RD&E
C E N T E R

Technical Report

No. 13202

DEVELOPMENT OF THE VEHICLE INTEGRATED DEFENSE SYSTEM FEASIBILITY DEMONSTRATION MODEL

CONTRACT NO. DAAE07-83-C-R098

31 JULY 1986

By R. TOM, R. SIMMEN, R. McGOVERN
DALMO VICTOR SINGER
1515 INDUSTRIAL WAY
BELMONT, CALIFORNIA 94002

U.S. ARMY TANK-AUTOMOTIVE COMMAND
RESEARCH, DEVELOPMENT & ENGINEERING CENTER
Warren, Michigan 48397-5000

NOTICES

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No 0704-0188 Exp Date Jun 30, 1986

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS NONE
2a. SECURITY CLASSIFICATION AUTHORITY DD-254 dated 8 November 1984	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) R-3811-11930	5. MONITORING ORGANIZATION REPORT NUMBER(S) 13202
6a. NAME OF PERFORMING ORGANIZATION DALMO VICTOR-SINGER 6b. OFFICE SYMBOL (If applicable) --	7a. NAME OF MONITORING ORGANIZATION U.S. ARMY TACOM R & D CENTER
6c. ADDRESS (City, State, and ZIP Code) 1515 Industrial Way Belmont, California 94002	7b. ADDRESS (City, State, and ZIP Code) Warren, Michigan 48397-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable) AMSTA-ZSC	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DAAE07-83-C-R098
8c. ADDRESS (City, State, and ZIP Code) Warren, Michigan 48397-5000	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. | PROJECT NO. | TASK NO. | WORK UNIT ACCESSION NO

11. TITLE (Include Security Classification)
Development of the Vehicle Integrated Defense System Feasibility Demonstration Model

12. PERSONAL AUTHOR(S)
TOM, RAY; SIMMEN, ROBERT; McGOVERN, RUSSELL

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
FINAL	FROM 9-83 TO 5-86	1986 JULY 31	

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Multispectral Sensor Integration, Data Fusion, Embedded Processing, Runtime Software Applications, Armored Vehicle Survivability, Threat Resolution (Continued on Reverse)
17	02		

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

The Vehicle Integrated Defense System (VIDS) is a demonstration of the feasibility of using a centralized Data Management System (DMS) to enhance the survivability of the armored vehicle. The Feasibility Demonstration Model (FDM) achieves data fusion by Multispectral Sensor Integration (MSI) over a standard interface (MIL-STD-1553B) using Ada[R], THE DoD Standard Higher Order Language as the Program Design Language and the runtime code. Programming was accomplished on the UNISTAR 200 Workstation hosting an Ada[R] Compiler and embedded system kit software developed by TeleSoft Corporation and targeting the MC68000 microprocessor. The operating system for the software development was UNIX V. In order to support the Ada[R] Application packages in runtime on the target, we designed and coded a special Ada[R] Runtime Operating System (ARTOS) which can be burned into PROM for embedded processing of the software. The entire DMS processor is housed in an enclosure 7-1/2 inches wide by 10 inches high by 17 inches long with a self-contained, 28-volt DC converter, making it suitable for temporary installation in an M1 for field test purposes.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Dr. Francis B. Hoogterp	22b. TELEPHONE (Include Area Code) (313) 574-6693 22c. OFFICE SYMBOL AMSTA-ZSC

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted.
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

18. (Continued from reverse)

Automated Counteraction Recommendations, Expert Rule-Based Systems

19. ABSTRACT (continued from reverse)

The DMS functions as an "Expert Rule-Based System" by virtue of the static and dynamic database information programmed into the software and loaded into the processor. This information is provided by the "Knowledge Engineer" aided by a former commander of armor who apply their experience to the decision tree logic of the software database. In this manner, the "expert system" benefits from the prior experience of what has been done in a similar previous battlefield engagement situation, yet enhanced by the automated analysis and recommendation processing of the embedded DMS. Integration of the sensor data converts the experience into recommendations for the present battlefield circumstances as perceived by the sensors and the Expert Rule-Based System.

Soldier Machine Interface (SMI) is provided as the end result of the DMS. A flat panel (thin film transistor) with touch-sensitive display screen indicates the type of threat, its relative location and lethality to the crew after a synthetic voice alert has notified them through the intercom system. The commander may touch the screen to designate a particular threat and then touch a soft key on the screen that may be blinking to indicate the recommended counteraction to be taken against the designated threat. The soft key then changes color when the counteraction actually is initiated.

Although the contract only specified a best effort required for performance, we accomplished an end-to-end operation (threat simulation sensing to voice alert and symbolic display) of approximately two seconds elapsed time for processing, bus management and SMI.

The status of the DMS and diplay is complete at the brassboard level. An Advanced Development Program of at least 18 months is required to transition toward full-scale Engineering development.

SUMMARY

Work was carried out to demonstrate the feasibility of integrating hardware and software to provide a Vehicle Integrated Defense System (VIDS). A breadboard system was assembled and tested, incorporating a Data Management System (DMS) that detects, tracks, correlates threat data, and makes reaction decision recommendations and execution in near-real time while communicating with the crew through a flat panel control/display (Soldier-Machine Interface).

Several multi-spectral sensors were modeled in lieu of having the actual sensors available for the demonstration. Inputs to the sensors were simulated in terms of threat observables and programmed in the DoD standard Higher Order Language (HOL), Ada*.

A unique Ada Run Time Operating System (ARTOS) was developed to provide run-time support for the Ada application packages operating on the embedded MC68000 microprocessor. The DMS applies Expert Rule-based logic to identify the threat platform, track the threat location, assign relative threat lethality, determine counteraction options, deduce optimum counteraction applicability, recommend the applicable counteraction to the crew by visual and audible means, and initiate counteractions either automatically or on crew command.

