issues involved with embedded computers and avionics as well as a broader range of programs that will be alluded to throughout the course of the three days.

Let me urge you as you approach your technical sessions in this conference, to keep in mind the broader issues - again standadization is not an end in itself but it is a way to achieve greater eficiency and effectiveness in the DoD systems and material acquisition process. The big challenge which I mentioned earlier of a general nature, that we need to wrestle with is funding. All too often the program manager has a legitimate concern as the standardization process for a particular subsystem commences with his program that it is going to cost his program greater dollars than it would if he simply went forward with his program on a unique basis. The upfornt funding, how do we properly allocate that up-front funding, those up-front dollars among not only the initial program which out not to bear the total cost but all the successive programs that benefit from the standardization effort. There are a variety of ways to handle this but so far we seem to get hung up in our internal budgeting and accounting process and we haven't found an easy way to escrow those funds, if you will, or an escrow account to spread those funds over successive programs. This community is a community made up of very bright people and I would challenge it to spend some time individually, in small groups and in large groups to work that problem and to come up with some suggested solutions.

the big caution that I would urge upon you, and which General McMullen also alluded to, is to incorporate in the program philosophically and practically a sufficient amount of flexibility so that we don't stub our toes within a structure that is too rigid. All too often throughout the history of the Department of Defense and other big institutions, this is not a foible unique to the Department, but a policy comes down without the flexibility that the executors need to do the right thing. And I think as this rational standardization program evolves, we have to be careful not to let ourselves fall into that pit.

The need for flexibility in a humorous note is best exemplified by a [illegible] that many of you from Wright-Patterson perhaps have heard, that perhaps those your from elsewhere haven't. It involves a subject near and dear to the hearts of many Ohioans; namely the Ohio State Michigan football game. Some years ago, Michigan late in the game, which was tied to 20-20, had just gotten a first down at its own 30 yard line but in the proces its second string QB had been knocked unconscious. The first string QB had been injured earlier in the game, so Bo Schembechler looked down the bench and he sees his 3rd stringQB, a kind of lumbering, ponderous chap who hadn't played much during the year. He sends him in with the instructions to run three plays and punt. So lo and behold the QB goes in and on the first play he startles Schembechler by throwing a pass that's complete for 25 yards. The next play he runs a double reverse that picks up 20 yards. The third play he completes another pass down to the 10 yard line, and of course, on the fourth play he punts. Schembechler is going quite beserk as the QB comes back off the field and he grabs him and says, "What in the world were you thinking of when you punted?" This ponderous QB looks into Schembechler's eyes and he says, "I was thinking we got the dumbest coach in the world".

So, let us as we evolve the rational standardization program, let's build into it sufficient flexibility, so that the executors, the people who are really charged with getting the job done, that they have sufficient flexibility to do the right thing.

I think that if it is not clear to the community yet, it will become clear as time goes on, that the Secretary of Defense and his staff are very much committed to the program that is so important to you folks that you are working on a daily basis. Let me tell you that if ther is anything that we can do to help ypou, either in a positive way or in a negative way, in the sense of removing something that we've done that yopu see as inhibiting you, don't hesitate to contact us; because we are simply a support organization her to support you folks in doing the job that you do so well. But if we can be of any help to you, let us know.

I wish you a successful and productive conference and as I say, if there is anything we can do, please let us know and "thanks" for the invitation to come here and share with you some of the thoughts on the OSD perspective.

William A. Long took office as the Deputy Under Secretary of Defense for Research and Engineering (Acquisition Management) in 1981.

Mr. Long was born in Cincinnati, Ohio in 1937. He graduated from Xavier University in 1959. Mr. Long's college education was interrupted for a two-year period during which he served on active duty with the U.S. Army. Following his graduation, Mr. Long attended the University of Pennsylvania, Wharton School of Business and received his M.B.A. degree in 1961. Several years later, Mr. Long enrolled in the Boston College Law School where he was awarded a degree in 1967.

Prior to his appointment, Mr. Long was a partner in Latham & Watkins. His activities were concentrated on business matters including securities, financing and government contracts. He has had extensive experience with the government acquisition process, including work on contracting for major systems. In his position as Deputy Under Secretary, Mr. Long serves as the principal advisor to the Under Secretary of Defense for Research and Engineering in all matters concerning management and policy for the Department of Defense acquisition process. Mr. Long is the Chairman of the DoD Acquisition Improvement Program Steering Group which is responsible for implementing major improvements both in weapons acquisition philosophy and the acquisition process itself. Under his guidance, many of the policy decisions required for implementation of improvements have been made and put into effect. Mr. Long is also responsible for making procurement system improvements in accordance with Executive Order 12352 of March [illegible], 1982 [illegible] Federal Procurement Reforms and is the DoD member of the [illegible] Committee on Procurement Reform.

Mr. Long and his wife, Janey, have six children, and reside in [illegible] Maryland

Speech To 2nd AFSC
STANDARDIZATION CONFERENCE
30 November 1982
Dayton Convention Center

by

MAJOR GENERAL JASPER A. WELCH
Assistant DCS/RD&A
HQ United States Air Force

Good morning! I have come this morning to talk about what some see as a double edged sword; on one side technology -- on the other side standardization. We are in the technology business, but without some standards the diversity and complexity of technology is threatening to stall our progress in applying our technology to systems.

There are some who are convinced that complexity is all bad. They would have us go back to the weapons of World War II. If you can get these people to sit still and listen, I have some facts for you to tell them:

In World War II, our fighter aircraft averaged one combat mission every four days. In Korea, the rate had increased to one mission every three days. By the Vietnam War, USAF fighters were averaging nearly one mission a day and current planning rates are even higher. Surge tests with F-15 units in Europe have demonstrated rates of better than four per day. Now you can't do that with piston engines and tube type radios. Technology and complexity have brought us a long way.

We do have some problems though: in Vietnam we found ourselves with aircraft grounded for lack of spare avionics while we had all sorts of spare avionics that would not fit the aircraft.

Following the Airlines Lead in F^3 Standardization

This problem stimulated us to look at how the airlines had solved their very similar problem. We found that they had two things going for them:

first, the concept of form, fit and function (or F^3) standardization, and

second, the process for development of standards.

The concept of F^3 standardization has been abstracted slightly to meet our needs and in its more general form goes by the name of "Interface Standardization."

Having looked at the airlines' success story, and after some study on how to adapt the essence for our use, the first thing we did was to try it out. We looked around for a susbsystem that might have a fairly wide application and one where we had a lot of non-interchangeable equipment in existing aircraft. The system couldn't be too close to the state-of-the-art, and should have a number of potential vendors.

The standard medium accuracy Inertial Navigation System was the choice. The first industry open forum was held in 1976 and a year later the spec was completed. A contract was awarded to Litton in 1979 for INS's for the A-10. First deliveries were taken in 1981 so that after five years we got our first system.

Interface Standardization As We Know It Today

In 1974, the Avionics Lab had become deeply involved in the Digital Avionics Information System or DAIS program and in 1975 began to advocate the adoption of interface standardization as we now know it. By this time, the Aeronautical Systems Division had enjoyed good success with the MIL-STD-1553 multiplex bus during the F-16 development. In 1975, the Avionics Laboratory identified a version of JOVIAL as the preferred higher order language for avionics and in 1976 began pursuing a computer instruction set architecture standardization effort in cooperation with the Aeronautical Systems Division. These efforts put into place the standards for a computer programming language and a computer instruction set architecture which eventually became MIL-STD-1589B and MIL-STD-1750A.

Establishment of the Deputy for Avionics Control

At about this same time an Air Force Scientific Advisory Board study identified the avionics acquisition process as something that needed attention. After additional study, discussion, and correspondence Dr. Martin, then Assistant Secretary of the Air Force for Acquisition and Logistics, approved a joint AFSC/AFLC plan and charter creating the Deputy for Avionics Control.

We put out a regulation (AFR 800-28) titled "Air Force Policy on Avionics Acquisition and Support." This regulation described in detail the responsibilities of the Deputy for Avionics Control.

Actions Speak Louder Than Words

Since we established the DAC we have put out a joint RD/LE policy letter requiring the use of JOVIAL J73, MIL-STD-1750A, and MIL-STD-1553B for avionics, and we have encouraged their use in other applications.

We have written the requirement to use these standards into just about every avionics related PMD we have issued over the last three or four years.

We have begun the process of putting a common bus-oriented avionics architecture into almost every aircraft in the Air Force inventory.

We have issued (jointly with the Army and the Navy) MIL-STD-1760 as our aircraft-to-stores electrical interface. We are requiring that all of our new weapon developments use this interface, and we are developing plans to incorporate it into our aircraft at the earliest opportunity.

Joint Service Review Committee

We have established a tri-Service group called the Joint Service Review Committee for Avionics Subsystems and Components or JSRC for short.

The Air Force representative on this committee is the Deputy for Avionics Control. The JSRC was established by a joint memorandum of agreement between the Army, Navy, and Air Force Assistant Secretaries for Research and Development to look for opportunities to achieve joint Service avionics standardization and to execute joint developments when a common requirement exists. They also serve to document cases where a joint development is not prudent.

We already have one joint program in development with the Navy -- the Standard Central Air Data Computer which the Air Force will put on the F-4, F-111, C-5A, C-141, and B-52 over the next few years.

Other projects in various stages of program formulation and planning are a Digital Audio Distribution System to replace many of our antiquated aircraft intercom systems, a crash survivable flight data recorder, a heading/attitude reference system, and a ground proximity warning system. The heading/attitude reference system will be an Army/Navy funded development while the rest will be tri-Service efforts.

Rational Standardization Benefits

CARA

- Replaces five high and eight low altitude radar altimeters
- 4000 inventory aircraft retrofit (TAC, SAC, MAC), plus F-16 MSIP
- Priced options 2700 (additional) (AF, USN, USA)
- Nuclear Hardened
- $5800 (ea) (Est $123M cost avoidance) 15 yrs LSC

F^3 INS

- 525 hours (fighter) MTBF (guaranteed)
- Acquisition/support cost avoidance - minimum $60M

CDU for F^3 INS (A-10 Aircraft)

- Modern technology (CRT and Keyboard)
- $11M cost avoidance

1750A Processor for F-111

- 2000 hour MTBF (guaranteed)
- $46K/Unit (estimated cost avoidance $20-40M)

C/KC-135 Update

- Specifications/standards - saved $20M
- MUX Bus/Architecture - saved $600M - 10 yr LCC

MRR (Multi-Role Radar)

- F-16 and B-1B Commonality (3 of 4 LRU's)
- Acquisition - $114M savings
- Shared production manufacturing - $130M savings (additional)
- Support equipment/training/data TBD (EST $10-15M)

SCADC (Standard Central Air Data Computer)

- USAF and USN commonality (5000 aircraft)
- First product of Joint Service Review Committee (JSRC)
- Estimated acquisition cost avoidance - $30M (4 standard CADCs vs Six unique CADCs)

GPS

- F^3 specification for receiver/processor unit
- NATO wide commonality
- Estimated $200M cost avoidance to USAF and NATO

Common Radios (ARC-164, 186, 190) and TACAN (ARN-118)

- USAF wide commonality
- LCC savings - $71M per year minimum

Our Record Speaks Loud and Clear

Overall, I think we have a very good record of successes in avionics standardization. Successes which have come from three main thrusts:

First, adopting as standard items those subsystems that were reliable, maintainable, and simple enough to be adapted to many different installations. Examples are the ARC-164 and ARC-186 radios, and the ARN-118 TACAN.

Second, developing standard subsystems when none of the existing equipment met the criteria for adoption as a standard item. The Standard Central Air Data Computer, Standard INS, and the Combined (high/low) Altitude Radar Altimeter (or CARA) are good examples of this type of standardization.

Third, where technology was changing rapidly, or where an aircraft unique piece of hardware was required, we have insisted only on adherence to the standard interfaces to enhance supportability, reduce future aircraft modification costs, and increase competition up front. Examples can be found in the F-111 digital computer replacement program and in both the F-16 and F-15 MSIP programs.

The results are impressive:

CARA will produce a $123 million cost avoidance over 15 years.

The F^3 INS will produce a $60 million cost avoidance; and that will increase as we buy more of them and put them into more aircraft.

The control/display unit for the F^3 INS in the A-10 will yield a cost avoidance of $11 million.

The new computer for the F-111 will yield at least a $20 million cost avoidance -- and because we had the forethought to put a MIL-STD-1553B multiplex bus interface into the computer, much larger savings will accrue as we proceed with the F-111 Avionics Modernization Program.

Inclusion of the multiplex bus architecture in the C/KC-135 update program will save $600 million over a 10 year life cycle, and the use of the interface standards will save about $20 million in acquisition costs.

Commonality in the F-16 and B-1 radars will save $244 million in acquisition costs with an additional saving of at least $10 million from common support equipment, training, and data.

Wide use of the ARC-164, 186 and 190 radios and the ARN-118 TACAN is saving about $71 million per year in reduced maintenance expense.

The standard CADC will save us another $30 million and NATO adoption of an F^3 specification for a Global Positioning System receiver/processor is anticipated to save at least $200 million dollars. I think it is clear that we are already reaping substantial benefits from our standardization program!

Reliability Efforts are Closely Related

There are other areas of electronics that we are interested in which complement our standardization efforts. First and foremost is our campaign for avionics reliability. You can read more about this campaign in the upcoming December issue of Electronic Business, but I would like to take a moment to touch on the highlights since they are so closely coupled to our standardization program.

I will make three claims about reliability:

1. You can get as much as you want
2. It costs less than you think
3. The payoff is greater than you expect

At the line replaceable unit level we need about 2000 hours mean time between removals. For a wing of 96 aircraft, this corresponds to about one removal of a particular LRU per month at peacetime flying rates, and under a wartime maximum sortie surge situation a box would have a 90% chance of no failure in 30 days. That's the real requirement and your Air Force leadership is determined to put on a full court press to meet it. This past summer the Air Force Scientific Advisory Board, with major support from the electronics industry as a whole, looked into the promise of the "New Electronics." They came away with the firm conclusion that very significant reliability improvements are possible and would be facilitated by current technological advances. They also observed that existence of a technology does not guarantee it will be correctly applied or will solve the right problem without proper management attention and direction.

Key Role of the Design Engineer

But to make these opportunities come to fruition requires hard work, innovation, and serious goals. In my view the key to success is a fired up design engineer with solid backing from his industrial organization and his government program office.

If I were given the job of evaluating whether a piece of equipment was going to be reliable, I would go to the design engineer and ask the following questions:

Does your design want to work?

Are you using robust components?

Do you have confidence that your test procedures will discover faults in design and manufacturing?

Will your design tell the operator when it's broken?

Will your design keep on working even when it's broken?

I believe you will find that if your engineer is on his way to a reliable product, he will have ready affirmative answers -- and a rationale that will withstand scrutiny. If you get a glazed or patronizing look -- get yourself another engineer. If you get a negative reply, then you and your program office are in a lot of trouble.

Software Reliability

Software reliability is another area that clearly needs work. Programmable digital avionics are here to stay. Unfortunately, most industrial managers and all too many Air Force managers view software management as a mystery world. Having written more lines of code than most people, and having designed and supervised many times more, let me assure you that standard management approaches work just fine.

In fact it is pretty easy to lay out a few analogues to illustrate: coding errors are like wiring errors -- an error prone craftsman cannot be tolerated in either case. Top Down Structured Programming is strictly analogous to the Work Breakdown Structure. Software Development Tools perform the same function as an oscilloscope and other bench testing apparatus for functional checking. In both cases, testers and designers must have a proper arms length relationship. Overall design architecture errors are as fatal as ever. Finally, make or buy decisions remain the first responsibility of management.

The underlying difficulties in software at the present time are three-fold:

1. The hardware is subject to rapid technological change so that detailed reapplication of prior accomplishments are scant;

2. The field is rapidly expanding and thus attracting a lot of marginal performers at both the individual and organizational level.

3. Managers still don't give software the attention it merits.

The software problem is real and its causes are fundamental, but it will yield to solid management attention. Some companies do a good job -- we intend to get them on our team.

Our emphasis on reliability and quality will increase very visibly in the near term as the new Deputy for Acquisition Logistics at Systems Command Headquarters begins to focus on this subject.

Lessons Learned

Coming back to standardization, I would like to turn for a moment to the topic of "lessons learned." Perhaps as much as anything else, we have learned that standardization is a continuing process of education and evolution. *Education* because the engineer that *doesn't* know about the standards will always invent a *different* way -- and *evolution* because the engineer that *does* know about the standards *can* invent a *better* way.

We must continually be alert to counteract the first case and to encourage and support the second. For everyone of you in this audience there are at least ten that need to know more about the standards -- not just that they exist, but how and when to use them well, how to avoid their misuse, and when not to use them at all.

A standard that doesn't evolve is a dead standard. The lack of evolution indicates a lack of interest. No interest, no educati no education, no utilization; no utilization, no standard! The reason for a lack of interest doesn't really matter. It may be that there is no need for the standard or that technology has passed it by because the standard was not adaptable to change -- the result is the same.

Now a standard is most useful if it has a long life.

So we must be careful when developing new standards and modifying our old ones to assure that we don't make the standards obsolete unnecessarily soon.

Future Directions

Now a word about what I see as our future direction in standardization.

Right up front, let me tell you that interface standards are still our main thrust. They allow us to put the pieces in place today that will shorten development schedules, reduce modification costs, and increase competition.

They also allow products developed with company funds to be brought to us in a form that is compatible with both our aircraft and our logistics systems.

I am not aware of any cases where we have completely eliminated a government funded engineering development phase (with the possible exception of the F^3 INS). But I do think that there are opportunities for innovative products to be brought to the military avionics market that will fit our aircraft and work the first time out.

Current Standards Have a Good Record

We have an excellent history with our current set of our avionics architecture standards. For those of you who came for the tutorials yesterday, I am going to ask for your indulgence for a few minutes. For those of you who didn't, I want you to know why I think we have a success on our hands.

Avionics Interface Standards

Current

MIL-STD-1553B	Multiplex Data Bus
MIL-STD-1589B	JOVIAL J73
MIL-STD-1750A	16-bit Computer Instruction Set Architecture
MIL-STD-1760	Aircraft-to-Stores Electrical Interface
MIL-STD-1679	Software Development

Future

MIL-STD-1815	Ada - DOD Standard Programming Language for Embedded Computers
MIL-STD-1862	Nebula - 32 bit computer instruction set architecture
MIL-STD-XXXX	High Speed Multiplex Bus
MIL-STD-YYYY	Packaging Mounting and Cool for Airborne Electronics

USAF Application of the Avionics Interface Standards

MIL-STD-1553

F-5G	A-10
F-15 MSIP	B-52
F-16	B-1B
F-111	C-5B
KC-135	

MIL-STD-1589

F-4G	HH-60D
F-5G	MX
F-15 MSIP	Satellite Control Facility
F-16 MSIP	GPS Ground Segment
F-111	Advanced Cruise Missile
B-1B	F-15 Radar
LANTIRN Pods	Wide Angle Raster HUD
F-16/B-1B Radar	MATE

MIL-STD-1750

F-4G	LANTIRN Pods
F-5G	Wide Angle Raster HUD
F-15 MSIP	F-16/B-1B Radar
F-16 MSIP	F-15 Radar
F-111	MATE
B-1B	Advanced Cruise Missile
HH-60D	

MIL-STD-1760

F-15 MSIP	Common Strategic Rotary Launcher
F-16 MSIP	Advanced Cruise Missile
A-10	AMRAAM
B-1B	WASP
B-52	MSER
LANTIRN Pods	30mm Gun
ASRAAM	Conventional Standoff Weapon
MRASM	

MIL-STD-1553

MIL-STD-1553 defines our digital multiplex bus. It has been around since the start of the F-16 development program. Industry is familiar with it, we have lots of systems that use it, you can buy off-the-shelf parts to implement it, and there is commercially available test equipment for it.

MIL-STD-1750

MIL-STD-1750 defines our standard 16-bit computer instruction set architecture. Anybody can build to it, and almost every computer vendor selling into the military market is building a product that implements this standard.

There are MIL-STD-1750 computers going into LANTIRN, F-16, F-15, F-111, F-5G, MATE, A-10, B-52 and B-1, as well as other programs that you may be aware of. Over the period from 1980 to 1984 brassboard performance for comparable computers will increase by about an order of magnitude while size, weight, power and cost will go down by over an order of magnitude. Individual machines will push one or more of these parameters by additional factors of from two to ten. It is this kind of dynamic performance improvement that makes a technology-independent standard so desirable, and in this case I think we have good evidence that 1750 really is technology transparent.

We probably have a fair number of people in the audience today who have helped make it happen. To those of you who are here today, and to the rest who are not -- well done!

Congressional Language on 1750

I am sure that many of you are aware that the Congress put language into the Defense Authorization bill which directed OSD to relook at issuing DOD instruction 5000.5X and directed the Air Force not to fund any new developments of MIL-STD-1750 computers without a determination from USDR&E that it was essential.

I don't think that the language will hurt us in the immediate future, and in fact it offers some guarantee to industry that the Air Force is not likely to fund a development program in direct competition with a known company funded effort. Meanwhile there are some 20 existing variants out there, many of which have passed through the SEAFAC certification process here at ASD.

The Arguments Against 1750

There are two principal arguments that have been offered against the widespread application of MIL-STD-1750.

First comes the claim that standardization inhibits progress in technology, and

Second, it is said that Ada can assure transportability of software so that instruction set architecture standardization is not needed.

I think we have adequate evidence that computer technology is progressing quite nicely. The performance growth of various 1750 implementations over the past two years and the promise of VHSIC performance in the next two years leads me to question the motives of those who raise this as an issue.

In the second claim however, there may be some merit. There would certainly be major benefits to both our computer hardware and software efforts if Ada were the only language required -- that is, if we could do away with all assembly language programming. But so far as I know, nobody has yet demonstrated that this is really possible. However, I think that it has enough potential to be looked into in more depth.

MIL-STD 1589

MIL-STD-1589B defines JOVIAL J73, the Air Force standard higher order language for programming avionics embedded computers. It is being used in conjunction with MIL-STD-1750 computers on all of the programs I previously mentioned as well as on a number of ground based applications in support of MX, GPS, and the Satellite Control Facility.

As more users come on line, improved support software for J73 is in great demand and the Language Control Facility here at ASD has a key role to play. I note that the Embedded Computer Resource Program Office is developing a compiler for the VAX computers. This should help to provide a relatively low cost stand-alone development facility.

In the past, it has always been less expensive to retarget the J73 compiler than to buy a machine for which an operating compiler already existed. There was, of course, a delay of 12 to 24 months to get the retargeting accomplished, and then there was the problem of residual errors. I think we may have finally reached the point in this case where it is cheaper and faster to buy a computer for which software already exists than to retarget existing software.

MIL-STD-1760

MIL-STD-1760 defines the electrical interface between an aircraft and a store. That store can be a bomb, a missile, a rack, a fuel tank or a pod of any type. This standard is the newest of our avionics architecture standards and is currently only one third of what it will eventually be. The current standard defines the wires for the two connectors associated with the electrical interface, but does not define all of the logical behavior of a loaded store. The connector design has been selected and a draft specification is available, but we have not yet definitized the fiber optic option that was left open in the first version of the standard.

MIL-STD-1760 is being used on AMRAAM, the Multiple Stores Ejector Rack, the 30-mm gun pod and WASP, and will be used on the Conventional Standoff Weapon, the Common Strategic Rotary Launcher, and the Advanced Cruise Missile. With a connector change or an adapter, both MRASM and LANTIRN will interface with aircraft equipped with the MIL-STD-1760 interface.

NATO is actively supporting the use of this standard since it will allow the NATO nations to buy and use US weapons, and also to sell compatible weapons to the US. We are installing MIL-STD-1760 as part of the F-16 MSIP program and we are looking for the best way to install it on the rest of our combat aircraft.

The pre-existence of MIL-STD-1760 on an aircraft should significantly reduce the cost of integrating a new weapon with that aircraft. It will also eliminate the three to four year delay that we currently experience in modification of aircraft to carry and employ new weapons.

How Do We Know We're Doing the Right Thing?

We are doing very well in the implementation of these four interface standards, but there are some important questions to ask:

First -- how do we know that interface standardization is the right way to go?

and second -- how do we know which interfaces to standardize?

I claim that the best designs maintain the independence of their functional requirements. That is -- when part of the requirement changes, only part of the design changes. Of course this necessitates that the requirement be stated in such a way that it identifies the minimum set of independent specifications which bound the acceptable solutions. A system architecture must allow the design to meet its requirements. So, if you will agree that a set of interfaces implies an architecture, then I challenge you to find a more appropriate set of interfaces than those that define the natural boundaries between functions!

Our four avionics interface standards do define natural boundaries between functions:

JOVIAL is the interface between the programmer and the software.

MIL-STD-1750 is the interface between the software and the computer hardware,

MIL-STD-1553 is the interface between the hardware subsystems, and

MIL-STD-1760 is the electrical interface between the aircraft and a store.

We are firmly set on continuing to standardize interfaces, and in the process we both need and want industry involvement.

Commercial Standards

We would like to adopt or adapt commercial standards when we can - for many reasons. We adopted some in the automatic test equipment area when the MATE program selected the IEEE-488 General Purpose Interface Bus and the ATLAS language for writing test programs.

We are also willing to consider changes to uniquely military interface standards to make them commercially viable. This might be something for you to think about as we develop new standards for a high speed multiplex bus and VHSIC packaging for subsystems.

New Standards

Speaking of new standards, I need to tell you where we are going with some specifics on the high speed multiplex data bus, our transition to Ada, the future of NEBULA, and our feeling about MIL-STD-1679.

High Speed Multiplex Bus

It has been clear for over two years - since the Scientific Advisory Board made its recommendations on the new bomber - that we need a multiplex bus that is much faster than MIL-STD-1553.

The PAVE PILLAR program was directed over a year ago to work with the SAE AE-9 subcommittee to produce a standard. Two VHSIC contractors have add-on contracts to look at the problem. The AE-9 high speed bus subcommittee has had a meeting or two, and there is some thinking going on -- but it is time to really get serious about building a new standard. Come on gentlemen - let's get with it!

JOVIAL/Ada Transition

Our plan for transitioning from JOVIAL to Ada involves a staged effort. We intend to take a lesson from our history with JOVIAL and assess the maturity of the production compilers and language support tools in a laboratory environment before we insist that industry use them for schedule critical full scale development programs.

I should point out that there are two steps between assessing the maturity of compilers in a laboratory environment and insisting that industry use them for schedule critical full scale development programs.

These steps are:

First - parallel operational system developments, where we identify schedule critical full scale development efforts that are being done in a mature language (such as J73 or FORTRAN) and fund parallel developments of the same software in Ada from completion of the A-spec to the start of preliminary system tests. This type of effort will get both Air Force and contractor experience using Ada on full scale development programs without putting those programs at risk.

and second - allowing programs to volunteer to use Ada where the program office and contractor feel comfortable with Ada; the tools are sufficiently mature; and the risks are low.

We are looking for opportunities to try out Ada in a wide variety of different functional areas so that we can get a feel for what changes or additions should be made to our Ada introduction program. Early applications of Ada to realtime systems, command and control, parallel processing and fault tolerant systems will be useful steps on the way to a policy of universal application.

In the interim, there is a set of rules for JOVIAL developed by Aerospace Corporation which should ease the conversion to Ada if that proves to be desirable. We also have begun to use Ada as a design language to facilitate later conversion.

NEBULA

As for NEBULA, I don't think we see any near term requirement for a 32-bit avionics processor. But, when one does arise we intend to use the NEBULA architecture just as we have used the MIL-STD-1750 architecture: as a language transparent basis for competition and technology insertion.

MIL-STD On Software Development (MIL-STD-1679)

Our approach to the use of MIL-STD-1679 in the software development process is somewhat different. We intend to use it! You will start seeing it in the requirements that we lay on in our contracts, but you will see it tailored to the needs of the individual program. It is not an interface standard, it is a process standard--and as such it is more readily adjusted to meet special case conditions.

Automatic Test Equipment Situation is Similar

As an aside, I would like to say a few words about our standardization efforts in the automatic test equipment arena.

Some of you may be familiar with the Modular Automatic Test Equipment (or MATE) program. It represents standardization in a somewhat different dimension than the avionics I have been talking about, yet the details of the standardization are quite similar--MIL-STD-1750A computers, JOVIAL and ATLAS software, the IEEE-488 interface bus, a standard user interface, technology transparency and increased competition.

But MATE is important for another reason. MATE will allow us to define and require a level of test equipment compatibility that has not been possible in the past. We will be able to mix F^3 avionics equipment from multiple vendors at the depot and run them all across a set of test equipment that is minimally different from what would be required to test a single vendor's equipment.

In the past, competition has been reduced in second round procurements of F^3 avionics where the Air Force has purchased test equipment from the winner in round one. In round two, there was an entry cost for any new vendor equal to the cost of the test equipment for his system.

With MATE we think that we will be able to reduce and perhaps eliminate this cost by appropriate testability and interface requirements written into the F^3 specification.

New Avionics Support Options

If we can make this work, then it opens a whole new range of avionics support options that the logisticians have not been willing to sign up to in the past.

For instance, we could mix F^3 equipment from multiple vendors in the field, buying spares instead of intermediate level support equipment and rewarding the vendor of the most reliable system by buying more systems from him.

We might plan on a shorter life cycle for some equipment and buy lifetime maintenance from the vendor, or insist that the factory test equipment conform to the MATE guides and make the test equipment deliverable to the depot after some long warranty period.

MATE is an important piece of our overall standardization program, and it will play a role in opening up new options for supporting the avionics of the future.

What Should Industry be Doing?

Before I step down off my soapbox I want to provide some further guidance to those I haven't put to sleep.

For those of you who have not been involved with our standardization activities -- we welcome and encourage your participation, both here at the conference and at the users group meetings for those standards that affect you.

Every company wants their product to be the standard. Work with us, and with the rest of the industry participants -- you have an opportunity to explain to a very compete ... your ... exactly why your design is the best. Your produ... can be the ...

If you don't have a product yet -- I woul... come and bring your very best engineers to the users group meet... And when you get home, I would strongly recommend that you build your marketing strategy for the Air Force around the use of the standards in high quality products.

The standards are here, they will be around for as long as they are useful. We intend to insist on their use in products t... we buy from you.

We need and want your inputs on <u>how</u> the standards should evolve and what <u>new</u> standards will be needed.

The progress that is being made is <u>very</u> impressive. Keep up the good work.

Major General Jasper A. Welch Jr. is assistant deputy chief of staff for research, development and acquisition, Headquarters U.S. Air Force, Washington, D.C.

General Welch was born Jan. 5, 1931, in Baton Rouge, La., and graduated from University High School. He earned a bachelor of science degree in physics, magna cum laude, from Louisiana State University, Baton Rouge, in 1952. He earned a master of science degree in 1954 and a doctorate in physics in 1958 from the University of California, Berkeley. He was a distinguished military graduate of the Reserve Officers Training Corps program at Louisiana State University and commissioned a second lieutenant in the Regular Air Force in May 1952. General Welch is also a distinguished graduate of the Industrial College of the Armed Forces, Fort Lesley J. McNair, Washington, D.C.

His first assignment was in August 1952 with the Armed Forces Special Weapons program at Sandia Base, N. M., as a student and then instructor in the early atomic energy program. From September 1953 to June 1954, he attended the University of California, Berkeley, under the Air Force Institute of Technology program.

After a short tour of duty at the Air Force Special Weapons Center at Kirtland Air Force Base, N.M., General Welch was assigned in November 1954 to the Lawrence Livermore Laboratory of the Atomic Energy Commission at Livermore, Calif. He led an experimental nuclear weapon design team which developed the basic design concept still used in most operational systems. While assigned to Livermore Laboratory he completed his doctoral studies under Nobel laureate Luis W. Alvarez.

He returned to the Air Force Special Weapons Center in September 1957 as chief of the Theoretical Physics Branch and scientific adviser to the director of research. For the next five post-Sputnik years, he led a team to determine the effects of nuclear weapons detonated in the upper reaches of the atmosphere and in space. During this period General Welch was invited to present the results of his pioneering scientific research in space physics to the National Academy of Sciences and several international symposia. In August 1962 he was assigned to the RAND Corporation where he was involved with the strategic and politico-military implications of ballistic missiles and space systems.

In January 1963 General Welch was assigned to Headquarters Air Force Systems Command, Andrews Air Force Base, Md. During 1963 he served as a member of the politico-military staff of Project Forecast, a major assessment of the future of the Air Force. In 1964 he was the Air Force member of [illegible] appointed by the secretary of Defense to oversee the first [illegible] analysis of resource allocation among strategic offensive and defensive forces. He moved to the West Coast [illegible] of Air Force Systems Command at Los Angeles in [illegible]. There he directed studies that led to the initiation of more than a dozen major Air Force programs, including the [illegible] cruise missiles.

In August 1969 he entered the Industrial College of the Armed Forces. Following graduation in July 1970 he transferred to Air Force headquarters as chief military analyst for the assistant chief of staff for studies and analysis.

From 1962 to 1969, General Welch was also a consultant to private industry on the peaceful uses of nuclear explosions for the production of petroleum. He served as a consultant to the Air Force Scientific Advisory Board, Defense Science Board, National Aeronautics and Space Administration, North Atlantic Treaty Organization Advisory Group on Aerospace Research and Development, and the President's Science Advisory Committee. He is author or co-author of many published technical reports, studies, journal articles and a book, "The Atomic Theory of Gas Dynamics." He is a member of the National Academy of Engineering, the American Physical Society, the American Geophysical Union and the Council on Foreign Relations.

He was assigned to the Office of the Secretary of Defense in September 1971 and served as assistant director of defense research and engineering for strategic systems review and analysis. In 1972 General Welch became staff director of a high level panel appointed by the secretary of Defense to examine certain key aspects of strategic policy. He served for a brief period in 1973 as special assistant to the deputy chief of staff, research and development at Headquarters U.S. Air Force.

The general resumed his work on strategic policy in August 1973 as special assistant to the assistant to the secretary of Defense for atomic energy. In October 1974 he returned to Headquarters U.S. Air Force as assistant for strategic initiatives to the deputy chief of staff for plans and operations. In September 1975 General Welch began a four-year tour of duty as assistant chief of staff for studies and analysis, also at Air Force headquarters.

In November 1979 he became defense policy coordinator for the National Security Council. After a short assignment as special assistant to the chief of staff, Headquarters U.S. Air Force, General Welch assumed his present duties in June 1981.

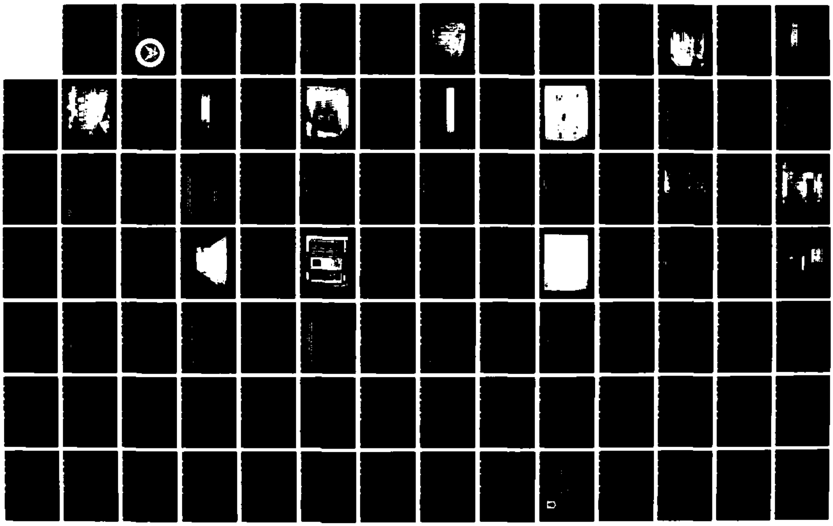
His military decorations and awards include the Distinguished Service Medal, Defense Superior Service Medal, Legion of Merit with two oak leaf clusters, Air Force Commendation Medal and Air Force Outstanding Unit Award ribbon.

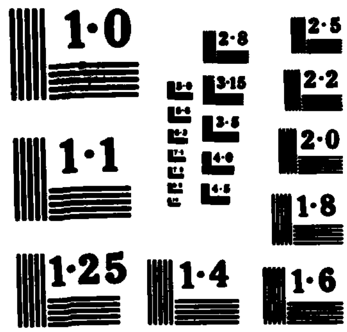
He was promoted to major general Feb. 6, 1976, with date of rank June 16, [illegible]

General Welch is married to the former Carroll Wright of New Orleans. They have three children: Jasper, Carroll and Brent, a student at the University of Colorado.

AD-A145 697 PROCEEDINGS PAPERS OF THE AFSC (AIR FORCE SYSTEMS COMMAND) AVIONICS STAND..(U) AERONAUTICAL SYSTEMS DIV WRIGHT-PATTERSON AFB OH DIRECTORATE O.. 2/6

UNCLASSIFIED C A PORUBCANSKY NOV 82 F/G 1/3 NL




1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8
3·15
3·5
4·0
4·5

EMBEDDED COMPUTER STANDARDIZATION IN THE NAVY

AN OVERVIEW

REAR ADMIRAL
WAYNE D. BODENSTEINER
DEPUTY CHIEF FOR
MATERIAL ACQUISITION
30 NOVEMBER 1982

Abstract

EMBEDDED COMPUTER STANDARDIZATION IN THE NAVY - AN OVERVIEW

The Navy has been practicing standardization of embedded computers and peripherals for over 20 years beginning with the Naval Tactical Data System in 1959. The basic attributes of Navy embedded computer standardization are:

- families of logistically identical high performance computers
- instruction set upward compatibility of existing computers with newer generation computers
- provision of standard Navy support software for the users of these computers

This talk will review the rationale behind the Navy approach to computer standardization. The primary factors being:

- minimization of at-sea hardware logistics supportability and costs
- graceful system degradation for casualty control by computer-to-computer backup
- preservation/reuse of a very large body of existing Navy unique software

VG #2 OUTLINE

GOOD MORNING. I APPRECIATE THE OPPORTUNITY YOU HAVE GIVEN ME TO ADDRESS THIS CONFERENCE. WE IN THE NAVY, VERY STRONGLY SUPPORT EMBEDDED COMPUTER STANDARDIZATION EFFORTS AND ARE COMMITTED TO ACQUIRING QUALITY EMBEDDED COMPUTER RESOURCES PURSUED IN ACCORDANCE WITH A "RATIONAL" POLICY. IN THE NEXT 20 MINUTES I WILL REVIEW THE NAVY'S COMPELLING REASONS FOR HAVING A STANDARDIZATION PROGRAM AND REVIEW THE HISTORY AND STATUS OF OUR ONGOING DEVELOPEMENT OF NEW STANDARDS. FINALLY, I WILL SPEND A FEW MINUTES ON WHAT WE SEE AS THE FUTURE OF NAVY STANDARDIZATION BEYOND THE CURRENT DEVELOPMENT, AS WELL AS WHAT WE NEED TO DETERMINE REGARDING THAT FUTURE.

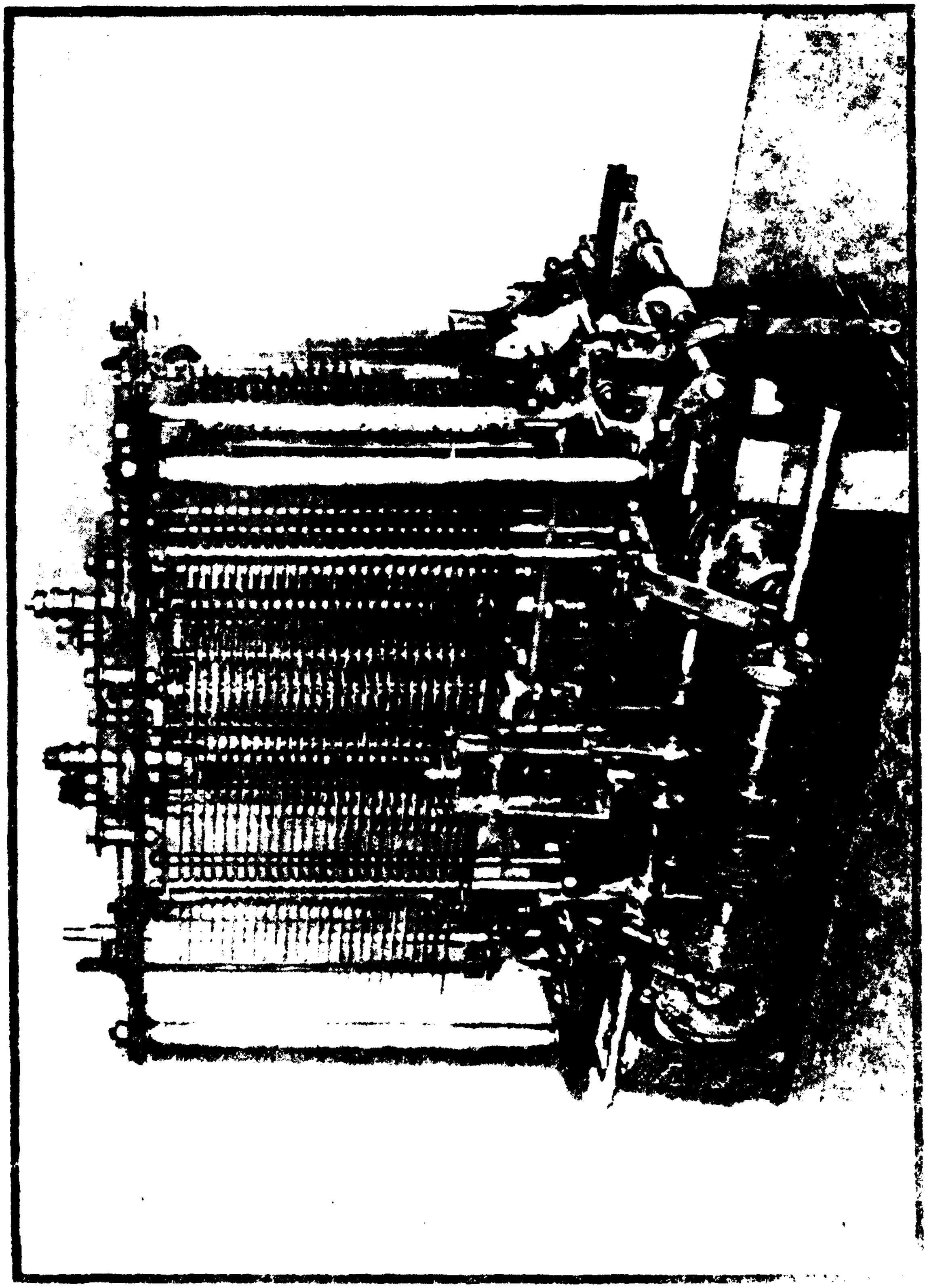
OUTLINE

- HISTORY
- STATUS
- WHY STANDARDIZATION?
- CURRENT DEVELOPMENTS
- THE FUTURE

VG #3 BABBAGE COMPUTER

CONTRARY TO POPULAR BELIEF, THIS IS NOT THE CURRENT NAVY "STANDARD COMPUTER". THE NAVY HAS NOT BEEN ABLE TO GET THAT MUCH OF A STANDARDIZATION HEAD START ON THE OTHER TWO SERVICES. INSTEAD, THIS MAY WELL BE RECOGNIZED AS THE FIRST REAL COMPUTER EVER ASSEMBLED. WHAT YOU SEE HERE IS ACTUALLY A RE-CREATION OF THE ORIGINAL. IT CAN BE SEEN IN THE SCIENCE MUSEUM, LONDON, ENGLAND. IT WAS PUT TOGETHER FROM THE DRAWINGS OF THE ANALYTICAL ENGINE LEFT BY CHARLES BABBAGE AND WAS CONSTRUCTED PARTLY BY HIS SON AND PARTLY BY THE FIRM OF R. W. MUNRO. IT IS THE "MILL" AND PRINTING MECHANISM CAPABLE OF PERFORMING THE ARITHMETICAL OPERATIONS AND PRINTING THE RESULT TO 29 PLACES. IT IS COMMONLY BELIEVED THAT THE WEALTHY DAUGHTER OF THE POET, LORD BYRON (AUGUSTA ADA BYRON) NOT ONLY PROVIDED SPONSORSHIP, BUT ALSO, AS AN ACCOMPLISED MATHEMATICIAN, PROGRAMMED THIS MACHINE FOR CHARLES BABBAGE. ADVANCEMENT OF SCIENCE DID NOT APPEAR TO BE THEIR PRIME MOTIVATION AS THE TWO SEEMED MOST INTERESTED IN USING THE MACHINE TO DEVISE A SURE-FIRE HORSE-BETTING SCHEME.

RUMOR HAS IT THAT ADA, ALSO KNOWN AS THE COUNTESS OF LOVELACE, ENTERED INTO A ROMANTIC ARRANGEMENT WITH MR. BABBAGE, A SITUATION APPARENTLY UNKNOWN TO LADY BABBAGE. AUGUSTA ADA BYRON ONLY LIVED TO THE TENDER AGE OF 37, DYING OF WHAT WAS PERCEIVED TO BE CANCER IN 1852. WE CERTAINLY HOPE THAT THE NEW DOD LANGUAGE STANDARD WILL NOT SUFFER SUCH AN UNTIMELY EARLY DEMISE.



VG #4 TACTICAL EMBEDDED COMPUTERS IN THE NAVY TODAY

TODAY, THE NAVY HAS 6,000 STANDARD MILITARIZED EMBEDDED COMPUTERS IN 450 TYPES OF TACTICAL SYSTEMS AND SUBSYSTEMS ENCOMPASSING SURFACE, SUBSURFACE, AIRBORNE AND SHORE-BASED PLATFORMS. THIS NUMBER IS EXPECTED TO GROW TO AROUND 27,000 BY 1990.

WE HAVE FIVE TYPES OF STANDARD COMPUTERS NOW IN USE. THE CP-642 AND THE AN/UYK-7 ARE USED ONBOARD SHIPS IN LARGE MAINFRAME APPLICATIONS; THE AN/UYK-20 IS USED IN SHIP BOARD MINI-COMPUTER APPLICATIONS; THE AN/AYK-14, AN AIRBORNE MINI-COMPUTER; AND THE AN/UYS-1, A PROGRAMMABLE SIGNAL PROCESSOR FOR ANTI-SUBMARINE WARFARE APPLICATIONS.

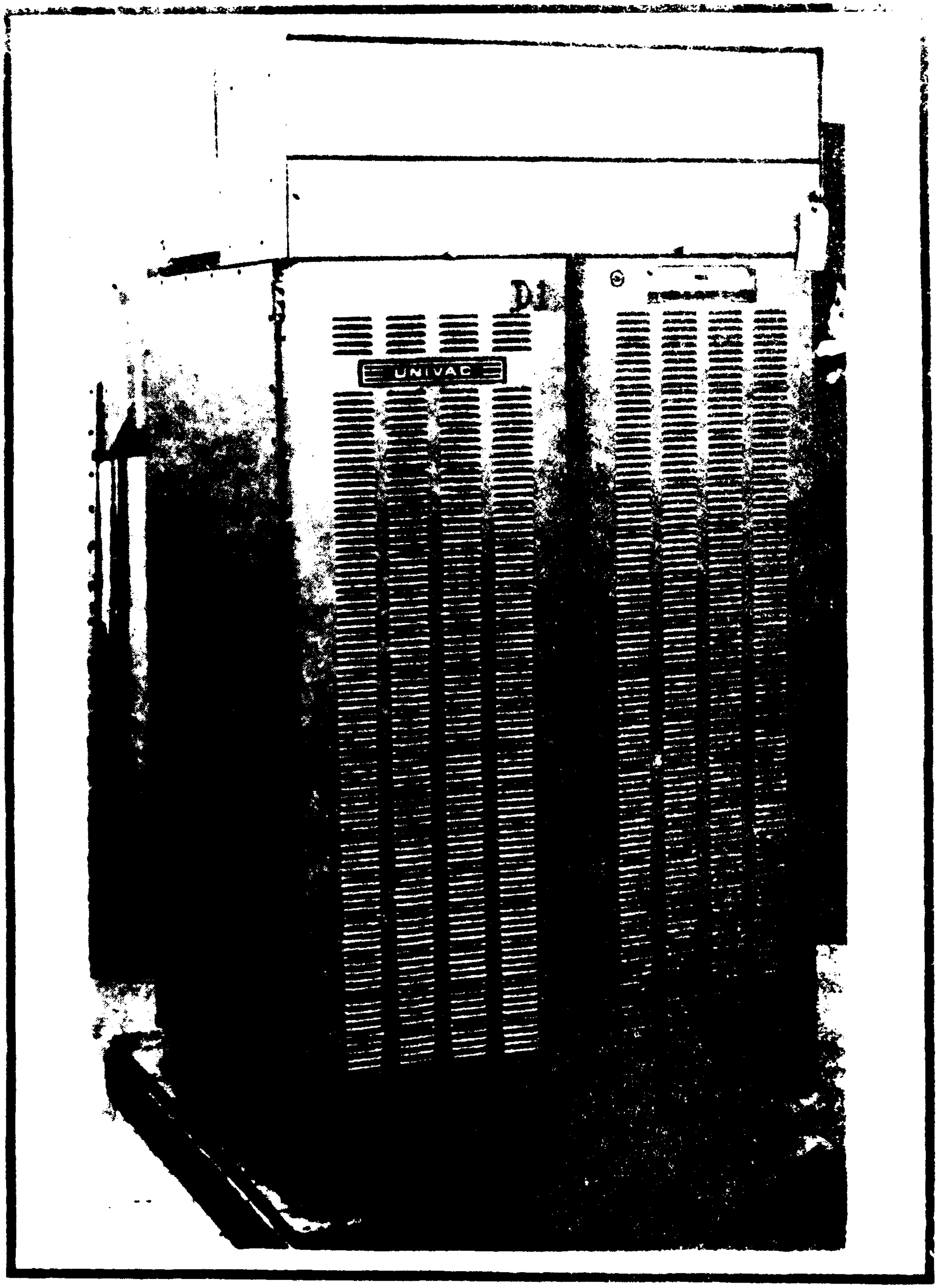
IT IS ESTIMATED THAT THESE COMPUTERS USE OVER 50 MILLION LINES OF COMPUTER PROGRAM CODE AND REQUIRE OVER 2 MILLION LINES OF STANDARD SUPPORT SOFTWARE. NUMBERED AMONG THE USERS OF SOME OF THESE COMPUTERS INCLUDE THE UNITED STATES AIR FORCE, ARMY, MARINE CORPS, COAST GUARD, AND THE NAVIES OF SEVEN OTHER NATIONS.

TACTICAL EMBEDDED COMPUTERS IN TODAY'S NAVY

- ESTIMATE 450 TYPES OF USER SYSTEMS ON SHIP, SUBMARINE, AIR, AND SHORE BASED PLATFORMS
- 6000 STANDARD COMPUTERS IN INVENTORY
- FIVE TYPES OF STANDARD COMPUTERS NOW IN USE
- 50 MILLION LINES OF APPLICATION PROGRAMS SOURCE CODE
- 2 MILLION LINES OF STANDARD SUPPORT SOFTWARE
- USERS INCLUDE
  - NAVY
  - ARMY
  - USAF
  - USMC
  - USCG
  - THE NAVIES OF SEVEN OTHER NATIONS

VG #5 PHOTO OF CP/642

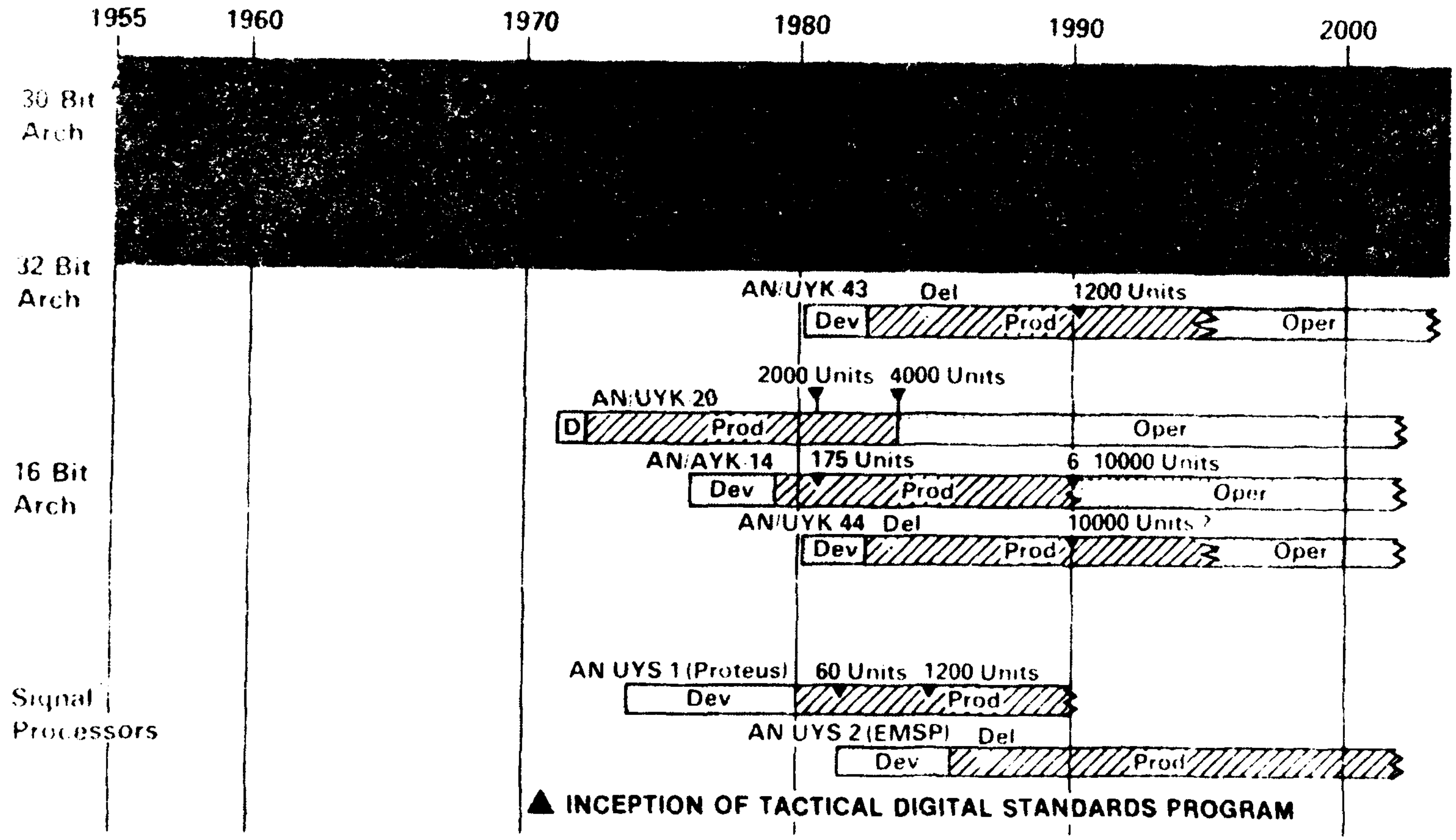
THE NAVAL TACTICAL DATA SYSTEM WAS FIELDED IN 1959, AND POWERED BY THE WORLD'S FIRST LARGE-SCALE, SOLID STATE, MILITARIZED DIGITAL COMPUTER--THE CP 642. IN ALL, 363 OF THESE 30-BIT WORD, 32K MEMORY MACHINES WERE DELIVERED; AFTER 20 YEARS, ALL ARE STILL IN ACTIVE SERVICE. NEEDLESS TO SAY, EVEN WITH AN EXTENDED CORE MEMORY UNIT, THEY ARE RUNNING AT THE LIMITS OF THEIR SPEED, MEMORY CAPACITY, AND SUPPORTABILITY.


UNIVAC

VG #6 LINEAGE OF NAVY PROCESSORS

THIS SHOWS THE HISTORY OF COMPUTER STANDARDIZATION IN THE NAVY. TO MEET INCREASED NTDS PROCESSING LOADS THE AN/UYK-7 COMPUTER WAS DEVELOPED AND BEGAN PRODUCTION DELIVERIES IN 1971 AS A SUCCESSOR TO THE CP 642--IN NEW INSTALLATIONS. IT HAD SUBSTANTIALLY INCREASED SPEED AND MEMORY CAPACITY.

LINEAGE OF NAVY PROCESSORS



VG #7 PHOTO OF AN/UYK-7

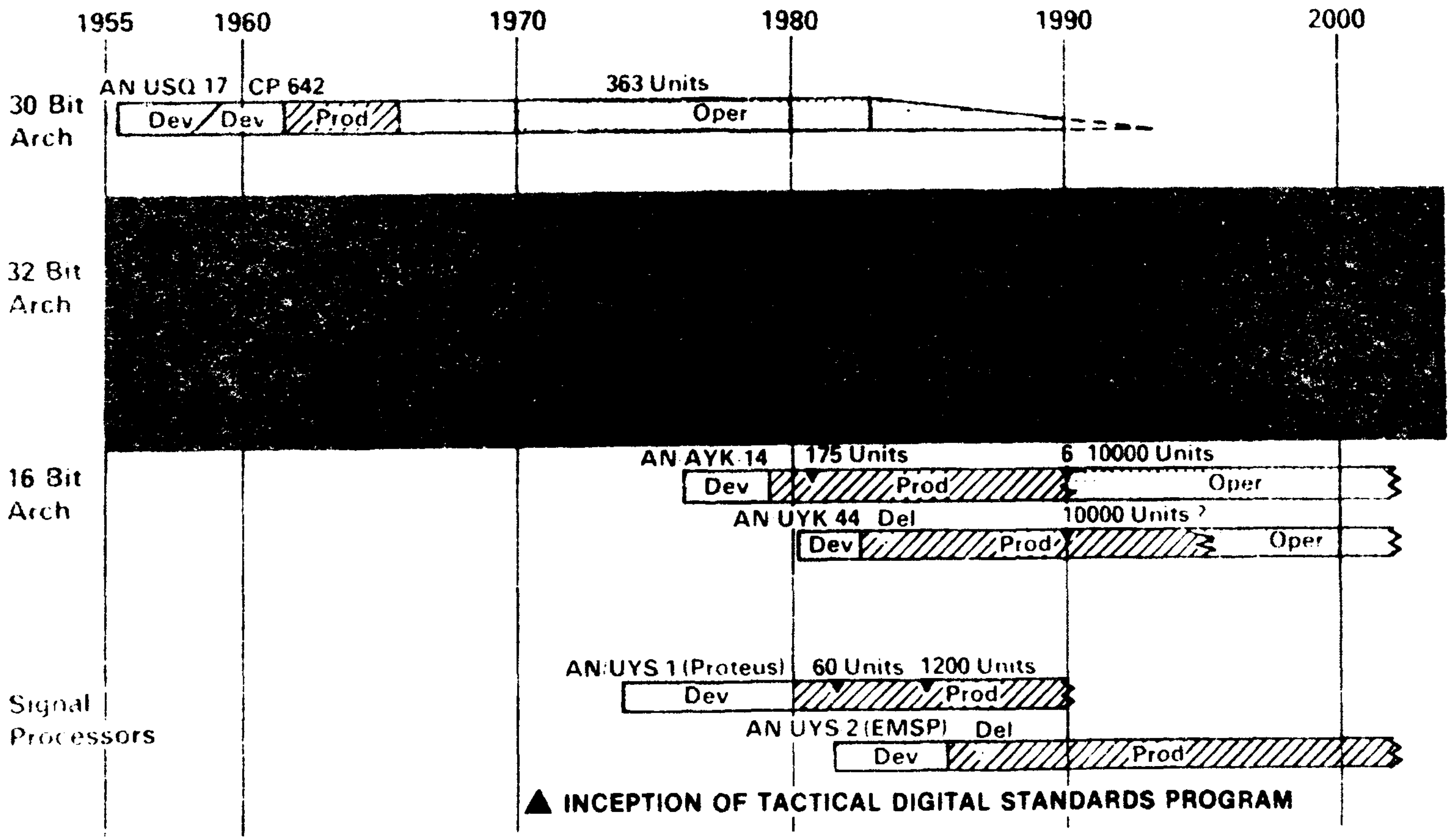
THIS SHOWS TWO-4 BAY UYK-7 ASSEMBLIES. ALMOST 1500 UYK-7 "BAYS" ARE NOW IN INVENTORY. THE INVENTORY OF UYK-7'S IS EXPECTED TO GROW TO 2,100 BY 1985; HOWEVER, THESE MACHINES ARE ALSO BEGINNING TO EXPERIENCE MEMORY SATURATION IN SOME APPLICATIONS.

VG #8 THE LINEAGE OF NAVY PROCESSORS

THE AN/UYK-43 IS NOW UNDER DEVELOPMENT AS A SUCCESSOR TO THE AN/UYK-7, FOR LARGE MAIN-FRAME APPLICATIONS.

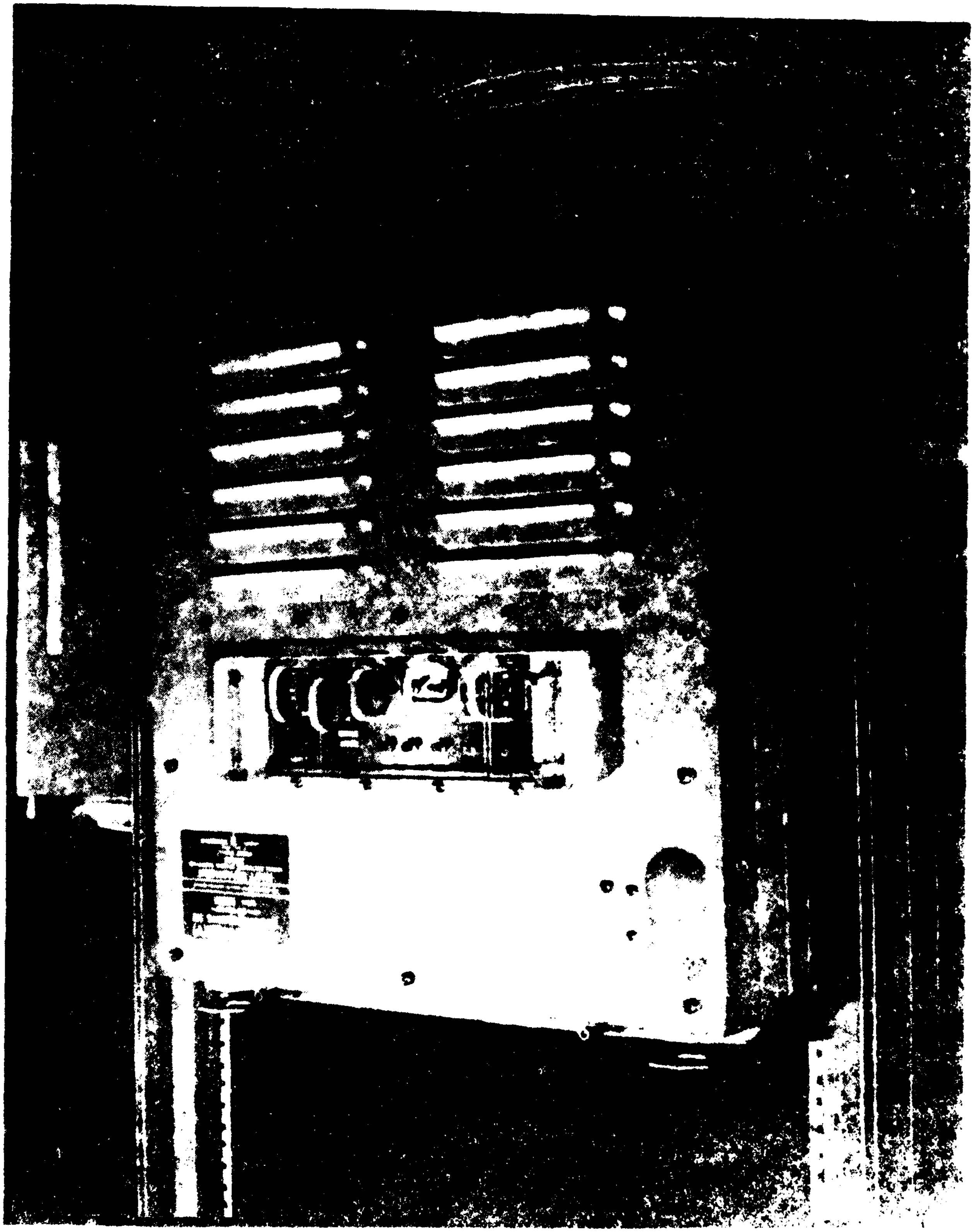
REGARDING SMALLER, OR MINI-COMPUTERS, A PROLIFERATION OF NUMEROUS TYPES OF THESE SYSTEMS BECAME EVIDENT IN 1971, AND IN 1973 THE STANDARD NAVY TACTICAL MINICOMPUTER, THE AN-UYK-20, WAS ACQUIRED COMPETITIVELY BY THE NAVAL ELECTRONIC SYSTEMS COMMAND IN ORDER TO STEM THE TIDE.

LINEAGE OF NAVY PROCESSORS



VG #9 PHOTO OF AN/UYK-20

THIS IS THE AN/UYK-20. TODAY 2,500 OF THESE 16-BIT MACHINES ARE IN USE IN 205 TYPES OF SYSTEMS, AND THE INVENTORY IS EXPECTED TO GROW TO 4,000 BY 1985. THIS MACHINE IS ALSO EXPERIENCING SATURATION OF ITS 64K WORD MEMORY IN SOME APPLICATIONS.

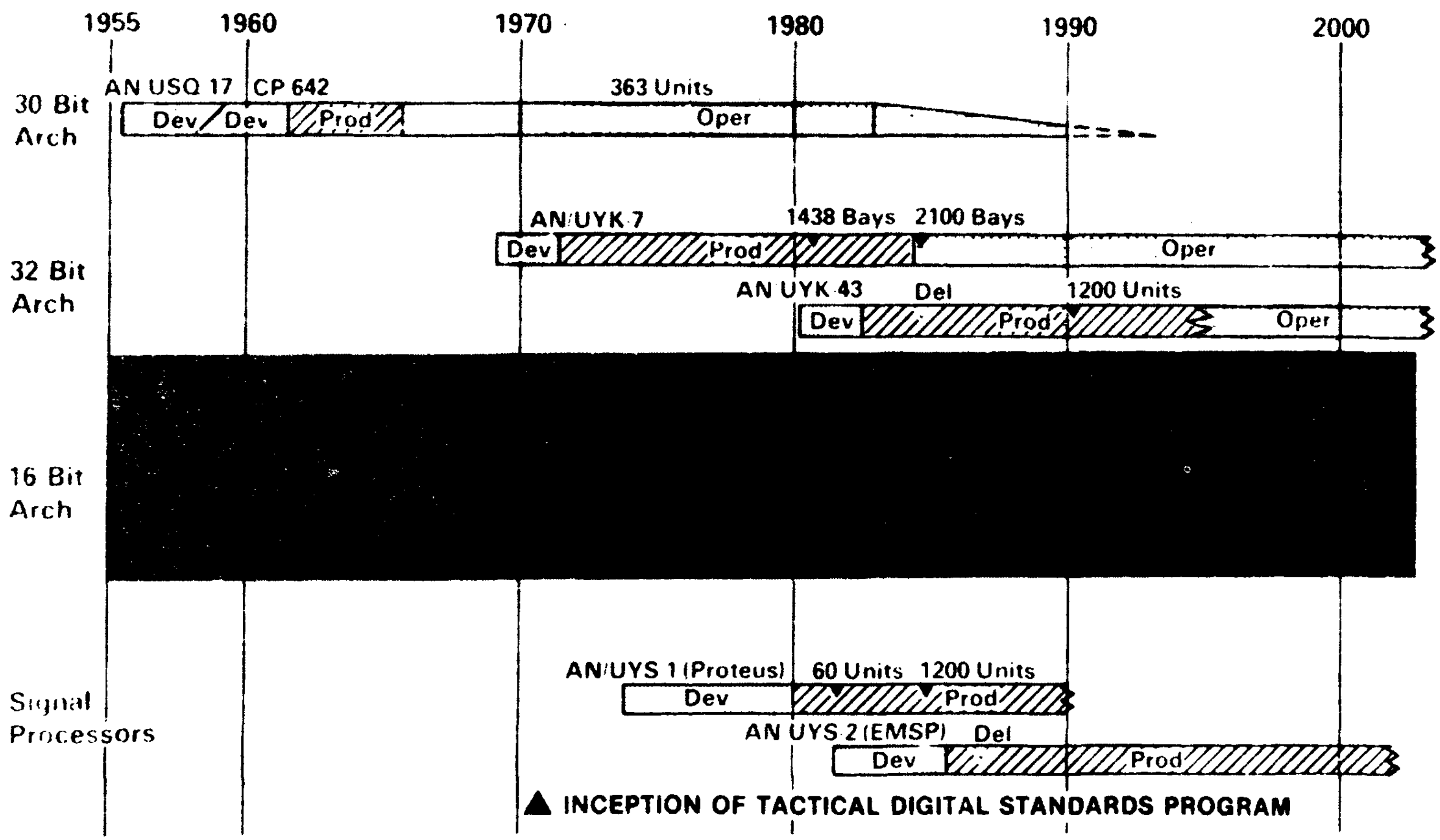


VG #10 LINEAGE OF NAVY PROCESSORS

THE AN/UYK-44 IS NOW UNDER DEVELOPMENT AS THE SUCCESSOR TO THE AN/UYK-20.

THE AN/AYK-14 COMPUTER IS A 16-BIT, 512K GENERAL PURPOSE MACHINE INTENDED FOR AIRBORNE, GROUND-MOBILE, AND SOME SHIPBOARD APPLICATIONS. THE SYSTEM'S BASELINE WAS ESTABLISHED IN DECEMBER 1979; OVER 200 PRE-PRODUCTION UNITS HAVE BEEN DELIVERED SINCE COMPETITIVE CONTRACT AWARD IN SEPTEMBER 1976. APPROXIMATELY 10,000 OF THESE COMPUTERS ARE ANTICIPATED TO BE IN USE BY 1990.

LINEAGE OF NAVY PROCESSORS



VG #11 PHOTO OF AN/UYS-1

THE AN/UYS-1 ADVANCED SIGNAL PROCESSOR IS PROGRAMMABLE AND DESIGNED TO SATISFY A RANGE OF ACOUSTIC APPLICATIONS. THE HARDWARE BASELINE WAS ESTABLISHED IN AUGUST 1979. TO DATE, 71 PRE-PRODUCTION UNITS HAVE BEEN DELIVERED TO USER SYSTEMS. THE ESTIMATED TOTAL PRODUCTION OF THE UYS-1 IS 1500 UNITS.

THE UYS-2 IS NOW UNDER DEVELOPMENT AS THE SUCCESSOR TO THE UYS-1.

VG #12 THE NAVY MUST ENFORCE STANDARDS

THERE ARE A NUMBER OF COMPELLING REASONS FOR EMBEDDED COMPUTER STANDARDIZATION IN THE NAVY, INCLUDING OPERATIONAL READINESS ECONOMIC REASONS, AND CERTAIN INTANGIBLE FACTORS (SUCH AS "MAINTAINER" FAMILIARITY - THAT IS, THE PEOPLE WHO RUN IT SHOULD KNOW HOW TO FIX IT) - HOWEVER, A CRITICAL DRIVING FACTOR IS THE COST OF AT SEA LOGISTICS AND MAINTENANCE. BECAUSE IT IS SO CRITICAL, I WILL SPEND A FEW MINUTES ON JUST THE COST OF AT-SEA SPARE PARTS FOR A FLEET WIDE INVENTORY OF DIGITAL COMPUTERS.

THE NAVY MUST ENFORCE STANDARDS

- TO IMPROVE OPERATIONAL READINESS
- TO REDUCE COST

VG #13 LOGISTICS, MAINTENANCE, AND TRAINING STRONGLY AFFECT OPERATIONAL AVAILABILITY

THE STANDARD CRITERION FOR MOST NAVY SYSTEMS IS SELF-SUFFICIENCY FOR A 90 DAY MISSION, AND THE OPERATIONAL AVAILABILITY (A_o) GOAL FOR STANDARD COMPUTERS DURING SUCH A MISSION IS 0.999 AVAILABILITY. A_o, IN TURN, IS DEFINED AS SHOWN, WHERE MTBCMc IS "MEAN DOWN TIME BETWEEN CRITICAL CORRECTIVE MAINTENANCE ACTIONS" AND MDT IS "MEAN DOWN TIME." MDT IS MADE UP OF A NUMBER OF FACTORS, INCLUDING:

- MEAN TIME TO DIAGNOSE
- MEAN TIME TO REPAIR, AND
- MEAN TIME TO REQUISITION SPARES

MEAN TIME TO REQUISITION SPARES TURNS OUT TO BE VERY CRITICAL. IT CAN VARY FROM A FEW MINUTES, (IF THE SPARES ARE ON-BOARD) TO AROUND 500 HOURS (IF THEY AREN'T) THE ACTUAL TIME CAN OBVIOUSLY VARY WIDELY AROUND THIS MEAN, DEPENDING ON WHERE IN THE WORLD THE SHIP IS.

THE MESSAGE IS, HIGH OPERATIONAL AVAILABILITY IS NOT ONLY A FUNCTION OF RELIABILITY BUT ALSO (AND PERHAPS EVEN MORE IMPORTANT) HOW FAST A FAILED UNIT CAN BE RESTORED TO OPERATION. AND THIS IS STRONGLY DEPENDENT ON THE AVAILABILITY OF ON-BOARD SPARES.

LOGISTICS, MAINTENANCE, AND TRAINING STRONGLY AFFECT OPERATIONAL AVAILABILITY

- MUST HAVE SELF SUFFICIENCY FOR A 90 DAY MISSION
- COMPUTER OPERATIONAL AVAILABILITY (A_O) GOAL FOR A 90 DAY MISSION IS 0.999

$$A_O = \frac{MTBCM_C}{MTBCM_C + MDT}$$

- HIGH A_O IS ACHIEVED BY A COMBINATION OF:
  - — HIGH MTBF
  - — LOW MTTR
- LOW MTTR IS ACHIEVED BY:
  - — KNOWLEDGEABLE MAINTAINERS
  - — ON-BOARD SPARES

VG #14 ON-BOARD SPARES AFFECT Ao AND COST

THIS SLIDE WILL GIVE YOU AN IDEA OF HOW ON-BOARD SPARES AFFECT A_o AND COST.

IF WE LOOK AT THE NEW NAVY STANDARD MAIN FRAME COMPUTER, THE AN/UYK-43, THE PROBABILITY OF ITS RUNNING WITHOUT FAILURE DURING A 90-DAY MISSION WORKS OUT TO 70%. OBVIOUSLY, REPAIR PREVISIONS MUST BE MADE TO ATTAIN A REASONABLE OPERATIONAL AVAILABILITY.

THE TABLE SHOWS THAT WHEN NO-SPARES ARE CARRIED ON-BOARD, REPAIR WILL TAKE AROUND 500 HOURS AND THE OPERATIONAL AVAILABILITY WILL ONLY BE 0.879. A 50% AVAILABILITY OF ON-BOARD SPARES WILL GIVE A 10% IMPROVEMENT IN MISSION AVAILABILITY -AT A PRICE OF $81,000. AT THE HIGH END OF SPARING, ONE SPARE FOR EACH LOGISTICALLY REPLACEABLE UNIT WILL BRING THE COMPUTER ALMOST TO THE .999 AVAILABILITY GOAL, AT A COST IN SPARE PARTS CARRIED ON-BOARD APPROACHING ONE-HALF OF THE ORIGINAL COST OF THE COMPUTER.

ON-BOARD SPARES AFFECT A_o AND COST

THE 6000 HOUR RELIABILITY OF THE AN/UYK-43 AND 44 YIELDS A 0.30 CHANCE OF ITS FAILING DURING A 90 DAY MISSION

	NO ON-BOARD SPARES	50% OF LRU TYPES ON-BOARD	ONE SPARE FOR EACH LRU TYPE ON-BOARD
OPERATIONAL AVAILABILITY	0.879	0.968	0.998
SPARED QUANTITY	0	27	54
ON-BOARD SPARES COST	0	$81,250	$162,500

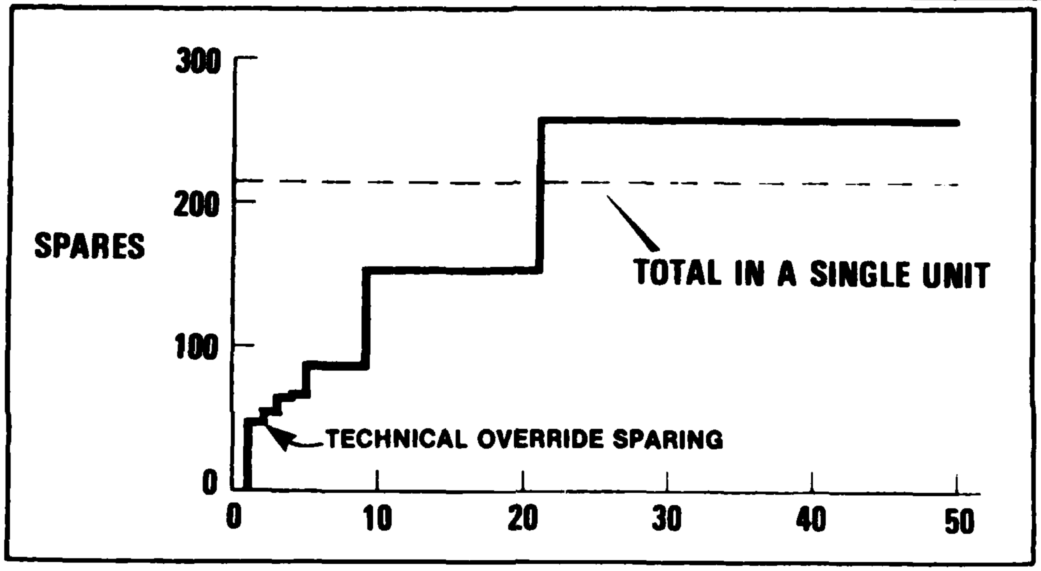
VG #15 ON-BOARD SPARES INVESTMENT COSTS FOR ONE COMPUTER FAMILY

OF COURSE, THE NEXT QUESTION IS WHAT IS THE EFFECT ON REQUIRED SPARES AND THEIR COST, IF MORE THAN ONE COMPUTER OF THE SAME TYPE IS ON-BOARD? TO ANSWER THIS, WE MUST FIRST SHOW THE COST OF ON-BOARD SPARES FOR A SINGLE COMPUTER SYSTEM.

THIS SLIDE SHOWS A SPARING CURVE REPRESENTATIVE OF THE AN/UYK-43 AND -44. AT THE LOW END, WITH ONE COMPUTER ON BOARD, APPROXIMATELY 50% OF THE LRU TYPES ARE AUTHORIZED TO BE CARRIED ON BOARD. THESE SPARES ARE TERMED "TECHNICAL OVERRIDE SPARES." THEY ARE THE MINIMUM NUMBER OF SPARES TO ENSURE THAT IN THE EVENT OF ANY FAILURE, COMPUTER PROGRAMS CAN BE LOADED AND THE MAINTENANCE PROCESSOR CAN BE MADE TO RUN SO THAT THE COMPUTER HAS A CAPABILITY TO DIAGNOSE ITSELF. THIS BASIC SET OF TECHNICAL OVERRIDE SPARES MUST BE ON BOARD, WHETHER THE SHIP CARRIES ONLY ONE OR 50 OF THE SAME TYPE OF COMPUTER. AN ADDITIONAL UNIQUE SET OF TECHNICAL OVERRIDE SPARES IS NEEDED TO SUPPORT EACH DIFFERENT COMPUTER TYPE CARRIED ON BOARD.

THE COST OF THESE MINIMAL SPARES (SHOWN ON THE BOTTOM OF THE VIEWGRAPH) IS BASED ON A TEN YEAR PROJECTION OF USE OF THE AN/UYK-43 MAINFRAME AND AN/UYK-44 MINICOMPUTER. USING THE POPULATION ASSUMPTIONS SHOWN, WE SEE THAT AT A VERY MINIMUM - ASSUMING ONLY TWO BASIC TYPES OF COMPUTERS, FROM THE SAME FAMILY - THE ABSOLUTE MINIMUM FLEET WIDE TECHNICAL OVERRIDE SPARING COSTS ABOUT 30 MILLION DOLLARS. OBVIOUSLY, AS THE NUMBER OF UNIQUE TYPES OF COMPUTERS INCREASES, THE COST ASSOCIATED WITH THE MINIMUM SPARING REQUIREMENTS INCREASES PROPORTIONALLY.

ON BOARD SPARES INVESTMENT COST FOR ONE COMPUTER FAMILY



SPARING ASSUMPTIONS

- 1500 NEW MAIN FRAMES ON 250 SHIPS
- 20,000 NEW MINI'S ON 500 SHIPS
- 50% SPARING COSTS APPROX. ¼ COST OF ONE COMPUTER

MINIMUM FLEET SPARING COST TO ACHIEVE A 0.97 OPERATIONAL AVAILABILITY WITH ONE COMPUTER FAMILY:

250 MAIN FRAMES × $80K = $20 MILLION
500 MINI'S × $20K = $10 MILLION
TOTAL $30 MILLION

VG #16 OTHER COSTS OF EACH UNIQUE COMPUTER FAMILY

SO FAR, WE HAVE TREATED ONLY THE COST OF FLEET-WIDE COMPUTER SPARING, AND WE HAVE SEEN THAT COMPUTER PROLIFERATION WOULD LEVY AN UNACCEPTABLE ECONOMIC BURDEN ON THE NAVY. THIS VIEWGRAPH SHOWS ADDITIONAL COSTS, BOTH TANGIBLE AND INTANGIBLE, FOR EACH COMPUTER FAMILY PUT TO SEA. WITH REGARD TO "LARGE VOLUME PRODUCTION COST BREAK" MENTIONED IN THIS TABLE, IT WAS THE CONCENSUS OF THE JOINT LOGISTICS COMMANDERS' MONTEREY SOFTWARE WORKSHOP (HELD IN JUNE 1981) THAT INDUSTRY GENERALLY PREFERS SERVICE COMPUTER STANDARDIZATION, BECAUSE OF THE LARGER PRODUCTION VOLUMES OF SINGLE TYPE COMPUTERS INHERENT IN STANDARDIZATION. AGAIN, MAINTAINER FAMILIARITY IS A KEY ELEMENT IN MAINTAINING HIGH SYSTEM OPERATIONAL AVAILABILITY.

OTHER COSTS OF EACH UNIQUE COMPUTER FAMILY

- SUPPORT SOFTWARE
- CONFIGURATION MANAGEMENT SYSTEM
- IN-SERVICE ENGINEERING ORGANIZATION
- TRAINING COURSES
- TRAINING TIME OFF-THE-JOB (1 TO 15 WEEKS)
- ADMIN COSTS OF EACH SUPPLY CHAIN ($3000/LRU)
- TECH MANUALS
- ON-BOARD SPARES STOWAGE AND REPAIR FACILITIES
- ACQUISITION COST OVERHEAD PER COMPUTER FAMILY
- DEPOT REPAIR FACILITIES
- FIELD CHANGE ACQUISITION OVERHEAD
- LOSS OF LARGE VOLUME PRODUCTION COST BREAK

INTANGIBLES

- MAINTAINER FAMILIARITY
- LOSS OF SYSTEM-TO-SYSTEM BACKUP
- ON-BOARD SUPPLY CONFUSION FACTOR

VG #17 TACTICAL COMPUTER DEVELOPMENT PROGRAM GUIDANCE

NOW WE WILL BRIEFLY REVIEW NEW NAVY STANDARD COMPUTERS IN DEVELOPMENT. UYK-20 AND UYK-7 LIMITATIONS, MENTIONED PREVIOUSLY, HIGHLIGHTED A REQUIREMENT EITHER TO COMMENCE A PROGRAM TO UPGRADE THESE MACHINES OR TO DEVELOP MORE CAPABLE SUCCESSORS.

UPON REVIEW OF THESE ALTERNATIVES, CONGRESS DIRECTED COMPETITIVE DEVELOPMENT OF SUCCESSOR COMPUTERS. REQUIREMENTS DRIVING THE DEVELOPMENT OF THE SUCCESSOR COMPUTER, INCLUDED:

A. THE CONTINUING NEED FOR A LARGE MAINFRAME COMPUTER FOR CENTRALIZED COMBAT DIRECTION, WEAPONS CONTROL, AND COMMAND AND CONTROL APPLICATIONS. THIS DEVELOPMENT IS THE 32-BIT WORD AN/UYK-43.

B. THE CONTINUING NEED FOR A STANDARD MINICOMPUTER AND A STANDARD "EMBEDDABLE" MICROPROCESSOR. THESE TWO REQUIREMENTS ARE BEING ADDRESSED BY EXTENSIONS OF THE STANDARD 16 BIT ARCHITECTURE AND WILL BE FULFILLED BY THE AN/UYK-44 DEVELOPMENT.

C. THE NEED FOR A SUCCESSOR SIGNAL PROCESSOR TO THE AN/UYS-1 ADVANCED SIGNAL PROCESSOR.

D. CONGRESS ALSO DICTATED THAT RELIABILITY, MAINTAINABILITY AND AVAILABILITY IMPROVEMENTS ARE TO HAVE PRECEDENCE OVER PERFORMANCE IMPROVEMENTS. (THUS FAR, SPECIFICATIONS ARE BEING MET, OR EXCEEDED, IN BOTH AREAS WITH NO COMPROMISE NEEDED.)

E. BOTH THE AN/UYK-43 AND 44 COMPUTER DEVELOPMENTS WERE TO BE SOFTWARE UPWARD COMPATIBLE WITH THEIR PREDECESSOR COMPUTERS. THIS IS BEING ACCOMPLISHED THROUGH INCLUSION OF PREDECESSOR INSTRUCTION SET ARCHITECTURES (ISA'S) AS SUBSETS OF THE NEW, EXPANDED ISA'S.

TACTICAL COMPUTER DEVELOPMENT PROGRAM GUIDANCE

- CONTINUING NEED FOR LARGE "MAIN FRAME" COMPUTERS (I.E. AN/UYK-43)
- CONTINUING NEED FOR "MINICOMPUTERS" (I.E. AN/UYK-44)
- RAPIDLY EXPANDING EMBEDDED "MICROPROCESSOR" APPLICATIONS (I.E. AN/UYK-44)
- EXPANDING PROGRAMMABLE SIGNAL PROCESSOR APPLICATIONS (I.E. AN/UYS-2)
- IMPROVE RELIABILITY AND MAINTAINABILITY
- ENSURE SOFTWARE TRANSPORTABILITY
  - — COMMON INSTRUCTION SET ARCHITECTURE (ISA)
  - — COMMON HIGH ORDER LANGUAGE (HOL)
- LOGISTICALLY IDENTICAL COMPUTER MODULES FOR LOWEST LIFE CYCLE SUPPORT COSTS

VG #18 AN/UYK-43 NAVY EMBEDDED COMPUTER SYSTEMS

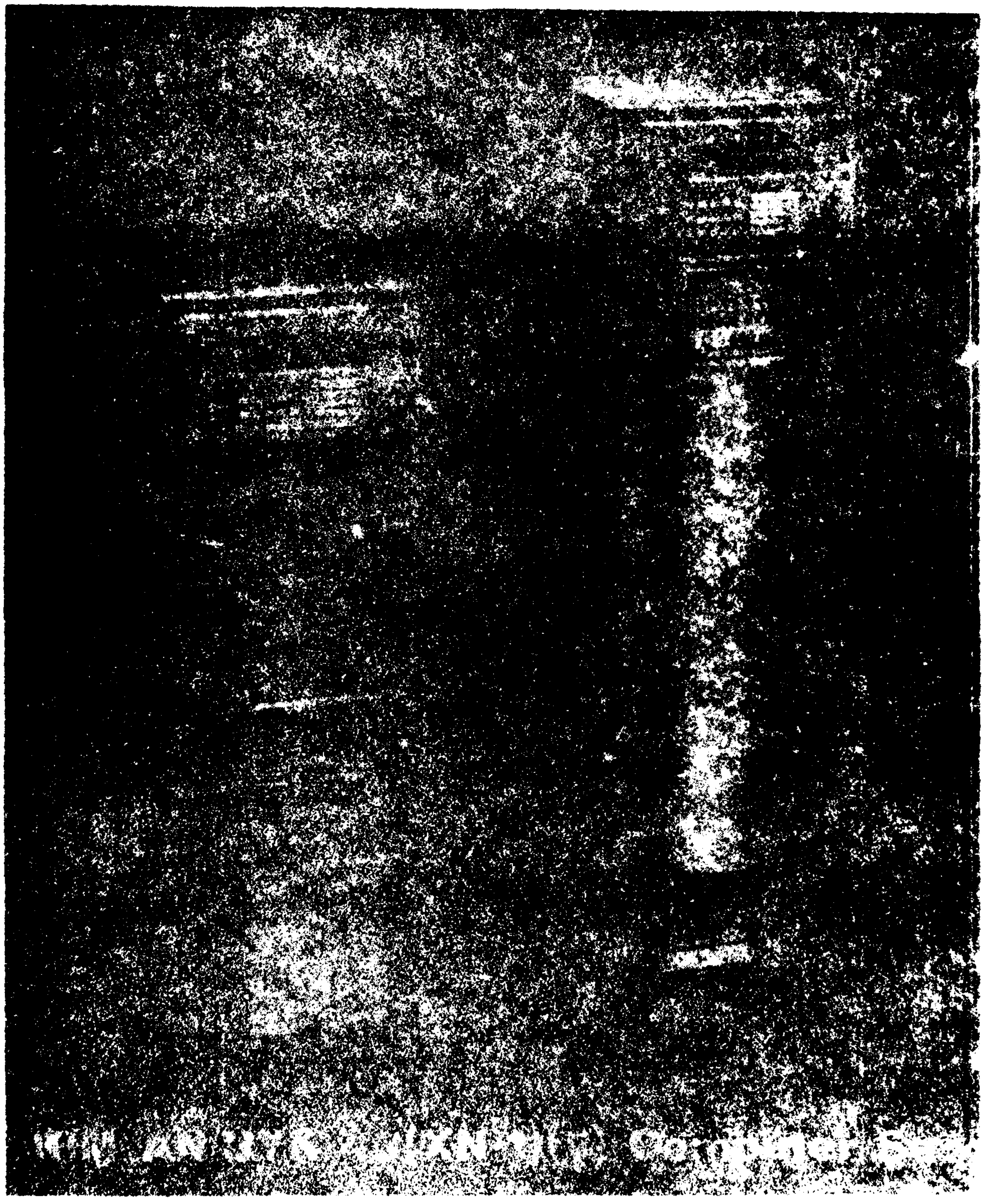
THE AN/UYK-43 NAVY EMBEDDED COMPUTER SYSTEM WILL BE THE NEXT GENERATION STANDARD NAVY 32 BIT COMPUTER SUPERSEDING THE AN/UYK-7. THE AN/UYK-43 WILL HAVE UP TO NINE TIMES THE PROCESSING SPEED COMPARED TO THE UYK-7 WITH SOFTWARE COMPATIBILITY SUCH THAT UYK-7 PROGRAMS WILL RUN ON THE UYK-43. IT WILL BE PRODUCED IN TWO BASIC ENCLOSURES WITH SOME PERFORMANCE DIFFERENCES BUT BOTH WILL BE CAPABLE OF PASSING THROUGH A 25 INCH SUBMARINE HATCH. THE PROGRAM IS CURRENTLY IN COMPETITIVE DEVELOPMENT BETWEEN IBM AND SPERRY-UNIVAC WITH PRODUCTION DELIVERY SCHEDULED FOR THE END OF FY-84.

AN/UYK-43 NAVY EMBEDDED COMPUTER SYSTEM (NECS)

- SUCCESSOR TO AN/UYK-7 FOR SURFACE SHIP AND SUBMARINE APPLICATIONS WITH HIGH PERFORMANCE REQUIREMENTS
- SOFTWARE UPWARD COMPATIBLE WITH AN/UYK-7
- LANGUAGE SUPPORT — CMS-2L (Ada IN 1986)
- MODULAR ARCHITECTURE, PERFORMANCE OPTIONS, COMPUTER INTERCONNECTION SYSTEM FOR ADAPTABILITY TO SYSTEM REQUIREMENTS
- FIRST USER — DDG-51 PROGRAM

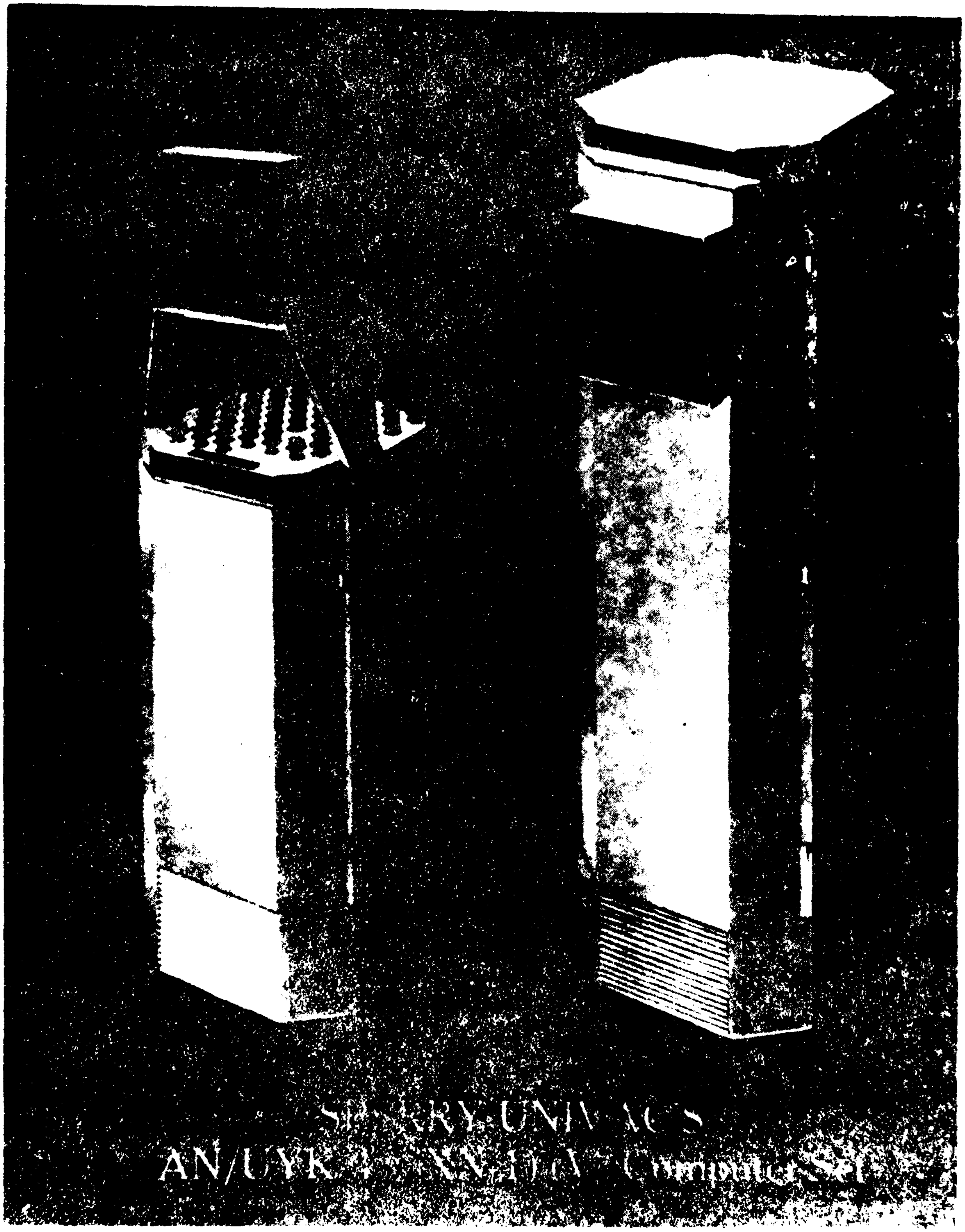
VG #19 PHOTO OF IBM'S AN/UYK-43

THIS SLIDE IS A PHOTO OF IBM'S VERSION OF THE AN/UYK-43. ENCLOSURE A IS ON THE LEFT AND ENCLOSURE B IS ON THE RIGHT.



VG #20 PHOTO OF SPERRY'S AN/UYK-43

AND THIS SLIDE DEPICTS SPERRY UNIVAC'S VERSION OF THE AN/UYK-43 AGAIN ENLCOSURE A IS ON THE LEFT AND ENCLOSURE B IS ON THE RIGHT.



VG #21 AN/UYK-44

THE AN/UYK-44 MILITARIZED RECONFIGURABLE PROCESSOR (MRP) AND COMPUTER (MRC) PROGRAM IS DESIGNED TO PROVIDE GREATLY INCREASED PERFORMANCE IN A FORM, FIT, AND FUNCTION SUCCESSOR TO THE AN/UYK-20. THIS NEW CONFIGURATION IS NEEDED TO ACCOMMODATE COMPLEX MULTIFUNCTION WEAPONS SYSTEMS, COMMUNICATIONS SYSTEMS, COMMAND AND CONTROL SYSTEMS, AND INTELLIGENCE SYSTEMS IN A DISTRIBUTED ENVIRONMENT. IT IS SOFTWARE COMPATIBLE WITH BOTH THE AN/UYK-20 AND AN/AYK-14. THE AN/UYK-44 WILL ALSO USE EXISTING SOFTWARE TOOLS INCLUDING CMS-2M AND SPL/I CROSS-COMPILERS. THE AW/UYK-44 ELECTRONIC MODULES (CARD SETS) WILL BE USED EITHER IN THE COMPLETE COMPUTER (MRC) OR WILL BE DIRECTLY EMBEDDED AS A PROCESSOR CARD SET (MRP) IN USER EQUIPMENTS.

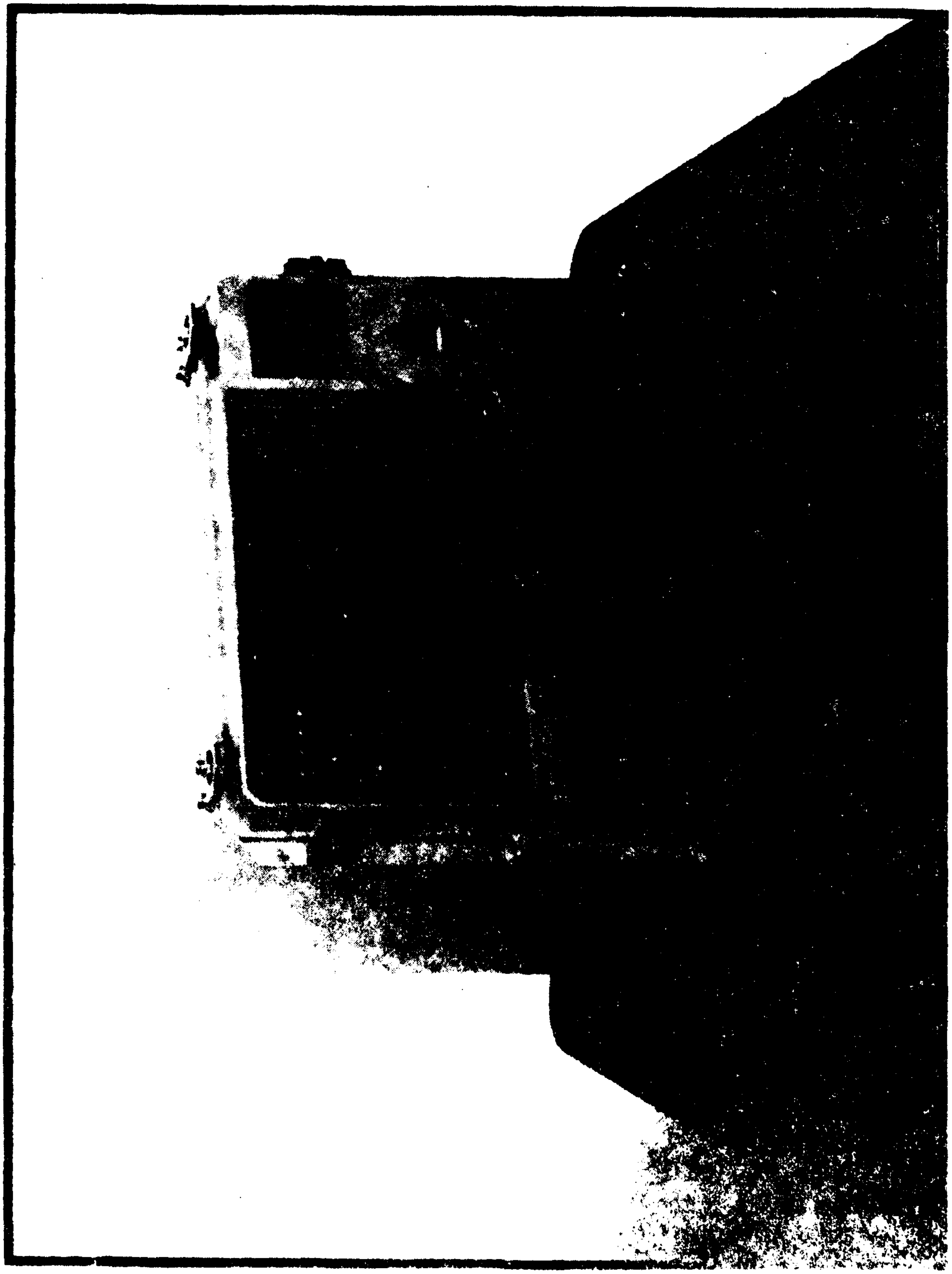
AN/UYK-44

- SOFTWARE UPWARD COMPATIBLE WITH AN/UYK-20 (EXTENDED UYK-20 ISA)
- LANGUAGE SUPPORT — CMS-2M AND SPL/I (Ada IN 1986 — WILL HAVE Ada ORIENTED INSTRUCTIONS)
- MILITARIZED RECONFIGURABLE PROCESSOR (MRP)
  - BUILT IN COMPONENT OF EQUIPMENT (E.G., SENSORS, DISPLAYS, LAUNCHERS) TO FACILITATE FUNCTIONALLY DISTRIBUTED PROCESSING
  - AVAILABLE AS STANDARD ELECTRONIC MODULES (SEM) OR CHIP SET
- MILITARIZED RECONFIGURABLE COMPUTER (MRC)
  - PACKAGED MILITARIZED COMPUTER USING MRP SEMs
  - SUCCESSOR TO AN/UYK-20 FOR SURFACE SHIP AND SUBMARINE APPLICATIONS WITH LOW AND MEDIUM PERFORMANCE REQUIREMENTS

VG #22 PHOTO OF SPERRY'S AN/UYK-44

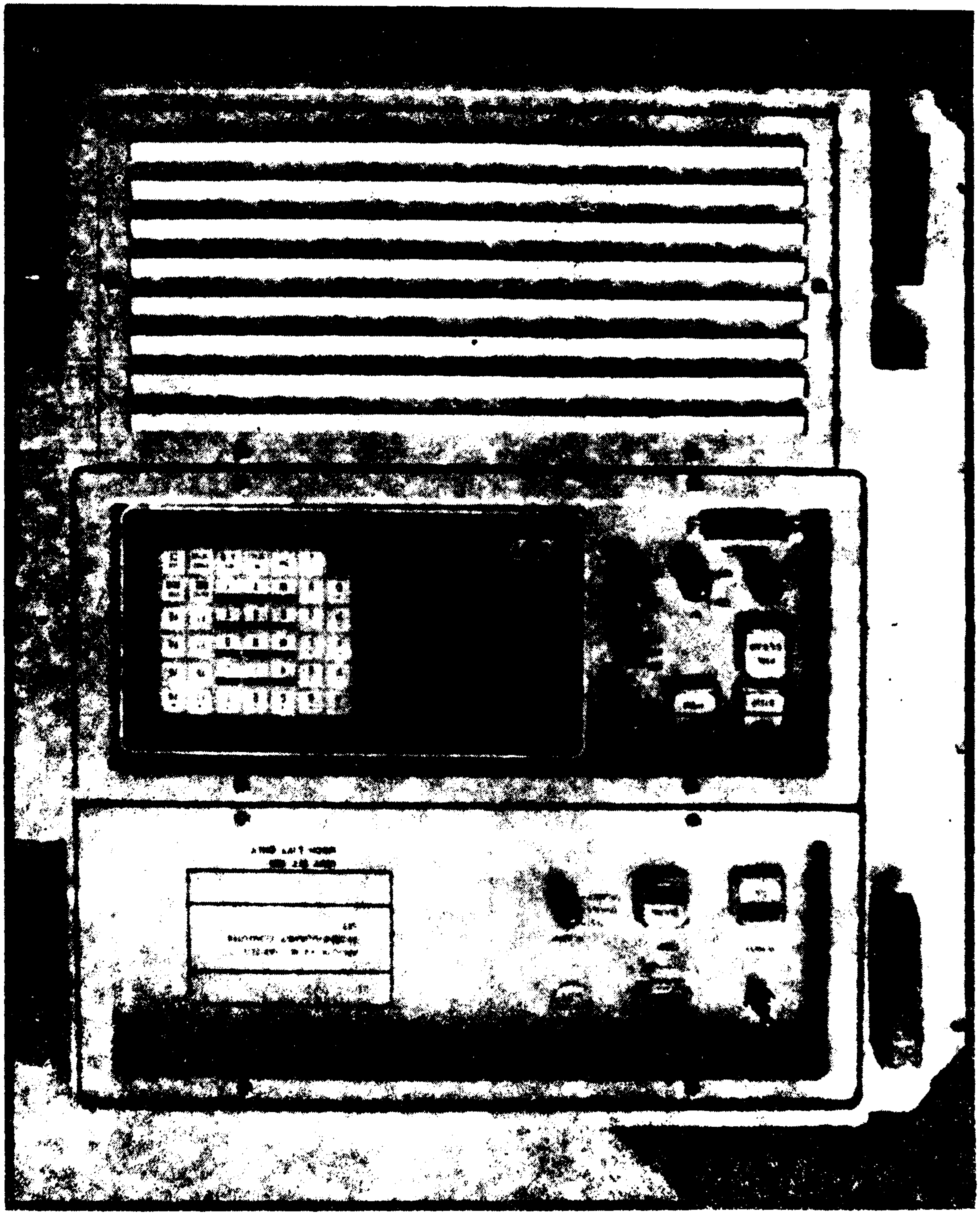
THE NEXT TWO SLIDES ARE PHOTOGRAPHS OF SPERRY UNIVAC AND IBM VERSIONS OF THE AN/UYK-44

FIRST SPERRY'S . . .



VG #23 PHOTO OF IBMS AN/UYK-44

AND IBM'S.



VG #24 AN/AYK-14

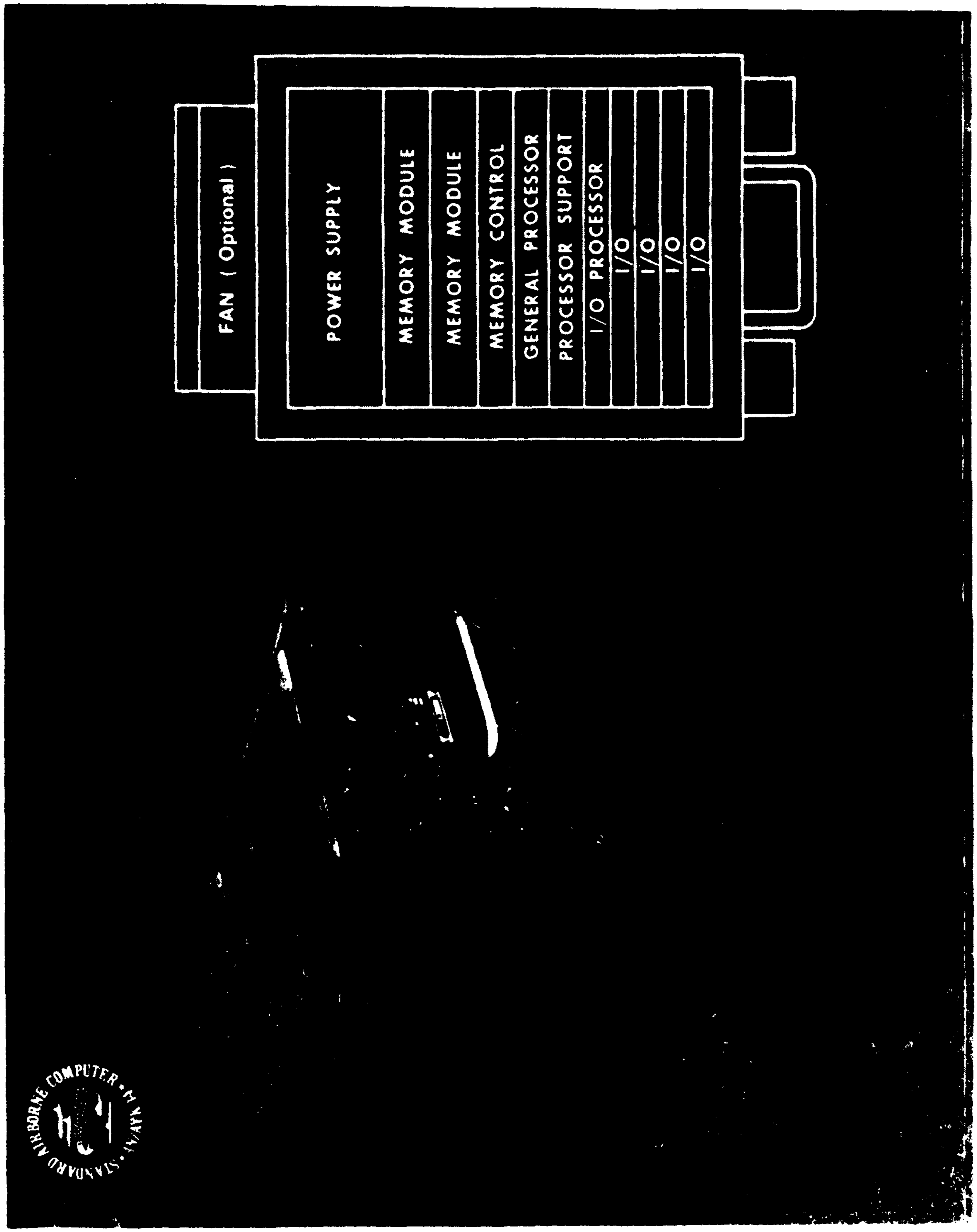
THE AN/AYK-14 BASELINE WAS ESTABLISHED IN DECEMBER 1979, AND OVER 200 COMPUTERS HAVE BEEN DELIVERED SINCE THEN. IT IS A GENERAL-PURPOSE, MODULARLY-DESIGNED COMPUTER WITH AN INSTRUCTION SET THAT IS A SUPERSET OF THE AN/UYK-20. THE STRINGENT SPACE, WEIGHT, POWER, AND FORM FACTORS PLACED ON AVIONICS COMPONENTS REQUIRE THAT IT HAVE A FLEXIBLE PACKAGING APPROACH. IN ADDITION TO FOUR BASIC CONFIGURATIONS, IT MAY BE RECONFIGURED TO MEET ALTERNATIVE REQUIREMENTS. THE MAJOR USES OF THE AN/AYK-14 INCLUDE F/A-18A, SH-60B (LAMPS), EA-6B, AV-8B, EP-3, E-2C, P-3C, AND BQM-111A (FIRERPAND RECOVERABLE DRONE). IN ADDITION, MAJOR NON-AIRBORNE USERS INCLUDE THE AIRBORNE LIGHTWEIGHT TORPEDO, THE AUTOMATIC CARRIER LANDING SYSTEM, AND SEVERAL OTHERS.

AN/AYK-14

- SOFTWARE UPWARD COMPATIBLE WITH AN/UYK-20
- LANGUAGE SUPPORT — CMS-2M
- MODULAR ARCHITECTURE WITH STRINGENT SPACE, WEIGHT, POWER, FORM REQUIREMENTS
- CURRENTLY PACKAGED IN FOUR CONFIGURATIONS
- CURRENTLY USED IN OR PLANNED FOR MAJOR AIRBORNE AND GROUND-MOBILE PROGRAMS

VG #25 PHOTO OF AN/AYK-14

THIS SLIDE SHOWS ONE CONFIGURATION OF THE AN/AYK-14.


FAN (Optional)
POWER SUPPLY
MEMORY MODULE
MEMORY MODULE
MEMORY CONTROL
GENERAL PROCESSOR
PROCESSOR SUPPORT
I/O PROCESSOR
I/O
I/O
I/O
I/O

VG #26 AN/UYS-2 ENHANCED MODULAR SIGNAL PROCESSOR (EMSP)

THE AN/UYS-2 WILL BE A HIGH TECHNOLOGY, HIGH THROUGHPUT PROGRAMMABLE SIGNAL PROCESSOR WITH A MODULAR ARCHITECTURE TO SUPPORT A VARIETY OF USER APPLICATIONS WITH VARYING THROUGHPUT REQUIREMENTS. IT WILL BE DEVELOPED TO MEET ENVIRONMENTAL SPECIFICATIONS FOR SUBMARINE, SURFACE SHIP, AND AIRBORNE PLATFORMS. THE AN/UYS-2 WILL BE DEVELOPED WITH FULL SUPPORT SOFTWARE BASED ON THE EXISTING NAVY STANDARD LANGUAGE, FOR SIGNAL PROCESSING (SPL/I). OUR PLAN INCORPORATES DEVELOPMENT OF A SUPPORT SOFTWARE ENVIRONMENT TO ALLOW GRACEFUL TRANSITION FROM THE CURRENT NAVY STANDARD THE AN/UYS-1, TO THE UYS-2.

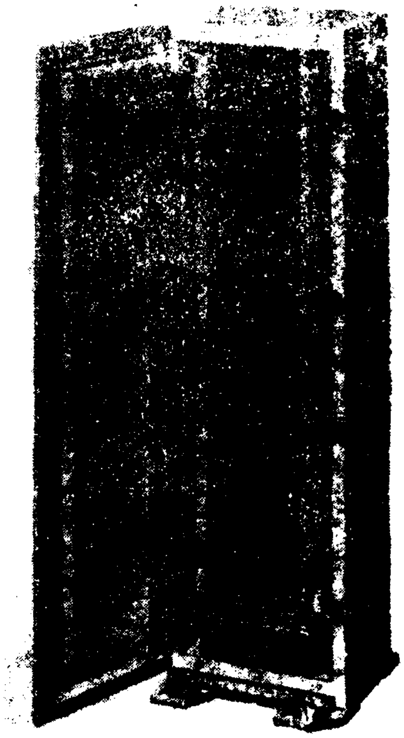
AN/UYS-2 ENHANCED MODULAR SIGNAL PROCESSOR (EMSP)

- SUCCESSOR TO AN/UYS-1 ADVANCED SIGNAL PROCESSOR FOR SURFACE SHIP, SUBMARINE, AND AIRBORNE SIGNAL PROCESSING APPLICATIONS
- SOFTWARE COMPATIBLE WITH AN/UYS-1 AT THE SPL/I HIGH ORDER LANGUAGE LEVEL
- LANGUAGE SUPPORT — SPL/I
- MODULAR ARCHITECTURE, PERFORMANCE OPTIONS, AND VARIETY OF ENCLOSURES FOR ADAPTABILITY TO SYSTEM REQUIREMENTS

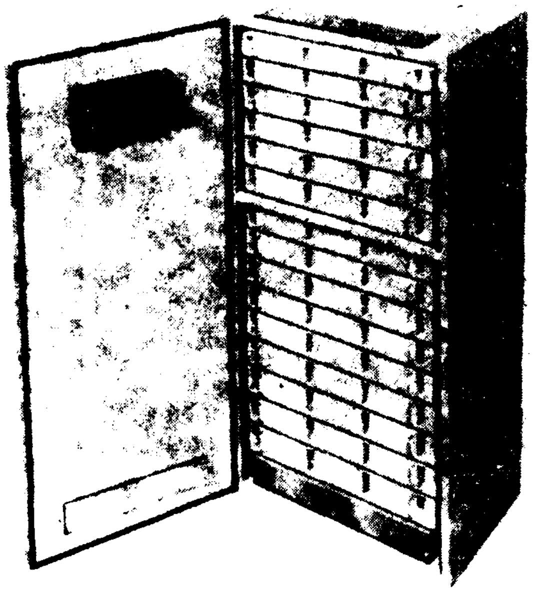
VG #27 PHOTO OF AN/UYS-2

HERE IS A PHOTO OF THE AN/UYS-2.

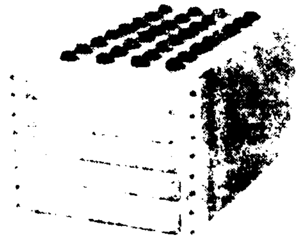
EMSP ENCLOSURES



Fixed Cage Cabinet
Water Cooled



Drawer Cage Cabinet
Air Cooled



Rack-Mountable
Enclosure

VG #28 NAVY EMBEDDED COMPUTER SOFTWARE

IN ADDITION TO OUR EFFORTS TOWARD STANDARDIZING HARDWARE, GREAT EMPHASIS IS BEING PLACED ON SOFTWARE ISSUES.

APPLICATION PROGRAMS ARE ONLY A PORTION OF THE SOFTWARE COST.

TO SUPPORT EXISTING SYSTEMS, THE NAVY HAS INVESTED $100 MILLION DOLLARS IN SUPPORT SOFTWARE. TO BETTER MANAGE SOFTWARE EXPENDITURES WE NEED TO CONTINUE TO EMPHASIZE STANDARD INSTRUCTION SETS, STANDARD LANGUAGES AND STANDARD SUPPORT SOFTWARE PROGRAMS.

NAVY EMBEDDED COMPUTER SUPPORT SOFTWARE

- APPLICATION PROGRAMS
- SUPPORT SOFTWARE INVESTMENTS
- INSTRUCTION SETS
- LANGUAGE
- SOFTWARE SUPPORT TOOLS

VG #29 NEED FOR A STANDARD NAVY INSTRUCTION SET ARCHITECTURE

IN ORDER TO CONTROL THE TOTAL COSTS OF DOING BUSINESS, STANDARDIZING INSTRUCTIONAL SET ARCHITECTURES IS ESSENTIAL. THESE ARE THE PRIMARY REASONS FOR AN ISA STANDARD.

TO SUPPORT COMPUTER CASUALTY BACKUP, SWITCHING ARRANGEMENTS MUST BE IN PLACE TO LOAD THE SAME MACHINE CODE FROM MASS STORAGE TO A DIFFERENT "FALL BACK" COMPUTER WITHIN SECONDS.

ALL SUPPORT SOFTWARE MUST BE REUSEABLE AND TRANSPORTABLE TO CONSERVE THE $100 MILLION INVESTMENT IN STANDARD SUPPORT SOFTWARE FOR STANDARD ISA MACHINES. WE CANNOT AFFORD TO RE-DEVELOP AT A COST OF $50 MILLION NON-RECURRING PLUS $10M ANNUAL MAINTENANCE FOR EACH ADDITIONAL ISA.

FINALLY, APPLICATIONS SOFTWARE MUST ALSO BE REUSEABLE AND TRANSPORTABLE, AGAIN TO REDUCE REDEVELOPMENT OF SYSTEMS.

NEED FOR A STANDARD NAVY INSTRUCTION SET ARCHITECTURE

- SHIPBOARD COMPUTER CASUALTY BACKUP
- STANDARD SUPPORT SOFTWARE REUSE/ TRANSPORTABILITY
- APPLICATIONS SOFTWARE REUSE/ TRANSPORTABILITY

VG #30 CURRENT SOFTWARE SUPPORT

THE NAVY CURRENTLY HAS TWO MACHINE TRANSPORTABLE SUPPORT SOFTWARE OR MTASS PROGRAMS. THE MTASS/M SUPPORTS THE 16-BIT INSTRUCTION SET ARCHITECTURE OF THE UYK-20, AYK-14, AND UYK-44 COMPUTERS. ITS COMPONENTS INCLUDE CMS-2M COMPILER, SIMULATION OF UYK-20 AND AYK-14, AND UYK-44, COMPILER AND UTILITIES. WRITTEN IN FORTRAN, IT EXECUTES ON MANY DIFFERENT HOST COMPUTERS AND IS INSTALLED AT 100 FACILITIES.

WITH SIMILAR CAPABILITIES, THE MTASS/L SUPPORTS 32-BIT INSTRUCTION SET ARCHITECTURE OF THE CURRENT UYK-7. IN MARCH OF NEXT YEAR IT WILL SUPPORT THE UYK-43.

CURRENT SOFTWARE SUPPORT — THE MACHINE TRANSPORTABLE SUPPORT SOFTWARE (MTASS) PROGRAM

- MTASS/"MINI" OR MTASS/M PROGRAM
  - SUPPORTS NAVY STANDARD 16 BIT INSTRUCTION SET ARCHITECTURE COMPUTERS
  - AVAILABLE UYK-20, AYK-14 AND UYK-44
- MTASS/"LARGE" OR MTASS/L PROGRAM
  - SUPPORTS NAVY STANDARD 32 BIT INSTRUCTION SET ARCHITECTURE COMPUTERS
  - INCLUDES UYK-7 AND UYK-43
  - AVAILABLE IN MARCH 1983 CONCURRENT WITH UYK-43 EDM DELIVERY

VG #31 FUTURE SOFTWARE PLANS

THE NAVY'S FUTURE SOFTWARE PLANS CENTER AROUND TRANSITIONING TO THE NEW DOD PROGRAMMING LANGUAGE, ADA EARLY IN 1986 -- WHEN THE NAVY EXPECTS TO HAVE AN ADA COMPILER, CODE GENERATORS, RUN-TIME EXECUTIVES, AND OTHER ADA BASED SUPPORT SOFTWARE READY FOR NEW SYSTEM DEVELOPMENT STARTS AND MAJOR UPGRADES. AS INDICATED IN THIS VIEWGRAPH, THE ADA PROGRAM IS A JOINT COOPERATIVE VENTURE AMONG THE ADA JOINT PROGRAM OFFICE AND THE SERVICES. AS MUCH DEVELOPMENT WILL BE DONE IN COMMON AS POSSIBLE.

THE NAVY IS ALSO DEVELOPING REQUIREMENTS AND PLANS FOR A COMPLETE STANDARDIZED NAVY SOFTWARE ENGINEERING ENVIRONMENT TO BE BUILT AROUND THE ADA PROGRAMMING SUPPORT ENVIRONMENT. THIS ENVIRONMENT WILL THEN BE USED IN NAVY SOFTWARE SUPPORT FACILITIES AND WILL BE TRANSPORTABLE TO CONTRACTOR AND GOVERNMENT SOFTWARE DEVELOPMENT FACILITIES.

IN ORDER TO MAKE A SMOOTH TRANSITION TO ADA, WE WILL RETAIN CMS-2 FOR OLDER SYSTEM USING BI-LINGUAL SUPPORT SOFTWARE.

FUTURE SOFTWARE PLANS

- TRANSITION TO Ada (NEW STARTS AND MAJOR UPGRADES) JAN 1986
  - V&V AGENT FOR ARMY AND USAF Ada EFFORTS
  - DEVELOP DoD INTERFACE STANDARDS FOR ADDITIONAL S.W. ENGINEERING TOOLS
  - DEVELOP THREE TOOLS FOR INTEGRATION WITH ARMY AND USAF MAPSES
  - DEVELOP Ada CODE GENERATIONS (AND OTHER NAVY-UNIQUE COMPILE-TIME COMPONENTS) FOR NAVY STANDARD COMPUTERS
  - DEVELOP STANDARD Ada RUN-TIME OPERATING SYSTEM FOR NAVY STANDARD COMPUTERS
- RETAIN CMS-2 FOR OLDER SYSTEMS AS LONG AS NEEDED (2001?) USING BI-LINGUAL SUPPORT SOFTWARE
- DEVELOP COMPLETE STANDARD SOFTWARE ENGINEERING ENVIRONMENT AROUND APSE

VG #32 BEYOND THE AN/UYK-43 & 44

WHAT ABOUT THE FUTURE BEYOND THE UYK-43, 44, AND AYK-14? IT CAN BE EXPECTED THAT THESE COMPUTERS WILL BE ON ACTIVE SERVICE WELL INTO THE NEXT CENTURY. HOWEVER, EVEN WITH THE CURRENT PLAN FOR TECHNOLOGY INFUSION DURING THEIR DEVELOPMENT AND PRODUCTION LIFETIMES, IT IS EXPECTED THAT THEIR PRODUCTION LIFETIME WILL BE 15 YEARS AT A MAXIMUM. THUS, TECHNOLOGY EFFORTS LEADING TO A SUCCESSOR NAVY STANDARD COMPUTER MUST COMMENCE IN A FEW YEARS. WHAT DO WE WANT TO SEE IN THIS MACHINE? HERE IS SOME OF OUR THINKING AND QUESTIONING, SO FAR. WE ARE ALL ANTICIPATING ADA AS THE FUTURE COMMON DOD HIGH ORDER LANGUAGE, AND THE NAVY IS FULLY COMMITTED TO ADA'S USE. BECAUSE OF ADA'S UNIVERSALITY, MACHINES BUILT TO RUN MOST EFFICIENTLY ON ADA AND WHICH ARE ABLE TO TAKE ADVANTAGE OF ADA'S FEATURES ARE CERTAINLY HIGHLY DESIRABLE. WILL THIS RESULT IN A FAMILY OF MACHINES WHICH CAN DIRECTLY EXECUTE ADA LANGUAGE STATEMENTS, AN ADA INTERMEDIATE LANGUAGE, OR, INSTEAD, A VERY FAST SIMPLE INSTRUCTION SET?

ONE FEATURE THE NAVY FIRMLY WANTS IN IT'S FUTURE COMPUTERS IS THE TOTAL ELIMINATION OF HARDWARE MAINTENANCE, DEPOT REPAIR, AND SUPPLY SUPPORT REQUIREMENTS THROUGH DEVELOPMENT OF MACHINES WHICH WILL NOT HAVE A DEBILITATING FAILURE DURING A NORMAL (15 YEARS?) LIFETIME. THE AN/UYK-43 AND AN/UYK-44 ARE ALREADY PREDICTED TO HAVE ABOUT A ONE YEAR MTBF THROUGH USE OF HIGH RELIABILITY COMPONENTS AND/OR MODEST FAULT TOLERANT AND SELF-HEALING CAPABILITIES. IT WOULD APPEAR THAT TECHNOLOGY CAN GIVE US A "NEVER FAIL" MACHINE WITHIN THE DECADE. SUCH A FUTURE GENERATION MACHINE, HIGHLY OPTIMIZED FOR ADA WOULD ALSO BE A GOOD CANDIDATE FOR THE CONCEPT OF "ACCREDITATION" -- WHEREIN A FEW CONTRACTORS WOULD PROVIDE STANDARD FORM, FIT, AND FUNCTION COMPUTER "BOXES". THE NAVY WOULD NOT BE CONCERNED ABOUT THE INTERNAL DETAILS OF SUCH ACCREDITED DEVICES BECAUSE OF THEIR HIGH PROBABILITY OF LASTING A NORMAL LIFETIME WITHOUT FAILURE; AND IN THE EVENT OF RANDOM FAILURE OR, MORE LIKELY--DAMAGE; REPLACEMENT WOULD BE MADE AT THE COMPUTER LEVEL--WHICH IN THE NEXT GENERATION WILL PROBABLY BE A SURPRISINGLY SMALL DEVICE.

WITH A STANDARD SOFTWARE SUPPORT ENVIRONMENT, FURTHER HARDWARE STANDARDIZATION WILL OFFER THE FEASIBILITY TO BUILD ON A MODULAR ARCHITECTURE TO SATISFY SPECIFIC REQUIREMENTS - NOT ONLY FOR NAVY SYSTEMS BUT FOR JOINT SERVICE USE AS WELL.

BEYOND THE AN/UYK-43 & 44

AN Ada COMPATIBLE MACHINE

- EXECUTE Ada DIRECTLY?
- EXECUTE Ada INTERMEDIATE LANGUAGE?
- NORMAL LIFETIME WITHOUT OPERATIONAL FAILURE?
  - — SELF HEALING
  - — REDUNDANCY
- SUPPORT ACCREDITATION?
- MODULAR FAMILY COVERING PERFORMANCE RANGE?
- JOINT SERVICE FAMILY?

Biography

RADM Wayne D. Bodensteiner, Deputy Chief of Naval Material for Acquisition, Naval Material Command, Washington, D. C. 20360

BS	Southern Methodist University	1954
MS	Naval Postgraduate School	1963
PhD	University of Texas	1970
Naval Aviator Designation		1956
Graduate - Test Pilot School, Naval Air Test Center		1959

Experience

28 years with the U. S. Navy, including the following assignments:

- VP-40 Squadron,
- Flight Instructor, NAS Corpus Christi
- Flag Lieutenant, Naval Forces Philippines
- VS-41 Squadron
- VS-29 Squadron
- Commanding Officer, VS-33
- S-3A Test Director, Naval Air Test Center
- Executive Assistant, ASW & Ocean Surveillance Programs (OP-095)
- Commanding Officer, NAS Jacksonville
- Director, Undersea & Strategic Warfare & Nuclear Dev Div (OP-981)
- Commander, Fleet Air Mediterranean

Current Assignment: Deputy Chief of Naval Material for Acquisition

Responsible for Navy material acquisition process; program evaluation; system engineering; production; test and evaluation; ranges and targets; acquisition and project management policy (including embedded computer standardization policy)

THE ARMY'S PERSPECTIVES ON STANDARDIZATION OF COMPUTER HARDWARE AND SOFTWARE

Brigadier General Robert D. Morgan

US Army Communication-Electronics Command, (CECOM)
Fort Monmouth, New Jersey

The Army has recognized the need for standardization of computer hardware and software on battlefield automated systems by promulgated policies on computer resources management and standardization of embedded computer resources.

CECOM as DARCOM's delegated systems engineer for C^2 has developed an approach to meet the major requirements for battlefield automation including; software and hardware standardization, maintenance of competition, reducing technological obsolescence, increasing survivability, providing for technology upgrade, maximizing affordability while minimizing proliferation of computer resources.

The program is based upon development of a standard Military Computer Family (MCF) and peripherals, the standard Ada* Language System and support software, interoperability standards between systems, a multi-level secure operating system, distributed processing research and an effective Post Deployment Software Support (PDSS) approach, all developed using Computer Resource Management (CRM) principles and techniques.

INTRODUCTION

The Army is concerned with Battlefield Automation Systems for each of the five functional areas of the Army's Tactical Forces, i.e. Maneuver Control, Fire Support, Air Defense, Intelligence/EW and Combat Service Support.

The development and implementation of systems for these applications has resulted in an inordinate amount of proliferation in the computer resources area.

Problems caused by proliferation are as follows:

- Impedes survivability during battle.
- Increases cost and complexity of production, logistics, maintenance and training.
- Impedes growth and evolution of systems.
- Increases cost and complexity of software development and post deployment support.

* Ada is a registered trademark of the Department of Defense (AJPO).

It is clear that the Army will not be able to fund or support all of these systems unless some degree of standardization is achieved in common hardware, common software, common support facilities and tools, and common hardware and software documentation formats are adopted. There is also the need to define interoperability standards between these systems.

POLICIES

The DOD has recognized that the control of the proliferation of computer resources can only be accomplished by standardization. Two Computer Resource policies have been issued, namely DODD 5000.29, Management of Computer Resources, and DODI 5000.31 which limits the number of high order languages (HOL's) in DOD systems.

On 9 August 1982, the Under Secretary of the Army updated a policy for "Standardization of Embedded Computer Resources", which states that the standard high order language Ada* must be used in all Army Battlefield Automated Systems (BAS) after January 1983 and the standard Military Computer Family (MCF) will be used in all BAS after the completion of FSD critical testing of MCF in about 1986. This policy memorandum also assigns DARCOM the responsibility for coordinating of all computer resources planning to minimize software development environment and to minimize the use of assembly language programming. The Army requirement for Ada and MCF is documented in AR 1000-1 and DARCOM R 70-16.

IMPLEMENTATION APPROACH

The Army monitors and reviews actions in the development of Battlefield Automated Systems (BAS) for C^2 and communications at the Command and Control System Program Review (C^2SPR). The first of which was held in November 1981. The C^2SPR at this review identified action items in the following technology product areas:

Information Processing
Input/output Devices
Information Networks
Survivable Communications

Army standardization efforts as applied to computers and software for Battlefield Automated Systems (BAS) are addressed in the Information Processing technology product area. Included are: a standard Military Computer Family (MCF), a standard high order language Ada and support tools, a multi-level secure operating system and the use of distributed processing architectures.

Standardization as applied to the Input/Output Devices technology product area, includes the development of advanced computer peripherals technology, the development of standard MCF peripherals including smart friendly terminals, soft copy imagery, tactical displays and an all electronic mass memory.

In the Information Networks technology products area programs have been established in the following areas:

Distributed Data Communications
Broadband Switching (Tandem)
Battlefield Spectrum Management
Fiber Optics Technology
VHSIC Exploitation
Dispersed Command Post Networks using millimeter wave and VHF technology.
JINTACCS (Army interoperability)

In the Survivable Communications technology product area programs established include:

Anti-Jam modulation techniques
Multi-level secure networking using the Mobile Subscriber Equipment
Communications Security technology
Fiber optics cable
Tactical Satellite Communications
Tactical Multi-channel Communications

The DARCOM responsibility for systems engineering for tactical command control and communications (C^3) in the battlefield has been assigned to the US Army Communications-Electronics Command (CECOM) at Fort Monmouth, New Jersey.

CECOM in undertaking this systems engineering responsibility for C^3 has addressed the following major requirements for Battlefield Automated Systems:

- Must have both computer hardware and software standardization
- Battlefield computers must be survivable
- The approach must consider cost factors and be affordable
- The approach must insure continuing competition
- Must keep pace with rapidly advancing technology
- Must accommodate evolutionary upgrade of Battlefield Systems

CECOM's approach in consideration of these requirements has been to develop a survivable cost effective standard family of computer equipment (MCF) and peripherals, supported by appropriate software, including a standard high order Ada language system and support tools, a secure multi-level operating system, interoperability standards between systems and making use of the latest developments in distributed processing architecture.

MILITARY COMPUTER FAMILY (MCF)

The development of a standard Military Computer Family (MCF) as a solution to the proliferation problem and to meet all Army requirements is based on the development of the following family members:

A Super mini-computer AN/UYK-41
A Micro Computer AN/UYK-49
A Single board micro computer and chip sets.

The MCF family will be based on a single instruction set architecture, MIL-STD-1862 (NEBULA) and will be based on use of the standard high order language Ada and Ada support tools including a secure operating system and will include three standard interfaces

CECOM in its approach to developing the Military Computer Family has addressed all six major requirements for battlefield automation.

The approach to standardization while maintaining real competition is based on the following approach:

- Open solicitation for Advanced Development
- Awarded four contracts for competitive advanced development
- Select two AD contractors to compete
  Full Scale Development (FDS) and ILS.
- Competitive FLY off for production
- Production Award for five years
- Repeat above cycle for next phase, every five years

The problem of reducing technological obsolescence is taken into account in the first MCF phase by requiring technology insertion of 1984 technology for the 1986 first five year production.

The concept of initiating a new phase of development for MCF every five years will provide the opportunity to insert new technology for each phase via competition, while providing a balance between technology and logistic support. The program will provide for upward compatible evolution of the NEBULA instruction set, the Ada language and interfaces.

The approach to assure maximum survivability is based upon the concept of a fault tolerant design using VLSI technology which is inherently reliable, built for the military specification environment and which will include built-in test features to assure maintainability.

The approach to providing for evolutionary upgrade is based on an approach which will: Provide initial capabilities in excess of known requirements; permit interfaces to support expansion via distributed processing; permit successive generations of products to be software plug compatible; provide improvements to NEBULA and will support improvements to Ada and will provide for higher computation speeds, memory, power and reliability.

The approach to affordability or reduction of life cycle costs includes the following:

- Extensive competition, competitive life cycle cost analysis
- All costs competed including fixed prices based on quantities ordered during 5 year production.
- Large production under single production contract will result in lowest unit cost.
- Simplified logistics--fewer spare parts in the pipeline.
- Emphasis on high reliability and maintainability will reduce ILS costs.
- Potential for software and plug-compatible upgrade to more cost-

effective units that emerge from each phase.

- Common Ada-NEBULA compiler, code generator and software environment—use of commercial hosts.
- Reduced costs for Post Deployment Software Support.

MCF PERIPHERALS

As part of the Military Computer Family (MCF) program, CECOM has initiated an MCF peripherals program to develop a family of standard MCF compatible militarized peripherals for Army wide use in battlefield systems. This is intended to reduce proliferation of types of terminals/peripherals, enhance battlefield survivability, reduce logistics, maintenance and training and simplify software development and support.

No technical barriers exist to the initiation of development of a family of standard peripheral devices under the MCF program. The role of the MCF program is to ensure that such peripheral devices are properly interfaced to MCF computers and can serve in multiple applications. The MCF program will only address the development of peripheral devices that contain significant risk so that no Project Manager need risk the success of his system. Examples of such devices are Large Screen Display using thin film electroluminescent (TFEL) technology (to be initiated under the MCF program in Advanced Development in FY-85) and an all electronic mass memory (to be initiated in FY-88). In addition, work will be initiated in FY-83 to interface high technology, commercial peripherals to MCF as models for the purpose of test, evaluation and demonstration.

THE ADA LANGUAGE SYSTEM

CECOM is committed to the development and validation of the Ada Language System for use in MCF and other battlefield automation system.

The approach in developing the Ada Language has been to provide the following:

- Language optimized for Embedded Computer Resource needs
- Reduce needs for assembly languages
- Real-time capabilities
- Parallel processing
- Separate compilation facilities
- Error resistant features

Potential benefits of the Ada language lies in commonality in training; transportability; communication; and support software tool focus.

CECOM initiated the development of an implementation of the Ada Language System and supporting tools in 1980 for introduction into operations in 1983.

This program is called the Ada Language System (ALS) development.

The Ada Language System development includes a complete programming environment. The ALS is a system of tools required to develop and implement

Ada Applications programs. The tool set includes the following:

Command Language Interpreter	Pretty Printer
Data Base Management System	Text Editor
Configuration Management System	Text Formatter
Ada Compiler	File Comparator
Linkers	Symbolic Dynamic Debugger
Assemblers	Test Coverage Checker
Stub Generators	Timing Analyzers
Set-Use-Static Analyzer	Loaders

The ALS, as currently being developed, can generate machine code for five target environments. These are the Digital Equipment Corporation VAX 11/780 (VMS), the VAX 11/780 (Bare), the ROLM 1602B, the Military Computer Family (MCF) and the Tactical Computer System (TCS). These capabilities will be introduced during the January - August 1984 timeframe.

The ALS is the developer's and maintainer's interface to the computer. Its aim is to provide an efficient implementation of the Ada Language as well as to provide a beneficial environment for programming in Ada. The ALS is written in Ada and will be placed under configuration management control via its own configuration management tooling. It is planned to use the ALS in all DARCOM Software Support Centers to maintain Army weapons systems utilizing Ada. It is expected that many Army development efforts will utilize the ALS during original development. This will be effected via making the ALS available to the industry. To this end current planning includes making ALS early versions available on a friendly site basis to selected companies during 1983. During this period the ALS will continue development toward forward introduction and use in 1984.

INTEROPERABILITY

Another major area for consideration of software standardization is in the intra-Army interoperability between battlefield automated systems in the five Army Tactical Forces functional areas. Interoperability development is based upon the following documents: the Battlefield Automation Management Plan (BAMP), the Automated Battlefield Interface Concept (ABIC), the Battlefield Interoperability Management Plan (BIMP) and the Technical Interface Design Plan (TIDP).

The impact of interoperability involves every segment of a system such as system design, interface characteristics, computer programs, data base, and communications. Externally it impacts upon the operator, the communication media, management and, because of its evolutionary nature, doctrine. The impact upon the management structure is great because it forces a higher degree of centralization and control due to involvement of two or more systems.

Key software interoperability consideration is involved in the following: man/machine interfaces, software versus firmware, flexible message generation capabilities, software interoperability training, provision of adequate multi-level security for joint Army/NATO interoperability and considerations for continuity of operations, and survivability of automated systems in the battlefield.

Security considerations, in the view of interoperability, cause increased complexity due to the number of systems and levels of security required within each system. There is need to provide a common security module that will provide adequate multi-level security for Joint Services and NATO interoperability in a hostile environment.

The approach to interoperability for the Army is to provide a common base for interoperability across all systems including design of standard software/firmware module, scenarios for interoperability testing, on-line and off-line software training routines, standard documentation and procedures.

MULTI LEVEL SECURITY

A major aspect of CECOM's program for standardization of Battlefield Automated Systems is the design and development of a set of multi-application real-time operating systems for both multi-level and dedicated secure computer-based applications which utilize the Military Computer Family (MCF), the MCF peripherals, and the Ada Language System.

The goals of this program are to develop operating systems which:

a. satisfy the broad resource control requirements of the entire range of Army computer systems, both dedicated secure (systems high) and multi-level secure applications.

b. are developed in Ada utilizing the Ada Language System (ALS) and compatible with applications developed in Ada using the ALS.

c. are modificable, expandable, maintainable, and easy for applications designers to utilize.

d. are efficient, i.e., do not incur a high overhead and allow for the development of real-time, high performance, embedded computer systems, as well as stand alone computer systems for the Army.

e. employs the current state-of-the-art in formal verification methodologies and tools, security mathematical models, and secure operating systems designs. A dedicated secure operating system for applications which do not require hardware/software security features.

The technology exists for the development of trusted computing systems wherein hardware and software security features are incorporated which can be certified and trusted to run in the multi-level secure mode. This technology consists of formal mathematical models of computer security, computer architecture features to support security, formal verification methodologies and tools, and Kernelized operating systems. The MCF Operating System (MCFOS) program will include an operating system for those applications that require the system for dedicated secure and systems high applications because there are many applications which can be naturally and appropriately operated in the dedicated mode without loss of performance, and because the security protection required for the multi-level mode requires a higher overhead with respect to performance.

DISTRIBUTED PROCESSING TECHNOLOGY

New Army tactical C^2 systems for the late 1980 early 1990 time-frame must put a premium on survivability, availability, and mobility of their system designs. The achievement of these objectives requires the implementation of new tactical systems which provide enhanced continuity of operations (CONOPS), survivability, and mobility through the utilization of distributed processing architectures and techniques, as well as the standardization of hardware and software (Ada and MCF).

CECOM is conducting research in the mathematical modeling of distributed systems, the use of Ada in a distributed processing environment, and the exploration of survivable systems. The mathematical modeling research is to develop mathematically precise models of distributed processing algorithms and techniques, and study and extend their properties in a precise mathematical setting. The Ada research projects are concerned with providing a support environment (development and maintenance) for distributed systems implemented with a Higher Order Language (HOL), and to examine those mechanisms that will allow Ada programs to be engineered to run on a distributed computer system. The research in the area of survivable systems is to define and explore issues in distributed processing that relate to tactical command and control functions. CECOM will provide a means to develop, evaluate, improve, validate and demonstrate state-of-the-art distributed processing techniques, including data replication and location, update synchronization, and error recovery to assure a consistent data base and fast accurate access to battlefield information.

An experimental distributed processing facility which will include six processing elements/nodes is under development to provide a flexible test bed for the integration of local and remote network distributed processing techniques. these integrations will be programmed in Ada and will be directly transferable to MCF. This facility will also provide a vehicle for investigation of distributed processing techniques as applied to the tactical battlefield, to support the requirements for more survivable Army tactical systems in the future.

POST DEPLOYMENT SOFTWARE SUPPORT

The Army Program for standardization of BAS and C^3 systems can only be successful if concurrent development of plans are initiated for effective support of these systems when fielded.

A study of the Army's Post Deployment Software Support (PDSS) problem was initiated in 1978 by an Army task force and working groups. A concept plan for PDSS was completed in May 1981.

The key feature of the plan are the establishment of eleven PDSS centers, each providing central management for a functional/mission area. The establishment of only eleven PDSS centers was a consolidation of resources for all Army requirements.

The PDSS centers will use both commercial and military computers with commercial computers as the host for development and support, and the military computers as the target system for test and debugging of fielded software. To be effective, a PDSS must have a role in all life cycle phases of a battlefield automated system.

In the conceptual and developmental phase, its role is to insure that management and technical decisions are compatible with support needs and to acquire design knowledge. In the deployment and support phase the PDSS role is to maintain, modify, and control all system software.

PDSS must be involved in the complete BAS software life cycle to deal with new and changing requirements, interface changes, and insertion of new technology. In its maintenance role, it is concerned with technical changes and correction of latent errors.

COMPUTER RESOURCE MANAGEMENT (CRM)

These major programs will be implemented using Computer Resource Management (CRM) principles as per DOD and Army Directives. CECOM, as well as other DARCOM MACOMS, has taken action in the formation of Computer Resource Working Groups (CRWG) and the preparation of Computer Resource Management Plans (CRMP) for each program. To educate and train system developers in CRM, a series of CRM Guidebooks have been prepared and a standard set of data item descriptions (DIDs) developed to coordinate documentation. DARCOM and CECOM has also been participating on a DOD basis with the efforts of the Joint Logistics Commander's Joint Policy Coordinating Group on Computer Resource Management (JLC-JPCG-CRM) since its formation in 1977.

CONCLUSION

The Army and CECOM's approach to standardization of computer hardware and software is based upon meeting the major requirements for battlefield automation, including: maintenance of competition, reducing technological obsolescence, increasing survivability, providing for technology upgrade and maximizing affordability while minimizing proliferation of computer resources.

The program is based on development of a standard Military Computer Family (MCF) and MCf peripherals, a standard Ada high order language system and support tools, a multi-level secure operating system, interoperability standards between systems, distributed processing research and effective Post Deployment Software Support (PDSS).

All of this development effort based upon use of Computer Resource Management principles and techniques.

This standardization approach is planned to result in a survivable cost effective approach to providing automation in the battlefield in the late 1980's and 1990's.

BRIGADIER GENERAL ROBERT D. MORGAN

Brigadier General Robert Daniel Morgan, Deputy Commanding General for Research and Development, US Army Communications-Electronics Command (CECOM), Fort Monmouth, NJ and Commander of the CECOM Research and Development Center, was born in Buffalo, NY, on March 2, 1934. He attended Lackawanna High School, NY, graduated from Canisius College, Buffalo with a bachelor's of science degree and earned his master's degree at Troy State University, Troy, Ala.

General Morgan is the recipient of the Legion of Merit, Bronze Star Medal, Meritorious Service Medal with 2 OLC, Air Medal with 6 OLC, Army Commendation Medal, National Defense Service Medal, Armed Forces Expeditionary Medal and the Vietnam Commendation Medal.

During his 26 years of military service, General Morgan's assignments have included Battalion Commander, 40th Signal Battalion, 1st Signal Brigade, Vietnam; Chief, Technology and Applications Directorate, Audio Visual Agency, Washington, DC; Deputy Project Manager for International Communications Systems, Communications Systems Agency, Fort Huachuca, Ariz. and Project Manager for Position Location Reporting System, Fort Monmouth, NJ.

UK MOD ACTIVITY IN AIRBORNE DIGITAL SYSTEM STANDARDS

by

Dr A A Callaway
Procurement Executive - Ministry of Defence
Flight Systems Department
Royal Aircraft Establishment
Farnborough, Hampshire
United Kingdom

1 INTRODUCTION

This paper covers a range of Ministry of Defence (MOD) activity concerned with the development and establishment of standards for both the hardware and software aspects of airborne digital systems. It considers the multiplex data bus and related digital interface standards, standards for programming languages and software development methods, and computer instruction set architecture (ISA) standards.

The formal recognition of a standard in the UK MOD is through its publication as a Defence Standard (Def Stan). The successful drive in UK towards airborne digital system standards is evidenced by a number of published Def Stans, which will be discussed within the paper.

One of the driving forces in airborne system standardization is the need to retain an international perspective. This has been recognized for many years, through MOD involvement with the Working Parties of the Air Standardization Coordinating Committee (ASCC) and the various NATO standardization committees, and has led in turn to fruitful cooperation between RAE and, in particular, USAF/ASD, who are the joint project authorities in a digital avionics Information Exchange Project. This cooperation has been notably valuable in the development of Mil Std 1553B and Mil Std 1750A.

Since the majority of the audience for this paper will not be familiar with the role and purpose of the RAE (Royal Aircraft Establishment), it is convenient here to say a few words concerning it. RAE is the largest of the UK MOD's research and development establishments. It is principally based at Farnborough, about 35 miles south-west of London.

RAE is at the centre of research and development into military (and some civil) aviation and space activities in the UK. Its primary function is to advance aerospace technology and to use it to assist Government agencies, the Armed Forces and industry in a variety of projects, evolving new operational techniques and solving problems which arise in the equipment when in service. Throughout, particular emphasis is placed on the rapid and effective transfer to industry of the knowledge and expertise stemming from the RAE work. This facet has been particularly important in our standardization work.

Section 2 of this paper, then, addresses interface and data transmission standards, Section 3 covers software and programming languages, and Section 4 discusses instruction set architecture standardization.

2 DATA TRANSMISSION

RAE has been involved with the development of Mil Std 1553B, in collaboration with USAF, since 1975. The history of this involvement was presented in a paper for the first AFSC Standardization Conference (Ref 1), which also detailed work done in UK in support of the adoption of the multiplex data bus in airborne systems.

The paper states that in consideration of such adoption and its impact on future avionic systems architecture, it became clear that, in addition to the data bus itself, other digital interfaces would still be required, and it was decided that the publication of the data bus standard in UK would be as part of a compendium of compatible interface standards.

In order to ensure that such standards would be universally acceptable, and to make use of the knowledge and expertise of industry, MOD formed the Data Transmission Standards Committee (DTSC) under the chairmanship of the DANav (Directorate of Air Navigation & Reconnaissance Development) branch of MOD Procurement Executive, with RAE acting as the technical authority and with extensive membership from both airframe and avionic system companies.

The vehicle for standards emerging from the work of DTSC is Def Stan 00-18, which is, in effect, the compendium referred to above. Def Stan 00-18 caters for the transmission of serial digital data through specification of an electrical multiplexed data bus and both electrical and optical single-source interface systems. The transmission of discrete signals is also catered for by both electrical and optical interfaces. The Def Stan is published in a number of parts, as will now be discussed.

It was clear from the outset that in order to support the introduction and maintenance of the standards, guides would be necessary. The guides have been produced, and they form Part 1 of the Def Stan. They give an official interpretation and amplification of the clauses in the standards, where experience suggests this is necessary, and also contain information to help the designer avoid known difficulties in implementation.

The Mil Std 1553B serial time-division command/response multiplexed data bus has been incorporated into the family of standards as Def Stan 00-18 (Part 2). This is technically identical to 1553B, but has been re-written in order to conform with the accepted Def Stan format and to 'anglicize' some of the phraseology. UK MOD has also supported the ratification of 1553B as NATO STANAG 3838 and ASCC Air Standard 50/2, and Def Stan 00-18 is the instrument by which these agreements are implemented.

Although there is wide experience in the United States of developments incorporating former versions of the Mil Std, UK has adopted the 'B' version from the outset. DTSC has, therefore, sought to rationalise the implementation of the standard by providing a focus for both UK Government and Industry development efforts, and this has resulted, for example, in the chapters in the guide (Part 1) on the definition of preferred remote terminal responses and on testing. MOD has also funded the development of devices to perform the remote terminal function, which has resulted in the products now available from Marconi and Smiths Industries.

Def Stan 00-18 (Part 3) defines a single-source, single/multi-sink serial digital transmission interface system for applications which do not require multiple data sources or that wish to implement a simpler interface to the multiplexed data bus. The salient features of the interface are that it retains the resilient electrical parameters, hardware configuration and data encoding of the multiplexed data bus whilst simplifying the interface protocol in a well-defined manner appropriate for point to point applications.

Def Stan 00-18 (Part 4) rationalises discrete signalling to three types of interface that meet the majority of aircraft requirements. Discrete signalling is divided by functional constraints into:

a. those which require fast response times (critical timing signalling),

b. those which are not particularly constrained (non-critical timing signalling), and

c. those which need to supply power with the signal (low power switching).

The development of Part 4 represents the first attempt on a national basis to standardize discrete signalling, and is aimed at ensuring electromagnetic and functional compatibility whilst reducing costs through simplification and optimisation.

Both Part 3 and Part 4 of Def Stan 00-18 are receiving attention in the Avionic Systems Working Party (AVSWP) of NATO-MAS and in ASCC Working Party 50 as potential international standards for data transmission.

Def Stan 00-18 (Part 5) contains four fibre optic single-source data transmission systems for use in aircraft: two serial data transmission interfaces, at 1 MHz and 10 MHz, respectively, a fibre optic stub for the multiplexed data bus and a discrete signal interface. The development of this standard posed problems due to the immaturity of the technological field, which were overcome by creating a hierarchical structure to system specification that allows scope for, and guides, continued development. On the topic of fibre optics, DTSC were also responsible for commenting on, and contributing to, the fibre optic version of Mil Std 1553B drafted by SAE A-2K.

To summarise, then, Def Stan 00-18 currently comprises five parts:

Part 1 - Application guide.
Part 2 - Multiplex data bus.
Part 3 - Single-source, single/multi-sink interface.
Part 4 - Discrete signalling.
Part 5 - Fibre optic interfaces.

These standards have been shown to satisfy the majority of existing requirements for digital signalling and to meet the aircraft physical and electromagnetic compatibility requirement. They have also served to focus both component and system development resources within the UK, to the benefit of both MOD and Industry.

Current considerations in DTSC include planning for updating the guides and for development of further testing and evaluation techniques, plus discussion of new options for standardization, such as high-speed buses, video distribution and standard data word formats. In such areas as these it is anticipated that cooperation with the United States through the established channels of the IEP, the DTSC/SAE dialogues and through common support of the NATO and ASCC committees will be as fruitful in the future as it has been in the past.

3 SOFTWARE DEVELOPMENT AND HIGH ORDER LANGUAGES

MOD has operated a standard high order language (HOL) policy since 1970. This is a tri-service policy and is based on the language CORAL 66. CORAL is an acronym for Computer On-line Real-time Applications Language. The original version of the language was specified in 1964 and the current version was specified, as its name suggests, in 1966, at the MOD establishment which is now known as RSRE (the Royal Signals and Radar Establishment).

The first Official Definition of CORAL 66 was published in 1970 (Ref 2), as was the first general users' guide (Ref 3), and the use of CORAL as a standard was ratified by the publication of Def Stan 05-47. Although the standard policy had operated from 1970, the Def Stan was not published until 1977.

CORAL 66 is based on Algol 60, with developments to make it more suitable for the embedded on-line task. Technologically it is a similar level language to Jovial J73. Since its adoption as a standard, many MOD development programmes have utilised the language in land, sea and airborne applications, and it has been implemented on many host and target processor types. There is, therefore, in UK a considerable body of knowledge and experience, not only in the technical application of the language but also in the problems and details of operating a standard language policy.

CORAL 66 is a sequential programming language; ie, it provides no facilities for such real-time concepts as process synchronisation and scheduling. It was originally designed to operate in conjunction with the operating system or executive extant on whatever the target machine happened to be. Thus, although this makes for an implementation-independent language definition, it inevitably leads to machine-dependent implementations, and this was one of the early problems encountered with the use of the standard. Also, with the trend towards host-target software development methods and standard software support suites, it became increasingly important to standardize on a particular software development and executive technique.

The method chosen for this purpose is MASCOT, which was also developed at RSRE. MASCOT is an acronym for Modular Approach to Software Construction Operation and Test. An Official Definition of MASCOT (Ref 4) was published in 1980, and the following extract drawn from that document describes its function. MASCOT:

a. defines a formal method of expressing the software structure of a real time system which is independent of both computer configuration and programming language;

b. imposes a disciplined approach to design which yields a highly modular structure, ensuring a close correspondence between functional elements in design and constructional elements for system integration;

c. supports a program acceptance strategy based on the test and verification of single modules and larger collections of functionally related modules;

d. provides for a small, easily implemented executive for the dynamic control of program execution at run time;

e. provides for a straightforward and flexible method for system building;

f. can be applied through all stages of the software life cycle from design onwards;

g. can form the basis of a standard system for software procurement and management.

MASCOT is a design method supported by a programming system. It is neither a language nor an operating system although it includes elements that are related to both these aspects of programming. It brings together a coordinated set of tools for dealing with the design, the construction (or system building), operation (or run time execution) and testing of software.

A powerful standard facility now exists in UK MOD for the development of software for embedded computer systems, consisting of MASCOT used in conjunction with CORAL 66, and extensions to CORAL have been designed which facilitate its interface with MASCOT. This combination is significantly different from, say, conventional languages with parallel processing features, which allow the creation of parallel processes and their intercommunication data areas within the context of a program. CORAL/MASCOT inverts this order, in that the expression of parallelism and intercommunication is placed at the highest point of software design, and programming, in the conventional sense, has a subordinate role, being used to express individual system components. This method, therefore, promotes a very high level of structured modularity in the applications software.

The principal problem with the use of CORAL 66 in future airborne systems, however, is that it is a national standard, whereas many of the programmes with which we are concerned are, or will be, international in nature. Also, CORAL is based on development which started nearly twenty years ago, and does not reflect modern thinking in language technology. These two factors are among the principal reasons for MOD's long term interest and participation in the US DOD's Ada language development programme.

There is little doubt that MOD will adopt Ada as a successor to CORAL 66 when this becomes practicable. One of the important factors to be considered in this respect is the nature of the Ada Programming Support Environment (APSE), to which considerable attention is being given in the DOD. MOD effort, coordinated by RSRE, has been responsible for certain investigations about APSE requirements and MOD is currently putting together a programme in conjunction with other UK agencies to develop a UK Ada environment which also supports the CHILL language.

It should be noted, however, that Ada will not be adopted in an uncritical way. There are clearly still imperfections which must be addressed, and it is certain that there will be restrictions imposed for the use of the language in, for example, safety-critical areas. UK is currently involved with other NATO nations in an AGARD Working Group studying the potential impact of Ada on aircraft guidance and control system programming, and this is just one example of the type of activity under way.

It should also be noted that there is an intention to carry forward the MASCOT type of development philosophy into the Ada era, and consideration is currently being given to the publication of certain aspects of the MASCOT approach in a Def Stan.

4 INSTRUCTION SET ARCHITECTURES

UK MOD currently operates a computer policy for embedded systems which specifies two UK proprietary processors: the Ferranti Argus M700 and the GEC 4000 series.

The Argus M700 architecture has been specified in Def Stan 00-21, and both the present policy architectures have been used in a variety of land, sea and air-borne applications. MOD is currently funding a VLSI version of the M700, known as the M700/40.

As was mentioned in the Introduction and in the discussion on CORAL 66, however, the air side, in particular, needs to keep an international perspective and to operate with standards which are internationally acceptable. This is particularly so when considering the requirements of international collaborative projects and the compatibility of home and export markets for avionic equipment, and is the reason for RAE participation in the US Mil Std 1750A exercise.

Again, the history of this participation is detailed in Ref 1. Suffice it to say here that RAE (and certain UK companies) have attended and contributed at all 1750A User Group meetings and RAE has also occupied a seat on the USAF Control Board. In addition RAE was responsible for construction of what is believed to have been the first single card 1750A processor.

Since the 1980 Conference, RAE has been responsible for the contract on Ferranti to provide two flyable 1750A prototypes with full 1553B bus control capability, and these are scheduled to be delivered by the end of 1982. RAE has also placed a contract on Systems Designers Ltd (SDL) for the development of a 1750A target to their CONTEXT (CORAL/MASCOT) development, hosted on VAX 11. This will provide full CORAL 66 and MASCOT facilities for 1750A, and will be ready in about the same timescale as the prototypes.

RAE has also run the McDonnell-Douglas assembly level package and has been responsible for distributing it to a number of interested firms in the UK. The Acceptance Test Program is now being run successfully at RAE.

On the policy front, RAE has written the case for 1750A to be adopted in the MOD policy, and this is currently under consideration. RAE is also in the process of drafting 1750A as a possible Def Stan. Internationally, MOD is currently supporting the introduction of 1750A as a draft ASCC Air Std and is also considering under a NATO AVSWP Study.

REFERENCES

1	Callaway, A A	United Kingdom Systems Application of Mil Std 1553B with Additional Discussion of Mil Std 1750 Activity. AFSC Standardization Conference, Dayton, 1980.
2	Woodward, P M et al	The Official Definition of Coral 66. HMSO 1970 & 1973.
3	Callaway, A A	A Guide to CORAL Programming. RAE TR70102, 1970.
4	MASCOT Suppliers Association	The Official Handbook of MASCOT 1980.

POLICY SESSION AND PANEL DISCUSSION

SESSION CHAIRMAN : J. M. Hoeferlin
ASD/ENAS

MODERATOR : Col. Eric B. Nelson
Chief of Staff
HQ AFSC

Understanding DoD's Standardization Objectives For Mission Critical Computers

D. Burton Newlin, Jr.

Defense Materiel Specifications and Standards Office
Office of the Under Secretary of Defense (Reseach and Engineering)

Introduction

The Military has placed increased emphasis on computer standardization to improve interoperability and reduce the acquisition and life-cycle cost of computer systems. With the accelerating advances in electronic technology, hardware costs are decreasing while software costs are escalating. Today the majority of DoD's costs associated with computer resources now involve the computer software associated with weapon systems. Management attention has been directed towards standardization as a management tool that can be used to help reduce these costs and prevent a software crisis.

To address this problem the Department of Defense has proposed a three-fold approach to achieve computer standardization. The first element is to strengthen the management attention directed to oversee defense-wide software standardization efforts.[1] The second element is to standardize on a common computer high level language called Ada*, which is designed to reduce DoD's increasing costs for the generation and maintenance of software for its increasing number of computer systems. In the interim until Ada is available, the Air Force has standardized on Jovial J-73 while waiting for Ada to mature. The third element of the plan is the implementation of DoD Instruction 5000.5X - a policy directive that defines the computer Instruction Set Architecture (ISA) levels which established a family of hardware-software interface standards. This minimum set of standards is to be used by Services to achieve increased interoperability, interchangeability and reduce the escalating cost of generating and maintaining software.

Evolution of Computer Policy

A series of Directives and Instructions were issued and others updated to formalize the policies, assign responsibilities and delineate authorities within the Services to address computer acquisition and standardization policies. The major Directives and Instructions are:

- DoDD 5000.29, "Management of Computer Resources in Major Defense Systems."[2]
- DoDI 5000.31, "Interim List of DoD Approved High Order Programming Languages (HOL)."[3]
- DoDI 5000.5X, "Instruction Set Architecture (ISA) Standardization Policy for Embedded Computers." Draft.[4]

* Ada is a trademark of the U.S. Department of Defense.

The objective of DoD Directive 5000.29 is to assure that computer resources are treated as important subsystems throughout the development, acquisition and support phases of the life cycle of defense systems. This Directive mandates that software developed for defense systems be developed in a DoD-approved High-Order Language (HOL) specified in DoDI 5000.31 of which one of the languages is the Air Force's Jovial J-73.

The objective of DoD Instruction 5000.31 is to reduce the proliferation of machine assembly languages and imperfect HOL's used in defense systems and to ensure that the preferred HOL's are used. The policy of moving to the minimum essential number of such approved languages was established and the search for a single language which would serve the broadest spectrum of DoD applications was undertaken. This activity has become the Ada* Program managed by the Ada Joint Program Office (AJPO). The Ada programming language is specified in MIL-STD-1815.[5]

Although use of an HOL in software development and standardizing on a minimum number of HOLs are helping to reduce life-cycle costs and improve the overall software process, there is still required a considerable specialization of the development environment. The computer program still must be tailored to the specific host/target combinations of interest and thus, the reusability and portability of both support tools and applications programs are attenuated. For these and other reasons, consideration has been given to reducing the number of unique environments within which HOL-based applications programs must run. This can be done by controlling the interface between software and the target environment, i.e., the Instruction Set Architecture (ISA) of the target machines.

DoD Instruction 5000.5X

What DoD has attempted with DoD Instruction 5000.5X is to control the interface between software and computer hardware by limiting the number of distinct sets of allowed operations that the software can be called upon to perform. Something like establishing a dictionary of words from which the programming language can be constructed. This dictionary or limited "set of instructions" would form the standardized architecture with which a computer program could be developed. This set of instructions that form the architecture interface for the hardware has been termed "Instruction Set Architecture" or ISA. The objective of DoD's policy is to standardize and limit the number of interface designs with which the software must be designed to perform and to improve software transportability from one machine to another and to enable, where mission dictates, hardware standardization. We are also hoping that the use of a minimum number of ISA's will reduce costs for software support tools. Standardization at the ISA level is the basis of a Proposed DoD Instruction 5000.5X.

The Establishment of a Standardization Area for Embedded Computer Resources

The recommendations made by the Defense Science Board Task Force on Specifications and Standards[6] and a GAO Study[7] has led to the establishment of a new standardization area for Embedded Computer Resources Standards (ECRS). An overall standardization document program plan has been developed to define the coordinated management program for standardization efforts in the Embedded Computer Resources Standards Area.[8] This plan, which is being [illegible] with both government and industry, is the principal source of management information and identifies the various services, NATO, and industrial activities -- either planned or underway --that involve embedded computer resources standardization issues. This plan also outlines objectives and establishes priorities, milestones and resource allocations consistent with the identified work backlog. The program plan is intended to promote conformance of standardization documents and associated data requirements with DoD policy, and ensure the elimination of duplication and overlapping requirements and promote more uniform technical requirements documents. DoD's standardization initiatives addressed in the Embedded Computer Resources Standards program plan are concentrated primarily on the standardization of software high order languages, architectures, software support tools, software documentation, quality control and configuration management.

The plan also identifies tasks being conducted by the Joint Logistics Commanders to minimize the types and kinds of data requirements and ensure that data requirements and applicable Data Item Descriptions can be selectively applied and tailored to match the technical requirements that are contractually imposed through standards.

Tasks are also identified within the program plan to review nongovernment specifications and standards for adoption, as well as identifying projects being accomplished by industry associations as they relate to software problems and opportunities which have been identified that are of interest to the Department of Defense.

Advantages of a Standard Instruction Set Architecture

The advantages of standardizing on a minimum set of Instruction Set Architectures (ISAs) is that it accrues cost benefits during the acquisition and support phases of the system life cycle. The reuseability of available support software such as compilers, editors, linkers, assemblers, and instruction level simulators, etc., significantly reduces program risk because the software development can be independent of the hardware development. Implementing a standard ISA on programs allows program managers to start coding efforts prior to the receipt of actual hardware and to use a common set of proven reliable software support tools. The first benefit mentioned should allow us to decrease software development time. The military can save many millions of dollars in life cycle cost across the forces, while enjoying shorter schedules, and fewer surprises during development. But most important, a standard ISA will allow the military to exploit the explosion in hardware technology without locking in on a single vendor's proprietary computer architecture.

Comparing Standardization Approaches

Each of the Services has chosen a separate ISA standardization approach. The Army has chosen to develop a military computer family (MCF) of computers and introduce a new government-owned, non-proprietary 32-bit architecture called Nebula under MIL-STD-1862, principally for ground-based command and control systems. The Navy has a large existing software base on just three types of computers, while the Army is faced with over 50 different types in the inventory. The Navy is attempting to inject the most modern technology as replacements for the AN/UYK-7, AN/UYK-20 and, to a lesser extent the

AN/AYK-14. The principal rationale for holding to these ISAs for the Navy is the conservation of the existing software investment. For the Navy, this investment is connected with CMS-2.

The Air Force has a very different maintenance and logistic situation than do the Army and Navy. The Air Force is standardizing at the computer architecture level by adopting its own set of instructions as defined in MIL-STD-1750. Under this approach, the computer manufacturers, will build these instructions into their computer equipment in a manner best suited by their manufacturing techniques and technology. The Air Force MIL-STD-1750 is a 16-bit architecture suitable for airborne, missile and ground-based real-time systems. Recent growth to MIL-STD-1750A, as a result of broad industry and user-group input, will add extended memory management and hence render 1750A germane to a wide range of applications as well. MIL-STD-1750A processors are being implemented in the F/FB-111, F-16, F-5G Tigershark, B-1B, LANTIRN, and is projected for a few other applications.

It may never be practical to go to a single ISA for all three Services, however, steps must be taken to minimize the number of ISA's that must be supported by DoD. Despite this objective, multiple sources for the hardware will be assured as will the injection of the very latest advanced technology. As one can see, the strategy which has been developed has led toward a significant reduction in both architectures and languages. By coupling the Ada Program and ISA standardization policies, we believe we have demonstrated an effective and efficient degree of control in a very rapidly changing technical and business environment.

Appropriate Level of Computer Standardization to be Achieved

Because of Congressional opposition to DoD's ISA proposed standardization policy, the DoD has been unable to provide central direction as to the level of standardization to be achieved. The Military Services are using different standardization philosophies and concepts, although a unified approach may accommodate most of their automation requirements. The purpose of the proposed DoD Instruction 5000.5X is to provide this central direction.

To assure that industry had an ample opportunity to review the proposed policy, under DoDI 5000.5X an "Open Forum" was held at the Andrews Air Force Base on November 2, 1979. This forum was announced well in advance in the Commerce Business Daily (CBD). The proposed policy was also provided to the major industry associations for comment. As a result of this open forum, numerous comments were received from those in industry who supported and opposed the proposed ISA standardization policy.

Many of the computer manufacturers and vendors who voiced opposition to DoD's ISA standardization approach to reduce escalating software costs, stating it will negatively affect hardware competition and that modern technology will resolve the related software transportability issue. An example is a new programming method called function-level computing, which is being developed with the capability of linking computers of radically different architectures and ostensibly simplifying software development tasks.(9) But function-level computing is still in its infancy and many theoretical and practical problems remain to be solved before it can become a reality.

Since industry agreement was not reached on the proposed ISA standardization policy -- as a result of a few dissenting industry votes --the Under Secretary of Defense (Research and Engineering) decided that a review by the Defense Science Board would be appropriate prior to any further processing of the Instruction. A special Defense Science Board Task Force on Embedded Computer Resources Acquisition and Management was chartered in September 1981 with representatives of both government and industry to review the ISA and other embedded computer standardization issues. The task force met between September 1981 and January 1982 to advise the Secretary of Defense and the Under Secretary of Defense for Research and Engineering an overall research, engineering, acquisition and management issues and to provide long-range guidance in these areas. Although the major issue which spawned the Task Force was DoD's concern over the need for DoDI 5000.5X and Industry's resistance to it, there were many related issues which the Task Force was also specifically asked to address.

Following completion of the DSB Task Force Study, a series of GAO reports were written and Congressional hearings conducted to review this entire ISA issue.

Congressional Research Service Report

The Congressional Research Service issued a report on February 3, 1982, entitled "Military Computers in Transition: Standards and Strategy".[10] This report outlined the evolution of DoD's computer standards policy up to the point in time the Defense Science Board Task Force on Embedded Computer Acquisition and Management had completed their study. This report openly questions the "integrity and professionalism" of the members selected to serve on the Defense Science Board's Task Force.

GAO Investigation of Charges "Favoritism"

In March 1982, Congressman Jack Brooks requested that the GAO undertake an immediate investigation to determine the validity of the charges raised by the Congressional Research Service Report. The Congressional Research Service Report[11] stated:

> "While it is recognized that it is difficult to get knowledgeable professionals who are not involved in some aspect or other of this specialized subject, the selection of task force members who have a recognized interest in the outcome of the subject may be open to criticism. For example, in some instances, members of the Board were from companies that were participating in related Defense programs and therefore might have biased views of the problem and therefore may be deemed to lack objectivity."

Congressional Hearings on Favoritism in Computer Procurements Within DoD

The Subcommittee on Legislation and National Security of the House Committee on Government Operations held hearings on July 21, 22, and August 4, 1982, on possible favoritism in computer procurements within the Department of Defense. These hearings addressed the Department's proposed policy (DoD

Instruction 5000.5X) to design and develop its own computer systems and the Defense Science Board's review of this instruction requested by the Under Secretary for Research and Engineering.

GAO Report on Objectivity of DSB Task Force

The GAO issued a report [10] 22 July 82 on the objectivity of the Defense Science Board's Task Force on Embedded Computer Resources Acquisition and Management. This report criticizes the DoD:

- For not taking adequate steps to form a balanced task force.
- For not properly reviewing the financial disclosure forms to prevent the appearance of conflicts of interest.
- For providing the task force with information drawn primarily from sources supportive of DoD's position.

DoD Directed to Table Issuance of DoDI 5000.5X and Restudy ISA Issue for the Armed Services Committees

The Armed Services Committees have expressed concern that standardization at the ISA level may adversely affect the options available to the department for achieving maximum effectiveness in new weapon systems.

Because of their concerns, the committees have directed DoD to postpone issuing DoDI 5000.5X pending the completion and submission of a study to the Committees on Armed Services of the Senate and House of Representatives that addresses the following issues:[13]

(a) A full assessment of the applicability of commercial computer technology.
(b) The desirability of standardization at ISA level.
(c) The degree of software transportability that the various approaches permit and how each approach would affect DoD's hardware/software logistics support requirements and the cost of computer system ownership.
(d) An assessment of the relative merits and liabilities involved in the incorporation of each approach into Department of Defense weapon systems.
(e) A justification for all on-going service computer development projects.
(f) A plan to reduce the proliferation of these computers.

The committee believes that this report would provide a necessary blueprint for Executive branch management and Congressional oversight of the DoD's efforts to streamline its computer logistics situation while laying the ground work for an approach that provides vigorous competition and, to the maximum degree possible, preserves the option for technology insertion.

Without ISA Standardization, Life Cycle Costs Will Continue to Escalate

Even with the development of the Ada* High Order programming language compiler programs must still be written, tested, corrected and maintained for the unique instruction set of architectures of a selected family of host machines.

The costs of maintaining these compilers and other support tools will proliferate as the languge becomes more widely accepted and as more companies develop compiler programs in order to implement Ada on the unique ISA of their machines. The same ISA software problem that is facing DoD will also be facing the software used within the Federal Government. The magnitude of the DoD's problem is greater than the rest of the Federal Government since DoD buys 70%, in quantity, of all government computers.

To address the software problems with general purpose computers facing the Federal Government, the General Services Administration established the Automated Data and Telecommunications Service. Under this organization four offices have been established: Software Development Office, Federal Compiler Testing Center, Federal Conversion Support Center and Federal Software Exchange Center. The Office of Software Development is concerned with reducing the costs spent on both the development and maintenance of general purpose software used within the Federal Government. The Federal Compiler Testing Center provides assistance to the private sector to meet the Federal Information Processing Standards (FIPS) by providing compiler validation systems and services to a wide variety of clients.

The ISA Stifling Technology Fallacy

It has been alleged that if DoD standardized on a minimum number of Instructions Sets it would result in DoD using stifled technolgy, but this is a fallacy since in the case of the Air Force's MIL-STD-1750 it does not dictate the technology which should be used to comply with the ISA. To prevent this from happening and provide documents for DoD to use on its current contracts; standards are constantly being revised, updated and developed to meet the requirements of the new technologies. Under the Defense Standardization program, it is mandatory that all standardization documents be reviewed every five years and either be revised, updated or cancelled. Under the proposed DoDI 5000.5X DoD would continually review proposed ISA that should be added to the list approved for new designs, just as obsolete ISA designs would be removed from this list for new designs.

Program Assessment

The major objectives of standardization within the DoD are to improve operational readiness, and reduce costs. Through the standardization of high order languages, instruction set architectures and improved computer software documentation, the efficiency of logistics support should be improved and life cycle costs should be reduced.

In the development of our standardization program for embedded computer resources, we are seeking to obtain participation and input from as wide and diverse a group as possible. An objective of the embedded computer resources standardization program is to ensure an orderly development of a cadre of standards. An effective standardization program in the computer technology area will require a close working relationship both within the Government and with industry.

Conclusion:

The standardization of ISA's is still a highly controversial issue, both from a technical and political point of view, since it will affect competition for hardware procurements. Several companies who opposed MIL-STD-1750A are not in the process of developing 1750 processors on their own funds. The Department of Defense is convinced that, because of advances in hardware technology and the advent of very high speed integrated circuits technology, standardization of hardware should be both vendor and technology transparent and that our computer standardization requirements should be expressed through a very limited number of instruction set architectures,

It is believed that more effective use of software interface standards which define a standard set of computer instructions or Instruction Set Architectures can significantly improve software productivity and transportability by reducing software proliferation and therefore reducing the costs for development and maintenance of computer programs and documentation. To attain this objective, we must have effective management controls in the acquisition process.

References

(1) Proceedings of the Twenty-First Annual Technical Symposium on Computing and Government, National Bureau of Standards, Gaithersburg, MD, DoD's Embedded Computer Standardization Initiatives, D. Burton Newlin, Jr., June 17, 1982.

(2) DoD Directive 5000.29, Management of Computer Resources in Major Defense Systems, dated 26 April 1976, (Being Revised).

(3) DoD Instruction 5000.31, Interim List of DoD Aproved High Order Programming Languages (HOL), dated 24 November 76 (Being Revised).

(4) DoD Instruction 5000.5X, Draft, Instruction Set Architecture (ISA) Standardization Policy for Embedded Computers.

(5) MIL-STD-1815, Notice 1, Ada* Programming Language, 10 June 1981. *Ada is a trademark of the U.S. Department of Defense.

(6) Defense Science Board Report of the Task Force on Specifications and Standards, 21 April 1977.

(7) GAO Report, LOD-80-69, The Department of Defense's Standardization Program for Military Computers -- A More Unified Effort Is Needed, 18 June 1980.

(8) Standardization Document Program Plan for Embedded Computer Resources Standards (ECRS) Area. Draft, Defense Materiel Specifications and Standards Office, Dated 15 September 1982.

(9) IEEE Spectrum, Function-Level Computing, John Backus, August 1982.

(10) The Library of Congress Congressional Research Service Report, Military Computers in Transition: Standards and Strategy, Dated February 3, 1982.

(11) Ibid.

(12) GAO Letter Report, GAO/FPCD-82-55, Objectivity of the Defense Science Board's Task Force on Embedded Computer Resources Acquisition and Management, 22 July 1982.

(13) Department of Defense Authorization Act, 1983, Congressional Conference Report No. 97-749, August 16, 1982.

AIR FORCE STANDARDIZATION ACTIVITIES

COL EDWARD T. AKERLUND
DIRECTOR OF COMPUTER RESOURCES
DCS/ACQUISITION LOGISTICS
AIR FORCE SYSTEMS COMMAND

AIR FORCE STANDARDS

MIL-STD-1750A	16 BIT ISA
MIL-STD-1553B	MULTIPLEX-BUS
FIPS-PUB-21-1	COBOL
MIL-STD-1589B	JOVIAL J73
IEEE STD 716-1982	C/ATLAS TEST LANGUAGE
IEEE STD 717-1982	C/ATLAS SYNTAX
ANS X3.9-1978 FIPS-PUB-69	FORTRAN
MIL-STD-1753	DOD SUPPLEMENT TO ANS X3.9-1978 (FORTRAN)

COMPUTER RESOURCE ACQUISITION

AFR 800-14	ACQUISITION AND SUPPORT POLICY
MIL-STD-480	SPECIFICATION PRACTICES
MIL-STD-483	CONFIGURATION CONTROL
MIL-STD-490	CONFIGURATION MANAGEMENT
MIL-STD-881	WORK BREAKDOWN STRUCTURE
MIL-STD-1521A	REVIEWS AND AUDITS
MIL-STD-52779	QUALITY ASSURANCE

COMPUTER SOFTWARE MANAGEMENT (CSM) SUBGROUP

JOINT POLICY

- UNDER REVISION

SOFTWARE DOCUMENTATION STANDARDS

- COMMENTS BEING ANALYZED

SOFTWARE DEVELOPMENT STANDARDS

- COMMENTS BEING ANALYZED

SOFTWARE QUALITY EVALUATION

- BEGINNING REVIEW OF DRAFTS

JOINT POLICY

PATTERNED AFTER AFR 800-14

DEFINES LIFE CYCLE

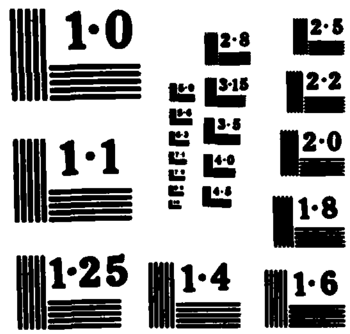
- SYSTEM/COMPUTER RESOURCE
- COMPUTER SOFTWARE

FOR EACH DEVELOPMENT PHASE, DEFINES

- ACTIVITIES
- PRODUCTS
- BASELINES

AD-A145 697 PROCEEDINGS PAPERS OF THE AFSC (AIR FORCE SYSTEMS COMMAND) AVIONICS STAND..(U) AERONAUTICAL SYSTEMS DIV WRIGHT-PATTERSON AFB OH DIRECTORATE O.. 3/6

UNCLASSIFIED C A PORUBCANSKY NOV 82 F/G 1/3 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.15
3.5
4.0
4.5

SOFTWARE DOCUMENTATION STANDARDS

DIDS DEVELOPED FOR

- MANAGEMENT PLANS
- ENGINEERING SPECIFICATIONS/DOCUMENTATION
- TEST PLANS/PROCEDURES/REPORTS
- SUPPORT DOCUMENTATION

WILL PROVIDE COMMON DOCUMENTATION SET

SOFTWARE DEVELOPMENT STANDARDS

SOFTWARE STANDARD DEVELOPED

- DRAWS UPON PAST EFFORTS [1679 & RADC SPEC]
- ENCOMPASSES DEVELOPMENT/SUPPORT PHASES
- TO BE APPLIED CONTRACTUALLY
- TIED TO POLICY & DIDS

CHANGES TO RELATED STANDARDS

- REVIEWS & AUDITS [1521A]
- CONFIGURATION MANAGEMENT [483]
- SPECIFICATION PRACTICES [490]

SOFTWARE QUALITY EVALUATION

TASKS COMPLETED

- DEFINITION/FRAMEWORK
- DRAFT POLICY
- DRAFT SOFTWARE QUALITY STANDARD

FOLLOW-ON TASKS

- DEVELOP GUIDEBOOK
- BUILD COURSE OUTLINE

EMBEDDED COMPUTER STANDARDIZATION IN THE NAVY -- POLICY AND PRACTICE

Captain David L. Boslaugh, USN

Director, Tactical Embedded Computer
Program Office
Naval Material Command Headquarters
Washington, D.C. 2036
(202) 692-3966

ABSTRACT

The Navy has 5000 standard embedded computers in 450 types of warfare systems using over 50 million lines of computer program source code. Some of these machines are approaching obsolescence and are experiencing speed and memory saturation in many applications. In mapping out a program for successors to these computers the Navy has considered a number of constraints, and has established a number of goals and policies which will be reviewed in this talk. The Navy organization which plans, develops, and enforces embedded computer standardization will be presented; followed by a short overview of the Navy thinking about the future beyond the the new Navy standard machines being developed today. The Navy's goal to reduce future software costs through transitioning to the new Ada programming language will also be reviewed.

INTRODUCTION

The initial introduction of shipboard digital computers some twenty one years ago in the Naval Tactical Data System (NTDS) was met by the first seagoing users with a form of enthusiasm. For example "No damned computer is going to tell me what to do" was an oft heard quote by the workers in the Bureau of Ships NTDS project office. Furthermore, the "outlandish" project office claims of 300 hours mean time between failure for such complex machines was viewed with skepticism and derision; and the makers of analog weapons computers went so far as to publish small treatises in their instruction manuals "proving" the impossibility of using digital computers for weapons fire control.

Twenty one years later the controversies still rage around the military use of embedded digital computers. The controversies have even risen to the halls of Congress. The issue, however, is no longer whether embedded digital computers will be used. They are fully accepted now--even an absolute necessity. The issue rather, is standardization of embedded computers and there are tens of billions of computer acquisition dollars at stake in the Navy alone. Likewise, battle effectiveness and billions of dollars in savings and cost avoidance are also at stake. Because of the very large coming market for embedded computers and because of the large and diffuse number of computer suppliers and Navy embedded computer users there are great pressures against standardization. The rapid change of technology also millitates against standardization. It is, therefore, absolutely essential that any viable service embedded computer standardization program have not only a line of state-of-the-art equipment and standard support software readily available for users, but also there must be an organization to manage and enforce the use of the standard product line. Additionally, the product line can not be allowed to become stagnant therefore, continuous long range requirements determination and planning for technology upgrades becomes an essential part of this organization.

POLICIES FOR STANDARDIZATION

Just what is an embedded computer? The concept of embedded computers means different things to different people. To some it involves processors (particularly microprocessors) physically embedded in, and a part of, some larger containing functional equipment such as a radio, display terminal, or a PAC-MAN. To others it is a stand-alone computer which serves as the heart of a larger combat system composed of a number of separate equipments--such as a gun system, command and control system or a missile system. The Navy definition of embedded computer embraces both of the above. Specifically, for Navy purposes an embedded commputer is:

> "a computer that is an integral component of any system or subsystem contributing to the combat capability of operating forces. This includes:
>
> - o ship, submarine, and shore applications
> - o non-militarized computers when used in combat systems."

SCOPE OF NAVY EMBEDDED COMPUTER MANAGEMENT POLICIES

All Navy warfare systems, and direct warfare support systems, using embedded computers are subject to the Chief of Naval Material's embedded computer resources management policies. System types covered include:

- o weapons
- o communications
- o command and control, and
- o intelligence

All platforms--ship, submarine, air and shore are included. Because of the wide variability of computer applications ashore, and because some warfare systems ashore can use embedded non-militarized computers, (because of beningn environmental conditions) application of informed judgement is sometimes required to distinguish between warfare support and non-warfare support systems using non-militarized computers. As specific examples of non-militarized computer applications ashore, Navy elements of the World Wide Military Command and Control System (WWMCCS) and the Ocean Surveillance Information System are classified as warfare support systems and are subject to CNM embedded computer standardization policies. On the other hand, logistic support systems, laboratory scientific computers, and software support facilities, inter alia, are not so subject.

The Navy ECR standardization policies do cover all life cycle phases of warfare systems development and acquisition, commencing with concept formulation and progressing through major upgrade and life cycle maintenance. The unique nature of computer software causes the requirement for coverage in the very early development phases. It has been found that the investment cost for prototype software even for "breadboard" conceptual systems is so high that it is usually not economically feasible to "throw" this software away when progressing to more advanced R&D. Rather, the software evolves from an initial conceptual base which is usually expensive. Thus, original very expensive software done in a non-standard language, for a non-standard computer, and not documented in accordance with standards can "lock" a system into continued use of non-standard equipments because of unacceptably high cost of re-working the software to meet standardization requirements. Thus, the requirement to start early on with standards. In addition as much as 70% of total software cost is in post delivery support. This has shaped Navy standardization policies which require "up front" software planning and design investments which will hold down overall life cycle software maintenance costs.

Standardization areas covered by Navy embedded computer policy include the following:

- o Hardware, including:
  - -- a line of computers and microprocessors (currently the AN/UYK-7 "mainframe" and the AN/UYK-20 minicomputer--to be succeeded by the AN/UYK-43 mainframe and the AN/UYK-44 minicomputer/microprocessor which are both nearing the end of development.
  - -- special purpose signal processors
  - -- display and operator terminal subsystems
  - -- magnetic tape handlers, and
  - -- rotating mass memories.
- o Standard support software, including:
  - -- operating and executive systems for the standard computers
  - -- basic software development "environments" called Machine Transportable Support Software (MTASS). (MTASS is hosted on a number of large commerical computer systems.)

o Software Development Methodology, in accordance with MIL-STD-1679 (Navy) Weapon System Software Development.

o Software Documentation Standards, as required in Secretary of the Navy Instruction (SECNAVINST) 3560.1

o Software Configuration Management, as required by NAVMAT Instruction 4130.2

- o Life cycle support of software, as required by NAVMAT Instruction 5200.27 which requires, inter alia, the assignment of a Navy software life cycle support activity for each embedded computer applications or support program.

- o Standard High Order Programming Languages -- CMS-2 supports the standard general purpose computers and SPL/I (Signal Processing Language/I) supports the AN/UYS-1 and AN/UYS-2 signal processors. Ada will begin to replace CMS-2 for new system starts in 1986.

All of the above "Navy Standards" must be used in all Navy weapons systems developments and acquisitions or a formal waiver must be obtained from the Chief of Naval Material (CNM) in advance of applying funds for acquisition or use of a non-standard. CNM policy on use of the standards is specified in a series of documents called Tactical Digital Standards (TADSTANDS) and is reiterated in periodic CNM policy letters and in Chief of Naval Operations (CNO) instructions and notices. More about the TADSTANDS will come later.

CURRENT EMBEDDED COMPUTER DEVELOPMENT POLICY

The Navy is in the final stages of developing computers to succeed the obsolescent AN/UYK-7 and AN/UYK-20 computers and the AN/UYS-1 Advanced Signal Processor (ASP). One of the controlling policies in these developments has been a requirement for industry-wide competition and a public statement that the winner of each competition will be awarded all Navy standards production in the applicable performance category of each standard equipment for a specified length of time. The Navy acquires all rights and data for these standard equipments in order to support possible second sourcing or recompetition; and central Navy configuration management/control is established for each equipment type.

Other specific policy guiding these developments is as follows:

- o Provide for logistically Identical Computer Modules (within a given equipment type) for Lowest Life Cycle Support Costs - The cost of at-sea maintenance (including inventories of spare parts) is the one largest compelling factor for Navy computer resources standardization. One generation of standard computers can be supported with a fleet-wide inventory of spare parts costing about \$60 million; whereas the shipboard spares for a proliferation of 35 logistically different computers would cost over one billion dollars. Thus, until failure-free, and consequently spares-free, computers are developed the Navy will require families of logistically identical computers.

- o Ensure Software Transportability from UYK-7 and UYK-20 Systems - The large Navy investment in weapon systems applications software (approximately \$5 billion in replacement value) and in standard support software (valued at \$100 million) has required that successor computers be able to upward "emulate" the instruction set architectures (ISA) of existing standard computers in order that software can be reused and "upward" transported as necessary. The ISAs of the new computers not only support the software of their predecessor computers but contain extended instruction sets with new and more powerful instructions for new applications. For example, it is planned that production UYK-43 and 44 computers will also include Ada optimized instructions.

- Improve Reliability and Maintainability - Current developments will have unprecedented improvements in reliability, maintainability, automated trouble shooting, fault isolation, and automated system reconfiguration and recovery. The developing agencies have been directed that these improvements must take precedence over performance improvements if compromises should become necessary. Thus far, both types of specifications are being met, or exceeded, and no compromises have been necessary.

- Support Rapidly Expanding Microprocessor Applications - The Navy has not previously had a standard shipboard embeddable microprocessor for direct use inside other equipments; however, microprocessor applications are expanding more rapidly than any other computer class. To stem a potential proliferation of "16 bit" embedded microprocessors, the AN/UYK-44 standard electronic module (SEM) card set and the AN/AYK-14 card set will be available, and required, for use in embedded microprocesor applications.

- Support Expanding Programmable Signal Processor Applications - The current standard Navy signal processor, AN/UYS-1, is being used in applications which tax its full speed and memory. One application even requires packaging three units in one cabinet, which is the economic limit for multiple packaging approaches. Thus, competitive development of a successor signal processor, AN/UYS-2 Enhanced Modular Signal Processor (EMSP), has commenced. The AN/UYS-2 support software environment will be compatible with the AN/UYS-1 level allowing graceful transition of software. The AN/UYK-44 will be used as a general purpose control processor for the AN/UYS-2.

POLICY FOR AIRBORNE APPLICATIONS

The Navy standard AN/AYK-14 airborne computer has entered production within the past two years and will remain the airborne standard, with technology upgrades, for the remainder of this decade. Specific elements of policy regarding development and use of the AN/AYK-14 include the following:

- Modular Expandable Versions - Aircraft weight and space requirements levy demands that a computer be no larger or heavier than need be for a specific airborne application. Thus, AN/AYK-14 versions are built from a common set of electronic modules and placed in different packaging to accommodate differing amounts of memory, power supplies, processors, etc. Four versions currently exist.

- AN/AYK-14 emulates AN/UYK-20 ISA (with extensions) - Emulation of the Navy standard "16 bit" ISA was required in order to reuse the existing $50 million investment in standard AN/UYK-20 support software.

- o AN/AYK-14 or AN/UYK-44 Card Sets for Embedded Microprocessor Applications - The AN/UYK-44 Standard Electronic Module card set is both ship and air qualified. Thus, airborne embedded microprocessor applications are required to use either of these, or obtain a formal waiver.

EMBEDDED COMPUTER SOFTWARE POLICY

Software investment is projected to become the dominant life cycle cost in embedded computer systems, and is thus a very appropriate target for cost reduction and productivity improvement initiatives. Navy policies in support of these goals include the following:

- o Support Software - The $100M existing base of AN/UYK-7 and AN/UYK-20 support software (such as Machine Transportable Support Software (MTASS) Systems) will be built upon and extended to support new upward compatible ISA hardware developments. The MTASS systems and other elements of "compile time" support software will eventually be incorporated into a complete Navy standard software engineering environment (SEE) which will also evolve to incorporate:

  - -- the Ada programming language, and the Ada programming support environment (APSE).
  - -- software requirments analysis and design tools
  - -- programmer productivity and software quality enhancement tools

- o Applications Software - In addition to upward compatible computer instruction set architectures, and software policy already discussed, the Navy will place new emphasis on reuseability of applications software from project to project. Key elements supporting software reusability will include:

  - -- Design of a highly accessible SEE data base containing applications module libraries.
  - -- Interface and module design standards for reuse of applications software modules
  - -- Heavy emphasis on software module reliability and maintainability
  - -- Rigorous configuration management of all applications software.

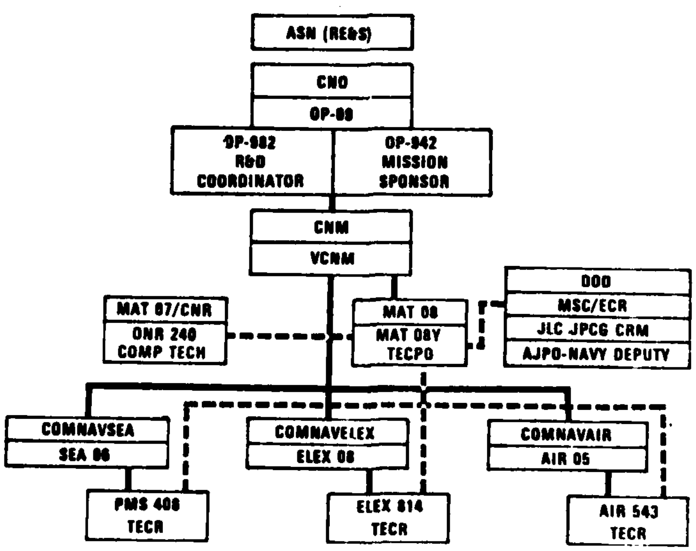
- o Ada Plans and Policy - The DoD Ada Joint Program Office was established in December 1980 and the Navy was the first service to provide a permanent service Deputy Project Officer in February 1981. The Navy is fully committed to transitioning embedded computer software to the Ada language and, in accordance with current plans and funds availability, production quality Navy Ada products will be available to Navy user systems for "new starts" and major upgrades in early 1986. The Navy Ada Language System will use the Army Ada compiler and the Army Ada Programming Support Environment (ASPE) with minimum changes and additions. Initial Navy-unique Ada products will include Ada oriented runtime operating systems for the AN/UYK-43 and 44 and the AN/AYK-14 computers, and Ada target code generators for all these computers.

NAVY ORGANIZATION FOR EMBEDDED COMPUTER STANDARDIZATION

The Navy has a single command, the Naval Material Command, responsible for development, acquisition, and life cycle support of all warfare related platforms and warfare systems. The Chief of Naval Material, a four star admiral, who reports to the Chief of Naval Operations is responsible for managing the Naval Systems Commands through a relatively small Naval Material Command Headquarters staff. The Navy Systems Commands in turn have the "muscle" to provide full material support to the fleet. Some of the systems commands such as the Naval Supply Systems Command and the Naval Facilities Engineering Command are not direct users or developers of embedded computer resources, however the three "platform" systems commands--the Naval Air Systems Command, the Naval Electronic Systems Command, and the Naval Sea Systems Command both use and develop/acquire the Navy embedded computer standards; and are integral parts of the Navy ECR standardization organization. This organization, which is depicted in Figure 1. is brought to a focal point at Headquarters Naval Material Command in the Tactical Embedded Computer Program Office (TECPO) which is code-named MAT 08Y. The TECPO office, headed by a captain USN, reports to the Naval Material Command's Deputy Commander for Acquisition (MAT-08), a rear admiral. Each of the three platform systems commands (SYSCOMs), in turn, has an office responsible for managing the use of the standards within the SYSCOMS, and in two cases these offices also develop assigned standard hardware and/or software.