The system was successfully demonstrated to exhibit an end-to-end integration of the hardware and software. Data processing was initiated and executed using realistic engagement parameters of threat observables. The DMS interface with the sensors and with the counteraction devices was provided by a MIL-STD-1553B data bus controller using a dual-port random access memory (RAM) "mailbox" data store and forward technique. Ada embedded software programming provided real-time performance in a discrete microcomputer.

The two most noteworthy achievements of the VIDS Feasibility Demonstration Model (FDM) were as follows:

- Multi-spectral sensor data fusion through the simple yet effective expediency of data normalization to a single (standard) processor interface for data input/output (I/O)

- Early success in the use of Ada as a program design language. This demonstrated the successful embedding of application software code using a unique operating system, ARTOS.

The FDM thus exhibited the overall VIDS DMS properties of automatic crew alert, semiautomatic counteractions, and embedded processing. All hardware developed under the contract is deliverable, as are the workstations on

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

which the software was developed, integrated, and tested. Complete documentation of all software design has been provided in both source code listings of the application packages and in the actual code on a 5¼-inch floppy disk. Complete specifications for both the software and hardware design are provided as well as engineering descriptions of the circuits that were developed for the hardware integration. All processes used in the integration of the FDM are summarized in this document. Brief summaries of the hardware and software development activities are also included. Detailed source code listings, software specifications, and hardware technical descriptions are contained in attachments referenced in this final report.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

Additional material prepared during the project and delivered to TACOM for reference purposes is listed below.

OPERATING SYSTEM SOFTWARE DESCRIPTIONS
APPLICATION SOFTWARE DESCRIPTIONS
ADAPTER UNIT SUBSYSTEM DESIGN DETAILS
DISPLAY/CONTROL PANEL DESIGN DETAIL
SYMBOLOGY FIRMWARE
VOICE SYNTHESIS FIRMWARE
TESS OPERATOR'S MANUAL

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS

LIST OF TABLES

THIS PAGE LEFT BLANK INTENTIONALLY

1.0. INTRODUCTION

This report describes the results of an FDM design and development program for the TACOM VIDS by Dalmo Victor under Contract No. DAAE07-83-C-R098. The heart of the VIDS is its DMS, which processes information from the threat detection sensors and then initiates the best reaction to counter the threat. Figure 1-1 shows the basic concept of the VIDS DMS. The key element of the DMS software is the Threat Resolution Module. It accepts the raw sensor output and detects, classifies, and prioritizes the threats. The work presented herein contributes to the enhancement of modern armored vehicle survivability, a highly critical area in an environment of numerically superior opposition.

The FDM development project was initiated to address the need for Government and industry experience in:

- Multispectral sensor integration

- Real-time processing of application software written in the Ada Higher Order Language.

The project goal was to prove the feasibility of not only developing algorithms for the necessary tracking and correlation of threat sensors but also the feasibility of coding the algorithms using the new DoD standard HOL, Ada, as the programming design language.

The algorithms have now been developed, written in Ada, coded, and tested. The results are described in this technical report. Program development was carried out on a Callan Data Systems Workstation in which the host processor was an MC68000 CPU. Testing was demonstrated in which the object code was targeted on the MC68000 CPU.

Although software development was the central thrust of this FDM effort, the necessary development of hardware and firmware associated with the software processing is described in this report. Specifics of the circuit designs are provided in attachments to the report. Commercial hardware was utilized wherever possible to reduce the technical risk and cost. New hardware was also developed for the FDM. It consisted of:

- A MIL-STD-1553B Terminal Controller board with host computer and dual-port RAM on board

- A thin-film electroluminescent flat panel display with touch-sensitive screen for use in demonstrating Soldier-Machine Interface.

All of this hardware is described in this report, with further details provided in the attached documents.