Figure 1.

NAVY ORGANIZATION FOR EMBEDDED COMPUTER STANDARDIZATION



THE TACTICAL EMBEDDED COMPUTER PROGRAM OFFICE

Major functions of the NAVMAT Headquarters Tactical Embedded Computer Program Office are listed in Table 1. which factors them into three areas: long range planning, policy development & enforcement, and program management.

Table 1. Tactical Embedded Computer Program Office (MAT 08Y) Functions

- o LONG RANGE PLANNING
  - o Master Plan for Embedded Computer Resources
- o POLICY
  - o Formulate and implement ECR policies (TADSTANDS) and procedures for the CNM (TADSTAND approval/disapproval authority)
  - o Review acquisition documentatin for Navy Systems
  - o Monitor CM and Life Cycle Support of all Navy standard ECR products
- o PROGRAM MANAGEMENT
  - o Program director for NAVMAT R&D related to Navy Standard ECR
  - o Manager Navy 6.3 and 6.4 ECR RDT&E Program Resources
  - o PDA for:
    - --UYK-43 and UYK-44 Standard Embedded Computers
    - --Standard support software
    - --Ada HOL development and implementation
  - o Assign DA's for Navy standard ECR products
  - o Direct CM of Navy standard HOL definitions, compilers, and ISA specs

Long Range Planning - The most tangible product of the long range planning function is the Navy Master Plan for Embedded Computer Resources. As shown in Table 2., this plan treats the future as near term, up to 1989, and long term, from 1989 to the turn of the century. It is updated yearly after Navy-wide review and comment to reflect new developments, suitations, and needs; and is issued over the signature of the Chief of Naval Material. It contains descriptions and plans not only for approved projects, but also goals and objectives not currently in the approved Five Year Defense Program (FYDP). The plan treats all aspects of embedded computer resources including computer hardware, peripherals hardware, support software, programming environments, applications software, manpower, training, and life cycle support issues.

Policy Functions - MAT 08Y develops ECR policies for the Chief of Naval Material which are written and disseminated in condensed documents called Tactical Digital Standards (TADSTANDS). The TADSTANDS cover such subjects as standard definitions, lists of standard hardware, and authorized programming languages. Each TADSTAND also delineates the conditions under which a waiver might be granted. The TECPO office conducts the waiver review and approval/disapproval process for the CNM; and as a part of the enforcement

Table 2. Summary of Master Plan Strategies and Objectives

Problems \ Solutions	Near Term (Pre 1989)	Long Term (Post 1989
CP-642 obsolescence	AN/UYK-7, AN/UYK-20, AN/UYK-43, AN/UYK-44	NAVY EMBEDDED COMPUTER PROGRAM
AN/UYK-7 obsolescence	AN/UYK-43	
AN/UYK-20 obsolescence	AN/UYK-44	
Lack of standard airborne computer	AN/AYK-14	
Lack of standard signal processor	AN/UYS-1 (ASP). Develop AN/UYS-2	AN/UYS-2 (EMSP)
Standard disk system obsolescense	Develop high technology mass memory	High technology mass memory system
Proliferation of non standard medium scale displays	Develop standard medium scale modular display subsystem	Standard medium scale modular display sub-system
Programming support deficiencies	Upgrade current Navy standard programming support systems; develop Ada based software engineering environment	Ada compatible standard Software Engineering Environment
Proliferation of Navy programming languages and dialects	Phase out divergent dialects; develop Ada language capability	Implement Ada as single Navy standard programming language

function, reviews all acquisition planning documents for all Navy system acquisitions to verify and/or require adherence to the TADSTANDS. As a further followup on Navy wide application of TADSTANDS, the TECPO office provides personnel to logistics review boards and project acquisition reviews.

- Program Management - The designated SYSCOM ECR project offices develop and acquire standard hardware and software products as assigned by the Chief of Naval Material and under "overview management" of the TECPO office as the Program Director for all NAVMAT TECR related research and development. For the most critical of the multiplatform standards such as:

  -- UYK-43 and UYK-44 development
  -- Standard support software, and
  -- Navy Ada products

  TECPO also serves as Principal Development Activity--meaning funds and final technical management authority remain at Naval Material Command Headquarters level for these programs.

THE SYSTEMS COMMANDS' ECR OFFICES

In addition to day-to-day management of assigned standards developments, performing commodity management, and ordering agent functions for their production programs, the SYSCOM ECR project office act as the first line standardization management agents for CNM. This latter function includes:

- assessing user requirements for new ECR needs
- monitoring and advising SYSCOM weapon system (user) projects
- enforcing CNM standardization policies including initial technical review of all waiver requests

Specific assignments of the SYSCOM project offices include the following:

- Naval Air Systems Command - Code AIR 543 (Avionics Systems and Embedded Computer Resources Division)
  -- AN/AYK-14 standard airborne computer

- Naval Electronic Systems Command - Code ELEX 814 (Computer Resources Division)
  --Development of standard software management procedures and practices (primarily Navy contribution to Joint Logistics Commanders software management initiatives)

- Naval Sea Systems Command - PMS 408 (Shipboard Tactical Embedded Computer Resources Project Office)

  -- AN/UYK-7 and AN/UYK-20 production acquisition
  -- AN/UYK-43 and AN/UYK-44 development
  -- AN/UYS-2 Enhanced Modular Signal Processor Development
  -- Standard support software development and life cycle support

-- Navy Ada products development
-- Standard shipboard peripherals acquisition (numerous types)
-- Standard shipboard display and operator terminal subsystems acquisition
-- Standard Navy compilers and associated support software

Even though a standard ECR product is developed and acquired by one SYSCOM it is universally used by all systems commands. Other users of most of the Navy standards include the U. S. Marine Corps, U. S. Army, U.S. Air Force, U.S. Coast Guard, and the navies of seven allied nations. Table 3. shows the developers and users of the standard computers.

Table 3. Navy Standard Embedded Computers
Who Acquires/Supports and Who Uses?

Computers	Acquisition/Support	Users	
UYK-20		NAVAIR	Australia
UYK-44	NAVSEA	NAVELEX	Germany
UYK-7	(PMS-408)	USMC	Italy
UYK-43		USAF	Japan
UYS-2		Army	Spain
		Coast Guard	Netherlands
AYK-14		NAVAIR	
UYS-1	NAVAIR	NAVSEA	
	(PMA-264)	NAVAIR	
		NAVELEX	
		USAF	

In some cases the acquiring SYSCOM is not the major user of a given standard. A case in point being the AN/UYS-1 Advanced Signal Processor which is acquired by the Naval Air Systems Command where it is used in airborne acoustic signal processing applications; however its dominant use is in a shipboard versions for surface ship and submarine acoustic signal processing.

THE PROCESS

Major elements of the Navy's ECR standardization process have already been reviewed. This section will serve to tie the process together by elaborating on the use of the Tactical Digital Standards (TADSTANDS). Approximately 400 Navy warfare systems are now using over five thousand standard Navy embedded computers; however there have been, and will probably continue to be, cases where the use of the standards might be:

--technically infeasible
--economically prohibitive, or
--operationally impractical

If either the using system or the standard cannot be adapted or modified to suit such situations, a waiver will be considered by the Chief of Naval Material. In addition to standard equipments, there are other areas of Navy ECR standardization policy coverage. Table 4. lists the full set of currently effective TADSTANDS.

Table 4.
Current NAVMAT Tactical Digital Standards (TADSTANDS)

TADSTANDS	
A	Standard Definitions
B	Standard Computers, Peripherals, and I/O interfaces
C	Standard Programming Languages
D	Reserve Capacity Requirements
E	Software Development, Documentation, and Testing Standards

Table 5. summarizes the required contents of a waiver request. It should be noted that the justification must emphasize why a standard can not be used rather than why a non-standard should be used. Also total life cycle costs and other life cycle considerations of standards vs. the proposed non-standard must be articulated.

Table 5. TADSTAND Waiver Requests

- CNM (MAT 08Y) via Appropriate Cognizant SYSCOM Office
- Justification to include:
    - System Name
    - Platform(s)
    - Embedded Computers
    - Storage, I/O Requirements
    - Software Constraints
    - Environmental Requirements/Constraints
    - Why Standards Cannot Be Used
    - Proposed Substitutes(s)
    - Plans, Schedule Cost Data on Proposed Substitutes (ILS)
    - R&M Requirements
    - Cost Comparisons

CHALLENGES AND OPPORTUNITIES

The Navy has excellent ECR standards, workable policies for standardization, and effective procedures in place for carrying them out. We should be able to lean back and relax--but such will never be the case! Technology and operational needs are advancing so fast that technology upgrade programs must start even before new standards enter production in order for new standard equipments to remain competitive with the general industry state of the art. Technology infusion studies and planning for the AN/UYK-43 and AN/UYK-44 are already under way even though they will not be delivering in large production quantities until 1984. Ada oriented instruction set extensions and VHSIC technology are being considered as well as "conventional" VLSI and memory density upgrades. Also, even with technology upgrades the new computers will reach the economic limits of upgrading in not too many years, and new next generation high technology successor competitive computer developments will be required. The Navy currently intends to start a next generation computer advanced development programs in 1985, and already next generation computer architecture studies are in progress. One of the objectives of these studies is to ascertain the best architecture (i.e., highest performance for lowest hardware and software life cycle cost) for run-time execution of Ada compiled programs. Another general objective of next generation Navy ECR standardization is the elimination of at-sea maintenance and repair through acquisition of "failure-free" machines. Such form, fit and function) (F^3) machines could potentially be acquired from a number of supplies without the need to have a prohibitively expensive proliferation of at-sea inventories of repair parts

Considerable Navy effort will also go into software quality, reuseability, and programmer productivity enhancement. This work will be centered around implementing Ada, and the Ada Programming Support Environment. Software engineering tools of all types will be integrated with ASPE and this entire body of support software will be evolved into a standard comprehensive modular software engineering environment which will form the basis for Navy "software factories."

Navy embedded computer resources management must remain forever sensitive to new operational needs, new technological capabilities, and continuously implement the same in order to retain a viable Navy ECR standardization program. The challenges will never cease.

CAPT David L. Boslaugh, USN, Director, Tactical Embedded Computer Program Office, Naval Material Command, Washington, D. C.

CAPT Boslaugh exercises overview management of the development and acquisition of standard embedded computers and associated peripherals for use in Navy warfare systems. For the most critical Navy multiplatform ECR products he is assigned as principal development activity. His office also performs long-range planning for standard computer developent and acquisition requirements, including both hardware and support software, and carries out the development-distribution-enforcement of related standardization policies throughout the USN.

Previously, CAPT Boslaugh was Assistant Project Manager for C^3 Software Development and Support, C^3 Systems Project Office, Naval Electronic Systems Command. He has also served as Director, Communications Research and Development, for the Naval Electronic Systems Command. Other assignments have included five years in the Naval Tactical Data System Project, three years at the NASA High Speed Flight Station, Edwards, California as an aeronautical research engineer, and command of the Naval Electronic Systems Security Engineering Center.

CAPT Boslaugh is a member of AIAA, IEEE, ASNE and Sigma Xi. He holds a BS degree in Aeronautical Engineering from the University of Minnesota, and an MS degree in Engineering Electronics from the U. S. Naval Postgraduate School.

ORGANIZATIONAL OVERVIEW

Policy, Regulations, Status,

Implementation Plans:

US Army

BY: Col J. Frank Campbell

SUMMARY

- "STANDARDIZATION" DOES NOT MEAN TECHNOLOGY LOCKOUT
- ARMY PROGRAM IS AIMED AT FOUR AREAS
  - LANGUAGE
  - ENVIRONMENT
  - ISA
  - HARDWARE
- ADEQUATE MODULARITY TO SUPPORT TECHNOLOGY INSERTION
- NEED COMMONALITY OF HARDWARE FOR COST, OPERATIONAL AVAILABILITY, & INTEGRATED LOGISTICS

MILITARY COMPUTER FAMILY

- PROTOTYPES TO BE DELIVERED, EARLY '83
- SAME ISA IN THREE SIZES:
  - SUPERMINI
  - MICRO
  - SINGLE MODULE (COMPONENT OF MICRO)
- CHOOSE BEST OVERALL IMPLEMENTATION

INSTRUCTION SET ARCHITECTURE

NEBULA

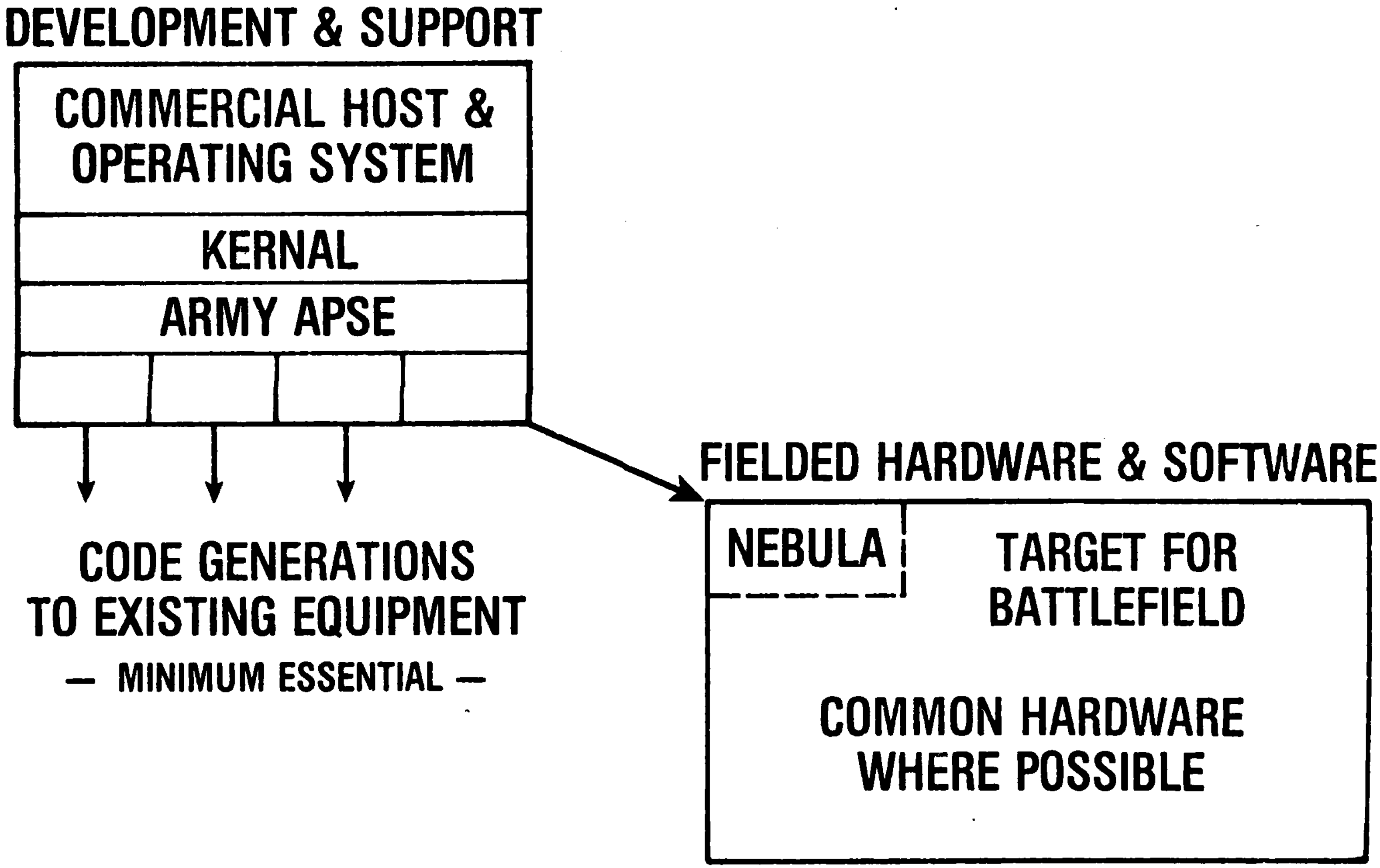
- JOINT AIR FORCE — ARMY CONTROL BOARD INDUSTRY PARTICIPATION
- TESTING IN 1983
- CODE GENERATOR FROM ALS

ADA®

- DOD APSE
  - ADA LANGUAGE SYSTEM (ALS)
    - ARMY'S INITIAL APSE
      - TEST
      - REHOST
      - USE
  - GOAL IS COMMON ENVIRONMENT FOR POST DEPLOYMENT SOFTWARE SUPPORT CENTERS

® ADA IS A TRADEMARK OF THE DEPARTMENT OF DEFENSE (AJPO).

ARMY DIRECTION



APSE: ADA PROGRAMMING SUPPORT ENVIRONMENT.

ARMY POLICY

- DOD 5000 SERIES
  - ARMY REGULATIONS AR 1000-1
AR 70 SERIES
  - DARCOM REGULATIONS 70-16
COMPUTER RESOURCE MANAGEMENT PLANS
- 1980 ASA (RDA) POLICY STATEMENT

STANDARDS VS SPECIFICATIONS

TAILORABLE?

CONFIGURATION MANAGEMENT!

— FOR INTEROPERABILITY —

COMMONALITY

EVOLUTION

MANAGEMENT

WHAT DOES STANDARDIZATION MEAN?

THE SOLUTION

". . .WHAT WE'RE ALL GOING TO HAVE TO DO IN THE NEXT TWO TO FOUR YEARS IS FIND A MINI OR MICRO-COMPUTER WITH GOOD COMMUNICATIONS CAPABILITY AND WRITE OUR OWN CONVERSION SOFTWARE, . . ."[3]

[3] COMPUTERWORLD, "SILENT AREAS OF OA CONVERSATION ADDRESSED," SEP. 20, '82.

THE STEPS:

"...THE FIRST STEP IS TO REDUCE THE NUMBER OF VENDORS USED BY THE ORGANIZATION..."[2]

"...THE NEXT STEP IS TO TIE THE THREE TYPES OF EQUIPMENT INTO A COMMON NETWORK..."[2]

" '...MANY OF YOU HAVE BEEN LED TO BELIEVE BY VENDORS THAT A LOCAL-AREA NETWORK IS A PANACEA FOR SOLVING INCOMPATIBILITY PROBLEMS...' "[2]

---

[2] COMPUTERWORLD, "SILENT AREAS OF OA CONVERSATION ADDRESSED," SEP. 20, '82.

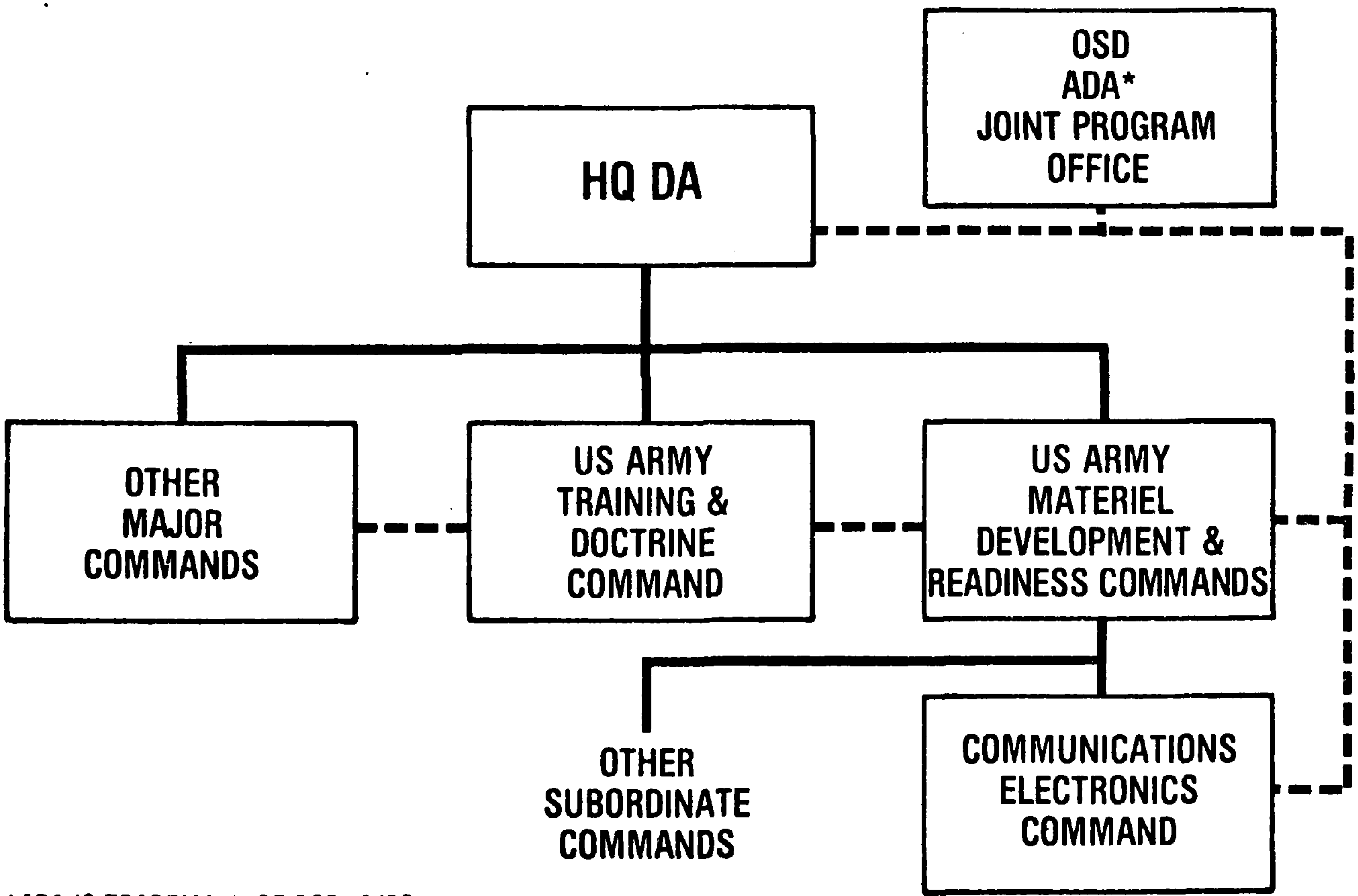
THE PROBLEM:

. . .“FIRST, INCOMPATIBLE WORD PROCESSING SYSTEMS OFTEN CANNOT FIND A COMMON LANGUAGE; SECOND, MANY COMPANIES’ OA AND DATA PROCESSING DEPARTMENTS APPARENTLY HAVE A HARD TIME TALKING TO EACH OTHER. . .”[1]

OA — OFFICE AUTOMATION.

[1]COMPUTERWORLD, “SILENT AREAS OF OA CONVERSATION ADDRESSED,” SEP. 20, ’82.

ORGANIZATION



*ADA IS TRADEMARK OF DOD (AJPO)

NAME: COL J. FRANK CAMPBELL

ENTRY ON ACTIVE DUTY: 3 June 1959

DATE OF RANK: 2 December 1980

BORN: 26 May 1935, Dunn, North Carolina

WIFE: Pat J. Campbell

CHILDREN: Robert 9 March 1962; Donald, 7 May 1964

BRANCH: Infantry

MILITARY SCHOOLING: Infantry Officers Basic Course, Sep 59
Airborne School, Oct 59
Ranger School, Dec 59
Infantry Advanced Course, Jun 64
Command and General Staff Course, 71

MILITARY ASSIGNMENTS (Past 15 Years)

JOB	OFFICE OR UNIT	COMMAND	LOCATION	YEARS
CO CDR, S-3 Air	3d Bde	1st Div	Lai Khe, VN	Jun 66 - Jun 67
Instructor	Dep of EE	USMA	West Point, NY	Jun 67 - May 70
Student	CGSC		Ft Leavenworth, KS	Aug 70 - May 71
Opns Adv	Binh Duong Province	MR-3	Vietnam	Jun 71 - Jun 72
XO	10th Bn, 2D AIT Bde	Ft Jackson, SC	Ft Jackson	Sep 72 - Jan 74
S-3	2d AIT Bde	Ft Jackson	Ft Jackson	Jan 74 - Dec 74
Div Chief	Inspector General	Ft Jackson	Ft Jackson	Dec 74 - Dec 75
Commander	15th Bn, 4th AIT Bde	Ft Jackson	Ft Jackson	Dec 75 - May 77
Div Chief	Automation Team, CSC, ODCSRDA	HQDA	Wash, DC	May 77 - Nov 80
Div Chief	Battlefield Auto Mgt Div DRA	DARCOM	Alex, VA	Nov 80 -

AWARDS/DECORATIONS

Legion of Merit, Bronze Star Medal (1), Army Commendation Medal (2), Meritorious Unit Citation (1), National Defense Service Medal, Vietnam Service Medal (5), RVN Gallantry Cross-BS/RVN Gallantry Cross Unit Citation/RVN Civil Actions Unit Citation, Senior Parachutist, Ranger, Combat Infantry Badge.

JOINT LOGISTICS COMMANDERS
JOINT POLICY COORDINATING GROUP
FOR
COMPUTER RESOURCE MANAGEMENT

Colonel John J. Marciniak, USAF

Rome Air Development Center
Griffiss Air Force Base, New York 13441
AC (315) 330-2165

BIOGRAPHICAL SKETCH

Colonel John J. Marciniak is the Chief of the Command and Control Division, Rome Air Development Center (RADC), Air Force Systems Command, Griffiss Air Force Base, New York. In this position he is responsible for emerging software technologies for application throughout the Air Force with special attention to command, control, communications and intelligence (C^3I) systems. Current activities under his cognizance are the development of the Ada support environment, knowledge based systems, the PAVE MOVER program, C^3CM Technologies, and Decision Aids. Col Marciniak was the Chief of the Information Sciences Division at Rome Air Development Center until a recent reorganization at RADC constituted the Command and Control Division. He came to RADC from HQ AFSC as the Director of Computer Resource Development Policy and Planning, where he was responsible for Air Force higher order language policy, implementation of computer architecture strategies, Air Force planning for the introduction of computer architecture strategies, and Air Force planning for the introduction of Ada. Previous assignments were with Headquarters United States Air Force, the Air Force Satellite Control Facility and the Electronic Systems Division. Col Marciniak received his BEE degree in 1957 from the City University of New York, and his MEE in 1969 from the University of Oklahoma. He is a graduate of the Air Command and Staff College, Maxwell Air Force Base, Alabama, 1973.

ABSTRACT

An overview of the Joint Logistics Commander's panel on Computer Resources Management is provided. The Computer Resources Management panel was organized to examine computer resource issues that are critical to the acquisition and support of defense systems. The past efforts of the panel, its current organization, and the current effort of the Computer Software Management subpanel to develop tri-service software policy, a Software Development Standard, a set of standard documents (Data Item Descriptions or DIDs) and changes to existing standards are detailed. The current industry/government review and implementation status of this effort is discussed. The author assesses the impact that these standards will have on industry/government and describes possible monetary savings that are possible.

INTRODUCTION

The Joint Policy Coordinating Group (JPCG) for Computer Resources Management (CRM) is a formally constituted panel under the management of the Joint Logistic Commanders (JLC): the four service commands; Air Force Systems Command, Air Force Logistics Command, U.S. Army Material Development and Readiness Command, and Naval Material Command. The activity of the JLC's is to insure cooperation and coordination of programs, eliminate unnecessary duplication and take advantage of mutual programs. There are currently 40 active panels dealing with subjects such as Automatic Testing to Simulators and Training Devices. Panels may be short term, dealing with an Ad Hoc issue, or continuing in nature dealing with longer term issues (of either a technical or policy nature) as long as the issue is of interest to the JLC's. The JLC's meet four times a year, at defined points, to review the progress of the panel's efforts. Each set of JLC's imparts their own collective personality on JLC activities. The current JLC's are extremely concerned with technology cooperation (they have just reconstituted the Joint Director of Laboratories panel) and meeting the Carlucci initiatives. They routinely meet with the Deputy Secretary of Defense after each quarterly JLC meeting.

The Computer Resources Management panel was formed to provide a focus for the JLC's on numerous embedded computer resources issues. The Charter, signed by the JLC's on 6 December 1977, provides that the JPCG-CRM coordinate policy, procedures, guidelines, and standards to implement effective management of computer resources in support of defense systems. Specifically the mission is to:

a. Provide a focal point for CRM activities within the JLC.

b. Coordinate and insure consistency in the preparation of new or revised regulations and standards to implement DOD Directives and Instructions on CRM.

c. Provide recommendations to the JLC on critical resource areas in CRM.

d. Provide a JLC focal point for coordinating service computer resour standardization programs.

Since its formation in 1977 the JLC has had some notable achievements. It prepared a JLC policy on the interservicing of operational software -- that is the utilization of common support resources in joint programs. That policy is presently being incorporated into the JLC joint software policy presently under review and discussed in more detail later. The CRM prepared a JLC position on a Government Services Administration action to reclassify several Federal Supply Code (FSC) equipment groups as Automated Data Processing (ADP) Equipment (FSC Group 70). The effect of this action was to place FSC groups 66 and 74, dealing with such items as test equipment, optical equipment, etc. under the procurement practices governing ADP and removed from normal supply acquisition practice. The JLC's got involved and the JLC-CRM developed a joint position which was forwarded to the Service Chiefs. The action itself has been deferred.

Perhaps the most significant action was the study of an Ad Hoc CRM panel on the acquisition of general purpose ADP integral to, or in direct support of, Defense Systems. The Ad Hoc group studied the acquisition procedures of the three services and developed recommendations for special acquisition procedures of ADPE integral to Defense Systems. These recommendations, presented to the Office of the Secretary of Defense (OSD), were timely since several key Policy Directives were under revision. Members of the CRM Ad Hoc group worked directly with members of the OSD drafting a new DODD 5000.29, Management of Computer Resources in Defense Systems. The policy provided that the acquisition of ADPE integral to Defense Systems be integral to the overall weapon system acquisition and not broken out for separate approval and procurement. Notably it was a short time later that the Warner Amendment was passed by the Congress providing for the above.

The current organization of the CRM is shown in Figure 1. Of the four panels, the Computer Software Management (CSM) subpanel has been in existence the longest, and it is the activity of this panel that is the primary emphasis of this paper. The Professional Development Subpanel objectives are to evaluate current professional development programs within the JLC's, provide recommendations for new and existing programs, and in general to propose better methods to enhance professional development of computer resource management personnel. The Technology subpanel was created with the objective of providing tri-service cooperation and even coordination of technology issues common to the services. Of particular interest are those areas that do not enjoy major interest or funding in any one service. It is believed that cooperative programs in these areas could produce a critical mass of technology effort that any one service is currently not able to sustain. Examples of such technologies are Direct Execution of Higher Order Language Machines and Software Requirements Management. Once an area of cooperation is identified, the panel would provide for cooperation either through arranging a formal tri-service program or insuring cooperation amongst the services.

The Terminal/Peripheral Standardization panel arose from a JLC initiative to evaluate potential ideas for further Joint Services Cooperation. The commands were canvassed and the JLC-CRM was tasked to evaluate nine issues. Of these nine the CRM decided that the issue of standardizing on terminals and peripherals used in tactical environments deserved further study and organized an Ad Hoc panel to investigate the issue. The panel is chaired by Al Selgas of the NAVMAT and the effort is expected to conclude with recommendations to the CRM in September 1983.

THE COMPUTER SOFTWARE MANAGEMENT (CSM) PANEL.

The most successful effort of the JLC-CRM, and perhaps the most significant, is under the responsibility of the CSM, that is the preparation of a standard tri-service software management policy and acquisition procedures. The effort took genesis in a very successful workshop on software quality sponsored by the JLC-CRM in April of 1979. Termed Monterey I, since it was held at the Naval Post Graduate School, Monterey, California, a group of 100 selected personnel from government and industry (80/20% mix) organized into four panels to discuss software quality, software acceptance criteria, software documentation and software acquisition/development standards. Their recommendations to develop a common software acquisition-development policy, a single unified set of acquisition and development standards, a comprehensive set of documents (Data Item Descriptions) and software acceptance criteria, and to define government practices, procedures, methodologies and responsibilities for software quality assurance were documented in a JLC report,[1] briefed to the JLC's and approved by them for implementation.

Since that time the CSM has been arduously at work carrying out their charge. A draft tri-service software development policy exists, and has been circulated amongst the JLC commands for comments. Those comments have been incorporated and the policy is ready for coordination. An early 1983 date is scheduled for implementation. While the policy will be binding on the JLC's only, it will be submitted to the respective services with the recommendation that it be issued as a joint service policy. The policy modifies the existing development cycle in a notable way by introducing several additional milestones (Figure 2). A preliminary design review will be held, ligitimizing a practice already in existence, and backed up with a document which will fold into the B-5 specification (Software Detail Design Document). The area of requirements will receive more attention. Called out as a special problem area in every major study of the "Software Management Problem" it is hoped that this special emphasis will solve the problem of requirements creep and maintain configuration control of requirements. It is important to realize that this software development "waterfall" represents an evolutionary change to the present lifecycle and not a completely new methodology.

The lifecycle will be backed up by a set of standard documents (Data Item Descriptions), a new Software Development Standard, a new Software Quality Standard and commensurate changes to existing standards such as MIL-STDS 483, Configuration Management Practices for Systems, Equipment and Computer Programs; 490, Specification Practices; and 1521, Technical Reviews and Audits for Systems, Equipments and Computer Programs. Action has already been taken to coordinate these changes with the appropriate Offices of Responsibility. The area has been given special emphasis by the DOD through the Defense Material Specification and Standards Office (DMSSO) in the preparation of a new standardization area for embedded computer resources. An Embedded Computer Resources Standardization Program plan[2] has been prepared incorporating the planned changes currently under preparation by the CSM.

BENEFITS.

The overall effort is expected to not only make improvements in the computer resource management process, but through streamling and standardizing the computer resource acquisition process achieve cost savings. While these are always difficult to quantify, we believe that we can make some projections based upon studies that have been conducted of the software development process and associated costs. It is estimated that approximately 3-7 billion dollars a year are spent on embedded systems software development. Assuming that documentation costs are about 25 percent of overall software development costs and based upon an expected savings of 5-10 percent of documentation costs, we forecast that we can accrue a savings of at least 40 million dollars per year. Increased emphasis on requirements also portends cost savings. Wolverton,[3] in 1977, postulated the cost of fixing errors at different life cycle phases to be:

Requirements Definition	- $	195
Design	- $	489
Coding and Checkout	- $	977
Test and Integration	- $	7,136

Extrapolating these figures into 1982 with an inflation rate of 10 percent per year we can show that an error in the requirements definition phase which costs $380 to fix would cost $13,906 to fix in the Test and Integration phase.

The JLC tri-service standards have been subjected to a broad industry and government review. Between now and the end of the year comments will be incorporated into the drafts and the formal process of coordination will take place. We hope to start using the products in 1983, and believe it feasible to use the drafts even earlier.

SUMMARY.

The JLC and the CSM are not sitting still resting on past laurels. In June of 1981 a second workshop, termed Monterey II, was conducted. While in part the workshop reported on the implementation status of the ongoing effort of the CSM, other issues were studied. Five panels looked at Documentation, Hardware/Software/Firmware Configuration Item Selection Criteria, Standardization and Accreditation of Computer Architectures, Estimating Software Costs, and Software Reuseability. A report was produced and some follow-on actions are planned.[4] Specifically, Hardware/Software/Firmware Selection Criteria will be studied and the CSM will prepare a Software Cost Estimating Guidebook.

The CRM is working an area that is important to the JLC's and one that can benefit from increased services cooperation from technology to management. Past accomplishments, notably the preparation of the new DOD Directive 5000.29, attest to the success of CRM activity. With the introduction of the tri-service software policy and standards an important milestone will be reached--for the first time the DOD will have achieved commonality of practice in a critically important systems area.

JLC-CRM ORGANIZATION

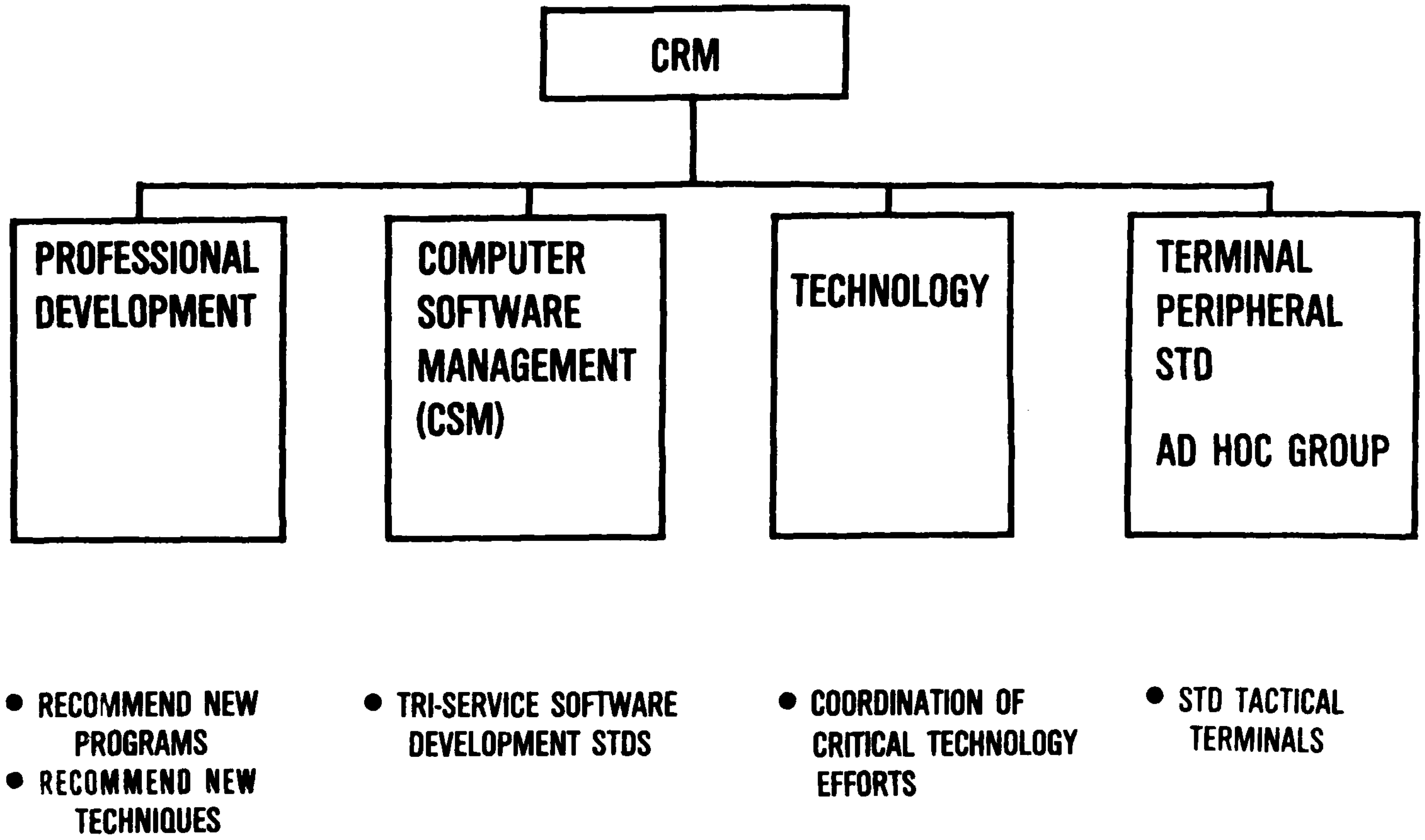


FIGURE 1

JLC SOFTWARE ACQUISITION MODEL

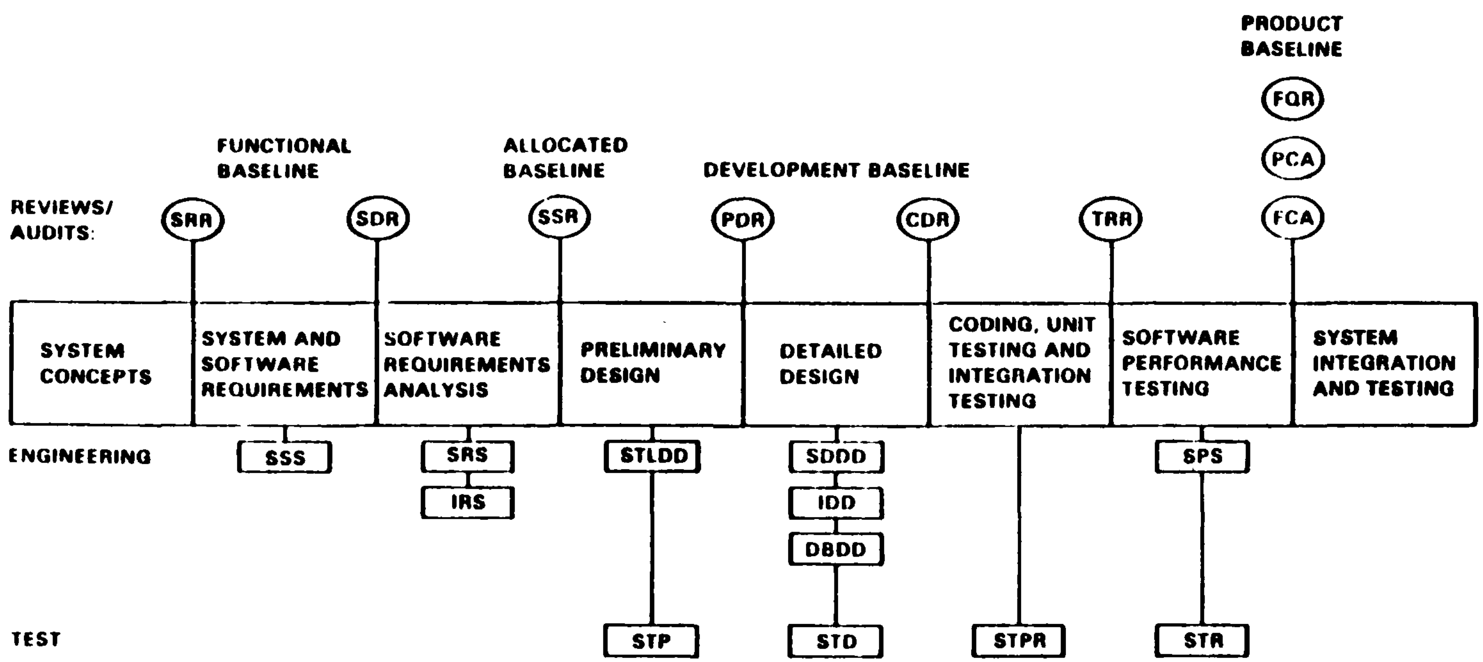


FIGURE 2

DEFINITIONS

SRR	System Requirements Review
SOR	System Design Review
SSR	Software Specification Review
PDR	Preliminary Design Review
CDR	Critical Design Review
TRR	Test Readiness Review
SSS	System/Segment Specification
SRS	Software Requirements Specification
IRS	Interface Requirements Specifications
STLDD	Software Top Level Design Document
STP	Software Test Plan
SDDD	Software Detail Design Document
IDD	Interface Design Document
DBDD	Data Base Design Document
STD	Software Test Description
STPR	Software Test Procedures
SPS	Software Product Specification
STR	Software Test Report
FQR	Formal Qualification Review
PCA	Physical Configuration Audit
FCA	Functional Configuration Audit

REFERENCES

1. Joint Logistics Commanders: Proceedings of the Joint Logistics Commanders Joint Policy Coordinating Group on Computer Resources Management, Computer Software Management Subgroup Software Workshop; 21 Aug 1979; NTIS AD 103485.

2. Department of Defense: Draft Embedded Computer Resources Standardization Program Plan; Directorate, Defense Material Specification and Standards Office, Falls Church, VA 22041; June 81.

3. Wolverton, Ray W.: The Cost of Developing Large-Scale Software, 1977 Annual Software Life Cycle Management Workshop; Aug, 1977, pp. 131-152; DTIC AD-A053014.

4. Joint Logistics Commanders: Proceedings of the Joint Logistics Commanders Joint Policy Coordinating Group on Computer Resources Management Computer Software Management Subgroup Second Software Workshop; 1 Nov 1981; NTIS ADA 109441.

MANAGING TACTICAL EMBEDDED COMPUTER RESOURCES (TECR)

IN THE NAVAL SEA SYSTEMS COMMAND

J. C. Stewart

and

T. L. Wallis

Naval Sea Systems Command

Washington, D.C. 20262

(202) 692-8204

Abstract

The Naval Sea Systems Command (NAVSEA) is charged with cradle-to-grave support of over 300 separately nomenclatured TECR items, employed in virtually all combatant classes to support signal processing, communications, navigation, and command and control. The wide variety of equipment supported, and variation in the requirements of over 150 different users poses a significant management challenge.

This paper will address the challenges associated with the management of TECR in NAVSEA, the techniques used to meet these challenges, and some of the lessons learned.

INTRODUCTION

Experience has confirmed that standardization of Tactical Embedded Computer Resources (TECR) results in the improved operational availability of deployed US Navy systems because of the commonality of spare parts and the availability of trained maintenance personnel; for similar reasons, hardware standardization reduces overall system life cycle cost. However, the limitations of current TECR may result in increased system life cycle costs for the following reasons:

a. A more complex system architecture is needed to overcome performance and capacity limitations.

b. Complex software designs are necessitated by the complex system architecture, thus adding significantly to software support costs.

c. Nonstandard computers are often needed to meet user system requirements resulting in higher costs for acquisition, maintenance, and logistics support. This practice also results in proliferation of nonstandard programming languages for software development and invariably increases software life cycle support costs.

To realize the benefits of a standards program, the Chief of Naval Material (CNM) has initiated policies and procedures covering all aspects of the use of embedded computer resources in Navy systems. They have been promulgated in the form of CNM Tactical Digital Standards (TADSTANDs), instructions and notices, military standards, and letter directives/guidance as appropriate.

Further, the Navy has embarked on a major program to develop and acquire state-of-the-art successors for the AN/UYK-7, AN/UYK-20, and AN/UYS-1 standards. The replacements will be capable of meeting embedded computer system requirements into the 1990s. As successor computers are developed, the Navy must take advantage of technological advancements on a competitive basis, while maintaining the significant investment in support and operating software. In addition, as the Navy improves upon and replaces standard embedded computer resources, technology upgrades must be introduced in a controlled evolutionary manner to best meet Fleet requirements.

BACKGROUND

In the 1950s, most of the computers used aboard Navy ships were special-purpose, analog computers designed to solve specific information-flow problems. In 1958, the Bureau of Ships designed a solid state shipboard digital computer to handle and manipulate the operational data that had previously been disseminated and displayed manually. This was the start of the Navy Tactical Data System (NTDS). These computers were complete units as they stood and could be configured in variable quantities to meet the differing mission profiles of several classes of ships. Most of the initial NTDS computers (CP-642) are still in the Navy inventory today.

From this modest beginning, the spread of the digital computer aboard ship proceeded rapidly, first with the expansion of NTDS and then into many other shipboard systems. By 1970 it was recognized that a proliferation of different types of digital computers in the Navy would have to be controlled in the interest of efficiency in logistics, training, reliability and maintainability, configuration control, system interoperability, and software support. As an initial effort, the AN/UYK-7 computer, the CMS-2 High Order Language (HOL), and certain NTDS operator consoles were designated as standards. This was followed by the competitive acquisition of a standard minicomputer (AN/UYK-20), cartridge magnetic tape unit (AN/USH-26), Alphanumeric display (AN/USQ-69), computer display set (AN/UYQ-21) and the development and control of standard HOL compilers and other support software. Standards were also developed and promulgated for embedded computer systems software documentation and quality assurance in SECNAVINST 3560.1 and MIL-STD-1679. These steps provided an initial respite from the proliferation problem. In recent years additional action has been taken to strengthen the Navy's management of Tactical Embedded Computer Resources (TECR).

In 1978, the Chief of Naval Material established the Tactical Embedded Computer Program Office (TECPO), MAT 08Y, as the NAVMAT program manager for embedded computer resources in the Headquarters, Naval Material Command. In 1979 MAT 08Y assumed the additional policy and standardization functions of the former Tactical Digital Systems Office (TADSO), MAT 09Y2. [illegible] the Deputy Chief of Naval Material for Acquisition [DCNM(A)], MAT 08Y is the Navy's Principle Development Activity (PDA) for research, development and acquisition of standard embedded computers, support software, and related digital equipment. MAT 08Y is also responsible for establishing NAVMAT policies concerning development and use of embedded computer resources, assigning SYSCOMS as Development Activities (DAs), controlling budget resources for standard embedded computer resource R&D projects, and appraising project results.

The Navy Shipboard Tactical Embedded Computer Resources Project Office (PMS 408), in the Naval Sea Systems Command, was formally established by NAVSEA Instruction 5400.59 on 14 November 1979. PMS 408 is the Development Activity responsible to the CNM for development, acquisition and life cycle support of standard embedded computer resources assigned to NAVSEA by MAT 08Y. This currently includes the AN/UYK-43 and AN/UYK-44 computers (successors of the AN/UYK-7 and AN/UYK-20 computers) and associated support of software; acquisition, configuration control and life cycle management of existing and future computers and peripherals for shipboard applications; and implementation of Ada (DOD standard HOL) for Navy use. PMS 408 responsibility also includes commodity management functions for the AN/UYK-7 and AN/UYK-20 standard computers and associated support software, and all standard shipboard tactical displays and peripheral equipment. Development Activity responsibility for the AN/UYS-2 Enhanced Modular Signal Processor (EMSP) has also been assigned to PMS 408. The PMS 408 Project Manager reports directly to the Deputy Commander for Combat Systems (SEA 06) who has the designated authority and responsibility for shipboard embedded computer systems.

POLICY/PHILOSOPHY

It is Navy policy to require the use of designated standard TECR unless use of a nonstandard can be shown to be in the best interest of the Navy. Presumably this means that if the designated standard TECR cannot compete in the DOD marketing arena, its use will not be required. NAVSEA, therefore, has adopted an operating plan of expending its efforts towards ensuring that Navy standard TECR is competitive rather than acting as a policeman for Navy policy. Figure 1 shows the types of effort which must be expended to support the user of a Navy standard during the four phases of his system. If the support provided in any phase is inadequate, then the user has legitimate cause to pursue the use of nonstandard products.

INFORMATIONAL SUPPORT

The chances that a given system will not use a standard product is greatest during the conceptual phase of the program, and once that decision is made it becomes virtually impossible to change that decision later due to "sunk costs." Programs may choose to use nonstandard equipment for any of the following reasons:

a. An unawareness that policy applies to him: The advent of the microprocessor has resulted in the automation of many systems outside the traditional electronics and weapons area. For instance, controllers for aircraft elevators and missile tube environment now use embedded computer resources. These project offices have no prior experience with TADSTANDS and are somewhat incredulous at the idea that TADSTANDS are applicable to them.

LIFE CYCLE SUPPORT REQUIRED FOR STANDARD TACTICAL EMBEDDED COMPUTER RESOURCES

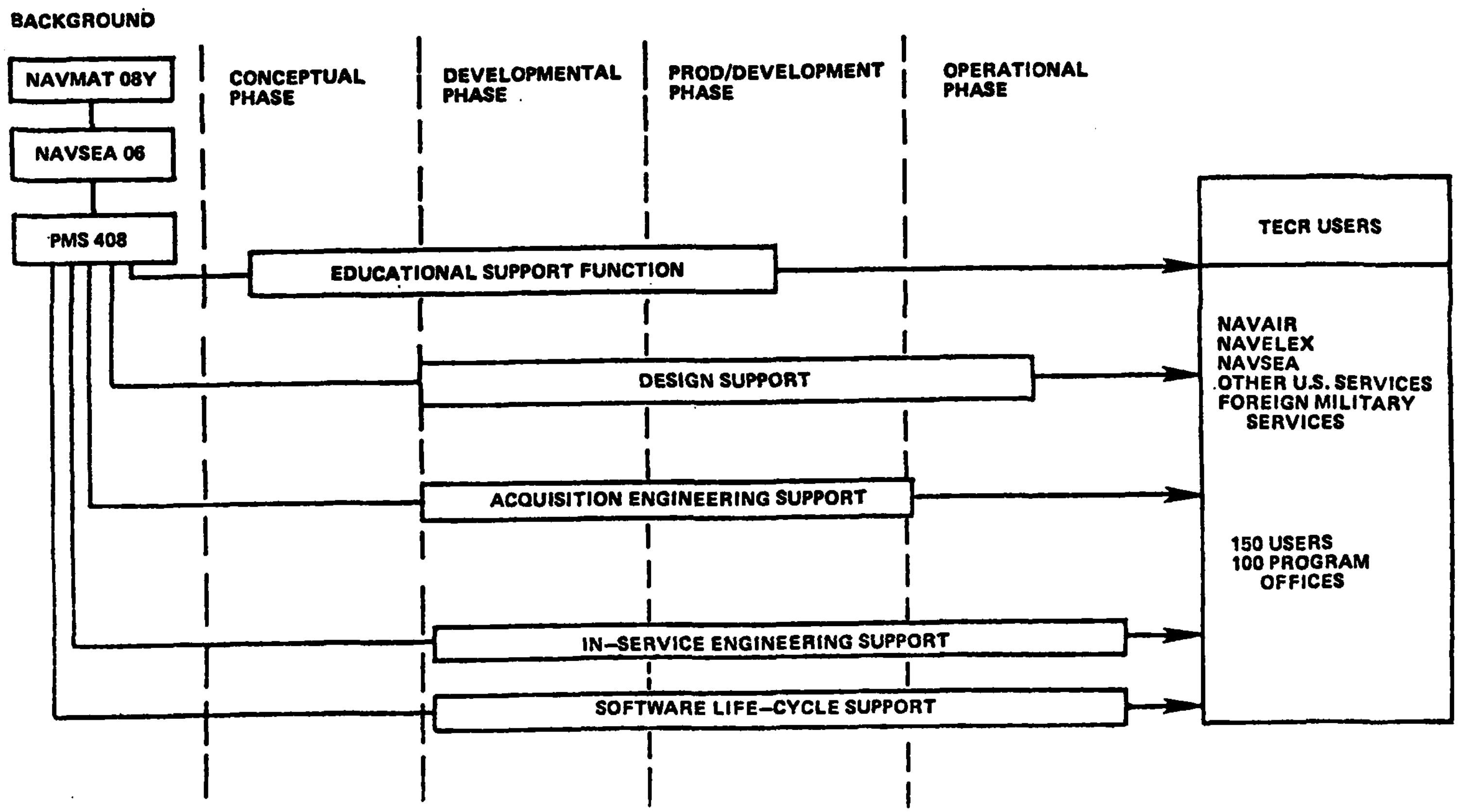


Figure 1

b. A lack of awareness regarding the capability of Navy standards: Many projects simply do not know what the performance characteristics are for Navy standards. There is a general belief that standards must cost more since they are general (vice specific) in nature, or are unable to meet commercial performance standards.

c. A lack of awareness of the hidden cost of using nonstandards: During the conceptual phase, system requirements tend to be "pie-in-the sky" and demand the use of future technology. In many cases, standards are dismissed out of hand as being of obsolete technology and unable to meet planned or perceived requirements. During the development phase, as a result of production delivery requirements, there is a tendancy to utilize the readily available current technology. An explanation of the added ILS costs associated with the use of a nonstandard may hasten this retreat to the real world.

The support, therefore, which must be provided during the conceptual phase of a project is largely educational. Users of embedded computer resources must be made aware of policy. They must be made aware of standard TECR performance capabilities, and, finally, they must be made cognizant of the logistics cost penalty associated with using a nonstandard TECR. At present the three major mechanisms used to provide information to users are listed as follows:

a. As part of the course provided to NAVSEA employees on Integrated Logistics Support, PMS 408 conducts training regarding the Navy policy requirements and the logistics penalties associated with the use of nonstandards.

b. An annual Navy Standards Users Conference is held to determine future users' requirements and to acquaint users with plans to improve or introduce new standards. The next conference is planned for Spring 1983 in Washington, D.C.

c. Marketing by Navy TECR suppliers is often the most effective educational method. If a potential DOD user's prime contractor can be convinced that the use of a standard TECR is in his best interest, it is not too difficult to convince the potential user.

The introduction of the AN/UYK-43 and AN/UYK-44, with the attendant high visibility competition, transition planning requirements, and competing marketing organizations, has significantly raised the awareness of TECR policy in the Navy.

DESIGN SUPPORT

The functions which must be provided under design support can be categorized into two areas, Project Design Support and TECR Design Support.

a. Project Design Support: New standards developments such as the AN/UYK-43, AN/UYS-2, AN/UYK-44 MRP and the Ada language, will have significant impact upon how Navy Combat Systems are designed. In the past, the user was presented with a package of installation control drawings for the hardware and a user's manual for the software. In most cases that proved sufficient. However, there were some problems:

(1) Restrictions on memory, processor speed and I/O ports led project managers to allow their contractors to program in assembly language in order to reduce memory usage or to accommodate special timing or I/O requirements. Although assembly language usage may allow more efficient use of available memory and performance capabilities, it increases the complexity and cost of programs during development and increases software maintenance costs throughout the system's life cycle. Further, unless your programmers are very experienced, you may well pay an efficiency penalty without realizing it.

(2) Interoperability among shipboard systems has often been considered as an after-the-fact element. As a result, costly modifications to operational software have been required, and ad hoc solutions have been developed for intersystem digital communications. Protocols developed in such situations have often been costly to implement and difficult to maintain and extend for new interoperability requirements. Intercomputer bus communications and protocol standards must be established.

(3) Imperfect understanding of TECR capabilities caused poor system designs, further hampering the user's ability to produce a viable, efficient, cost-effective system.

The AN/UYK-43, AN/UYK-44, and AN/UYS-2 TECRs provide a solution to the first of these problems by their very nature, and a potential solution to the second because of their flexibility and design for growth. However, they also enhance the opportunity to make design mistakes. The necessity to provide information and assistance to the user in his design efforts is well understood. A critical portion of the Ada development effort will be the procurement of a training package which teaches not how to code in Ada, but how to design based on Ada. The AN/UYK-44 MRP will have two technical manuals: the traditional maintenance-oriented manual and an equipment designer-oriented manual to facilitate its embedding. In the case of the AN/UYK-43, AN/UYK-44, and AN/UYS-2, laboratories supporting the development of these equipments also provide support to major potential users. Finally, TECR contracts are designed to provide technical engineering support upon request.

b. <u>TECR Design Support</u>: The march of technology is the most significant challenge facing the Navy standards program. Many think that standardization and technological advancement are incompatible. While standardization complicates the influx of technology into Navy systems due to the necessity to meet new users needs without perturbing old users, these two objectives are not mutually exclusive. The lack of prior planning and funding have caused most problems in the past.

The shortcomings of the AN/UYK-7 and AN/UYK-20 computers were recognized in the mid-70s. The lack of a continuing budget line following the initial development precluded any action. Attempts to establish this needed funding for product improvement ran afoul of sole-source problems resulting in the competitive procurement of the AN/UYK-43 and AN/UYK-44. This is not to say that there is anything wrong in the procurement of this new equipment, however, improvements to the AN/UYK-7 and AN/UYK-20 were delayed pending the introduction of their successors.

The AN/UYK-43 and AN/UYK-44 will not eliminate AN/UYK-7s and AN/UYK-[illegible] [illegible] not economically feasible to replace some 4500 existing [illegible]

Future improvements to the AN/UYK-7 and AN/UYK-20 will be the cost-effective method of meeting the emerging requirements of systems which utilize the current standards. However, funding remains unbudgeted for this effort. It is expected that transition planning results will provide the justification needed to obtain these funds.

Similar problems with the AN/UYK-43 and AN/UYK-44 will, hopefully, be avoided. Funding is currently budgeted to begin definition of successor computers, and funding has been requested to provide for preplanned product improvements.

The designer of TECR is constantly on a knife edge. He must choose between the newest technology and potential impact upon existing users. Thus, the initial design of the TECR must be sufficiently flexible to allow the infusion of state-of-the-art technology without impact to existing systems and/or their software. In the case of the AN/UYK-20, the Navy has had great success in ensuring modifications which are upward compatible, less so with the AN/UYK-7. This relative degree of success reflects a difference in the modularity of the design with respect to growth. The AN/UYK-43 and AN/UYK-44 are specifically designed to capture existing software and a major concern of design reviews was to ensure that the flexibility existed to insert future technology.

ACQUISITION ENGINEERING SUPPORT

The TECR commodity manager must ensure that users are delivered the equipment they require in a timely manner. Further, he must ensure that cost to the user remains relatively stable with increases being limited to inflation.

a. TECR Availability: TECR are procured through firm fixed price, production requirements type contracts. In many cases the contracts cover multiple years of procurement. This means the contractor is required to deliver an unspecified quantity of equipment over the life of the contract. Normally, delivery is required within a specified time after an order is placed. To date no contractor has defaulted with respect to a delivery. When the product is delivered late it is the result of Navy problems. In many cases, the user waits until the 11th hour to place his order. Sometimes this is beyond his control, but the result is still such that it is impossible to meet his needs. The delivery times, After Receipt of Order (ARO), for the AN/UYK-7 and AN/UYK-20 computers are approximately 12 months and 6 months respectively - extremely good for a DOD procurement action, but a finite time none-the-less.

Another reason which causes an inability to meet a required GFE date results from delays introduced in cost negotiations which preclude the placing of the order in a timely manner. This occurs when contracts come up for extension.

b. TECR Costs: Standardization leads to a sole-source situation. This does not necessarily mean that TECR costs could be significantly reduced by open competition. There are other factors involved which tend to limit interest in TECR contracts, primarily the fact that off-the-shelf commercial equipment is not acceptable. The AN/UYK-43 and AN/UYK-44 were open competitions, yet only two contractors chose to bid. TECR costs are controlled through the contract negotiation process. The initial competition for the contract produces reasonable base prices. By requiring the contractor to justify increases over this price in the negotiation process, unreasonable cost increases are prevented. Of course, these negotiations are not always easy, and delays in the delivery of GFE can occur. The object, therefore, has been to reduce the number of negotiations which must take place.

Our oldest standing contracts have been for AN/UYK-7(V) computers and for tactical displays. These contracts must be negotiated yearly and are highly dependent on quantities being procured. Therefore, fluctuation in user requirements affect build rate and thus cost; thereby making these contracts very difficult to negotiate. The AN/UYK-20 contract, on the other hand, is fixed on a five-year cycle. The first two years are bid and negotiated at a fixed price with automatic adjustment factors for inflation and ordering quantities. These adjustments are always for future orders and never retroactive to existing orders. At the end of the first two years, prices are renegotiated to adjust for any factors not taken into account by the automatic features, and then a fixed price is set for the remaining three years of the contract. At this point the cycle is repeated. The AN/UYK-43 and AN/UYK-44 contracts are patterned after the AN/UYK-20 contract.

IN-SERVICE ENGINEERING SUPPORT

The In-Service Engineering Support function can be described in two words: Correct and Control.

a. Correctional Support: The use of TECR is predicated on the proposition that it improves system availability, reliability, and lowers cost. Program Managers tend to be skeptical of these claims. Thus, any problems which occur are blown out of proportion. It is vital to the success of the Navy's standardization policies that rapid and effective correctional action be taken in response to reported problems. The early AN/UYK-20s suffered greatly from intermittent problems. Analysis indicated these problems resulted from the design of some circuits improperly accounting for worse case situations and the lack of sufficient margins testing. The designs were corrected and improved testing procedures were implemented, but, most important, the deficient units were replaced at no cost to the users. This example demonstrates the type of support which must be provided. Unfortunately, the Navy's ability to provide this type of support for all TECR does not exist.

By NAVSEA charter, PMS 408 is tasked with providing equipment-level engineering support in the following areas: design, safety, test support, technical documentation, performance and maintenance data analysis, maintenance engineering, installation, fleet engineering support, training and manning, integrated logistics support, configuration management, data management, test equipment supply support, and repair facilities. To accomplish this task, NAVSEA requires personnel resources far exceeding its headquarter's staff, specifically an In-Service Engineering Agent (ISEA). ISEA's, however, do not work without pay and no funding line exists for the life cycle support of TECR.

How, then, has the Navy's TECR standards program managed to survive the past decade? The answer is fairly simple. Major system users have assumed equipment-level support engineering functions and contractor support has be bought within the funding provided for hardware procurement. The [illegible] tion significantly undermines the logistics savings achieve[illegible]

TECR standards. The latter becomes impossible when the equipment is no longer being sold. The establishment, therefore, of a life cycle support funding line for TECR is the highest priority for PMS 408.

b. Control Support: The TECR manager must provide for the users a product which is stable and cannot change without their knowledge and approval. Any process implemented must be sensitive to the potential of causing cost increases on user systems. Conversly, the process cannot be so rigid as to preclude the injection of new technology to meet the requirements of new users.

The consolidation of life cycle responsibility for shipboard TECR products into one NAVSEA systems command organization has assisted in configuration management of TECR by allowing the establishment policy and a set of standard procedures for the communication and coordination of CM within the entire community of TECR users. It has facilitated the review of requested engineering changes to ensure that all aspects of impact are considered including maintenance, training, software, and other TECR configuration items.

The establishment of a rigid standard policy on when to approve an engineering change is impractical. Each request must be considered on its own merits. General guidelines include:

(1) The change should be upward compatible or implemented as a nonmandatory change.

(2) The change should be of demonstrable benefit to performance, maintainability, or reliability.

(3) The change should benefit more than one user.

The approval of engineering changes based upon the desires of only one user should not be undertaken even when the user is willing to fund the change. Since these changes are not likely to be purchased by other users, the result is a unique configuration which not only results in increased life cycle cost, but produces no general benefit to the Navy as a whole. When a user's requirements cannot be met by a design which is beneficial to all users, or the user's requirements cannot be adjusted to meet the common design, then it is appropriate to use a nonstandard. In the past it has not always been possible to separate the greater benefit from the individual user need because the individual user has been funding the development or maintenance of the TECR. The establishment of sound configuration management requires the TECR agent to be financially independent of the user project. While NAVSEA has achieved this goal in the development of the AN/UYK-43 and AN/UYK-44, the Navy's inventory of standard displays and peripherals also requires updating. The identification of a first user or group of first users will assist in achieving the needed redevelopment funds, however, control of the funds must continue to be placed in the engineering hands of a single TECR commodity manager to ensure that standardization objectives are met.

SOFTWARE LIFE CYCLE SUPPORT

When the first Navy solid state general purpose digital computers were introduced, hardware costs still outweighed software costs. The result has been a slower recognition of the advantages of and the need to provide standard support software to reduce system software development and maintenance costs. The support software that PMS 408 manages can be classified as Compile Time and Run Time.

a. Compile Time Software: This classification, by PMS 408 definition, includes compilers, editors/librarians, system generators, simulators, debuggers, and some utility software. It is this software that is used to generate applications software. In the early 1970s, the Navy developed an interactive real-time program generation system, known as Share-7. It executes on the AN/UYK-7, the target computer for the software it generates. This development, in its time, represented a major step forward from the batch processing program generation centers that it replaced. However, Share-7 still suffers some major drawbacks:

(1) It can only run on the AN/UYK-7. Since most contractors do not own this host, it is a GFE cost to the user.

(2) Contractors do have extensive commercial computer program generation facilities. Their personnel are familiar with the use of these commercial systems and therefore, requiring the use of Share-7 often involves an extensive learning experience.

(3) The tools available to assist programmers on commercial systems tend to stay a step ahead of those on Share-7 because of industrial competition.

Concurrent with the introduction of the AN/UYK-20, the Navy took another step forward with the introduction of Machine Transferable AN/( ) Support Software/Medium (MTASS/M). This set of program development components is transportable across a wide range of commercial computers. This then allows contractors to utilize the development support environment with which they are familiar and to utilize the tools they have available on these systems. The MTASS/M system is being upgraded to support the AN/UYK-44. An MTASS/L system is being developed to support the AN/UYK-43 and AN/UYK-7. Machine Transferable AN/( ) Support Software/Large (MTASS/L) will have the same attributes as MTASS/M with one addition. Those features of the system which are unique to the host computer are being developed as a set of Common Interface Routines (CIR). This provides two benefits: first, the rehosting time has been reduced from several weeks to a few days; and second, by following the CIR interface specification, the user can add to the MTASS/L system any special tools he feels necessary for his development. Once the MTASS/L CIR designs are tested, they will be incorporated into MTASS/M.

b. Run Time Software: Run time software consists of executives, operating systems, and utilities. Here again, the predominance of hardware in the mind of the Navy has caused problems. In the 32-bit world, the hardware was developed and released before any standard run time software was available. The result has been the multiple development of executives and operating systems. Learning from experience, the AN/UYK-20 had a standard [illegible] available on release (SDEX-20). This executive is in use on over 100 systems. A second standard executive now exists as a result of competitive procurement to support the AN/AYK-14. This newer executive (SDEX-M) is the currently preferred executive and to this end, as part of the AN/UYK-44 upgrade, it has been restructured to better allow the user to build an operating system around it. The lack of a 16-bit operating system has prevented the 100 percent acceptance and use of the 16-bit standard executives. While it was intended to provide a CMS-2 operating system for the AN/UYK-43 and AN/UYK-44, this funding has now been diverted to develop an Ada based system.

THE FUTURE

Two major issues face the Navy's Shipboard Tactical Embedded Computers standardization. In the near term, the Navy must come to grips with transition from current standards to planned standards. In the longer term the Navy must establish a policy with respect to the control of microprocessors.

a. Transition: Navy planning calls for all tactical systems requiring standard computers after FY83 to use the AN/UYK-43 or AN/UYK-44 or to submit justification as to why the continued procurement of AN/UYK-7 or AN/UYK-20 computers is required. At no time has the backfitting of current AN/UYK-7 and AN/UYK-20 computers been a consideration. In consonance with this line of reasoning, there exists four reasons why transition should take place:

(1) Accommodate expanded operational requirements.

(2) Improve Reliability and Maintainability.

(3) Achieve significantly lower life cycle costs.

(4) Avoid logistics obsolescence.

For tactical systems that currently use standards, the issue of a shortfall in capacity to meet operational requirements is the only true motivator. The cost of logistically maintaining a tactical system with two different computer configurations outweighs any economic benefits that might be achieved by transitioning to the new computers in mid-project.

It is hoped that Navy planners will accept the fact that procurement of AN/UYK-7s and AN/UYK-20s will continue into the late 1980s, and will provide the resources necessary to refurbish and maintain these computers until their true operational demise in the 21st century.

b. Microprocessors

The AN/UYK-44 Militarized Reconfigurable Processor is the low end of the Navy standards product line. It is not a microprocessor. In today's technology, microprocessor-based systems can be designed and built that will outperform the AN/UYK-44 and, at the same time, will cost less. The distributed design of these systems with microprocessors performing small dedicated tasks makes the use of the AN/UYK-44 impractical. This is not a condemnation of the AN/UYK-44(V). It does mean that a substantial hole remains in the Navy

Standards Program. A significant penalty is being paid in software development and maintenance costs because no effective policy exists for microprocessor applications.

It has not yet been fully ascertained whether microprocessors represent only a software cost problem or pose a logistics problem as well. If the problem is purely one of software support costs, then the answer lies in Ada program support environmental development in conjunction with the accreditation of more than one processor type to execute a standard Navy Instruction Set Architecture (ISA). At the present time, the concept of "accreditation" has not been accepted because it does not resolve the logistics supportability, maintainability, and training issues faced by the current and planned Navy TECR. However, these problems tend to be eliminated or are significantly reduced when the processor element is confined to a single replaceable unit that is so reliable debilitating failures seldom if ever occur and the unit is inexpensive enough to be discarded when such errors do occur.

It is not clear that state-of-the-art microprocessors or TECR system designs have reached the level required to say that the logistics aspects can be ignored. However, the march of technology is certainly in that direction. In the interim, the Navy requires a policy to control the proliferation of microprocessors in deployed systems.

GLOSSARY OF ABBREVIATIONS

ARO	After Receipt of Order
CIR	Common Interface Routines
CNM	Chief of Naval Material
DA	Development Activity
DCNM(A)	Deputy Chief of Naval Material for Acquisition
DOD	Department of Defense
EMSP	Enhanced Modular Signal Processor
HOL	High Order Language
ILS	Integrated Logistic Support
ISA	Instruction Set Architecture
ISEA	In-Service Engineering Agent
MRP	Militarized Reconfigurable Processor
MTASS/L	Machine Transferable AN/( ) Support Software/Large
MTASS/M	Machine Transferable AN/( ) Support Software/Medium
NAVMAT	Naval Material Command
NAVSEA	Naval Sea Systems Command
NTDS	Navy Tactical Data System
PDA	Principle Development Activity
R&D	Research and Development
SYSCOMS	Systems Commands
TADSO	Tactical Digital Systems Office
TADSTANDS	Tactical Digital Standards
TECPO	Tactical Embedded Computer Program Office
TECR	Tactical Embedded Computer Resources

MIL-STD-1589

JOVIAL (J-73) HIGH ORDER LANGUAGE

SESSION CHAIRMAN: Donna K. Gant
General Dynamics Corporation

MODERATOR: Melvin R. Barlow
VP & General Manager of General Dynamics
Data Systems Division

PREVIOUS PAGE
IS BLANK

JOVIAL STANDARDIZATION

Austin J. Maher

The Singer Company-Kearfott Division
150 Totowa Road
Wayne, New Jersey 07470
(201)785-6607

ABSTRACT

MIL-STD-1589B defines the JOVIAL (J73) high order programming language, the most recent dialect of a long line of AF languages in the JOVIAL family. Some highlights in the development of the JOVIAL language family and J73 in particular are reviewed as an introduction to this session.

At the present time, the J73 language is the only AF standard language for embedded computer systems. Some technical highlights of the language are presented as well as a review of its current use in AF systems. The key role played by the JOVIAL-Ada Users Group (JUG) in the evolution and use of J73 is reviewed as well as the procedures for handling proposed language changes and compiler validation.

With the apparent imminent availability of Ada*, some have questioned the wisdom of continued JOVIAL development and use. To help clarify this situation, the relationship between JOVIAL (J73) and Ada is discussed in the framework of the recently released AF plan for phased introduction of Ada as the replacement for its current standard language, J73.

INTRODUCTION

Dialects of the JOVIAL Language have been in use for computer software embedded in USAF weapon systems for over twenty years. At the present time, the JOVIAL (J73) language has been designated by the Air Force as the preferred implementation language for all operational weapon delivery software. As such, it is one of only seven high order languages accepted by the DoD as standard languages during this pre-Ada time period. This paper reviews some features

*The Ada Joint Project Office asserts that Ada is a registered trademark of the U.S. Department of Defense.

INTRODUCTION (Continued)

of the JOVIAL family of languages and highlights of their development. The role of JOVIAL as a precursor of Ada will also be covered. Subsequent papers will cover development of the J73 dialect, and its use in current systems and associated software support tools.

HISTORY

The origin of the JOVIAL programming language family dates back to the early 1960's. The name "JOVIAL" is a rather curious acronym representing: Jules Own Version of the International Algebraic Language, where "Jules" refers to Jules Schwartz, one of the language's principal original designers. The International Algebraic Language was Algol, of course, and the influence of this block-structured language has been felt in all subsequent JOVIAL language dialects.

Many dialects of the JOVIAL Language were developed in the 1960's. Probably the most widely used one being J3, which was adopted by the AF as a standard language for Command and Control applications (MIL-STD-1588).

In the early 1970's, two new dialects were developed. The Rome Air Development Center (RADC) conducted a study which culminated in the development of a much expanded dialect of JOVIAL designated as "J73". A simplified subdialect of this language (J73I) was subsequently implemented and used for the DAIS project at WPAFB. The full "J73" language was never implemented, falling victim to its own size and complexity.

At the same time, the B-1 avionics project was getting under way. Prior projects of this type had typically been implemented in Assembler language, due to the high efficiency requirements of the avionics environment. Since both the AF SPO and Boeing, the responsible contractor, were convinced that the use of High Order Language (HOL) for avionics was long overdue, and since their schedule did not permit waiting for the results of the "J73" committee's deliberation, a JOVIAL compiler was developed by Boeing and SofTech for the SINGER-Kearfott SKC2070 computer based upon an adaptation of J3. It was designated J3B. This pioneering use of HOL for avionics was eminently successful and the J3B language was subsequently used successfully to implement the F-16 Central Avionics Software on a Delco computer and the B52G Offensive Avionics Software on an IBM computer. JOVIAL (J3B) appeared to be on its way to becoming the de facto standard for large AF avionics projects.

At this point (approximately 1976), the DoD High Order Language Working Group (HOLWG) issued Directive [illegible] ... [illegible]it proliferation of different HOL's in weapon [illegible]

HISTORY (Continued)

The list included two JOVIAL dialects: J3 and J73. Since the full "J73" language had never been implemented, the AF decided to merge the best features of the J3B language (used for B-1, F-16, and B-52G) with the best features of J73I (used for DAIS) and to call the result J73*.

This was very effectively accomplished with the support of many industry representatives who united to form the JOVIAL Users Group (JUG) for this purpose. The resulting language has been widely implemented and has since become the only JOVIAL dialect permitted by DoD Directive 5000.31 (supplanting J3 in a recent update). The JUG organization proved to be such an effective vehicle for communication between the AF and inaustry on embedded computer software issues that it continues to be a viable entity, recently changing its name to the JOVIAL-Ada Users Group (JUG) as its scope expanded to address the introduction of Ada which will eventually supercede and obsolete JOVIAL (J73).

For further detail on the development of the first J73 compilers, see the paper by J. Pepe in this session.

JOVIAL LANGUAGE FEATURES

While JOVIAL language dialects differ from each other in some features, the JOVIAL family resemblance is provided by certain features which are typically common to all JOVIAL compilers. The principal JOVIAL family characteristics are outlined below:

- COMPOOL - From the earliest dialects, JOVIAL languages have contained a COMmon POOL of information to be shared among subroutines. This COMPOOL may contain shared data and shared subroutine definitions. This permits JOVIAL compilers to check type consistency between subroutine argument definitions and their invocation.

- Strong Typing - The type of a JOVIAL data name must be explicitly declared and its precision (minimum word length) can be stipulated. J73 permits a wealth of data types including: floating values, signed and unsigned integer values, fixed values, bit strings, character strings, status values, and pointers.

*In this paper, J73 refers to the modern JOVIAL dialect defined by MIL-STD-1589 A and B, while "J73" refers to the language defined by the RADC committee in 1973 (approximately).

JOVIAL LANGUAGE FEATURES (Continued)

- o Rich Data Structures - The memory layout of data can be explicitly defined in JOVIAL and complex tables of data can be defined to represent vectors, matrices, arrays, etc. J73 permits tables and blocks to be defined.

- o No Hardware Dependent Facilities - JOVIAL dialects typically have no provision for input/output or interrupt handling. It is presumed that these capabilities (which are strongly dependent upon the target computer hardware) are provided by procedures written in assembler language with an interface which permits their invocation as JOVIAL subroutines.

- o Status Type - A non-numeric representation is permitted for variables which distinguish between a finite number of states, an early predecessor of the enumeration data type in Ada.

- o Separate Compilation - Subroutines may be separately compiled. There are two types of subroutines, procedures which are invoked in a call-statement and functions which return a value and are invoked in a formula (expression).

- o Block Structured Flow Control Facilities.

The J73 dialect, faithful to its lineage, contains each of these generic JOVIAL characteristics. For a complete definition of the language see MIL-STD-1589B.

JOVIAL VS ADA

The energy devoted by the AF to JOVIAL in recent years has been perceived by some as a dilution of effort that might be better spent on Ada-related activities. Some even perceive J73 as a competitive language to Ada with some potential for undermining Ada's support. Is Ada therefore threatened by JOVIAL?

Occasional attendance at the quarterly JUG meetings should convince any observer that this is far from the case. That segment of industry which engages in the use of HOL's for embedded computer systems is clearly united in the drive to eliminate the proliferation of languages used with embedded computers. Moreover, many of those user organizations support all the DoD services, so a single DoD-wide language is much preferred over service-dependent "standards". The inclusion of a number of powerful new language features in Ada (which promise to facilitate the development of more [illegible] readable code) serves to further strengthen [illegible] their attraction to Ada.

JOVIAL VS ADA (Continued)

More importantly, though, Ada language processors are not yet developed to a level useful for embedded computer systems. The J73 language processors have been developed to fill this gap in timely fashion. Many projects are currently under way using the J73 language which could not have waited for Ada developments to be completed. A partial list of such projects includes: F-111 avionics update, F-16 avionics, DIS, LANTIRN, MRASM, MATE, MX, etc. So compilers for J73 have matured (there are now at least four viable suppliers) while the compilers for the more complex Ada language are still in their gestation period.

Just as it took several years for the JOVIAL (J73) compilers to mature, the Ada language processors can be expected to have some growing pains also. This is especially true for avionics applications, where the reliability and efficiency requirements are very stringent. Compilers for avionic systems must incorporate object code optimization algorithms which squeeze the utmost in efficiency out of the target computer's architecture. Such algorithms are understandably complex and troublesome. Often they are being refined for a year or more after the initial delivery of a full working compiler.

Since the Ada language is even more complex than J73, we can expect a similar (if not longer) period of refinement before production quality compilers are available. In a sense, the J73 development process serves as a preview of the Ada development process. AFSC is determined to apply all the lessons learned in the J73 developments to improve the Ada development process. Toward that end, a four-phased Ada Introduction Plan has been announced which introduces Ada deliberately, not hurriedly, into AF systems. The intent is to avoid the problems that a hasty introduction of the new language would create, no matter how well intended. The four phases have been identified as follows:

1) Laboratory Developments - Ada compilers and tool sets are developed and used by AF laboratories to gain initial experience and develop the necessary tools.

2) Parallel Development - AF Product Divisions choose project(s) whose software is developed both in a mature language (such as FORTRAN 77 or JOVIAL J73) and in Ada, without introducing any risk into the project since the Ada effort does not detract from the original project plan.

3) Selected Use - Programs would volunteer to use Ada to implement their operational software with Headquarters making the final selection.

JOVIAL VS ADA (Continued)

4) **Mandatory Use - AF regulations would be changed to discontinue the stated preference for J73 and require the use of Ada, unless a formal request for waiver is approved.**

Having learned from some of the more painful J73 experiences, explicit criteria are defined for making the transition from each phase to its successor. In short, Ada will not be prematurely mandated onto a project before the necessary maturity level has been accomplished. This prudent strategy is a valuable legacy from the J73 era which promises that the transition to Ada will be a smooth one.

CONCLUSION

The current AF standard language, JOVIAL (J73), is a worthy successor to the long line of JOVIAL dialects. Developed as it was by distilling the best features of its predecessors, it undoubtedly is the most powerful member of the family. While it will see widespread use in AF systems over the next few years, it will be only a matter of time before this JOVIAL dialect is superceded by Ada. thus transforming this best JOVIAL dialect into the last JOVIAL dialect.

Mr. Maher has been active in the field of avionics for over 20 years. Currently, he is the Manager of the Computer Software Engineering Department at the Kearfott Division of the SINGER Company. This Department is the focal point for computer software technology at Kearfott, including the development of realtime software for avionic products and associated automatic test equipment, support software for computer products and general purpose computation facilities for Engineering applications.

Recent activities have included significant contributions to the support of DoD standardization activities, including the AF standard architecture (MIL-STD-1750A), AF standard language (MIL-STD-1589B) and the Ada standard language development. Mr. Maher was selected to participate in the Monterey Workshop sponsored by the Computer Software Management subgroup of the Joint Logistics Commanders' Joint Policy Coordinating Group on Computer Resource Management. He recently completed a two year term as Chairman of the Jovial-Ada Users Group (JUG), a very active organization of industry and government participants. Since its inception in 1978, the JUG has stimulated a high level of communication and cooperation among its participants on computer software standardization initiatives.

Mr. Maher received a BS Physics degree from Holy Cross College and an MS Computer Science degree from Stevens Institute of Technology. His professional affiliations include the IEEE Computer Society and the Ada TEC, SIGPLAN and SIGARCH organization within the Association for Computing Machinery.

MIL-STD-1760

STANDARD STORES INTERFACE

SESSION CHAIRMAN: Claude Connell
AD/DLJA
Eglin AFB

MODERATOR: Major Lewellen Dougherty
HQ USAF/RDPV

Opening Remarks and Air Force

Overview of MIL-STD-1760

by

Major L. S. Dougherty
(Session Chairman)

Armament and Avionics Division
Operational Requirements Directorate
DCS/RD&A
HQ USAF, Washington DC

prepared for
MIL-STD-1760 Session

2nd AFSC Standardization Conference

Dayton Convention Center

1 December 1982

Good afternoon ladies and gentlemen. Before we get to the detailed papers today I would like to review how we got to where we are today and some of the assumptions that underlie the current version of MIL-STD-1760. I will talk for a moment about the kinds of questions that I am getting on the use of the standard from both program offices and industry, and I will tell you what questions I think we should be asking ourselves about the future of MIL-STD-1760.

What Problem Were We Solving

First, I would like to answer the question "What problem were we solving when we invented MIL-STD-1760?".

Back in the good old days of analog airplanes and analog weapons, there was a need to optimize the aircraft-to-store interface to minimize the cost of the weapon system. Signals were unique to a weapon both in terms of voltage levels and types of signals. Displays were unique to a combination of aircraft and weapon, and the control of a weapon was unique to a particular aircraft. It was occasionally possible to reuse a power line or a discrete, but the basic architecture of the avionics/weapon control system was so inflexible that adding wires to an aircraft for each new weapon became a way of life.

Because of the uniqueness of a particular aircraft/weapon combination it was almost impossible to do much effective planning for future weapons other than simply running some spare wires between accessible splice areas.

The evolution of smart weapons increased the amount of wiring required in an aircraft to near the limit of physical space available. The wire bundle behind the F-4 wing leading edge is a classic example.

Multiple adapters were required to make a particular weapon fit on multiple aircraft types. This was a maintenance and supply burden and a significant cost factor in some cases.

What Did We Decide To Do

It became clear to a few smart guys at Eglin and China Lake that somebody needed to solve the general problem of integrating new weapons with aircraft.

The first try looked at a single connector for all weapons. The result was a huge connector with something like 500 pins if it was to be 100% backward compatible -- clearly an unacceptable answer.

An alternate approach took into account the evolution of digital systems and multiplex technology, and limited the scope of the problem to new weapons and new aircraft. The solution that we settled on would put one new connector on existing aircraft for all new weapons, and would require that only that connector could be used on new weapon developments and new aircraft developments.

Under this approach, old interfaces would slowly disappear, and after 15 or 20 years would be about as useful as tonsils, at which time the old aircraft would be gone, hopefully replaced with new ones.

What Have We Done?

The strategy that we pursued called for three phases:

In phase one, we issued a draft MIL-STANDARD that only defined the signal set and basic wiring characteristics for the interface. We found out that there was a need for an auxiliary power connector for such things as ECM pods and external sensors such as LANTIRN.

We included extra wires for wideband signals over and above what was required for current systems. We built in provisions for possible future growth to DC prime power and fiber optics.

In order to meet the safety related requirement for two independent series switches to activate critical functions, we define the "critical store power" line.

. This line, combined with a multiplex bus message, would provide two independent means to enable critical functions.

Because the dual redundant multiplex bus is so flexible and reliable, MIL-STD-1760 implicitly requires all discretes to be sent over the multiplex bus -- the implementor can define the way the discretes are sent and the level of checking that is accomplished.

In phase two of our strategy for introducing MIL-STD-1710 we will add the connector specifications to the standard.

The insert arrangement has been agreed to and published as annex 25-20 to MIL-STD-1760 (insert arrangements for MIL-C-28999 and MIL-C-27599 electrical, circular connectors).

This insert arrangement has been reviewed by the nuclear community who found no fault with it.

Eight other slash sheets for pins, sockets, and tooling have been published including the triaxial pins for the 1553 mux bus and the coaxial pins for both video and high bandwidth lines.

The fiber optic pins will be developed by PAVE PILLAR.

Notice 1 to MIL-STD-1760 which specifies the intermatability characteristics of the common armament connection is in coordination with approval expected by 15 December 1982.

Phase three of the MIL-STD-1760 introduction strategy calls for the development and coordination of the logical (functional) element of the STD which will define protocols and common data formats for use of MIL-STD-1760.

International Adoption

MIL-STD-1760 has been proposed to NATO and to the Air Standardization Coordinating Committee nations as a basis for future air armament interoperability.

Programs to use MIL-STD-1760

The Air Force has committed to putting MIL-STD-1760 on all of our new weapons and retrofitting our aircraft as soon as possible.

Programs Using 1760

AMRAAM
LANTIRN adapter required
MRASM
30mm Gun Pod - in PMD
ASRAAM - NATO implications
WASP
CSW
SAW
F-15 - study complete
F-16
A-10 - study about to start
B-52
B-1B
Advanced Cruise Missile
Common Strategic Rotary Launcher - provisions for

Most Often Heard Concerns

In the course of normal business, I get some questions about how to use and interpret MIL-STD-1760. It is probably useful to go over some of the concerns for which I have answers. Then I would like to show you a list of questions that need to be answered.

Concern: There is not a requirement for all of the wires called out in MIL-STD-1760.

Answer: In the conventional sense you are right. However, a lot of the very best (and by that I mean far-sighted) people in the weapon and pod design business feel that there is a high probability that they will need all the pieces that make up the signal set. This is not to say that one system would use it all (though LANTIRN comes close) but that all of it will probably get used.

Concern: How do I make provisions for fiber optics?

Answer: Since we don't have a spec on the fiber optic cable, I would suggest putting in some small thin wall conduit at those places where it would be expensive to string fiber optic cable later.

Concern: We need more discretes!

Answer: The MIL-STD-1553B miltiplex bus that is part of the standard is just <u>full</u> of discretes. Every weapon I have looked at has internal digital logic anyway - the real choice is between adding something to the weapon to decade the bus, or adding wires to the connector, aircraft, pylon, rack, etc., increasing the size of the connector, and forcing the other users to deal with another pin that they don't use.

Concern: The high bandwidth cable is too big.

Answer: Generally this is a misinterpretation of the intent of the standard. There are coaxial cables available that will meet the bandwidth requirements and are less than one quarter of an inch in diameter. They will not transmit a kilowatt of power, but that was not why the 50 ohm and 75 ohm lines were included in the standard. They were intended for relatively low voltage and low power signals that were pre-conditioned to drive the line.

Concern: How come the video lines are 75 ohm coax when MAVERICK and GBU-15 use 91 ohm coax (triax)?

Answer: The mismatch between 75 ohm and 91 ohm cables is too small to worry about for the video bandwidths of MAVERICK and GBU-15. The NATO STANAG that covers video displays requires a 75 ohm impedance, and some of our current displays already have a 75 ohm input impedance.

Concern: Do we have to put in the auxiliary power connector at all store stations?

Answer: Not unless you expect to need it. Generally the heavy weight carrying stations or those that would carry ECM pods or sensor pods are good candidates.

Concern: Do we have to power all stations simultaneously at full current load?

Answer: No. Whatever limitations exist in terms of available current will simply have to be a constraint on simultaneous operation of multiple stores. I view it as similar to the situation in our office -- we have one 20 amp circuit that feeds about 10 outlets. We can operate three word processors on 3 outlets but don't plug in a 10 amp heater at the same time or your will have a clotch of angry secretaries, a popped circuit breaker and maybe even a messed up work processor disk.

Now I have a short list of questions for which I have no answers.

How will the avionics system get access to data on the stores management system multiplex bus?

How will video (RF) and power switching be controlled?

How many different video/RF channels will be needed in a particular system and how will they be divided up?

How will MIL-STD-1760 be adapted to a future video bus?

How can the high speed multiplex bus be incorporated into MIL-STD-1760?

What is the story on subsetting?

How can weapons that must fit old aircraft with existing interfaces be made to also fit aircraft that only have MIL-STD-1760 interfaces?

Can some aircraft (like the F-4 and F-111) use existing wiring to implement parts of MIL-STD-1760?

How is the Air Force going to control MIL-STD-1760 and certify aircraft and weapons?

If anyone in the audience has any ideas, I will be available after this session to discuss them with you.

NAVY PERSPECTIVE ON MIL-STD-1760

PRESENTED TO THE SECOND AFSC DOD EMBEDDED
COMPUTER STANDARDIZATION CONFERENCE
(30 NOV - 2 DEC 1982)

BY GERALD E. KOVALENKO
NAVAL AIR SYSTEMS COMMAND
AIR-323B, RM. 472, JP-1
AUTOVON 222-2522
WASHINGTON, D.C. 2036

Abstract

Prior to MIL-STD-1760, an aircraft and the stores which it carried were typically developed independently of each other or were developed exclusively for each other. This usually resulted in new aircraft/store electrical interface requirements and the general proliferation of overall store interface designs. The lack of standards within DOD for the aircraft/store electrical interface led to low levels of interoperability and costly aircraft modifications to achieve required store utilization flexibility. Application of this standard to new aircraft and stores will serve to significantly reduce and stablilize the number and variety of signals required at the aircraft/store interface, and increase store interoperability among the services, and within NATO. The Navy is supporting the development of the standard aircraft to store interface with the advanced aircraft armament system advanced development program.

INTRODUCTION

Interoperability is presently precluded by a set of obstructions. Within this set, a primary obstruction is nonstandard aircraft-to-store and store-to-aircraft interfaces. Interfaces between aircraft and stores are becoming increasingly sophisticated and complex. At the same time, there is an increasing desire on the part of the Department of Defense to increase service and allied nation interoperability between aircraft and stores. The Air Force and Navy have an on-going Joint Service program that address the problem of interoperability. Objectives of the Joint Service program are: 1) a validated Aircraft Armament Interoperable Interface Standard and Specification, 2) a demonstration of the interoperability of the armament-to-store interface by Navy and Air Force aircraft armament test beds, 3) examination of the feasibility of modifying representative existing aircraft and stores to enable compliance with the developed standards, and 4) a provision for a joint aircraft/store data base which may be efficiently and effectively used by the services.

BACKGROUND

Aircraft and munition systems are rightfully designed and built to accomplish limited and often very specialized objectives. In addition to the constraints imposed by these objectives, there are many other constraints such as overall physical envelope, weights, aerodynamic characteristics, etc. There has been a notable lack of constraint, however, on the configuration of the physical and functional interfaces between store and aircraft. Interface configurations have tended to be optimized around weapon systems, with little attention given to applications outside a specified, usually narrow list of aircraft and stores. This philosophy unwittingly leads to a lack of aircraft/store interoperability with those not on the original armament list.

As a new store is developed, it must be compatible with a specified set of new and old aircraft. And as a new aircraft is developed, it must be compatible with a specified set of new and old stores. The net effect has been a proliferation of interface designs and costly modification to achieve any degree of interoperability. This intertwined and increasing spiral of unique aircraft and store systems is at the root of the store interoperability problem, and contributes heavily to the high cost of ownership. Other factors affecting and contributing are:

a. Rapid advances in technology
b. Trend toward sophisticated stores
c. Acquisition management processes with cost and schedule constraints being of primary importance
d. Requirements for new stores to be compatible with old aircraft systems configurations and vice versa

Until recently, there has been little emphasis or requirement for general store/aircraft interoperability. The recognition that interoperability between countries' weapon systems would significantly enhance the NATO defense posture has led to decisions to correct the growing interoperability problem. Recently, the requirement for interoperability has been included as part of system development direction. This kind of direction and emphasis will require designers to give more attention to the problem while making trade-off decisions, but it will not of itself produce total interoperability. [illegible] store and/or aircraft system modifications is one approach, but as [illegible] rule these are inordinately expensive. The military service headquarters has concluded that the long term solution to this dilemma requires that concentrated armament system research and development be conducted leading to a set of standards and specifications which will support a physical and functional interoperable interface.

STATEMENT OF PROBLEM

Interoperability is presently prevented by a number of obstructions. Within this set, a primary obstruction is the nonstandard electrical aircraft-to-store interface. The electrical interface between aircraft and stores is becoming increasingly sophisticated and complex. At the same time, there is an increasing desire on the part of the Department of Defense to increase service and allied nation interoperability between aircraft and stores.

The number of different types of stores is large (more than 100) and continues to grow as a result of development and acquisition programs. Stores include conventional general purpose bombs, guided bombs, missiles (air-to-air and air-to-ground), nuclear weapons, sensor pods, dropped sensors, camera pods, countermeasure pods, fuel tanks, sub-munition dispensers, guns, rockets, etc. The electrical interface between aircraft and stores is only partially guided by standards and, therefore, has tended to evolve into system peculiar adapters/connectors, electronic signals, power connections, and other armament assemblies which make interoperability impossible without major modifications to aircraft/and or stores on a case-by-case basis. The trend toward more complex store function which require increasing amounts of avionics data from aircrft systems is causing the problem to become increasingly acute. Examples of this situation are AMRAAM, HARPOON, PHOENIX, HELLFIRE, ATLAS POD, ALCM, etc.

On the aircraft side of the interface, stores management systems are unique to each aircraft type and sometimes each model. Old aircraft Stores Management Systems (SMS) are generally hardwired, not integrated, not automated, and reflect outmoded state-of-the-art in electronics and electronic design. Although new aircraft SMS designs reflect current technologies in electronics and communications, they are still tailored to a specific store list and are not designed for growth. Invariably, the store list changes requiring modifications almost as soon as the aircraft begins its operational life. The adoption of acquisition methods which result in aircraft systems with SMS's which are tailored to handle a specified list of stores has limited weapon system capability growth and flexibility. The methods yield weapon systems which are well defined within themselves, but are inflexible and costly to modify.

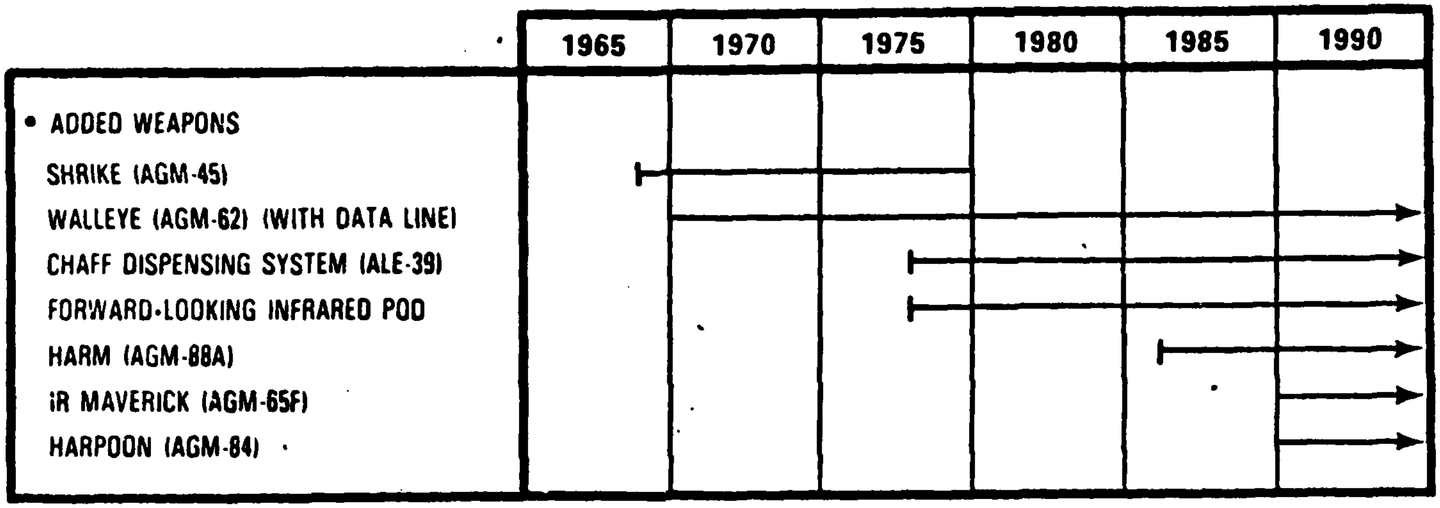
Two examples of aircraft weapon update costs are the A-7 and F-18. The A-7 being an older aircraft one would asume that this aircraft would naturally require costly modifications with the new modern F-18 requiring only minor modifications. The F-18 is more compatible with the new weapons because of the digital data bus, etc. However, significant modifications are required for the F-18 also. Fig. 1 shows armament changes and their associated cost for the A-7. Fig. 2 and 3 shown armament changes and their associated costs for the F-18.

CONCEPT OF SOLUTION

Standardization of interfaces between aircraft and stores is one major determinant of the amount of interoperability which may be achieved. The ultimate solution to the interoperability dilemma will depend to a large extent upon the removal of the nonstandard interface obstruction and the establishment of a standard which will govern all future aircraft/store interfaces and serve as a guide for the appropriate conversion of selected existing equipments.

Prior to ML-STD-1760, an aircrft and the stores which it carried were typically developed independently of each other or were developed exclusively for each other. Application of this standard to new aircraft and stores will serve to significantly reduce and stabilize the number and variety of signal required at the aircraft/store interface, and increase store interoperability among the services, and within NATO. The Navy is supporting the development of the standard aircraft to store electrical interface with the advanced aircraft armament system advanced development program.

MIL-STD-1760 addresses the electrical interconnection system and specifies the parameters required to obtain electrical compatibility between aircraft and stores.The complete aircraft/store electrical interconnection system is comprised of three hierarchical elements: electrical, logical, and physical. The electrical element specifies the aircraft-to-store interface signal set. The logical element defines the communications architecture, message content and formatting, and data transfer protocol. The physical element specifies the aircraft-to-store wiring system including connectors, umbilicals and their terminations. Requirements for mechanical, aerodynamic logistic, and operational compatibility are specified in other existing or in development DOD documentation. Another MIL-STD being deveoped under the auspices of the Hoint Technical Coordinating Group, Working Party 6 - Aircraft/Stores Compatibility will address the relationships of these documents and their use in the overall system level compatibility process. That MIL-STD will be titled Aircraft/Store Interconnection System Standard. Figure 4 shows the hierarchical relationship of the proposed MIL-STD with MIL-STD-1760 and other documentation. Figure 5 shows the MIL-STD Interface Definition for various aircraft carriage modes.



WEAPON/COST	R&D	COST PER A/C	NO. OF A/C	TOTAL
SHRIKE (AGM-45) (WITH DATA LINE)	2.0M	.075M	411.0M	30.83M
WALLEYE (AGM-62)	1.5	.004	411.0	1.64
DISPENSER (ALE-37)	.5	.015	380.0	5.70
FORWARD LOOKING INFRARED (FLIR) POD	4.0	.667	229.0	152.74
HARM (AGM-88A)	2.0	.705	124.0	87.42
IR MAVERICK (AGM-65F)	3.0	.296	176.0	52.10
HARPOON (AGM-84)				
	11.0M			330.43 11.00
TOTAL				341.43M

FIGURE 1 A-7 ARMAMENT UPDATES AND ASSOCIATED COSTS

A/C – COST	R & D	COST PER A/C	NO OF A/C	TOTAL
II. F-18				
A. PRODUCTION LINE UPDATE	1.0M NON-RECUR FOR SMS ONLY	35K	BASED ON 1029 A/C	36M
B. RETROFIT OF INSERVICE A/C		115K	BASED ON 337 A/C	38.76M
C. SUB-TOTAL			(NO OF A/C IS BASED ON) THE AVALIABILITY SCHEDULE OF THE AMRAAM (337 A/C WILL BE INSERVICE)	74.76 1.0 ——— 75.76M

F-18/AMRAAM ARMAMENT UPDATE COST ESTIMATE

FSD REQUIREMENTS	NEW REQUIREMENTS
AIM-7F	AIM-7M
AIM-9L	AIM-9M
MK-20	AGM-65F
MK-82 LD/HD	AGM-84
MK-82 CONICAL	BLU-80
MK-82 LGB	MK-36
MK-83 LD	MK-40
MK-83 LGB	MK-41
MK-84 LD	MK-52
MK-84 LGB	MK-55
CBU-59	MK-56
AGM-65E	MK-60
AGM-88A	MK-65
WE I ER/DL	
B57/BDU-11, 12, 20	WE II ER/DL
B61/BDU-36	
MK-106/MER	BDU-48
MK-76/MER	FMU-139/B
LAU-10/61/68	IMPROVED AGM-88A

COST ESTIMATE

1. $12M R&D NON-RECUR. COST
2. AIRCRAFT MOD
   a) PRODUCTION LINE-UPDATE
      $35K/AC X1029AC = $36M
   b) RETROFIT OF INSERVICE AIRCRAFT
      $115K/AC X 337AC = $38.76M
   c) TOTAL AC MOD = $74.76M
3. TOTAL
   a) R&D - $12M
   b) AC MOD -$74.76M
   c) TOTAL - $86.76M

FIGURE 3 F-18/ARMAMENT UPDATE AND ASSOCIATED COST ESTIMATE

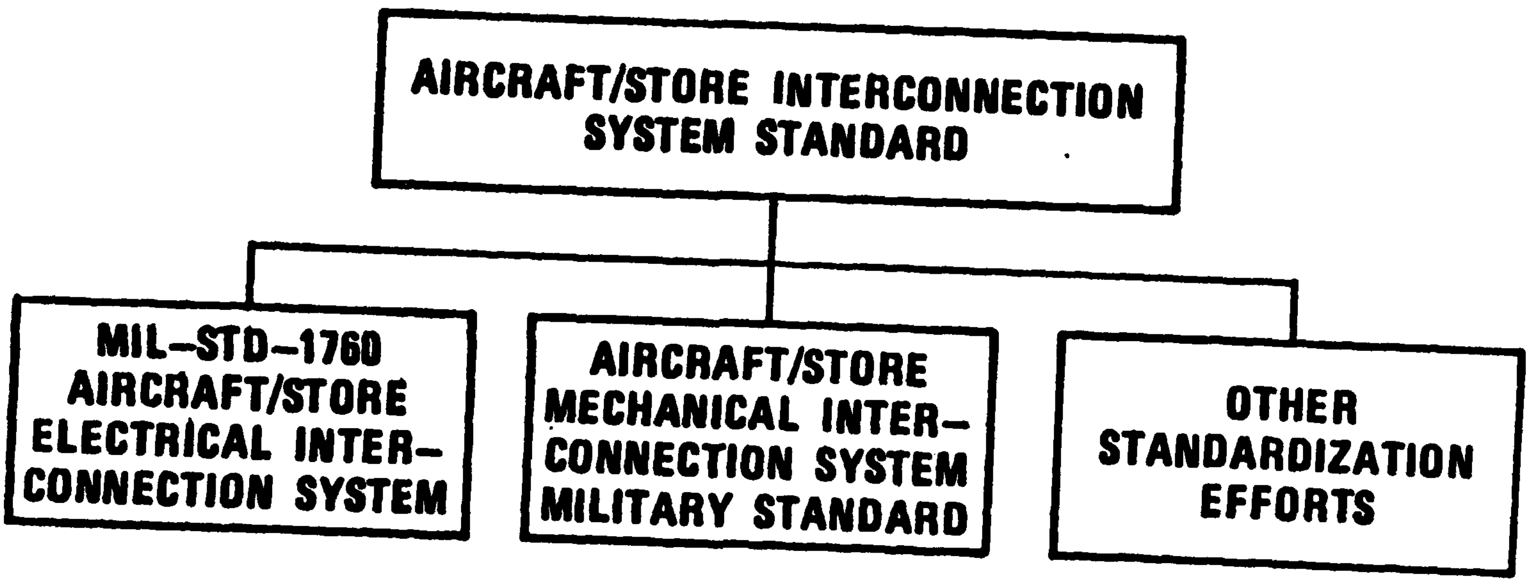


FIGURE 4 AIRCRAFT/SOTRE INTERCONEECT SYSTEM STANDARDS PLAN

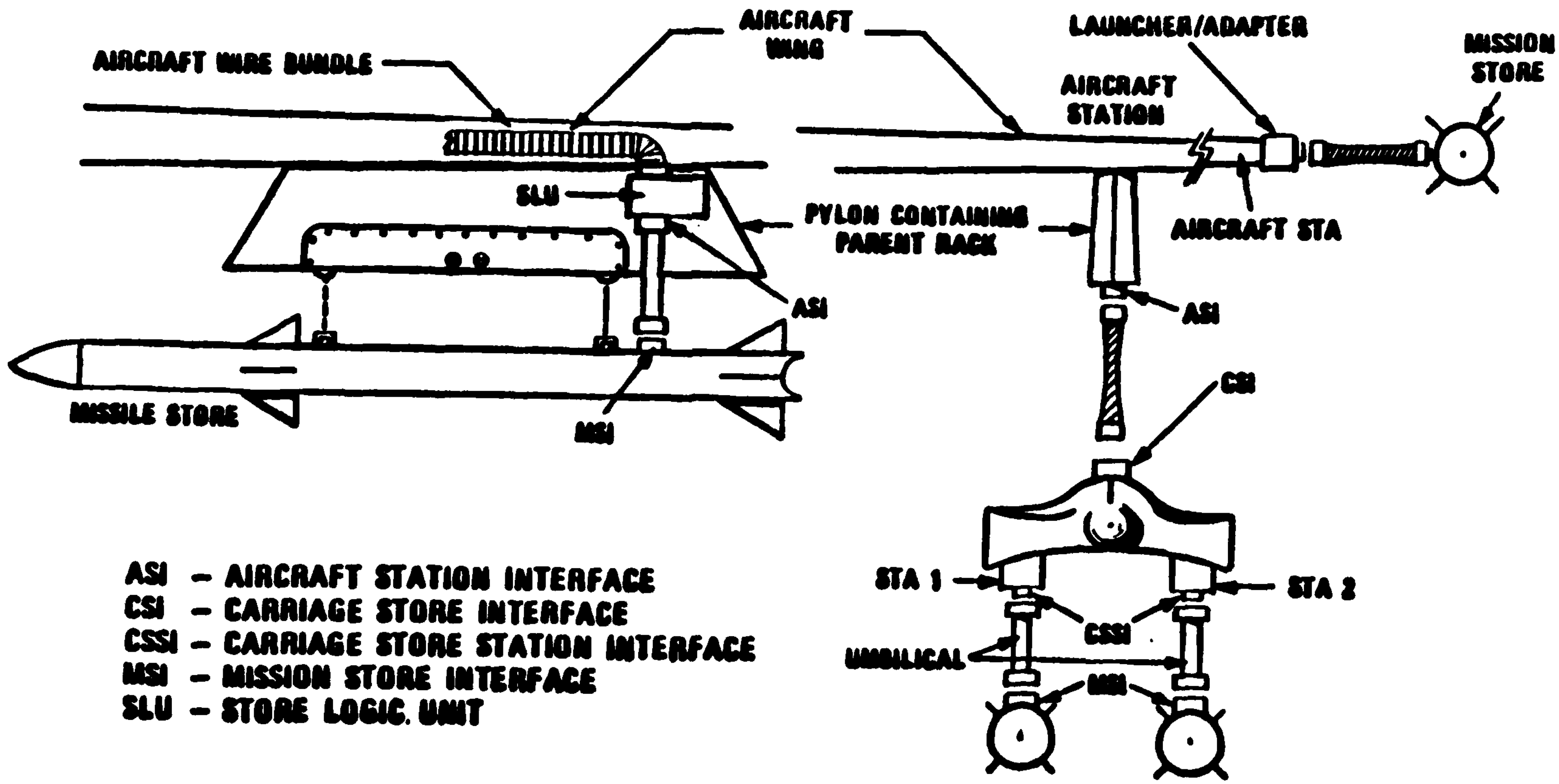


FIGURE 5 MIL-STD-1760 INTERFACE DEFINITIONS

BIOGRAPHY

Mr. Gerald E. Kovalenko graduated from North Dakota State University in May 1960 with a BS degree in Mechanical Engineering. Mr. Kovalenko has continued his education through graduate school extention courses in Engineering and Management from University of California at Los Angeles and at Santa Barbara, University of Southern California, and University of Michigan with specially short course training through the Governments continued training program. In May of 1960, Mr. Kovalenko accepted a position with the Corps of Engineers in Omaha, Nebraska where he worked in the Design Groups and the Atlas "E" Missile Installation and OAHE Dam Construction Sites. In January 1962 Mr. Kovalenko accepted a position with the Naval Weapons Center, China Lake, California. During his 17 years at the Naval Weapons Center, Mr. Kovalenko worked on a variety of Air and Surface Launched Armament Subsystem Development Projects. The Exploratory and Advanced Development Projects included: Surface Weapons Fire Control Exploratory Development Program, Point Defense, Sidewinder, AGILE, SHRIKE, HARM, HARPOON, Aircraft Survivability, Aircrft Armament Equipment, Liquid Propellant Gun Technology, 20mm General purpose projectile, and Gunfire Control System Technology. During the 17 years at the Naval Weapons Center, Mr. Kovalenko was head of the Mechanical Design and Analysis Branch for 7 years. In November 1978 Mr. Kovalenko accepted a position with the Naval Air Systems Command, Washington, D.C.. As an Armament Engineer, the Technology responsibilities included Gun System Technology, Pyrotechnics Tehnology, Air Armament System Technology and Cartridge and Cartridge Actuated Device Technology. The responsibility as Program Manager is primarily for the "Advanced Aircraft Armament System" (AAAS) Advanced Development Program which includes Stores Management and suspension and release equipment. The "AAAS" program is a Joint Service (Air Force/Navy) development program which is developing the aircraft store electrical interconnect system. As a technology administrator and program manager interacting with international groups for technology coordination is required. The two international groups are NATO and Air Standardization Coordination Committee (ASCC) Air Armament Working Groups and Inter-Service JTCG/MD working party for aircraft/stores compatibility and working party for guns and ammunition.

MIL-STD-1815

ADA HIGH ORDER LANGUAGE

SESSION CHAIRMAN: Paul Wood
Sperry Univac Corp.

MODERATOR: Clyde E. Allen
Vice President, Product Engineering

USE OF ADA* IN SYSTEM DESIGN: A CASE STUDY

Hal C. Ferguson and Michael B. Patrick

General Dynamics Data Systems Division, Central Center
P. O. Box 748, M.Z. 5400
Ft. Worth, Texas 76101
(817) 731-0741

ABSTRACT

Since Ada has been adopted by the Department of Defense as the standard programming language for embedded computer systems, it is vitally important that government and industry personnel understand the consequence of using Ada in system development. A case study was recently completed in which Ada was used throughout the development of a large digital message switch. Prior to the start of system design, personnel were trained in the use of Ada and a methodology incorporating Ada compatible requirements and design techniques was developed. With judicious application of the methodology, a system design was produced. One major component of the system was programmed in Ada. In this paper, the case study effort is described. Examples of system design structures and Ada code are presented. Lessons learned and conclusions regarding the use of Ada are discussed.

INTRODUCTION

The Ada Capability Study was performed by the Data Systems Division (DSD) of General Dynamics Corporation, Fort Worth, Texas, under contract with the Department of the Army Communications - Electronics Command (CECOM), Fort Monmouth, New Jersey. The purpose of the contract was to provide a documented case study and analysis of the use of Ada in the design, development, and implementation of a large scale digital system. An existing real time system, the AN/TYC-39 message switch, was selected by the Army for this case study and the actual "A level" specifications were provided to General Dynamics.

This task involved the performance of four subtasks: (1) develop a methodology for the use of Ada in the specification of requirements, the design, and the implementation of a digital system; (2) train personnel in the use of the Ada language and the methodology; (3) use the developed methodology to design a system for the AN/TYC-39 message switch; and (4) program one selected module of the designed system.

*Ada is a registered trademark of the U.S. DOD (AJPO)

ORGANIZATION AND PERSONNEL

A project organization was established with the group divided into three teams: methodology development, system design, and training. Consultants from industry and academia were used to support methodology development, to provide expert consultation, and to review the entire case study effort.

The methodology team was composed of three General Dynamics employees and two consultants from North Texas State University. A total of seven employees were assigned to the design team during the contract effort, with a maximum staff of six at any one time. Personnel with varied backgrounds were chosen so that the case study effort would be accomplished in a real world environment. Two of the people had Master's degrees in computer science; five had Bachelor's degrees, three in mathematics and two in electrical engineering. The leader of the design team was chosen because he had specific experience in communications and telephone switching systems. Other team members had varied backgrounds which included real time systems programming, compiler development, and business data processing. Four of the people had experience in assembly language and Fortran; three had used structured higher order languages including Pascal.

TRAINING

The training of personnel was an integral part of the Ada Capability Study. Early in the program, the project management sought effective Ada training in the form of books, video tapes, short courses, and tutorials. It soon became evident that the availability of training materials at the level required for the Ada Capability Study was limited, and that it was necessary to develop a training curriculum to meet the training requirements of the project.

In-house training of Ada project personnel was conducted in two phases. The first phase consisted of ten 2- to 3-hour presentations given to the original members of the Ada project team by a consultant from North Texas State University (NTSU). These presentations were given in seminar fashion, in the form of lectures with accompanying viewgraphs and handouts, and with free class discussion. All features of the Ada language were covered in this phase. Because of time constraints, most topics were covered rather quickly.

The second phase consisted of a reprise of the first phase given primarily for new team members, but open to anyone on the Ada project. These sessions covered fundamentals of the language in more detail (in seven 2-hour meetings). Attendees in this phase had, on average, less broad experience in higher order languages than those in the first phase. Special emphasis was given to overall program structure, data types, packaging, and tasking. Phase I experience was useful in identifying and anticipating student difficulties.

METHODOLOGY DEVELOPMENT

The methodology team conducted a study of existing methodologies and published an Ada Integrated Methodology (AIM). It is comprised of two main parts described in more detail below: (1) the Ada Requirements Methodology (ARM) and (2) the Ada Design Methodology (ADM). It integrates several existing methodologies and some important design concepts with the power of the Ada language.

The purpose of the requirements phase of the software life cycle is to promote an understanding of the problem at hand by clearly (unambiguously) and succinctly specifying the functional and non-functional goals and objectives of a project. ARM accomplishes this purpose in a four-part process as follows:

PART I - Develop Functional Requirements in four steps:

1. Create a data flow diagram (DFD) model of the system to be developed;
2. Develop and maintain a data dictionary of all data flows on the DFD model;
3. Develop a logical data structure model;
4. Write functional requirements using an Ada-based structured English.

PART II - State the Non-Functional Requirements (reliability, performance, accuracy, etc.) in English narrative format.

PART III - Develop Concurrency Requirements using concurrency charts and/or English narrative.

PART IV - Formulate and organize the Ada Requirements Document which is primarily an accumulation of the outputs from PARTS I, II, and III.

Collectively, the components of the ARM output are used to state and graphically illustrate system requirements (both functional and non-functional). It is not intended to constrain or limit the designer, although the functional decomposition and creation of DFDs in the requirements phase is considered by some as moving toward design. However, the intent of ARM is to produce a requirements document that will aid and facilitate the design process. The inclusion of functional decomposition and DFDs in ARM is consistent with a trend in contemporary system development to incorporate more planning, discipline, and decision-making up front (prior to program development). From experience gained in this project, the researchers feel that ARM could replace the old military A-specification document, which proved unsuitable in adequately documenting the message switch modified by this project.

ADM is a design methodology that converts the output of ARM to a system design expressed by graphic models and Ada PDL. Several methodologies (including Structured Design, the Jackson data structure approach, and object-oriented design) and design concepts were combined with the Ada language to form ADM. Therefore, the designer that applies ADM should be equipped with a design tool bag. This tool bag approach was favored by the design team as it attempted to achieve the two-fold purpose of ADM - (1) produce a design that satisfies the requirements in the Ada Requirements Document and (2) produce a design that exhibits maintainability (flexibility for design changes during development and after implementation). ADM achieves its purpose in three parts as described below:

PART I - Architectural Design

a. Identify data structures
b. Perform object-oriented design pre-analysis
c. Develop concurrency requirements
d. Generate structure chart
e. Validate "goodness" of design
f. Validate correctness of design
g. Document system interfaces graphically
h. Perform preliminary design review
i. Develop Ada Unit Specifications (beginnings of Ada PDL)
j. Perform hardware/software partitioning

PART II - Detail Design

a. Express system design in Ada PDL
b. Perform a final design review
   i. Design Walk-Thru
   ii. Preprogramming Ada Evaluation
   iii. Requirements-to-Design Traceability
   iv. Design Philosophy Review

PART III - Compilation of the Ada Design Document

In summary, the application of ADM facilitates the development of programming requirements and design documentation. This is normally within the scope of a design methodology. However, because ADM is an Ada-based methodology, there is a tendency toward using the Ada language as much as possible during design. Also, the nature of the Ada language requires the designer to function somewhat like a chief programmer during design. For example, the designer must evaluate the overall design in terms of data types, overloading of unctions, etc. Any encumbrances created by improper data typing or improper overloading of functions should be eliminated before actual program development begins. ADM's use of Ada as a PDL for developing programming specifications is another example of the impact and use of the Ada language during design.

There are many elements of existing methodologies which are beyond the scope of this methodology development effort. Specifically, this methodology excludes the following:

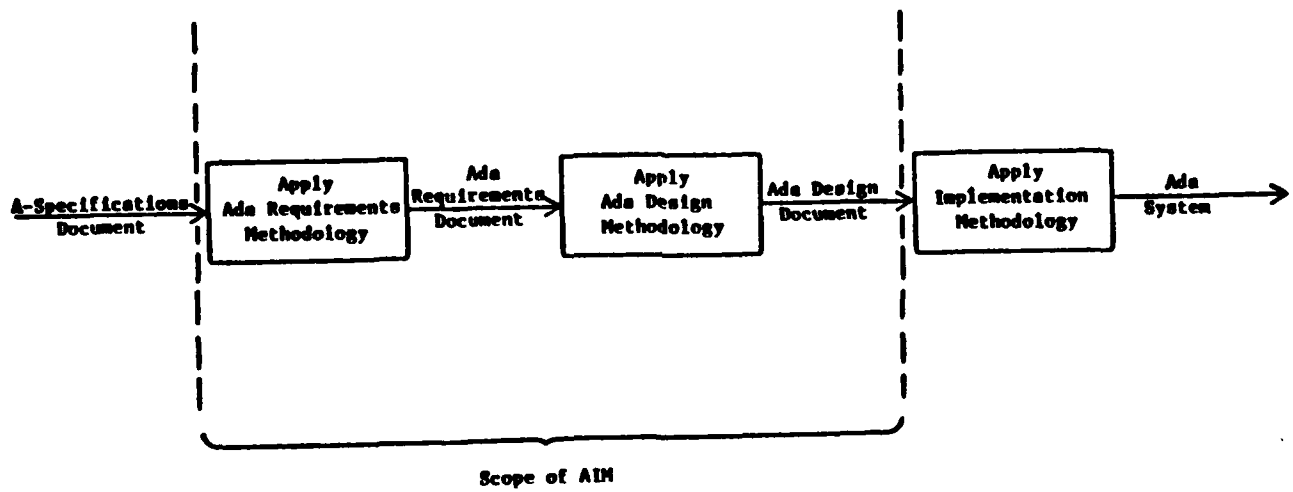


Figure 1 AIM Project Life Cycle Concept.

- Development of documentation and plans in support of the implementation of a new system such as (1) user guide, (2) operations guide, (3) test plan, (4) new procedures, and (5) training aids and courses.

- Survey of the present system

- Conversion from present system

- Implementation phase (does not address programming and testing - goes through design only)

- Constraints such as time, money, and personnel

- Detailed explanations of contributing methodologies, concepts, and the Ada language (provides only basic guidelines and examples for the requirements analyst and designer).

The AIM project life cycle concept can be used to help illustrate the scope of AIM (See Figure 1). The dotted lines indicate where AIM begins and ends. Although there is no implementation methodology within AIM, one chapter does provide some Ada development standards.

REQUIREMENTS DEFINITION PHASE

The message switch requirements definition began simultaneously with the methodology development at the start of the contract in July 1981. The requirements analysis team then consisted of three persons, a chief systems engineer, chief programmer, and an assistant. The major problems confronting the team were (1) to understand the message switch application (2) to apply the appropriate methods during requirements that would facilitate Ada oriented design and coding phases and (3) to determine the appropriate limit to the scope of the project so that completion could be accomplished in one year. An additional uncertainty, the lack of understanding of the Ada language, was to be resolved by training, as previously discussed.

The solution to limit the scope of the project was to set up a meeting with the Army representatives to discuss the matter. It was determined that certain complicated I/O devices would be eliminated, and minimal effort would be spent on the operator interface and duplication of similar message format processing. Ordinarily, there would be frequent meetings with the customer to resolve misunderstandings and incongruities in the specifications; however, the message switch application is a standard part of the army equipment inventory and the requirements are essentially static. For the purposes of this contract, any conflicts between wording of specifications was evaluated in the requirements group and the most practical approach was taken. This undoubtedly expedited the requirements phase.

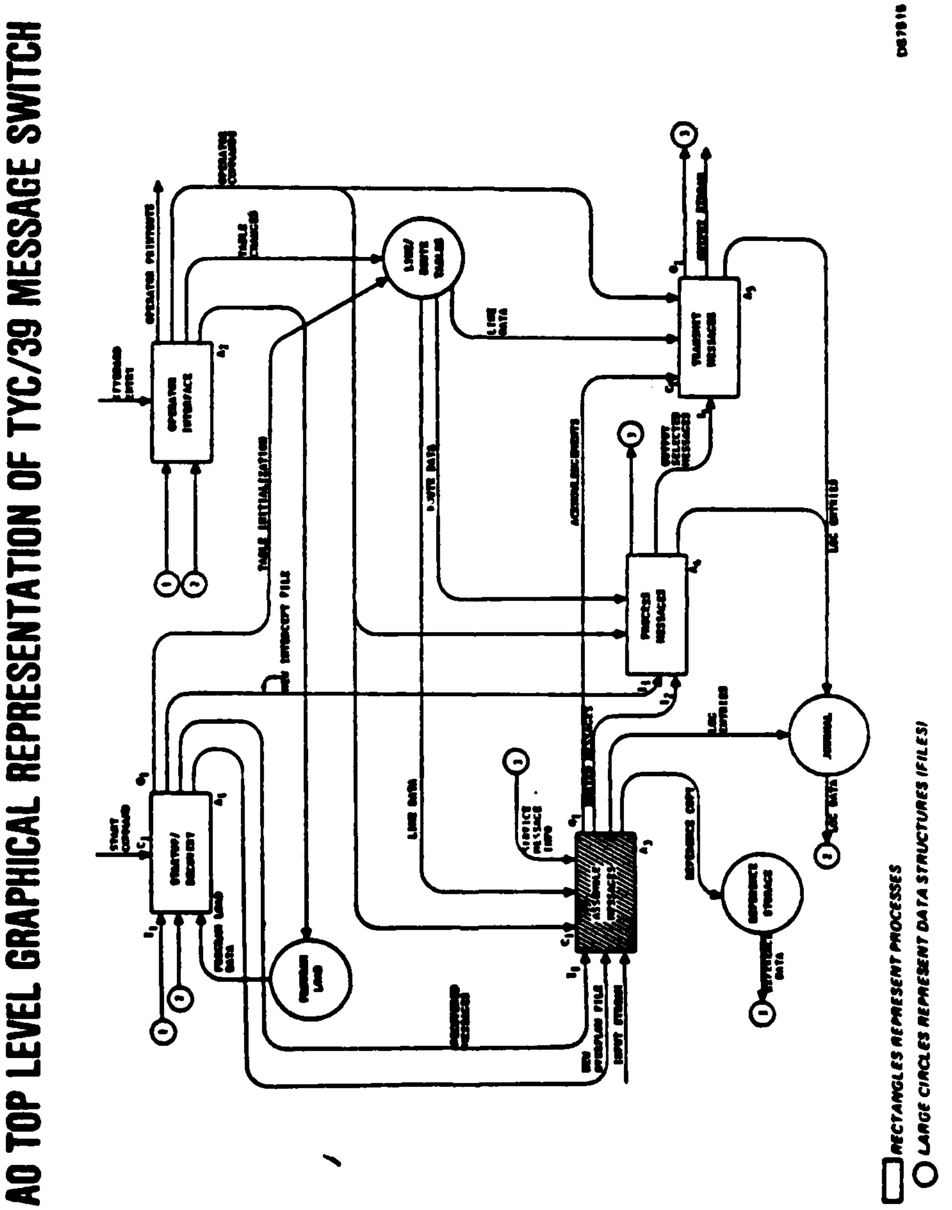


Figure 2. Example of a Data Flow Diagram (DFD)

The understanding of the message switch application was gained by studying the "A level" specifications provided by the customer. Within these, many other specs were referenced. All specifications were in narrative form, and the "A level" spec described an additional system not related to this contract.

Because of the organization of the specifications, understanding of the message switch, a large complex system, came in fragmented parts, and the whole learning process was iterative. Therefore, a graphical approach utilizing data flow diagrams (DFD) was used to represent the team members' understanding as it evolved. A new data flow diagram (DFD) could be quickly sketched out with any radical change in thinking. Minor refinements could be easily added to the existing diagrams. Since the chief engineer had prior experience with structured analysis (SA) techniques, the structured analysis and design technique (SADT*) was used until the methodology group formulated a requirements methodology. SADT has the benefit of showing the separation of data and control elements, considered by the chief engineer to be important for real time system design. An example of a modified top level SADT diagram for the message switch is shown in Figure 2.

This diagram is the final requirements version and supplemental information provided by the methodology team has been added. The rectangles represent traditional SADT processes, whereas the large circles, taken from Yourdon/De Marco SA, represent data repositories (files). The small circles are on-page connectors to reduce line crossings. For example, all circles labeled with 1 connect to each other. In reading the DFDs, data is input at the left side of a rectangle, control is input at the top, outputs exit the right side, and flow is in the direction of the arrows.

The label on each line connecting the rectangles and circles is described in a data dictionary which provides further decomposition of the information. For example, on the Node A0 diagram (Figure 2), the "sorted message" data flows from box A3 to A4. Because multiple inputs and outputs are numbered in ascending order from top to bottom, the "sorted messages output" from A3 is O2. In comparing the "sorted messages" entry in the summary data dictionary of Figure 3, it can be observed that a connection from output 2 of node A3 exists, as well as connections from input 2 of node A4, input 1 of A41, and input 1 of A411. Further, it can be seen that sorted messages are made up of a message plus a message control block (MCB). These terms can be further decomposed as seen in the "message" entry.

For the message switch application there are thirty-two additional graphic diagrams that represent successive refinement of the top level functions illustrated here. Eventually, a point is reached where a function becomes so trivial that it is no longer feasible to represent graphically. At this stage, an Ada subset is used as a requirements specification language (RSL) to express the specifications for each lowest level (primitive) function of the DFD. The subset of the Ada constructs utilized by the Ada Requirements Methodology (ARM) is provided below:

*SADT is a registered trademark of SofTech, Inc.

DATA DICTIONARY - Example

FIRST_LMF_CHARACTER = [B|D|I|S|C|A|T|R|Q|F]

JANAP_FMT_LN_2 = PRECEDENCE + LMF_PAIR +
CLASS + CIC.CAI + 'b' + OSRI
+ OSSN + 'b' + DATE_TIME +
'–' + REDUNDANT_CLASS +
'––' + RI + <P .49>(RI) +
'.' + LN_END

JANAP_HEADER = (JANAP_FMT_LN_1) +
JANAP_FMT_LN_2 +
(JANAP_FMT_LN_3) +
(FMT_LN_4) + (FMT_LN_5) +
(FMT_LN_6) + (FMT_LN_7) +
(FMT_LN_8) + (FMT_LN_9) +
(FMT_LN_10)

JANAP_MESSAGE = [CARD_MESSAGE|TTY_MESSAGE]

MESSAGE = [JANAP MESSAGE|
ACP_MESSAGE]

NEW_ROUTE_LINE = ROUTE_LINE

RI = [R|Y|U] + <3..8>(LETTER)

ROUTE_LINE = [JANAP_FMT_LN 2|
ACP_FMS_LN_2
CARD_FMT_LN_2]

02A3
12A4
11A41
11A411
SORTED_MESSAGES = MESSAGE + MCB

TTY_MESSAGE = JANAP_HEADER + SEPARATOR +
BODY + SEPARATOR +
JANAP TRAILER

Figure 3. Condensed Data Dictionary for the Message Switch

if statement
case statement
loop statement
assignment statement
code statement
expression
relation
exit statement
procedure calls for pre-defined procedures
raise statement
exception handler

Displined English is used to supplement the Ada requirements language constructs when knowledge is insufficient for representing a particular requirement in detail. Additionally, comments are used where appropriate to add clarity to the requirements. The symbol for a comment line is "—", the same as the Ada comment notation. A "->" symbol is used to indicate a line of requirements specifications that is stated in disciplined English.

In addition to the Ada RSL, a primitive contains a brief narrative which explains its purpose, shows traceability back to the "A level" specification provided by the customer, and contains the date and initials of the analyst performing the work. Figure 4 is an example of an Ada requirements specification which uses Ada constructs, disciplined English, and comments. During the requirements phase approximately 150 primitives were produced.

By late October, one analyst had been transferred to the methodology group and another had been added, leaving a total of four, and the first draft of the Ada Requirements Methodology was released from the methodology group. Fortunately, the data flow diagrams were still based on SADT, but additional enhancements were supplied by structured analysis techniques. Some redrawing of the diagrams was required as a result of the methodology arrival and additional levels of decomposition occurred. Another requirements analyst was then assigned to the project. The chief engineer assigned the analyst to the output message section. The chief programmer was continuing with message routing and the previously-assigned analyst continued on the input message section. Initial assignments were made by the chief engineer based on individual interest, which was somewhat related to the team members' backgrounds. Hardware designers worked in areas of I/O and software designers in transform analysis. The chief engineer was coordinating the requirements development, assisting in message input, and spending part of his time completing another project unrelated to the message switch (real world problem).

The decomposition process continued into December. Initially, it was expected that the design document would be completed by Christmas, but this was not accomplished. The message input requirements were considerably behind, and the requirements analyst assigned to the message input function asked to be removed from the project. The chief engineer and chief programmer completed the section by late January while the other analyst finished the message output. Also, in January, non-functional requirements

```
--A413 PERFORM ROUTING LINE SEGREGATION
--PURPOSE : GENERATES MULTIPLE ROUTES, SEGREGATED BY OUTPUT
    -- TRUNK TYPE
--REQUIRED BY PARAGRAPH: 3.2.1.2.7.6, FURTHER REFERENCES IN
-- DCAC-370-D175-1, SECTION 9-4.
-- REV BAAA
-- 11/5/81 HF/PD

procedure SEGREGATE is
begin
  for NEXT_RI in ROUTE_LINE loop
    --OBTAIN NEXT RI IN HEADER
    if NEXT_RI = RI_TERMINATOR (2EH) then
      --> COPY RI_TERMINATOR TO NEW_ROUTE_LINE;
      exit loop;
    end if;
    if NEXT_RI = RI in GROUP_RI then
      --> COPY NEXT_RI TO NEW_ROUTE_LINE;
    else
      --> REPEAT UNTIL ALL RI IN GROUP_RI HAVE BEEN COMPARED TO
          --> NEXT_RI;
    end if;
    if NO RI=NEXT_RI then
      if OUTPUT DEVICE is DTE then
        if LMF_PAIR = "SC","CC","BB","DD" or "II"then
          --> COPY AN EQUIVALENT NUMBER OF SPACES TO
              --> NEW_ROUTE_LINE;
        end if;
      elsif LMF_FIRST = "S","C","B","D" or "I"then
        --> COPY A SINGLE SPACE TO NEW_ROUTE_LINE;
      elsif LMF_FIRST = "T","R","F","Q" or "A"then
        --> COPY "SI"(OFH) TO NEW_ROUTE_LINE;
      else
        raise LMF_ERROR;
      end if;
    end if;
    --> COPY ONE SPACE CHARACTER TO NEW_ROUTE_LINE;
  end loop;
end SEGREGATE;
```

Figure 4. Example of an Ada Requirements Primitive

relating to performance, reliability, and operating constraints not describable elsewhere, were completed. Concurrency (high level) was studied, and a concurrency chart was produced.

The efforts of the design team during the requirements phase resulted in a 174-page Ada requirements document. This document restated the "A level" specifications in a more structured, organized format with many graphic illustrations (DFDs and concurrency), Ada subset RSL, and narrative.

The application of ARM produced a very good understanding of the problem during the requirements phase. The success in this area can be mainly attributed to the functional decomposition, DFD approach used. Although it was not known at that time, the design and coding phase went considerably faster than expected. It is felt that this is due partly to the thorough understanding gained during this phase.

DESIGN PHASE

A transition from the requirements phase to the design phase took place in late January, 1982. The four requirements analysts continued on the project and two new personnel were brought in from engineering. The engineers were given the Ada requirements specification as their primary source of information about the message switch, as well as a briefing on the methodology. The design team met as a group for several weeks, sometimes in full day sessions. Various issues arose during the sessions and the need for individual thought dictated half day sessions on many occasions.

The first two weeks were spent on object-oriented design. Since none of the designers had ever participated in an object-oriented design session, the methodology group was frequently invited. It was suggested by the chief methodology engineer that the search for objects should begin with top level DFD (node A0). This turned out to be a good idea since four out of seven objects appeared at that level. The process of identifying operations to be performed on the objects gave the design team the opportunity to more closely observe the relationships between data structure components. Thus, information hiding techniques could be applied that allowed operations to be hardware independent. This was especially evident when designing the "reference storage" object, where the "construct" operation was defined to make a sequential operation work acceptably for random access or sequential hardware devices. In addition, the "message" data structure and its components provided the basic structure for the internal system software design.

During the design sessions, one designer was in charge of updating the chalkboard as the design evolved. The chief engineer refereed the discussions, especially when it was felt that enough time had been spent on a topic. The chalkboard was copied to paper by the participants at the end of each design session or when a new topic was to be considered.

The object-oriented design sessions could have continued longer, but opinions varied as to what additional usefulness would be gained from this relatively new approach. The next step in the methodology lasted about four weeks and began by utilizing traditional structured design techniques to generate a structure chart of the message switch. Although consideration was given to startup/restart, operator interface, maintenance programs, and runtime support, the primary design emphasis was related to message processing. This was because the message processing is the most important real time aspect of the system and all other software in the switch is present for its support. The support functions were somewhat limited because of the time and scope of the project. The emphasis on support functions was at the interface to the message processing function.

After several rounds of refinements, the chief engineer made assignments to team personnel. For those who participated in the requirements phase, the assignments were in a different functional area. This was not only to provide each person with some variety, but also to see if a designer could interpret the requirements written by another requirements analyst. From the time the assignments were made, the person responsible for a particular area of the message switch would be the resident "expert" during a design session involving that area. This helped to ensure that a specific designer was responsible for issues arising during the sessions pertaining to his area of expertise, and part of his time outside the sessions was to be utilized solving these problems.

Throughout February and March, the structure charts were refined, additional concurrency requirements were identified and coupling and cohesion ("goodness" of design) were evaluated. In mid-March, an in-house preliminary design review was held. All design and methodology team members were present as well as an Army representative and consultants. An overview of the message switch was presented for the benefit of the consultants. Then the object-oriented design, structure charts, and concurrency were discussed. Some minor errors were detected during the process and it was generally agreed that the review was worthwhile.

In late March three areas of the message switch were identified as potential condidates for coding as required by the contract. Message output was suggested as the number one choice and the Army subsequently agreed.

In early April interconnectivity charts were made by each designer for his area of responsibility. The two primary data structures, linetable and routing indicators, were formally organized and recorded. One designer with much hardware experience wrote a description of the executive initialization and user interface modules in narrative form. This was done to show the completeness of the system design. The area of user interface, startup/restart, fault detection, and maintenance diagnostics are just as important to the system as the application. Due to the size of the message switch and the scope of the project, the level of detail in these areas was necessarily limited. The first seven steps of the design methodology were complete and a one hundred page document was compiled for the critical design review held in mid-April.

All steps in the methodology were useful for design purposes except the interconnectivity charts, which were intended for documentation of interfaces. The information in these charts was derived directly from the structure charts.

After the CDR, the Ada Unit specifications for each Ada design unit of the structure chart were created. This served the purpose of formal specification of the calling sequences, tasks, declarations, parameter lists, and other items required by an Ada specification. The example shown in Figure 5 is one of over one hundred written for the message switch, and forms the basis for a PDL. The designers accomplished this mostly as an individual effort, with reviews by the chief programmer.

The hardware/software partitioning was done concurrently with the unit specifications. The designer who wrote the "run switch" description worked on partitioning full time. The chief engineer assisted with this process on a part-time basis. In addition to task coordination, time was spent assisting with the data structure unit specs. It was discovered that the distributed processing approach decided upon by the hardware designer could have significant impact on the structure that had been defined up to this point. The level of impact depended on where the partition was drawn. A group meeting was called to discuss the matter, which resulted in a partitioning that had a minimal design impact, yet provided good interprocessor load sharing and cost effectiveness. The conclusion drawn from the expriece was that the hardware/software partitioning should be considered earlier, particularly in a distributed environment.

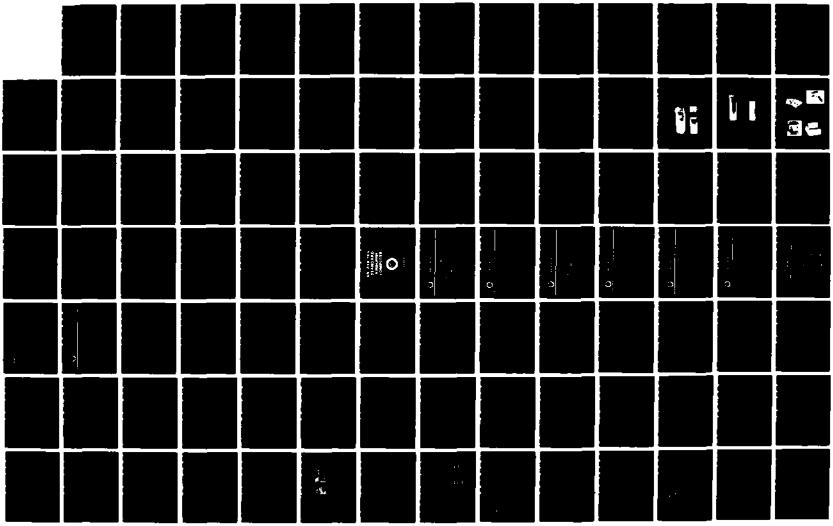
Because of the scope of the project, the detailed design was done only on the selected module and its interfaces.

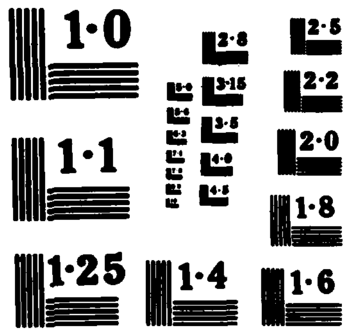
The detailed design phase was carried out differently than recommended by the Ada design methodology, mostly because the expression of the system design in Ada PDL would not be very different from the requirements RSL that already existed, except in the area of message processing support routines. A two day group meeting was held to establish the exact routines and functions needed for this support, including the memory allocation/deallocation scheme for message buffering. After establishment of this structure, each designer/programmer was to use these support (library) routines as necessary and report to the chief programmer any new support needed but not yet defined. All routines in the MESSAGE_OPS, SEGMENT_OPS, and MANAGE_INTRANSIT packages resulted from these sessions, as well as the message schema, a diagram showing the basic internal message structure. Having these packages at the start of the detail design provided a certain amount of consistency to the resulting design because each designer worked with the same building blocks, not creating individual special purpose routines that partially duplicate functions. This approach worked extremely well. Two designers were paired to design the output message validation routines, another defined the queueing to output port interface, another designed the output port task call/accept structure and support routines, and another continued on hardware/software partitioning.

The final design review called for in the methodology was held at the technical interchange meeting of May 25 and 26 near Ft. Monmouth, N.J. Essentially, there was a complete design walk through (informally held at General Dynamics the week before), a design philosophy review and explanation of the hardware/software partitioning. The requirements-to-design traceability had been completed at the prior design review meeting.

AD-A145 697 PROCEEDINGS PAPERS OF THE AFSC (AIR FORCE SYSTEMS COMMAND) AVIONICS STAND..(U) AERONAUTICAL SYSTEMS DIV WRIGHT-PATTERSON AFB OH DIRECTORATE O.. 4/6

UNCLASSIFIED C A PORUBCANSKY NOV 82 F/G 1/3 NL




1·0
2·8
2·5
3·2
2·2
3·6
2·0
4·0
1·1
1·8
1·25
1·4
1·6

MANAGE_INTRANSIT

```
package MANAGE_INTRANSIT is
  ----------------------------------------------------------------
  -- NAME: MANAGE_INTRANSIT
  -- PURPOSE: contains procedures and tasks to manage the intransit
  --          storage memory resource.
  -- PROGRAMMER: Paul Dobbs
  -- DATE: May 17, 1982
  ----------------------------------------------------------------

  type MEM_SIZE is INTEGER;
  type THRESHOLD is range 0..100;

  procedure SET_THRESHOLD(LOWER:THRESHOLD := 60;
      UPPER:THRESHOLD := 70);
  -- SETS LOWER AND UPPER THRESHOLD AS SPECIFIED IN CALL
  -- MIDDLE THRESHOLD IS SET TO THE AVERAGE OF LOWER AND UPPER

  procedure READ_THRESHOLD(LOWER,MIDDLE,UPPER: out THRESHOLD);
  -- READS CURRENT THRESHOLD SETTINGS

  task CALCULATE_THRESHOLD is
    -- CALCULATES THRESHOLD VALUES AND INITIATES OVERFLOW
       -- ACTIONS
    entry GET(BITS:MEM_SIZE);
    -- USED WHEN GETTING STORAGE
    entry PUT(BITS:MEM_SIZE);
    -- USED WHEN RETURNING STORAGE TO INTRANSIT
  end CALCULATE_THRESHOLD;

end MANAGE_INTRANSIT;
```

Figure 5. Example of an Ada Unit Specification

PROGRAMMING PHASE

Although there was no formal preprogramming Ada evaluation, a set of standards was developed by the Methodology group as programming progressed, and members of the design group consulted with each other on a regular basis to ensure that the development stayed on the right track. The selected module (output message) was programmed in Ada utilizing the primitives (from the requirements phase) and the Ada Unit specifications (from the design phase) for the application software, and the machine dependent structure diagrams and Ada unit specifications (both from the design phase), for the support (library) software. The support software consists of packages of functions and procedures (such as segment_ops) required to perform operations stated in disciplined English in the application primitives. Most of the message switch support routines, queueing interface and validation routines had been written by late-May. The "send message" routines, which required a much closer orientation to the hardware were written in June. It was quickly determined that Ada has some deficiencies when interfacing at the hardware level. Most of these concerns have been rectified by revisions to the Ada language subsequent to the July 1980 version used in the project. Also, in June, time was spent formalizing various documentation.

Since no one had any Ada programming experience, the New York University Ada Ed Compiler/Interpreter was used frequently to find syntax and semantic errors. Even though some problems were uncovered in Ada Ed, it was deemed a valuable tool. It was necessary to restructure and break up some of the larger modules in order to allow compilation, but it was also agreed that using "separate" procedures made higher level routines more understandable. This increases the need for a good cross reference tool, however.

An error analysis was accomplished to determine the frequency and types of errors made by the individual programmers and to relate this information to the experience level and training provided for each programmer. It was no surprise that programmers experienced in Pascal and Jovial understood the concepts of Ada the easiest and turned out more error free code. Typographical errors were the largest category (about 25%), but declaration errors, failure to import packages, and type mismatching accounted for half of the errors made after correction of syntax. Better training and experience should lower declaration errors, but good tools should help lower the errors attributed to import errors, and maintenance of a type dictionary would probably have decreased type mismatches. The team members who did Ada programming became very proficient in its use, partly with help from other programmers proficient in Pascal and Jovial, and partly through use of AdaEd. Approximately 7500 lines of commented code were generated by four programmers with an overall rate of approximately 7.5 lines per programmer hour. The 7500 lines of code generated represented nearly 15 percent of the total system.

CONCLUSIONS

The Ada Capability Study has been a success and has demonstrated that Ada can be used effectively in the definition, design, and programming of a large scale digital system. In a span of twelve months, a methodology was developed, personnel were trained, system requirements were defined, a design was accomplished, and a module of the system was coded. Since an Ada compiler and run time support package are not yet available, it was not possible to execute any of the implemented code. It is recognized that certain embedded real time applications may present Ada implementation problems heretofore not realized, particularly in the area of hardware interfacing and tasking.

A case study such as this is a good beginning, though it is only the beginning. Continuing research in methodology development and in the use of Ada is required with the development of compilers and an Ada environment.

In a paper of this length it is difficult to discuss all aspects of a project such as this one. Several documents were produced which describe each segment in great detail. The titles are listed in the appendix and are now available from the National Technical Information Service.

APPENDIX A

List of Documents Produced by the Army CECOM Capability Study

All were produced under contract no. DAAK80-81-C-0108

U. S. Army CECOM
Joe Kernan, DRSEL-TCS-ADA
Ft. Monmouth, N.J. 07703

by GENERAL DYNAMICS CORPORATION
Data Systems Division
Central Center

(1) Ada Integrated Methodology, NTIS #ADA 123305

(2) Ada Equivalent System Requirements Specification for AN/TYC-39 Store and Forward Message Switch, Revised, NTIS #ADA 123306

(3) Ada System Design for the AN/TYC-39 Store & Forward Message Switch, NTIS #ADA123307

(4) Ada Capability Study Source Code Document, Revised, Not assigned to NTIS

(5) Ada Capability Study Final Report, NTIS #ADA 123304

The NTIS order desk telephone number is (703) 487-4650

Michael B. Patrick, an employee of General Dynamics Data Systems Division, has 20 years of experience in computer programming and software design, specializing in real time embedded computer systems. He served as project manager for the recently completed Ada Capability Study. In that capacity he was instrumental in selecting the team members, securing expert consultants, and coordinating contract activities. Mr. Patrick has a B.S. in mathematics from the University of Texas at Arlington. He is a member of the ACM and the AdaTec special interest group.

Hal C. Ferguson, an employee of General Dynamics Data Systems Division, has extensive experience in both hardware and software design and the development of real time process control systems. He served as chief systems engineer on the Army Ada Capability Study contract. In that contract he led the design team effort which produced a well-documented redesign of an AN/TYC-39 message switching system, using Ada throughout the development process. In 1962 he received an A.S. in Electrical Engineering Technology from the University of Texas at Arlington and in 1971 a B.S. in Math/Computer Science from Texas Christian University. He is a member of the ACM and the AdaTec special interest group.

STANDARDIZATION ISSUES - NEAR TERM

SESSION CHAIRMAN: Robert Harris
AFWAL/AAAI

MODERATOR: Colonel Hugo Weichel
AFWAL/AAR

SUCCESSFUL DEVELOPMENT AND SUPPLY OF STANDARDISED COMPUTER HARDWARE AND SOFTWARE DURING A PERIOD OF RAPID TECHNOLOGICAL CHANGE

KB Dixon

Sales Manager

Ferranti Computer Systems Limited

Cwmbran Department
Ty Coch Way
Cwmbran, Gwent
South Wales

Telephone: (06333) 71111

ABSTRACT

This paper overviews the experiences of Ferranti Computer Systems as major developers, suppliers and users of UK MoD standard computer and interface hardware and software items over fifteen years.

Ferranti's leading role in this field currently centres on the development of the Military Argus computer range and associated CORAL/MASCOT compilers and operating systems with emphasis on the high performance bipolar VLSI radiation hard M700/40 processor due for release early 1983. These developments follow the earlier computer/module ranges of the FM1600 series, widely applied in Naval Command and Weapons Control and Air Traffic Control/Air Defence; and the F100-L military microprocessor, used mainly in missile, armoured vehicle and space applications.

For the future, Ferranti are heavily committed to the MIL STD 1553 system interface, and are engaged in developments relating to MIL STD 1750 in preparation for VLSI implementation, probably in the Ada/APSE era.

The paper seeks, in the light of experience, to identify the vital ingredients of success for standardisation policies, where market size can be limited, but performance and quality requirements are extremely high, during a period of technological change.

BACKGROUND

With the notable exceptions of the Royal Navy and the United States Navy, the application of standards to computer architecture, system interfaces and high order languages for military real time applications is a comparatively recent development.

For both the New World and the Old, it is not perhaps surprising that 'the Navy got there first'. It was the ships of the fleets that first provided a basis for reasonable numbers of identical computer systems, operating in an environment which, although to a degree rugged, with careful design was compatible with the computer technology of the 1960s.

In the course of being leading developers and suppliers of real time computers and associated systems to the UK MoD for nearly twenty years, Ferranti have made a major contribution to the introduction of standards within the Royal Navy.

In the early 1960s Ferranti were primarily engaged in the design and supply of computer systems for fixed ground systems for Air Traffic Control and Air Defence and for the first mobile Air Defence missiles such as Bloodhound and Thunderbird. This was in the days when each different system requirement tended to result in the development of its own, special computer. The first naval systems were for 'capital' ships such as the carrier HMS Eagle and later for the County Class Guided Missile Destroyers. The scale and operational requirements of these systems resulted in the development of the Poseidon computer and various specialised associated equipment for interfacing to the radars and other main sensors of the day; but the systems, whilst functionally complex, were in no sense modular. As the real costs of these developments became apparent one did not have to be a prophet to realise that if such systems were going to spread across the different classes of ship in the fleet, the development of a modular equipment and software base was essential.

It was soon realised that the achievement of modularity was conditional upon the introduction of standards. However, it has to be recognised that the application of 'standards' requires the acceptance of an element of constraint, whether the standards apply to processor architecture in the form of approved instruction codes, local computer to peripheral interfaces in terms of physical form and protocols, or - as we are now seeing - to system architectures by interfaces such as 1553B. In other words for a standard to be successful a balance must be struck between compromise and overkill. It cannot be denied that in the early days of standards, the element of compromise was sometimes found to be irksome, and this prevented the spread of some standards beyond the range of applications foreseen at their time of definition or choice. The integrated circuit and its followers - MSI and LSI - have, however, progressively enabled the element of compromise to reduce, as the element of overkill has become more economic. This has now reached the point where, with VLSI, the advantages of intelligently chosen standards become overwhelming.

It is important however, to recognise that technology is developing and will continue to develop at an extraordinary rate. For this reason the choice of standards must be matched to natural 'plateaus' in the general development advance, so as to allow effective exploitation. This means that some inherent flexibility of implementation is essential within the framework of each standard so that, for example, evolutionary advances in semiconductor technology can be embodied or taken advantage of without loss of compatibility, until a new 'plateau' can be recognised which implies major change.

A CASE HISTORY

A look at major milestones in the past fifteen years or so illustrates how these lessons have been learned and applied in the UK.

By 1965 the advent of the integrated circuit enabled development of the first truly modular equipment ranges to take place.

Forseeing the integrated circuit as a key element of a natural 'plateau' - although this was before the emergence of an accepted 'world standard IC range' - we in Ferranti developed and put into production our own range of fast bipolar logic - DTL Micronor II - which had the inherent environmental ability to meet military requirements. Around this we created, by complementary technology development, a controlled electrical and mechanical enviroment based upon multi layer printed circuits for cards and back planes. We engineered modular shelf based computers - such as the FM1600 and FM1600B processors - and a matching range of peripheral control units. In all, a range of over 130 modules which could be interconnected by the Ferranti 'B' Standard Interface.

From these we were able to construct systems ranging from Aircraft Carrier Command and Control systems - as on Invincible and Hermes - through more complex systems involving direct digital control of weapons and sensors, as on the Type 42 guided missile destroyers, the Type 21 and Broadsword Class 'Seawolf' frigates, down to the nuclear 'hunter killer' submarines. In the recent Falklands conflict, these systems performed reliably and well, an achievement of which we are justly proud.

In addition to operational systems, the adoption of this modular equipment range has enabled the configuration of both tactical and operations rooms trainers.

We were able, within the concepts of the F1600 series computers, to take advantage of advances in technology.

* The FM1600 and FM1600B computers gave way to the FM1600D and FM1600E computers....

* The Equipment Standard evolved, enabling a higher degree of modularity, more flexible maintenance choices for first and second line repair.

* And the printed circuit card sizes enabled, in some cases, the packaging of modules in ATR cases for airborne application. For example, the FM1600D computer, heart of the Nimrod MR aircraft 'Searchwater' radar.

In all, systems based on this modular equipment range accounting for several hundred millions of pounds sterling have been supplied by Ferranti, of which over £100M have been for export - and sales continue to grow.

This is a resounding success to be attributed to the adoption of standards.

There is a wider lesson to be learned from this, and taken into account in the choice of standards today. It is simply that, despite claims to the contrary by manufacturers of equipment designed for 'soft' civil applications, on balance military applications demand differences in design. Principally these arise from differences in environmental ability, affecting:

* choice of semiconductor technology because of need for wide temperature operating ranges consistent with high noise immunity and in some areas need for radiation hardening;

* choice of printed circuit board sizes and connectors, so as to meet conditions of vibration, shock, humidity and cooling;

* choice of interface standards, so as to give the required transfer and response speeds within the number of connections that can be allowed, consistent with pin-out limitations and emp considerations.

* choice of packaging technology to meet both environmental and maintainability requirements.

It is for these sorts of reasons that 'hardening' a commercial design can become not only a re-engineering exercise, but, more often, a re-design exercise involving the loss of compatibility with its 'soft' counterparts.

MOD POLICY - THE FERRANTI ROLE

The introduction of the FM1600/FM1600B computer equipment range coincided with the first serious attempt within MoD to define computer standards.

Driven by a concern to take advantage of development costs spread over a wider market than that provided by the limited volume of UK military requirements, the MoD 'Christchurch' committee came into being. Its aim was to choose a restricted number of 'approved' computer ranges and architectures for development and use in MoD applications, each preferably having a sound 'commercial' base. It was assumed that programming would be in CORAL. The arguments raged - and three computer ranges were chosen. Of these, however, only the Ferranti F1600 Series had an effective standard interface, and moreover, one that met the requirements of fast, real time response oriented systems. This was adopted by MoD and known as the 'Christchurch' interface. In retrospect this can be seen as a key reason why - and contrary to many people's inital expectations - the Ferranti FM1600 and FM1600B ranges have come to dominate the UK Naval computing scene, despite the claims of alternatives to have a wider 'civil' market.

The evolution and acceptance of computer standards during the 1960s and early 1970s in the naval scene has not, however, been paralleled in the UK Army and Airborne scenes.

By the late 1970s the MoD, alarmed by the implications of supporting an ever increasing range of different computers, programming languages and support software, took a fresh look at computer policy.

This time, whilst the underlying aim has been to achieve a common software support structure, based on CORAL 66 and MASCOT with a controlled change to Ada/APSE in time, practical considerations have again resulted in the choice of a restricted number of computer architectures. The principal choice is Ferranti Argus M700 series, embodying the MoD defined local bus - the EUROBUS.

Having its origin in the Ferranti commercial Argus series, a largely MoD funded development programme has produced the first military Argus processor - the M700/20. This is based on bipolar bit/slice devices and is engineered on two double Eurocards. It takes its place in an increasing range of Argus M700 series modules, ranging through memories, peripheral controllers, data link interfaces and control, monitor and test devices. It also includes interfaces to the 1553 data bus and to the UK Naval standard interface - the ASWE Serial Highway - together with System Development Aids such as our versatile 1553B Bus Controller.

Although the M700 range is sold and supported by Ferranti as an OEM product range, licence arrangements have been signed with the MoD enabling its manufacture by other MoD sub-contractors. In this context licences have been taken up by Marconi, relating to its use in the TORNADO programme; and by British Aerospace

The M700/20 is, only the first of the M700 series, having been largely configured from 'off the shelf' semiconductor components, albeit meeting military standards.

Again drawing on our in-house semiconductor capabilities, a VLSI M700 is currently under development - the M700/40.

The Ferranti semiconductor technology which makes this development possible is based on three aspects:

* the attractions of the Ferranti advanced Collector Diffusion Isolation simple bipolar process technology - FAB II;

* the pioneering and leading position of Ferranti in gate arrays;

* advances in process technology and CAD in military chip design derived from the successful F100-L military microprocessor programme.

Based on these attributes, the Argus M700/40 offers:

* High performance - operating speed over 1.6 MIPS

* Low Power Dissipation - typically 4 Watts

* High Resistance to Radiation Damage.

It is designed for use in all naval, land mobile and avionic applications - and hence to meet the full military temperature range of -55°C to +125°C.

The processor package comprises four LSI circuits, mostly based on special high performance gate arrays, mounted in 84 pin leadless ceramic chip carriers in a single hybrid package of 56mm x 91mm.

Also under development is the Argus M700/41 single card microcoputer. Based on the Argus M700/40, this includes cache memory, control and monitor logic, and FILL control; whilst for interfacing to devices off the card, a EUROBUS peripheral interface is included, and a local 'private memory' interface.

Ferranti are heavily committed both to the sale of the Military Argus family as OEM products, and to their use in Ferranti systems. These include, in addition to the new areas of Naval Commmand and Control systems wherein the M700 processors are used in a distributed form, activities in army missile, EW and avionics applications. In short, the military Argus family will be with us for a long time to come.

THE FUTURE

The emergence of Standards in the United States is undoubtedly having an increasing impact on the UK scene. Few would deny that there is every reason for the UK to join in.

In this repsect Ada/APSE is a most promising start, whilst the success of 1553B speaks for itself.

With regard to computers, the question has to be asked:

> 'With the rapid development of commercial microprocessors embodying architectural features hitherto associated with mainframe machines, will the comparatively limited MIL STD 1750A architecture stand the test of time?'

The answer, we think, is 'Probably yes'

The answer to the same question in respect of the Nebula 1862 as presently concevied is, in our view, debatable. Yet, a powerful public domain 32-bit standard machine architecture is urgently needed.

WHAT CONCLUSIONS CAN BE DRAWN

In our experience the application of standards has proved beneficial both to ourselves as a supplier of military hardware, software and systems, and to the customers we seek to serve.

In the choice and implementation of standards, cognisance of certain facts of life is essential for success:

* The standards must be imaginatively and yet practically chosen and matched to natural 'plateaus' in the advance of technology.

* Combinations of functional, environmental, reliability and maintainability requirements militate against the simple hardening of commercial equipment for military use. Design for military requirements from the outset is necessary.

* Bearing in mind the limited volume requirements of the military market, if industry is to have the confidence to put its full weight behind the development of equipment to meet such standards, it has the right to expect Government Procurement agencies to exert an element of compulsion on contractors to ensure that the standards are applied. It also follows that MoD/DoD funded development programmes are essential.

* The time for 'public' domain standards has come. There is every reason to expect the United Kingdom to join with the DoD in the choice, development and insistence on use of such common standards.

Ferranti, as a major supplier to the UK MoD are committed to support such standards as they emerge, with products engineered in leading, competitive technology.

THE AUTHOR

Keith Dixon has been associated with Ferranti computer developments since 1965, initially for Naval applications but since 1970 with emphasis on Avionic and other hard environment applications areas. He assumed marketing responsibilities for military computer LSI development and investment in 1975, and is now responsible for marketing and sales of FCSL standard military computer products, and of Army, Avionic and ATC/Air Defence systems produced by Ferranti, South Wales.

THE AN/UYK-43(V) AND AN/UYK-44(V)

A PROGRAM OVERVIEW

Captain James P. O'Donovan, USN

Naval Sea Systems Command

Washington, D.C. 20262

(202) 692-8204

Biography

Captain James P. O'Donovan
Project Manager
Navy Shipboard Tactical Embedded Computer Resources Project
Naval Sea Systems Command
Washington, D.C.

Captain O'Donovan has served in a variety of R&D, Fleet Support and acquisition engineering billets during his career.

Tours at the Long Beach Naval Shipyard, Electronics Officer aboard USS Hornet and Fleet Maintenance Electronics Officer on COMSERVPAC and CINCPACFLEET staffs have been interleaved with Washington, D.C. assignments as NTDS Project Officer for the CGN-36 and CGN-38 ship class Combat Direction System designs, and as Director for Electronics Programs on the staff of the Assistant Secretary of the Navy (Installations and Logistics).

Captain O'Donovan received his degree in Electrical Engineering from Manhattan College and has obtained both his Masters in Electronic Engineering from the Naval Postgraduate School and in Administration from George Washington University. He is also a graduate of the Industrial College of the Armed Forces.

He comes to his present job as PMS 408 from a tour as Commanding Officer of the Naval Electronics Engineering Center, Charleston, South Carolina.

Abstract

The AN/UYK-43(V) and AN/UYK-44(V) program will provide replacements for the current Navy standard computers, the AN/UYK-7(V) and AN/UYK-20(V) respectively. The AN/UYK-43(V) and AN/UYK-44(V) are planned for use in all Navy shipboard tactical systems requiring digital computers. The AN/UYK-43(V) is a militarized general purpose large scale 32-bit computer. The AN/UYK-44(V) is a militarized general purpose 16-bit embeddable processor or a standalone fully packaged minicomputer.

Both the AN/UYK-43(V) and AN/UYK-44(V) are modular in construction to permit optimal configuration according to the unique requirements of each user system, enhance maintainability and logistics supportability, and permit orderly infusion of advanced technology into the computer hardware.

Both the AN/UYK-43(V) and AN/UYK-44(V) are micro-programmed emulat^rs of their respective antecedent computers. This emulation permits use of the same support software for systems application software development, and it allows capture of existing software that runs on the respective antecedent computers. Moreover, the Instruction Set Architecture (ISA) can be extended or enhanced to meet new requirements.

INTRODUCTION

The AN/UYK-43(V) Navy Embedded Computer System (NECS) and the AN/UYK-44(V) Militarized Reconfigurable Processor/Computer (MRP/MRC) programs will provide a new generation of highly reliable computing resources for support of Navy tactical systems developed and deployed in the 1985-to-1995 time frame. Both programs are built on the current extensive base of Navy tactical embedded computer resources (TECR). The term tactical embedded computer, as used in this program overview, is defined as a computer or processor that is an integral component of any system or subsystem contributing to the combat capability of the operating forces. It follows that TECR then, are those computers/processors integral to a tactical equipment/system combined with all the software, data, support, training and personnel associated with the combat ready status of these computers/processors.

Since 1971, U.S. Navy experience has confirmed that standardization of TECR results in improved operational availability of deployed U.S. Navy systems because of commonality of spare parts, documentation, and availability of trained maintenance personnel; for similar reasons, hardware standardization reduces overall system life cycle cost. More importantly, however, Navy standards for High Order Language (HOL) use and software documentation result in significant savings in development and life cycle support of both applications and support software because of reusability, ease of configuration control, and consistency in documentation for software support activities.

GENERAL DESCRIPTION

The AN/UYK-43(V) and AN/UYK-44(V) computers are planned replacements for the current Navy standard computers, the AN/UYK-7(V) and AN/UYK-20(V) respectively. The AN/UYK-43(V) and AN/UYK-44(V) will be used in all Navy tactical systems requiring digital computers.

The AN/UYK-43(V) is a militarized general purpose large scale 32-bit computer that will be available in two enclosures, the "A" enclosure (a direct replacement for the AN/UYK-7(V) computer) and a taller "B" enclosure. (Figure 1a and 1b). The AN/UYK-43(V) is a software-compatible extension and enhancement of the AN/UYK-7(V) computer system which represents the Navy's 32-bit computer instruction set architecture base. There are currently over 2,000 AN/UYK-7(V) units in Navy applications with a similar number of AN/UYK-43(V) applications planned. The AN/UYK-43(V) computer system will represent a ma[illegible] stride forward in computer capabilities compared to the current AN/UYK-7(V) 32-bit baseline. This includes 9 times the processor throughput speed, 25 times the memory space, 6 times the input/output (I/O) throughput and greater than 3 times the reliability when configured in the "B" enclosure. The AN/UYK-43(V) computer system features advances in maintainability, built-in test (BIT) design and fault-tolerant architecture unmatched by current Navy systems. Single-button maintenance actions performed by Navy technician A-school graduates with one additional week of specialized training will suffice to diagnose and fault isolate failures in the AN/UYK-43(V) computer system. A combination of hardware, firmware, and software modules called the Fault-Tolerant Reconfiguration Module (FTRM) combine to provide a computer system with no single-point failures (when properly configured). Extended mission availability in the presence of degraded hardware is then achieveable and tactical applications software can be executed to obtain system MTBFs significantly beyond the 6,000-hour basic computer MTBF realizable without FTRM (the degree of hardware resource availability in degraded mode is an end-user system design prerogative).

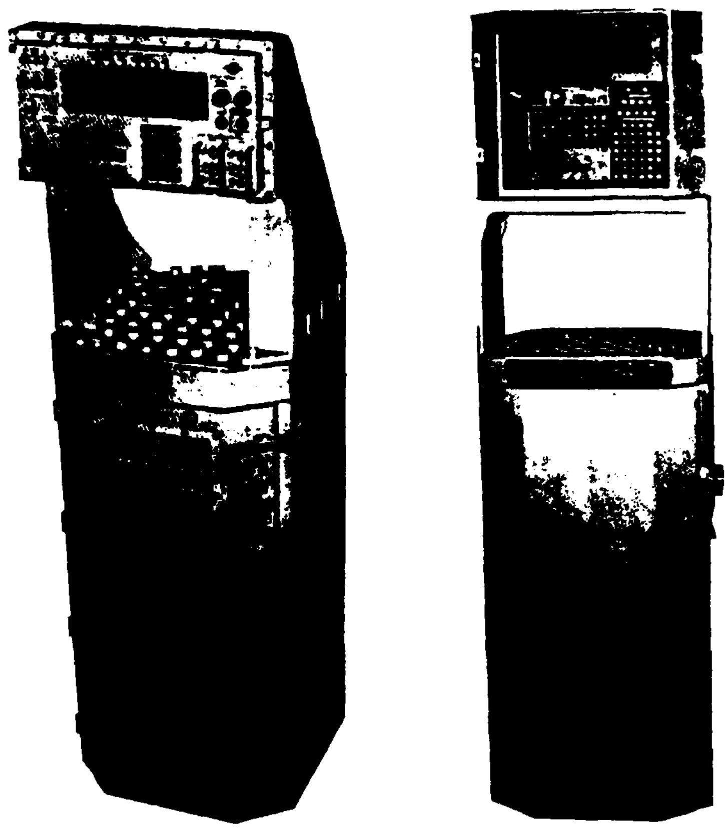


Figure 1A. AN/UYK-43(V) Computers, "A" Enclosures [IBM (Left), Sperry Univac (Right)]

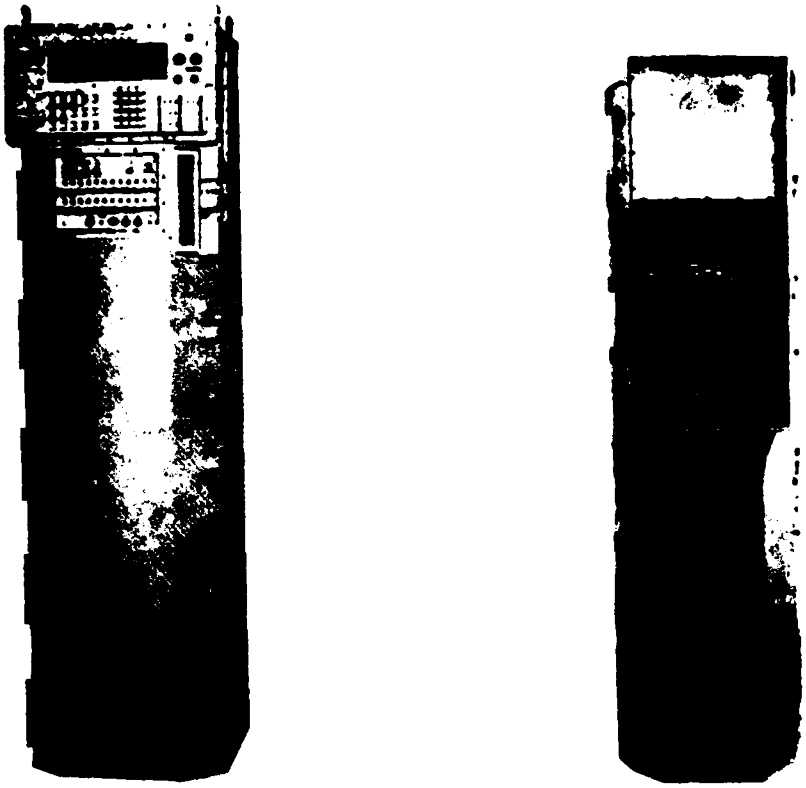


Figure 1B. AN/UYK-43(V) Computers, "B" Enclosures [IBM (Left), Sperry Univac (Right)]

The AN/UYK-44(V) is a militarized general purpose 16-bit processor and/or computer. It is available either as an unbundled set of Standard Electronic Modules (SEM), intended for direct embeddment in users equipment, called the Militarized Reconfigurable Processor (MRP) or as a traditional standalone fully packaged minicomputer called the Militarized Reconfigurable Computer (MRC). (Figure 2a and 2b).

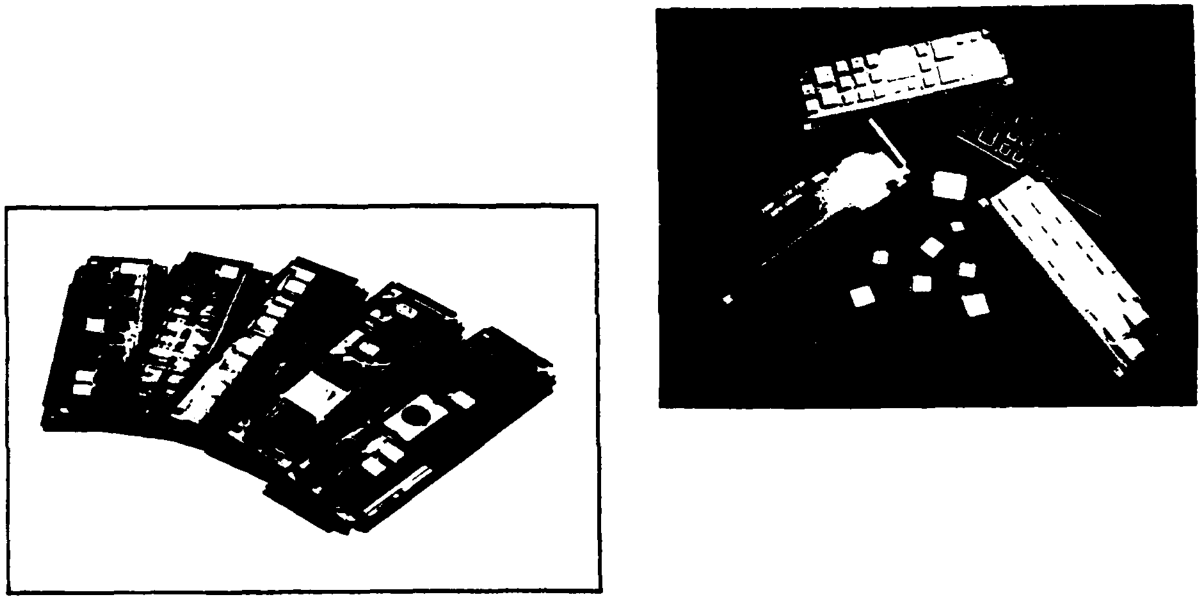


Figure 2A. Standard Electronic Modules (SEM) Used in AN/UYK-44(V) Militarized Reconfigurable Processors (MRP) [IBM (Left), Sperry Univac (Right)]

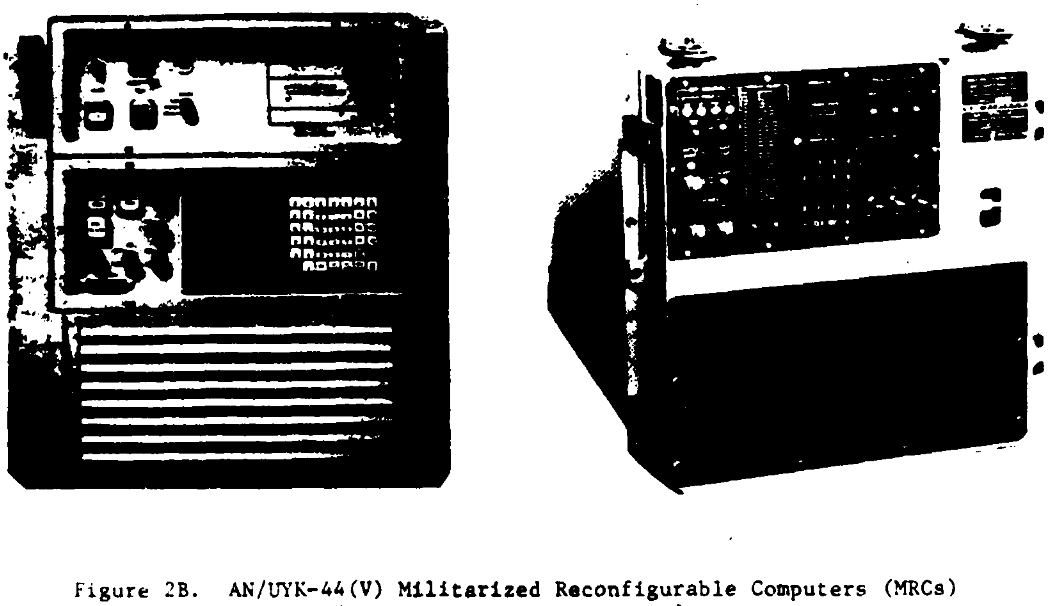


Figure 2B. AN/UYK-44(V) Militarized Reconfigurable Computers (MRCs) [IBM (Left), Sperry Univac (Right)]

The AN/UYK-44(V) Militarized Reconfigurable Processor and Computer (MRP/-MRC) is a software-compatible extension and enhancement of the AN/UYK-20(V) and AN/AYK-14(V) shipboard and airborne minicomputer family which represents the Navy's 16-bit computer instruction set architecture base. There are more than 5,000 units of the Navy's 16-bit computer family currently planned or installed in tactical systems. The MRP/MRC population planned over the next decade will reach more than 10,000 units. The MRP/MRC is a noted departure from current computer standard efforts in that the design of the component circuit cards has been specified at the card level. With the aid of the Standard Electronic Module (SEM) program, the Navy has directed the development of standard printed circuit cards and card sets which may be separately ordered and inserted into existing systems' physical equipment structures, thereby supporting the first truly embeddable military processor. Through this engineering approach, system designers will be able to acquire and install as many or as few computer parts and capabilities as their processing needs demand. The MRP/MRC additionally represents advances in computer capabilities as compared to the current AN/UYK-20(V) 16-bit base, including 2 times the processor throughput speed, 16 times the memory space, 3 times the reliability and 4 times the I/O throughput.

Both the AN/UYK-43(V) and AN/UYK-44(V) are micro-programmed emulators of their respective antecedent computers. This emulation permits use of the same support software for systems application software development, and it allows capture of existing software that runs on the respective antecedent computers. Moreover, the Instruction Set Architecture (ISA) of both the AN/UYK-43(V) and AN/UYK-44(V) computers can be extended and/or enhanced to meet new requirements. In addition, the functional partitioning of the AN/UYK-43(V) central processing unit (CPU) and flexible bussing structure of the AN/UYK-43(V) make it possible to emulate new or different ISAs and effectively create a family of computers sharing common hardware with different ISAs. However, there are no plans to do so at this time.

DEVELOPMENT OBJECTIVES

The Navy's strategy underlying the current TECR development effort can be summarized as follows:

a. Ensure that the AN/UYK-43(V) and AN/UYK-44(V) computers meet the full spectrum of systems requirements in terms of performance and operational support capabilities.

b. Capture and build on the existing tactical applications software base.

c. Ensure that the AN/UYK-43(V) and AN/UYK-44(V) computers are supported by existing programmer environments of machine transportable support software.

d. Emphasize reliability, maintainability, and enhanced operational availability (RM and A_o) in developing the AN/UYK-43(V) and AN/UYK-44(V).

e. Provide a family of computers with common instruction set architectures and high order languages.

f. Provide new computers in a variety of modular form factors and performance ranges.

g. Do not sacrifice RM and A_o for performance in the development of the AN/UYK-43(V) and AN/UYK-44(V).

The engineering objectives of the AN/UYK-43(V) full scale engineering development effort that implements this strategy are summarized as follows:

1. Provide a family of 32-bit computers capable of emulating a variety of instruction set architectures.

2. Develop a machine that emulates an ISA based upon that of the AN/UYK-7(V) with extensions to increase the capability of the final product, i.e., the ISA of the AN/UYK-43(V) computer is a superset of the ISA of the AN/UYK-7(V) computer.

3. Capture current AN/UYK-7(V) tactical applications software as a primary design objective.

4. Develop a modular design to allow configurability to meet current system demand, and at the same time, permit ease of field modification to meet future growth needs.

5. Provide complete rehostable support software concurrent with computer development to facilitate program generation facility operation.

Performance requirements for the AN/UYK-43(V) have been summarized in Table 1.

PROCESSOR SPEED	2281 KIPS (CACHE). 1521 KIPS (SCM). 860 KIPS (CORE)	
I/O SPEED	3M WORDS (32 BITS) SECOND PER IOC	
MULTI–CABINET CONNECTION	CIS PERMITS SYSTEM WIDE ADDRESSABILITY (2^{32} WORDS) 1 OR 8 CPUs AND 8 IOCs. UP TO 16 ENCLOSURES MAY BE INTERCONNECTED.	
I/O CHANNELS	32 INDEPENDENT IOAs PER IOC	
MEMORY ADDRESSING	32 BITS	
SIZE (IN.) (H, W, D)	A–40X19.88X22.33 B–72X19.88X22.33	FIT THROUGH 25" SUBMARINE HATCH
REGISTERS	1 SET OF TASKS 4 SETS OF EXEC	
BREAKPOINT REGISTERS	8 SETS OF 5 REGISTERS	
CPU I/O CONTROL	CPU CONTROLS UP TO 8 IOCs	
MTBF (MINIMUM)	≧ 6000 HRS	
AVAILABILITY (MINIMUM)	> 0.75	
COOLING	AIR/WATER	
POWER CONSUMPTION	2,500 WATTS ("A" ENCLOSURE) 5,500 WATTS ("B" ENCLOSURE)	

Table 1. AN/UYK-43(V) Performance Requirements

The engineering objectives of the AN/UYK-44(V) computer system development are summarized as follows:

1. Provide a family of 16-bit computers capable of meeting a variety of Navy applications and performance requirements.

2. Develop an embeddable version, to be identified as the Militarized Reconfigurable Processor (MRP), as a set of modules packaged in Navy Standard Electronic Module (SEM) Format B type hardware. Figure 2a)

3. Develop a standalone version, to be identified as the Militarized Reconfigurable Computer (MRC), that utilizes the MRP and is a form, fit, and function replacement for the AN/UYK-20(V). (Figure 2b)

4. Both the MRP and MRC versions emulate an ISA based upon that of the AN/UYK-20(V) and AN/AYK-14(V), with extensions to increase the capability of the computer, i.e., the ISA of the AN/UYK-44(V) is a superset of the ISA of the AN/UYK-20(V) and AN/AYK-14(V)

5. Capture current AN/UYK-20(V) tactical applications software as a primary design objective.

6. Provide a commercial grade system for MRP testing and system development, to be identified as the Microprocessor Development System (MDS)

7. Provide complete rehostable support software concurrent with computer development to facilitate program generation facility operation.

The key performance requirements for the AN/UYK-44(V) are summarized in Table 2 and Table 3 for the MRP and MRC respectively. The cycle times specified for core and semiconductor memory in the MRC are 900 nanoseconds and 350 nanoseconds respectively. However, performance data shown in Table 3 depicts throughput characteristics for an MRC with 1000 nanosecond core memory and 250 nanosecond semiconductor memory respectively.

	LOW PERFOR-MANCE MRP		LOW PERFOR-MANCE MRP	
THROUGHPUT PERFORMANCE REQUIREMENTS	CORE MEMORY (1000 NSEC)	SEMICONDUCTOR MEMORY ([illegible] NSEC)	CORE MEMORY (1000 NSEC)	SEMICONDUCTOR MEMORY ([illegible] NSEC)
CP THROUGHPUT (KOPS)	[illegible]	[illegible]	[illegible]	[illegible]
MATH PAC THROUGHPUT (KOPS)	25	[illegible]	[illegible]	[illegible]
IOC CHAIN INST (KOPS)	100	[illegible]	150	[illegible]
IOC DATA TRANSFERS (k WORDS)	250	750	[illegible]	1500
RELIABILITY	10000 HOURS		[illegible] HOURS	
MAINTAINABILITY	BUILT-IN TEST (BIT) AND DIAGNOSTICS			
OPERATING TEMPERATURE (BOTH MRP's)	[illegible]			

ADDRESSABILITY [illegible] 16-BIT [illegible] SETS OF PAGE REGISTERS

I/O [illegible] CHANNELS OF THE FOLLOWING TYPES

- MIL-STD-1397 A, B, C
- [illegible]
- MIL-STD-1397 SERIAL
- RS-232-C
- MIL-STD-[illegible]
- NATO-STD-[illegible]
- NATO-STD-[illegible]
- MIL-STD-[illegible] ([illegible])

Table 2. AN/UYK-44(V) MRP Performance Requirements

THROUGHPUT REQUIREMENTS	LOW PERFORMANCE MRC CORE MEMORY	LOW PERFORMANCE MRC SEMICONDUCTOR MEMORY	HIGH PERFORMANCE MRC CORE MEMORY	HIGH PERFORMANCE MRC SEMICONDUCTOR MEMORY
CP THROUGHPUT (KOPS)	250	350	350	900
IOC THROUGHPUT (k WORDS)	250	750	500	1500
RELIABILITY	5000 HOURS		5000 HOURS	
SIZE	19 IN W X 20 IN H X 24 IN D			
INPUT POWER	• SINGLE PHASE, 115 VOLT, 60 OR 400 Hz • THREE PHASE WYE, 115 VOLT, 60 OR 400 Hz • THREE PHASE DELTA, 200 VOLT, 60 OR 400 Hz			
COOLING	• AIR COOLED • WATER COOLING (OPTIONAL)			
MEMORY CAPACITY	• 256K OF CORE MEMORY, 512K OF SEMICONDUCTOR MEMORY OR MIX			
I/O CAPACITY	• UP TO 16 I/O CHANNELS OF ANY TYPE, ANY MIX			
WEIGHT	• 105 LBS			
OPERATING TEMPERATURE	• −54°C TO +65°C			
MAINTAINABILITY	• BIT AND DIAGNOSTICS; 15 MIN MTTR			

Table 3. AN/UYK-44(V) MRC Performance Requirements

PROGRAM SCHEDULES

In both the AN/UYK-43(V) and the AN/UYK-44(V) projects there is a competitive development of two candidate systems. The AN/UYK-43(V) development contracts were awarded to IBM, Owego, New York and to Sperry Univac, St. Paul, Minnesota in September 1980. The AN/UYK-44(V) contracts were awarded to IBM, Manassas, Virginia and Sperry Univac, St. Paul, Minnesota in September 1980. Commander, Naval Sea Systems Command (PMS 408) is the Development Agent for both systems with Navy laboratory support teams providing technical assistance in the specification and testing process of the candidate systems.

The AN/UYK-43(V) Engineering Development Models (EDMs) are scheduled for March 1983 delivery upon completion of contractor-performed, government-witnessed testing. After delivery, additional government lab testing will be conducted to validate Fault-Tolerant Reconfiguration Module (FTRM) performance, software transportability, automated maintenance features, and other related performance characteristics. Source selection of the prime contractor for the production phase of the AN/UYK-43(V) project is planned for third quarter FY83 after delivery of candidate units. First production units are scheduled for December 1984 delivery. (Figure 3)

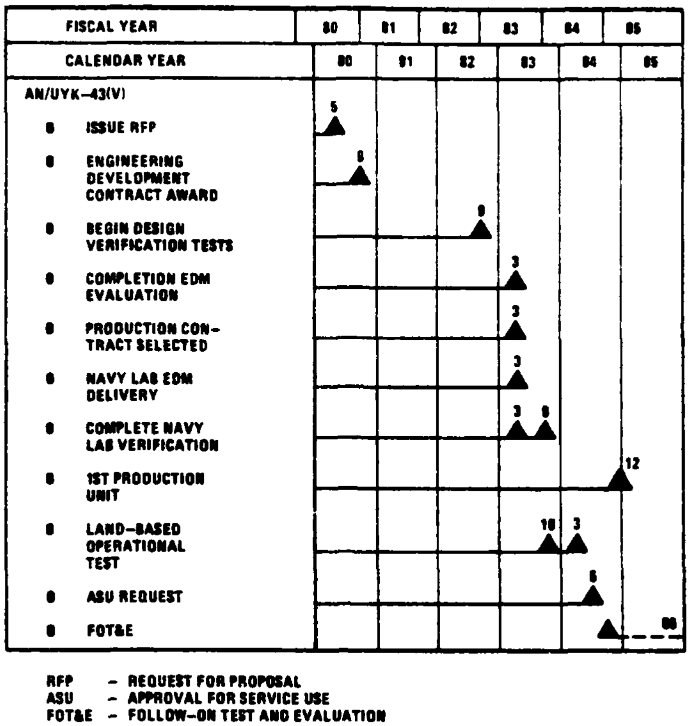


Figure 3. AN/UYK-43(V) Program Milestones

The AN/UYK-44(V) MRP/MRC EDMs have been undergoing periodic delivery (card sets and support systems) since 18 December 1981. Advanced Production Equipment (APE) deliveries will be complete in November 1982, except for the MRC packaged computer. Both contractor and independent lab testing is in process. Testing will be completed by January 1983. Source selection is planned for not later than the end of the third quarter FY83 contingent upon completion of testing. First production MRP units will be available prior to December 1983. Final MRC testing and MRC production unit deliveries are anticipated by December 1984. (Figure 4)

COMPARABILITY WITH ANTECEDENT COMPUTERS

The AN/UYK-43(V) is designed to be used in one of two enclosures. The "A" enclosure is a direct physical and plug-compatible replacement for a single bay AN/UYK-7(V) computer. The "B" enclosure has the same footprint as a AN/UYK-7(V) single bay, but is taller. The AN/UYK-43(V) captures any AN/UYK-7(V) software that does not rely on:

a. Precise AN/UYK-7(V) instruction execution time dependencies.

b. AN/UYK-7(V) physical partitioning (implementation dependent configuration or hardware dependent software).

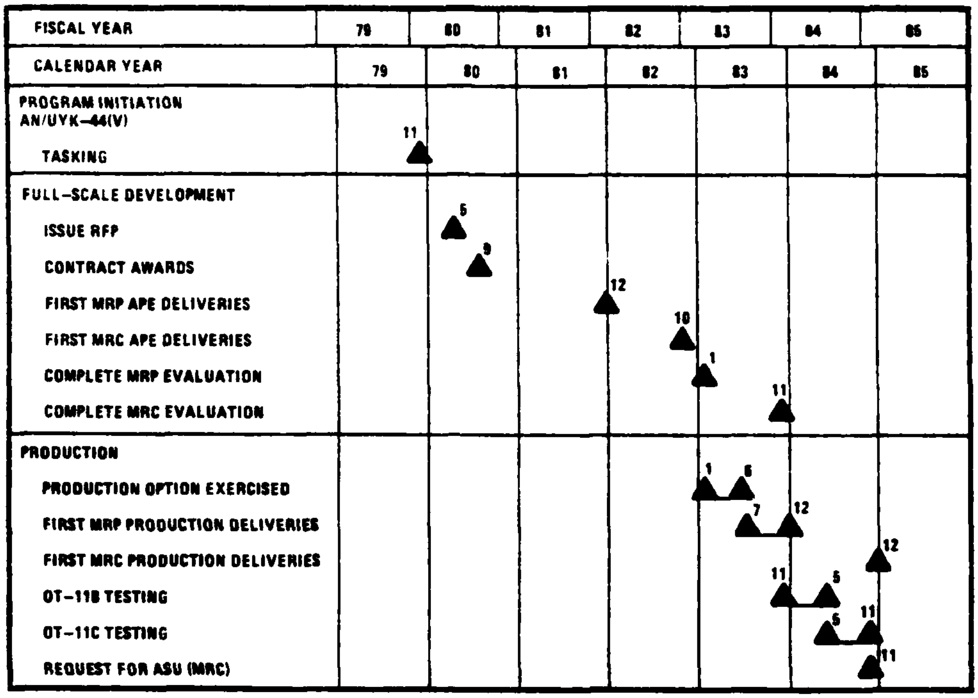


Figure 4. AN/UYK-44(V) Program Milestones

c. Use of AN/UYK-7(V) instructions which are incompatible with Navy approved AN/UYK-43(V) ISA extensions.

Software transportability is further facilitated by the provision of an AN/UYK-7(V) compatibility mode in the AN/UYK-43(V) that is a faithful emulation of the AN/UYK-7(V) CPU and IOC ISAs. The AN/UYK-43(V) can operate in the following compatibility modes:

1. AN/UYK-43(V) executive and task program state

2. AN/UYK-7(V) executive and task program state

3. AN/UYK-43(V) executive, AN/UYK-7(V) task "mixed mode" program state.

In comparison to the AN/UYK-7(V), the AN/UYK-43(V) provides IOC memory protection and significant IOC Instruction Set Architecture (ISA) and computer hardware technical features of major benefit to combat system designers and tactical applications programmers.

The following features are noteworthy computer system state-of-the-art implementations in the AN/UYK-43(V) design.

a. A Computer Interconnection System (CIS) that:

1. Provides the extension of the internal computer bus outside of the computer enclosure.

2. Allows processors in one computer to address memory and processors in other computers.

3. Allows direct processor to memory data transfers without I/O channels.

4. Functions within currently defined ISA specification of AN/UYK-43(V), i.e., it is transparent to the programmer.

5. Results in a virtual computer with 4 billion words of memory, 8 CPUs, 8 IOCs, and 256 I/O Channels.

b. A fault tolerant concept for the computer system that provides for:

1. Detection of a fault through hardware/firmware/software containment of the fault and prevention of error propagation.

2. Localization of a fault to a functional module and isolation of that functional module from the computer.

3. Applications software notification of available computer resources for applications software reconfiguration, if required.

4. On-line repair of failed functional modules and verification of repair action.

5. Module activation and return to the computer resources inventory. Applications software is notified for restoration of full capability, if required.

c. Maximum allowable integrated circuit junction temperatures of 80°C in the water-cooled enclosure and 90°C in the air-cooled enclosure under worse case environmental conditions (i.e., 50°C ambient inlet air temperature and 40°C ambient inlet water temperature).

d. Designed system redundancy that essentially eliminates single point failures with proper resource configuration in the AN/UYK-43(V) "B" enclosure.

e. Maintainable by operator ratings with one week of maintenance training.

A summary comparison of other AN/UYK-43(V) computer system characteristics in comparison to the AN/UYK-7(V) are shown in Table 4.

The AN/UYK-44(V) provides a performance range to allow the user to trade performance requirements against cost. The high performance AN/UYK-44(V) MRC has:

a. 2 times the speed (in KIPS) of the AN/UYK-20(V),

b. 8 times the memory capacity,

	AN/UYK-7(V)	AN/UYK-43(V) (ENCL A)	AN/UYK-43(V) (ENCL B)
THROUGHPUT (MINIMUM)	507 KOPS	2,028 KOPS	4,056 KOPS
MEMORY CAPACITY	96K	1,25M	2.5M
CENTRAL PROCESSORS	1	1	2
INPUT/OUTPUT CONTROLLERS	1(≈1M W/S)	1(≧ 3M W/S)	2(≧ 6M W/S)
INPUT/OUTPUT CHANNELS	16	24	64
MEMORY ADDRESSABILITY	256K	4G	4G
MTBF (MINIMUM)	2,500	≧ 6,000	≧ 6,000
AVAILABILITY (MINIMUM)	N/A	> 0.75	> 0.75
COOLING	AIR	AIR/WATER	AIR/WATER
DIMENSIONS (WXHXD)	20X41X22 IN	AN/UYK-7(V) FOOT-PRINT 41 IN HEIGHT*	AN/UYK-7(V) FOOT-PRINT/6FT HEIGHT*
POWER CONSUMPTION (WATTS)	2,500	2,500	5,500

I/O INTERFACES (MIL-STD-1397)
(*NEW DESIGNS)

TYPE A - NTDS SLOW
TYPE B - NTDS FAST
TYPE C - ANEW
TYPE D - SERIAL (NTDS)
TYPE E - LOW-LEVEL SERIAL (NATO)
*TYPE F - MIL-STD-1553 B
*TYPE G - RS-449 (RS-232 COMPATIBLE)
*TYPE H - TYPE C COMPATIBLE HIGH-THROUGHPUT PARALLEL (500,000 32-BIT WORDS/SEC)

Table 4. AN/UYK-43(V) Comparability to AN/UYK-7(V)

c. the same number of I/O channels (higher throughput capability), and

d. 1.25 times the reliability.