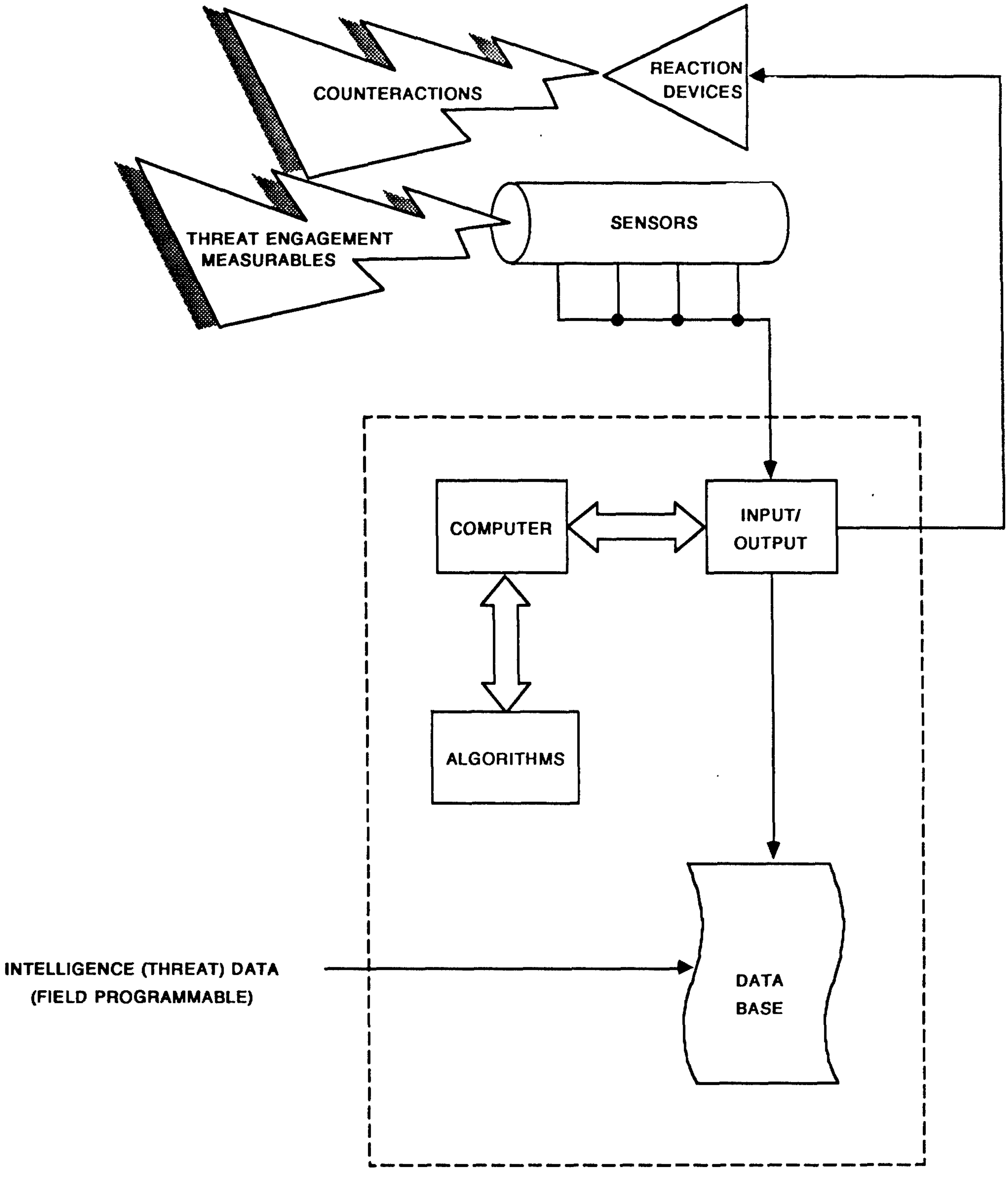


Figure 1-1. VIDS DMS Concept

2.0. OBJECTIVE

The objective of the VIDS is to provide increased vehicle and crew survivability. A DMS automatically correlates the input from multi-spectral sensors. It identifies threat platforms, determines appropriate counteractions, advises the crew, and executes the counteractions either automatically or manually. The system is designed to interface with the crew with minimum disruption of their combat operations. Crew reaction is kept as simple as possible. To demonstrate these DMS capabilities, the following requirements were specified and met:

- Demonstrate ability to integrate hardware and software for these VIDS processing characteristics:
  - Detection of multiple threats using at least four different sensors
  - Detection and track of threat "signals" from moving platforms
  - Correlation of data from multiple "emitters" on one platform
  - Lethality prediction based on tactical situation evaluation
  - Appropriate counteraction recommendation
- Simulate threat observables in lieu of actual sensors
- Interface DMS with sensors and peripherals using a standard data bus
- Develop software for real-time performance in an embedded microcomputer
- Program and code the software in Ada
- Exhibit properties of:
  - Automatic crew alert
  - Semiautomatic counteractions
  - Embedded processing.

This report provides the following information on the project:

- Thorough description of the technical effort and the system developed
- Discussion of "lessons learned" during the development

- Overview of software and firmware developed on the project
- Description of the integration process and demonstration of the FDM.

Attached to the report as separate documents are:

- Revised "Enhanced Software Specification"
- Annotated descriptions of the source code for the applications packages (software)
- Results of the FDM software integration on a 5¼-inch floppy disk.

An illustration of the overall FDM and software development system is shown in Figure 2-1.

The schedule for performance of the FDM was basically a 30-month program plus preparation of this final technical report and a short video tape illustrating the operation of the fusion processor when exercised by the Tactical Engagement Situation Simulator (TESS). The original schedule is shown in Figure 2-2, with dotted lines and solid triangles showing the actual completion dates. The main reasons for the 3-month slip were:

- A planned delay of hardware development to bring it more nearly into proper time relationship with software integration
- Continued difficulties with the Ada Run Time Operating System (ARTOS) which had to deal with an incomplete compiler and early tools for embedded systems.

Further discussion of these difficulties is included in Section 3.0, Conclusions.

3.0. CONCLUSIONS

The following statements summarize the conclusions of the FDM development activity.

- The VIDS, comprising automated onboard sensors and countermeasures, is a feasible concept.

The overriding conclusion must be that a microprocessor-based data management system can be successfully developed to be small enough to reside within the confines of a combat vehicle, yet powerful enough to perform the real-time processing of multispectral sensor data.