Table 5 compares an AN/UYK-20(V) maximum configuration to that of a high performance AN/UYK-44(V) MRC. The ISA of the AN/UYK-44(V) is a superset of the AN/UYK-20(V) ISA, which allows the direct capture of AN/UYK-20(V) software and use of existing AN/UYK-20(V) program generation facilities. The advancements that the AN/UYK-44(V) provides over the AN/UYK-20(V) that will be of most interest to combat system designers and tactical applications programmers are summarized as follows:

a. Low and high performance embeddable card sets designed on SEM Format B form factor for use in distributed processing and qualified to Level II SEM environmental requirements (including 70K feet altitude testing).

b. Memory addressing increased to 4096K words.

c. Two modes (executive and task) of program operation.

d. Memory protection and malfunction detection.

e. Memory-mapped I/O provided for 64 devices independent of the IOC

	AN/UYK-20(V)	AN/UYK-44(V)MRC
THROUGHPUT:	450 KIPS	350 KIPS TO 900 KIPS
MEMORY CAPACITY:	65K	512K
CENTRAL PROCESSORS:	*	1
I/O CONTROLLERS:	*	1
I/O CHANNELS:	16	16
MEMORY ADDRESSABILITY:	65K WORDS	4M WORDS
MTBF:	4000 HRS	5000 HRS
COOLING:	AIR	AIR/WATER
DIMENSIONS	19"X20"X24"	19"X20"X24"
POWER	1000 WATTS	900 WATTS
WEIGHT	220 LBS	165 LBS

*THE AN/UYK-20(V) CONTAINS A SINGLE MICRO-CONTROLLER FOR CPU AND IOC INSTRUCTIONS SUCH THAT THE CPU AND IOC ARE NOT INDEPENDENT.

Table 5. AN/UYK-44(V) MRC Comparison with the AN/UYK-20(V) Computer

f. Memory management instructions allowing unpaged access.

g. IOC functionally independent from CPU.

h. Up to 4 IOCs for a total of 64 I/O channels.

i. Specified AN/AYK-14(V) CP Instructions in addition to AN/UYK-20(V)

j. AN/AYK-14(V) Extended Arithmetic Unit (EAU) instructions added to MATH PAC.

k. Page registers increased from 1 to 4 sets of 64 registers.

l. Improved RMA

m. Lower power and weight.

n. Increased throughput.

o. Packaged Militarized Reconfigurable Computer that uses either the low or high performance MRP card sets.

COMPUTER MODULARITY

Both the AN/UYK-43(V) and AN/UYK-44(V) are modular in construction to permit optimal configuration according to the unique requirements of each user system. In addition, their modular nature enhances maintainability and logistics supportability. This modularity also permits orderly infusion of advanced technology into the computer hardware to improve performance and/or reliability, or to reduce size, weight, power consumption, or cost.

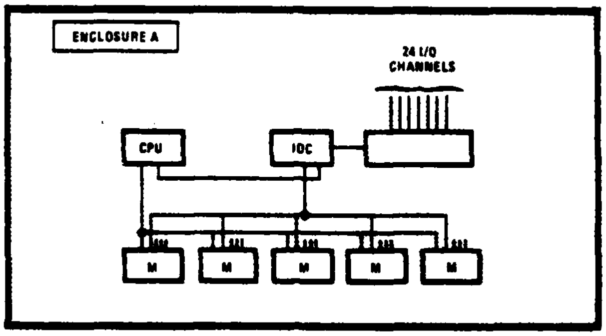
The AN/UYK-43(V) and AN/UYK-44(V) are required to meet or surpass all the applicable Navy invoked specifications for shock, vibration, air and structureborne noise, electromagnetic compatibility, electromagnetic protection, safety, electromagnetic interference and environmental stress testing in according with MIL-E-16400(G). In addition, the AN/UYK-44(V) design discipline is specified in accordance with MIL-M-28787 for standard Electronic Module (SEM) design.

The details of the AN/UYK-43(V) computer system's modularity are presented in Figure 5. The architecture of the AN/UYK-43(V) "A" and "B" enclosures previously described are depicted in Figures 6 and Figure 7 respectively. In each figure, the salient features of the architecture are listed.

MODULES	ENCLOSURE TYPES	
	A	B
CPUs	0–1	0–2
IOCs	0–1	0–2
I/O CHANNELS	0–24	0–64
MEMORY MODULES	0–5	0–10
POWER SUPPLIES	1	2
CIS	0–1	0–1
D/CP	1–2	1–2
ROCU	0–1	0–1
SIZE (WXHXD INCHES)	20 X 41 X 22	20 X 72 X 22
WEIGHT (POUNDS)	500	750
POWER (WATTS) CONSUMPTION	2500	5500

Figure 5. AN/UYK-43(V) Modular Enclosure Configurability

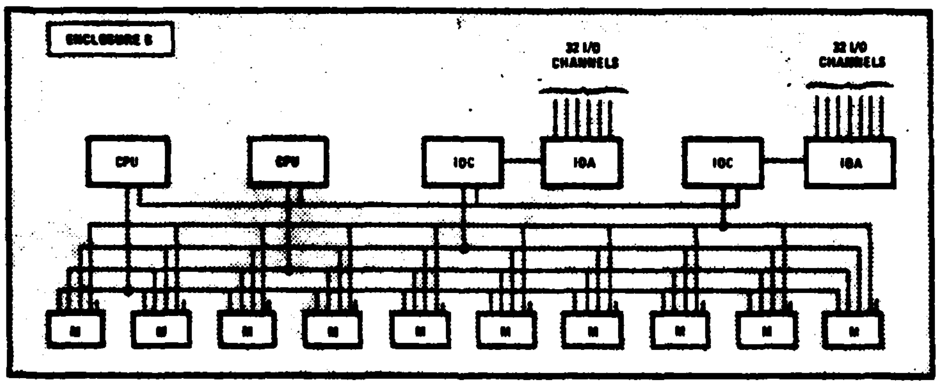
The "A" enclosure is intended primarily as a plug-compatible replacement for the single-bay AN/UYK-7(V). The "A" enclosure provides improved performance (e.g., with cache memory, 4.5 times the AN/UYK-7(V) processing capability, 3.5 times the I/O throughput capability, 50 percent more I/O channel capacity, and 14 times the memory capacity), a Fault-Tolerant System Reconfiguration Module (FTSRM), and an extended architecture in the same physical envelope.



SALIENT ATTRIBUTES

- FAULT-TOLERANT DESIGN
  - AUTOMATIC DETECTION
  - AUTOMATIC RECOVERY
  - CASUALTY REACTION HIERARCHY
  - FAULT-TOLERANT AND SYSTEM RECONFIGURATION MODULE (FTRM)
  - BIT (BUILT-IN TEST)
  - PARITY ON CORE MEMORY AND MODULE DATA TRANSFERS
  - ERROR CORRECTION ON SEMICONDUCTOR MEMORY
- AUTOMATED MAINTENANCE FACILITIES
- EXTENSIVE PERFORMANCE MONITORING AND PROGRAM DEBUG FEATURES
- INPUT/OUTPUT CONTROLLER - (IOC) ARITHMETIC PROCESSING CAPABILITY TO OFFLOAD MAIN CPU INTERACTION
- CPU-IOC EFFECTIVELY CREATES DUAL PROCESSING CAPABILITY
- 32KX34 BITS CORE MEMORY MODULE MINIMUM SIZE (64KX34 BITS MAX)
- 64KX39 BITS SEMI-CONDUCTOR MEMORY MODULE MINIMUM SIZE (1MX39 BITS MAX)

Figure 6. AN/UYK-43(V) "A" Configuration Architecture



NOTE: DIAGRAM ILLUSTRATES EACH REQUESTER HAS SINGLE MEMORY BUS; DESIGN MAY INCLUDE TWO BUSES PER REQUESTER

SALIENT ATTRIBUTES

- ENCLOSURE A ATTRIBUTES PERTAIN
- DESIGN REDUNDANCY
  - DUAL POWER SUPPLIES
  - FULL INTERCONNECTIVITY OF REQUESTERS AND MEMORY
  - SYSTEM SURVIVABILITY (BUILT-IN HARDWARE REDUNDANCY WITH SOFTWARE ABILITY TO RECONFIGURE)
- CPUs-IOCs EFFECTIVELY CREATE QUAD PROCESSING CAPABILITY

Figure 7. AN/UYK-43(V) "B" Configuration Architecture

The "B" enclosure is the centerpiece of the AN/UYK-43(V) system. It is, in effect, a fully redundant, duplexed system in a single enclosure. It provides more computing power than two four-bay AN/UYK-7(V)s (five times the memory), while occupying one-eighth the deck space and consuming one-fourth the electric power. The "B" enclosure can contain twice the functional modules of the "A" enclosure. In addition, a complete fault tolerant reaction and on-line automated maintenance facility can be employed to yield a very high system level MTBF/availability. The "B" enclosure has independent memory modules providing a total of 2.56 million 32-bit words of main memory or 10 million bytes of memory. The two CPUs are capable of performing 4.5 million instructions per second (cache), and two IOCs can transfer and process I/O data at a rate greater than 6 million words per second over 64 I/O channels.

For either enclosure type, each I/O channel may be independently selected from a set of eight standard interfaces. The Computer Interconnection System (CIS) allows the system designer to implement a virtual multicomputer processing complex in which a processor in one enclosure may have direct access to memory modules and initiate I/O chains in as many as 15 other enclosures or peripherals.

AN/UYK-43(V) makes extensive use of the latest Large-Scale Integration (LSI) and Very Large-Scale Integration (VLSI) technology, providing for computer reliability, maintainability, performance, and capacity far superior to earlier military computers at lower costs. The basic architecture and physical and functional partitioning of the AN/UYK-43(V) are designed to facilitate the infusion of future technologies. The design allows upgrading of a functional module with no impact on other functional modules or the interconnection bus system. The AN/UYK-43(V) design allows for the emulation of a wide variety of ISAs through microcode changes and/or processor replacement. New instructions may be easily added to the present ISA via microcode changes. A complete ISA change may be accomplished by the replacement of the processor with one that executes a different ISA. The new processor need only meet the physical, bus electrical, and protocol specifications.

The AN/UYK-44(V), as previously noted, is deliverable as an MRP card set implemented on SEM Format B modules with standardized I/O, common data bus, low power consumption, user configurable memory, and power supplies. Additionally, it can be ordered as an MRC computer in a standard mechanical enclosure that is compatible with the AN/UYK-20(V) footprint, possesses standard I/O interfaces, incorporates the MRP and many of its options, possesses up to 1 million words of memory, up to 2 IOCs, a 32-bit serial and/or parallel I/O channel mix, and is rack or deck mountable in either an air or water cooled configuration. Further, the AN/UYK-44(V) MDS can be acquired to support the embedded MRP with optional features, and utilized to support operation and test of the MRP and AN/UYK-44(V) software debug and test functions through an interactive display/ keyboard interface.

The generic architecture of the AN/UYK-44(V) is shown in Figure 8. The MRP as shown may consist of:

a. A low performance or high performance processor and some combination of the following optional features.

1. I/O Controller (IOC)

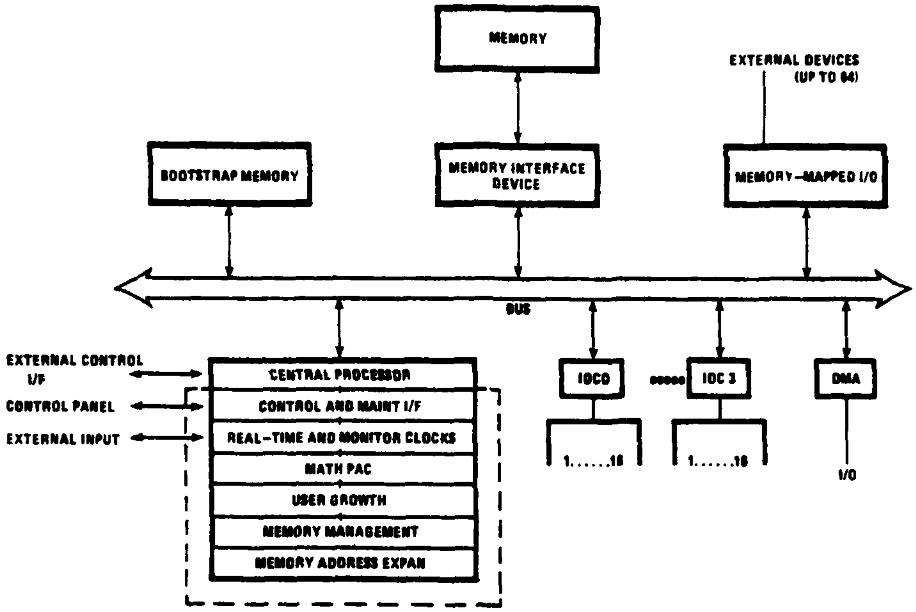


Figure 8. AN/UYK-44(V) Generic Architecture

2. I/O Channel Adapters

3. Control and Maintenance Interface

4. MATH PAC

5. Bootstrap Memory

6. Real Time and Monitor Clocks

7. Memory Address Expansion

8. Memory Interface Modules

9. Memory Mapped I/O Modules

10. Direct Memory Access (DMA)

Key performance characteristics of several of these options are summarized in the following paragraphs.

a. The AN/UYK-44(V) I/O function allows for:

1. One to four I/O controllers

2. Independent operation of each IOC

3. Up to 16 I/O channel adapters (serial and/or parallel) per I/O controller

4. Up to 32 program initiated I/O chains per I/O controller

5. I/O instruction repertoire - same format as CPU

6. Optional memory address expansion to 4 million words

b. The I/O Channel Adapters (IOAs) are implemented in two channel groups for the following standard interfaces:

1. PARALLEL CHANNELS

(a) MIL-STD-1397, TYPE A

(b) MIL-STD-1397, TYPE B

(c) MIL-STD-1397, TYPE C

2. SERIAL CHANNELS

(a) MIL-STD-188C, SYNC, ASYNC

(b) EIA-STD-RS-232-C SYNC, ASYNC

(c) VACALES

(d) MIL-STD-1397, TYPE D

(e) NAT-STD-4153 (PROPOSED TYPE E)

(f) NAT-STD-4156

c. The MATH PAC module functions include:

1. Square root

2. Trigonometric and hyperbolic vector and rotate

3. Floating point arithmetic, square root, trigonometric, exponential and natural logarithm

4. Double precision multiply and divide

5. Algebraic left and right quadruple shifts

d. The Real Time and Monitor Clock and Bootstrap modules provide:

1. 192-word Bootstrap capacity

2. Preprogrammed or customer defined microcoded instructions

3. 32-bit Real Time Clock

4. 16-bit Monitor Clock

5. 1 KHz, 32 KHz, or external clock rate up to 50 KHz

e. The Memory Address Expansion module allows for addressability up to 4 million words via four sets of 64-page registers.

f. The Memory Interface module provides access to a maximum of 8 memory modules of up to 256K core memory words or 512K semiconductor memory words (or a mix of both) with a cycle time range of 250 nanoseconds to 1000 nanoseconds.

g. Memory mapped I/O (MMIO) modules allow the CP and/or the IOC to communicate with peripherals by treating the control and data registers of each memory mapped I/O module as memory locations. Features of the MMIO capability include:

1. Reduction in memory conflicts

2. Maximum of 64 MMIO devices per processor with a 250 nanosecond cycle time

3. Asynchronous timing

4. Parallel or serial data transfers are permitted.

h. The AN/UYK-44(V) MRP provides a DMA capability which allows interfacing directly to the common data bus. The DMA capability is compatible with the AN/UYK-20(V) design.

The AN/UYK-44(V) MRC includes the MRP and selected options plus a single DMA channel per MRC cabinet. In addition to the features shown in Table 5, the MRC design provides for an expansion cabinet option containing one IOC, 16 I/O channels, and 512K semiconductor or 256K core memory.

COMPUTER MAINTAINABILITY

a. AN/UYK-43(V) Summary

The AN/UYK-43(V) achieves a 15 minute Mean Time To Repair by using on-line and resident diagnostics to isolate computer faults to a single Line Replaceable Unit (LRU) 90% of the time, and to within three LRU's 98% of the time. Ease of access, modular construction, fault tolerant design concepts, on-line repair capability and simplified personnel training all contribute to ease of maintenance. The AN/UYK-43(V) requires no periodic preventive maintenance other than monthly cleaning of filters, lubrication of doors and similar actions not of a time critical nature and not requiring system shut-down to accomplish. All LRUs are readily accessible and can be quickly removed and replaced without special tools.

Maintenance can be performed by a class "A" electronics school graduate with 36 hours of special training.

Fault tolerance is achieved through built-in reliability, hardware redundancy, and hardware/software features which isolate and identify faults with the aid of automated maintenance facilities.

On-line repair is implemented in a user-friendly fashion and is concurrent with other computer actions. The operator sees an alphanumeric display with clear, concise repair instructions. Only a single button is pushed to execute diagnostics. Modules that have not experienced failure continue to process normally. The system is not taken down or off-line to isolate the failed LRU.

b. AN/UYK-44(V) Summary

Using advanced Built-in-Test (BIT) microcode firmware and macro self-test programs, the AN/UYK-44(V) achieves a Mean Time To Repair (MTTR) of 15 minutes. The diagnostic and BIT system detects at least 98% of all MRP/MRC malfunctions, including those preventing successful program loading, and will isolate at least 80% of all detected malfunctions to a single module, at least 95% to two modules and at least 98% to three modules. The microcode firmware coupled with the macrodiagnostic provides an easy to use, minimum experience to operate and fast malfunction detection capability as a means of achieving ease of maintenance on the AN/UYK-44(V). Corrective maintenance is primarily accomplished with pluggable modules, with only one preventive task (cleaning filters) required of the MRC. All maintenance tasks can be performed easily and do not require high skill level personnel, extensive support equipment or extensive documentation. With the exception of the rear-mounted I/O channel cable connectors, all replaceable elements of the MRC are accessable through the front cabinet doors.

An easy to use operator and maintenance panel mounted on the front of the MRC simplifies and reduces MTTR. This panel consists of a function/mode select keyboard and an operator interface display. The microcode firmware can be executed as an in-line macroinstruction, as a press-to-test operation enabled through the operator and maintenance panel, or at initiation when the stop/master clear switch is momentarily depressed. Once diagnostic testing is initiated, fault isolation is accomplished by display of fault group numbers providing direct reference to the failed module(s).

CONCLUSIONS

Current U.S. Navy standard shipboard computers no longer have sufficient performance and capacity to satisfy current and projected operational requirements. The limited performance and capacity of current standards is causing increased life cycle costs for tactical embedded computer resources due to requirements for complex system architectures to overcome performance/capacity limitations, and complex system software designs to support these complex architectures.

The new generation AN/UYK-43(V) and AN/UYK-44(V) will provide required Navy operational enhancements and will yield significant gains in reliability, maintainability, availability and fault tolerance at a substantially lower life cycle cost to the Navy. Software capture will retain the substantial investments in existing Navy software. Commonality of spares, training, technical manuals, software, maintenance and operator interface will allow interchangeability of equipment and personnel between platforms and between systems on the same platform. During critical operations, computer system availability will be significantly increased. Standardization will allow cost effective evolutionary preplanned product improvement upgrades of memory capacity, ADA capabilities, and VHSIC-like technology insertion. The physical and functional partitioning of the AN/UYK-43(V) and AN/UYK-44(V) are designed to facilitate the infusion of future technologies to maintain these computers on the leading edge of the produceable computer state-of-the-art.

GLOSSARY OF ABBREVIATIONS

A_o	Operational Availability
APE	Advanced Production Equipment
BIT	Built-in Test
CIS	Computer Interconnection System
DMA	Direct Memory Access
EAU	Extended Arithmetic Unit
EDM	Engineering Development Model
FTRM	Fault-Tolerant Reconfiguration Model
FTRSM	Fault-Tolerant System Reconfiguration Model
HOL	High Order Language
ISA	Instruction Set Architecture
LRU	Line Replaceable Unit
LSI	Large-Scale Integration
MDS	Microprocessor Development System
MMIO	Memory-Mapped I/O
MRC	Militarized Reconfigurable Computer
MRP	Militarized Reconfigurable Processor
MTBF	Mean Time Between Failures
MTTR	Mean Time to Repair
NECS	Navy Embedded Computer System
RM	Reliability and Maintainability
RMA	Reliability, Maintainability and Operational Availability
SEM	Standard Electronic Module
TECR	Tactical Embedded Computer Resources
VHSIC	Very High Speed Integrated Circuit
VLSI	Very Large-Scale Integration

AN/AYK-14(V)

STANDARD

AIRBORNE

COMPUTER

Henry H. Mendhall

Computer Resources and Avionics Systems Division

Naval Air Systems Command

Washington D.C. 20360

The AN/AYK-14(V) Program Overview

The AN/AYK-14(V) Program represents a family of Navy designated standard logic modules. These modules can be configured in a variety of combinations to produce processing elements for incorporation into subsystems of a weapon system or produce complete self contained general purpose computer systems. The design of the AN/AYK-14(V) modules includes features which permit AN/AYK-14(V) users to select only those functions necessary to meet individual application needs. The flexibility of this "building block approach" also allows future hardware reconfiguration and growth to accomodate changes in an applications needs with minimal impact to the weapon system. The functionally partitioned modular design also enables the AN/AYK-14(V) Program to capitalize on advances in technology in a planned orderly manner which is transparent to the users.

This brief presents program status and an overview of the AN/AYK-14(V) as a computer system and as a processor. Examples will be provided which demonstrate the inherent flexibility in the AN/AYK-14(V) design which enables it to meet a wide range of applications and take maximum advantage of technology infusion and pre-planned product improvement.

AN/AYK-14(V) STANDARD AIRBORNE COMPUTER

OVERVIEW

211050.0

AN/AYK-14(V)

OUTLINE

- PROGRAM GOALS
- AN/AYK-14(V) CONCEPT
- GENERAL FEATURES
- PROGRAM MILESTONES
- NEXT STEP

211051.0

AN/AYK-14(V)

AN/AYK-14(V) PROGRAM GOALS

- REDUCE PROLIFERATION OF UNIQUE COMPUTER SYSTEMS
- UTILIZE COMMON SOFTWARE SUPPORT AND LOGISTICS BASE FOR ALL USERS
- BE ADAPTABLE TO CHANGING OPERATIONAL NEEDS
- SATISFY MAJORITY OF AIRBORNE COMPUTER SYSTEM REQUIREMENTS BETWEEN 1976 AND 1990

211062.0

AN/AYK-14(V)

THE AN/AYK-14(V) IS *NOT* A COMPUTER

211063.9

AN/AYK-14(V)

THE AN/AYK-14(V) IS

- A FAMILY OF NAVY STANDARD FUNCTIONALLY PARTITIONED MODULES
  - PROCESSOR MODULES
  - MEMORY MODULES
  - INPUT/OUTPUT MODULES
  - POWER SUPPLY MODULES
  - CHASSIS MODULES
- A BUILDING BLOCK APPROACH TO COMPUTER SYSTEMS
- AN EMBEDDED PROCESSOR FOR SUBSYSTEMS

211054.0

AN/AYK-14(V)

AN/AYK-14(V) GENERAL FEATURES

- GENERAL PURPOSE, 16 BIT
- THRUPUT UP TO 800 KOPS
- DUAL PROCESSING
- UP TO 512K WORDS MAIN MEMORY
- COMPATIBLE WITH AN/UYK-20/UYK-44 SUPPORT SOFTWARE
- EXTENSIVE BUILT IN TEST
- GENERAL PURPOSE PARALLEL BUS (2) ARCHITECTURE

211085.0

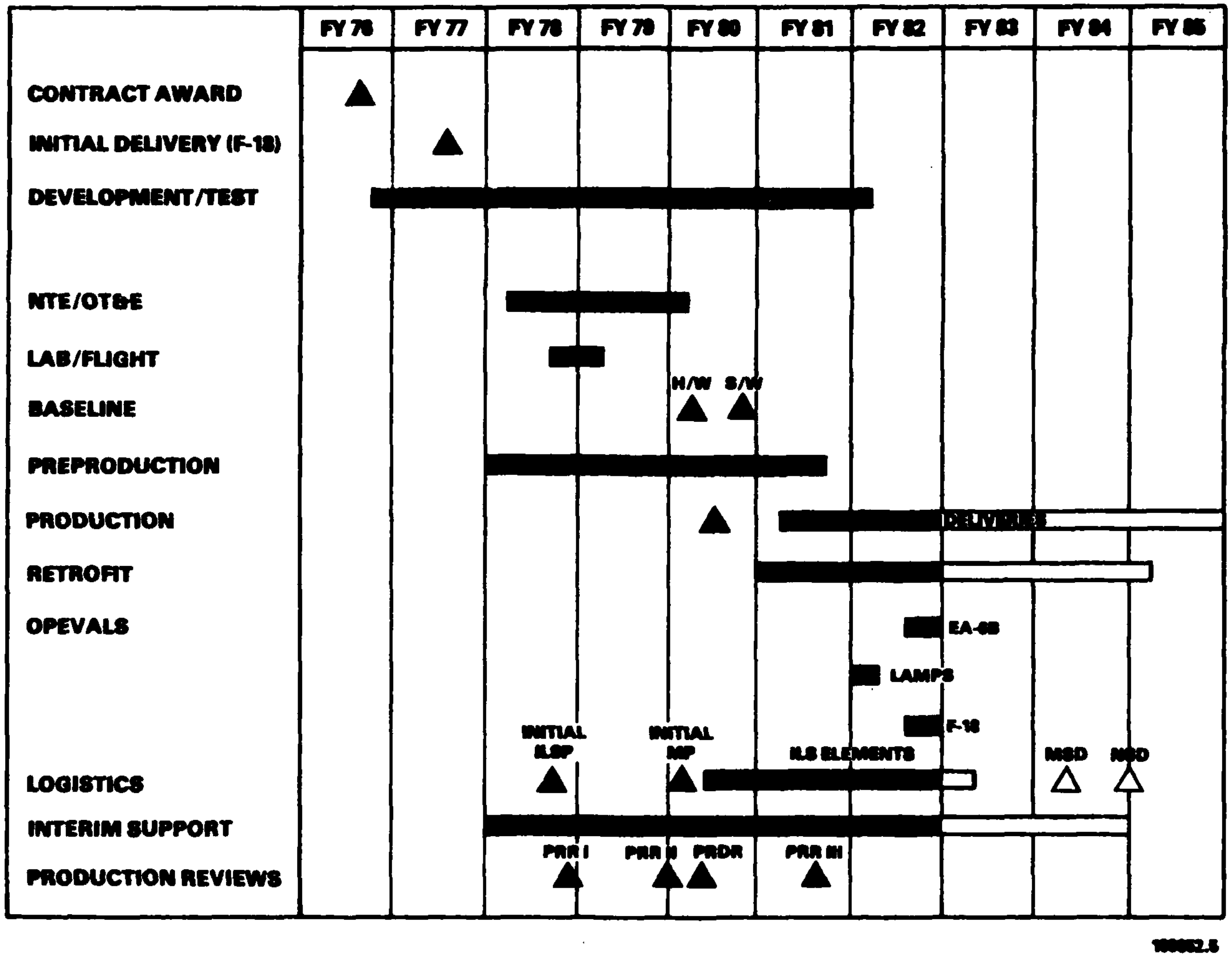
AN/AYK-14(V)

AN/AYK-14(V) PROGRAM MILESTONES

• CONTRACT AWARD	SEP 76
• PREPRODUCTION DELIVERY	JUL 77
• BASELINE	DEC 79
• PRODUCTION OPTION	MAY 80
• 60/MONTH CAPACITY	APR 84
• 575 SYSTEMS DELIVERED TO DATE	

211056.0

AN/AYK-14(V)
PROGRAM MILESTONES



AN/AYK-14(V)

AN/AYK-14(V) PRODUCT IMPROVEMENT

- DESIGN PERMITS ORDERLY PRODUCT IMPROVEMENT WHILE RETAINING DESIGN AND LOGISTICS INVESTMENT
  - BUS ARCHITECTURE
  - FUNCTIONAL PARTITIONING
- IMPROVEMENTS TO DATE
  - INCREASED MEMORY DENSITY (16K-64K MODULE)
  - NEW INPUT/OUTPUT (6 MHZ, 1553B)
  - ENHANCED INPUT/OUTPUT PROCESSOR

211067.0

AN/AYK-14(V)

AN/AYK-14(V) FUTURE PLANNING

- INCREASE MEMORY DENSITY
- INCREASE PROCESSOR THRUPUT
- INCREASE INPUT/OUTPUT CAPABILITY
- ACCOMMODATE ADA
- ACCOMMODATE TECHNOLOGY ADVANCES

211058.0

Biography

Henry H. Mendenhall
Computer Resources and Avionics Systems Division
Naval Air Systems Command
Washington, D. C.

BSEE Drexel University, Philadelphia 1968

EXPERIENCE

14 years with the Naval Air Systems Command in Avionics computer and software systems.

CURRENTLY

Section Head for Advanced Computer Systems

Responsible for planning, engineering and acquisition of Navy standard and planned standard computer resources.

ADVANCED SYSTEMS ARCHITECTURE

SESSION CHAIRMAN: Major I. Caro
AFWAL/AAAF

MODERATOR: Colonel David J. Teal
F-16 Deputy System Program Director

B-1B AVIONICS APPLICATIONS OF MILITARY STANDARDS

L. M. Carrier
and
G. A. Kinstler

Rockwell International, North American Aircraft Operations - Avionics
2770 E. Carson St.
Lakewood, CA 90712
(213) 420-0290

Military standards applied to the B-1B avionics program are discussed. Emphasis is placed on aircraft electronics systems design and interface with the aircraft. Subsystems discussed include the Central Integrated Test System (CITS), Electrical Multiplex (EMUX) system, control and displays, weapons interfaces, and Communications And Traffic Control (C&TC). Standards applied to offensive and defensive avionics are also summarized. Program constraints and rationale pertinent to the partial or deferred application of some standards are discussed. The extent currently applied and options planned for future incorporation in these areas (e.g., MIL-STD-1589B, -1750A, and -1760) are presented.

INTRODUCTION

The application of military or government standards to the architecture and subsystem design of the B-1B is advantageous to minimize the total cost of ownership. Cost savings accrue from such factors as:

1. Reduced acquisition cost due to standards encouraging the use of previously developed techniques, components, subsystems, or software applicable to other military programs.

2. Reduced support costs resulting from training transferability and commonality of maintenance actions and logistics support from other programs.

Rockwell has supported the application of current standards to the B-1B, consistent with the B-1B development schedule. In some cases, standards have been partially incorporated or other special considerations have been made to allow future application of developing standards. Equipment elements selected for the total B-1B avionics suite were, where possible, the USAF standard or preferred item, or a current inventory item common to other programs such as the B-52 Offensive Avionics System (OAS).

This paper presents an overview of the B-1B avionics suite and identifies the standards applied, partially incorporated, or provisions made for future standards incorporation. The discussion is organized into the following six avionics areas:

1. Communications and Traffic Control (C&TC)
2. Central Integrated Test System (CITS)
3. Electrical Multiplex (EMUX) system
4. Flight Instruments Subsystem (FIS)
5. Offensive Subsystem Group (OSG)
6. Defensive Subsystem Group (DSG)

Though not further discussed separately within the following subsections, the B-1B program is currently in the acquisition process for automatic test equipment supporting all of the above equipment areas at the intermediate depot and shop level. It is intended to employ the new Modular Automatic Test Equipment (MATE) standard in this area to the maximum extent possible within B-1B program constraints.

COMMUNICATIONS AND TRAFFIC CONTROL SYSTEM

Figure 1 presents a block diagram of the Communications and Traffic Control (C&TC) system. Dual AN/ARC-171 UHF radios provide redundant line-of-sight communications capability. The TSEC/KY-58 (USAF standard) secure voice equipment provides encrypted voice transmissions over UHF. The AN/ASC-19 Satellite Communications (SATCOM) terminal permits world wide communications capability for voice or teletype messages. The new USAF standard TACAN, the AN/ARN-118, provides navigational range and bearing information. High frequency (HF) communications provide two-way voice command and control capability for the B-1B. The AN/ARC-190 HF radio is a new production equipment. The AN/ARC-190 HF coupler is a CFE item, modified from existing hardware to match the B-1B shunt-type antenna system. The AN/ARN-108 Instrument Landing System (ILS) is the same as used on the B-1A and provides localizer, glideslope, and marker beacon functions. The AN/APX-101 Identification Friend or Foe (IFF) transponder used on the B-1B is the current USAF standard and, in conjunction with the KIT-1A/TSEC secure IFF, provides a comprehensive identification capability. The AN/APX-105 rendezvous beacon used for the B-1B is an X-band transponder. It is the same as used on the B-1A, except for a connector change made to comply with current B-1B connector requirements. It is being provided as CFE on the B-1B. The ICS-150 intercom system is also being provided as CFE. This system does not yet have a military nomenclature assignment since it is a new development for the B-1B. It is a lower cost form, fit, and function replacement for the intercom system used on the B-1A. Low Frequency/Very Low Frequency (LF/VLF) receive-only teletype message terminals are expected to become available for the B-1B during the later years of production. This is expected to provide a very long range jam-resistant command and control line to the B-1B.

CENTRAL INTEGRATED TEST SYSTEM

The Central Integrated Test System (CITS) is an on-board B-1B aircraft subsystem interfacing with but completely independent of aircraft avionics and non-avionic operational systems. It provides real time functional testing and in conjunction with the B-1B offensive and defensive systems provides functional testing of all subsystems aboard the B-1B aircraft. To accomplish this, the CITS utilizes techniques and procedures proven on the B-1A aircraft which maximize passive monitoring of end-to-end subsystem testing during normal operation of the subsystem and minimize active interference during ground testing of subsystems without the aid of ground support equipment.

The CITS is essentially a digital data processing and control system with elements dispersed throughout the air vehicle for acquisition and conditioning of parameters from electrical, mechanical, hydraulic, etc. systems. A block diagram of the CITS is shown of Figure 2. The CITS system consists of a digital computer and a resident stored software program to control processing, four Data Acquisition Units (DAU) for interfacing with aircraft systems to transmit/receive test signal data, a CITS Control and Display (CCD) panel for operator interface, an Airborne Printer (AP) to provide a "hard copy" of test result data, and CITS Maintenance Recorder (CMR) which provides a magnetic tape output for ground data processing and analysis.

The CITS testing functions are essentially automatic, with a minimum of operator participation. All test logic is predetermined and fixed within the software program, thus eliminating the necessity for the operator to interpret results or make decisions as a part of the normal testing process. In flight, CITS continuously monitors parameters from the systems under test and utilizes the signal data in performing over 4000 tests each second. Malfunctions of systems under test are displayed to the flight crew with isolation of failures being made to the Line Replaceable Unit (LRU) level. This permits an immediate evaluation of the situation and allows the pilot to make mission-oriented decisions. If necessary, further data may be manually accessed by the flight crew using CITS to individually select and display the value/state of over 10,000 signals on the B-1.

All malfunction data displayed by CITS (identification of the failed system, identification of failed LRU, time of failure, and associated messages) are printed on a paper strip tape which provides maintenance personnel with an immediate view not only of trouble areas requiring maintenance action, but also of most probable corrective action required. This information, substantiated and supplemented by flight crew "squawks", allows maintenance operations to begin unhampered by the normal procedures of investigation of flight crew reports, hook-up of ground support test equipment, conducting of tests, and interpretation of test results. These time consuming tasks delay corrective action, after which the same tasks must be repeated to verify that the original problem was corrected. The success of such procedures is highly dependent upon the skill level and special knowledge of the technicians performing the maintenance, the availability of properly trained personnel, and the availability of a wide variety of special test equipment. Each of these negative elements is affected positively through the use of CITS. Special knowledge of a particular system undergoing test of maintenance by a technician is reduced without detracting from confidence in results, which in turn means that lower skill-level personnel can be assigned to maintenance tasks; elimination of the need for special test equipment does away with not only the time required to position and connect the equipment, but also reduces the number of people needed to handle and maintain the equipment. This is of paramount importance in an operational environment where austere bases demand that aircraft be self-sufficient. The CITS computational system employs the current military standard multiplex data bus, MIL-STD-1553B, to interconnect all elements of the system, except for the CITS Maintenance Recorder (CMR) which employs a B-52 OAS version of MIL-STD-1553A.

The CITS central processor is a high speed, general purpose, dual architecture machine derived from the merger of the B-52 OAS Instruction Set Architecture (ISA) and the MIL-STD-1750A ISA. It is common with the processors used for offensive and defensive systems. The implementation of MIL-STD-1750A ISA in this processor is presently scheduled for SEAFAC verification during 1983. The processor in conjunction with special test equipment can execute either an advanced version of the B-52 OAS instruction set or the MIL-STD-1750A instruction set. The common processor is currently being configured to execute coding using the improved B-52 OAS instruction set architecture. Common processor conversion to the MIL-STD-1750A ISA can be accomplished by minor firmware changes at such time that the software is ready. The CITS software is currently 75% programmed in the J3B higher order language (the remainder coded in assembly language). It is anticipated that conversion to the MIL-STD-1589B (J73) higher order language will occur concurrently with the conversion to the MIL-STD-1750A ISA discussed above.

ELECTRICAL MULTIPLEX SYSTEM

The Electrical Multiplex (EMUX) system is a digital time division multiplex system which transmits control and data signals over redundant data buses. All B-1B aircraft electrical control signals are multiplexed where practical. The EMUX system reduces the point-to-point "hard" wiring, conventional signal wiring and the associated hardware required, permitting a savings in both weight and installation costs. The use of EMUX provides additional advantages of (1) improved reliability, (2) more flexibility, (3) greater maintainability, and (4) reduced battle damage vulnerability for the aircraft electrical systems.

The EMUX system provides the function of accepting, formatting, transferring, performing logic and control functions, and outputting signals required for aircraft subsystem electrical control, data transfer, and function monitoring. The EMUX system serially transmits the signals over redundant circuits (data buses) by using Time-Division Multiplex (TDM) techniques and baseband transmission. The EMUX system consists of the following equipment:

two Control (CONT) boxes (bus controllers), ten digital Data Distribution (DD) boxes, and one CITS Interface (CI) box. Their placement in the aircraft is illustrated in Figure 3.

A redundant data bus interconnects all EMUX boxes which are distributed throughout the major equipment areas of the aircraft. Data transfer is controlled by the CONT boxes which have logic processing capabilities sufficient to perform sequencing, interlock and other control and load management functions involving the discrete signals associated with the aircraft subsystems electrical power distribution and control. The EMUX data sequencing, logic processing, and load management functions are capable of being changed, providing the required control flexibility.

The EMUX system utilizes high density microelectronics and solid-state parts, components, and circuitry to achieve small size, low power utilization and high reliability. The EMUX is hardened to withstand nuclear radiation and Electromagnetic Pulse (EMP) environments. Component and circuitry redundancy provide that no single failure within an EMUX section will affect the data transfer and processing of that section. Each EMUX box has self-test functions to determine failures and to switch over to redundant circuits or, if appropriate, to the redundant CONT Box. The EMUX self-test data is transferred to CITS via the CI Box. Each CONT, DD, and CI box has provisions to prevent box damage due to overtemperature conditions during ground maintenance modes.

Rockwell recognized that significant development effort was required to implement MIL-STD-1553B in the EMUX system. While Rockwell supports the Air Force's initiatives for standardization, it was recognized that substantial schedule risk and cost impact would result. Since the EMUX system is an existing design and is a stand-alone-system, few of the benefits of standardization would be achieved.

The evaluation of incorporation of MIL-STD-1553B into the B-1B EMUX design required consideration of the following signficant factors:

1. Development cost
2. Production cost
3. On-aircraft maintenance cost
4. Off-aircraft maintenance cost
5. Training cost
6. System modification and upkeep cost

Development costs would increase significantly because EMUX with -1553B is more complex and development schedule requirements will not permit the incorporation of an EMUX system with -1553B into the Lot I and Lot II aircraft. A new EMUX system, incorporating MIL-STD-1553B could only be installed beginning with Lot III. This would necessitate two separate parallel development efforts. Production costs would increase because of the increased complexity of the EMUX units with -1553B incorporated and also the production learning curve would be set back due to the introduction of a new design at Lot III.

On-aircraft maintenance would not be significantly different with -1553B. In either case, EMUX reports its status to CITS which displays a maintenance message identifying any malfunction. Should an item of support equipment be identified as a requirement, the use of -1553B would have no impact.

With consideration given to the above factors, the benefits derived from modifying EMUX to comply with the MIL-STD were minimal since:

1. The EMUX is an existing design
2. The design is proven and extensively flight tested
3. The EMUX bus is a specialized closed bus

Additionally, compliance would:

1. Cost additional money
2. Increase hardware and software complexity
3. Potentially reduce reliability
4. Have little effect on life cycle cost

FLIGHT INSTRUMENTS SUBSYSTEM

The B-1B Flight Instruments Subsystem (FIS) includes sensors and transducers, and associated display components and electronic assemblies related to air-data,attitude direction measurement systems as well as specialized subsystem electronic interfacing equipment. The interfaces and locations of subsystem elements are shown in Figures 4 and 5, as further amplified in the following discussion.

Redundancy is employed so that loss of any single element will not affect mission completion capability and no two component failures will result in a hazard Category III or IV as defined in MIL-STD-882. The entire system is designed with emphasis on simplicity of mechanisms, with minimum dependence on support equipments, and it will meet the requirements of AFCS DH 2-1 and 2-2. Self test provisions require that any undetectable failure, when followed by a single in-flight failure, will not result in an unsafe condition.

Principle interfacing systems are the Central Integrated Test System (CITS), Communication and Traffic Control (C&TC) system and the Offensive Subsystem Group (OSG). Communication between FIS components and the OSG is accomplished via a serial digital data bus conforming to the requirements specified in MIL-STD-1553B. Communication between aircraft subsystem components is also accomplished by dedicated serial digital, analog, and DC discrete signals.

The FIS includes a fully redundant air data system including four side mounted probes, installed to comply with requirements of MIL-P-26292, to sense air pressure and flow angle, and two Total Temp Probes to sense temperature.

Two Central Air Data Computers (CADC's) utilize the sensors' data to provide highly accurate information including velocity, altitude and angle of attack for primary flight instrument displays and twelve other aircraft systems. In addition, mechanical stand-by instruments with direct pressure connections from the air data probes provide vital altitude and velocity information to the pilots.

The Gyro Stabilization Subsystem (GSS) is an all-attitude Auxiliary Heading Reference System (AHRS) which provides pitch, roll, heading, and turn-rate to the pilot's and copilot's Vertical Situation Displays (VSD's); in addition, heading is provided to the 1553B data bus. For maximum accuracy AHRS requires True Airspeed (TAS) indications from the CADC's. AHRS consists of a Gyro Reference Unit (GRU), a Gyro Reference Unit-Mounting Base, Electronic Control Amplifier (ECA), a Magnetic Aximuth Dectector (MAD) and a Control Panel (CP). As a back-up for the AHRS pitch, roll, and turn-rate, a Self-contained Attitude Indicator (SAI) and a rate gyro is provided. As a back-up for AHRS heading, a 'whiskey' magnetic compass is provided.

The Flight and Control Instruments (FCI) is a dual (redundant) system with each individual system consisting of Flight Director Computer/Monitor Unit (FDC/M) and a VSD system. The FDC/M selects attitude reference from either the GSS or the Inertial Navigation System (INS) and computes steering commands in various navigation and traffic control modes selected on the FDC/M panel. Output signals are displaysed on the VSD and the Horizontal Situation Indicator (HSI), and used by the Automatic Flight Control System (AFCS) for control computations. The VSD is a CRT display, providing pilot Attitude Director Information (ADI) and flight parameter symbology. It also provides video overlays of Terrain Following (TF) or Terrain Avoidance (TA) modes when selected. The Surface Position Monitor System (SPMS) includes the Surface Position Indicator (SPI), surface position sensors, transducer excitation power supply and signal conditioning unit.

The Data Transfer System (DTS) includes dual (redundant) Flight Instruments Signal Converters (FISC) and a single Data Conversion Unit (DCU) which provide data conversion and processing needed for a compatible interface between various aircraft equipments. In addition, the DTS includes a Multiplex Interface Module (MIM) that is capable of receiving/transmitting asynchronous serial digital data on either of two data buses as specified in MIL-STD-1553B; the MIM is installed inside the using subsystem LRU, and a total of 10 MIM's will be used on the aircraft. Finally, the Data Link Terminal (DLT) is used to provide transformer coupling between the data bus and the subsystems as well as bus termination. The auxiliary equipment includes the Flight Instruments Test/Mode (FITM) panel, two clocks, two prepare to eject bells, an aural tone generator, a windshield temperature controller, and five GFE instruments.

OFFENSIVE SUBSYSTEM GROUP

The B-1B Offensive Subsystem Group (OSG) utilizes state-of-the-art off-the-shelf and modified system components to achieve unprecedented performance capabilities at minimum cost. The major elements of the OSG are shown in Figure 6.

Accurate navigation is provided by a Singer-Kearfott dual dry tuned-rotor gyro inertial navigation system derived and improved from the F-16. A Westinghouse Offensive Radar System (ORS) derived from the APG-66 radar (F-16) provides high-quality ground map imagery for navigation system updates as well as numerous other modes including terrain following. A second radar channel identical to the first performs backup functions. The terrain following function provides much improved accuracy, sensitivity, and countermeasure immunity relative to the B-1A avionics equipment. Both radar channels operate through a shared electronically scanned phased array antenna which permits simultaneous radar modes on a time shared aperture basis. The phased array antenna design plays a key role in the B-1B's acheiving a low observable signature. The doppler radar supports the navigation system and is unmodified off-the-shelf equipment (the current Air Force standard).

The computational system employs the current military standard multiplex data bus, MIL-STD-1553B, and AP-101F computers derived and slightly modified from the B-52 Offensive Avionics System (OAS) upgrade program. A total of eight such identical processors are utilized on the B-1B including four for the central processing of offensive and defensive systems (one redundant), two being utilized for the ORS terrain following computations, one as a defensive system pre-processor, and one for central integrated test system functions. More than 25% capacity for growth is provided in the central processing functions. Controls and displays are a combination from B-52 OAS and B-1 modified as necessary. The navigation system, computational system, and missile interface units have the accuracy, capacity, and interface compatibility required to accommodate ALCM carriage at a later date.

Weapon interface units include existing units from B-52 OAS and B-1 programs as well as new designs. B-1B weapons interface compliance with MIL-STD-1760 is summarized in Table 1. Buried provisions to accommodate MIL-STD-1760 requirements will be provided on B-1B aircrafts No. 2 and subs. Fiber optics and 270 VDC transmission line requirements, which are part of MIL-STD-1760, have been excluded since they lack B-1B applicability in the reasonably foreseeable future. Thus, the additional buried wiring required on the B-1B to accommodate MIL-STD-1760 will consist of audio, video, and RF lines from each of the three weapons bays to the central equipment bay. The wire run from each of the three weapons bays will consist of a twisted and shielded pair of wires to accommodate audio, a pair of 75 ohm coax lines to carry video, and a pair of 50 ohm coax lines for RF. Each of the three wire runs will be capped and stored in the central equipment bay as well as in the forward, intermediate, and aft weapons bay.

Additional discussion of the OSG and standards is presented by Boeing in a companion paper.

DEFENSIVE SYBSYSTEM GROUP

The B-1B Defensive Subsystem Group (DSG) consists of a modified version of the AN/ALQ-161 defensive avionics system developed for B-1A aircraft (A/C-4), Expendable Countermeasures (EXCM) developed for A/C-4, a Defense Management System (DMS) and a Tail Warning System (TWS). The major elements of the DSG are shown in Figure 7.

The AN/ALQ-161 defensive avionics equipment includes transmitters, receivers, antennas, a Preprocessor Avionics Control Unit (PACU) and interconnecting electronics. Utilizing this equipment, the AN/ALQ-161 acquires radiating signals from the external environment, processes these threat signals to determine their identity, and based upon the type of signal acquired, applies an appropriate electronic countermeasure (jamming) to defend against the detected threat. Improvements made to the A/C-4 system consist of frequency expansion to provide bands 1-3 receive and band 8 receive and transmit and the addition of techniques to counter sophisticated radars.

The Expendable Countermeasures (EXCM) consists of eight interchangeable flare or chaff dispensers located on top of the aircraft aft of the crew compartment, a controller and controls and displays. Dispensing of EXCM (chaff or flares) is controlled by the DMS. It can be manual (operator initiated) or automatic, based on missile warning data supplied by the TWS.

The Tail Warning System (TWS) detects aircraft and missiles in the aft sector and provides information to the DMS for operator warning and automatic dispensing of EXCM as applicable.

The Defensive Management System (DMS) consists of controls and displays and a preprocessor that provides the man-machine interface required to operate the AN/ALQ-161 defensive avionics, EXCM and TWS. Primary data transfer is accomplished via MIL-STD-1553B and data bus interfaces. The preprocessor and certain other components of the DMS are shared with the offensive avionics controls and displays.

SUMMARY

In summary, military standards have been utilized for B-1B subsystems to the maximum extent possible where shown to be cost effective and consistent with B-1B development schedules. Standardization in the C&TC subsystem is accomplished by equipment selection. For subsystems employing common processors (CITS, OSG, and DSG), the standard data bus is fully incorporated and provisions are being made for future incorporation of MIL-STD's -1589B and -1750A to realize software support cost savings in the future. Some elements of MIL-STD-1760 are being incorporated now for weapons interfaces, whereas other elements are deferred. The mature development status of the B-1B EMUX subsystem as well as B-1B schedule limitations indicated no program/field support cost savings could be afforded in that area by redesign to more current standards.

ACRONYMS

ADI	Attitude Director Indicator
AFCS	Automatic Flight Control System
AFCS DH	Automatic Flight Control System Design Handbook
AGE	Avionics Ground Equipment
AHRS	Attitude Heading Reference System
ALCM	Air Launched Cruise Missile
AP	Airborne Printer
CADC	Central Air Data Computer
CCD	CITS Control and Display
CFE	Contractor Furnished Equipment
CI	CITS Interface
CITS	Central Integrated Test System
CMR	Cits Maintenance Recorder
C&TC	Communications & Traffic Control
DAU	Data Acquisition Unit
DCU	Data Conversion Unit
DD	Data Distribution
DLT	Data Link Terminal
DMS	Defense Management System
DSG	Defensive Subsystem Group
DTS	Data Transfer System
ECA	Electronic Control Amplifier
EMP	Electromagnetic Pulse
EMUX	Electrical Multiplex
EXCM	Expendable Countermeasures
FCI	Flight and Control Instruments
FDC/M	Flight Director Computer Monitor
FIS	Flight Instruments Subsystem
FISC	Flight Instruments Signal Converter
FITM	Flight Instruments Test/Mode
GFE	Government Furnished Equipment
GRU	Gyro Reference Unit
GSS	Gyro Stabilization Subsystem
HF	High Frequency
HSI	Horizontal Situation Indicator
IFF	Identification Friend or Foe
ILS	Instrument Landing System
INS	Internal Navigation System
ISA	Instruction Set Architecture
LF/VLF	Low Frequency/Very Low Frequency
LRU	Line Replaceable Unit
MAD	Magnetic Azimuth Detector
MATE	Modular Automatic Test Equipment
MIM	Multiplex Interface Module

ACRONYMS (Continued)

OAS	Offensive Avionics System
ORS	Offensive Radar System
OSG	Offensive Subsystem Group
PACU	Preprocessor Avionics Control Unit
RF	Radio Frequency
SAI	Self-contained Attitude Indicator
SATCOM	Satellite Communications
SPI	Surface Positon Indicator
SPMS	Surface Position Monitor System
TA	Terrain Avoidance
TAS	True Airspeed
TDM	Time Division Multiplex
TF	Terrain Following
TWS	Tail Warning System
USAF	United States Air Force
VAC	Volts Alternating Current
VDC	Volts Direct Current
VSD	Vertical Situation Display

Table 1. B-1B MIL-STD-1760 Compliance

	AVAILABLE	BURIED PROVISIONS	DEFERRED
28 VDC	X		
115 VAC	X		
270 VDC			X
DIGITAL DATA	X		
AUDIO		X	
VIDEO		X	
RF		X	
FIBER OPTICS			X

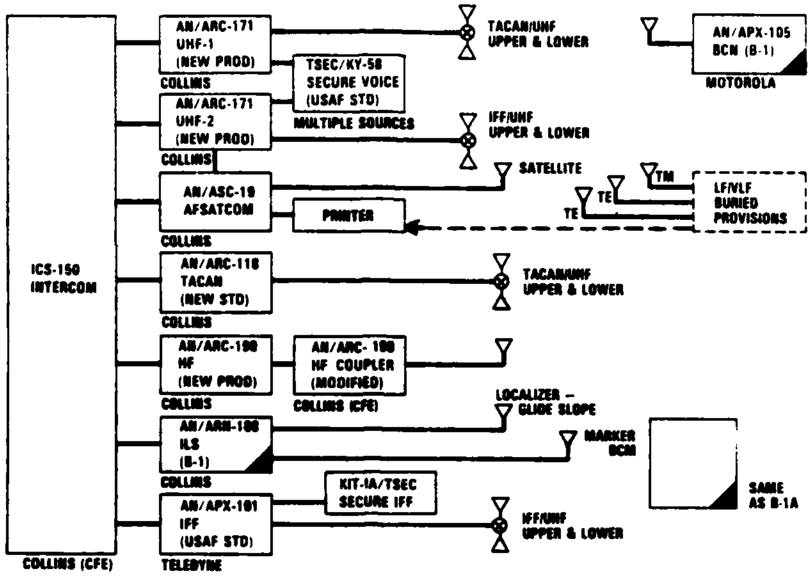


Figure 1. B-1B Communications and Traffic Control (C&TC) System

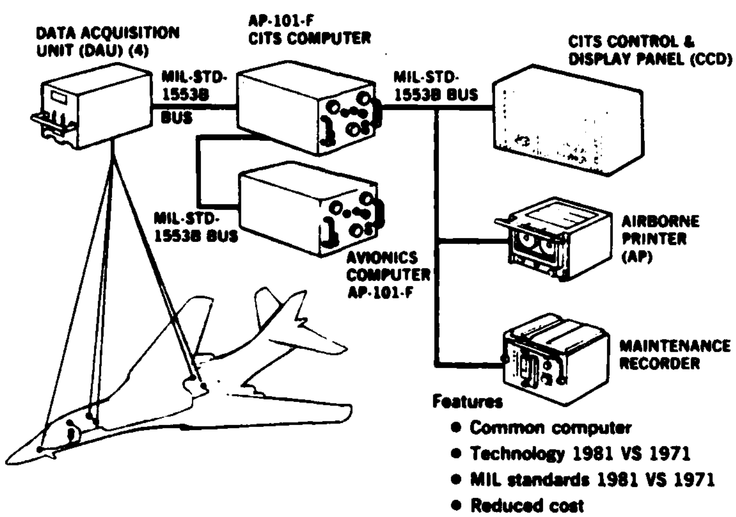


Figure 2. B-1B Central Integrated Test System (CITS)

Control Box = Bus Controller
DD Box = Remote Terminal (Interface with Subsystems)
CI Box = Interface with CITS

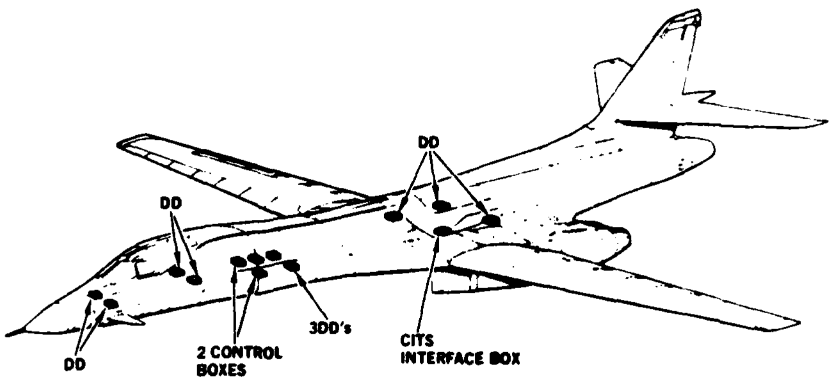


Figure 3. B-1B Electrical Mutiplex (EMUX) System Component Location

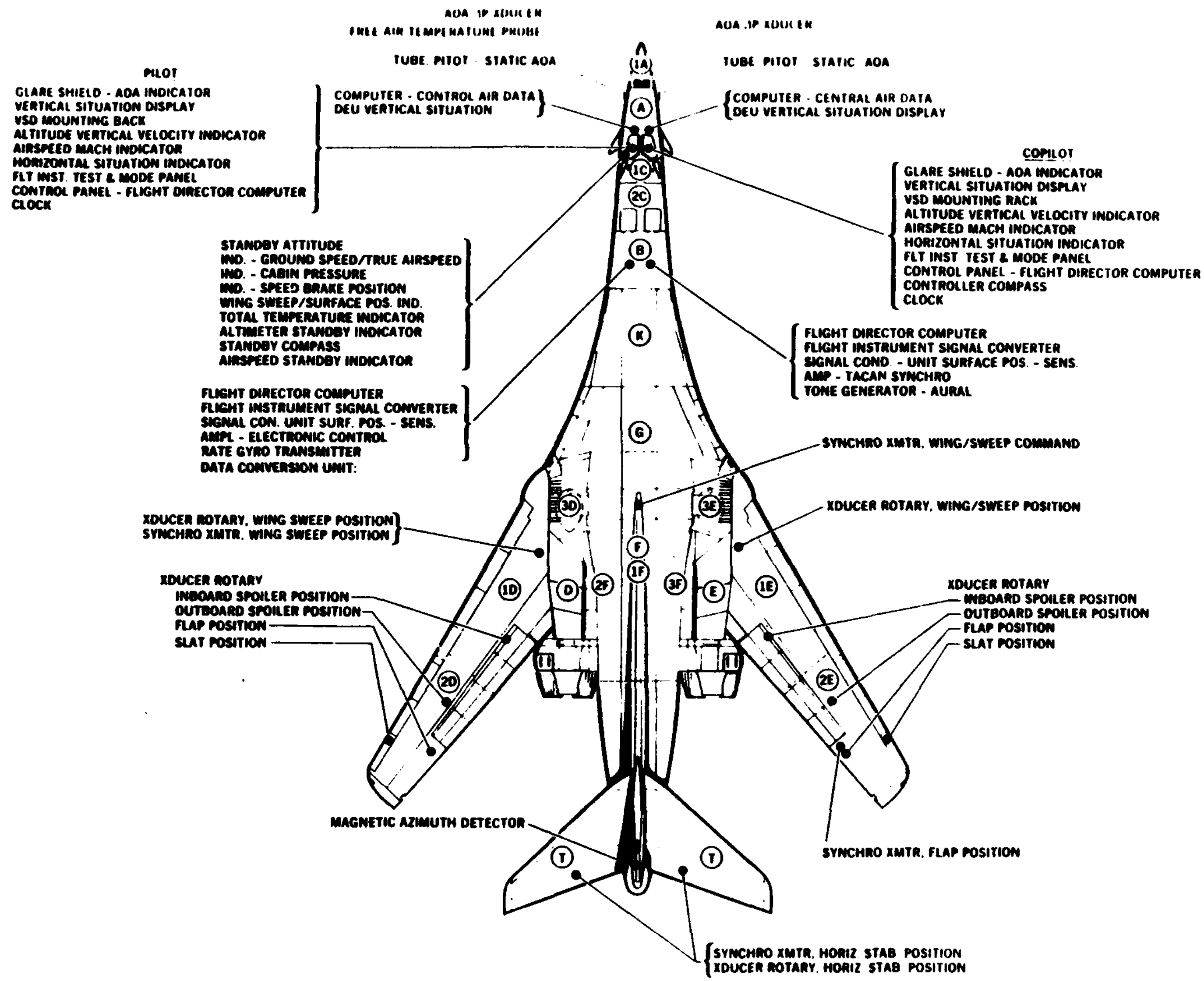


Figure 4. Flight Instruments Subsystems Configuration

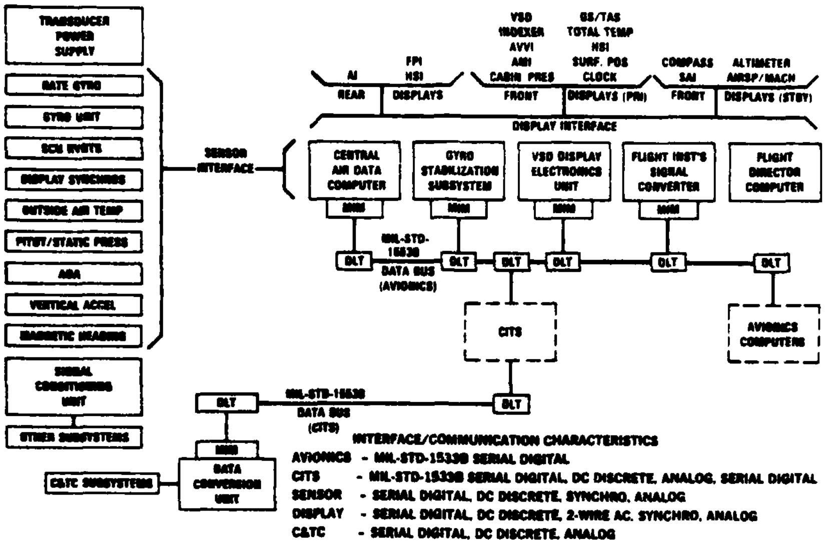


Figure 5. Flight Instruments Subsystem Interfaces

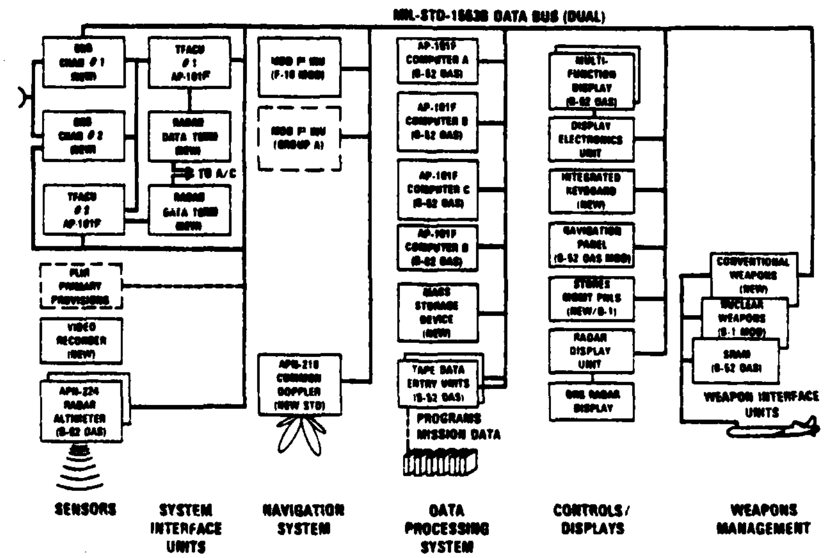


Figure 6. B-1B Offensive Subsystem Group (OSG)

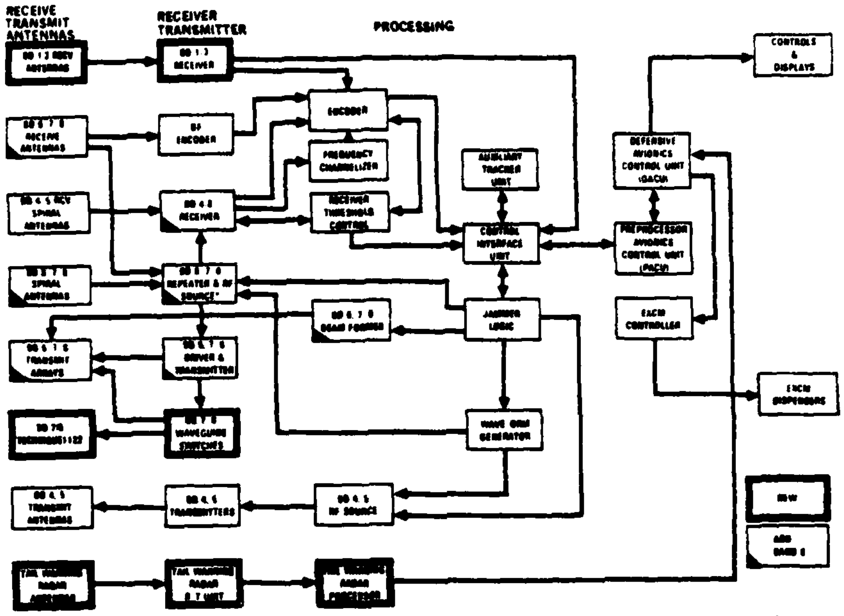


Figure 7. B-1B Defensive Subsystem Group (DSG)

STANDARDS APPLICATION TO B-1B AVIONICS PROGRAM

H. L. Ernst

Boeing Military Airplane Company (BMAC)
Mail Stop 41-14
P.O. Box 3707
Seattle, Washington 98124
(206) 655-1130

This paper covers the B-1B Avionics Program as related to the recently developed USAF Avionics Standards. These USAF Avionics Standards include MIL-STD-1553B Data Bus System, MIL-STD-1589B High Order Language (HOL), and MIL-STD-1750A Computer Instruction Set Architecture (ISA). The B-1B Avionics System is described covering the system architecture, major subsystem, and equipment. The recently conducted B-1B Standards Program (MIL-STD-1589B and MIL-STD-1750A) is explained. Also the tasks are defined and the summary of results and conclusions are included. The B-1B program action, taken subsequent to the Standards Program wrap-up, is discussed. Finally, a B-1B program plan for application of the military avionics standards is discussed.

BASELINE B-1B AVIONICS Program

The B-1 Avionics system was originally contracted in April 1972, well before the USAF Avionics Standards were defined. The B-1 Program was subsequently redirected from a production program to a test and development activity in the latter 1970's. The B-1B Avionics Program RFP was released in January 1981 and the final contract signing was completed in May and June 1982. The B-1B Avionics Program as contracted in 1981-82 was defined to be a low risk program (i.e., few developments). At the time the J73 Compiler was not proven (matured on a production application), and the MIL-STD-1750A ISA had been only recently released. These factors resulted in definite risk for standards application to a low risk program with tight budget and schedule and a fixed price contract. Therefore, the B-1B baseline definition included the IBM AP-101C Avionics Control Unit (ACU) and JOVIAL J3B HOL transfer from the B-52 Offensive Avionics System (OAS). With further B-1B program activity the IBM AP-101C ACU evolved to the AP-101D model.

The B-1B Avionics Configuration includes (1) navigation subsystem, (2) computational subsystem, (3) control and display subsystem, (4) stores management subsystem, and (5) various features of the aft crew station control console along with some features on the Front Station (pilots') arrangement (Figures 1, 2 and 3). The AP-101D ACU was designated as the common B-1B processor and as such is used in eight places on each B-1B airplane. These applications are listed and defined in table 1. The CRACU provides backup (redundancy) for the critical functions assigned to the

PREVIOUS PAGE IS BLANK

GNACU, WDACU, and the CDACU. The dual TFACUs provide redundancy for the terrain following mode. The electronic countermeasures (ECM) has backup capability provided in event of failure of the PACU. The AP-101D ACU included an IBM MMP (Multipurpose Midline Processor) ISA, 400 KOPS thruput, 128K Words of memory, four dual channel 1553 interfaces, and parallel input/output (I/O) for the ECM system interface.

PARALLEL STANDARDS PROGRAM

To pursue the possibility of applying MIL-STD-1589B (JOVIAL J73 HOL) and MIL-STD-1750A ISA Processors to the B-1B program without impacting schedule or costs, a separately funded parallel standards program was established in January 1982. This program was initiated with a 90-day phase 0 study period, January 11 through April 12, and a planned follow-on phase I. To ensure minimal B-1B program impact, this program was conducted by ASD/EN at WPAFB and by BMAC Avionics Technology in Seattle, Washington. Rockwell International (RI), the Weapon System Contractor (WSC), and AIL Division of the Eaton Corporation, the Electronic Countermeasures Contractor (ECMC), also participated in the study.

The Phase 0 Program objectives were to initiate studies, identify alternative incorporation approaches, develop plans, and initiate long lead activities to:

- o Replace B-1B ACU, AP-101D, with a MIL-STD-1750A processor.
- o Replace J3B HOL with J73 HOL.
- o Develop and validate J73/1750A compiler.
- o Rehost BMAC support software for compatibility with J73/1750A standards.
- o Establish viable B-1B standards incorporation plans within current program constraints.

The Phase 0 SOW tasks are listed in table 2, and the key results are outlined in figure 4. Data available by mid-May 1982 indicated that (1) a decision to incorporate the standards at any time during 1983 or 1984 would result in a schedule slide and high incorporation costs and (2) an August - September 1982 decision date would have less impact than a November - December 1982 decision date because of the planned start date of B-1B software coding by both BMAC and RI. In all cases, the ECMC (AIL) did not plan to make a 1982 or 1983 standards incorporation decision or to comply with incorporating the standards into B-1 aircraft 4 or B-1B aircraft 1 first flight (figures 5 and 6). The phase 0 program was completed and activity continued under contract into the phase I period through 7 June 1982.

Summary of results and conclusions reached during the Parallel Standards Program (phases 0 and I) by task are listed below.

a. Task 1, B-1B ACU modification kit to MIL-STD-1750A ISA:
Results of this task indicated that the current IBM AP-101D could not provide sufficient thruput margin beyond the original 400-KOPS to allow for the expected thruput efficiency degradation of a new compiler (J-73), as compared to a mature compiler (J3B). As a result, this task was terminated in mid-March and not carried on into the phase I program.

b. Task 2, Preparation of competitive procurement (MIL-STD-1750A ISA processor);

A B-1B MIL-STD-1750A (ACU) competition within industry was completed, source selection made, and the results were briefed to the B-1B SPO, along with various ASD personnel. The WSC (Rockwell International, ECMC (AIL), and the USAF(ASD/EN) participated in the competitive procurement package preparation to ensure concurrence on a common B-1B MIL-STD-1750A ACU definition. The competitive procurement package was released to seven suppliers that were selected from an earlier listing of 13. Six suppliers submitted proposals in early May 1982. The proposal evaluation (technical, operations, Program Management, schedule, cost, etc.) and source selection were completed by the end of May 1982. The selected MIL-STD-1750A ACU, IBM AP-101E, was off-the-shelf, developed by IBM, exceeded the specified requirement of 525 KOPS thruput, and complied with the requirements of 256K words of memory capability. The B-1B Program office then chose the IBM AP-101F dual-architecture ACU rather than the AP-101E selected by the standards program competition. This choice was made because of program factors that indicated a lower B-1B program cost and schedule risk while still providing for standards incorporation. These program factors are further explained in the next section of this paper, "Current B-1B Avionics Program."

C. Task 3, Modification of the AFWAL (SEA) J73/1750A compiler, and Task 4, Evaluation of alternate J73/1750A compiler approaches:

Subsequent to definition of B-1B J73/1750A compiler requirements, industry review, and identification of viable alternate compilers phase 0 contracts were awarded as listed below.

TASK	SUPPLIER	BASELINE COMPILER	REMARKS
3	Software Engineering Associates (SEA)	AFWAL (SEA)	Developed AFWAL compiler
3	Proprietary Software Systems (PSS)	AFWAL (SEA)	Under contract to GD/F-16
4	Advanced Computer Techniques (ACT)	F-16 MSIP compiler development	Under contract to GD/F-16
4	SofTech (Software Technology)	JOCIT	Previous RADC, current MX, and other related contracts

As a result of the evaluation of Phase 0 activity, supplier Phase I design approach reports, and supplier Phase I proposals with associated not-to-exceed costs, Phase I development contracts were awarded to PSS on the AFWAL (SEA) Compiler and SofTech on the JOCIT Compiler. The Phase I contracts required initial compiler deliveries from each supplier on 15 July 1982 and final compiler deliveries on 15 September 1982. These dates were selected to be compatible with the planned alternative program decision dates of 1 September and 1 December 1982 for B-1B Program incorporation of the MIL-STDs.

d. Task 5, Retarget BMAC support software to MIL-STD-1750A:

All critical areas of the BMAC support software rehost to the MIL-STD-1750A were completed on schedule to ensure compatibility with other elements of the B-1B Parallel Standards Program. The activities under this task were (1) definition of change requirements, (2) assembler updates, (3) link Editor updates, and (4) simulator update.

e. Task 6, Development of benchmark (J73/1750A compiler evaluation system):

The objectives of this task were to develop a benchmark plan to assess capabilities of the J73/1750A compiler using operational software, conduct a review of the AFWAL (SEA) compiler status, and develop a detailed compiler evaluation plan to measure the J73/1750A compiler acceptability. Results of the review of the AFWAL (SEA) Compiler based on limited benchmark code from RI, AIL, Logicon, and SofTech were that the J73 compiler used 16% more memory and had a 54% decrease in compilation time than the J3B compiler. It was concluded that the AFWAL (SEA) compiler did provide a basis for development of a mature compiler for B-1B application. This task was phased to support the planned alternative program decision dates of 1 September and 1 December 1982 for B-1B Program incorporation of the MIL-STDs.

f. Task 7, Development of common J73/1750A implementation:

The objective of this task was to develop the J73/1750A implementation plan for Phase I. Activities undertaken included (1) definition of facilities and training, (2) impact of the conversion upon the executive, (3) impact of the conversion on applications software, and (4) definition of a master implementation schedule. The Phase I plan developed, figure 5, is keyed to Program Incorporation decisions in August 1982 and November 1982 to support effective dates of 1 September and 1 December 1982.

g. Task 8, Definition of alternative program plans:

The objectives of this task were to define plans for implementing Avionics standards, MIL-STD-1589B and MIL-STD-1750A, in the B-1B in coordination with the WSC (RI) and ECMC (AIL). Risk items and impact on costs and schedules were identified and engineering trade studies were performed to select a plan for standards implementation. Initial plans included four decision dates to meet B-1B first flight, ranging from April 1982 to December 1983. Also, an alternative plan was included for delayed incorporation of the standards, figure 6.

CURRENT B-1B AVIONICS PROGRAM

The MIL-STD-1750A industry competition was completed, including source selection of the IBM AP-101E computer by the parallel program (BMAC Technology). Studies and developments relative to the other program tasks (compiler, support software and flight and laboratory operational software) had indicated that significant program penalty (cost and schedule) would [illegible] experienced with a B-1B program changeover to the new standards. Also, [illegible] was indicated that future activities of the B-1B avionics and standards would require B-1B program office funding.

Concurrently (late May and early June 1982) IBM had submitted a supplier change proposal to the B-1B program covering a dual-architecture computer (AP-101F), providing for the MIL-STD-1750A ISA and the AP-101D MMP architecture. This proposal included provisions for validating both architectures during the development period including tests by the USAF SEAFAC Laboratory of the MIL-STD-1750A ISA and providing the "user option" of either architecture initially with a minimal impact changeover to the other architecture at a later time (figures 7 and 8 and table No. 3).

Therefore, two alternative approaches were available for the incorporation of the standards into the B-1B program. One approach was to use the IBM AP-101E computer, selected by the industry competion during the parallel study program, and the J73/1750A compiler, with an immediate changeover from the J3B HOL to the J73 HOL and from the AP-101D MMP assembly language (AL) to the MIL-STD-1750A ISA AL. The second approach was to use the IBM AP-101F dual-architecture computer with the MMP architecture as an interim step along with J3B HOL and a subsequent change-over to the Avionics standards with the AP-101F using the MIL-STD-1750A ISA. The B-1B program office selected the second approach thereby minimizing B-1B program cost and schedule risk. Further, the B-1B SPO concurred with the follow-on J73/1750A AFWAL (SEA) compiler development by PSS and that a continual monitoring of other J73/1750A compiler developments such as the ACT/GD F-16 compiler be conducted. The USAF directed that the follow-on development of the SofTech JOCIT compiler be terminated.

This course of action allows for program incorporation of the standards when (1) a production level J73/1750A compiler has been demonstrated, (2) the B-1B flight software development is stabilized, and (3) the B-1B program schedule and cost impacts are optimum for the change.

B-1B AVIONICS STANDARDS INCORPORATION

The Phase 0 and Phase I Parallel Standards Program and the current B-1B Avionics Program use of the IBM AP-101F dual-architecture ACU allow later incorporation of the standards in an orderly manner. A plan has been prepared and discussed with the B-1B SPO and the associate contractors (RI and AIL), figure 9. Specific actions and approximate schedules for incorporation of the USAF Avionics Standards into the B-1B Avionics are as listed below.

- ECP authorization, January 1985.
- Compiler and support software enhancements, January through December 1985.
- MIL-STD-1750A executive development and J3B to J73 translator completion (version 2), January through October 1985.
- B-1B test tape conversion, January through December 1985.
- SDL upgrade to accommodate MIL-STD-1750A ACU, August to October 1985.
- Avionics Flight Software (AFS) translation from J3B HOL to J73 HOL, translation from AP-101F MMP assembly language to AP-101F MIL-STD-1750A assembly language, and validation of translated software in the BMAC B-1B SDL, August 1985 through June 1986.

- AP-101 FIELD MOD to incorporate user proms for utilization of existing MIL-STD-1750A architecture, January through August 1985.
- Aircraft changeover from J3B2 to J73 software, June 1986 effective with B-1B aircraft 9 first flight.
- Flight validation of B-1B Avionics, ACU, and J73/1750A software, B-1B aircraft 9, June 1986 first flight.
- Also, companion ECPs would be required from the other associates for changeover of HOL and Assembly Language software.

SUMMARY

In summary, the B-1B program has incorporated the IBM AP-101F dual-architecture computer that includes MIL-STD-1750A ISA, while presently continuing with the current software approach using J3B HOL. This selection provides improved B-1B ACU reliability, memory capability and thruput over the AP-101D (table 3). Also, the B-1B program is continuing development and evaluation of the J73/1750A compiler to ensure meeting the objectives of the Parallel Program to incorporate the avionics standards (MIL-STD-1589B and MIL-STD-1750A) at minimal program risk. Efforts are underway to identify the incorporation approach that optimizes B-1B program schedule and cost impacts.

Mr. Ernst is currently chief, B-1B Avionics Technical Staff. He was program manager of the BMAC B-1B Avionics Parallel Standards Program (MIL-STD-1589B, J73 HOL, and MIL-STD-1750A ISA) from January through June 1982. After receiving his BSEE degree from the University of Kentucky, and graduate work at Notre Dame, Mr. Ernst joined The Boeing Company in 1953. He joined the Boeing Military Airplane Company in 1972, after 19 years in jet transport programs (KC-135, 707, 720B, 727, SST, and new airplane program NAP).

Subsequent to joining BMAC in 1972, Mr. Ernst was systems technology chief on the AMST prototype (YC-14) program covering the technical areas of flight deck, avionics systems, electrical systems and equipment, hydraulics, and mechanical systems and equipment. Then on the AMST (C-14) and C-X studies and proposals he was avionics technology chief, covering the areas of flight deck, avionics systems, avionics equipment and associated installations, such as equipment installations, cabling and antennas. Also, human factors including the cargo compartment as well as the flight deck were covered.

ACRONYMS/ABBREVIATIONS

ACT	Advanced Computer Techniques
ACU	Avionics Control Unit
AFWAL	Air Force Wright Aeronautical Laboratories
AIL	AIL Division of Eaton Corporation
ASIC	Avionics System Interface Contractor
BMAC	Boeing Military Airplane Company
CDACU	Controls and Displays ACU
CITS	Central Integrated Test System
CRACU	Critical ACU
ECMC	Electronic Countermeasures Contractor
GD	General Dynamics
GNACU	Guidance and Navigation ACU
HOL	High Order Language
IBM	International Business Machines
ISA	Instruction Set Architecture
MMP	Multipurpose Midline Processor
PACU	Preprocessor ACU
PSS	Proprietary Software Systems
RADC	Rome Air Development Center
RI	Rockwell International
SEA	Software Engineering Associates
SofTech	Software Technology
TFACU	Terrain Following ACU
USAF	United States Air Force
WDACU	Weapons Delivery ACU
WSC	Weapon System Contractor

List of Tables

List of Figures

Table 1. B-1B AP-101 ACU Applications

Nomenclature	Description	Quantity per airplane	Associate contractor responsibility
GNACU	Guidance and navigation ACU	1	BMAC—ASIC
WDACU	Weapons delivery ACU	1	BMAC—ASIC
CDACU	Controls and displays ACU	1	BMAC—ASIC
CRACU	Critical ACU	1	BMAC—ASIC
TFACU	Terrain following ACU	2	BMAC—ASIC
PACU	Preprocessor ACU	1	AIL—ECMC
CITS ACU	Central integrated test system (CITS) ACU	1	RI—WSC

Table 2. Parallel Standards Program—Phase 0 Statement of Work Tasks

Task	Description/Title
1.	B-1B avionics control unit modification
2.	Preparation of competitive procurement (MIL-STD-1750A ISA processor)
3.	Modification of the AFWAL (SEA) J73/1750A compiler
4.	Evaluation of alternate J73/1750A compiler approaches
5.	Retarget of BMAC support software to MIL-STD-1750A
6.	Development of benchmark (J73/1750A compiler evaluation system)
7.	Development of common J73/1750A implementation
8.	Definition of alternate programs
9.	Monthly status briefings
10.	Program management
11.	Phase 1 proposal

Table 3. Avionics Processor: Performance Comparison

Parameter	AP101D	AP101F
Power	600 watts	450 watts
Reliability MTBF	> 2000 hrs	> 3000 hrs
Thruput	400 kop	1000 kop
Instruction set architecture	Multipurpose midline processor (MMP)	MMP MIL-STD-1750A
Memory system	128K words (4 QMCM or 2 HMCM) 4K words ROM	[128K word (2 HMCM)-shadow 128K words SCM (active memory)] 8K words ROM

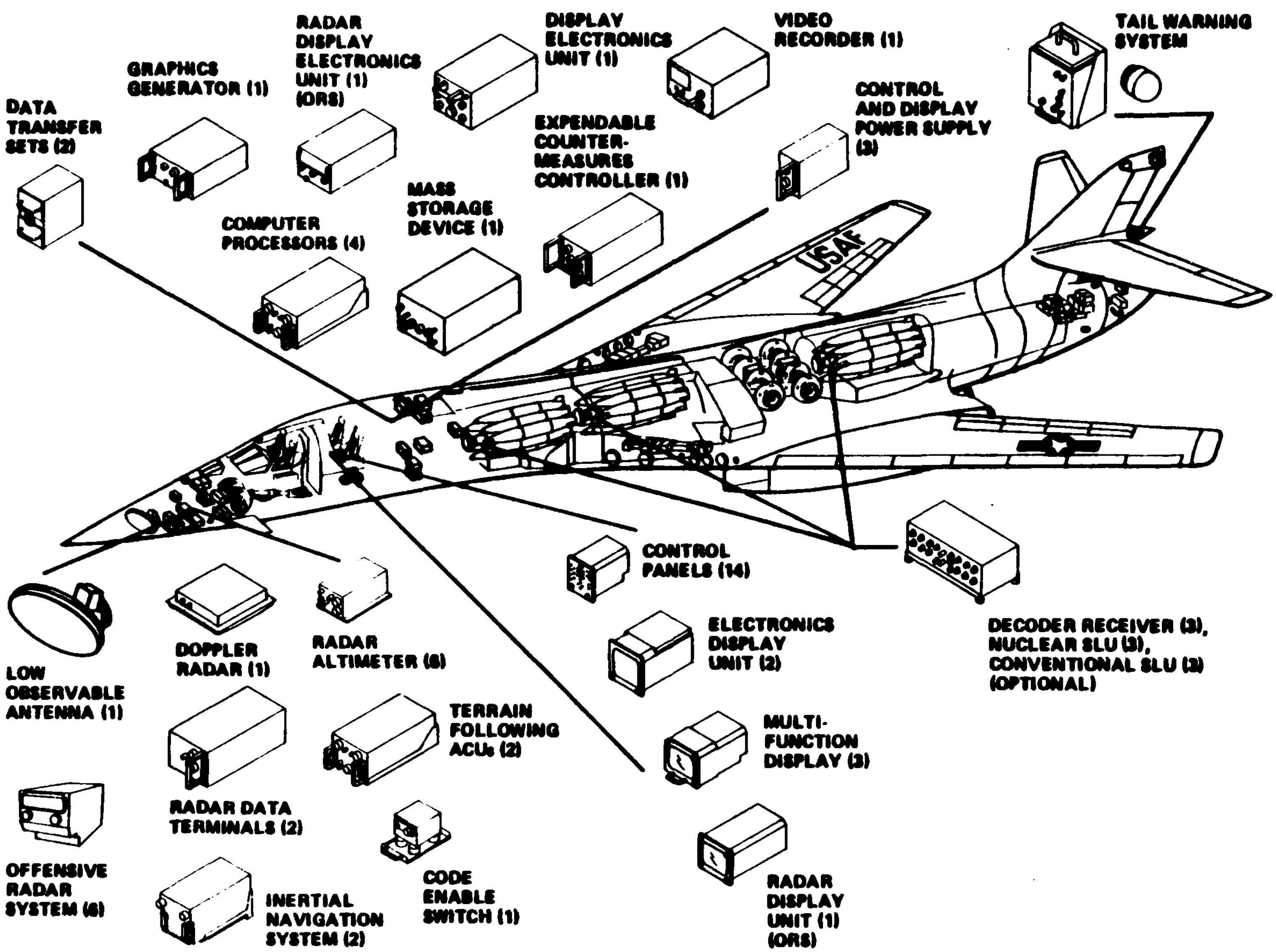


Figure 1. B-1B Offensive and Defensive Avionics Systems

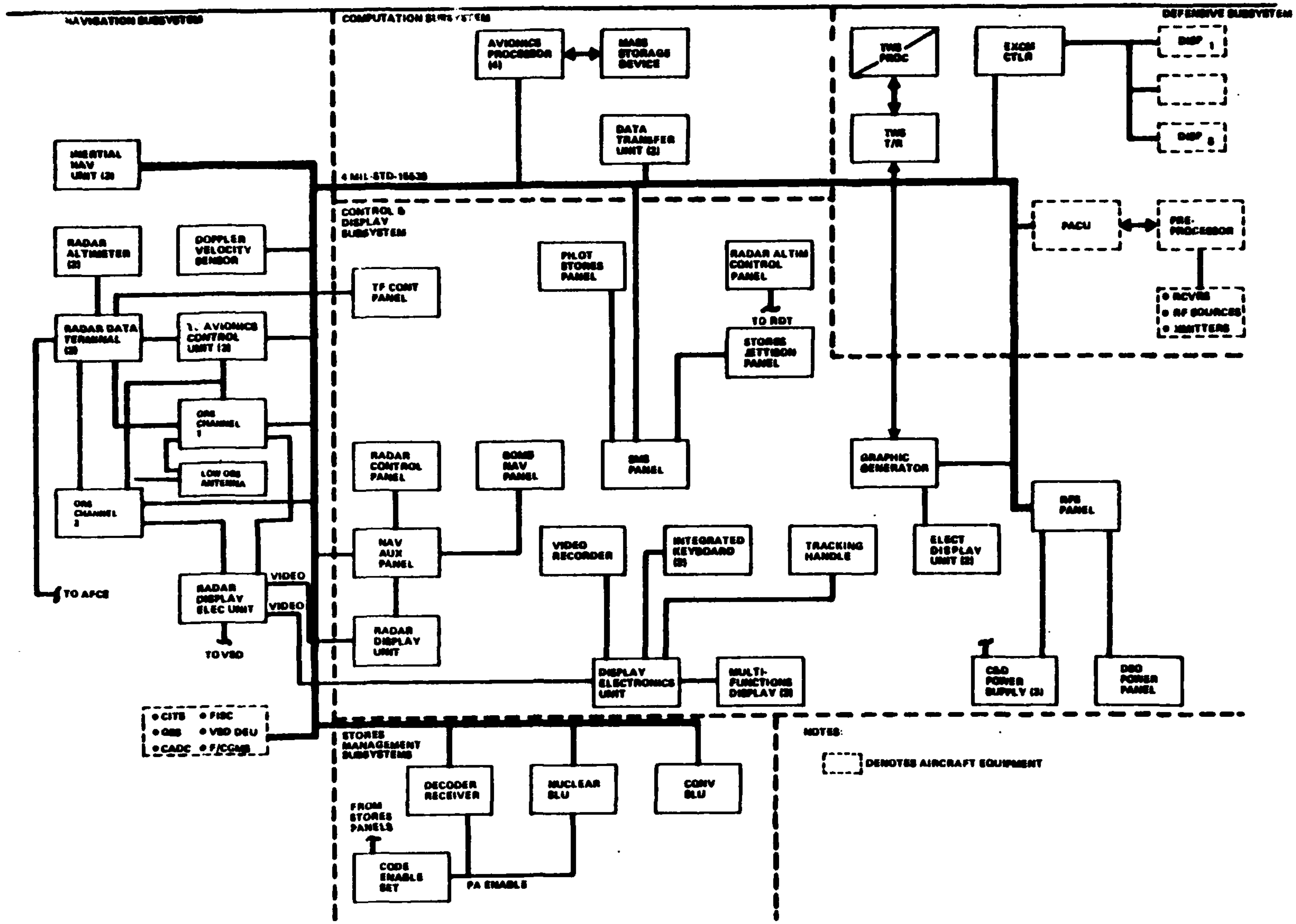


Fig 2. B-1B Avionics Configuration

AFT CREW STATION CONTROL CONSOLE

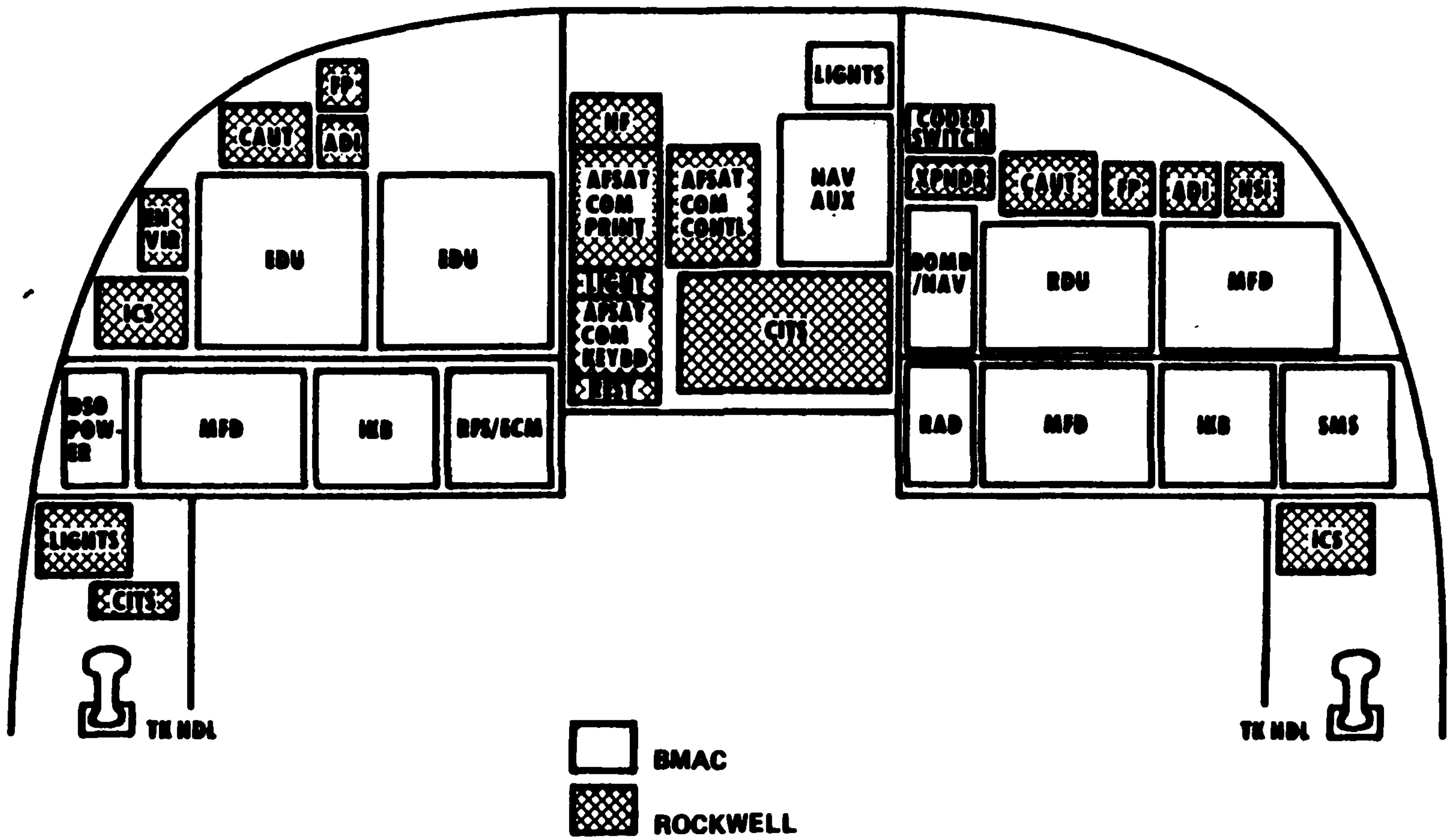


Figure 3. B-1B C&D Avionics

- 4 compiler suppliers under contract—SEA, PSS, ACT, and Softech—1 March 1982
- Compiler benchmark development initiated—20 January 1982
- MIL-STD-1750A ACU computer procurement package released to industry—25 March 1982
- B-1B Standards incorporation date alternatives of 1 September 1982, 1 December, 1982, and June 1986
- B-1B program standards incorporation impacts and status
  - MIL-STD-1750A processors meet schedules for first flight of B-1 aircraft 4 July 1984
  - Three J73/1750A compilers under development
  - ECMC (AIL) will not meet first flight incorporation
  - Dual processor development and termination liability of IBM AP-101D contract
  - Avionics flight software (block 0 and block 1) slides
    - (a) Software training
    - (b) Software rework
    - (c) Decreased software commonality (B-52 OAS/B-1B)
    - (d) Dual support software configuration