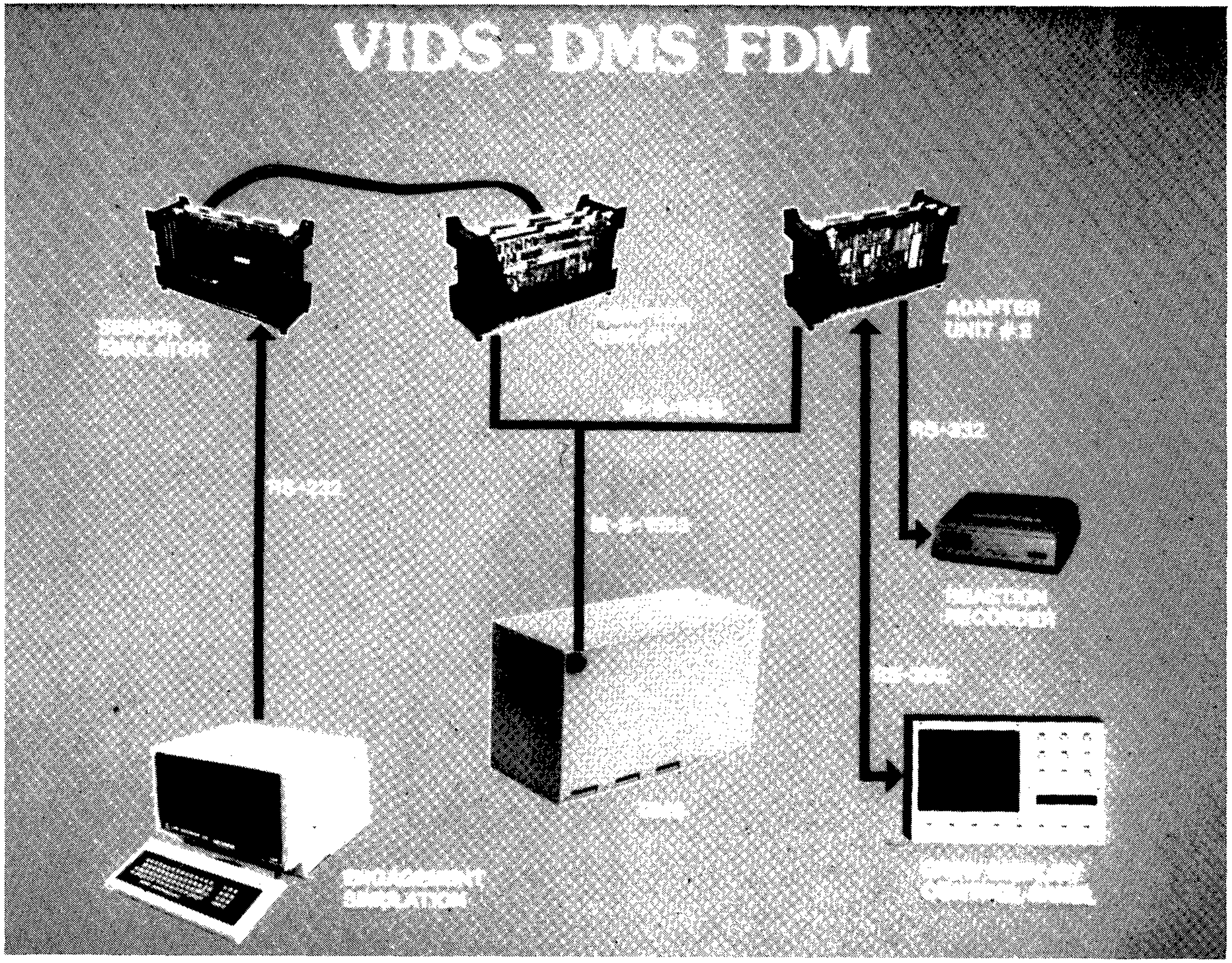


Figure 2-1. Feasibility Demonstration Model

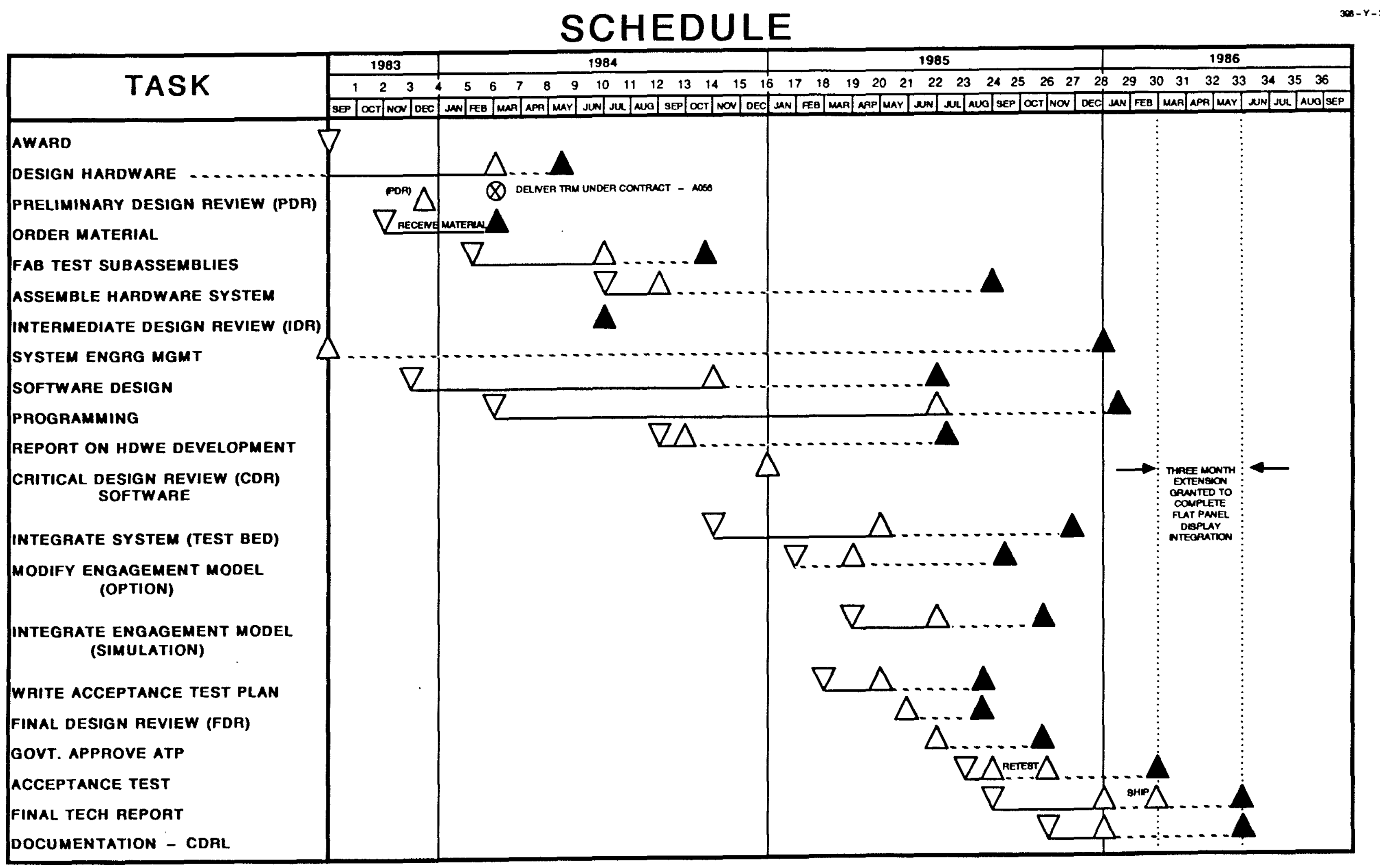


Figure 2-2. Program Performance Milestones

- The system can handle and operate at least 20 sensors and countermeasures.