Figure 4. Key Results of Phase 0—B-1B J73/1750A Parallel Standards Program

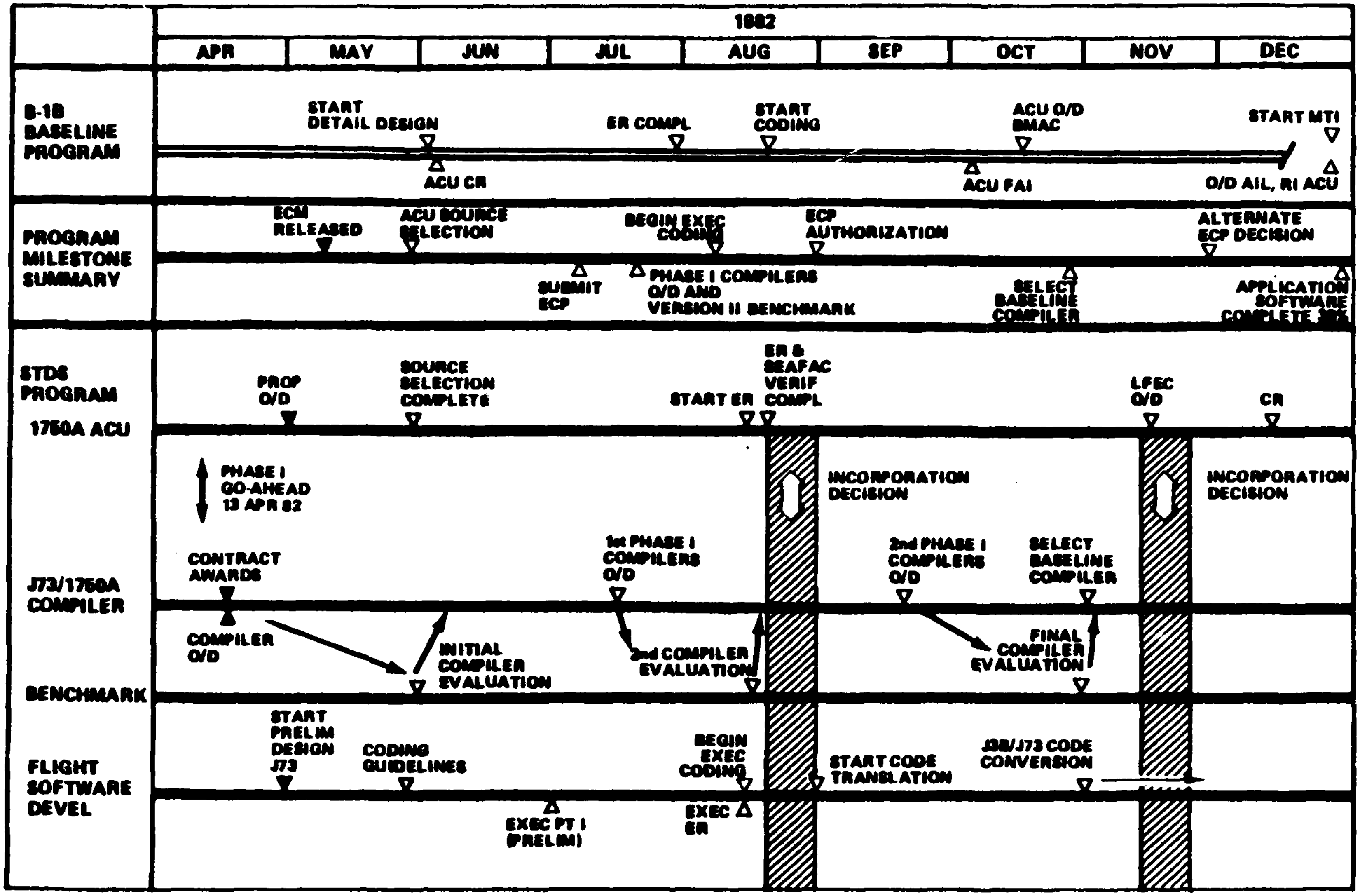


Figure 5. J73/1750A Parallel Standards Program Incorporation Plan

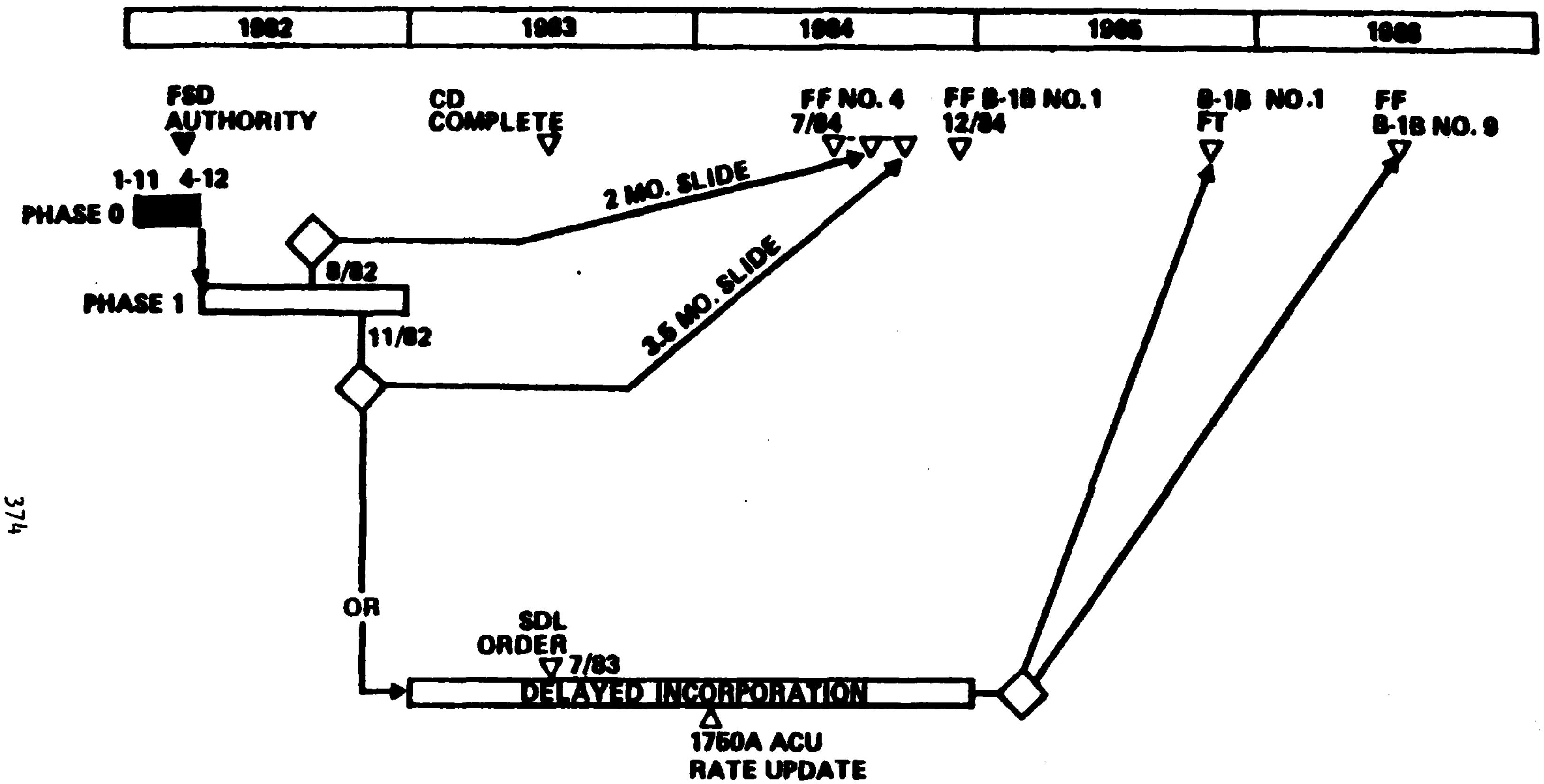


Figure 6. B-1B J73/1750A Standards Program Plans Summary

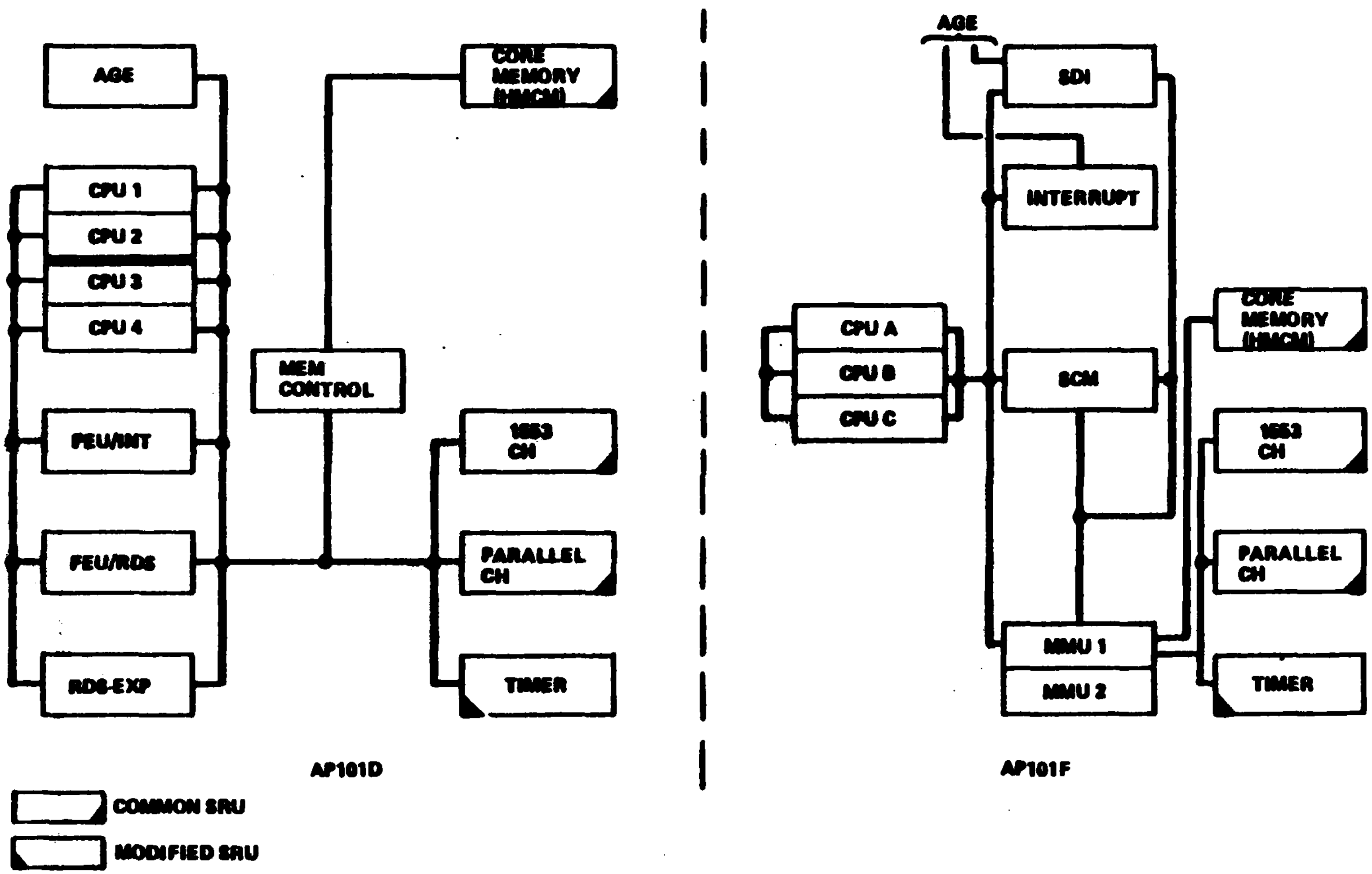


Figure 7. Avionics Processor: Block Diagram

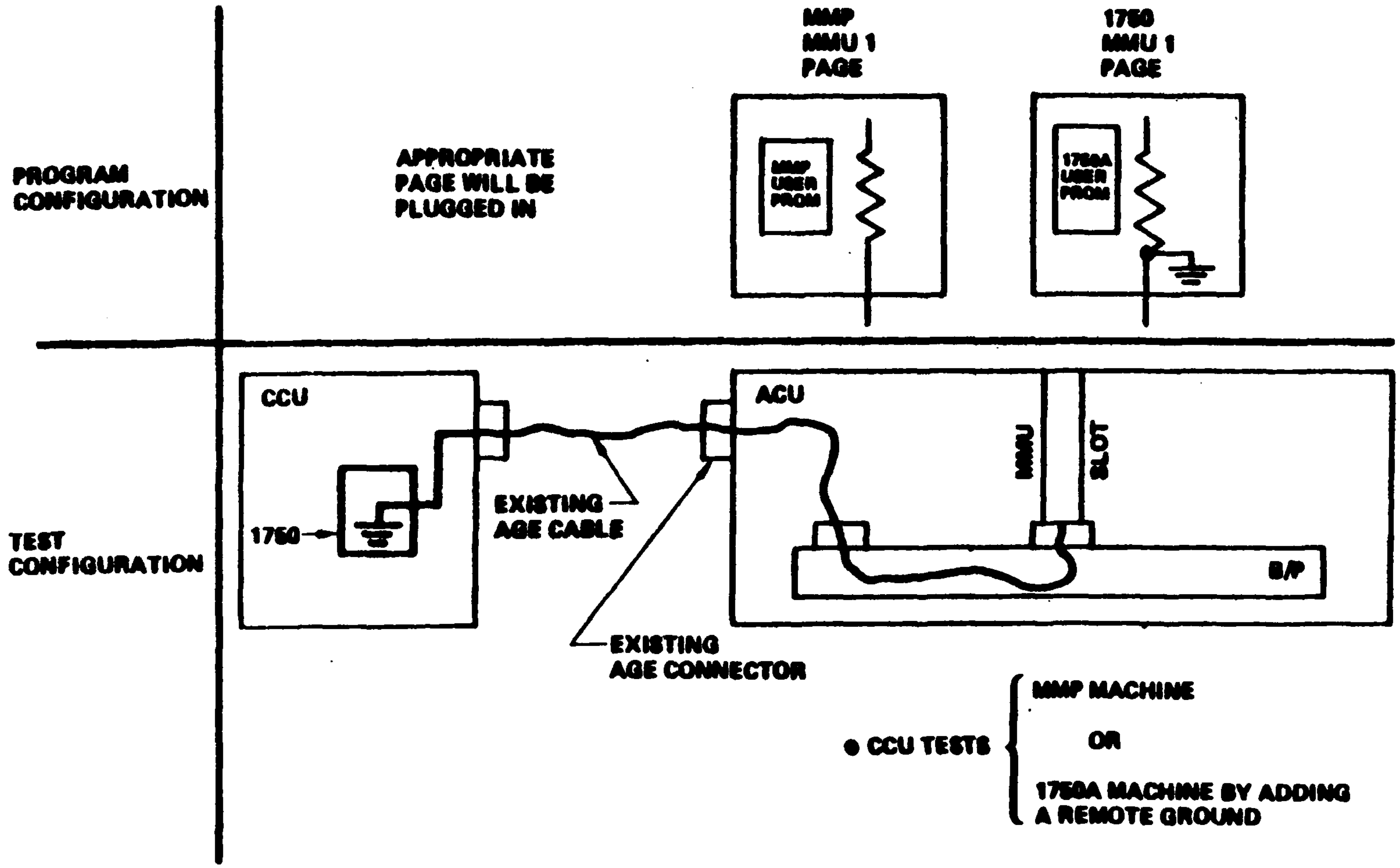


Figure 8. AP-101F Architecture Switch Concept

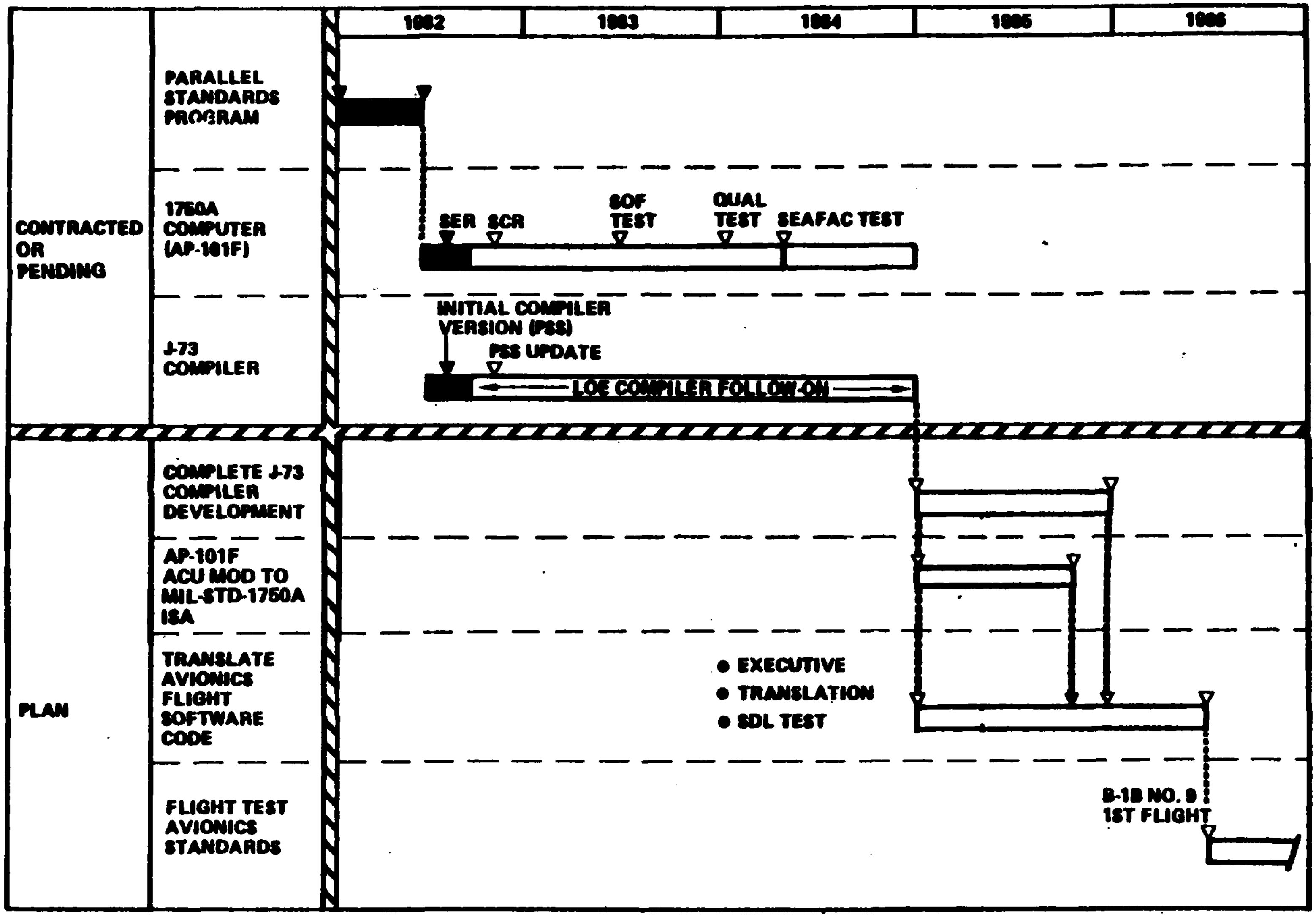


Figure 9: B-1B Avionics Standard Program

THE DIGITAL INTERFACE CHALLENGE

SESSION CHAIRMAN: Major Lee Cheshire
ASD/ENAS

MODERATOR: Colonel J. Clifford
[illegible]/AF

PREVIOUS PAGE
IS BLANK

HH-60D Advanced Avionics Architecture

November 1982

Ira Glickstein, Sr. Engineer

HH-60D Systems Engineering
304AH21
IBM Federal Systems Division
Owego, New York 13827
(607) 751-4202

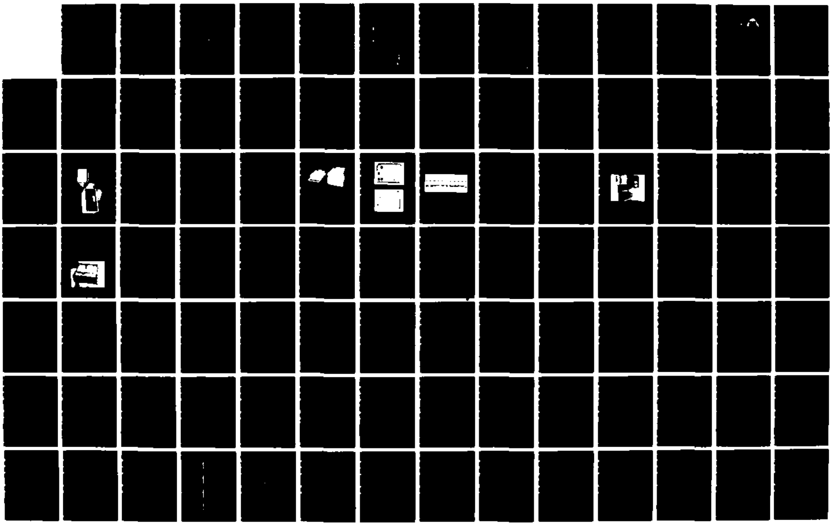
Abstract

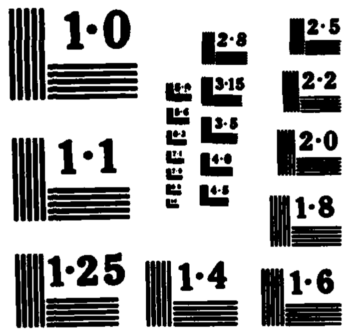
The HH-60D Night Hawk Combat Rescue Helicopter avionic system design makes extensive use of USAF/DoD interface and processing standards:

- Standard Interfaces
  - All data interfaces use a dual-redundant MIL-STD-1553B data bus, where cost and safety considerations permit. This provides operational reliability and growth flexibility.
  - Signal conversion for equipments that are not data bus compatible is performed by four separately located Remote Terminal Units that serve the cockpit/nose and transition areas. This permits use of unmodified inventory units, and reduces risk, schedule, and life-cycle costs.
  - The control and display subsystem is compatible with either 525 or 875 line TV (EIA RS-343A/RS-170) and can handle either 1:1 or 4:3 aspect ratios. This allows use of current forward looking infrared, multi-mode radar, and remote map reader technology, and future infusion of technology improvements.
- Standard Processing Hardware and Software
  - All mission processing is performed in dual-redundant MIL-STD-1750A Mission Computers, using the standard USAF higher-order language, JOVIAL J73, and structured software development disciplines. This controls the software development process and provides inherent growth flexibility through transferable software.
  - Existing distributed processors, with proven software and firmware designs, are used for peripheral tasks, where cost effective. Processors in the Display Electronics Units and Multi-Mode Radar are programmed in JOVIAL J73, making possible future transfer of software to standard USAF processors. Much of the software in the embedded processors has already been paid for under other military programs.
  - The HH-60D architecture is fully compatible with distributed processor avionics now in development for use on multiple military air vehicles.

AD-A145 697 PROCEEDINGS PAPERS OF THE AFSC (AIR FORCE SYSTEMS COMMAND) AVIONICS STAND.. (U) AERONAUTICAL SYSTEMS DIV WRIGHT-PATTERSON AFB OH DIRECTORATE O.. 5/6

UNCLASSIFIED C A PORUBCANSKY NOV 82 F/G 1/3 NL




1·0
2·8
2·5
3·15
2·2
3·5
1·1
2·0
4·0
4·5
1·8
1·25
1·4
1·6

HH-60D Advanced Avionics Architecture

Introduction

IBM Federal Systems Division has recently been selected by the Air Force as Avionics Integration Contractor for the HH-60D Night Hawk Combat Rescue Helicopter. The Night Hawk is designed for search and rescue and special operations missions. It will operate under day, night, and adverse weather conditions, and will be capable of penetration of hostile territory, using very low-level terrain-masked flight. An advanced, highly integrated, multi-sensor avionics system provides the performance, reliability, and survivability features the aircrew requires to successfully complete these demanding missions.

The Air Force HH-60D Night Hawk air vehicle is supplied by Sikorsky. It is a variation of the Army UH-60A Black Hawk and the Navy LAMPS SH-60B Seahawk. This is a good example of tri-service standardization. As prime contractor for the Navy LAMPS program, we are familiar with the H-60 airframe, and with installation and integration of sophisticated avionics on helicopters. The Air Force HH-60D has much in common with the Army and Navy versions, but is distinguished from them by external fuel tanks, air refueling capability, and an advanced, multi-sensor avionics suite.

The Night Hawk makes use of many standard Air Force/DoD avionics units, and existing or slightly modified designs. However, it has been put together around a core of integrating elements that provide a modern, highly integrated system, with inherent growth flexibility. Our design has allowed us to introduce standard Air Force processors, software, and interfaces without the need to modify existing hardware, for the most part.

The core integrating elements of the Night Hawk avionics system (see Figure 1) consist of:

- Central processing and data bus,
- Signal conversion (Remote Terminal Units), and
- General purpose controls and displays.

Central Processing and Data Bus —

This part of the Night Hawk avionics system is fully compatible with USAF/DoD processing and interface standards:

- Two MIL-STD-1750A Mission Computers,
- JOVIAL J73 mission software, and
- MIL-STD-1553B, Notice 1, dual-redundant data bus.

Mission Computers —

One of the strong candidates is the F-16 computer, with 64K words of core storage. This computer meets the MIL-STD-1750A requirement, is "off the shelf," and is in the Air Force inventory, offering standardization benefits.

We intend to use two Mission Computers, with identical software, to perform mission processing and provide primary and backup data bus control. With this architecture, failure of either Mission Computer will cause absolutely no degradation in mission performance.

Alternatives to Multi-Computer Architecture —

We could have used a small, non-MIL-STD-1750A processor for "degraded" backup bus control. This approach could have saved the cost and weight of one of the two Mission Computers, however, it would have increased software design, development, integration, and test costs. We would have had to define the "minimum essential get-home" modes, and develop and test operational procedures for the pilots to use under these degraded conditions.

Advantages of Multi-Computer Architecture —

With the multi-computer approach, only one operational flight program need be designed, developed, integrated, tested, and maintained. Only one set of procedures need be learned by the pilots. This saves time and money, and reduces risk, during the full scale development phase of the HH-60D program.

For the production phase, use of standard Air Force processors for centralized processing and data bus control will be more cost-effective than introduction of non-standard processors in this area. We recognize that the processor area is rapidly changing due to infusion of new technology. We expect the MIL-STD-1750A chip sets, and fast, dense, low-cost monolithic storage devices, to make future

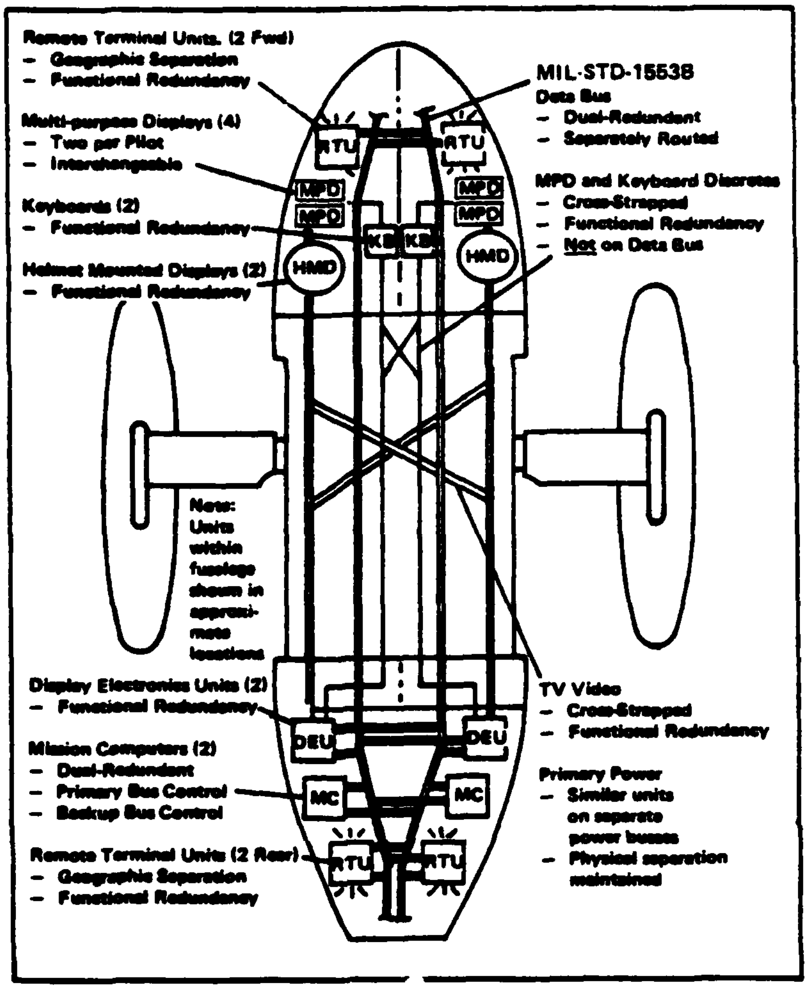


Figure 1. HH-60D Core Integrating Elements use Standard Processors, Software and Interfaces

standard processors far more cost-effective than current ones. By the time the HH-60D goes to production, we may find higher speed and storage capacity in the same size box, at attractive prices. Should this be the case, our multi-computer architecture, utilizing JOVIAL J73 software in MIL-STD-1750A processors, will be ready to exploit the new technology.

JOVIAL J73 Software --

As noted above, the software in the Mission Computers utilizes the Air Force standard higher order language. In addition, some of the software in embedded processors will also be written in JOVIAL J73. This includes the system processor portion of the Display Electronics Units (DEU), and a portion of the multi-mode radar software. In addition to providing better documentation and control of the software development process, this opens the possibility of transfering some of this software into MIL-STD-1750A chip set processors. This may be attractive, in the future, as a way of reducing size, weight, and cost of subsystems, and increasing overall performance by exploiting software flexibility.

Data Bus --

The equipments listed in Figure 2 are interfaced directly with the MIL-STD-1553B, Notice 1, dual-redundant data bus. This totals 12 units, providing 18 spare addresses for future growth. We expect all future equipments under development by the Air Force/DoD to be bus compatible. This includes candidates for future HH-60D upgrade such as GPS, Electronic Survivor Location Equipment (ESLE), and defensive subsystems.

Mission Computer #1 (Primary Bus Controller)
Mission Computer #2 (Backup Bus Controller)

Remote Terminal Unit #1
Remote Terminal Unit #2
Remote Terminal Unit #3
Remote Terminal Unit #4

Display Electronics Unit #1
Display Electronics Unit #2

Inertial Navigation Set (ASN-141)
Multi-Mode Radar (LANTIRN Modified)
Remote Map Reader
Doppler Radar

GPS (Growth)
Electronic Survivor Location Equipment (Growth)

Figure 2. Units on HH-60D MIL-STD-1553B Data Bus

In addition to growth flexibility, use of the MIL-STD-1553B data bus provides reliable, redundant interfaces between equipments. This has allowed us to make use of shared display units for engine instruments, warning/caution/advisory indications, and other functions that traditionally have dedicated indicators. Our dual-redundant data bus, redundant processors, and multiple, interchangeable displays are actually more reliable and survivable than simplex dedicated indicators. Our approach also saves panel space and puts critical indications in primary viewing areas for better pilot response.

In the highly unlikely case that both Mission Computers fail, or the dual-redundant data bus fails, the following functions continue to operate: 1) Standby flight instruments, 2) Intercom, 3)UHF radio, and 4) Automatic Flight Control System.

Signal Conversion (Remote Terminal Units) —

In order to reduce full scale development and life cycle costs, we selected hardware that was already in Air Force/DoD inventory. Some of these inventory units, such as the ASN-141 Inertial Navigation System (INS), are compatible with the data bus, but most are not. Units that are not data bus compatible require signal conversion. DC analog, synchro, discrete, and digital signals are converted to MIL-STD-1553B by four Remote Terminal Units (RTUs). Two RTUs are located in the forward part of the helicopter (nose/cockpit area), and two are in the rear (transition section).

Each of the RTUs has a common design, except for plug-in subsystem interface modules and associated interface wiring. Figure 3 is a simplified overall view of the Night Hawk's avionics architecture and interconnection, and indicates the preliminary partitioning of signal conversion between the four RTUs. We have partitioned the signals in accordance with the following guidelines:

- Critical signals, such as warning/caution/advisory and engine parameters, are converted by two different RTUs, so that loss of one RTU does not cause degradation.
- Where completely redundant signal conversion was not practical, we used functional redundancy, such as interfacing each of the VHF radios via a different RTU, so the VHF function remains despite loss of one RTU.
- If practical, we intend to make the two forward RTUs identical.
- Signals are generally routed to the RTU that is closest, to minimize aircraft wiring.
- To the extent possible, spare capability is evenly divided between RTUs.

Our use of interchangeable, physically separated RTUs, and dual or functionally redundant signal conversion, assures reliable, cost-effective MIL-STD-1553B compatibility. We have utilized existing, standard avionics units, for minimum acquisition and operational costs, making full use of Air Force/DoD maintenance, training, and logistics assets.

Shared Controls and Displays —

Most HH-60D control and display functions, for both pilot and copilot, utilize shared control and display units, located in prime viewing and reach areas. Figure 4 shows the instrument panel and center console. These shared resources include:

- Four interchangeable Multi-Purpose Displays (MPD).
- Two Helmet Mounted Displays (HMD).
- Two alpha-numeric keyboards.

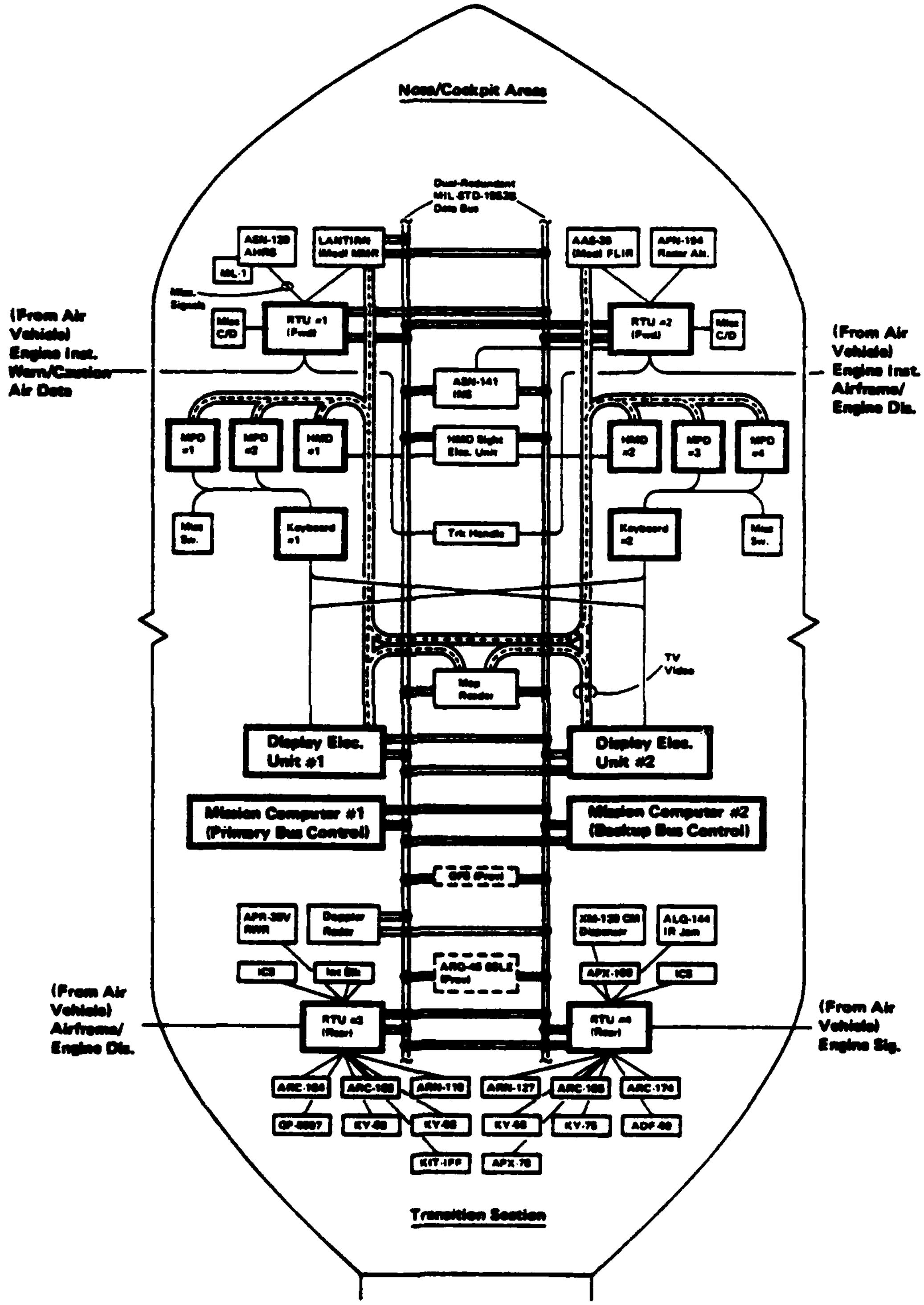


Figure 3. Avionics Subsystem Block Diagram

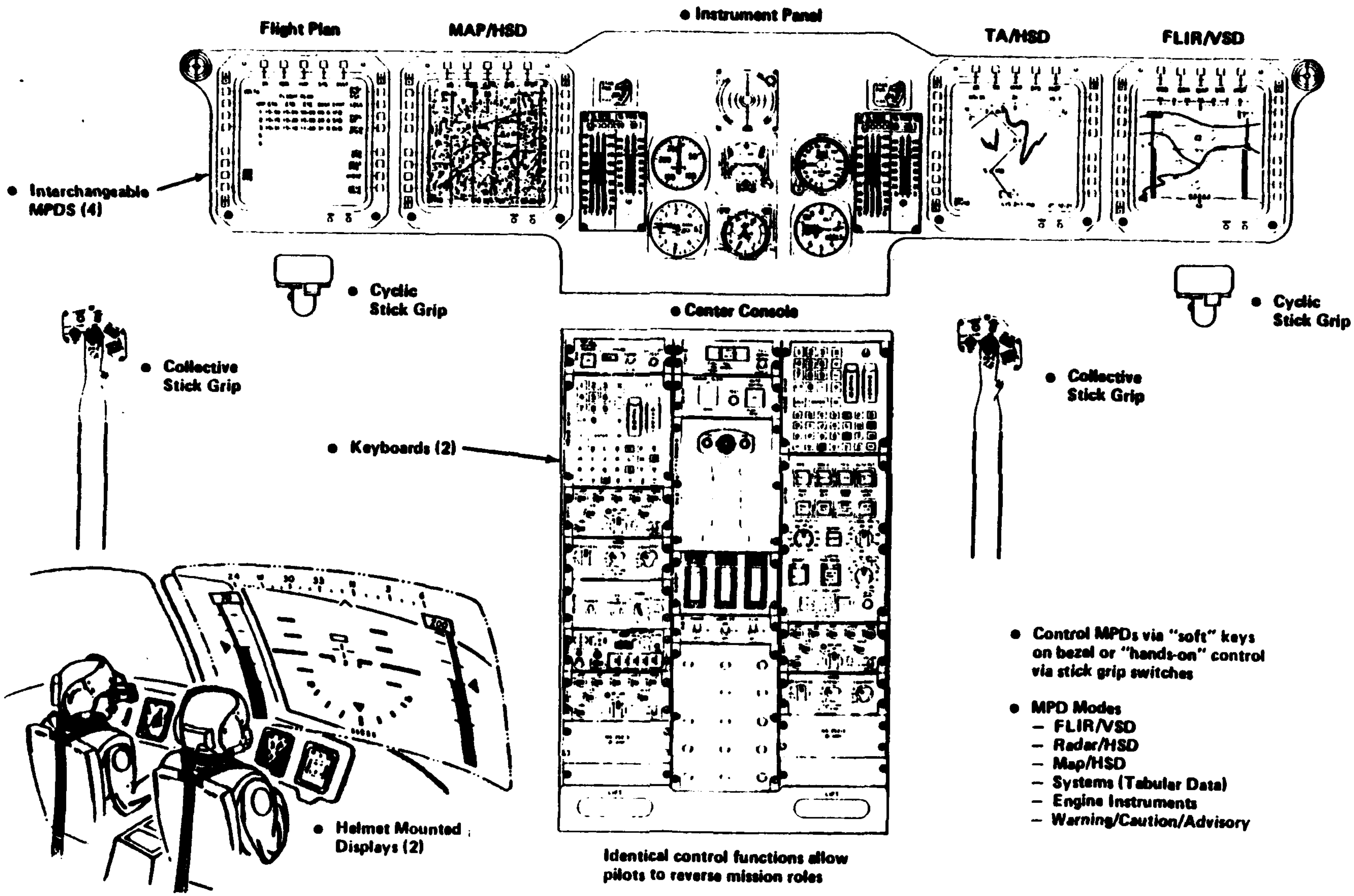


Figure 4. Standardized, Shared Controls and Displays

- A tracking handle,
- Cyclic and collective stick grip switches, and
- Two display electronics units (DEU).

Any display format may be selected on any MPD. If both DEUs are operational, up to four different display formats may be in use by pilot and copilot at any time. It is thus completely up to the pilot and copilot which display formats will be displayed on their MPDs and HMDs.

The MPDs also provide control capability via 21 "soft" keys located in the bezel of each MPD. The function performed by each key is software controlled, and is displayed on the surface of the MPD near the key. When, for example, the IR mode (Forward Looking Infrared) is selected on a given MPD, that MPD becomes a complete control and display surface for the IR sensor. If the RDR mode (Multi-Mode Radar) is selected, the MPD becomes a complete control and display surface for the RDR sensor. We do not have a separate control panel for either the IR or RDR sensors, nor do we have separate panels for the remote map reader, INS, etc. Should any MPD fail, the function may be performed on any of the remaining MPDs. Should either of the two DEUs fail, all MPDs and HMDs continue to operate, but the pilot and copilot are limited to a total of two display formats.

We do not have separate keysets for entry of navigation data, communications data, etc. All alpha-numeric data entry is via the two interchangeable keyboards. Should one fail, any type of data entry may be performed on the other.

We have standardized our video interfaces to utilize 875 line, 1:1 aspect ratio TV in accordance with Electronic Industry Association (EIA) RS-343A. However, our MPDs and DEUs can also handle 525 lines, and 4:3 aspect ratio (EIA RS-170) signal standards.

We expect that new or modified sensors will be added to the HH-60D avionics suite in the future to extend the operational life of the system. With our shared control and display architecture, featuring standardized, interchangeable control and display units, standard data and video interfaces, and software control, we have inherent growth flexibility. We believe that the Night Hawk's mission controls and displays are easier to use, have higher operational reliability and survivability, and are lighter and less expensive than separate, dedicated control and display panels.

Future Applications

HH-60D Night Hawk use of standard Air Force/DoD equipments, processors, software, and interfaces is a good example of how these standards may be used without stifling creativity or limiting overall system performance or growth flexibility. In fact, the use of standard processors, interfaces, and software has improved performance and growth flexibility. We expect very similar core architecture to be utilized on future projects in the helicopter/VSTOL area, such as the Joint Services Advanced Vertical Lift Aircraft (JVX), and other avionics systems.

Biographical Sketch

Ira Glickstein is a Senior Engineer at IBM's Federal Systems Division, Owego, New York. He served as lead systems engineer on the HH-60D proposal, with responsibilities in the system architecture and controls/displays areas. He is currently in the HH-60D systems engineering department.

Ira has been involved in new business and initial development engineering for several projects at IBM, including: Coherent Emitter Location Testbed (CELT), Army Command and Control Master Plan, Joint Tactical Information Distribution System (JTIDS), Adaptable Surface Interface Terminal (ASIT), A-4M Angle Rate Bombing System, Light Airborne Multipurpose System (LAMPS), AC-130E Gunship, and A-7D/E Navigation and Weapon Delivery System. He received Outstanding Contributon awards for his work on a computer memory correction method and for systems engineering efforts.

Prior to coming to IBM, he was employed by Lockheed Electronics and Norden. He holds an electrical engineering degree from City College of New York and a Professional Engineering License (New York).

FAIRCHILD'S DATA TRANSFER SYSTEM - UTILIZING THE LATEST MILITARY STANDARDS

AUTHORS

Stephen L. Belechak-Becraft/George D. Farmer
Fairchild Space & Electronics Company

ABSTRACT

Fairchild's F-16 Data Transfer Unit (DTU) is a cockpit mounted, microprocessor-controlled unit which provides for automatic exchange of pre-flight, in-flight, and post-flight information with the avionics system over the MIL-STD-1553B (or A) serial digital multiplex data bus. This information exchange is facilitated by a detachable nonvolatile Data Transfer Cartridge (DTC). On the ground and away from the aircraft, the information in the DTC is managed via the computer-based Ground Support Equipment.

The unit is implemented with the latest military standards encompassing mux bus operation (MIL-STD-1553B), computer hardware implementation (MIL-STD-1750A), and software development and maintenance (MIL-STD-1589B). Employment of these standards enhances commonality and compatibility with the latest avionic systems, improves maintenance, and reduces life cycle costs.

BACKGROUND

Avionics systems of current military aircraft have substantial and ever-increasing informational requirements. Specifically, this information takes three forms:

a) PRE-FLIGHT:

Pre-program information into aircraft for initialization of subsystems and for storage purposes so that the data can be recalled during the flight. Such information may include aircraft status, waypoints, stores management initialization and sequencing, threat data, navigation and communication data and coordination data.

b) IN-FLIGHT:

Record all flight parameters and pertinent mission data to be used for analysis and debriefing. This data would typically include threat location, tactical data, maintenance data, system "built-in-test", training information, airframe and engine fatigue life. Also, during flight, provide subsystem initialization parameters and a flexible data base access.

c) POST-FLIGHT:

Retrieve all mission data for debriefing and maintenance purposes.

The increase in digital communications and data processing influenced by microprocessor development has given rise to system designs that are programmable, flexible, and capable of handling a multitude of options.

At the present growth rate of computing power that can be embodied in each subsystem, the demand for a method and mechanism to provide efficient "data transfer" is necessary.

The magnitude of these informational avionic requirements have reached a point where direct communication of this data to and from the flight crew is not effective. The scenario is time consuming, prone to errors, and quantity limited. In a fighter aircraft, the pilot must routinely enter a thousand digital alphanumeric characters before take-off if he intends to make use of present avionics processor capabilities. This is clearly unacceptable from an operation standpoint and unthinkable for next generation avionics.

An alternative is to use a memory storage device which can, on command, automatically transfer data directly to/from aircraft avionic systems. This device would complete this task faster and with fewer errors than the flight crew. The memory storage element of this device would be portable and nonvolatile; thereby, initialization, access, and analysis of this information need not be conducted at the aircraft. Such a device has been developed under the F-16 Data Transfer System contract award to Fairchild Space & Electronics Company from General Dynamics.

F-16 DATA TRANSFER SYSTEM

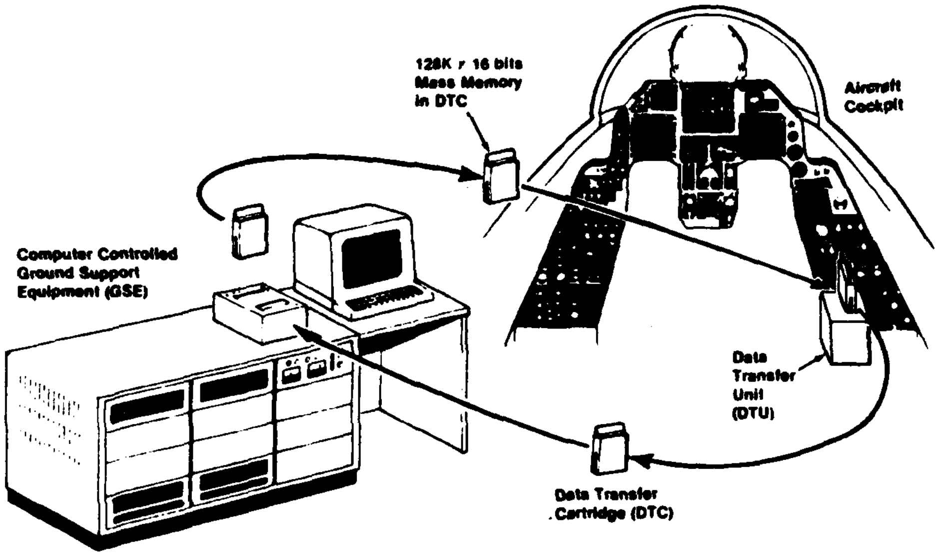
The Data Transfer System concept is depicted in Figure 1. The Data Transfer System consists of Data Transfer Equipment (DTE) and computer controlled Ground Support Equipment (GSE). The DTE consists of a Data Transfer Unit (DTU) and a Data Transfer Cartridge (DTC).

Utilization of this system begins at Pre-Flight. The ground-based computer, GSE, accurately loads and verifies preprogrammed mission data into the portable DTC for initialization of aircraft subsystems. The DTC is then inserted into the cockpit resident DTU. In flight, the microprocessor-controlled DTU provides real-time access to the DTU memory over the MIL-STD-1553 data bus. After the flight, the DTC is removed from the cockpit and inserted into the GSE. The GSE is then used to retrieve all mission data for debriefing and maintenance purposes.

The DTE may be compared to secondary storage devices found in computer systems. Relative to this analogy, the DTU is considered to be the unit controller and the DTC is considered to be the unit storage medium (e.g., floppy disk). As a secondary storage device, the DTE is designed to interface as a remote terminal on a MIL-STD-1553 type data bus. It is over this data bus that the DTE will receive commands and data and transmit status and data. This data exchange is depicted in Figure 2, Avionic Architecture and Data Exchange.

The DTU contains a complete file management system which allows the bus controller (mission computer) to access data in the DTC without knowing the address location of the data, only the "filename" is needed. This also allows testing and data verification to be performed by the DTU. The use of file management clearly reduces the new software development overhead required by the bus controller in order to use the DTU.

DATA TRANSFER SYSTEM



PREFLIGHT:

- **PREPROGRAM INFORMATION INTO A DATA CARTRIDGE (USING GROUND SUPPORT EQUIPMENT) FOR INITIALIZATION OF AIRCRAFT SUBSYSTEMS AND STORAGE OF INFLIGHT MISSION DATA.**
- **INSERT PREPARED DATA CARTRIDGE INTO A COCKPIT-RESIDENT RECEPTACLE.**

IN-FLIGHT:

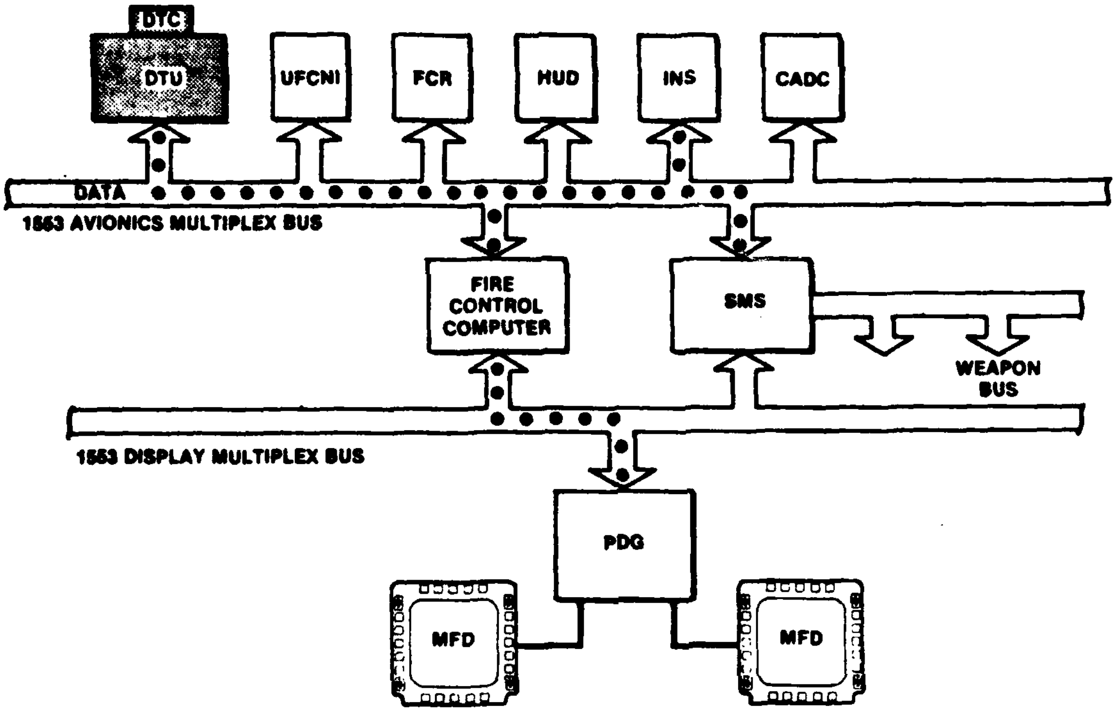
- **PROVIDE SUBSYSTEM INITIALIZATION PARAMETERS AND A FLEXIBLE DATA BASE.**
- **PROVIDE INFLIGHT MISSION DATA BASE**
- **RECORD FLIGHT PARAMETERS, MISSION DATA AND MAINTENANCE INFORMATION USED FOR ANALYSIS AND DEBRIEFING.**

POST-FLIGHT:

- **REMOVE DATA CARTRIDGE FROM COCKPIT**
- **RETRIEVE ALL MISSION DATA FOR DEBRIEFING AND MAINTENANCE PURPOSES**
- **UTILIZING THE GROUND SUPPORT EQUIPMENT (GSE), THE DATA TRANSFER EQUIPMENT (DTE) SUPPORTS THE AUTOMATED CORRELATION OF INFORMATION FROM MULTIPLE MISSIONS**

Figure 1. Data Transfer System Concept

AVIONIC ARCHITECTURE AND DATA EXCHANGE



- AVIONICS BUS CONTROLLER TRANSFERS DATA TO AND FROM DTU
- SYSTEM SUPPORTS REAL TIME 1553 DATA TRANSFERS INCLUDING REMOTE TO REMOTE MODES
- MULTI-FUNCTION DISPLAY HAS ACCESS TO THE DTC DATA BASE THROUGH FUNCTION SELECT SWITCHES. INTEGRATED BY THE FIRE CONTROL COMPUTER
- CHANGES IN DATA CONTENT ARE ACCOMPLISHED WITH THE MFD/KEYBOARD
- COMPUTER CAN ACCESS FILES/RECORDS WITH SYMBOLIC REFERENCES (FILE NAME RATHER THAN BY ABSOLUTE MEMORY LOCATION)
- MULTIPLE MODES OF FILE ACCESS PROVIDE FLEXIBILITY TO BUS CONTROLLER AND SIMPLIFY CONTROL ALGORITHMS

Figure 2. Avionic Architecture and Data Exchange

Further, the mission computer does not have the burden of executing a file management system, and associated error routines, otherwise needed if the DTU did not contain same.

DTS APPLICATIONS

Figure 3 highlights the major types of information to be stored and or retrieved from the DTC, such as: communication channels, waypoints, threat data, etc. The DTU does address the proper handling of threat data. A DTC erase switch in the cockpit is hardwired to the DTU and can be activated by the pilot. The DTC contains CMOS RAM and, therefore, erasure is near instantaneous. If the DTC is removed from the DTU, erasure can still take place by closure of the local erase switch on the DTC itself.

Not only can the DTU be used for the obvious purpose of data transfer, but in addition may contain program overlays from the mission computer, as well as being treated as an extension of the mission computer memory.

Also listed in Figure 3 are the benefits of using the DTS, ranging from reduced pilot workload to increased mission effectiveness.

DTS APPLICATIONS

PREPROGRAMMED INFORMATION

- RADIO CHANNELS
- TACAN STATION LOCATIONS
- WAYPOINTS
- WEAPON DELIVERY PROGRAMS
- RELEASE POINTS
- NAVIGATIONAL UPDATES
- AIRCRAFT STATUS
- STORES MANAGEMENT DATA

RECORDED INFORMATION

- TACTICAL DATA
- THREAT LOCATION
- MAINTENANCE DATA
- SYSTEM BIT
- TRAINING INFORMATION
- AIRFRAME & ENGINE PARAMETERS

FUTURE APPLICATIONS

- PREFLIGHT/POSTFLIGHT SYSTEM TEST ENHANCEMENT
- SUPPORT LARGE MEMORY STORAGE FOR PREPROGRAMMING (TERCOM, ASPJ, ECM, ETC.)
- MISSION ORIENTED DATA BASE MANAGEMENT SYSTEM
- FLIGHT MANAGEMENT COMPUTATION

- REDUCES PILOT WORKLOAD
- DECREASES REACTION TIME
- REDUCES PROGRAMMING ERRORS
- INCREASES MISSION EFFECTIVENESS
- SUPPORTS MISSION COMPUTER
- SUPPORTS MULTI-FUNCTION DISPLAYS
- SUPPORTS ADVANCED SYSTEM AVIONICS
- IMPROVES MAINTAINABILITY
- IMPROVES FLIGHT HISTORY RECALL
- IMPROVES OPERATIONAL EFFECTIVENESS
- PERMITS REAL TIME OPERATION WITH A/C MULTIPLEX DATA BUS

Figure 3. Data Transfer System Applications

AVIONIC SUBSYSTEMS

The present day avionic suite in modern military aircraft is very "integrated". This term recognizes the need for subsystems to be highly interactive on future aircraft. In the past, many of these subsystems were significantly independent in function. However, improved performance can be obtained by correlating the information developed and the control processes applied by these subsystems. Figure 4 (F-18 Avionics Multiplex Bus) and Figure 5 (Advanced F-16 Avionics System Architecture) represent present and near future avionic suites. Common to these aircraft avionics is the following:

- Multiple MIL-STD-1553 Multiplex Buses (trend to hierarchical buses)
- Proliferation of Digital Equipment
- Software Intensive
- Intelligence distributed to each subsystem (Embedded Processors, Microprocessors)

The DTU is consistent with the above trend. In support of distributed processing, off-loading tasks from bus controller (mission computer), the DTU contains a file management system to facilitate DTC data accesses. Thereby, the mission computer needs only to know the data organization within its file and not how the information is stored within the DTC. This also allows testing and data verification to be the responsibility of the DTU rather than the data user devices. DTC data access is achieved by specifying a sixteen-bit logical file name. This opens a 512-word window (called a page) within the specified file. Data access pointers are used as the pointers within the page where data transfers occur.

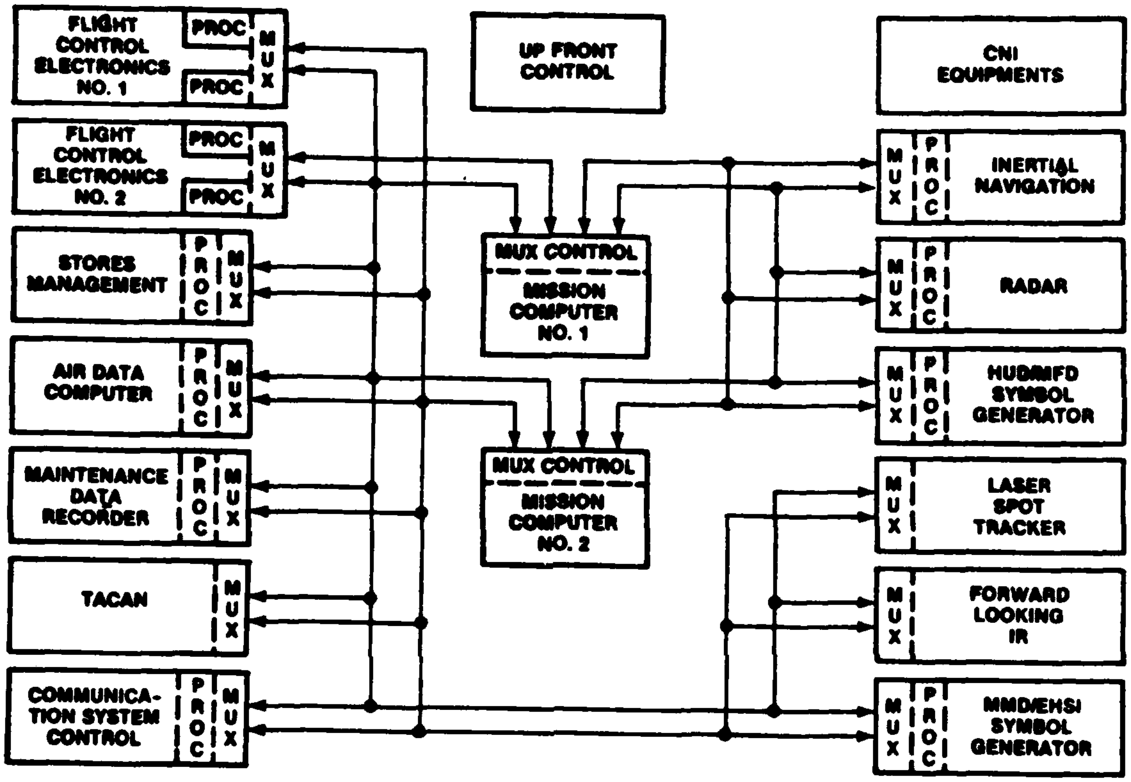


Figure 4. F-18 Avionics Multiplex Bus

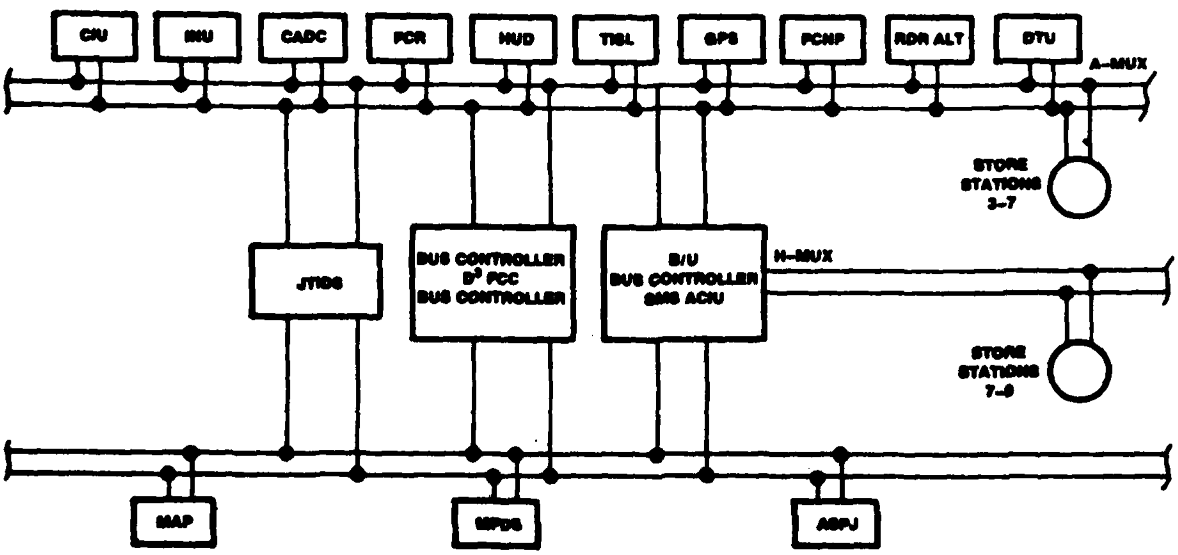
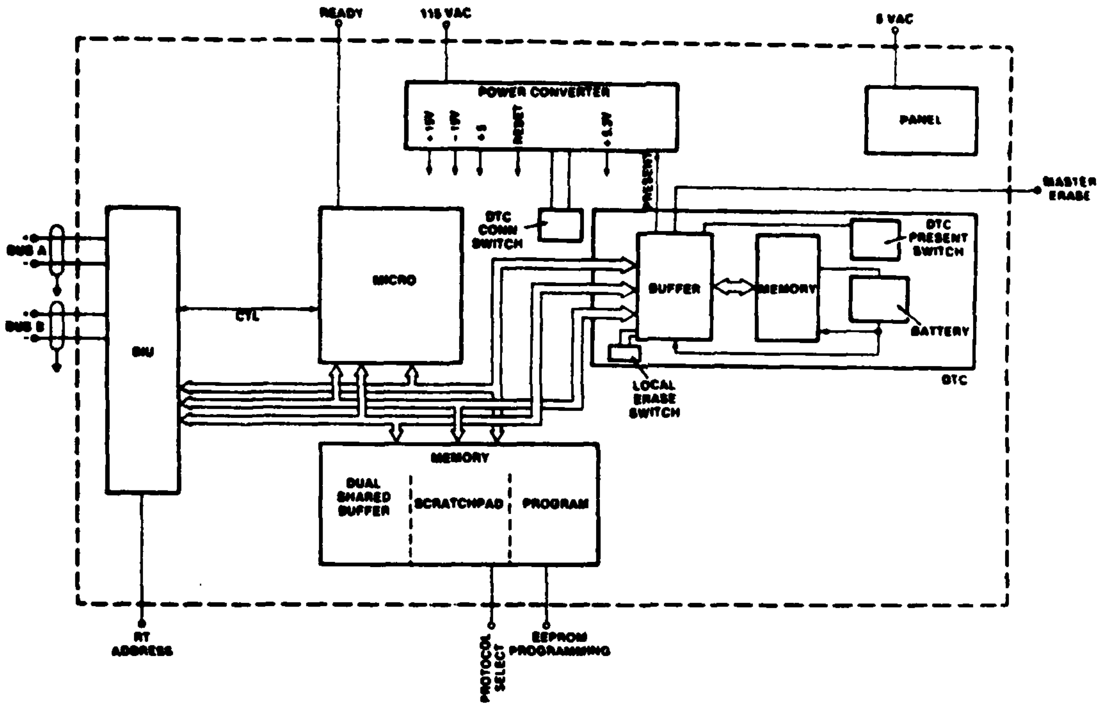


Figure 5. Advanced F-16 Avionic System Architecture

Commands are provided that allow bus devices to alter these pointers, change the access window and examine system status including the page number, file name, file size as well as other information. Data access modes may also be specified to allow the using device the flexibility to optimize data access for its particular application. Files may also be write-protected either in flight or by the Ground Support Equipment prior to the mission allowing read-only access to vital information. This file management technique is analogous to accessing a floppy disk drive in a commercial home microcomputer system.

Higher throughputs will be required from avionic subsystems in near future years. The F-16 DTU contains full parallel address and data line interface to the DTC, thereby providing an inherent high speed architecture. This architecture is depicted in Figure 6, DTU/DTC Block Diagram.

The DTU also addresses the importance of exhaustive self-test and built-in-test (ST/BIT). The proper ground work for ST/BIT begins with functional card partitioning within the unit. Each card in the DTU is an isolated function: 1553 bus interface (BIU), microprocessor controller (MICRO), memory and power supply (see Figure 6, DTU/DTC Block Diagram). Testing in the DTE is divided into four categories: initialization testing, real-time testing, background testing, and foreground testing. Initialization testing verifies that the DTE is operational before allowing any 1553 bus commands to be received. These tests include checking RAM, ROM, and the CPU for proper operation. Should any one of these fail, the DTE does not respond to any bus commands. Some hardware timing tests are also checked at this time since bus traffic would prevent accurate timing measurements but errors for these tests are only reported is the test status word and do not



- DTC INSERTION (POWER ON)
- MULTIPLEX BUS INTERFACE OPERATION
- PROCESSING OF BUS COMMANDS
  - 1553 MODE COMMANDS
  - CONTROL COMMANDS
  - STATUS COMMANDS
  - READ/WRITE COMMANDS
- DTC FILE MANAGEMENT
- BIT/SELF-TEST
- DUAL A/B PAGE BUFFER
- POWER FAIL/RECOVERY
- DTC REMOVAL (POWER DOWN)
- DTC ERASE — MASTER & LOCAL ERASE
- EEPROM PROGRAMMING
- DUAL 1553 PROTOCOL (SELECTABLE)

Figure 6. DTU/DTC Block Diagram

suspend DTE operation. Tests performed in real-time verify proper bus communication. The Remote Terminal (RT) address in the received 1553 command word is checked to ensure that the DTE is only responding to the assigned terminal address. File management commands are checked to ensure they are received in a meaningful sequence (such as reading data before a file is opened for access). Data access commands are also checked for validity (e.g., not reading more data than the file contains or writing to a write-protected file). Other tests, such as illegal file names and data words are also checked during this real-time phase of testing. Background testing is conducted whenever the DTC is not busy processing bus commands. All aspects of the DTU and DTC (with the exception of the BIU) are tested. Checksums and parity are used to verify both local and DTC RAM. ROM and the CPU as well as other hardware are checked for proper operation. Foreground testing is initiated by a bus command. This allows the DTE to perform tests on the BIU. These tests verify proper operation of the Bus Interface up to the bus transmitters (no data is actually sent out on the bus). All of the other tests indicated above in the background test are also performed.

There never seems to be adequate memory storage space in avionic systems for future needs. The F-16 DTE design anticipated future memory requirements for such units as weapon systems, JTIDS, etc. The DTU can address up to two megawords of DTC data. The present DTC can be configured for up to 128K words (16 data bits plus one parity bit). This high density is accomplished by the memory carriers used. Each memory carrier, measuring less than 3.6" x 1.25" x 0.25", provides 8K words of buffered memory. On every carrier are 32 4K x 1 CMOS memories with associated data, address, and control buffers, each packaged in a leadless chip carrier. The DTC will allow up to sixteen memory carriers providing from 8K words to 128K words of memory in 8K word increments. The electrical design of the interface circuits, both on and off the carriers, was implemented such that only one memory carrier would be enabled during any given access. This yields dynamic power requirements to be equal to that of just one memory carrier regardless of the size of the DTC memory. The DTC was designed to contain a large amount of memory but still maintain the access speed, power consumption, and volume associated with much smaller memory systems. Todays 2Kx8 CMOS RAM provides DTC memory expansion to 256K words with the development of new memory carriers.

STANDARD ISSUES

Many avionic subsystems today are dedicated designs optimized to an aircraft type or system. This magnifies DOD support costs (life cycle costs) since support tools (hardware and software) are not common from aircraft to aircraft nor even from program to program on the same aircraft.

Military standardization provides an effective means to lower unit life cycle costs. MIL-STD-1553 bus interface is an example of a standardization that permits a common front end solution to avionic boxes.

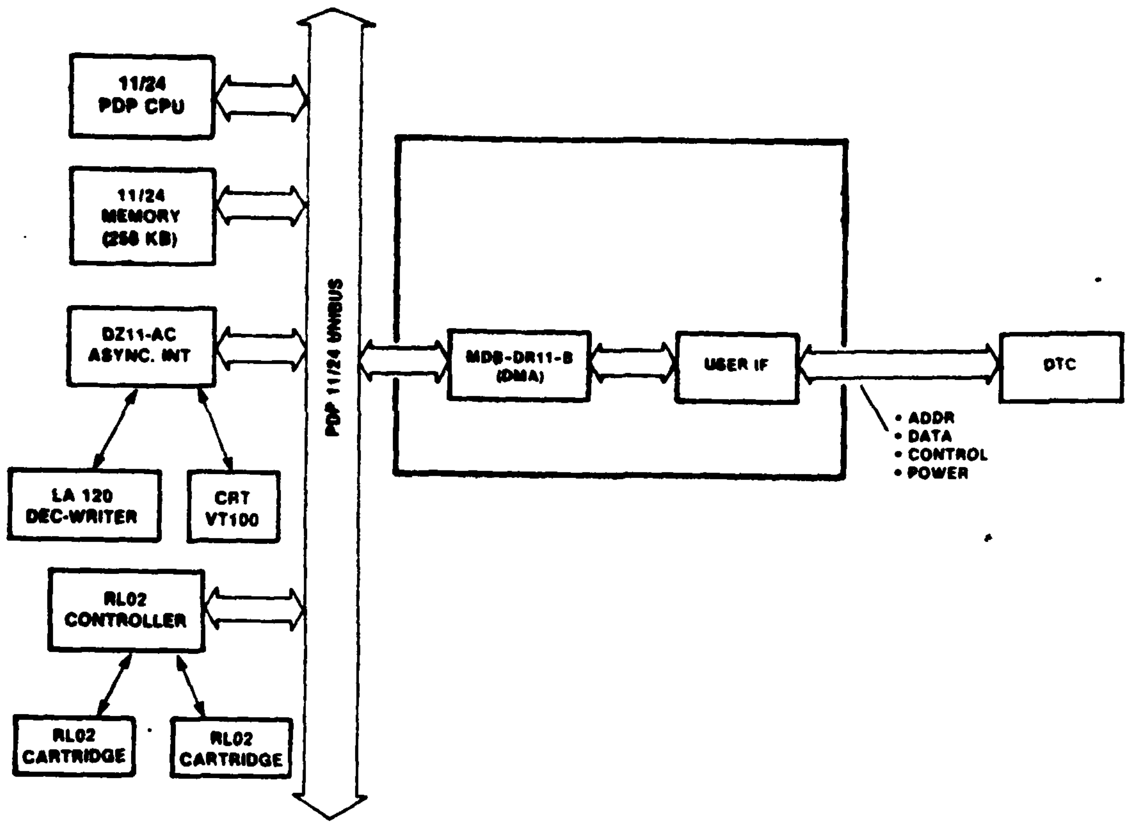
The Data Transfer Units is developed with current Air Force requirements and standards. The primary standards pertinent to the DTS are MIL-STD-1553, MIL-STD-1589, and MIL-STD-1750. The 1553 Serial Multiplexed Data Bus is implemented in the bus interface unit. Either 1553A or 1553B may be selected by a jumper on the external signal connector. The two protocols as implemented in the DTU are different only by the valid mode code definitions. Therefore, no changes will be required when retrofitting older aircraft (1553A). The operational flight program (OFP) is currently written in assembly language but will be replaced by a Jovial/J73 (MIL-STD-1589B) version over the next three months. The assembly language OFP will be converted to IEEE standard mnemonics to facilitate 1750A conversion as well as making the assembly language routines needed for the J73 OFP compatible with the J73 assembler and linker. The internal architecture of the DTU is designed to be compatible with Z8002 or F9450 microprocessors. The DTU's 1750A microprocessor card is designed to be form, fit, and functionally equivalent to the present Z8002 microprocessor card in the DTU. No electrical or physical changes to the DTU will be required for the interchange of cards. Delivery samples of the Fairchild Semiconductor 1750A microprocessor chip (F9450) is expected in first quarter 1983. The 1750A micorporcessor board in the F-16

DTU will meet the requirements of a MIL-STD-1750A processor. The 1750A microprocessor board provides the following functions required for compatibility with MIL-STD-1750A: 1750A instruction set and functional architecture, trigger-go timer, timer A, timer B, fault register, pending interrupt register, interrupt mask register, status word, exhaustive decode of memory and I/O address for the detection of illegal addresses, and detection of incorrect parity during memory and I/O transactions. The 1750A microprocessor board also provides address and data demultiplexing, I/O strobe generation, wait state insertion, DTC decoding, control signal translation, and segment number latching necessary for compatibility with the Z8002 microprocessor card. The prescaler for timer A and timer B, the trigger-go timer, the illegal address detection, and the parity generation and checking are implemented in circuitry external to the Fairchild Semiconductor F9450 CPU which implements the remainder of the functions required by MIL-STD-1750A.

GROUND SUPPO'

G und Support Equipment (GSE) provides three primary functions: pre-flight DTE support, post-flight DTE support, and DTE testing and fault isolation. Prior to a mission, initial mission data must be organized, formatted, and entered into the DTC. This information depends on the aircraft mission but includes data for such systems as the inertial navigation unit, communications, stores management and navigation waypoints. Space is also allocated for files that will be generated during the mission. All checksums as well as all other DTC format linkages are provided by the GSE DTC formatting programs. When the DTC is returned to the GSE after a flight, the files that have been created during the mission can be analyzed. Data files such as new target positions can be used to generate pre-flight data files for subsequent missions. Maintenance data files can be examined by the appropriate personnel on the GSE without tying up the aircraft, thereby conserving fuel and allowing the plane to be configured for its next flight. DTC testing is normally performed both before each use to ensure that the memories are properly functioning and that the battery has sufficient energy to sustain the data between the time it is loaded and when the information is retrieved after its mission.

Two versions of the Ground Support Equipment are available: one implemented using a Digital Equipment Corporation PDP 11/24 (see Figure 7, F-16 GSE), and the other using a portable desktop computer (see Figure 8, Desktop GSE). Both systems provide full support capabilities for the DTC and both have RS232C interfaces for host computer connections where mission planning and analysis can be performed. Also, both units use menu displays to simplify data entry. The PDP 11/24 system also contains provisions for full test and fault isolation for the Data



- END TO END COMPATIBLE SYSTEM OPERATION
- INITIALIZATION OF THE DTC
- ENTER MISSION DATA INTO DATA TRANSFER CARTRIDGE (DTC)
- POSITIVE VERIFICATION OF STORED DTC MISSION DATA
- VERIFICATION OF BATTERY/MEMORY DATA RETENTION
- INTERACTIVE ACCESS AND CONTROL OF DTC DIRECTORY, FILES AND DATA
- CONVERSION OF DATA FROM ENGINEERING UNITS TO APPROPRIATE BINARY FORMATS AND LOCATION IN FILES
- CONVERSION OF DTC FORMATTED DATA TO ENGINEERING UNITS FOR ANALYSIS
- OPERATIONAL SUPPORT SYSTEM PROGRAM (OSSP) CAN BE MODIFIED TO MEET VARIOUS DATA TRANSFER SYSTEMS REQUIREMENTS
- MONITOR DTC BATTERY FOR END OF LIFE
- PROGRAM TWO DTC'S SIMULTANEOUSLY

Figure 7. F-16 Ground Support Equipment (GSE)

Transfer Unit for shop-level support. The portable desktop computer GSE has been delivered to Sperry Flight Systems on the Army Helicopter Improvement Program (AHIP).

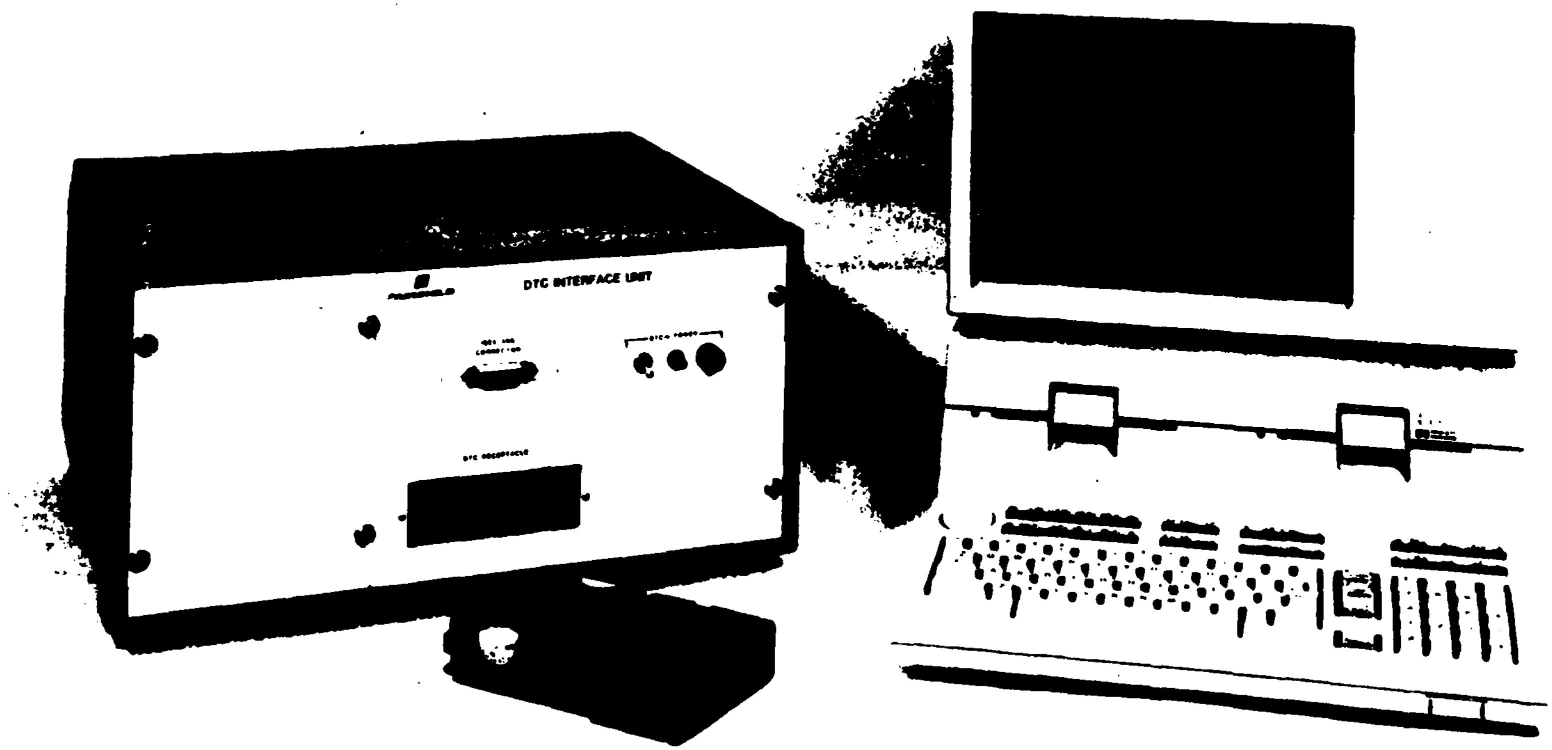


Figure 8. Desktop DTC GSE

SUMMARY

Fairchild's Data Transfer Equipment (DTE) is a general purpose solid state memory cartridge and memory management system providing avionic computers and subsystems real time access to mission planning data. A multimode file access system, under microprocessor control, performs all file management functions, thereby reducing user overhead processing and simplifying user interface.

This system, in conjunction with a ground-based computer, provides an end-to-end capability to accurately load pre-flight mission data, maintain in-flight memory access recordings, and perform post-flight analysis. The system features high reliability, real time data exchanges with the 1553 data bus, flexible file access system, and a high density/capacity solid state nonvolatile memory cartridge.

Data Transfer Equipment features include:

- Compact high capacity real time memory access system.
- Portable nonvolatile memory cartridge expandable from 8K words to 128K words with present memory technology (CMOS).
- Memory growth to 2M words provided in hardware and software for future high density memories.
- Integrity of data storage and transfers enhanced with high degree of error checking/detection throughout system.
- Rugged construction designed for expected handling environment.

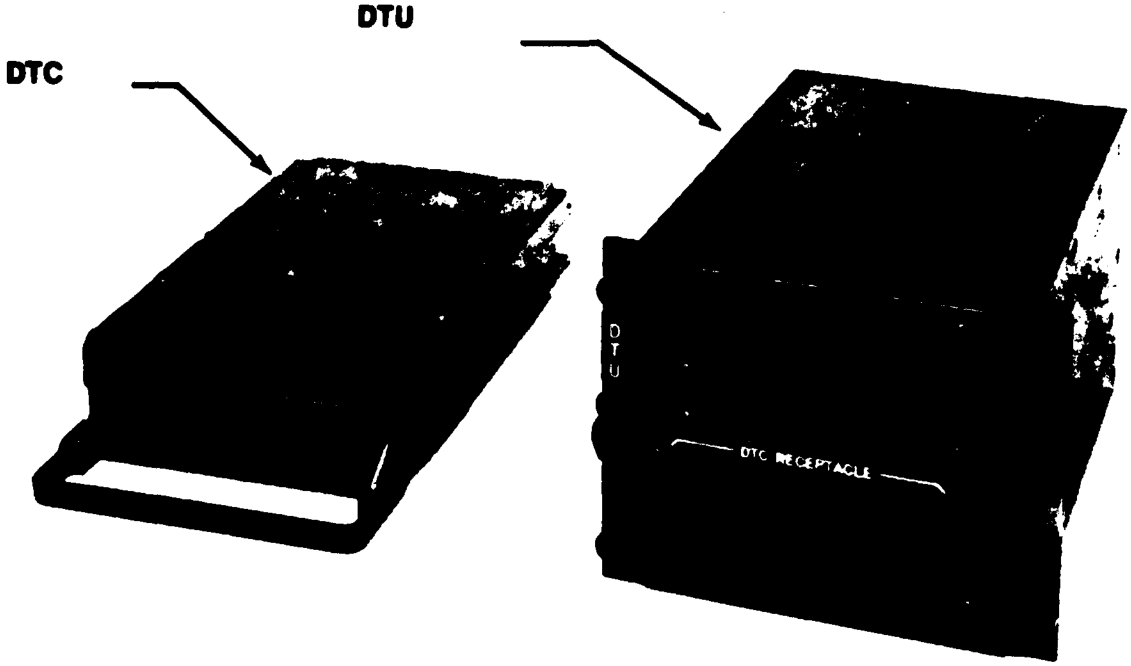
- Implementation with current MIL standards:
  - MIL-STD-1553 A/B Multiplex Data Bus
  - Z8002/MIL-STD-1750A Microprocessor
  - MIL-STD-1589B Jovial (J73) Higher Order Language Software
- EEPROM in DTU memory permits reprogramming of DTOP software through box connector.
- Multimode file management system.

Fairchild's employment of military standards enhances commonality and compatibility with the latest avionic systems, improves maintenance, and reduces unit life cycle costs.

F-16 DTU Hardware Pictorials and Drawings are enclosed.

DATA TRANSFER FLIGHT HARDWARE

DATA TRANSFER EQUIPMENT



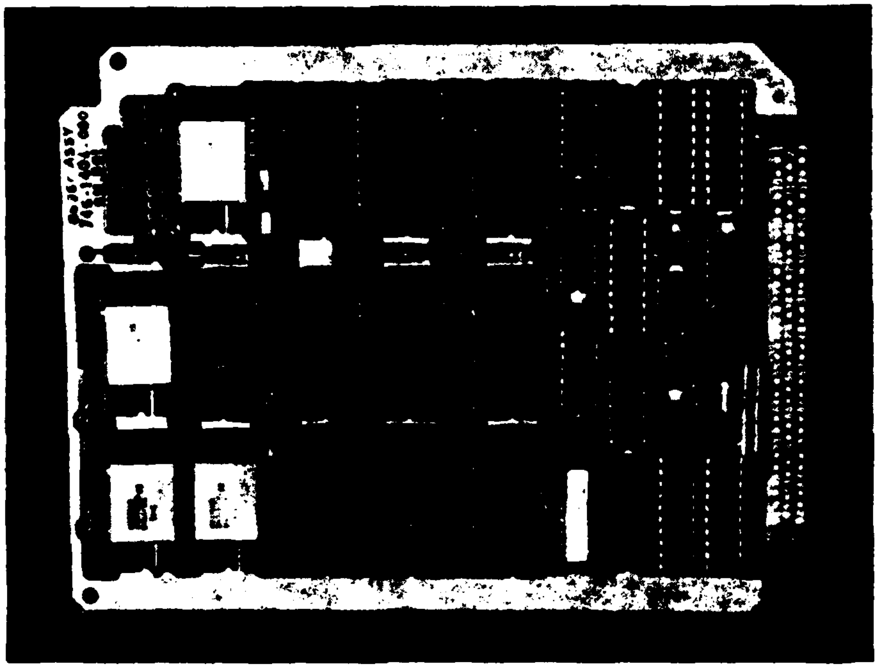
DTU PHYSICAL CHARACTERISTICS

SIZE	5.75" WIDE x 4.5" HIGH x 7.0" LONG
WEIGHT	6.25 LBS.
CONSTRUCTION	ALUMINUM ALLOW INVESTMENT CASTING
POWER	26 WATTS WITH Z8000 PROCESSOR 30 WATTS WITH 1750A PROCESSOR

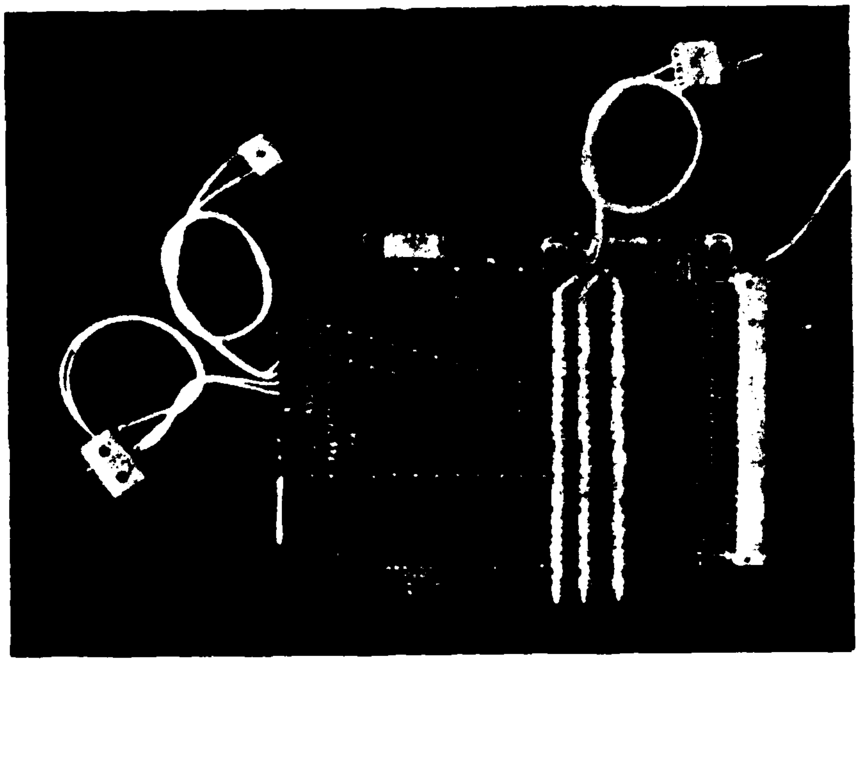
DTC PHYSICAL CHARACTERISTICS

SIZE	1.625" HIGH x 4.72" WIDE x 7.5" LONG
WEIGHT	1.5 LBS. MAX.
CONSTRUCTION	ALUMINUM ALLOY INVESTMENT CAST HOUSING & COVER
INTERFACE	LOW FORCE CONNECTOR LOCKING PINS/HANDLE ALIGNMENT PINS

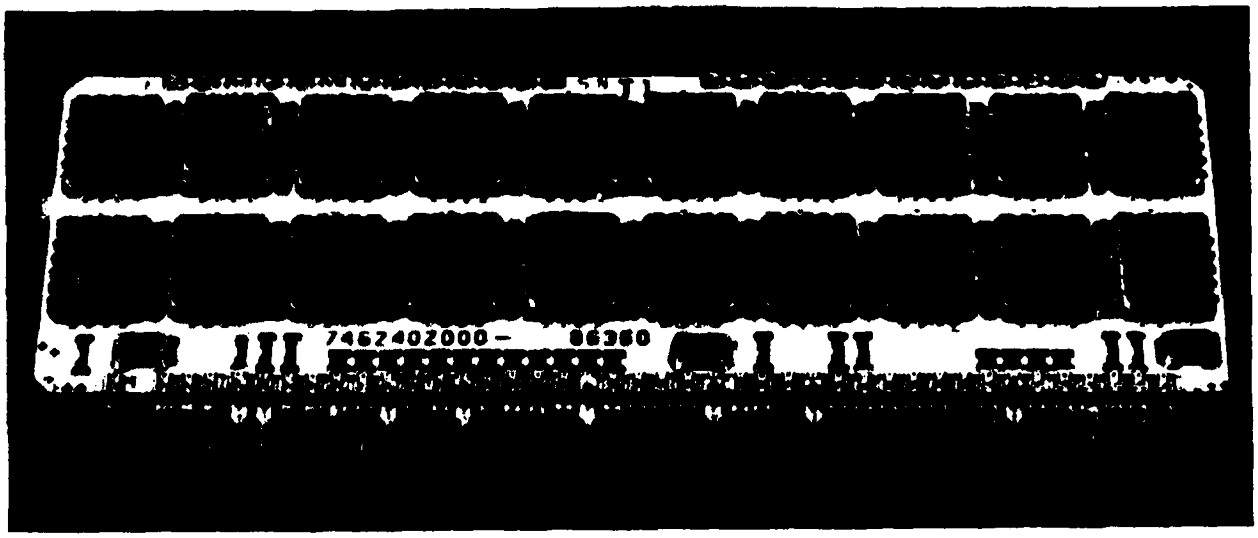
F-16 DTU MEMORY CARD



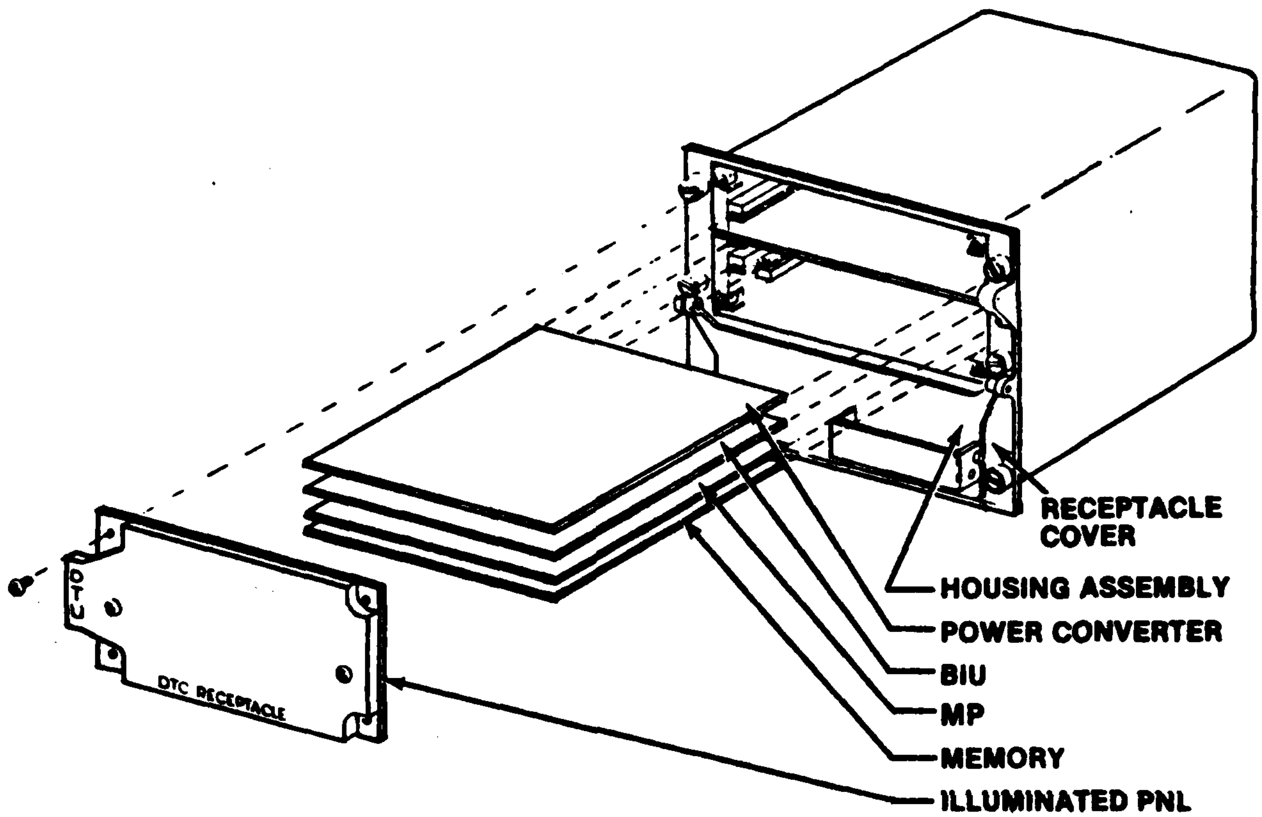
F-16 DTC CARRIER MOTHER BOARD



F-16 MEMORY CARRIER

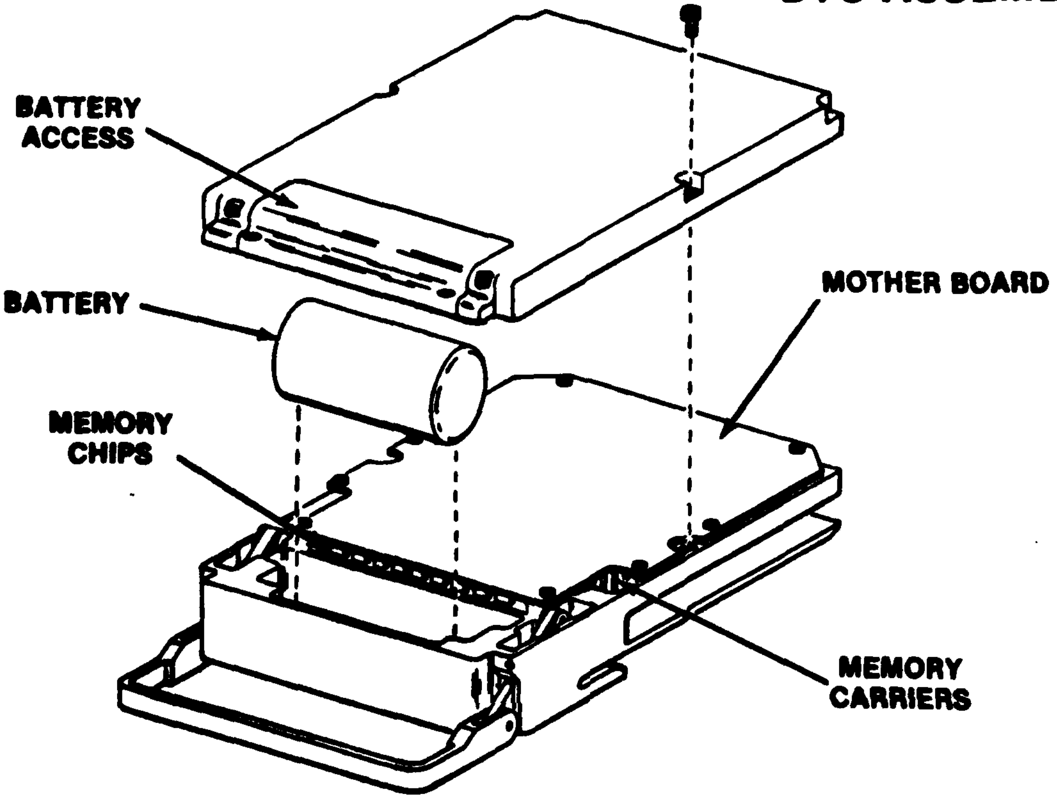


DTU ASSEMBLY



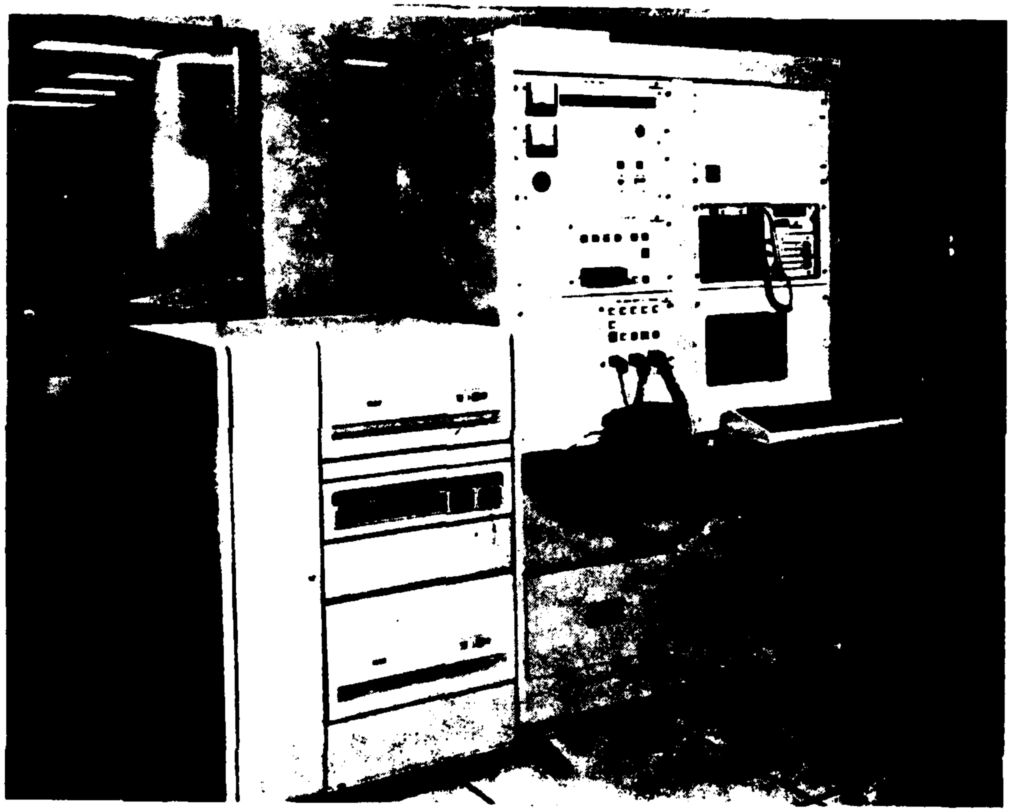
- DESIGNED FOR MAINTAINABILITY AND ASSEMBLY
- DESIGNED FOR ONE-HAND INSERTION/EXTRACTION OF DTC
- INVESTMENT CASTING FOR RUGGED CONSTRUCTION
- FLEXTAPE MOTHERBOARD INTERCONNECT FOR ELECTRICAL CONTROL OF INTERFACE
- ILLUMINATED PANEL (REMOVABLE WITH FOUR SCREWS) PROVIDES ACCESS TO ELECTRONICS
- SPRING-LOADED COVER REMAINS OPEN FOR INSERTION OF DTC AND CLOSES UPON EXTRACTION OF DTC
- POWER CONVERTER CARD AND ELECTRONIC PRINTED CIRCUIT CARDS ALL PLUGGABLE FROM FRONT OF THE UNIT
  - BIU — BUS INTERFACE CARD
  - MP — MICRO PROCESSOR CARD (Z8002/1750A)
  - MEMORY — PROGRAM MEMORY FOR DTOP SOFTWARE AND BUFFER RAM
  - POWER CONVERTER — 110 VAC, 400 Hz TO DC CONVERTER

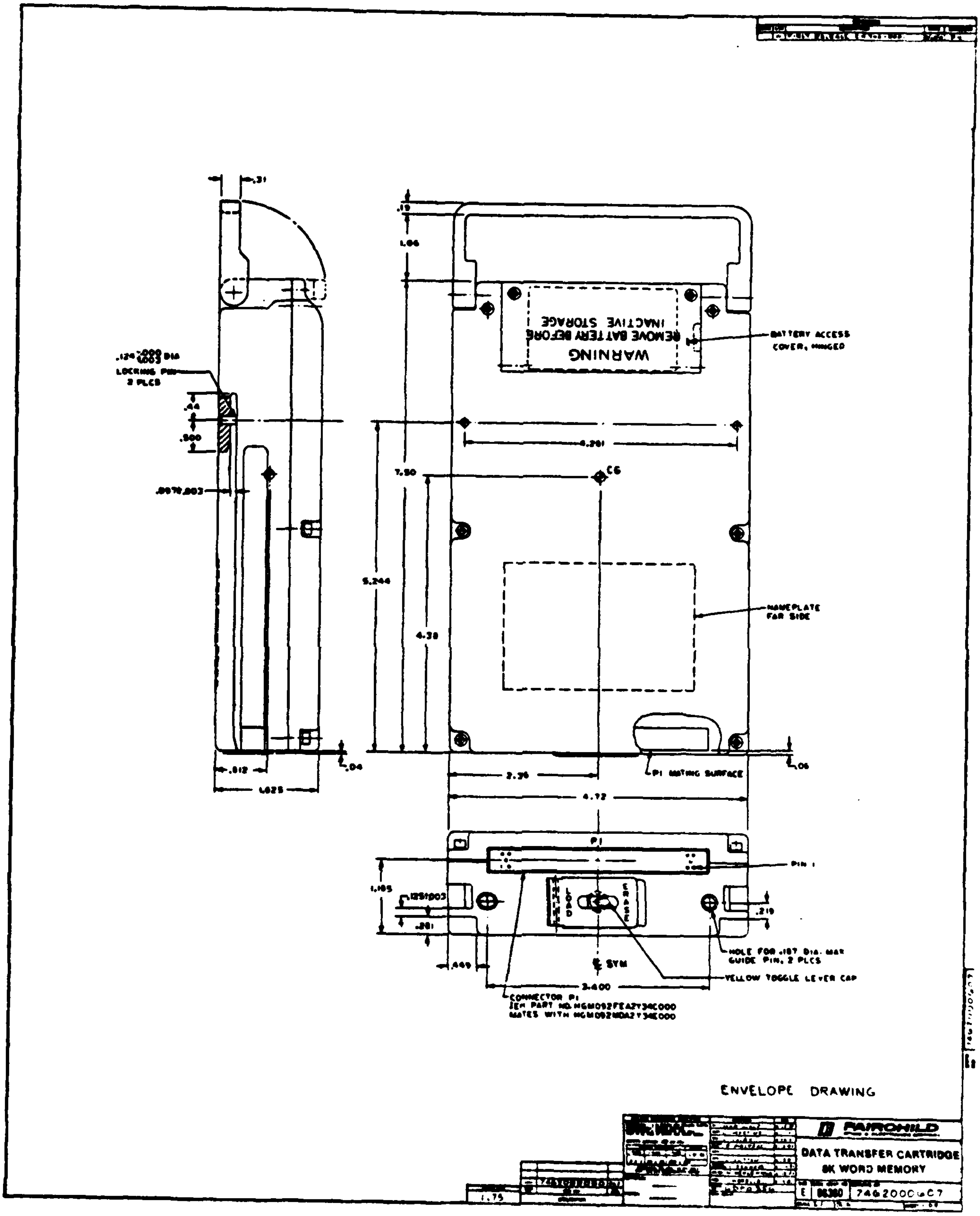
DTC ASSEMBLY

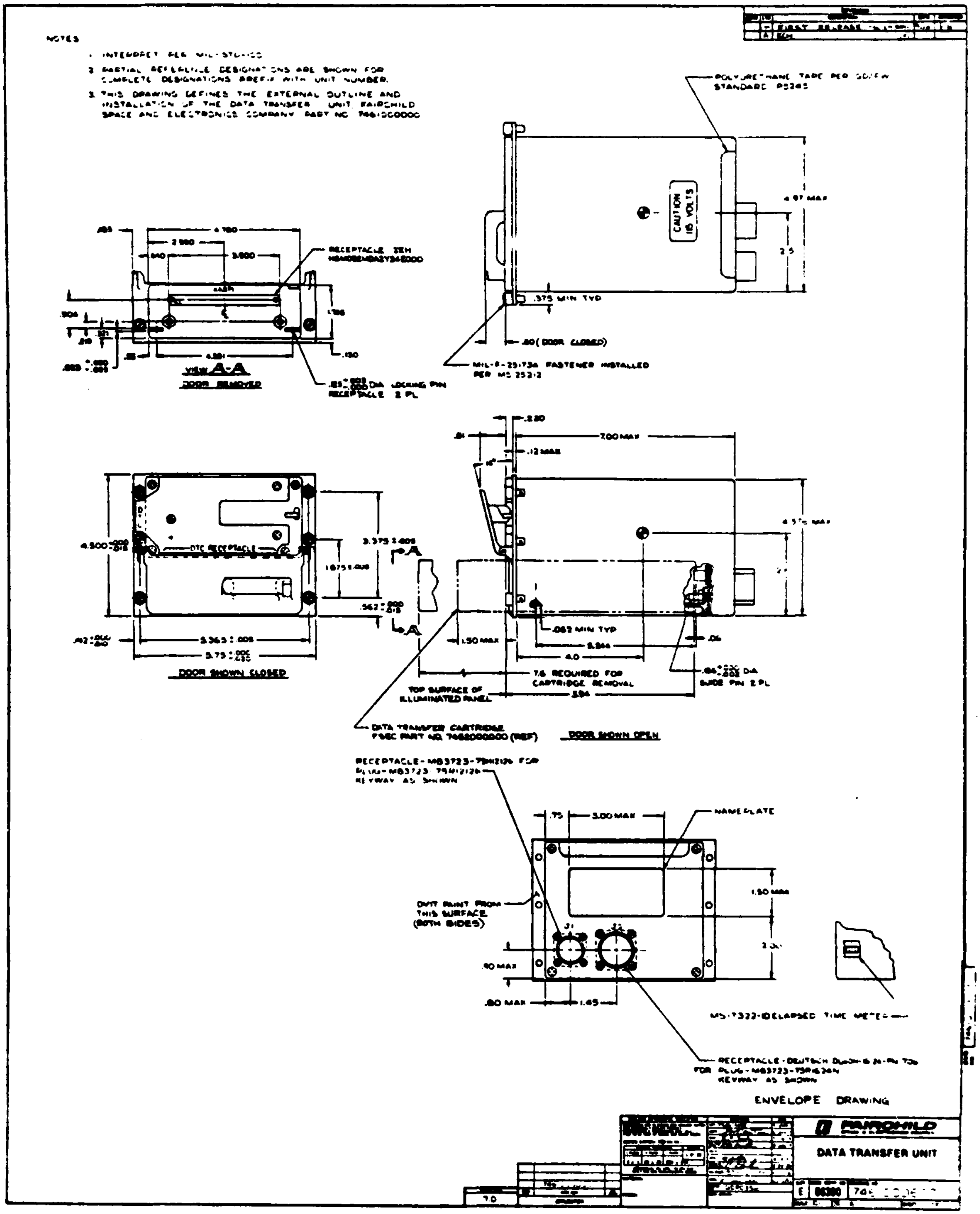


- 8K WORD BASELINE
- EXPANDABLE TO 128K WORDS WITH CURRENT CHIP MEMORY DENSITY - CMOS LCC
- MOTHERBOARD COMPATIBLE WITH 256K WORDS WITH DOUBLE DENSITY MEMORY CARRIERS
- MOTHERBOARD COMPATIBLE WITH 2M WORDS WITH 8X DENSITY MEMORY CARRIERS
- LOCAL AND AUXILIARY ERASE CAPABILITY
- BATTERY BACKED DATA RETENTION
- FAIL-SAFE INSERTION PROCEDURE FOR DTE POWER ON
- PROGRAMMABLE WAIT STATE OUTPUTS TO DTU TO ACCOMMODATE RANGE OF MEMORY TECHNOLOGY
- POSITIVE LOCKING MECHANISM PROVIDES INDICATION TO PROCESSOR THAT CARTRIDGE IS INSERTED AND LOCKED

F-16
GROUND SUPPORT EQUIPMENT




WARNING
REMOVE BATTERY BEFORE
INACTIVE STORAGE
BATTERY ACCESS COVER, HINGED
NAMEPLATE FAR SIDE
P1 MATING SURFACE
CG
P1
PIN 1
LOAD
ERASE
HOLE FOR .187 DIA. MAX GUIDE PIN, 2 PLCS
YELLOW TOGGLE LEVER CAP
℄ SYM
CONNECTOR P1
LOCKING PIN 2 PLCS
ENVELOPE DRAWING
FAIRCHILD
DATA TRANSFER CARTRIDGE
8K WORD MEMORY


NOTES
VIEW A-A
DOOR REMOVED
DOOR SHOWN CLOSED
DOOR SHOWN OPEN
CAUTION 115 VOLTS
DTC RECEPTACLE
NAMEPLATE
ENVELOPE DRAWING
FAIRCHILD
DATA TRANSFER UNIT

STEPHEN L. BELECHAK-BECRAFT

Mr. Belechak-Becraft is presently Manager of Avionic Products Development at Fairchild Space and Electronics Company. His present duties include technical management of Avionic IR&D programs as well as new Data Transfer System applications. He is also serving as Technical Advisor to the F-16 DTS program for MIL-STD-1750A processor application.