We successfully demonstrated an implementation of a MIL-STD-1553B Data Bus Controller terminal.

The specifications for the various sensors, which are candidates for eventual VIDS implementation, are presently incomplete. This is due to the fact that most sensors are presently under development. Although this situation caused some uncertainty during the development of the FDM, it will be a long-term advantage if the sensors eventually become equipped as MIL-STD-1553B remote terminal interfaces.

- Ada HOL has been confirmed as a feasible software language for the DMS.

Perhaps the single most significant accomplishment during the development of the FDM was that we successfully demonstrated that a large amount of Ada code (requiring an estimated 500,000 bytes) will run in near real-time when embedded in a high-speed 32-bit microcomputer. Although the average time for presentation of a voice alert and a display symbol following receipt of sensor input packets (SIPs) from the simulator was an average of two seconds, there are a number of refinements which may be made in the improved software of full-scale engineering development (FSED) processor. We are confident that the eventual DMS and its Ada software will provide threat warning on a time scale equivalent to or better than that of most current warning systems used by tactical aircraft.

- Expert Rule-based logic has been successfully demonstrated in processing threat data and recommending appropriate counteractions.

Another significant accomplishment was the establishment of an "Expert System" which not only warned the crew of an impending threat but also recommended the specific counteraction to be taken against that threat.

This Expert System incorporated that static data base of the various types of threats and compared them with the given reactions that would take place if an experienced tank commander were evaluating the proper response to a particular threat during an engagement. These Expert "rules" were incorporated in the dynamic data base and used in the reaction module software processing to determine the appropriate counteraction recommendation to be made.

- The system can receive and process threat data from sensors and activate countermeasures with minimum disruption to other crew operations. Soldier-machine interface is simple and unambiguous in concept and can operate in minimum time.

A further significant development beyond the scope of the contract was the demonstration of an interactive display and control panel which provided an important soldier-machine interface (SMI) for a very realistic demonstration. The inclusion of a touch-sensitive membrane over the flat panel symbolic display, combined with the synthetic voice alert, allowed the tank commander to designate the particular threat that he wished to take counteraction against and then either follow the computer's advice or made his own selection for counteraction, depending on his final evaluation of the engagement. We believe this to be the most realistic scenario for the "man in the loop" in the combat vehicle. That is, this systems responds to the dictim "let the computer do what it does best and man do what he does best."

The FDM conclusions on SMI capabilities are:

- The display/voice alert has been successfully demonstrated as an operator interface.

- The touch-sensitive, flat panel screen is a feasible operator control device.

- A data link device would increase system effectiveness through communications with outside stations.

Other conclusions that may be drawn from the successful completion of the VIDS FDM DMS are as follows:

The 1553B data bus controller with dual-port RAM mailbox and dedicated onboard Central Processing Unit (CPU) has been sucessfully demonstrated as a feasible method for data message handling.

The Threat Resolution Module designed during VIDS Phase II program has been a valuable and operational software program.

The unique Ada Run Time Operating System, ARTOS, developed on internal funding by Dalmo Victor Systems and Software Engineering (formerly Bell Technical Operations) in Tucson, Arizona, supports Ada packages in run time when embedded with the MC68000 microcomputer.

The hardware and software which was developed for the feasibility demonstration model is suitable for full-scale engineering development to military qualifications. All of the hardware components used in the FDM are available as Military Specification components and nearly all of the software requirements can be met using Military Standard 1815 (Ada) programming.

- The FDM developed for this program can be adapted readily for field test by adding a memory board to replace the present floppy disk drive.

- The DMS developed for this program to interface between threat warning sensors and system operations with crew display/control capability has numerous applications to other ground battlefield requirements.

4.0 RECOMMENDATIONS

The two most important recommendations are:

- Continue the program to permit efficient integration of the VIDS FDM into the VETRONICS demonstrator.

- Commence a brassboard development which would provide a smaller, more ruggedized processor with an update to the software using a fully validated compiler and embedded system kit.

Subsequent recommendations include the previously suggested development by the U.S. Army of a series of time-line analyses for each particular threat to our vehicle in a one-on-one engagement situation. This would allow more accurate selection of counteractions based on realistic lethalities and probable fly-out times of the threat weapons systems.

Many recommendations could be made with respect to the continuing development of sensors for the VIDS. This is beyond the scope of this current requirement, but it is of vital importance to the overall success of the VIDS.

The Ada Run Time Operating System (ARTOS) was developed on internal funding by Dalmo Victor Systems and Software Engineering (SSE). Because of limited time and resources, we were able to develop an ARTOS which provides only the minimum essential functions. This was caused by the extreme difficulties encountered in the use of a less than efficient Embedded System Kit which was developed for the early Ada Compiler by TeleSoft. As a result, we elected to code and test only those functions of the ARTOS that were essential to the demonstration of the FDM. The original design, if completed in code and tested, would provide a richer set of utilities. Therefore, in order to take the next step toward a field demonstration, the ARTOS should be completed.

In the short term, if a field demonstration is required, the ARTOS must be upgraded to include such functions as 68000 central processor allocation to individual software processes based on priority and real-time clock management.

In the long term, a greater portion of the ARTOS code must be rewritten in Ada to allow for greater transportability to different target processors (such as the 80186 planned for the APR-39A unit).

Other Army battlefield requirements should be examined for feasibility of adapting the DMS to their needs. For example:

FAADS - Target acquisition and fire control

ETAS - Sensor integration (ground and air)

VISTA - Data fusion - Rule-based logic - Transmission

BMS - Data integration - information display

OTHER - System requiring sensor/data fusion from multi-spectral sensors and expert rule-based logic to manage reaction devices.

The final recommendation, overall, is to promote and strengthen the U.S. Army's use of Ada. It is an excellent language when used as program design language in a well organized, object-oriented design methodology. Further, we have shown that compilers can be and have been developed which are efficient to the extent that they can be used in the near real-time environment required by threat warning systems. Thus, we believe that all further specifications for development programs of this sort should include the absolute requirement that application packages be programmed in Ada with no waivers permitted.

Again, the VIDS DMS program should be continued into an advanced development stage to field test system validity using available operational sensors and countermeasure systems.

5.0. DISCUSSION

5.1. Overview of Report

Our 1980 study addressed the principal VIDS purpose of improved vehicle and crew survivability. The technical approach of the study encompassed several issues:

- The ability of sensors to detect the threat before launch
- The need for autonomous sensor and reaction operation
- The sensors contained microprocessors which required an architecture of distributed, rather than centralized, processing
- The system design for automatic operation with manual override capability
- The need for crew alert and display options which effectively aid the crew without distracting them from their primary mission.

It was recognized that the VIDS FDM program would be highly software intensive which, because of its complexity, would have to be organized into a well-structured family tree. The overall organization is shown in Figure 5-1. This system is further described in Section 5.2.2. The hardware necessary to support the FDM consisted of two types of equipment: the development and support equipment, which was purchased, and the breadboard DMS and the Crew Display and Control Panel, which were developed. This system is described in Section 5.2.1. The VIDS configuration is shown in Figure 5-2.

Detailed descriptions of the hardware subsystems and software packages are described further in Sections 5.3, 5.4, and 5.5.

Integration of the entire Feasibility Demonstration Model, first as groups of hardware, then as groups of software modules, and finally as a system, is described in Section 5.6.

TESS, which was necessary to energize the input portion of the FDM, is described in Section 5.7.

5.2. General System Description

The VIDS FDM was developed to demonstrate the feasibility of enhancing the survivability of armor through the real-time processing of battlefield information. This information is available to a VID system from several sensors on board (or off) the vehicle. The information can be detected and classified as various emissions or other observables of the threat. The sensors generally contain one or more embedded microprocessor chips or boards to perform computation of the fundamental characteristics of the emissions. This is signal processing in the sense of analog-to-digital conversion and analysis. It is preprocessed, normalized data that the sensors input to the VIDS DMS. The first job of the DMS, prior to management, is to sort out the various inputs. This is done by providing a controller for a dual MIL-STD-1553B data bus that polls the sensors (and other peripheral devices) to determine how and where the preprocessed signal information is to be handled.

The management tasks of the DMS involve analysis of location, relative lethality of the threat, priority (and propriety) of warning the crew, display of information, and initiation of appropriate counteraction(s). Thus, the DMS is a true manager of data for the entire VIDS system. Because it interfaces with so many different types of peripheral devices and subsystems, it can function (to a limited extent) as a data manager for the entire vehicle.

This approach illustrates our concept of data fusion. If this technique is carried further, using Artificial Intelligence methods, the system can provide total data management. For example, several sensors may have relevant data but require "human experts" to decipher and correlate the data to make