Mr. Belechak-Becraft has ten years of experience in military avionics development projects, from initial system concept through production integration and test. Prior to joining Fairchild, he was supervisor of an avionics digital signal processor design and development team. Program assignments have included: F-16 Airborne Radar, F-16 Data Transfer System, Divisional Air Defense Radar, Standoff Target Acquisition System and the Advanced F-16 Radar Programmable Signal Processor.

EDUCATION

Mr. Belechak-Becraft holds the Bachelor of Science in Electrical Engineering from University of Maryland, and the Master of Science in Computer Science from Johns Hopkins University.

ACADEMIC AND PROFESSIONAL SOCIETIES

ETA KAPPA NU
MIL-STD-1750A User's Group
American Management Associations

GEORGE D. FARMER

Currently, Mr. Farmer is responsible for software related issues for Advanced Avionics Development. In this role, he has been involved in a number of proposals as well as the development of IR&D systems. Previously, Mr. Farmer served as lead software engineer for the Data Transfer Unit (DTU) for the General Dynamics F-16 aircraft responsible for both the Operational Flight Program as well as the test equipment software developments. Previously, he developed the software for fault detection/isolation and a network status display package for a fault tolerant processor for use on the NASA Digital fly-by-wire F-8 test aircraft. A study was also conducted to utilize a similar instruction-synchronous fault tolerant processor as a Command Augmentation System in the Fairchild A-10A aircraft.

While working for MIT Lincoln Laboratory, he developed a Z-80 based cockpit information display system under the FAA DABS/Datalink program. This system used aircraft surveillance data supplied by the ground DABS computer to track nearby aircraft and provided collision avoidance via a graphic representation of the encounter on a color CRT.

EDUCATION

Mr. Farmer received his BS degree in Aeronautics and Astronautics/Guidance and Control from Massachusetts Institute of Technology in January 1980.

THE FAIRCHILD MIL-STD-1553 SERIAL MULTIPLEX BUS TESTER

Alan M. Dunn
Fairchild Space & Electronics Co.
Germantown, MD

ABSTRACT

The Fairchild Serial Multiplex Bus Tester (SMBT) is a general purpose computer controlled MIL-STD-1553 stimulus/response/analyzer module designed for use in maintenance and production test systems. Moreover, SMBT has been selected to fulfill the data bus testing requirements of the Air Force Modular Automatic Test Equipment (MATE) systems. The SMBT integrates Monitor, Controller, and Remote Terminal functions into a single unit, which can be rack mounted and remotely controlled via a IEEE-488/ATLAS compatible protocol. Modular design in hardware and software allow the SMBT to be modified to customer requirements. The design enables the SMBT to be controlled and synchronized by a master computer performing a coordinated total checkout of a unit under test.

Conceptually, the SMBT may be thought of as three separate devices or functions: a monitor, a bus controller, and a remote terminal simulator. These functions may be operated in any combination. They are programmed via the IEEE-488 interface with a protocol that is semantically compatible with ATLAS-type test scenarios. The monitor allows the ATE system to collect 1024 word "windows" of bus traffic and simultaneously accumulate error statistics on all bus traffic. Trigger modes may be programmed which aid in the positioning of the collected "window" in relation to selected bus events.

The bus controller allows the ATE system to generate a major frame of up to 128 unique messages. The gaps between these messages may be programmed. Also, various word and message errors may be injected into each message of the major frame.

The remote terminal simulator allows the ATE system to simulate up to 31 remote terminals. As many as 128 different RT responses may be programmed. The response time is programmable and, as with the bus controller, various word and response errors may be injected into each response.

Architecturally, the SMBT is divided into a real-time "front end" and a secondary "back end". The back end is a Z8000-based microcomputer which is responsible for managing the IEEE-488 interface and for setting up the front end. The front end is a sequencer-driven design which handles monitor, bus controller, and remote terminal functions in real-time. The interface between the front end and the back end is several dual-port memory boards which are easily expandable. Operationally the back end accepts test scenarios, translates them into a special format which is loaded into the dual-port memory, and commands the front end to start. The front end then begins operating per the specially formatted scenarios. When monitor information is required, the back end pulls it directly out of the dual-port memory and transmits it back to the host computer over the IEEE-488 interface.

In summary, the SMBT is designed to provide necessary stimulus/response/analysis capability for MIL-STD-1553 Line Replaceable Unit (LRU) testing. Furthermore, the SMBT is designed to provide this capability in a manner that is consistent with ATLAS-oriented ATE systems. The SMBT, with its substantial baseline capability, compatibility with existing standards, and its modular, expandable architecture, is targeted specifically for "off-the-shelf" ATE systems.

INTRODUCTION

The SMBT shown in Figure 1 is designed to perform extensive testing of a MIL-STD-1553 type multiplex bus system. The three major operational modes of the SMBT are simulating the bus controller, simulating up to 31 remote terminals, and monitoring all transactions on the bus. The bus controller is capable of emulating any bus controller with the added task of selected error injection. The remote terminal simulator is able to emulate up to 31 remote terminals concurrently while also injecting selected errors. The monitor verifies all transactions on the multiplex bus, recording all errors detected and providing for triggering capabilities to trap certain selected bus traffic information. In addition to each of the above, the SMBT is able to operate with all or any combination of the above modes concurrently.

There are also two non-operational modes of the SMBT. These are the setup mode and built-in-test (BIT). The setup mode is the period during which the parameters for the operational modes are chosen. Bus controller commands, remote terminal responses, monitoring functions, triggering modes, and error injection are but a few of the parameters which must be determined. BIT can be initiated by the host computer via the system control port (IEEE 488 interface). BIT will verify the operational status of the SMBT to a high degree of confidence and, if a fault is detected, it will isolate the fault to its associated printed circuit card.

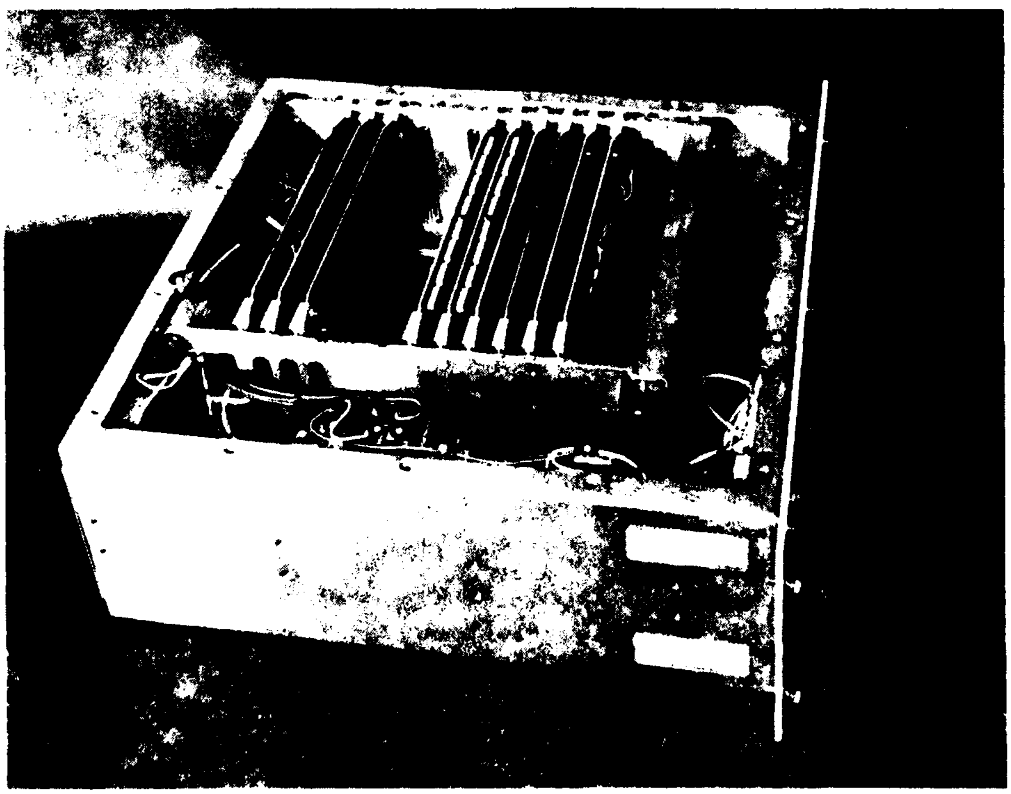


Figure 1. Serial Multiplex Bus Tester

BUS CONTROLLER SIMULATOR

The bus controller for a MIL-STD-1553 type bus system initates all messages on the bus. It transmits messages on a frame basis where the frame is broken up into equally spaced subframes. Each subframe may contain a unique set of messages. The SMBT is capable of transmitting error free messages in this format, or it may be commanded to inject selected, timing, protocol, or electrical errors into the transmission. The bus controller of the SMBT will be discussed in two parts. First, the normal bus controller functions will be explained, followed by the error injection capabilities of the bus controller.

The bus controller of the SMBT is able to initiate normal bus traffic as stated in MIL-STD-1553 (A or B). The basic format for a frame is shown in Figure 2. Each frame is divided into equally spaced subframes. Within each subframe is a uniquely specified set of messages. The content of each message within the subframe may be uniquely specified from all others.

The normal operation of the SMBT bus controller is to output a frame of commands and then to repeat this frame continuously until directed otherwise. There are only two ways to alter this normal pattern, one is by adaptive polling and the other is to program a one-shot frame.

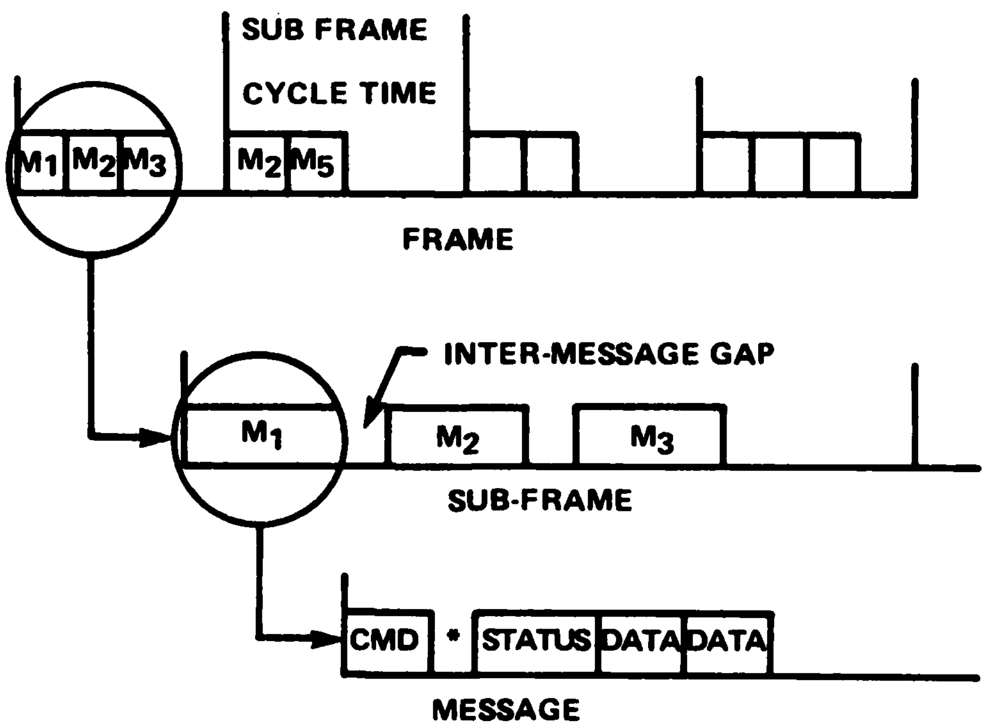


Figure 2. Frame Sub-Frame and Message Format

The adaptive polling feature allows branching from normal sequencing when a particular status word is received in response to a particular command. Adaptive polling is enabled on a command by command basis. An adaptive polling word is compared to the associated status word on a bit by bit basis. If a match is not found, normal sequencing ensues. Upon a match, an adaptive polling vector determines the new bus controller path to follow.

The one-shot frame is a subset of normal sequencing. The bus controller may be commanded to stop transmissions after a user selected number of frames has been issued.

Subframes within a frame are all equally spaced. The two programmable aspects of the subframe are the number of subframes per frame and the length of a subframe. The number of subframes within a frame can be from one to a upper limit equal to the number of messages in a frame. The number of subframes within a frame can also change during bus controller operations. Since an adaptive polling acceptance vectors the bus controller to a different sequence and an unlimited number of adaptive polling sequences can occur, the length of a frame can vary widely during operation. The length of a subframe can be varied by the user, but once chosen will remain constant for all subframes within a transmission. The subframe interval may be varied from 40 microseconds to 650 seconds as follows:

- Ten microsecs to 64 millisecs in one microsec increments.
- Ten microsecs to 650 millisecs in ten microsec increments.
- One hundred microsecs to 6.5 sec in 100 microsec increments.
- One millisecs to 65 secs in one millisec increments.
- Ten millisecs to 650 secs in ten millisec increments.

The number of messages and their contents can vary from subframe to subframe. The command in a message can be uniquely specified from all other commands in the frame. The programmable aspects of a message are:

- Command Type
- Specific Command Word(s)
- Specific Data Word(s), if any
- Bus Used
- Inter-Message Gap

There are three command types, bus controller to remote terminal (BC-RT), remote terminal to bus controller (RT-BC) and remote terminal to remote terminal (RT-RT). BC-RT and RT-BC commands both contain one command word, RT-RT contains two command words. Under normal conditions only the BC-RT command has associated data word(s) transmitted by the bus controller.

Each command word can be specified with different address, subaddress, word count, and transmit-receive fields as specified by MIL-STD-1553.

After the command has been issued, the data must be specified, if required. The set-up sequence of the SMBT specifies the desired data patterns to be transmitted with the command. Each bit of every data word can be uniquely determined for every command. But, since the SMBT hardware is table driven, two commands with identical data patterns can share the same data table.

This allows for less memory to specify a given frame providing for either a memory savings or increased message quantity for a given memory size. As another user option, a random data table can be chosen. The data from this table is generated from a pseudo-random number generator.

In addition to specifying the bit patterns of the command and data words, the bus on which the message is to be transmitted can be specified on a per message basis. The spacing between each message within a subframe can vary. This time, called the inter message gap, can be programmed from 4 microsecs to 32K microsecs in 0.5 microsec increments. The accuracy of this time will be -0.1 to +0.2 microsecs.

ERROR INJECTION

The bus controller of the SMBT not only acts as a versatile bus controller, but it is also able to inject errors within its transmission to verify remote terminal(s) operation. The SMBT bus controller can inject a wide variety of errors, on a bit, word, or message basis. The following errors can be generated:

- Manchester error
- Inverted sync
- Parity
- Bit Count
- Discontiguous Data
- Word Count
- Amplitude Deviation
- Simultaneous transmission

A brief description of each of these errors follows:

MANCHESTER ERROR - A manchester error is an error in which a bit is transmitted without a zero crossing. The SMBT can inject Manchester errors in bits 1 thru 15 of any command or data word. As many words as desired may contain a manchester error.

INVERTED SYNC - An inverted sync error is one where a data sync is transmitted in a command word or a command sync is transmitted in a data word. The SMBT bus controller is able to specify the sync for every word in every message allowing for as many sync errors as desired.

PARITY - Each word transmitted onto the multiplex bus should have odd parity. The SMBT bus controller has the capability to transmit odd or even parity for every word in any message of the frame.

BIT COUNT - The normal bit count of a MIL-STD-1553 type word is 20 bits, three for sync, 16 for data, and one for parity. The SMBT bus controller is able to transmit from 12 bits (three sync, eight data, one parity), to 27 bits (three sync, 23 data, one parity). The bit count is programmable in every word of every message the bus controller transmits.

DISCONTIGUOUS DATA - All data words transmitted by a bus controller should be preceded by either a command word or another data word with no inter-word gap. The second command word of an RT-RT transmission should also follow the first command word with no inter-word gap. The SMBT bus controller is capable of inserting an inter-word gap in each of these cases. The gap is programmable in any word of any message. The limits of the gap are from 0.5 microsecs to 7.5 microsecs in 0.5 microsec increments.

WORD COUNT - The number of data words transmitted in a BC-RT command should be equal to the word count field in the command word. The SMBT is able to select the number of words to be transmitted from none to 255 words. This capability also applies to RT-BC and RT-RT commands. The data is sent contiguously unless a gap is specified by the discontiguous data error.

AMPLITUDE DEVIATION - The amplitude of the output voltage of the SMBT onto the 1553 data bus is programmable on a message basis. The range is programmable from 250 millivolts to 6.5 volts peak to peak, line to line. The output voltage of the SMBT is adjustable in 50 millivolt increments. The worst case slew rate from one level to another is less than 20 microseconds.

SIMULTANEOUS TRANSMISSION - A normally operating bus controller in a dual redundant system will not transmit on both buses simultaneously. The SMBT bus controller is able to specify two independent commands, each to be transmitted on a different bus. The two commands may be programmed to begin concurrently, or the start of the second command may be delayed from 0.5 to 63 microseconds beyond the start of the first.

REMOTE TERMINAL SIMULATOR

The remote terminal(s) simulator for the SMBT is capable of emulating up to 32 remote terminals simultaneously. The SMBT is able to simulate the bus traffic on a given dual redundant multiplex bus system or any part thereof. In addition to the remote terminal(s) emulation, the SMBT remote terminal has the capability to inject errors in the response. The requirements of the SMBT remote terminal(s) simulator will be discussed in two parts, normal simulation and error injection.

REMOTE TERMINAL FUNCTIONS

The command word of a 1553 type message is broken up into four fields; the address, the transmit/receive (T/R) bit, the subaddress, and the word count/mode field. The address specifies the remote terminal to which this command is directed. There can be up to 32 different remote terminal addresses. The T/R bit specifies if the remote terminal is to accept or transmit information over the bus. The sub-address field provides a means to specify up to 32 different tasks for the remote terminal to perform. One of these tasks can be to use the word count/mode code field as a mode code which specifies another possible 32 tasks (usually front end related) that can be performed. The word count/mode code field specifies the number of data words to be transferred for this command when in the non-mode code function. For a a more detailed explanation refer to the MIL-STD-1553 Specification.

The SMBT remote terminal is able to emulate the above terminal. It monitors for the selected addresses then, when a match occurs, the remaining fields are decoded to determine the action required to take place. The SMBT creates a vector address by combining the address field, the T/R bit, and either the subaddress field or mode code field, whichever is applicable. This eleven bit quantity is used as an address pointer to a command-status response table. This table contains information about the transmission of the status words and the transfer of all data words.

The capabilities of the SMBT remote terminal(s) simulator is as follows. The SMBT is capable of emulating up to 32 remote terminals at the same time. The status response returned on the bus is uniquely determinable for every address, T/R bit, and subaddress or mode code combination in the command word. The data associated with each status is also uniquely determinable for every command word. However, to conserve memory space, status and data responses may be shared by multiple commands. The SMBT configuration for the MATE program allows for 128 unique status responses and up to 8 unique data tables. The response time of the status word is programmable in 0.5 microsecond steps from 4.0 microseconds to 127 microseconds, and is uniquely programmable for each status response.

The data table requirements are identical to those of the SMBT bus controller. Each status response points to its desired data table. The data tables specify the data pattern desired for the particular status response. Random data is also provided. Each data table can be shared by multiple status responses to conserve memory. Incidentally, since these tables are specified identically to the bus controller data tables, the bus controller and remote terminal simulators can share data tables to provide for additional memory savings. Under normal, error free operating scenarios, the SMBT will transmit the status response on the bus from which it received the command.

REMOTE TERMINAL ERROR INJECTION

The SMBT remote terminal(s) simulator has the capability to inject errors in its transmissions much as the bus controller does. The errors which can occur in a status response area are as follows:

- Manchester Error
- Inverted Sync
- Parity
- Bit Count
- Discontiguous Data
- Response on wrong bus
- Word Count
- Response Time

MANCHESTER ERROR - As in the controller mode, the SMBT can transmit any word in the status response with a Manchester error. A Manchester error is an error in which a bit is transmitted without a zero crossing. The SMBT can inject Manchester errors in bits 1-15 of any status or data word. As many words as desired may contain Manchester errors.

WORD COUNT - The number of data words transferred in a non-mode code response should equal the value in the word count field of the command word. The SMBT remote terminal(s) simulator is able to select the number of data words to be sent in relation to the word count field. Both a word count high, or a word count low error may be introduced by the SMBT remote terminal simulator.

RESPONSE TIME - The time that a remote terminal must respond to a valid command is referred to as the response time. MIL-STD-1553 A and B define the maximum and minimum limits of this time. The SMBT is programmable, on a response basis, to provide a response time from 4.0 microseconds to 127 microseconds. The response time is programmed in 0.5 microsecond increments.

INVERTER SYNC - An inverted sync is when a data sync is transmitted in a status word or a command sync is transmitted in a data word. The SMBT remote terminal(s) simulator is able to specify the sync for every word in every response allowing for as many sync errors as desired.

PARITY - Each word transmitted onto the multiplex bus should have odd parity. The SMBT remote terminal(s) simulator has the capability to transmit odd or even parity for every word in all responses.

BIT COUNT - The normal bit count of a MIL-STD-1553 word is 20 bits, three for sync, 16 for data, and one for parity. The SMBT remote terminal simulator is able to transmit from 12 bits (three sync, eight data, 1 parity), to 27 bits (three sync, 23 data, 1 parity). The bit count is programmable in every word of every response transmitted.

DISCONTIGUOUS DATA - All data words transmitted by a remote terminal should be preceded by either a status word or another data word with no inter-word gap. The SMBT remote terminal simulator is capable of inserting an inter-word gap. The gap is programmable in any word of any response. The limits of the gap are from 0.5 microseconds to 7.5 microseconds in 0.5 microsecond increments.

RESPONSE ON WRONG BUS - Under normal operating conditions the SMBT remote terminal simulator will respond with the status word or the bus which transmitted the command. As an error condition, however, the SMBT can be programmed to respond with the status always on bus A, always on bus B, or on the bus opposite the one which transmitted the command.

MONITOR

The SMBT monitor is responsible for transferring all information from the multiplex bus to the system processor. In addition to the data bits of the command, status, and data words, this information includes additional parameters characterizing each word. Information such as word type, gap time, error(s) present, sync polarity, and on which bus the word was received is recorded. The SMBT monitor is also responsible for maintaining various error tables and for detecting and the reporting of triggers. The monitor will be discussed in three parts, the bus traffic table, the statistical tables, and the triggering capabilities.

BUS TRAFFIC TABLE

The bus traffic table contains all received data from the multiplex bus including annotated information. The standard table is 1024 words long and is double buffered. Information gathering takes place in one buffer while the system processor is examining the data in the other buffer. Buffer switching occurs when triggers are detected. The trigger is placed at the middle of the buffer. This provides the trigger and the 500 words before and the 500 words after the trigger in the buffer for examination by the system processor.

In addition to gathering the data information from the bus, the bus traffic table also gathers status information for each word as listed below.

- Gap Time - the gap time from the beginning of this word to the end of the preceding word is recorded. This information provides a means to calculate inter-message gap (last word of previous message to first command word), status response time (last word of command to status word), and word contiguity (first command to second command in RT-RT, status word to data word, and data word to data word). Gap time is recorded with a 160 nanosecond resolution.

- Type of Word - Specifies if this word is a first command word, second command word, status word, or data word of the message.

- Bus ID - This reports the bus associated with the word.

- Errors - A bit is assigned for every error the monitor provides as trigger inputs. A complete list of the errors detected by the SMBT is contained in Table 1.

- Trigger - A bit which indicates if a trigger occurred during the reception of this word.

- Transmit Status - This information is placed in the bus traffic table by the SMBT bus controller or remote terminal(s) simulator. The bit pattern is determined by the user. This feature is provided so that the bus controller or remote terminal(s) simulator has a means to indicate to the monitor what their current status is, allowing some real time monitoring of the transmit logic.

STATISTICAL TABLES

The bus traffic table is a very complete record for the windows of information it gathers on the bus, but it can miss sections of bus traffic while the system processor is busy analyzing data, and transmitting it back to the host computer. The statistical tables provide all information concerning bus activity in selected areas of interest. These tables compile information continuously from the start of the monitor's activity, until directed to stop. The statistical tables generated by the SMBT are the following:

- Terminal Error Count Table
- Status Response Analysis Table
- Bus Activity Table
- Error Time Analysis Table

Each of the above tables is generated simultaneously along with the bus traffic table.

TERMINAL ERROR COUNT TABLE - The terminal error count table is a tabulation of all detected error types vs. all remote terminal addresses. Each entry represents the number of times the particular error type occurred for the particular remote terminal. The range of this number is from 0 to 65,535. When the maximum count is reached, the value will roll over to 0. Table 1 lists all error types detected by the SMBT monitor.

TABLE 1. SMBT Detected Error Types

Invalid Word - parity, Manchester, bit count
Word Count High
Word Count Low
Simultaneous Bus Traffic
Late Response Time
Early Response Time
No Response
Discontiguous Data
Intermessage Gap Time High
Intermessage Gap Time Low
Response on Incorrect Bus
Inverted Sync

STATUS RESPONSE ANALYSIS TABLE - This table is a listing of the number of times each of the status bits was set for each remote terminal address. The status bits include the eleven non-address bits. The range of each entry is 0 to 65,535. When the maximum value is reached for any entry the value will roll over to 0.

BUS ACTIVITY TABLE - The Bus Activity Table lists the number of commands which have been issued to each remote terminal. Separate counts are kept for each bus, as well as a total count on both buses for each remote terminal. The range for each entry is 0 to 65,535 with roll over occurring if the maximum limit is exceeded.

ERROR TIME ANALYSIS TABLE - The Error Time Analysis Table preserves a record of each time a "trigger" occurs in the SMBT monitor. This table contains up to 1024 entries which record the trigger condition, the 1553 bus word, all error bits, and the time of occurrence of the trigger.

TRIGGER

The monitor continually checks for triggers. When a preselected trigger(s) occurs, the monitor will switch buffers in the bus traffic table and inform the system processor that a trigger has occurred. There are five sources of triggers, the monitor error detector, the monitor word detector, the SMBT bus controller, the system processor and an external trigger. Each of these triggers can be masked and combined to form the trigger condition.

ERROR DETECTOR - The monitor detects the errors listed in Table 1. The errors, when detected can be used as triggers. The monitor will apply a trigger mask to the errors to enable only user desired errors. The enabled errors will be ORed together to form the error detector trigger.

WORD DETECTOR - A second source for triggers in the monitor is the word(s) detector. The monitor is able to compare from one to eight words of a particular type to a preprogrammed sequence of words. The types allowed are command words, status words, data words, or all words. The pre-programmed sequence of words will contain a desired bit pattern of ones, zeroes, or don't cares. Each word of the desired type is ANDed with a mask word and then compared to the pre-programmed words. If type consecutive words from the bus compare to the pre-programmed words in bit pattern and order, a trigger is generated.

SMBT CONTROLLER - The SMBT bus controller simulator is capable of generating triggers. One of the bus controller instructions provides for a trigger generation. This allows triggers to be generated at any point within the bus controller routine. In this manner the system processor can be informed in real-time of events requiring its attention.

SYSTEM PROCESSOR - The system processor is able to generate a trigger to the monitor logic. It will also be able to inhibit all triggers to allow freezing of the double buffer bus traffic table.

EXTERNAL TRIGGER - A differential external trigger input is provided to the SMBT to allow synchronization of the SMBT monitor with real-time events. A pulse greater than 100 nanoseconds in length at the external trigger input will trigger the SMBT monitor.

BIOGRAPHY

ALAN M. DUNN

Mr. Dunn is presently Section Manager of Digital Circuit Design at Fairchild Space and Electronics Company, Germantown, Maryland. During his ten year association with Fairchild, he has participated in a variety of avionic and space programs. His primary responsibility is in the design and development of microprocessor based equipment. Mr. Dunn provided the technical leadership for the design and development of the Fairchild Data Bus Monitor Controller, the first commercially available MIL-STD-1553 tester.

As with the DBMC, Mr. Dunn is now providing technical leadership for the MATE/SMBT Program.

EDUCATION

Mr. Dunn holds a Bachelor of Science degree in Electrical Engineering from the University of Maryland and a Master of Science in Computer Science from George Washington University.

SOFTWARE ENGINEERING SERIES IN THE FEDERAL GOVERNMENT

PRESENTOR: GWENDOLYN E. HUNT
ASSOCIATE, SYSTEMS TECHNOLOGY
OFFICE, PACIFIC MISSILE TEST CENTER

ABSTRACT

SOFTWARE ENGINEERING PROJECT
PROFESSIONAL COUNCIL OF FEDERAL SCIENTIST
AND ENGINEERS

The West Coast Region, Professional Council of Federal Scientists and Engineers has been exploring the difficulty encountered by the lack of specific classification and qualification standards in the Federal government for positions in the emerging field of software engineering. This paper provides an account of the efforts to develop the software engineering occupational series. It also describes interim approaches by the Department of Navy, and some of its field activities, to alleviate the problems of recruitment and retention, undefined career paths and salary inconsistencies.

Finally a prognosis is provided of the success of the project, together with the new initiatives which are being studied.

SOFTWARE ENGINEERING SERIES IN THE FEDERAL GOVERNMENT

PROFESSIONAL COUNCIL OF FEDERAL SCIENTISTS AND ENGINEERS
ADVISORS TO THE DIRECTOR, SAN FRANCISCO REGIONAL OFFICE
FRANCIS V. YANAK, DIRECTOR

BACKGROUND

In the mid-sixties, the Department of Defense (DOD) began its trend toward the acquisition of weapon systems embodying digital computers as a primary component of a total system.

These computers performed the functions of data storage, process control, and complex decision making. In order to perform these functions, the computers required a set of instructions and data which was narrowly called software. The continued miniaturization of computer hardware through gigantic technological advances forced an acceleration of this trend in the seventies. Simultaneously, decreased national productivity, an unstable economy, energy shortfalls, and the increase in Federally supported social services fostered the proliferation of computers in the non-defense sectors of the Federal government.

In both environs, the complexity of the development, operation, support and management of these computer-based systems spawned a community of government employees whose required knowledges, skills and abilities (KSA's) transcended traditional academic disciplines such as mathematics, engineering, physics, and the like. These required KSA's have been described in subject matter literature since the late sixties as "Software Engineering", with the unique individual processing these KSA's being known as a "Software Engineer." A recent software engineering text provides the following:

"Among the many definitions of software engineering proposed since 1970, the most accurate and descriptive was by F.L. Bauer of the Technical University, Munich, Germany, in 1972. His definition can be stated:

> The establishment and use of sound Engineering Principals (methods) in order to obtain economically software that is reliable and works on real machines

This definition of software engineering encompasses the keywords that are the heart of all engineering discipline definitions: sound engineering principles, economical, reliable, and functional (works on real machines)." (2:9)

To paraphrase, Software Engineering is the application of knowledge of mathematical and physical sciences acquired by special education, training and experience to the various aspects of software system design, development, and management essential to insure effective, efficient and economic utilization of computer system resources.

The Software Engineer is responsible for various aspects of software system design, development, and management essential to insure effective utilization of computer system resources as elements of major physical or environmental systems which incorporate one or more specific engineering disciplines. The computer systems are generally embedded and/or integrated within a major system complex and provide direct real-time control of and/or perform specific tasks within one or more of the system functional elements.

While there is an equally important required skill of understanding the computer and its languages, the software engineer must understand the operation, functions, and interfaces of the total system in order to effectively solve complex problems necessary to design algorithms and instructions for the computer, resulting in an economical and reliable system.

Private industry has already recognized this emerging technological area and has established a career field for the software engineer. With the demand far exceeding the supply, industry is also offering premium salaries to applicants. Without a designated software discipline, the Federal service lags behind industry and this compounds the recruiting problems. The software engineering job category must be recognized in the public sector in order to overcome the recruitment and retention obstacles which already include salary disparity within the industry and lack of specific, well-defined career patterns in this field.

The West Coast Regional Council of Professional Scientists and Engineers had for some time been exploring the difficulty caused by the lack of appropriate classification and qualification standards in the Federal government. The Council is composed of senior civilian representatives of the DOD and other Federal civilian activities in the Western Region which employ fifty or more professional employees engaged in research, development, test and evaluation. It was established as an advisory group to the San Francisco Regional Office of Personnel Management (OPM) by the director, Francis V. Yanak. Through its involvement with the many initiatives to upgrade the quality of the Federal technical work force, it was successful in obtaining special pay rates for engineers; first in the Western region, then nationwide. A more recent effort is to obtain approval to extend this special pay status to all scientists.

The software engineering project was officially intiatiated in 1979. The approach was that the Western Regional Office, utilizing field activity resources accessible through its Professional Council of

Scientists and Engineers, undertake the development of a new engineering series for Software Engineers. The first phase of the effort was to conduct an occupational survey resulting in a definition of a new series. Later efforts would be concerned with full classification and qualification standards under the guidance of the Standards Development Center OPM, Washington, D.C.

At the same time, the Department of the Navy System Commands, Material Command and Field Activities were being plagued by the same problem. The Master Plan for Tactical Embedded Computer Resources, a Naval Material Command document states:

> "Weapon systems software acquisition and maintenance are becoming sufficiently important that consideration should be given toward establishing a separate career field for weapon system software engineers. Also, software acquisition and maintenance is sufficiently complex and challenging so that career incentives should be developed to attract and retain good personnel." (4:3-39)

The document suggests an action program which would:

- Identify the software personnel requirements.
- Review the current classification procedures.
- Review training requirements and training capabilities for preparing software engineers.
- Recognize this expertise as critical and develop necessary career programs and incentives.

Separate and independent initiatives began sprouting throughout the government as more and more dependency upon the computer was evidenced.

The Air Force added software engineering courses to the curriculum at the Air Force Institute of Technology (AFIT) and the Naval Postgraduate School offered a degree in Computer Science.

The prime thrust of all this activity, however, was a reaction to the computer technological explosion and many efforts proceeded redundantly rather than synergistically.

MAJOR ISSUES

The problems of providing sufficient numbers of qualified software personnel did not lend themselves to short-term management resolutions. The full impact on qualification requirements, and hence on recruitment and placement, from the establishment of a highly specialized series such as software engineering had to be thoroughly analyzed to assure that the benefits to be gained outweighed the disadvantages to be incurred. The vast number of Federal employees already performing duties requiring specific KSA's of computer resources and classifed to other occupational series, some non-professional such as the Computer Specialist, caused the following issues to be considered:

1. Is Software Engineering a "professional" occupation; i.e. requiring a engineering academic preparation. According to the position classification standards for the Engineering series:

> "Whether an occupation is placed in the engineering group of physical sciences group depends upon whether the nature of the work and the qualifications required for its performance are predominantly identified with an engineering or physical science discipline. It is the common core of professional knowledges and abilities representing a discipline required for perfornance of the work which distinguishes series within occupational groups. Areas of application or investigation are of secondary significant. . . .
>
> In some cases, where large numbers of multi-discipline positions constitute what may be considered to be a new profession or occupation with a common core of duties and qualifications required, a new series is established, e.g., Soil Conservation Series." (5:11,12)

By definition, the Software Engineer is responsible for the application of engineering principles, theory and concepts to the development of the software which works, reliably and economically. These requirements imply a specific academic progression to a certified level of competence, even though many years of experience may sometimes narrowly suffice. For example, an individual who "engineers" weapon system software must have knowledge of the avionics subsystem within the weapon system if he is to develop, design, or maintain the embedded weapon software and/or firmware. One must understand the environment in which the weapon system is employed, i.e., threats, countermeasures, etc. In addition, there is the equally important skill of understanding computer and languages. Both these skills are required to be effective in designing algorithms and the instructions for the weapon system embedded digital computer. This example holds true for large, complex, non-embedded systems which control many subsystems as an integrated entity.

2. What is the impact of the lack of series definition, specialization, classification and grade criteria?

Since there are no classification standards in Federal service for this engineering field, Federal agencies have been forced to establish and fill professional software positions from related disciplines such as Electronic Engineers, Mathematicians, Computer Specialists, Aeronautical Engineers and General Engineers. This clouds the career patterns of the selectee, since the individual is not "set apart" in title and/or series as to his specific expertise in software engineering. The lack of series definition and specific classification criteria create an opportunity for misunderstanding in grading positions since the classifier must search for an appropriate standard or standards which will properly grade the duties of the Software Engineer.

3. Are there recruitment and retention obstables?

Industry has recognized the need for a career field for the Software Engineer. Given the fact that demand for Software Engineers far exceeds the supply, and that premium salaries are being offered by industry, the Federal government is lagging behind in recruitment and retention. At the entry level, salaries offered by private industry exceed Federal salaries by as much as $10,000 or more.

Once in the Federal service, the lack of a defined career path for Software Engineers impacts upon retention.

Although the government provides the opportunity for challenging work, and the opportunities for meaningful advancement, competitive salaries are necessary in order to attract and retain highly qualified personnel in this new and emerging discipline.

4. What is the impact on employees with other series designations who might be subject to reclassification?

Positions currently properly classified will not be impacted. Those classified to a professional engineering or mathematics discipline and whose primary duties are software engineering would require a change to the new engineering series. Those properly classified in the non-professional GS-334 series will remain as they are.

The qualification standards must be developed such that the basic requirements and alternate requirements permit the placement of all incumbents of "certified" software engineering positions.

These issues were given primary consideration; and, after many Ad Hoc discussions, separate attempts at alleviating the problem within specified time windows were initiated. In the following paragraphs, an account of representative efforts is discussed.

DEPARTMENT OF NAVY EFFORT

The Navy's investigation of the problem began with 20 Navy representatives attending a meeting in August of 1979 to study the computer software engineering field in order to better understand this emerging occupation and the role it plays within the Automatic Data Processing (ADP) community.

The Navy's plan of action included:

1. Define the scope of the Study
   - Develop an overall plan for the study including
     - Study format
     - Methodology
     - Milestones
   - Tie-in with the OPM - Professional Council for Federal Scientists and Engineers, San Francisco Region, Project for a Software Engineer Classification standard
   - Identify major functional areas
   - Identify major problem areas and issues
2. Identify Procedures and Assignments
   - Define workload and assignments
   - Conduct fact-finding
   - Develop a draft of findings
   - Coordinate with other offices and activities in geographical area/systems command
   - Corroborative fact-finding by OP-141C1
3. Develop Final Draft of the Study
   - Review and develop draft Interpretive Memorandum
   - Circulate draft for review and comments
   - Complete final draft for publication
4. Publish Interpretive Memorandum on Computer Software Engineering Positions.

Some resistance to the study effort was evident by those activities with a large number of software personnel classified under the 334 standard; however, the majority of the new activities agreed that a special classification series should be established for the long term.

Many problem areas and issues were exposed, the most pervasive issue was an uncontested description of the software engineering discipline and the associated minimum basic qualifications.

The final results of this effort was an Interpretive Memorandum to aid in the classification process. (3)

INTERPRETIVE MEMORANDUM

The Interpretive Memorandum, which was formally issued in October 1981 from Navy Personnel Headquarters, offers extensive classification guidance on Software Engineering positions which was completed at Navy Personnel Headquarters. It provides criteria for determining the classification of engineer and other professional positions involved with computer software engineering for embedded or similar computer systems. "Software Engineering", as defined in this memorandum, covers the engineering work pertaining to the research, development, design, testing, production, installation, maintenance, operation and other functions relating to computer programs and the data required to allow the computer to perform its functions. This guide provides occupational information and grade level criteria for the classification of Navy positions requiring the performance of professional technical work in the field of computer software engineering.

NAVAL SURFACE WEAPONS CENTER (NSWC)

For the Naval Surface Weapons Center, the Interpretive Memorandum has not solved the total problem. The Naval Surface Weapons Center has been concerned with identifying the academic preparation and the actual preparation of personnel. (Reference Appendix A). Because of the different backgrounds of those involved in the study, an early decision was made to ignore the titles of positions and the job descriptions. NSWC began by identifying the knowledge areas important to the development of the Nayy systems for which NSWC has, or could have, reponsibility. These knowledge areas, considered to be necessary for those individuals developing successful systems consist of:

1. Controls - controls, information feedback systems, basic systems distinctions (open loop, closed loop, hierarchical, etc.).

2. Process exposure/dynamic interrelationships - time-dependent behavior, system interactions, the "process" concept, cross effects, binding time, process communications, cooperation and competition.

3. Design Principals - the principals of engineering (or the scientific method), elements of the design activity (specification, analysis, decomposition, synthesis, testing), maintenance and reliability.

4. Interpersonal communication skills - written and verbal communication, team participation and team leadership.

5. Functional capabilities of digital hardware - logical structure and composition. (This area was recognized to be potentially divisible into computer hardware and digital non-computer hardware.)

6. Software design technology - system life cycle, specication techniques (e.g., PSL/PSA, Workbook, Jackson, etc.), development techniques (e.g., chief programmer, structured walk-through, design reviews, builds, code reading), documentation, modification and maintenance.

7. Evaluation - systems analysis techniques, models and modeling identification or creation of alternatives, characterization of trade-offs.

8. Systems integration - component and subsystem testing, system reliability, progressive testing, diagnostic capability, degraded mode options, recovery.

9. Programming systems techniques - programming languages, systems programs, structured programming, modularity, stubs, program documentation, program testing.

10. Human factors engineering - human/machine interface, dialogue design, prompting, "trainability" and "learnability", adaptability and design and change. (This area was recognized as potentially divisible into software design for human use and hardware design for human use.)

In addition, a plan has been devised by the Software Engineering Committee for the acquisition and training of Software Engineers for the Naval Surface Weapons Center. It was recommended that all newly hired persons who are intended for organizations involved in software development and do not already have an appropriate background be included in the Software Engineering Development Program. Others who wish to change from their present jobs to developing software will also be included.

The plan assumes that the trainees have no detailed knowledge of computers but that they do have at least a BA degree or equivalent in one of the scientific fields. Training opportunities will also be provided for experienced software development personnel through a series of short courses conducted at the Center.

The Plan includes a two-step training program including formal classes as well as On The Job Training (OJT) with specific training experiences to be added as needed. (Reference Appendix A).

Pacific Missile Test Center (PMTC)

At the Pacific Missile Test Center, the impact of the shortage of employees having software engineering expertise is severe. It causes understaffing of current operations and overworking of available people. There is great fluidity in labor supply which leads to round-robin attrition, lost continuity and an unbalanced work force. High attrition leads inevitably to a high risk environment in terms of performance, schedule, and cost.

It became increasingly evident that a plan to overcome this growing problem was needed to increase the supply of skilled software engineers from within the organization. The proposed program addressed a modular approach which:

(a) offered a career training program at the undergraduate level.

(b) provided for career branching after the undergraduate program is completed.

(c) offered several graduate certificate programs to employees currently having a bachelor's degree in the professional fields.

(d) provided for up-dating state-of-the-art to employees currently working in the software field.

The objective of this program would be the potential creation of new resources for the areas of Software, Electronic Warfare, and Test and Evaluation.

The Career Development Division was tasked to design a Software Engineering Career Development Program which would offer modular training opportunities and attract potential resources to the software engineering function at the Center.

A survey of need for this function was conducted in late fiscal year 1977. The scope was limited to software support for scientific and engineering functions and excluded management, business, and supply functions. The training program was to be designed around requirements for Tactical Fighter Weapons Systems software requirements, since the biggest "gap" was occurring within this function. Further, it was felt that any training program designed to meet needs for this highly specialized area would automatically include the knowledge, skills, and abilities necessary to meet the requirements in the excluded areas.

The survey indicated an immediate requirement for 46 new journey employees to meet current demands; 10 additional requirements for fiscal year 1978; and 11 additional requirements for fiscal year 1979. When coupled with retirements and attrition rates over a five year period, it became evident the need for qualified software engineers far exceeded the supply. In addition, many of the current employees badly needed training to keep abreast of the exploding technological changes occurring in the software field.

An intensive recruitment program was then mounted to attact such resources to the Center. In addition to attracting recent college graduates at the entry level, attempts were made to capture intermediate or journey professionals in the field. Although some gains were made, the competition for resources in other government agencies and in private industry created an environment where the attrition rate was higher than the accession rate.

Concurrent with the above efforts to attract employees to the software function at the Center, a major Personnel Evaluation and Audit was conducted by the Office of Personnel Management. Findings of the evaluation team included many high grade positions recommended for down-grading. As a result, Reduction-in-Force placements occurred, and many of the supervisory/managerial positions in the software function suddenly had new employees who were limited in the knowledge/abilities of the software function.

It was in this climate that the need to design a program aimed at attracting new employees into the field; provide up-date training to current software employees; provide supervisory/managerial overview orientations; and provide for some type of upward mobility opportunity which would attract and retain employees over a longer period of time in an effort to "grow our own" software employees became paramount.

This effort was separated into two phases:

Phase I - provide for a Task/Competency Needs Assessment. This phase was completed during fiscal year 1978.

Phase II - Design Modular Training which would include all of the elements addressed above. The objective of this phase was to design and implement modular training which would include requisite knowledges/skills to satisfy each element of the overall program. The highest priority in the modular development was the providing of state-of-the-art training to current employees. This was due to the high attrition rate of employees having the requisite skills; the rotation of new supervisors into software functional areas; and the recruitment of personnel having less than adequate knowledge/skills/abilities. This combination presented a serious performance/p oduct problem for the Center.

The Phase I product was the identification of knowledges, skills, or abilities required of employees in each of the occupations assigned to the tactical fighter weapons systems function. These knowledges, skills, or abilities were then translated into the development and presentation of 25 courses which offered up-date and state-of-the-art training believed to have the most urgent need. Three-fourths of the courses in this module have been presented to current employees at least once, and feedback from supervisors indicate they see immediate results in improved performance or motivation of their employees.

Appendix B addresses the complete modular training program; the Phase II product.

PROFESSIONAL COUNCIL EFFORTS

The intiatives described in the preceding paragraphs represent short-term, stop-gap measures and may in themselves solve some of the problems of the organizations involved.

However, a long-term commitment by the public sector mangagement to develop personnel programs which ensure adequate numbers of software personnel for complex system software acquisition and maintenance is needed.

The effort by the Professional Council is considered the first step to that end. In 1979, the Professional Council established a committee to work with OPM in the development of appropriate classification and qualification standards for engineering positions. The Pacific Missile Test Center, Pt. Mugu (PMTC) member, K.I. Lichti, was designated action officer for the project. His sub-committee consisted of the following persons:

Technical Experts

G. HUNT	Navy, Pacific Missile Test Center	Sub-Committee Chairperson
S BERMAN	Navy, Pacific Missile Test Center	Sub-Committee Member
G. WROUT	Navy, Pacific Missile Test Center	Sub-Committee Member
D. NAURATH	Navy, Pacific Missile Test Center	Sub-Committee Member
J. BOK	Navy, Pacific Missile Test Center	Sub-Committee Member
H. JOHNS	Navy, Pacific Missile Test Center	Sub-Committee Member
J. SALAZAR	Air Force Vanderberg	Sub-Committee Member
D. FARREL	Navy, China Lake	Sub-Committee Member
J. HOWELL	Air Force, Edwards Air Force Base	Sub-Committee Member

Personnel Analysts

J. BENNISON	Office of Personnel Management WR	Analyst Team Chairperson
D. PIUSER	Navy, Pacific Missile Test Center	Analyst Team Member
V. VERNEUILLE	Air Force, McClellan Air Force Base	Analyst Team Member

The tentative schedule for 1980 was established as follows:

ACTION	DATE	LEAD ACTIVITY
Draft Questionnaire & Series Description	20 June	OPM-Western Region
Establish Action Plan	20 June	PMTC/OPM-WR
Draft Cover Letter	3 July	OPM-WR
Solicit Distribution Lists from Council Sub-Committee	3 July	PMTC
Distribution Lists due to PMTC	10 July	Council Sub-Committee
Generate Composite Distribution List	11 July	PMTC
Finalize Questionnaire and Series Description	14 July	PMTC
Distribute Questionnaire	15 July	PMTC
Establish Analysts Committee	18 July	PMTC/OPM-WR
Questionnaire Response due to PMTC	1 August	Distribution
Complete Data Search	5 August	PMTC
Complete Initial Data Analysis	8 August	Analyst Committee
Review Analysis	19 August	Council Sub-Committee
Develop Series Definition Package	8 September	All
Deliver Package to Council	15 September	PMTC
Review	26 September	Council
Deliver Package to Standards Development Center, OPM, Wash., D.C.	30 September	Council

The sub-committee distributed a questionnaire to a cross-section of Federal activites. The analyst team convened on August 19-20, 1980, and completed the initial review of responses. Of 285 questionnaires mailed to 69 activities in 8 departments/agencies, 26% were returned by the activities. The quality of the input ranged from cursory to very thorough. The enclosures and additional materials furnished by activities such as the Naval Surface Weapons Center, Dahlgren, Virginia, and the Naval Weapons Center, China Lake, California were indicative of the interest in this topic, and other studies which had already been undertaken. The respondees indicated that approximately 1200 positions would potentially fall within coverage of the Software Engineer Series GS-8xxx as defined in the questionnaire. The number of positions identified exclusively with a requirement for an engineering degree was a significantly lesser number than 1200. The most significant population, however, was in the Electronics Engineering Series GS-855. Though some jobs were included in the Computer Specialist Series GS-334, the vast majority, estimated at over 95%, were classified to professional series in the GS-800, GS-1300 or GS-1500 groups.

The overwhelming response reflected the viewpoint that the nature of the work to be performed required entry level professional training comparable to that attained in a four year degree program leading to a BS in engineering or computer science or mathematics or closely related disciplines. In addition to the discussion of Minimum Qualifications, a discussion of the required knowledges, skills, and abilities for full performance was provided. The majority viewpoint characterized the occupation as grounded principally in engineering fundamentals and computer knowledges, with related knowledges of mathematics and physical sciences, and abilities in general management and communications.

The questionnaire which asked for distinguishing characteristics of Software Engineering from other occupations was not addressed by a significant number of respondees even though principal differences were detailed. The team did not summarize inputs regarding the Computer Specialist series since the trend on minimum qualifications clearly characterized the work as "professional".

From the data presented it was the Council's opinion that the software engineering occupation should be established as a fully professional engineering series. A very significant number of 1239 positions identified with the proposed series in the questionnaire responses are currently classified in the Electronic Engineering Series GS-855. This fact substantiates the conclusion that professional engineering training is the over-riding qualification requirement.

The Council further concluded that with approval of the new series, the Federal service will be competitive with industry and will significantly benefit in its efforts to attract and retain a fully qualified staff to meet the software requirements of the Federal government.

Appendix C contains excerpts from the report as sent to OPM, Washington, D.C. on 30 September 1980.

STATUS AND PROGNOSIS

The current state of the individual short term efforts appear somewhat encouraging while attainment of the long term objective looks dismal. The reports on these individual efforts are:

Naval Surface Weapons Center (NSWC)

- The plan for the acquisition and academic training of Software Engineers for NSWC has been approved and endorsed by their Technical Director.
- NSWC is currently implementing Steps 1 and 2 (Reference Appendix A).

Pacific Missile Test Center (PMTC)

- The PMTC Software Engineering Career Development Program effort is currently progressing.
- Software Management Awareness Core has been proposed and consists of the following:
    - Executive overview of Software Life Cycle Management.
    - Software Project Management Requirements and Planning.
    - Software Task Management Responsibilities for NAVAIR Projects.
- The Software Technology Update Core is in process.
    - The objective is to provide opportunity for current software employees to increase their knowledge/skills/abilities in software technology and bring productivity to optimum level.
- The Software Retraining Core Graduate Certificate Program was initiated 1 Sept. 1980.

- The use at this time of the term "engineering" in association with computer programming and systems analysis would require engineering societies and engineering granting institutions support. Such support has not been obtained. Will PMTC buttress their report to overcome these concerns?

- Are private sector employers using software engineering titles? More detailed information is needed to confirm this concern. What is PMTC's plan for contacting private sector employers to obtain this information?

- Will the establishment of a software engineering series/occupation would substantially reduce turnover among such personnel? Will PMTC conduct some kind of a pay survey and if so what is PMTC's plan for conducting such a pay survey?

Most recently, Mr. Jerry Reed, Executive Director of Acquisition for Chief NAVMAT has expressed great interest in the software problem in DOD. He is exploring what can be done at NAVMAT Headquarters to assist. His past successes allow for a great degree of optimism.

Nevertheless, the Federal government is at the threshold of an ever escalating software requirement. To continue to ignore the need for a positive, pro-active pursuit of a viable solution is irresponsible and can result only in disaster in the long term.

- The objective is to provide a core series of courses in software at the graduate level which provides expertise in software and qualify employees with a Bachelor's degree to successfully complete/qualify for positions in the software function.

- The Undergraduate Software Training leading to an Undergraduate Certificate Program has not yet been approved.
- The Master of Science Program in Electrical Engineering and Computer Science has been in existence since 1975 and is designed for students who need to update their background, experience and operational ability in the computer area.

Department of the Navy

- Interpretive Memorandum was issued October 1981.
- There is no indication at this time that it is in use for classification purposes.

Professional Council

The long term solution attempted by the Council has been stymied and no new strategy has been developed. To date:

- There has been several meetings with Paul Katz, Director of the Standards Development Center.
- These meetings have resulted in no new actions to date.
- PMTC is investigating answers to the following questions and concerns raised by Mr. Katz:
    - What can PMTC do to overcome the evidence that indicates the continued requirements for "professional" qualifications?

Gwendolyn Hunt
Director, Data Processing Service Center, West
Pacific Missile Test Center
Point Mugu, California 93042

BA	Tennessee A&I	1956
MS	University of Southern California	1976
Graduate, Industrial College of the Armed Forces		1979

Experience:

26 years of software engineering involvement including:
Pacific Missile Range - Range Software Development
Pacific Missile Test Center - Head, Tactical Software Design
Pacific Missile Test Center - Assc Director, Systems Technology Dept.

Current Assignment:

Director, Data Processing Service Center West

Responsible for all Automatic Data Processing support for MAVMAT Activities located within the proximity of the Pacific Missile Test Center.

STANDARDIZATION ISSUES OF THE FUTURE

SESSION CHAIRMAN: Darlow G. Botha
AFWAL/AAAI

MODERATOR: Frank A. Scarpino
AFWAL/AAA

THE APPLICATION OF STANDARD SYSTEM SPECIFICATION TECHNIQUES

TO THE DESIGN OF VERY LARGE SCALE INTEGRATED CIRCUITS

D. Jordan

Airborne Software Division
Marconi Avionics Limited
Elstree Way
Borehamwood
Hertfordshire
01-953 2030

Dave Jordan obtained his honours degree in Mathematics from the University of London in 1974. After some years in the telecommunications industry he joined the Airborne Software Division of Marconi Avionics Limited where he is now the senior member of a team which has developed the MENTOR system specification system.

The design of large-scale integrated circuits has traditionally been the province of silicon real-estate artists. Standardisation has meant keeping the same artist for each design iteration. The infeasibility of this approach for circuits with gate counts between 10,000 and 100,000 is apparent.

The use of modelling programs, using hardware description languages such as ELLA (1), leads to the use of standard circuit modules with their descriptions held in a model library. At Marconi Avionics this approach has been extended through integration with MENTOR (2) a system specification system.

The MENTOR approach is based on a standard specification language which can be applied at all levels of design and which allows the chip designer to both differentiate and correlate the behavioural and functional descriptions of circuits and modules. Advanced language analysis techniques enable detailed design verification and the construction of a perfectly consistent design database.

INTRODUCTION

The design of large-scale integrated circuits has traditionally been the province of silicon real-estate artists. Standardisation has meant keeping the same artist for each design iteration. The infeasibility of this approach for circuits with gate counts between 10,000 and 100,000 is apparent.

The effective design of such large scale circuits can only be achieved through an increased emphasis on standardised methods of modularisation and specification. To some extent this trend is already visible. Hardware description languages such as ELLA have been developed enabling the detailed specification of standard circuit modules. Modelling programs can then be generated which represent the specifications of designs using those standard modules, and which enable some verification and testing of designs.

At Marconi Avionics we have been separately examining the use of system specification languages for the verification and validation of designs in a more abstract way. This work has led to the development of MENTOR, a sophisticated new specification system incorporating powerful automatic techniques of specification analysis. With some extensions we believe that MENTOR will prove a valuable addition to the arsenal of specification methods for very large scale integrated circuits, which is fully complementary to the hardware description language approach.

The MENTOR approach is based on a standard specification language which can be applied at all levels of design and which will allow the chip designer to both differentiate and correlate the behavioural and functional descriptions of circuits and modules. Advanced language analysis techniques enable detailed design verification and the construction of a perfectly consistent database.

SPECIFICATION: A DOUBLE EDGED SWORD

It is usual to regard specification as a simple interfacing activity. On the one hand the specification informs potential users of a device of the designer's intentions in respect of the behaviour which the device will display. On the other hand the specification informs the implementor of the functions which the device must implement.

The specification is indeed subject to two forms of checking. The potential user must satisfy himself that he can operate the proposed device in such a way as to satisfy his goals and needs. The implementor must be able to demonstrate that his implementation in fact achieves the specified functionality of the device. These two activities comprise respectively the validation and verification of designs.

The central problem which must be addressed by any specification method is that of achieving a balance between the requirements of verification and validation. It is important to recognise in this context that a specification is not a pure and simple document which is either the only correct specification or not the correct specification at all. Specifications incorporate decisions: decisions which determine the "operability" of the proposed device, and decisions which determine the functional structure of the device.

An important role of the specification is to make explicit all of the decisions which have been made. The validation activity must be able to distinguish the impact of functional structure on the operability of the device, and the implementation /verification activity must be able to distinguish the impact of operability constraints on the interpretaion of functional structure.

We believe that this can be achieved by means of a strongly structured specification approach, and that functional abstraction at high levels is of paramount importance.

THE ROLE OF HARDWARE DESCRIPTION LANGUAGES IN SPECIFICATION

Hardware description languages are designed to enable the behaviour of a device to be specified in terms of its functional structure. The device is represented in terms of the procedural, or algorithmic interactions between its functional components without any regard to the way that those procedures are to be realised in terms of the interconnection of physical circuit modules.

Individual functional components may in turn be specified in terms of yet more elementary components and their procedural interactions, or they may be elementary building blocks whose behaviour is specified in terms of their register transfer functions.

This structure serves the purposes of the implementation/verification activity rather well. A device may be specified to any level of detail without predjudicing the physical design of functional components or of their interconnection net. Subsequent designs can be modelled similarly in terms of circuit components whose specifications are maintained on a central library, and the descriptions can be exercised by simulation software and tested to verify that the required functional behaviour can be achieved.

It is my view, however, that despite their hierarchical nature, hardware description language specifications fall short of satisfying the requirements for validation of device operability. To be sure, simulations can be constructed to any required degree of complexity, so that the operability of a complete device can be tested in simulation. However, for a device of any appreciable size it will never be possible to exercise more than a small proportion of its potential states in this way.

Furthermore, the exercising of a hardware description language specification is dependent upon the behaviour of functional components, at some level, being specified in detail in terms of register transfer functions. Consequently, by its very nature, the simulation of a device from a hardware description language specification produces a great deal of detailed information which, although valuable, is to some extent peripheral to the issue of operability.

The essence of the specification approach being proposed here is that functional abstraction should be employed at high levels of specification in order to clearly exhibit the relationship between device behaviour and operability. The introduction of functional detail should be carried out in a structured manner, parallel with, and in support of the hierarchical elaboration of behaviour. This in turn should be a guide to the synthesis of designs in terms of interacting functional components.

Nevertheless it is to be anticipated that ultimately the specification will contain sufficient functional detail to be capable of expression by means of a hardware description language. In view of the advantages noted above in the use of this style of specification for implementation and verification activities it would seem to be essential to provide some means whereby the hardware description language can be generated. It would however appear that by orienting the specification process towards the creation of a design at the level of a predefined library of functional components the procedures of the hardware description language could be automatically and painlessly generated.

THE ROLE OF ABSTRACTION IN SPECIFICATIONS

The MENTOR approach is based upon the fundamental idea that any device has significance only when it is embedded as a component in some wider system, and conversely that any component in any system can be regarded as a device in its own right.

In view of this we can say two things; firstly, that the significance of any device can only be expressed in terms of the way in which it interacts with other components of the system in which it is embedded. And secondly, that the process of design concerns itself with the replacement of one device by a number of other more elementary devices interacting in such a way that each has a significance in the achievement of the objectives of the parent.

The important point here, I believe, is that we do not need to know, in detail, what any component does in order to assess the operability of a given design. It is sufficient to know, in the abstract, the significance of what the component does so that we can determine whether it is being operated in an appropriate way to achieve the objectives of the design in which it participates.

Abstraction, therefore, is a crucial factor in the understanding of any design. If this were not so, surely, we could never design by the divide and conquer method. The point which I am trying to address, however, is this:-

When we set out to create a design for some device, all that we can ever know is:-

i) the way the device is going to be operated.

ii) the environment in which it will operate.

And

iii) the significance, in the abstract, of the operations which it will perform.

Certainly, once a certain level of detail has been achieved this is equivalent to knowing, if you like, the register transfer functions which must be implemented. However, above this threshold, specification and validation of designs can and should be carried out in the abstract.

What does this mean in practical terms? The main point, I suggest, is that a design comprises three distinct kinds of information (see Figure 1). Firstly there is information about the (abstract) significance of each component in the design. This can be expressed in terms of the abstract behaviour displayed by each component, and amounts to a specification of the component which can be used either as input to a further stage of design, or to select a standard component to fill the specified role.

Secondly there is information concerning the way that components operate together. And, thirdly, there is information about the relationship of the limited environments of individual components to one another and to the environment in which the parent device operates.

In the MENTOR system these distinctions are explicitly supported by means of distinct specification primitives so that:-

i) The abstract behaviour of a device can be expressed in terms of the type of functional relationships which are established between registers in given operational circumstances. Details of these functional relationships cannot be explicitly given except in terms of the functional structure of the device. However their significance can be made explicit by the use of appropriate names to identify register contents.

ii) The operational interactions between devices can be expressed in terms of the circumstances and manner in which they are activated, and

iii) The structural relationships between device environments can be expressed in terms of the significance of each register assembly of the parent device, which is, of course, expressed by an appropriate name identifying the register assembly in the parent device behaviour specification.

MENTOR therefore enables the designer to specify his device in a way which both differentiates and correlates its behaviour and functional realisation. This is particularly valuable for validation purposes since it enables the potential user to confirm that all significant aspects of device operation have been identified. Provided that each component is designed in a manner which fully reflects it operational significance then the interactions of the components will achieve precisely the desired effect in the environment of the parent device.

AUTOMATIC VERIFICATION IN THE ABSTRACT

A major issue in the use of an abstract specification style, and one of the reasons why a more concrete form of specification is desirable for detailed design work is the possibility that:-

i) different users may interpret the same abstraction in different ways, or conversely:

ii) two distinct users may use different abstractions for the same fundamental concept.

An important way of avoiding these problems is to make the specification process as visible as possible and encourage communication between users of the specification. In order to achieve this MENTOR maintains the database of specification information and provides a report generator which will provide, on demand, up-to-date diagrams, summaries, and analysis of the specification. This facility is, however, no guarantee that conflicts of interpretation will not from time to time arise.

There is however a certain amount of verification which can be performed in the abstract, since it is possible to see whether the operational interactions of a set of components achieve any effects which can be interpreted as the behaviour specified for the parent device. MENTOR incorporates a consistency checking mechanism, based on a detailed logical representation of its specification primitives. This interprets as inconsistent any situation in which the abstract function of a device is not realised by the interactions of the abstract functions of its components.

Thus MENTOR will generate diagnostic reports whenever it becomes apparent that the abstractions used by different users do not correspond. This mechanism will, of course, trap both:

i) situations where a design is simply wrong and will not work, and.

ii) situations where detailed design considerations have identified significant features of the behaviour of a device which were not anticipated and reflected in the specification of the functional structure of the device.

MENTOR therefore has an important role in ensuring that the abstract design specifications produced in the concept elaboration phase of a hardware design project are complete and consistent in advance of any detailed design work based on hardware description language style specifications. This contrasts with a situation in which the same information would be kept on paper in natural language and would be subject to misinterpretation at the detailed design stage when its significance must be clearly understood.

INTEGRATING MENTOR WITH A HARDWARE DESCRIPTION LANGUAGE

The application of a MENTOR-like specification tool in order to support the achievement and validation of device operability offers significant advantages in the design of complex, special purpose,device architectures. However, the detailed logic design activity will, we anticipate, continue to be based on hardware description languages such as ELLA. Therefore, reliable mechanisms will be required for the transfer of specification and design information from MENTOR to the hardware description language format.

Ideally a fully automatic transcription process is required, and the general purpose report generator facility of MENTOR provides an appropriate vehicle for the automatic construction of the hardware description language specification as a special purpose MENTOR report. This strategy, however, imposes some constraints on the specification process, which in turn have implications for the MENTOR system itself.

We have assumed that the devices to be specified will consist largely of standard generic functional elements. This will, we believe, be necessary in order that designers will be able to cope with the complexity of these devices and, more basically,because the circuit proving task will, otherwise, be infeasible. Detailed hardware description language style specifications will, therefore, be obtained by instantiating corresponding generic specifications maintained on a central library.

A consequence of this assumption is that the abstract specification activity supported by MENTOR must be specifically oriented towards the generation of a design in terms of the library generic elements. Therefore, MENTOR must provide structuring primitives which facilitate the specification of library elements by specialist hardware designers.

One mechanism which can be employed for this purpose is the general purpose property attribution mechanism. This mechanism enables each functional element to be described by a set of arbitrary attribute values identifying specific properties of the functional element. For example a given functional element could have the "generic type" property assigned the value "integer adder" and the "data precision" property assigned the value "8" to identify it as an 8 - bit integer adder. An alternative approach which shows some promise for the longer term, but which is currently still under investigation would be to extend the range of structuring primitives in MENTOR to correspond precisely to the library generic elements.

In addition to this, provision must be made to handle those parts of a required device which cannot conveniently be specified in terms of standard predefined elements. Clearly, for these elements the hardware description language specification must be produced manually. For the sake of completeness, however, these manually produced specifications should be represented on the MENTOR system. MENTOR will provide special primitives for this purpose.

CONCLUSION

As the potential complexity of integrated circuit devices increases, and as the demand for purpose built integrated circuits is stimulated the issue of operability of devices will be seen to be at least as significant as the current concern with ensuring functional correctness.

Future special purpose VLSI devices implementing complex functions will require many more control options than previous generations of easy to operate general purpose devices implementing relatively simple functions. For this reason it will be necessary to adopt specification standards equivalent in power to those already in use in the fields of systems and software design, and emphasising the issue of device operability

While very valuable for the achievement and verification of functional correctness, current standardised hardware description languages, such as WHDL (3) in the USA and ELLA in the U.K., appear to have severe limitations as regards the achievement and validation of operability. This is because hardware description language style specifications are intrinsically functionally detailed, whereas the issue of operability is best addressed in terms of functional abstractions.

Current work on more general abstract forms of specifications, including Marconi Avionics MENTOR system offer the prospect of standardised design specifications which are both verifiable and amenable to behavioural validation. The special requirements of the integrated circuit design process, however, demand that the specification be readily transformable to the hardware description language form.

For design tasks which require a preponderance of standard generic circuit elements which can be predefined in a library this apparently presents no major problems and MENTOR-like systems can readily be accommodated to the automatic generation of equivalent hardware description language. It therefore seems likely that MENTOR-like systems will prove a valuable addition to the arsenal of standard hardware specification techniques.

REFERENCES

1) "ELLA: A hardware description language" J. D. Morison, N.E.Peeling and T.L. Thorpe.
Proc. IEEE. Int. Conf. on Circuits and Computers. Sept 23-Oct 1 1982 NY PP604-607

2) "The MENTOR approach to requirements specification" D. Jordan and B. Hauxwell. AGARD Symposium on Software for Avionics. Sep 8-10 1982 The Hague

3) "Report of IDA Summer Study on Hardware Description Language" G. W. Preston.
IDA Science and Technology Division, Paper P-1595, DTIC ref AD-A-110866.

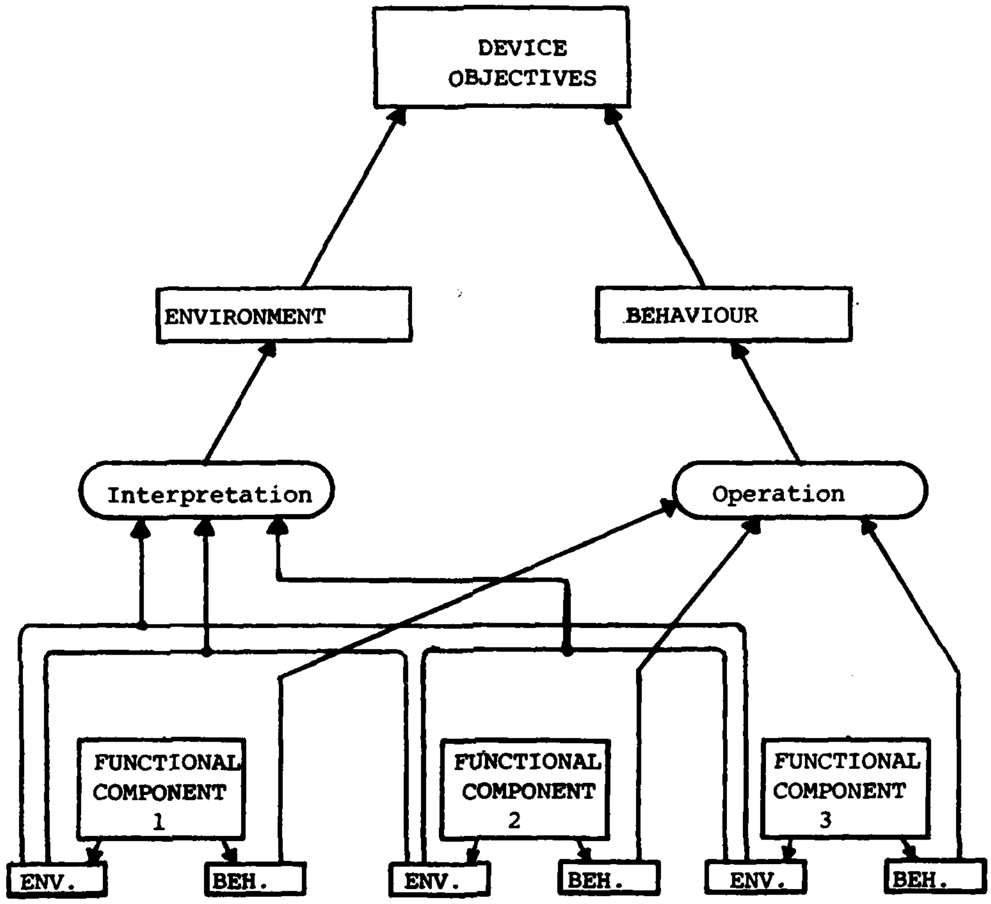


FIGURE 1. THE STRUCTURE OF ABSTRACT DESIGNS

ADVANCED STANDARDIZED SYSTEMS/SUBSYSTEM

SESSION CHAIRMAN: Major I. Caro
AFWAL/AAAF

MODERATOR: Colonel Darrold D. Garrison
AVRADA/DAVAA-D

COMPUTER STANDARDIZATION IN THE SUBMARINE ADVANCED COMBAT SYSTEM (SUBACS)

by Ronald L. Ticker

Naval Sea Systems Command

Biography

Mr. Ronald L. Ticker
Systems Engineer
Submarine Combat Systems Project (PMS 409)
Naval Sea Systems Command
Washington, D.C. 20362

BS University of Maryland - 1979
Graduate Studies - George Washington University
Member - IEEE and IEEE Computer Society

Abstract

The Submarine Advanced Combat System is a modular, distributed combat system for SSN 688 class attack submarines, currently in concept development. SUBACS will be among the first weapon systems to employ the AN/UYK-44(V) computer and the AN/UYS-2 Enhanced Modular Signal Processor (EMSP). The SUBACS architecture utilizes commonality and distributed processing to provide flexibility in reconfiguration and growth. To achieve this, standardization is exhibited at many levels within the system. This talk will focus on the development and evolutionary plans of the SUBACS and the utilization of standard computing devices in those plans.

I. INTRODUCTION

The Submarine Advanced Combat System (SUBACS) will be installed on new construction SSN 688 class nuclear attack submarines with delivery in 1988. The SUBACS consists of several subsystems and associated programs which will be integrated into a total ship combat system. It will replace existing sensor and fire control equipments providing significant performance improvement in acoustic detection, data base management, system reliability and flexibility plus reductions in manning, space and weight requirements crucial in the submarine environment.

The SUBACS Basic, illustrated in Figure 1, will be comprised of some new equipments integrated with retained portions of existing systems. This approach will permit advanced but mature capabilities to be introduced in a manner responsive to fleet needs while minimizing development costs. The SUBACS A, is shown in Figure 2. It will build on the Basic SUBACS and provide new capabilities, particularly in the area of fire control. Much of the equipment retained from older systems will be replaced by newly developed, more capable equipments. The SUBACS B, depicted in Figure 3, will provide additional improvements.

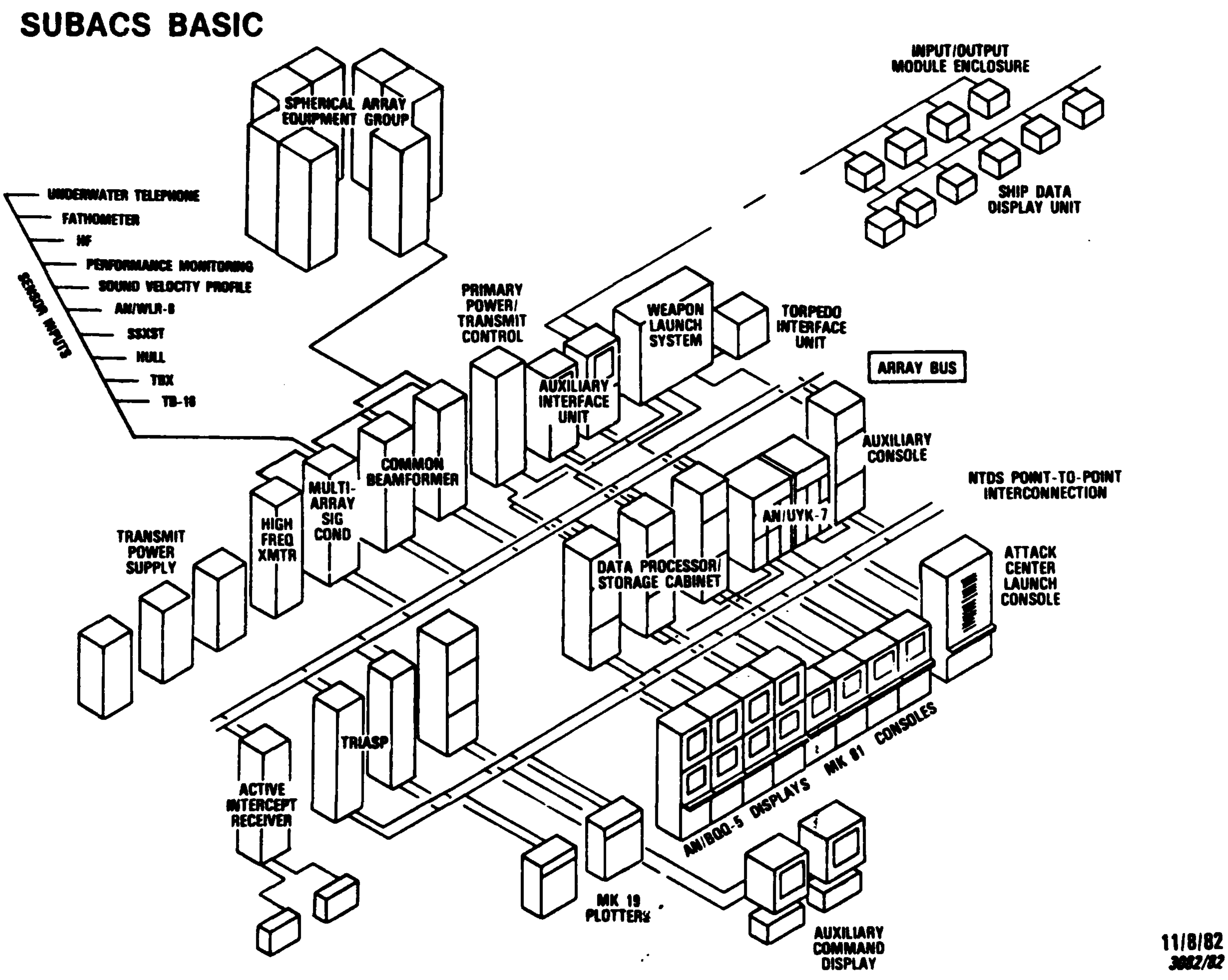
A summary of the SUBACS evolutionary, or pre-planned product improvement program is provided in Figure 4. Each stage of SUBACS development planned at three year intervals, is under-taken with the evolution to the next, and later stages being considered. The progression from stage to stage is faciliated by two important features of the SUBACS. The first feature is the commonality which the SUBACS provides both within and across systems. The second is the distributed, bussed architecture.

The SUBACS will be one of the largest combat system development activities the Navy has ever undertaken. By the time SUBACS B first goes to sea, over 2 billion source lines of new code will have been written. It will integrate more than 24 AN/UYK-44(V) computers and over 150 Motorola 68000 microprocessors.

II. COMMONALITY IN SUBACS

The SUBACS will be a modular system with a high degree of fault tolerance, flexibility, and capacity for technology insertion. Full system availability will be improved through the use of "Hot Spares." Failures occurring in existing systems result in loss of capability, the severity of which depends upon the nature of the malfunction. Each hardware element of the SUBACS will have at least one identical element capable of maintaining full system performance.

The modularity is largely due to the packaging of SUBACS elements at the drawer rather than the cabinet level. The SUBACS common enclosure, shown in Figure 5, consists of nine common drawers arranged in a three-by-three matrix. Each has its own power supplies and identical form and fit. Processors, such as the AN/UYK-44(V), are embedded four to a drawer in SUBACS Basic and eight to a drawer in SUBACS A. This allows replacement by drawer rather than cabinet permitting simpler, cheaper system upgrades.



SUBACS BASIC
SPHERICAL ARRAY EQUIPMENT GROUP
UNDERWATER TELEPHONE
FATHOMETER
HF
PERFORMANCE MONITORING
SOUND VELOCITY PROFILE
AN/WLR-8
SSXST
HULL
TBX
TB-16
SENSOR INPUTS
INPUT/OUTPUT MODULE ENCLOSURE
SHIP DATA DISPLAY UNIT
PRIMARY POWER/ TRANSMIT CONTROL
WEAPON LAUNCH SYSTEM
TORPEDO INTERFACE UNIT
ARRAY BUS
AUXILIARY INTERFACE UNIT
AUXILIARY CONSOLE
COMMON BEAMFORMER
MULTI-ARRAY SIG COND
HIGH FREQ XMTR
TRANSMIT POWER SUPPLY
NTDS POINT-TO-POINT INTERCONNECTION
AN/UYK-7
DATA PROCESSOR/ STORAGE CABINET
ATTACK CENTER LAUNCH CONSOLE
TRIASP
MK 81 CONSOLES
AN/BQQ-5 DISPLAYS
ACTIVE INTERCEPT RECEIVER
MK 19 PLOTTERS
AUXILIARY COMMAND DISPLAY
11/8/82
3082/82

Figure 1

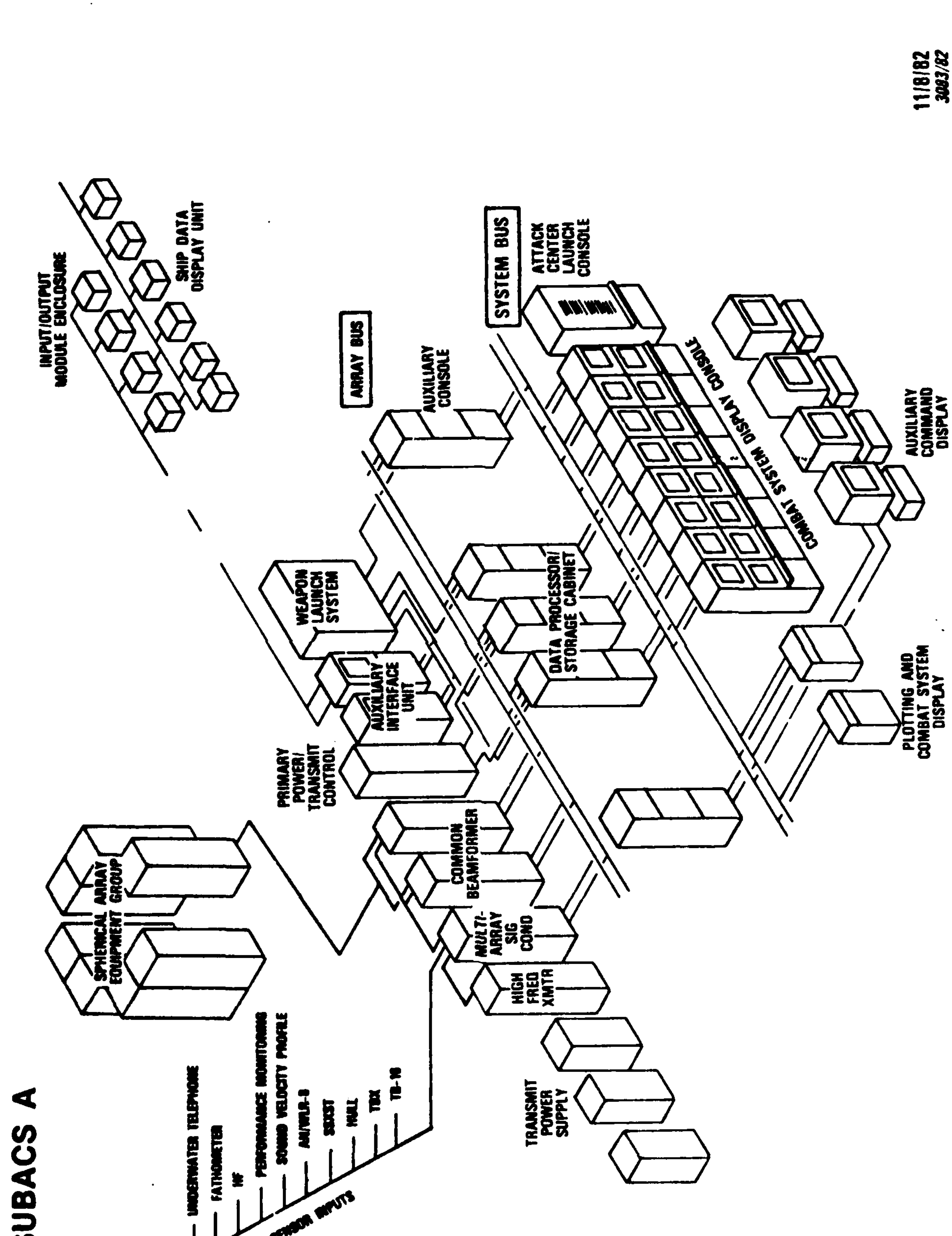

SUBACS A
UNDERWATER TELEPHONE
FATHOMETER
WF
PERFORMANCE MONITORING
SOUND VELOCITY PROFILE
AN/WLR-8
SSXBT
HULL
TBX
TB-16
SENSOR INPUTS
SPHERICAL ARRAY EQUIPMENT GROUP
TRANSMIT POWER SUPPLY
HIGH FREQ XMTR
MULTI-ARRAY SIG COND
COMMON BEAMFORMER
PRIMARY POWER/ TRANSMIT CONTROL
AUXILIARY INTERFACE UNIT
WEAPON LAUNCH SYSTEM
DATA PROCESSOR/ STORAGE CABINET
PLOTTING AND COMBAT SYSTEM DISPLAY
COMBAT SYSTEM DISPLAY CONSOLE
AUXILIARY CONSOLE
ARRAY BUS
SYSTEM BUS
ATTACK CENTER LAUNCH CONSOLE
AUXILIARY COMMAND DISPLAY
INPUT/OUTPUT MODULE ENCLOSURE
SHIP DATA DISPLAY UNIT
11/8/82
3083/82

Figure 2

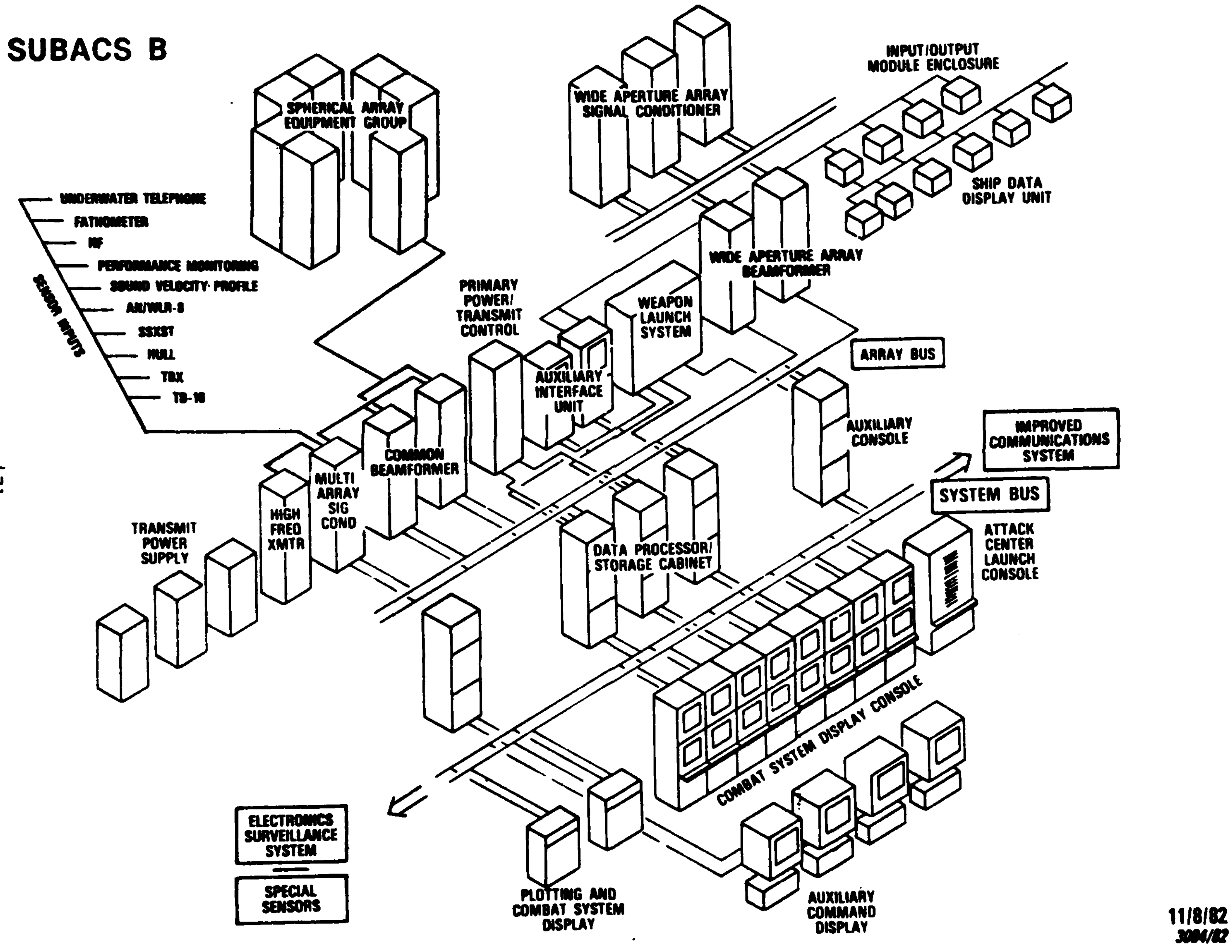

SUBACS B
SPHERICAL ARRAY EQUIPMENT GROUP
WIDE APERTURE ARRAY SIGNAL CONDITIONER
INPUT/OUTPUT MODULE ENCLOSURE
SHIP DATA DISPLAY UNIT
UNDERWATER TELEPHONE
FATHOMETER
NF
PERFORMANCE MONITORING
SOUND VELOCITY PROFILE
AN/WLR-8
SSXST
NULL
TBX
TB-16
SENSOR INPUTS
PRIMARY POWER/ TRANSMIT CONTROL
WEAPON LAUNCH SYSTEM
WIDE APERTURE ARRAY BEAMFORMER
AUXILIARY INTERFACE UNIT
ARRAY BUS
COMMON BEAMFORMER
MULTI ARRAY SIG COND
HIGH FREQ XMTR
TRANSMIT POWER SUPPLY
AUXILIARY CONSOLE
IMPROVED COMMUNICATIONS SYSTEM
SYSTEM BUS
ATTACK CENTER LAUNCH CONSOLE
DATA PROCESSOR/ STORAGE CABINET
COMBAT SYSTEM DISPLAY CONSOLE
ELECTRONICS SURVEILLANCE SYSTEM
SPECIAL SENSORS
PLOTTING AND COMBAT SYSTEM DISPLAY
AUXILIARY COMMAND DISPLAY
11/8/82
3004/82

Figure 3

SUBACS PRE-PLANNED PRODUCT IMPROVEMENT (P³I) PLAN

PRE-SUBACS

- BQQ-5C
- CCS MK 1 W/VLS
- NAVIGATION (ESGN)
- ESM (WLR-8)
- EXTERIOR COMMS (DLCS)
- MK 48/4 AND ADCAP
- TASM/TLAM(C)
- HARPOON
- SLMM
- 3" CM'S

SUBACS BASIC

- ACOUSTICS (SADS, TBX, MIDAS)
- COMBAT CONTROL (WLS)
- NAVIGATION (ESGN)
- ESM (WLR-8)
- EXTERIOR COMMS (DLCS) IMPROVEMENTS
- ACOUSTIC ARRAY BUS
- TRIASP
- NETWORK OPERATING SYSTEM (ACOUSTIC ONLY)
- MK 48/4 AND ADCAP
- TASM/TLAM (C)
- HARPOON
- SLMM
- 3" CM'S

SUBACS A

- ACOUSTICS IMPROVEMENTS
- COMBAT CONTROL IMPROVEMENTS
- NAVIGATION (ESGN IMPROVEMENTS)
- ESM (WLR-8 SEA VIXEN)
- EXTERIOR COMMS (DLCS) IMPROVEMENTS
- ENHANCED MODULAR SIGNAL PROCESSOR
- COMMON CONSOLE/PACS
- SYSTEM BUS (CC/A)
- NETWORK OPERATING SYSTEM (CC/A)
- COMPUTER AIDED DETECTION
- COMPUTER AIDED CLASS
- OPERABILITY IMPROVEMENTS
- ASW STANDOFF WEAPON
- TLAM (C&N)
- MK 48 ADCAP
- TASM IMPROVEMENTS
- 2 ADDITIONAL ACD'S

SUBACS B

- ACOUSTICS IMPROVEMENTS
- COMBAT CONTROL IMPROVEMENTS
- NAVIGATION (ESGN IMPROVEMENTS)
- SEA NYMPH VARIANT
- INTEGRATED COMMS
- ENHANCED MODULAR SIGNAL PROCESSOR IMPROVEMENTS
- COMMON CONSOLE/PACS IMPROVEMENTS
- SYSTEM BUS (CC/A, ESM, COMMS)
- NETWORK OPERATING SYSTEM (CC/A, ESM, COMMS)
- COMPUTER AIDED DETECTION
- COMPUTER AIDED CLASS
- OPERABILITY IMPROVEMENTS
- WEAPON IMPROVEMENTS
- WIDE APERTURE ARRAY
- MINING
- COUNTERMEASURES
- IR SENSOR (DARKEYES)
- RADAR SENSOR (BPS-1)

Figure 4

STANDARD ENCLOSURE

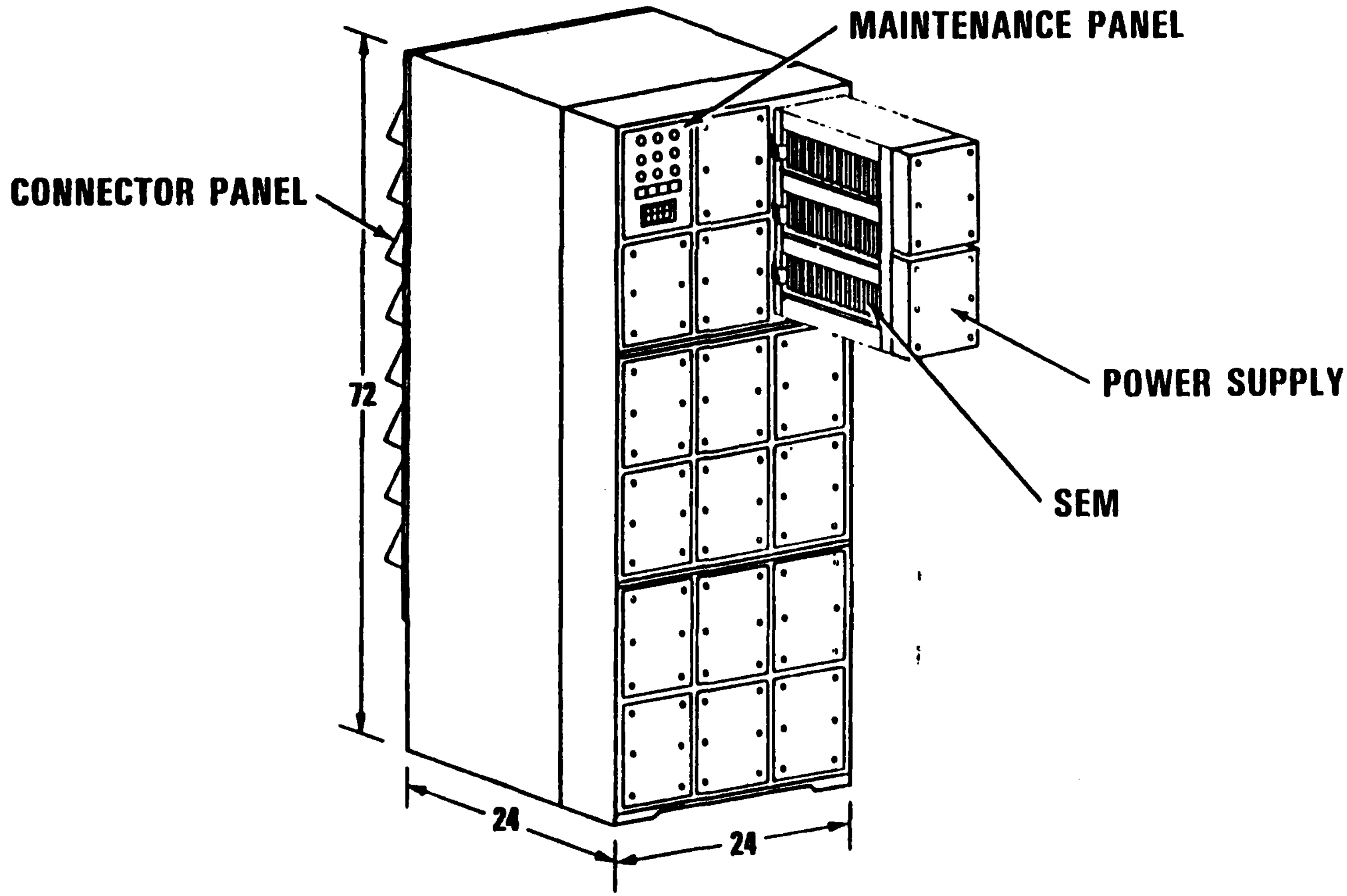


Figure 5

The modular cabinet concept provides a building block for other SUBACS units. The SUBACS Combat System Display Consoles consist of three common drawers in the lower third of the cabinet with the display surfaces and controls located in the upper two-thirds. The data processing and storage cabinets will house two Random Access Storage Systems (RASS) disks each in one-third of the cabinet with three drawers again occupying the lower third.