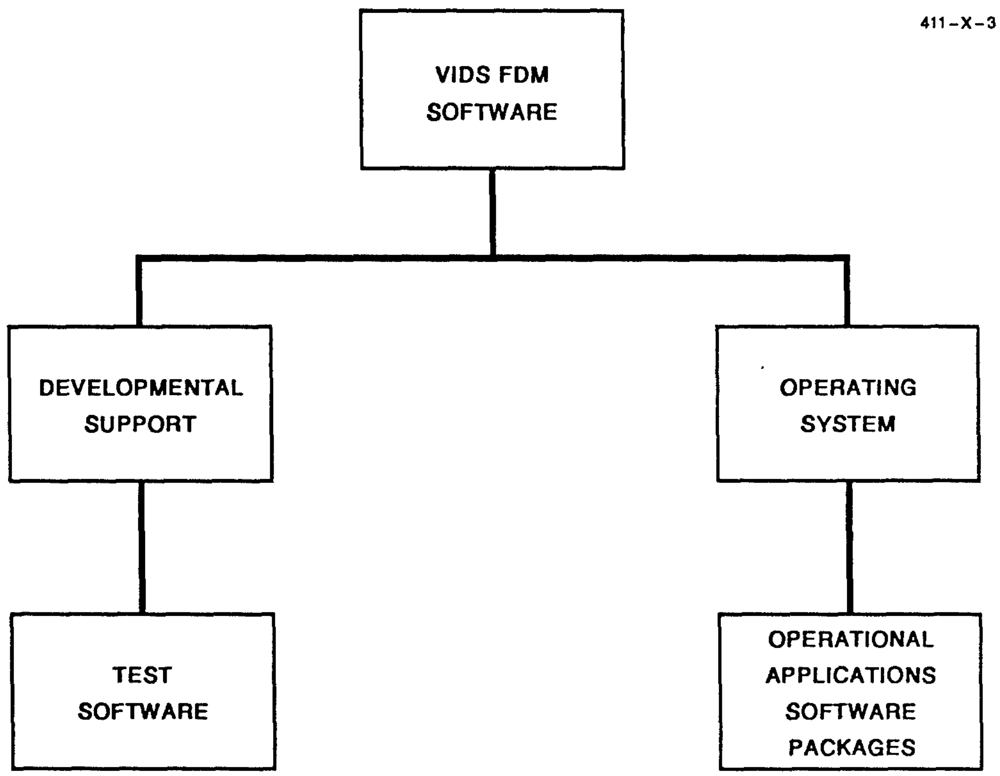


Figure 5-1. VIDS FDM Software Structure

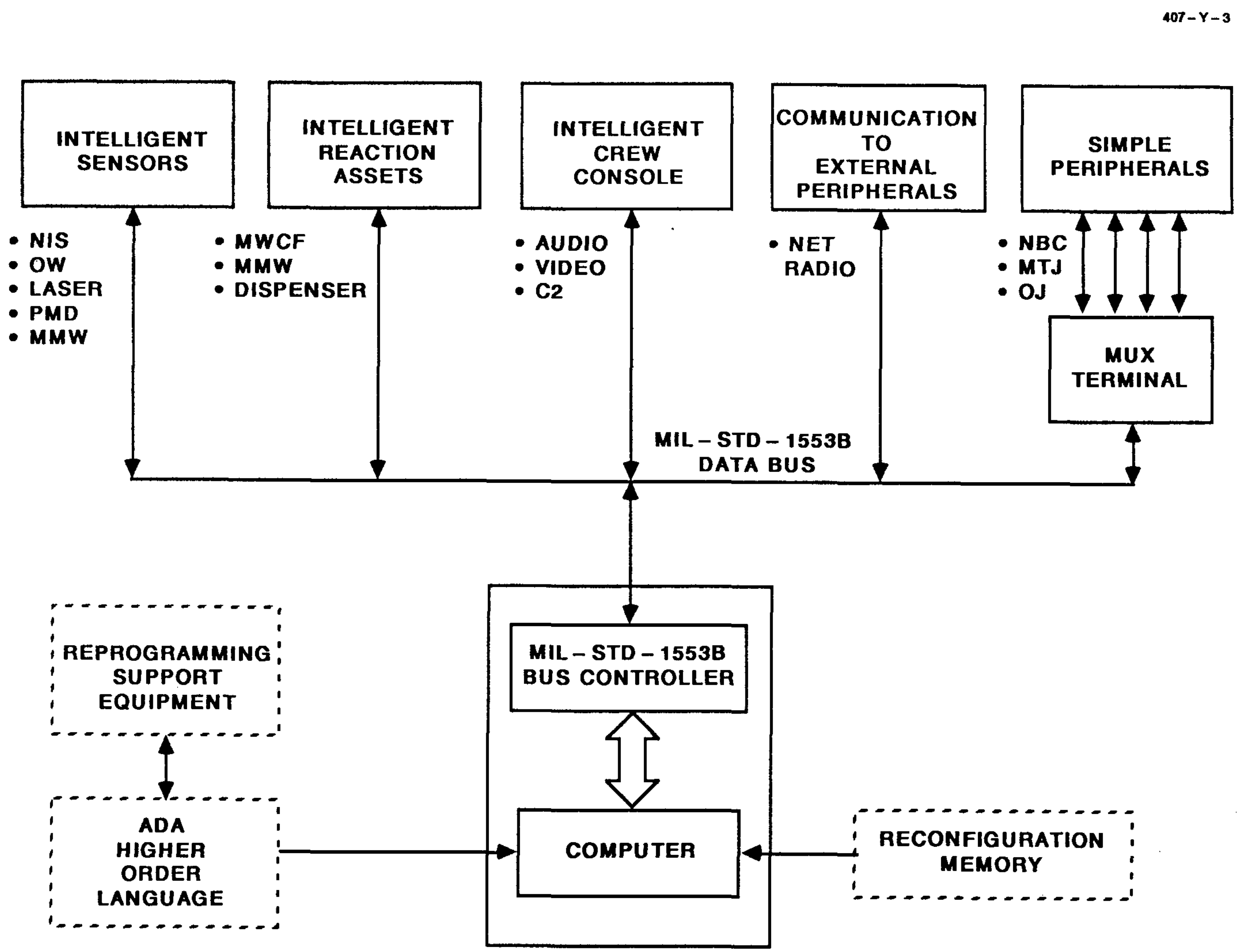


Figure 5-2. VIDS Configuration

a more accurate judgement. The DMS contains several data files to relate the nature of the threat observables and individual sensor characteristics (static data base) plus the real-time recording of sensed data (dynamic data). We can thereby program the DMS to consider various combinations of threats, locations, tactical situations, environmental conditions, Electronic Order of Battle (EOB), and availability of counteractions as if we were witnessing a previous situation on an actual battlefield.

In this manner, we are able to give the DMS the characteristics of a true "Expert System" by giving it the capability to output the same recommendations an experienced battlefield commander would have made for a given set of circumstances. Since the individual tank commander is generally too busy directing his own tank to be able to run the necessary data analysis of sensor information, we claim that the DMS is able to enhance survivability by supplying the commander with processed information (intelligence) that he would otherwise be completely unable to effectively acquire or utilize.

5.2.1. Hardware System Overview. To understand the VIDS sensors and counteraction devices, we offer the following abbreviated definitions of the various peripherals:

5.2.1.1. Peripheral devices. The several devices and subsystems presently served by the DMS in the 1986 Feasibility Demonstration Model are listed below with a brief description of their function.

Sensors

Optical Warning/ Optical Jamming	Detects and counters optical focal planes (telescopes, binoculars, range finders)
Laser Detector	Detects coherent light from laser designator range finder or beam rider missile
Non-Imaging Sensor	Detects, locates, and classifies helicopters
Passive Missile Detector	Senses propulsion (plume) of rocket motors
Nuclear, Biological, Chemical	Detects nuclear, biological, and chemical agents
Millimeter Wave	Proposed detector (passive and active) for future application to VIDS

Responses and Counteractions

Crew Alert and Warning	Synthetic voice annunciation
Threat Display and Control Panel	Flat panel display of graphics and alphanumerics with operator controls (touch sensitive, voice synthesis)
Laser Decoy	Counter to laser designator weapons by offset illumination or reflection
Flare, Smoke, Chaff	Decoy and obscurant (expendables dispenser)
Main Weapon Counterfire	Semiautomatic turret slew and tube elevation
Maneuver	Predetermined series of maneuvers based on timing, location, bearing, and tactical conditions
Missile Tracker Jammer	Infrared seeker countermeasure

5.2.1.2. FDM configuration. To demonstrate the integration of the stated peripherals (or their simulations), a system was configured as illustrated in Figure 5-3. These eight hardware assemblies are summarized in the following descriptions.