Since each AN/UYK-44(V) is identically configured, the use of a particular processor to support a given function is transparent to the operator. The loss of an AN/UYK-44(V) will not degrade the system since its function may be performed by another AN/UYK-44(V). The spare AN/UYK-44(V) need not be located in the same drawer or cabinet as the failed processor, but may be found elsewhere in the system.

The SUBACS system commonality is further enhanced through the widespread use of the Navy's Standard Electronic Modules (SEM). The new, large SEM format to be used in SUBACS will be compatible with VHSIC/VLSI technology insertion. The new format will provide for functional partitioning by card and the additional input/output capability required by new technology.

III. DISTRIBUTED ARCHITECTURE AND THE BUS

Existing, non-distributed combat systems are generally controlled by a single, central computer. For example, the AN/BQQ-5(V) sonar and fire control system Mk 117 both use AN/UYK-7 computers as central controllers. The occurrence of a computer failure forces a system reconfiguration or degradation. Maintaining system critical functions are subject to available computer resource limitations and a predetermined set of priorities. Probability of mission success is therefore very much dependent upon the reliability of single pieces of equipment.

A distributed architecture, like that of the SUBACS, replaces the central computer complex with a network of independent processors. Reliability and operability are improved by the presence of unused "hot spare" processors. Processing capacity is increased by the parallel processing capability of several computers operating simultaneously.

Communication between SUBACS processors and other elements is accomplished via a hierarchy of busses. The array bus connects beamformers, signal processors, mass data storage and system control units. A high bus bandwith of 32 megabytes per second is required including 100% reserve capacity, due to the high data rate associated with sensor data, Fiber optic technology is used to provide the increased throughput and also eliminate electromagnetic interference and reduce requirements for large, bulky cables. A fiberoptic system bus, having a 8 megabyte per second bandwidth, will transfer control and display information beginning with SUBACS A.

Most SUBACS units have internal cabinet busses. The cabinet busses extend the array and system buses to the drawer level. This effectively allowed subcabinet elements to communicate in the way cabinets do in more traditional system architectures. The independent communications permitted the subunits provide reliability and reconfigurability improvements. Product improvement is facilitated by placing the interface at a lower level than in traditional architecture thus reducing the extent of the required modifications. A subelement could be added in a spare slot without a major redesign of the unit.

Each AN/UYK-44(V) processor, EMSP, disk, as well as many other SUBACS sub-elements, are connected via a bus interface unit to the bus network. The bus interface is controlled by a Motorola 68000 microprocessor which performs message scheduling and status accounting. It has its own associated memory which is used for message queuing and routing.

The busses are interfaced through bridges. These are similar to the bus interface devices. The bridges are also controlled by a Motorola 68000 microprocessor. The bridge provides a mechanism for interfacing busses of different speeds and technologies. It also makes the location of distant processes appear as if they were local by acting as a relay point.

Each bus is paired with a redundant bus for reliability. Either bus can provide sufficient bandwidth on its own to completely service its users. All bus interface units and bridges are connected to both (redundant) busses.

The bus hierarchy provides a structure upon which the system may expand to meet future needs. Additional cabinet buses or subsystem busses may be added by bridging onto an existing bus. The number of bus ports usually limited by addressability, bandwidth considerations and driving distances, could therefore be increased according to system requirements.

IV. SUMMARY

The SUBACS, using standardization and state-of-the-art technology, will be the Navy's submarine combat system far into the twenty-first century. Not only will it be adaptable to meet changing threats and reliable in countering those already present, the SUBACS equipped attack submarine will pose a significant threat of its own. Vertical Launch Tomahawk cruise missiles with over-the-horizon targeting could provide both conventional and strategic capabilities. Advanced Capability Mk 48 torpedoes and the Anti-Submarine Warfare Standoff Weapon will enhance the submarine anti-surface and anti-submarine warfare roles. Improved target detection and localization techniques and equipment will reduce the time required for weapon targeting. Improved data management would simplify operations in heavy contact areas. In general, the submarine equipped with the SUBACS will be a force to be reckoned with.

Defence Materiel Administration
Airborne Electronic Division
115 88 STOCKHOLM, Sweden

Computer standardization in Swedish Air Force

(Mr G Elg, chief engineer, Airborne Electronic Division)

The Swedish Defence Materiel Administration, has been engaged in a standardization effort for airborne computors since 1975. The effort has resulted in the SDS80 Standard Computer System, which is now under full scale development for the JAS Aircraft program.

The SDS80 includes a high order language, based on Pascal, a programming environment PUS80 and a modular computer D80. All three components are well matched to each other to be an efficient solution to the computing requirements in the Swedish aircrafts and related systems. Details on the system will be presented seperately.

This presentation convers the specific background for the Swedish program and the approach we have taken to develop the standard. It also explains how we have got it accepted by the Swedish industry as a basis for a fix price aircraft development and production contract.

SDS80 - STANDARD COMPUTER SYSTEM

BACKGROUND

OBJECTIVES

APPROACH TO STANDARDIZATION

SYSTEM DESCRIPTION

COSTS

TIME SCHEDULE (PLANNED)

STATUS AND FUTURE

DISCUSSION

SDS 80 Standardized Computing System

Background

- Software costs become dominating
- Several different computers in a system

Solution

A standardized computing system consisting of

- A high order language
- A program development system
- A standardized multi-processor computer supporting the high order language

SDS80. A CONCEPT FOR STANDARDIZED HIGH-ORDER LANGUAGE COMPUTER SYSTEMS

1. BACKGROUND

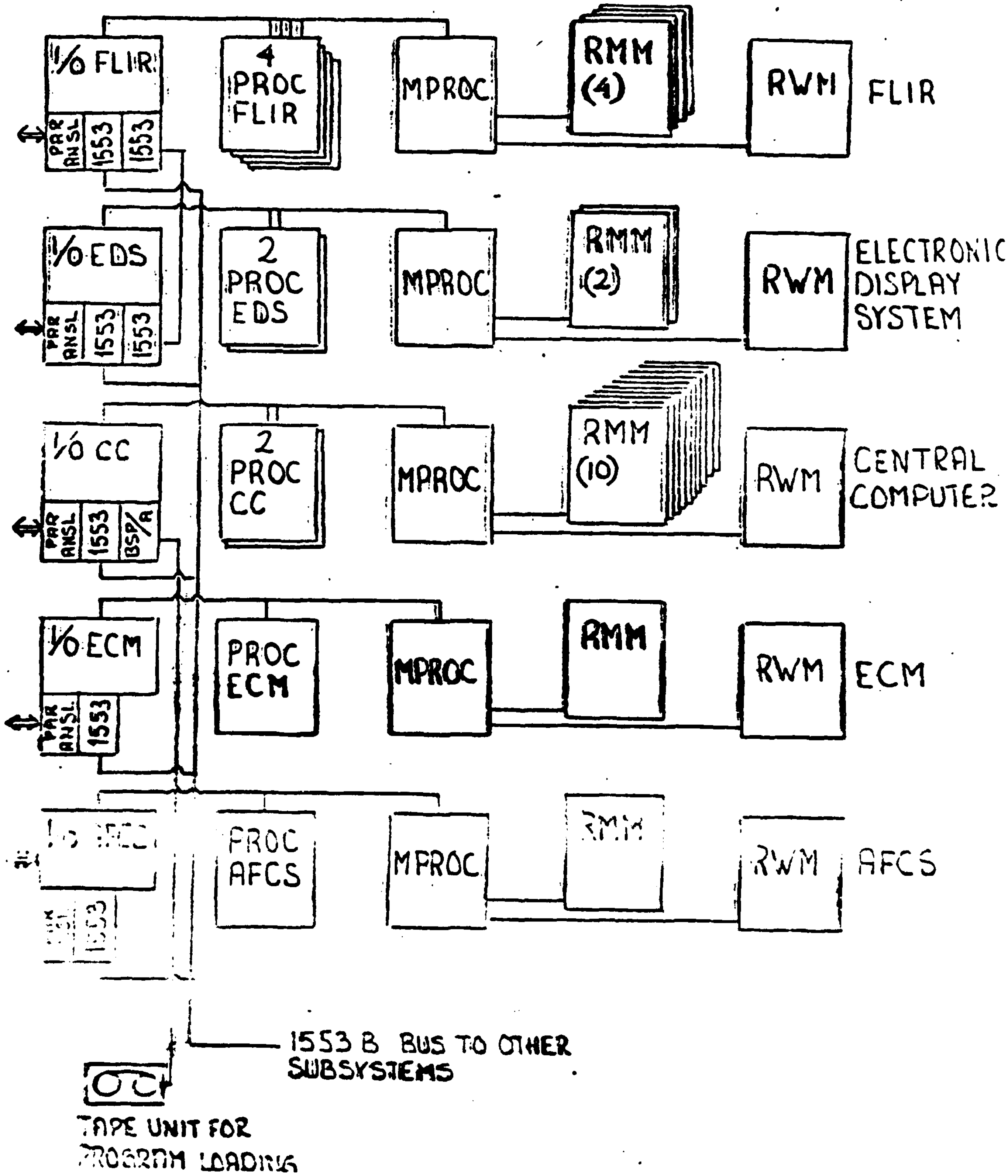
2. SDS80

   HIGH ORDER LANGUAGE
   PROGRAM DEVELOPMENT SYSTEM
   COMPUTER MODULES

3. APPLICATIONS OF SDS80

4. EXTENSION TO ADA

5. SDS80 PRESENTATIONS TO US GOV'T



33LA. COMPUTERSYSTEM
PRINCIPLE SCHEMATICS

INVOLVE AND COMMIT

RESISTANCE TO STANDARDIZATION:

LIMITS FREEDOM OF COMPANIES AND INDIVIDUALS

STYMIES TECHNOLOGICAL ADVANCES

REQUIRES INORDINATE AMOUNT OF COORDINATION

ALL EGGS IN ONE BASKET

NIH

ETC

OUR SOLUTION:

INVOLVE POTENTIAL USERS HEAVILY IN SPECIFICATION OF REQUIREMENTS

LET THE BEST EXPERTISE, REGARDLESS OF AFFILIATION, COOPERATE IN CONCEPT FORMULATION

INVOLVE MAJOR USER EQUIPMENT CONTRACTORS IN DESIGN AND DEVELOPMENT OF THE STANDARD EQUIPMENT

GIVE FREE ACCESS TO TECHNICAL INFORMATION

DESIGN TO ACCOMODATE TECHNOLOGICAL PROGRESS

SIDE EFFECTS THROUGH STANDARDIZATION

- STANDARDIZING ON A MODULAR LEVEL AUTOMATICALLY GIVES STANDARDS ON COMPONENT LEVEL.

- STANDARDIZING IN A JOINT DEVELOPMENT PROJECT GIVES THE POSSIBILITY TO USE RESOURCES FOR DEVELOPMENT, BOTH QUALITATIVE AND QUANTITATIVE, IN A MORE EFFICIENT WAY.

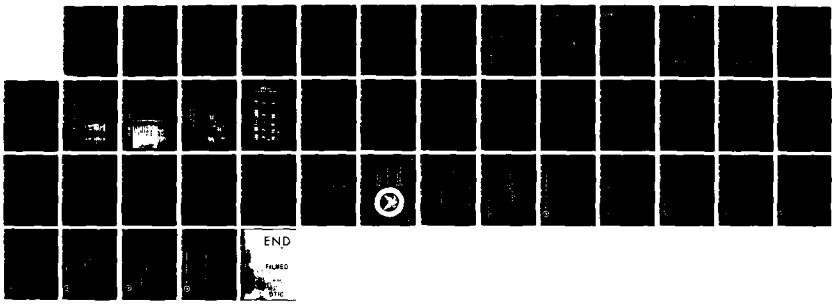
- STANDARDIZING GIVES THE POSSIBILITY TO USE MOST COMPETENT PEOPLE FOR DESIGN AND DEVELOPMENT INDEPENDENT OF WHICH FIRM THEY BELONG TO.

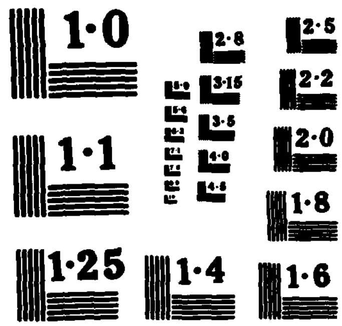
- STANDARDIZING GIVES INDUSTRY THE POSSIBILITY TO COOPERATE, LEARN FROM AND SPUR EACH OTHER.

- STANDARDIZING GIVES A LARGER BASE FOR SUPPORTING RELIABILITY AND QUALITY ACTIVITIES.

- STANDARDIZING SIMPLIFIES THE PROJECT MANAGEMENT TASK OF THE CUSTOMER, AND REDUCES THE WORKLOAD FOR MAINTENANCE- AND TEST-DEVELOPMENT.

AD-A145 697 PROCEEDINGS PAPERS OF THE AFSC (AIR FORCE SYSTEMS 6/6
COMMAND) AVIONICS STAND..(U) AERONAUTICAL SYSTEMS DIV
WRIGHT-PATTERSON AFB OH DIRECTORATE O..
UNCLASSIFIED C A PORUBCANSKY NOV 82 F/G 1/3 NL




1·0
2·8
2·5
3·15
2·2
3·5
2·0
1·1
4·0
4·5
1·8
1·25
1·4
1·6

COMPUTER ARCHITECTS

CONCEPT LAYOUT

GUNNAR CARLSTEDT, HYLAB

DESIGN REALIZATION

STELLAN MENNERFELT, SRA

D & D ORGANIZATION

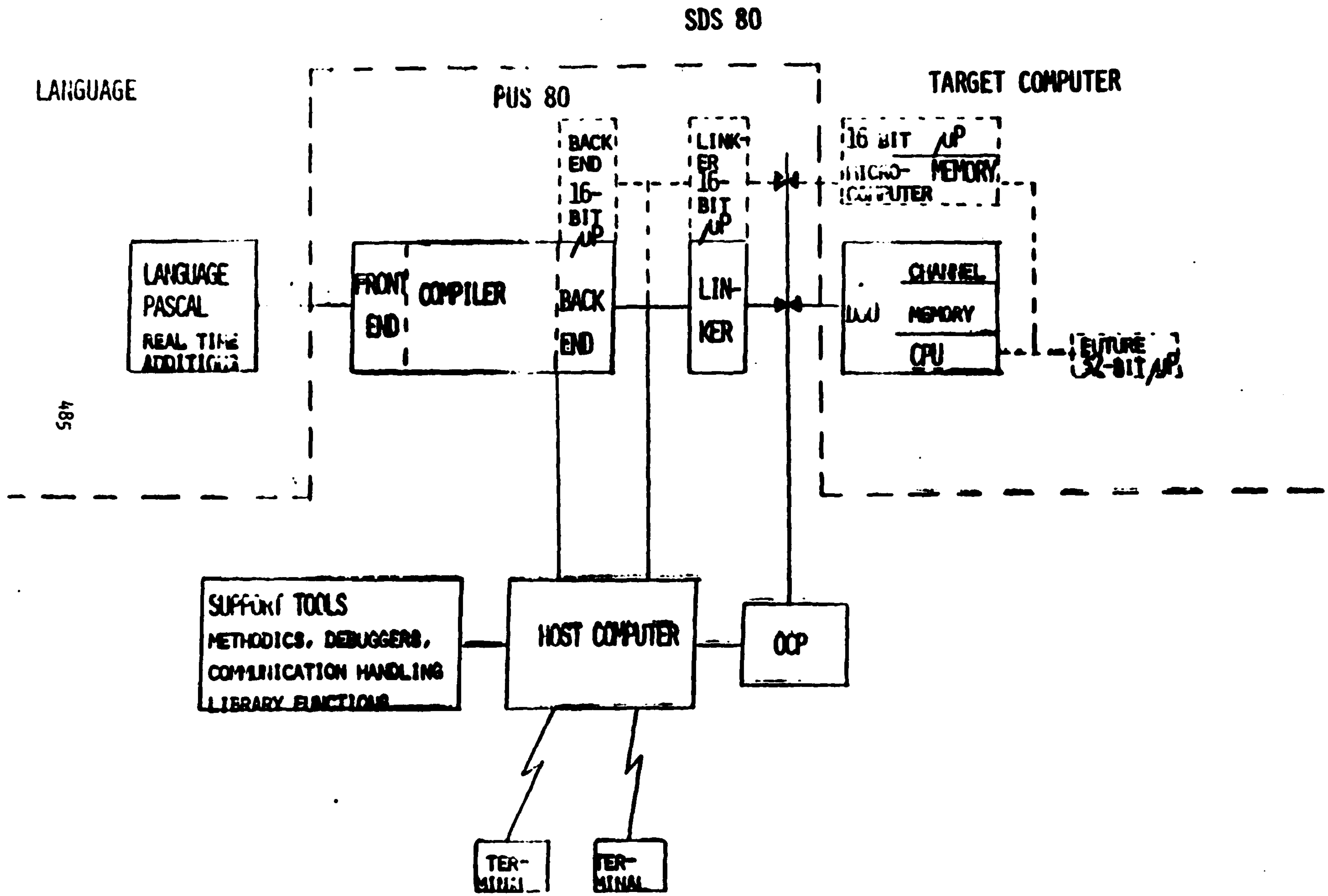
DATORKONSORTIUM 80

LM ERICSSON (PROJECT MANAGEMENT)

DATASAAB

SRA COMMUNICATIONS

(SYSTEMS PROGRAMMING LTD: HLL & COMPILER)


SDS 80
LANGUAGE
PUS 80
TARGET COMPUTER
LANGUAGE PASCAL
REAL TIME ADDITIONS
FRONT END
COMPILER
BACK END 16-BIT µP
BACK END
LINKER 16-BIT µP
LINKER
16 BIT MICRO-COMPUTER
µP
MEMORY
CHANNEL
MEMORY
CPU
FUTURE 32-BIT µP
SUPPORT TOOLS
METHODICS, DEBUGGERS,
COMMUNICATION HANDLING
LIBRARY FUNCTIONS
HOST COMPUTER
OCP
TERMINAL
TERMINAL

ATTENDANT STANDARDIZATION ITEMS

EQUIPMENT BOXES

ARINC 600 OUTLINE, 3 SIZES

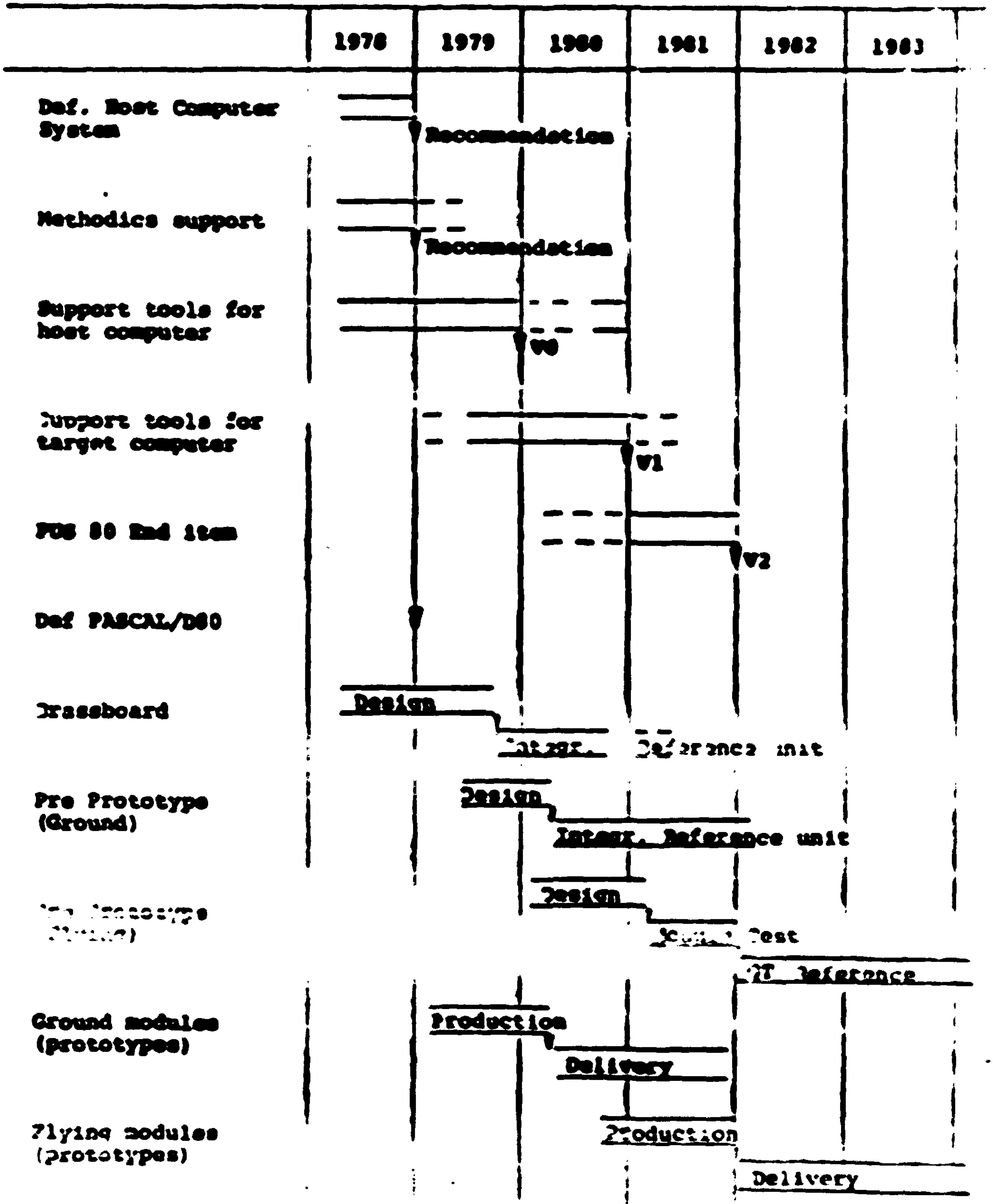
STANDARDIZED CARD DESIGN

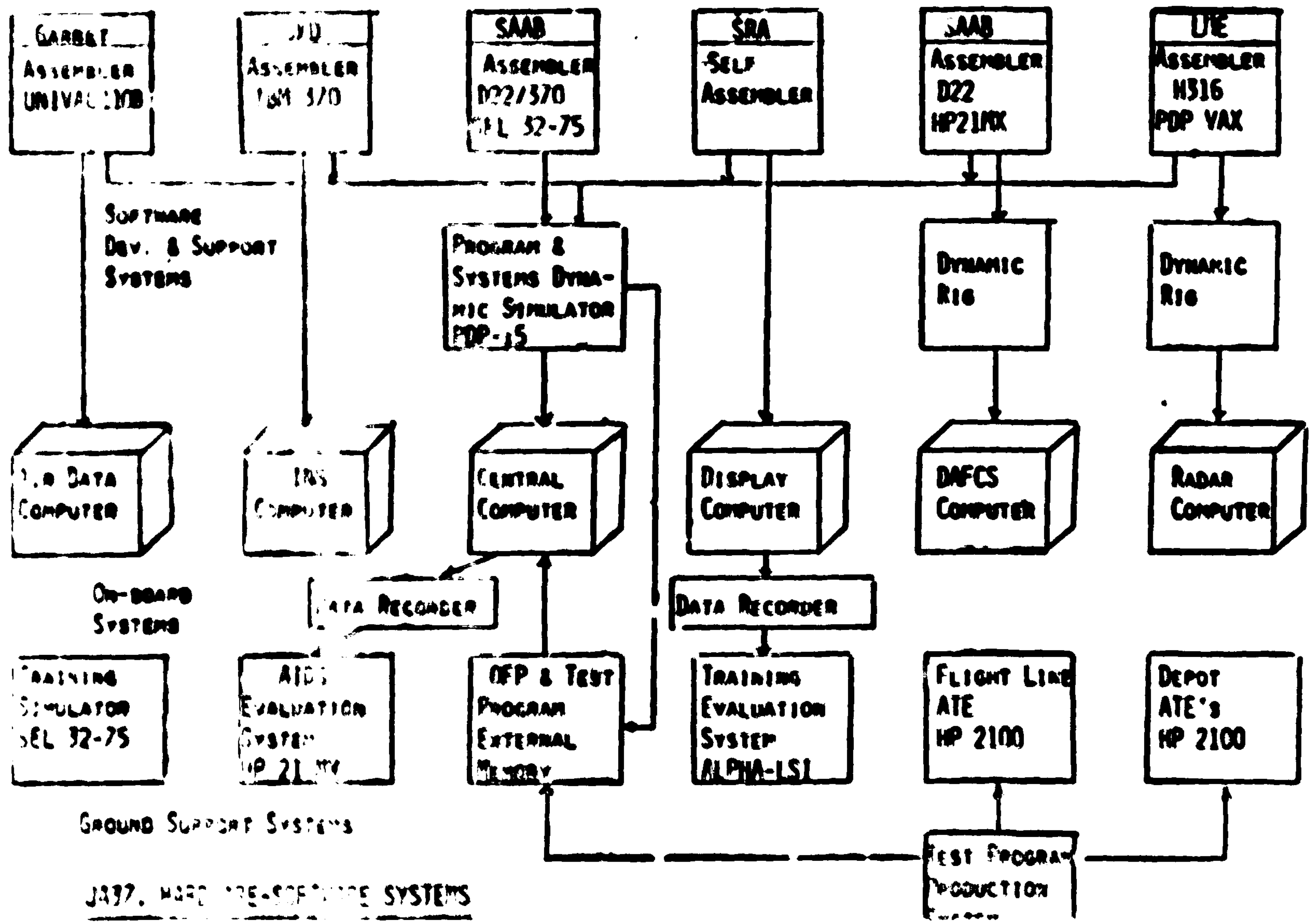
POWER SUPPLIES

2 SIZES: 200 & 350 W

STANDARD COMPUTER SYSTEM SDS 80

TIMESCHEDULE

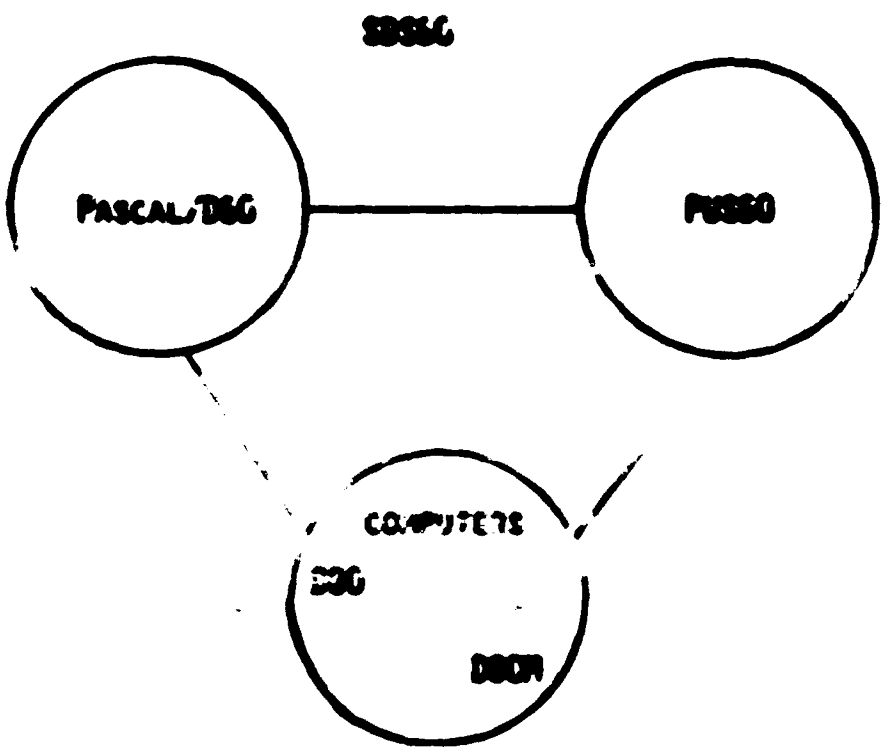




JA37. HARD WARE-SOFTWARE SYSTEMS

STANDARDIZED COMPUTER SYSTEM SDSBG

- HIGH ORDER LANGUAGE
- EFFICIENT PROGRAM DEVELOPMENT SYSTEM
- COMPUTERS SUPPORTING HOL



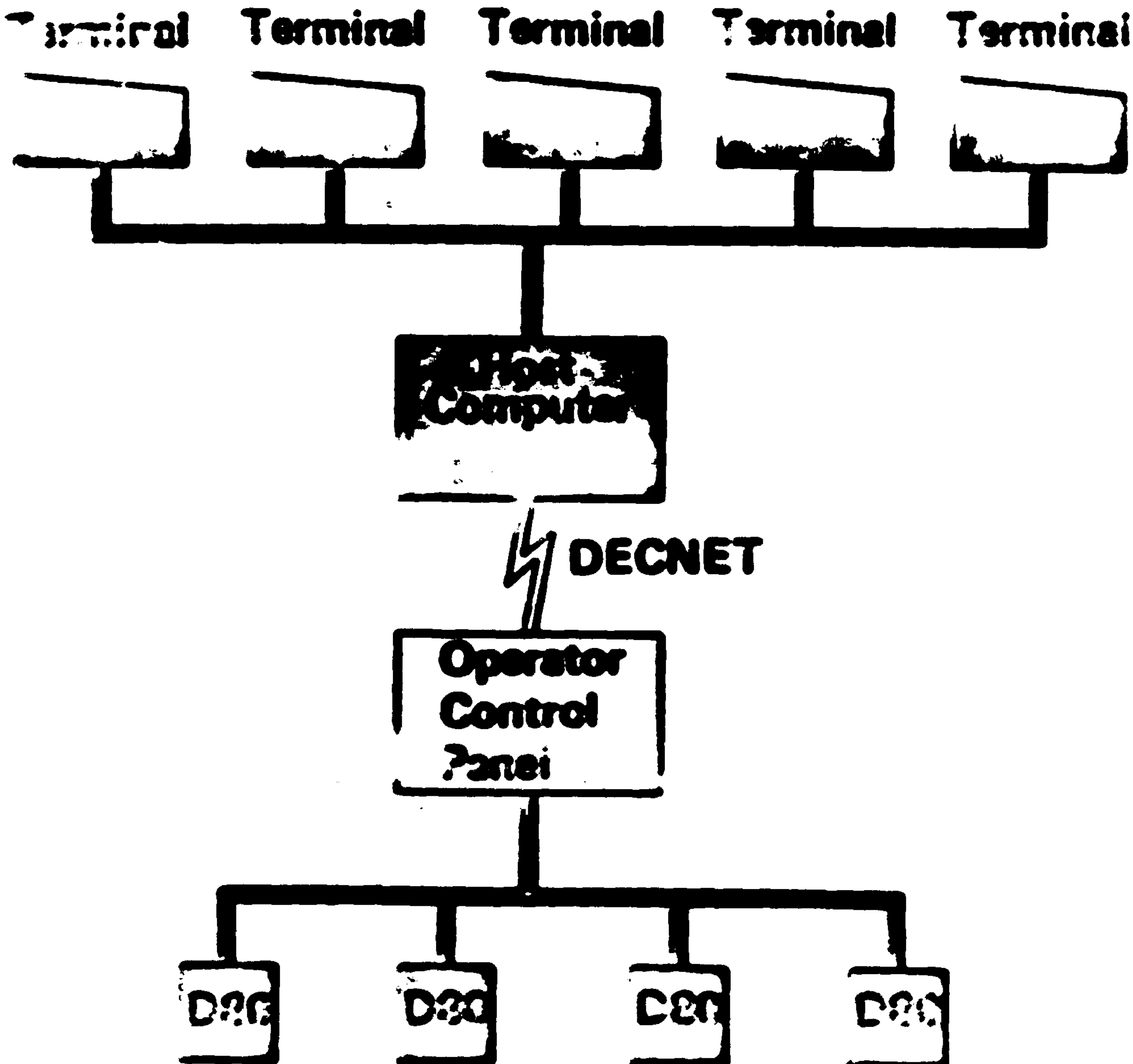
SDS 80 Standardized Computing System

The High Order Language
PASCAL/D80

- Based on standard PASCAL
- Improved for real-time systems
- Modular language
- Supports parallel processes
- Facilities for synchronization and communication
- Error handling

SDS 80 Standardized Computing System

PUS80
Program Development System



SDS 80 Standardized Computing System

Operator Control Panel (OCP):

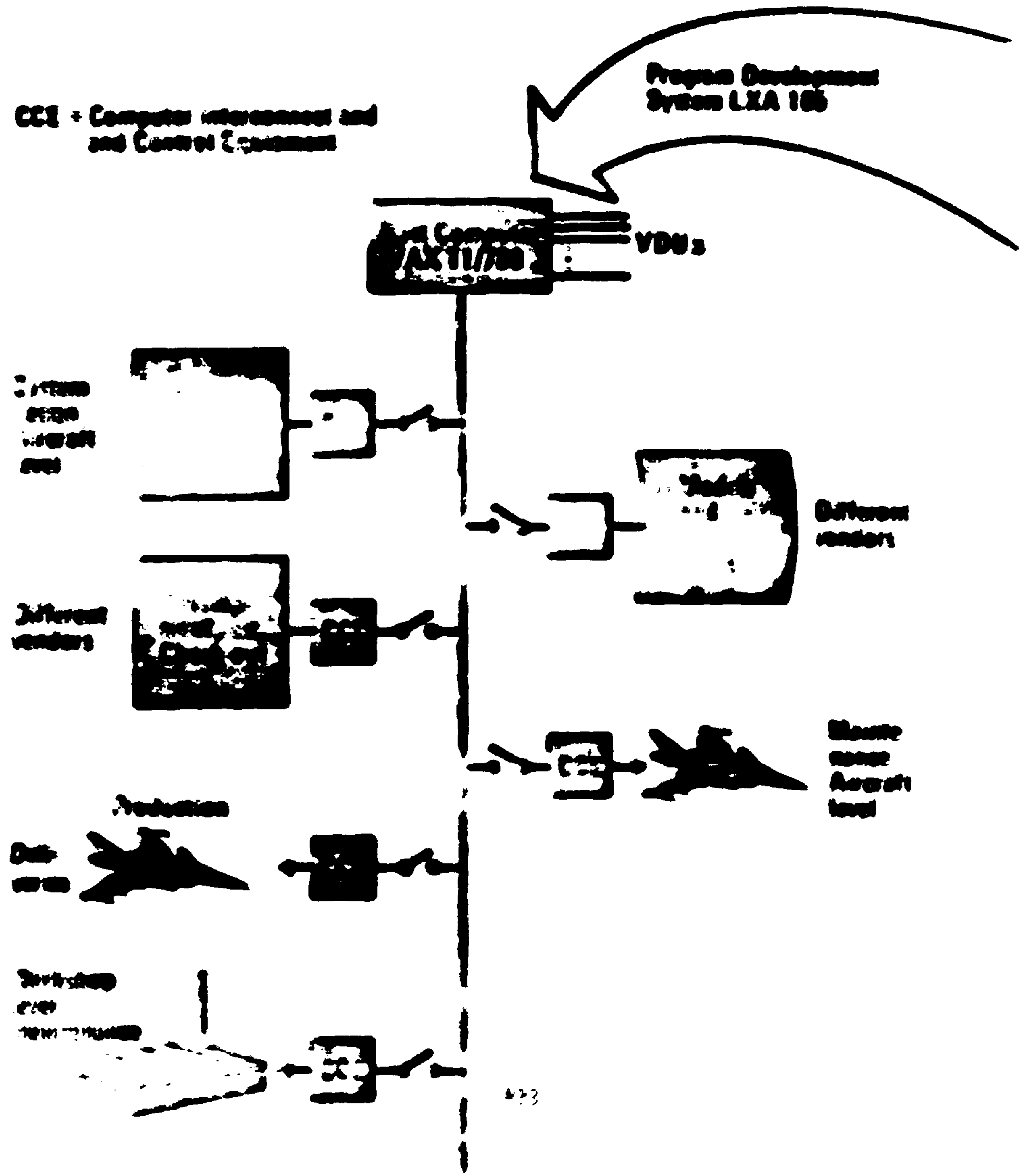
- Serves up to 8 D80 computers simultaneously
- Contains floppy disk
- Can be operated from host computer terminals

Major PUS80 Software Tools:

- PASCAL/D80 compilers and linkers for host and target
- Symbolic debuggers for host and target
- Editor
- Pretty print program
- System variable handler
- System variable lister
- Text change handler (Source Code Control System)
- I/O-channel program generator
- I/O-channel program translator
- Subsystem generator

Programming Support Environment

Standardized Computing System

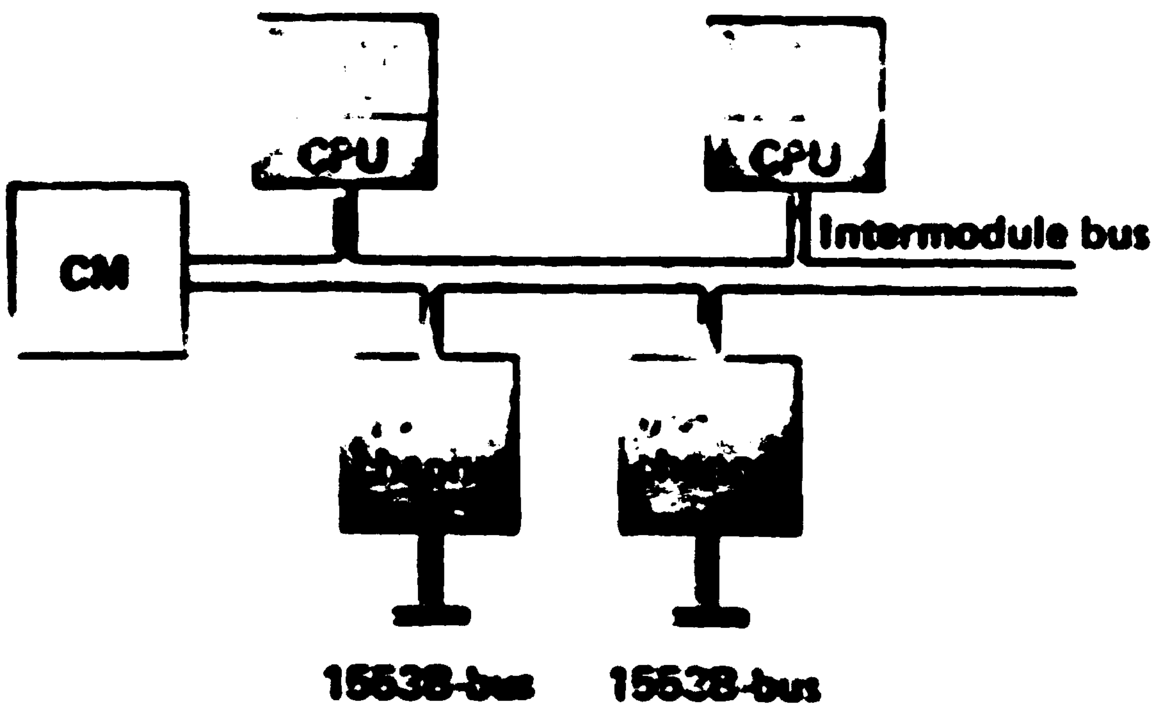


SDS 80 Standardized Computing System

D80 Computer

- Up to 7 processors and I/O-channels
- Common variable memory (CM)
- Intermodule bus

Several parallel processes per processor.



SDS 80 Standardized Computing System

CHARACTERISTICS OF THE D80 PROCESSOR

HOLM with support for parallel processes/multi processors
HOLM (High Order Language Machine)

- Stack machine
- Machine instructions reflecting HOL (High Level Language) structure (block structure, addressing)

Parallel process support

- **Hardware support for parallel processes**
- **Machine instructions for synchronization and communication between processes in the same or another processor**

Fast execution

- **Architecture supports HOL primitives**
- **Separate address calculation**
- **Separated program and data buses**
- Local data memory for fast data access
- Fast arithmetic and logic unit with hardware supported floating point

Result: Fast execution of HOL in a real time environment with a low operating system overhead

SDS 80 Standardized Computing System

Datorkonsortium DK80

The consortium DK80 consists of

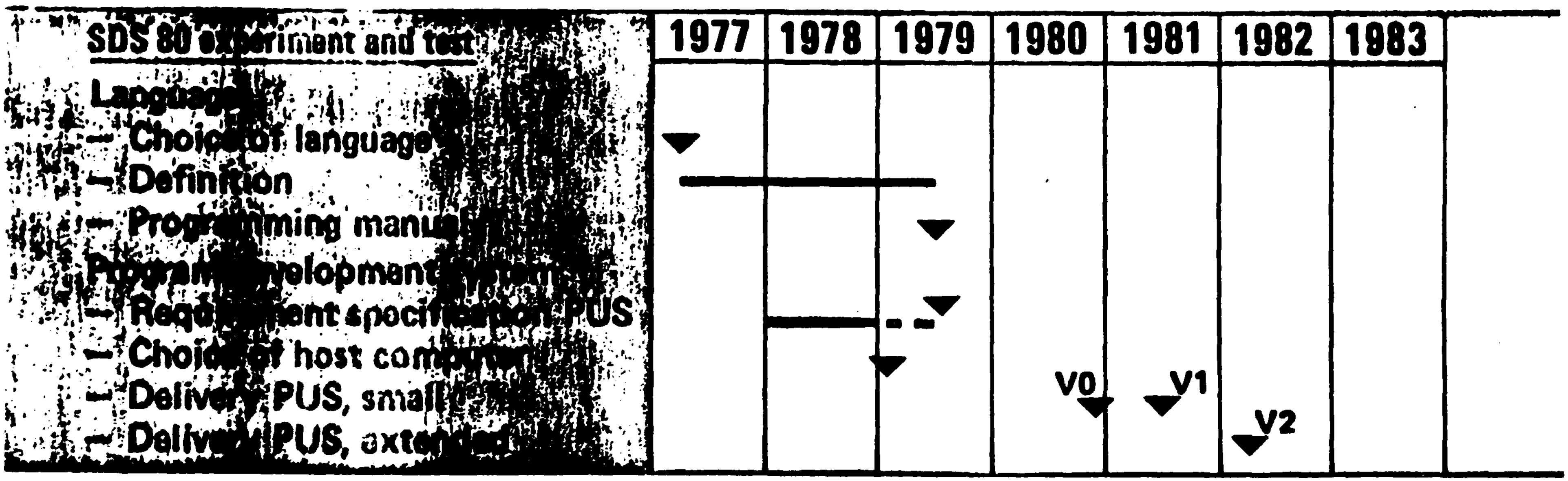
- **LM Ericsson, MI Division, Mölndal**
- **DATASAAB AB, Linköping**
- **SRA Communications, Stockholm**

Standardized Computing System, SDS80

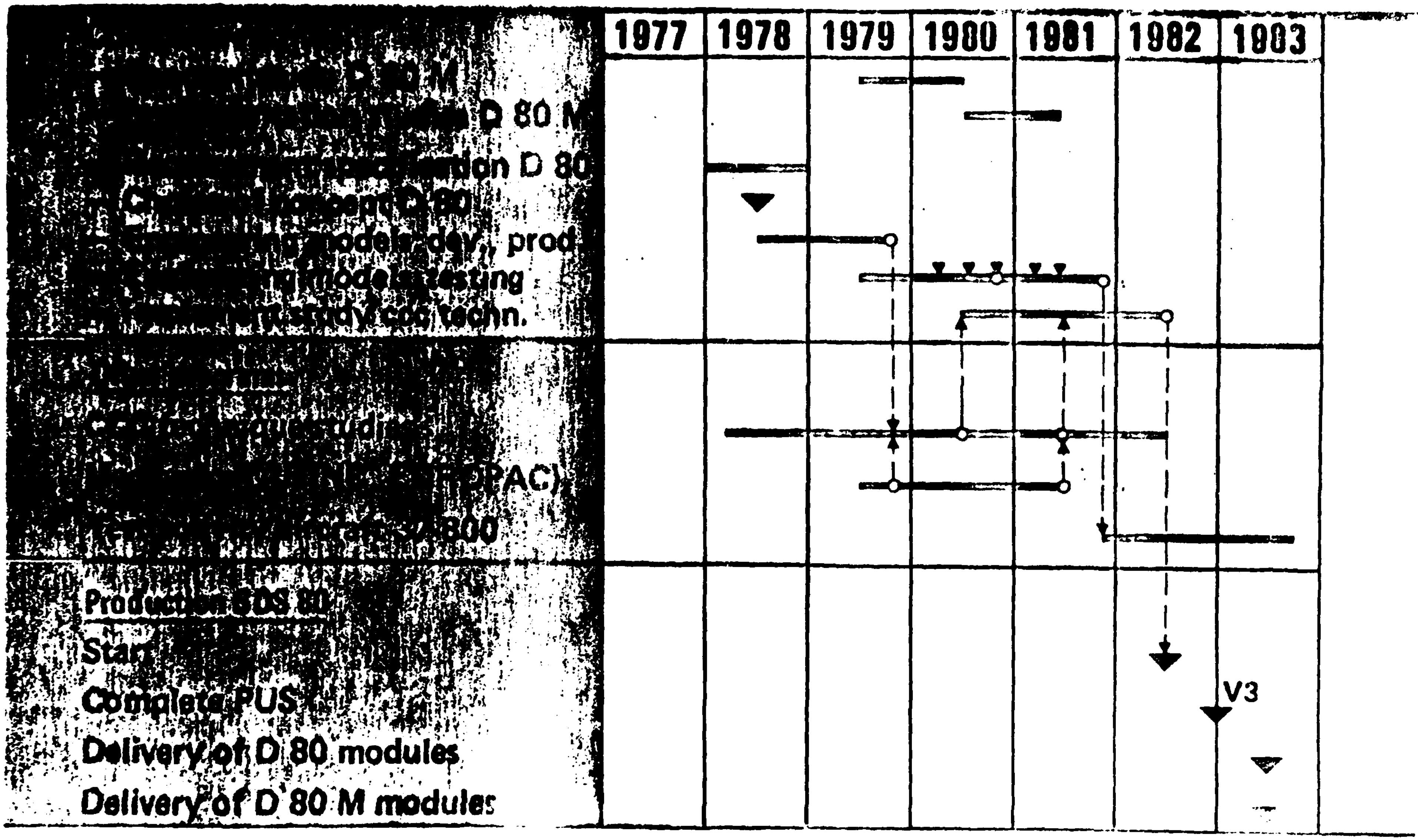
Consists of

- PASCAL/D80 high order language
- PUS80 program development system
- D80 multi-processor computer

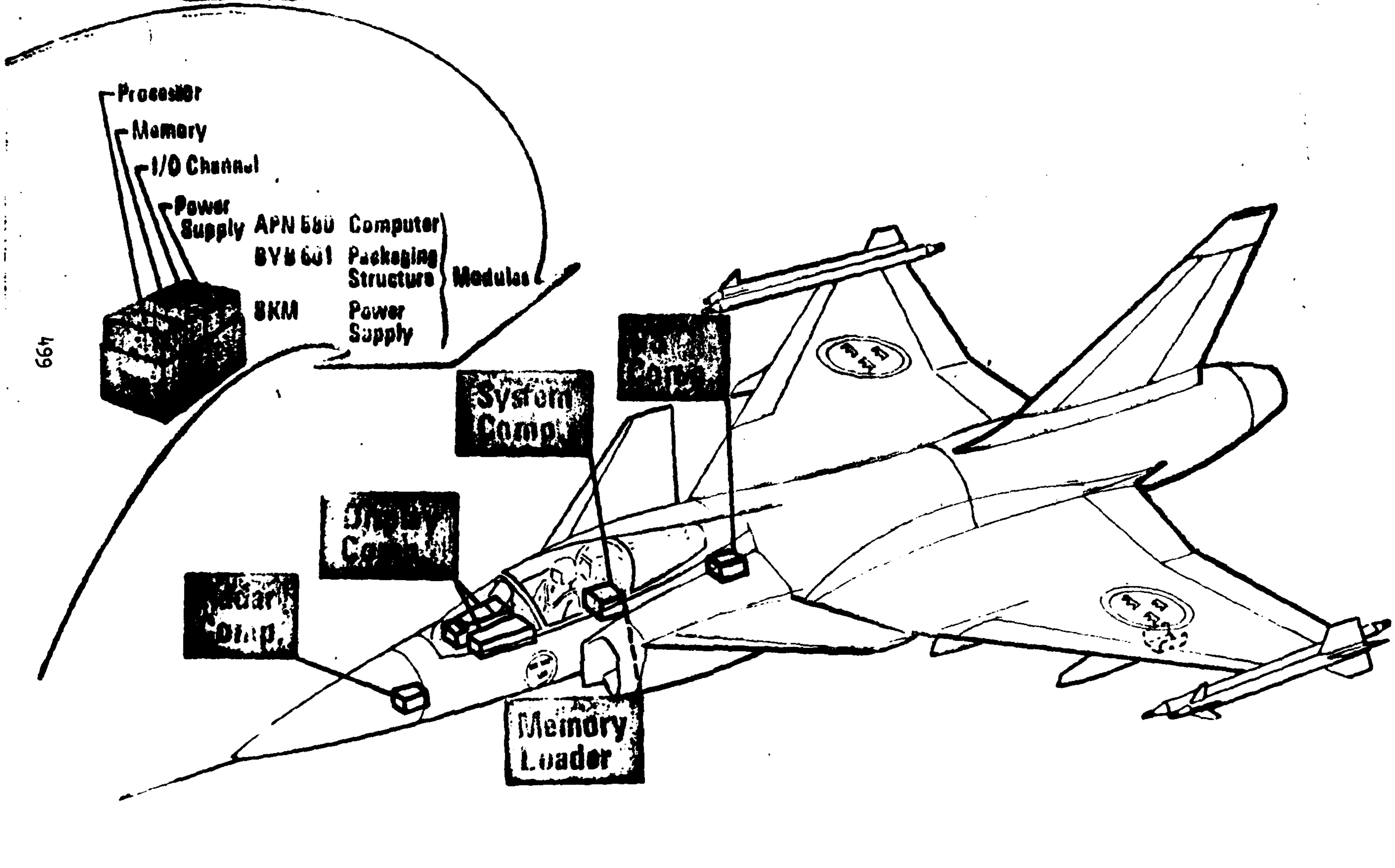
SDS 80 Standardized Computing System



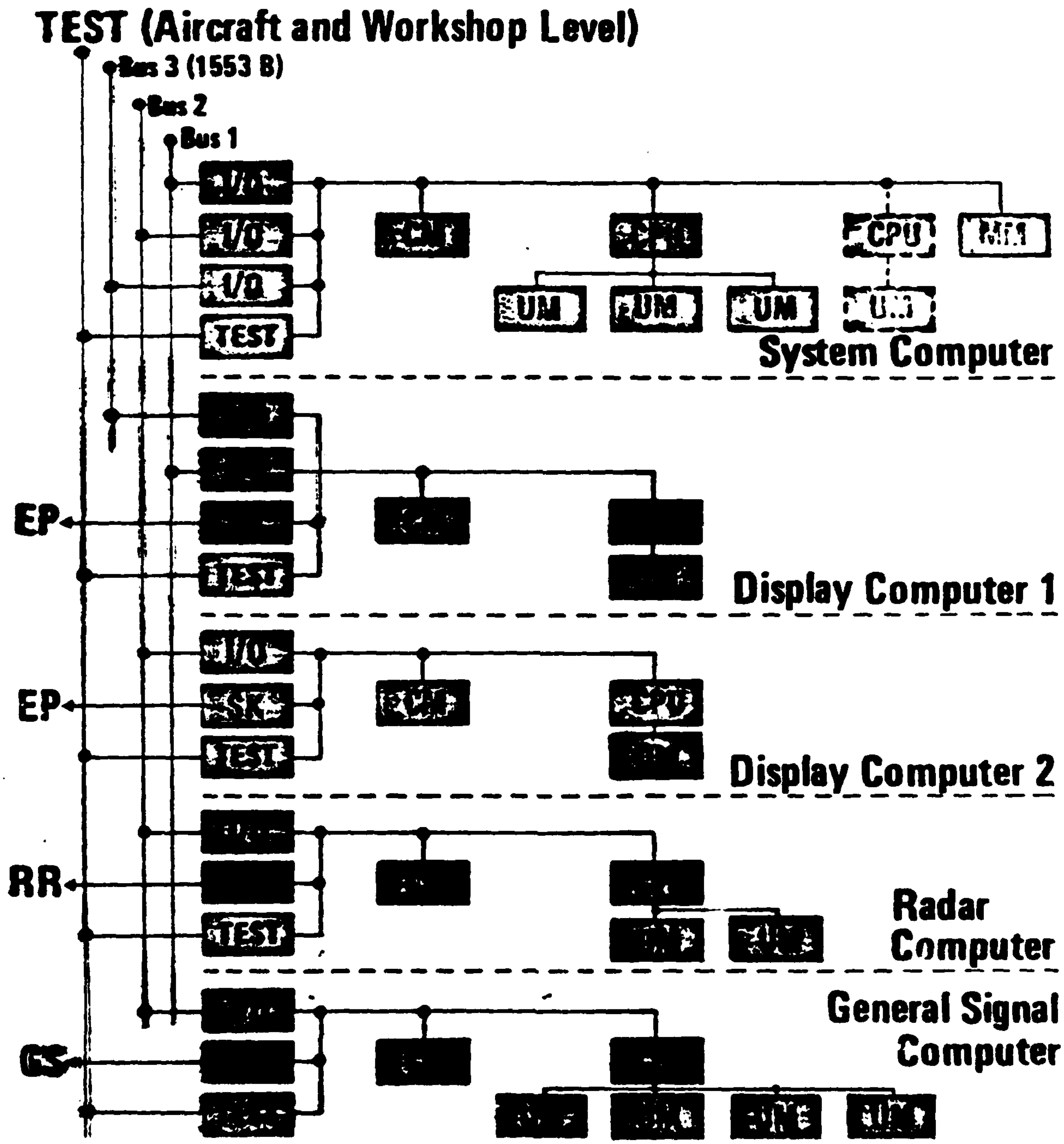
SDS 80 Standardized Computing System



JAS Aircraft
Standardized Computing System



JAS Aircraft
Standardized Computing System



Navy Real Time Signal Processor Development: Second Generation Planned Service Standard

C. B. Robbins

Deputy Project Manager, Navy Shipboard Tactical Embedded Computer Resources Project (PMS-408), Naval Sea Systems Command, Washington, D. C. 20362

Abstract

Planned growth in the coming decade of Navy combat systems will generate signal processing performance requirements that far exceed the capability of the current Navy standard signal processor. There is a further need to improve the programming environment of Navy standard signal processors to increase programmer productivity. The Navy has initiated development of a second generation standard signal processor, the Enhanced Modular Signal Processor (EMSP), nomemclatured as the AN/UYS-2. This paper describes the Navy program to develop the EMSP as a multi-processor signal processing system. The approach to specifying system performance and programming environment along with the acquisition approach that resulted in vigorous competition for the engineering development contaract recently awarded is discussed. The commodity management concept for EMSP's in-service lifetime involves interface management within the system and controlled technology infusion. This important plan to stay abreast of technology while meeting user community requirements for product stability is described.

Introduction

The Navy is committed to the use of standard computers and programmable signal processors in sea going and airborne weapons systems and sensor information processing systems. Extensive use of these Tactical Embedded Computer Resources (TECR), standard computers and processors embedded in larger systems, is the basis for one of the major strenghts our Navy enjoys. Aggressive acquisition policies are being pursued to add new members to this standard family. The AN/UYK-43 and AN/UYK-44 are being developed as planned standard mainframe, mini-computer, and micro-processor. The AN/AYK-14 is being developed as standard airborne mini-computer. The AN/UYS-1, the first standard programmable signal processor for shipboard and airborne applications is in production.

Even as the first standard programmable signal processor begins service life, a clear need to begin development of the second generation standard programmable signal processor is seen. Current processing capacity is limiting the effective aperatures of our sensors to less than that which the spectral, temporal and spatial coherence of progation media supports. Implementation of newalgorithms expected to mature this decade that process information from larger aperatures to cope with reduced target information must not be limited by programmable signal processor throughput or the economic limitations of non-programmable or difficult to program processors. Fleet introduction of these new information processes is essential to maintaining the qualitative advantage of our forces. Development of a next generation programmable signal processor to support sensor and algorithm improvements is being undertaken now as the AN/UYS-2, the Enhanced Modular Signal Processor (EMSP).

Requirements to meet the coming decade's processing needs and improve in-service support of standard signal processors have been establsihed for the EMSP. First, the EMSP must be a complete product line, five different sized versions will be available with five corresponding performance levels, sizes, power requirements, and weight. At the high end, at least an order of magnitude throughput improvement is necessary, 80 million real multiples and adds per second steady state performed in the context of the complex arithmetic in typical signal processing applications. Secondly, there is an urgent need to improve the programable signal processor programming environment. Machine level programming of programmable signal processors, the current practice, poses severe limitations on practical use of signal processing power of new technologies and architecture. The direct relationship between lines of machine code and number of gates to be programmed requires major improvements in programmer productivity if signal processor hardware advances are to be fully utilized.

Through the course of development of the AN/UYS-1, the Navy has slowly advanced from machine level programing to partial use of a High Order Language (SPL/I). Most programs remain assembly code programs. Although a true High Order Language exists architectural constraints of the current standard result in applications code that is not machine independent. Systems programmers are required to effectively program the current standard. The small number of applications engineers with this programming expertise limits productivity. There is a requirement for an applications code that is not machine independent. There is a requirement for an applications oriented interface to the users at a level of abstraction above the HOL, a problem oriented notation system. This notation system is to be part of the machine/language independent application programmers interface. Production enhancing environmental tools based on the Navy's Machine Transportable Support Software (MTASS) tools must also be made available for programable signal processor program developments.

Thirdly, there is a requirement to develop an in-service commodity management concept that accommodates the conflicting goals of providing long term product stability for economic and logisticsupport reasons and following a program of agressive technical infusion during service life. This requirement dictates the recognition of EMSP as a system rather than a device. Interfaces between architecturally significant elements must be formally managed to allow technical infusion in independent elements. The software interface to the applications programmer must isolate applications programs from hardware or system software upgrades. And, system upgrade must be accomplished on a module replacement basis to decouple EMSP upgrades form ship overhaul periods.

Finally, there was an absolute requirement for competitive selection of the engineering development contractor. Because the Navy standard computers and processors, once in production, are used in all systems with processing requirements as a matter of policy, development contracts must be competitively awarded.

The acquisition approach taken to selecting an engineering development contractor has worked well, beyond Navy expectations. Seven teams composed of twenty-one prominent firms submitted proposals. Five of these firms were selected to participate in a *Machine Definition/Proof of Principle testing phase.* Competitors developed a Principal of Operations (POPs) document describing EMSP to the level necessary to program it at the assembly level and performed proof of principal demonstration necessary to satisfy preliminary Navy technical testing requirements. Western Electric Company was selected from this highly competitive field.

Technical Approach

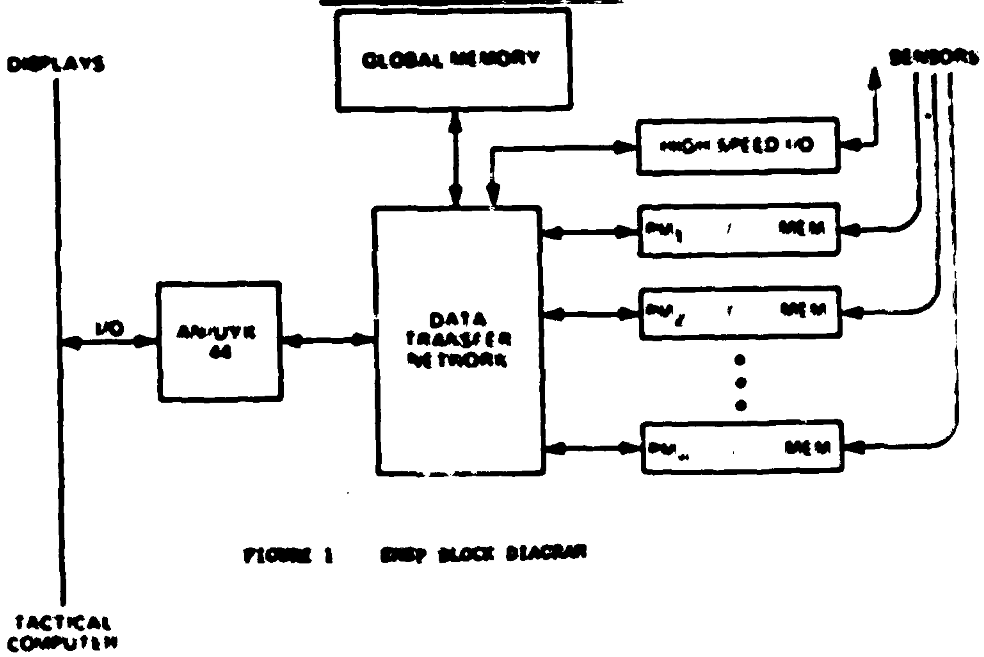


FIGURE 1 EMSP BLOCK DIAGRAM

This section presents the technical approach to EMSP. The technical *approach was developed to encourage industry participation* and innovation.

Figure 1 is a block diagram of EMSP. It was used by the program office to present the EMSP architectural requirements, not support a particular architecture. The parallel set of processing modules consist of programmable array processors, special purpose single function devices, and high speed input/output modules. The number and types of modules in parallel and system performance will vary from one proposed system to another. The effective agregate of computational power of the set is the system's performance. Sensor data is input to these modules. The high speed I/O module interfaces the system to combat system data busses or other devices requiring data rates that exceed those provided by NTDS standard interfaces. A global memory is shown as one of the major architectural elements. This memory is to be accessable by all processing modules. The Data Transfer Network (DTN) shown as the center block is the physical interconnection of processing modulesand memory and the control processes and processors that control module operation and intermodule data transfers. It is in this area of networking in which the Navy is properly taking what is considered measured technical risk and expects significant payoff. There were a number of excellent array processors available to the Navy in implentations that permit their stacking in large numbers in a single cabinet. The cumulative throughput of that stack would more than meet the performance required. Realizing an effective system level of performance requires agregating individual module performance by the DTN without unacceptance degradation caused by network contention or control problems.

The performance of the system has been specified in considerable detail in terms of a set of benchmark algorithms. Input bandwidths have been specified to establish computational performance. Algorithms are:

a. <u>Inter Array Processing</u>

This algorithm produces large, two dimensional ambiguity surfaces from independent input data streams. Peaks in sequential ambiguity surfaces are tracked to establish convergence to mean values of surface parameters and uncertainty area about mean values.

b. <u>Adaptive Beam Forming</u>

Individual sensor data from an array of sensors is constructively added to form a directional array response. Sensor phase and amplitude weights are adaptively modified to steer side lobe nulls at directional interference. Modifications are constrained to prevent unacceptable main lobe degradation.

c. <u>Sonobuoy Processing</u>

Sensor Data from a large number of sonobuouys is filtered through 8 octaves. Each octave of each sonobuoy is spectrum analyzed in parallel paths.

d. <u>Sonar Processing</u>

Sonar data from a multi element sonar array is beam formed by a linear beam forming process. Beam data is spectrum analyzed in wide band and narrow band spectrum analysis paths.

e. Wide Band Intercept Processing

Very wide band data from a single sensor is spectrum analyzed in wideband and narrow band paths. Energy detection in either path initiates finer grain signal analysis.

Each algorithm is specified in signal flow graph form, and computational operations at each node are specified.

The set of benchmark algorithms establishes requirements for data flow, scheduling processing tasks, computations, intermodule data transfers, and memory loading performance for the system. This set was meant to stress the design in ways representative of anticipated applications of EMSP. The downsized members of the EMSP family are specified in terms of proportionally scaled sets of the Benchmark algorithms.

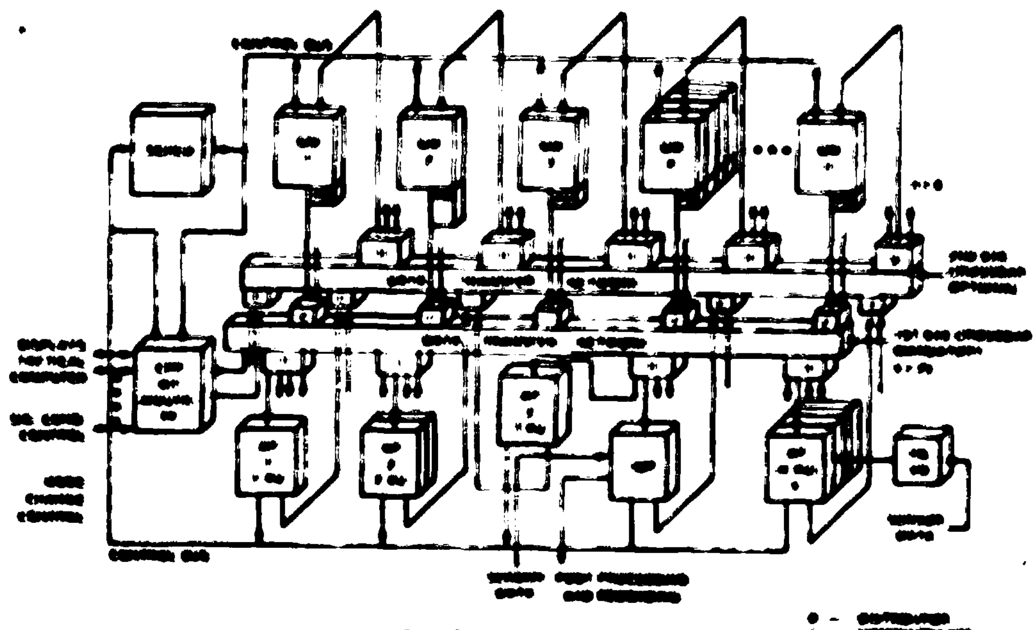


FIGURE 2

Figure 2 shows the specific EMSP architectural approach. Independent memory modules are connected to processing modules through a pair of non-blocking switching networks. The paths are 32 bits wide. At the 5mhz clock rate, 16 bit words are communicated at a 160 mhz rate. The set of processing modules have an aggregate of 300 m mult/sec burst rate processing power of which 150 m mults/sec plus can be realized in steady state processing benchmarks. Scheduling is provided via a separate control bus. The system can schedule and dispatch 40 knodes/Sec signal processing tasks (FFT, Average, Correlate, etc.) which far exceeds benchmark requirements.

Software Approach

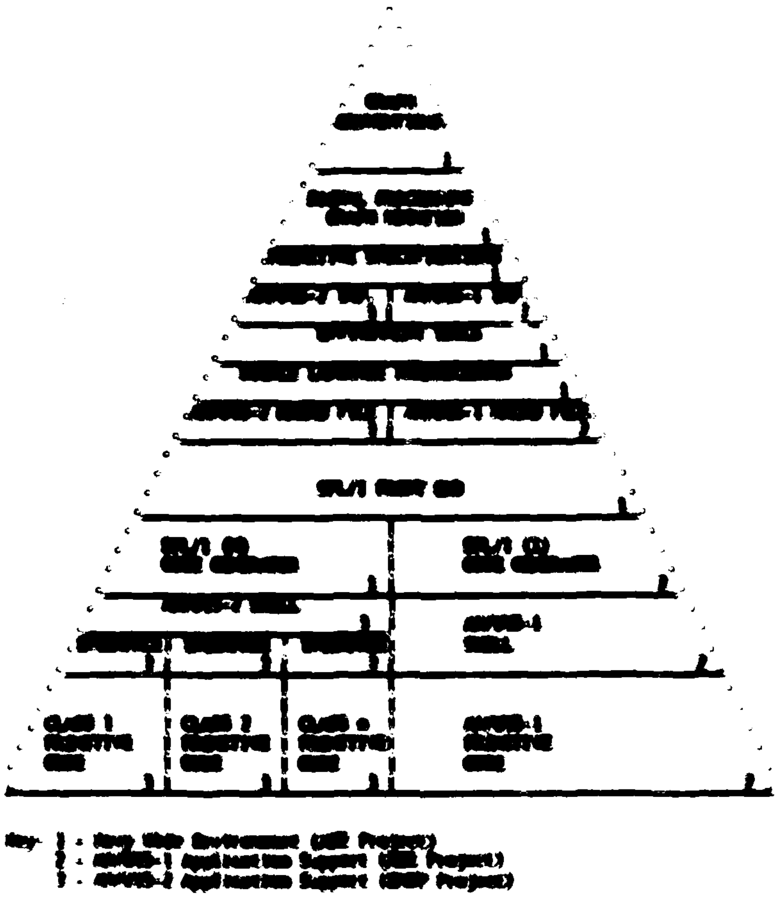


FIGURE 3

The Navy is developing an upgrade to the Navy standard signal processing environment, the ACOS/ECOS environment. This environment common to both the current standard and EMSP will support transport of applications programs from the current standard to EMSP at a level above machine dependence. Figure 3 depicts this environment. The programming methodology is based on a directed task graph method for specifying applications programs, a problem oriented notation set for linearizing the application graphs into machine readable form, a library of array processor programs (primitives) for each signal processing task specifiable other program development tools, a preprocessor to translate the notation into SPL/I the Navy's signal processing HOL, the SPL/I compiler, a run time program called shell that schedules and dispatches nodes of the graph based on availability of input data to nodes (i.e., a data flow scheduler) and AP micro code for each AP primitive.

Applications programs are specified in graphical form and hand coded in graph notation. The graph notation can be machine processed to executable code on either AN/UYS-1 or AN/UYS-2 coding at this level represents a reduction of an order to magnitude or more in the number of lines of code to be hand coded. Since it ismachine independent, applications readily transport. Using graph notation to code application programs at a level of abstraction above HOL provides compression of the number of lines of user code to be written and managed by more than order of magnitude.

Figure 4 shows a typical ECOS sub graph, part of the sonobuoy processing algorithm. This subgraph is part of the graph shown in figure 5 which defines an entire process. In this example, the data source is connected to the AGC node and the AGC node is connected to a sequence of octave filter nodes. Each Octave filter node is connected to the next octave filter node in the sequence and outputs to another sub graph as well; the final queue (UNUSED) is the bit bucket. The AGC node executes when the to_AGC queue exceed threshold value and outputs to the INTO8 queue. After several executions of the AGC node, the INTO8 queue data accumulation will exceed threshold and the OCTAVE primative executes on the data in that queue. The data is filtered into half bands; the upper half band is outppt to the OCTAVE8 queue and the lower into the INTO7 queue. Successive octave nodes execute as their respective input queue threshold values are exceeded.

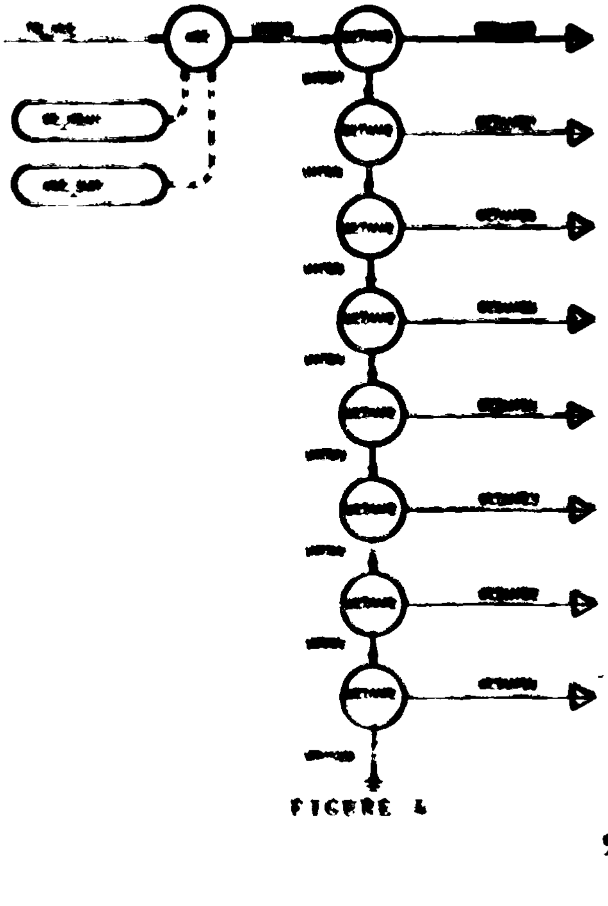


FIGURE 4

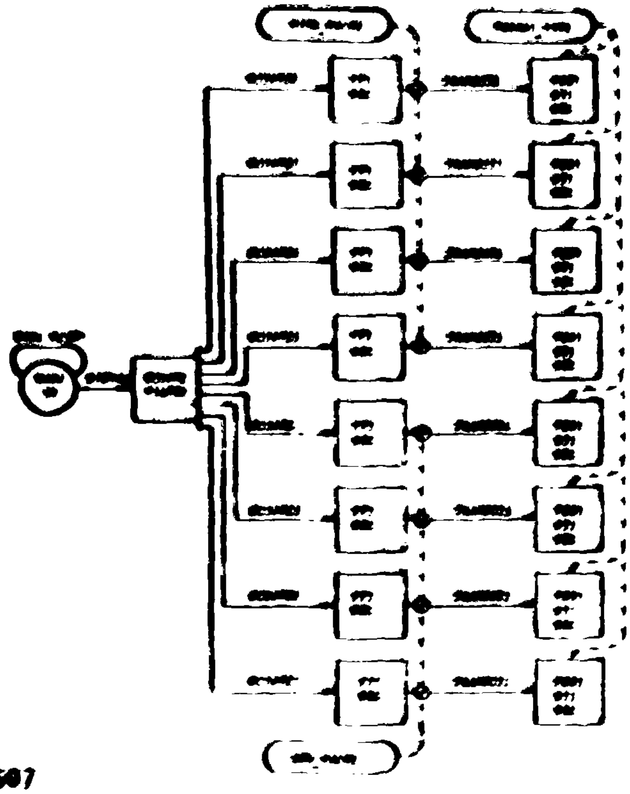


FIGURE 5

engineer. The HOL (SPL/I) is the application/machine interface. The SPGN is language independent as well and will support expansion into Ada as well when the Ada environment is available.

The EMSP support software approach is built around this ECOS (EMSP Common Operating Software) concept. The graph specification rules, notation (SPGN) and HOL (SPL/I) are common with that being developed for the current standard. The EMSP applications programmer is provided, as an interface to the machine, a library of primitive operations available, graph notation system (SPGN) to represent graphs in machine readable format, and a HOL programmingenvironment to program graph execution control programs. These tools (some already MTASS(M) components which supports Navy 16 bit computers) will be part of the ACOS/ECOS environment. This code generation software MTASS(M), is hosted in many commercially popular machines along with other software programming tools (librarians, test editor, linking loaders, etc).

Requirements on the EMSP developer are for support of the ECOS compile time environment, an operating system that implements the data driven graph execution concept for the system, machine level coding of the primitives, system software to provide fault detection, isolation and system reconfiguration control software, and various items of product support software. The Navy is specifying compile time software to be the Navy standards and established performance requirements for run time software. The operating system requirements are being met with an event driven operating system in which both graph events and machine events are processed to determine node schedulings. These schedulings are dispatched to the appropriate processing resource. Task schedules are queued within the processing resources and executed as the processing modules complete previous tasks.

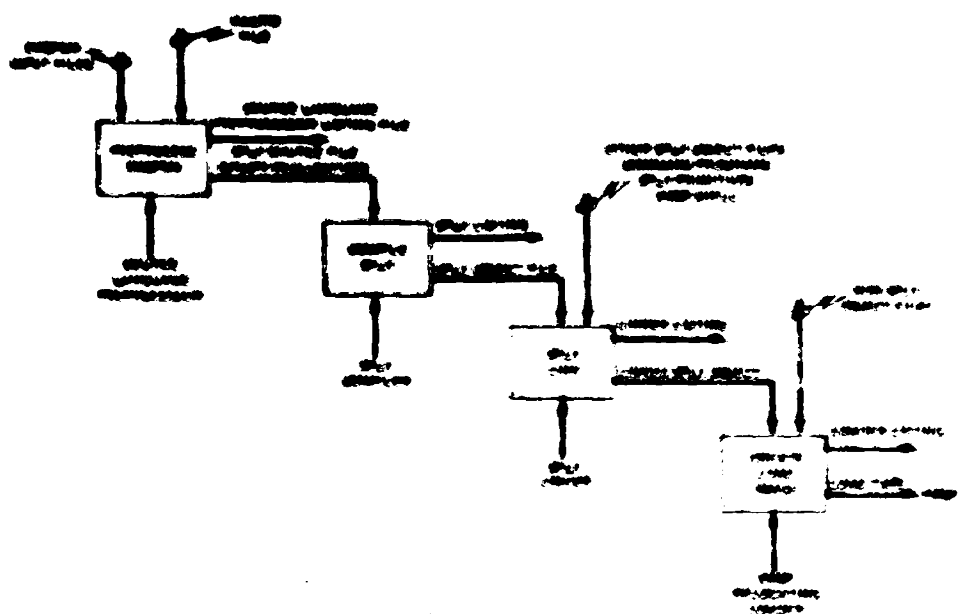


FIGURE 6

Table 1 lists the SPGN that specifies this subgraph to the preprocesor. Although notation conventions have not been covered in this paper, The notational representation of the graph is self-explanatory when compared with the subgraph. Each of these node statements expands into 12 lines of SPL/I, the HOL. These 12 lines of SPL/I expand into approximately 180 lines of machine level control code for the current standard.

[illegible]

[illegible]

The operating system of the current standard is being modified by the addition of a data flow based scheduling and dispatching program. This algorithm called a SHELL, monitors data accumulation in data queues (including trigger queues) and schedules execution of nodes when data queue threshold values have been reached. The resultant is a data driven operating system asynchronously executing the nodes specified by signal flow graphs with the Signal Processor Graph Notation (SPGN). The user interface SPGN is machine independent and oriented towards the expertise of the applications

The compile time environment, part of the programming environment, is shown in greater detail in figure 6. Machine processing of application programming inputs commences with the input of graph notation to the preprocessor. A macro file for the particular target, EMSP in this case, must be utilized. Source code is generated and input to the compiler. The compiler outputs object files to the linker. The linker accepts object files for the operating system program, the SHELL, SPL/I primitives and the SPL/I command program which have been previously compiled. Linked non-SPL/I object files such as micro coded primitives are input as well. A load tape is then generated for EMSP.

The EMSP operating system, the EMSP SHELL, is considered at the heart of the EMSP development and inseparable from the physical architecture. The SHELL efficiency in supporting data driven task execution will determine how much of the theoretical maximum machine throughput is realized as effective throughput.

Interface Management

Commodity configuration management of EMSP during its service life will be based on establishment of standard interfaces and protocols within the system and strict configuration management of those standards during service life. Management of technology infusion will be based on upgrading architectural elements between interfaces and requiring upgraded modules be design to meet existing interfaces.

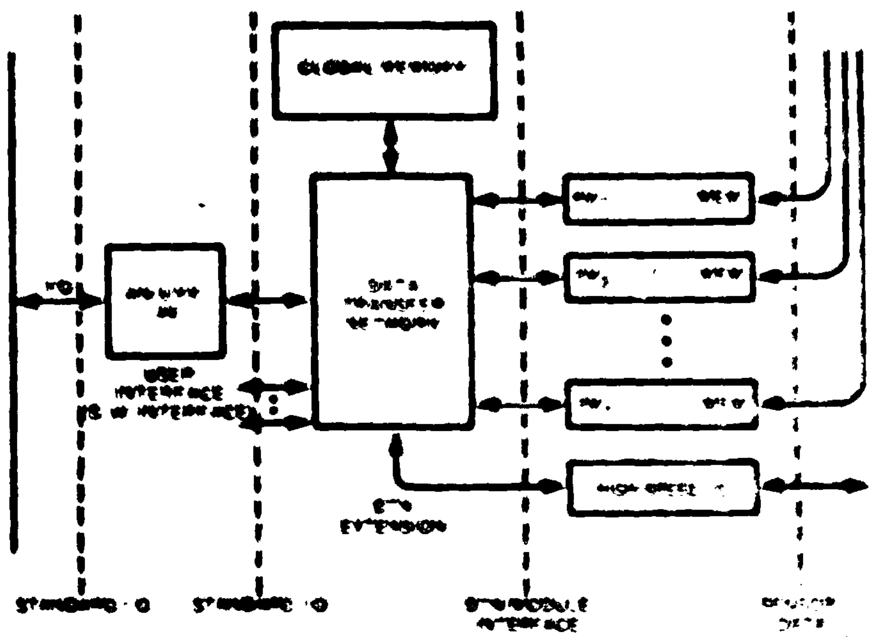


FIGURE 7

Figure 7 shows the EMSP architectural block diagram with interfaces to be configuration managed identified. The interfaces to the sensor data are expected to be a reasonably small set of interfaces. As new applications require interfacing EMSP to new sensor types or other devices via a high speed I/O additional interfaces will be developed. The interface between the Data Transfer Network and the set of Processing Modules is of utmost importance. As new applications require new types of special purpose processing modules, new modules will be designed to meet the existing interface and incorporated into the EMSP system. Technology upgrades of processing modules shall be accomplished such that the upgraded module meets the existing DTN, interface and protocol. Back fit of upgraded, modules may be accomplished conveniently without expensive equipment rip out of reprogramming of applications programs. Recent congressional language directed the EMSP program to insert VHSIC technology and conduct proof of principle demonstration to validate the concept of VHSIC technology insertion. The Navy will address VHSIC insertion at the chip level and the module level. VHSIC chips will be incorporated into appropriate modules as VHSIC chips became available. Navy VHSIC brassboards will be reimplemented in the EMSP brassboard from factor and interfaced to the EMSP DTN. These VHSIC brassboards will be integrated into EMSP as front end modules. Operation of benchmarks at much increased performance levels will be demonstrated. The interface to the support computer will be one of the standard NTDS interfaces. Upgrade of the DTN when technology supports such an upgrade must be accomplished to meet both the processing module interfaces and NTDS interfaces. The interface to the user, the applications programmer is the software interface previously discussed. It is intended to pursue an agressive technology infusion program during the service life of the EMSP system. As expensive as technology upgrades and system software changes are, they are bearable as non-recurring costs of keeping abreast of technology and ahead requirements. Recurring costs of application software changes for each upgrade must be avoided. Upgrade of programmable processing modules will require new primitive microcode to be generated and operating system changes, but not changes in applications programs. From the users point of view, the EMSP system should remain stable.

Reliability and Availability

The high degree of parallelism in the architectures of candidate EMSP's will support development of operational availabilities that far exceed present levels. Support costs, the cost of spare parts and repairs will remain a function of fundamental hardware reliability. Both attributes are important to EMSP.

Fundamental hardware reliability is being specified as a MeanTime to Fault (MTTF) with faults defined as any hardware condition requiring a logistic action, i.e., repair, spare part consumption, etc. All faults, whether their repair is deferred or immediate, are counted. Grooming equipments for extended missions by replacing cards with built in redundancy because of fault of one or more of the parallel paths is a deferred repair.

Operational availability is specified in terms of a Mean Time Between Failures of the system (MTBF). Failures are those single faults or aggregation of uncorrected faults that cause the system to operate at less than full capacity. MTBF must equal or exceed MTTF. Those architectures which have signicant redundancy in architectural elements in reserve will have MTBF's far exceeded MTTF. Two levels of degraded mode operation are specified, an 80% or 50% performance level. These levels mean that all application programs can be executed at the specified percent of the input bandwidth. The Mean Time Between Failure below the 80% and 50% performance level of the system is greater than the MTTF and MTBF for full performance system failure. A final availability to be specified is availability of a particular application program. This quantity is dependent on the amount of extra capacity available for reconfiguration in the event of a fault for a particular EMSP configuration and application program. This is specified in terms of Mean Time Between Critical Failure (MTBCF). A critical failure is defined as a fault or aggregate of faults that causes a particular application program to be unable to execute at its full performance level. An application program that loaded the EMSP 50% would sustain a critical failure if the system degraded below the 50% performance level. The MTBF for 50% degraded performance level would equal the MTBCF in this case. Since a variety of configurations will be available to users, including multi EMSP systems, application engineers may control the availability of their program through designing in over capacity for their particular application programs

Acquisition Approach

The constraints of supporting the first users concurrent weapons system development, the technical risks involved, and meeting the normal requirement for competitive selection of the engineering development contractor, dictated a non-traditional acquisition strategy. The Navy is has completed a two phase extended service selection process which resulted in the selection of a single engineering development contractor. The first phase was a normal proposal phase; the participation of all potential contractors was been solicited. As mentioned previously, a variety of system architectures were proposed. Offerors were required to identify the technical risks involved in their particular approach and propose Critical Item Demonstrations (CID's) to demonstrate manageability of technical risks in engineering development. These CID's were proofof principle type demonstrations on partial brass boards with supporting analyses. An outline of the Principles of Operations (POPs) was proposed as well.

For the second phase of the extended source selection process, the Navy awarded five contracts for performance of the proposed CID's, completion of the POPs, and delivery of plans and studies such as hardware and software development plans and reliability analyses for evaluation. Based on the POPs, the plans and studies, and the Critical Item Demonstrations, the Navy selected the single engineering development contractor, Western Electric Company.

Major Engineering and production milestones are listed below:

Engineering Development Contract Award -- August 1982
Delivery of System Engineering Facility (SFF) -- August 1983
Delivery of Functional Development Model (FDM) -- March 1984
Delivery of Engineering Development Models (FDM) -- Sep 1985
Availability of Advanced Production Equipments (APE) -- 1986
Availability of Production Equipments -- May 1987

The System Engineering Facility (SEF) is a software simulation of the POPs which is hosted in a commerical machine. The Functional Development Model (FDM) is non-militarized version of the EMSP, a POPs emulator. The Engineering Development Models (EDM's) are fully militarized units delivered for Navy acceptance testing. Advanced Production Equipments (APE's) built to EDM specifications will be made available to users for system development. Upon completion of Navy acceptance testing of EDM's any discrepancies found will be corrected in APE's upgrading them to production quality machines.

Software deliveries are scheduled to commence 6 months after award of the engineering development contract and continue through delivery of the FDM's.

Testing

A vigorous program of testing is planned during engineering development with the goal of supporting an early decision to authorize a limited amount of pilot production in advance of authorization of unlimited production by the Navy. This limited production authorization is important to accomplishing fleet introduction of a new processor on an economical basis.

Using the SEF as a vehicle, software Validation and Verfication (V&V) will be initiated early in the development. An independent first users group will be formed within the Navy Project management team. This team will program the benchmark algorithms for execution on the FDM's. Algorithms from selected in service systems as well will be programmed using the FDM. This group will do much of thesoftware and functional debugging that in the past has fallen to the first combat system user. Functional testing of the EMSP and further software (V&V) will be conducted on the FDM.

Technical testing of the EDM's will consist of environmental qualification testing including maintainability and reliability demonstrations and Design Verfication testing. Operational testing will consist of execution of selected algorithms in shore based test facilities on actual sensor data. Based on successful technical and shore based operational testing of the EDM's authorization for limited production will be granted, and a production line opened.

Conclusion

The Navy is undertaking an agressive program to introduce a second generation standard signal processor EMSP into the fleet by 1987. This processor will meet emerging processing requirements for this decade and accommodate technology infusion to upgrade it for an additional decade's services life. EMSP is a multi-processing system, network of programmable array processors and special purpose devices. System performance is the effective aggregate of individual processor performance. EMSP software, ECOS, is based on support software under independent development for all Navy standard signal processors. It provides a user interface a level of abstraction above the HOL SPL/I. The system will have a data driven operating system and support tools to increase application programmer productivity. The software, particularly the operating system, will influence the system architecture and implementation. The Navy is in effect building an ECOS machine. There is technical risk involved in this development. The acquisition plan and test program are designed to manage risks, keeping them at acceptable levels. While this development is to meet future Navy signal processing needs, it is hoped that many multi-processing concepts may be validated in the course of development that are extensible beyond signal processing applications.

References

1. Purchase Description No. 101 for the Enhanced Modular Signal Processor (EMSP), Naval Sea Systems Command Purchase Request Number N00024-81-PR-28807, April 24, 1981
2. ACOS Methodology Program Performance Specification, Naval Sea Systems Contract N00024-80-C-7198, January 7, 1981

STANDARDIZED SOFTWARE DEVELOPMENT

SESSION CHAIRMAN: David J. Krile
ASD/ENAMR

MODERATOR: Colonel Harold C. Falk
AFWAL/AA
Special Assistant for Simulators

STATUS OF MIL-STD-1679A

AN OVERVIEW

MR. BILL EGAN
SOFTWARE ACQUISITION
AND POLICY MANAGER
(MAT 08Y)

MILITARY STANDARD 1679 (NAVY)

- ISSUED 1 DEC 1978
- REVISED DEC 1982
- CONTRACTUAL VEHICLE TO STANDARDIZE PROCUREMENTS

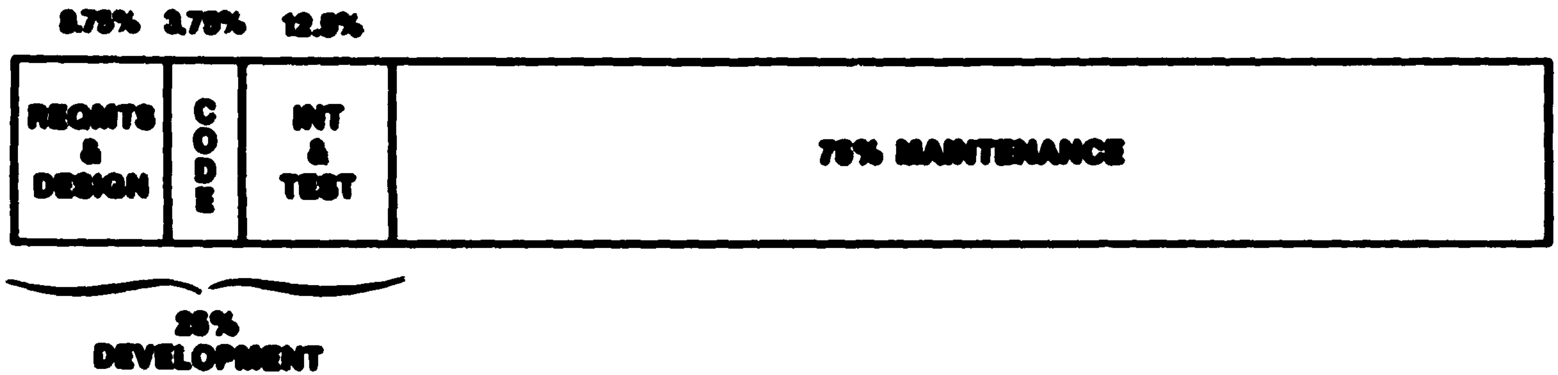
MIL-STD-1679

"ESTABLISHES UNIFORM REQUIREMENTS FOR THE DEVELOPMENT OF WEAPON SYSTEM SOFTWARE WITHIN THE DEPARTMENT OF DEFENSE."

MIL-STD-1679 INCORPORATES

- TOP DOWN METHODOLOGY
- STRUCTURED PROGRAMMING
- VERIFICATION & VALIDATION
- STANDARDS & CONVENTIONS
- RESERVES
- WALK-THROUGHS
- HOL
- QUALITY ASSURANCE

SOFTWARE COSTS



35-15-50 (GOVT)
40-20-40 (INDUSTRY)

GOALS OF 1679

- CONTRACTING VOID
- IMPROVE MAINTAINABILITY
- REMOVE GOTCHA's

NO SYSTEM DEVELOPED UNDER MIL-STD-1679 HAS YET TO REACH MAINTENANCE PHASE.

WHY REVISE

- DoD AND NAVY ECR POLICIES
- SUBMITTED COMMENTS
- INTERIM MEASURE

SIGNIFICANT CHANGES

- DELETE "WEAPON SYSTEM"
- UPDATE DEFINITIONS
- DELETE SOFTWARE ENHANCEMENT PROPOSAL
- TOP DOWN TESTING
- CLARIFICATION

1679A SCHEDULE

- 20 MAY 82 DISTRIBUTE FOR REVIEW
- 20 JUL 82 COMMENTS DUE
- 30 SEP 82 LATE COMMENTS DUE
- 30 DEC 82 RELEASE

PROBLEM — SIGNIFICANT COMMENTS RECEIVED (OVER 1200 SPECIFIC)

HARD HIT

- DEFINITIONS
- CONVENTIONS
- PRODUCTION
- TEST
- ACCEPTANCE

MIL-STD-SDS NAVY REVIEW

- LATE
- SIGNIFICANT COMMENTS
  - GENERAL
  - SPECIFIC
- SDS = NAVY POLICY

END

FILMED

10-84

DTIC