TESS is a software program designed in Ada as the Program Design Language (PDL). The TESS will provide a simulation model of the threat environment; it will be used to exercise the entire FDM and to run an operational scenario to provide representative SIPs. These SIPs are communicated in sequence, according to a written engagement scenario, over an RS-232 bus to the sensor emulator.

A Hardware Emulator (Adapter Unit No. 3) is used to convert the software from the TESS into hardwired outputs at TTL levels, serial and/or parallel, just as they exist in the actual (or planned) sensors. These sensor outputs are interfaced electrically to Adapter Unit No. 1 (below) where they are buffered for retransmission to the DMS.

A Hardware Format Converter (Adapter Unit No. 1), which accepts the various forms of data input from the sensors (emulated), commutates the samples and converts the samples of the sensor data into serial format in accordance with MIL-STD-1553B bus requirements.

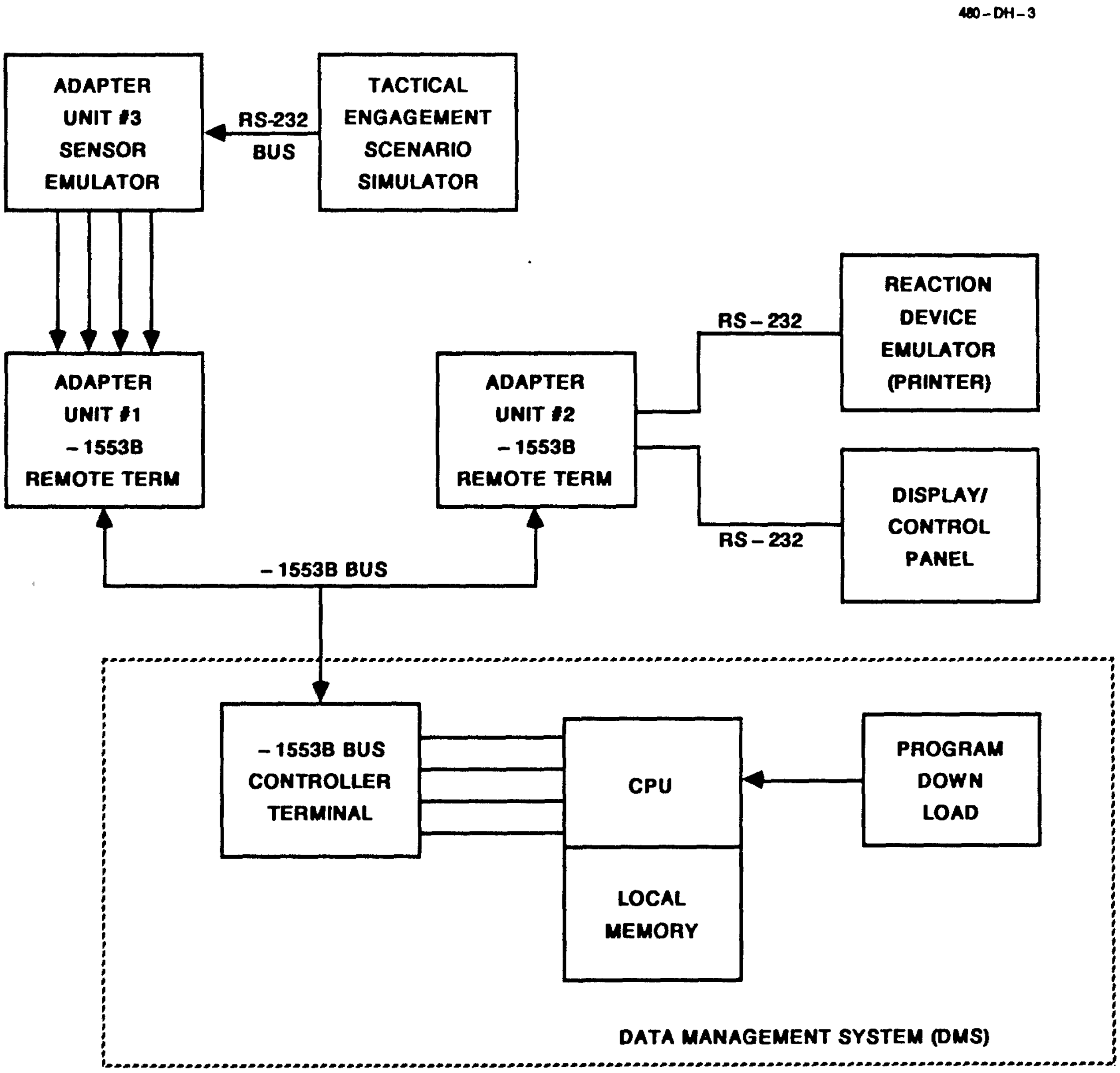


Figure 5-3. FDM Hardware Configuration

A DMS Hardware Assembly which provides the following subassemblies:

- 1553B Bus Controller and mailbox board
- CPU for the entire DMS
- Memory board with one megabyte of no-wait state RAM
- Floppy-disk controller board for (temporary) download of program store for demonstration only.*

A DMS Software Assembly consisting of the following two major components:

- An operating system which supports Ada code in run time on an embedded MC68000 CPU.
- Application software packages (written in Ada) which provide the processes and data bases for the active VIDS operation.

A Hardware Adapter (Adapter Unit No. 2) which accepts command messages from the DMS over the 1553B bus and translates them into one of two RS-232 links to either the display/control panel or the counteraction devices. This adapter contains a 1553B remote terminal for reception and transmission of signals from and to the DMS, plus look-up tables to convert 1553B bus messages into RS-232 messages.

A Display/Control Panel to provide the crew with the interface functions described in detail in Section 5.3.5.

A Counteraction Device Simulator which represents the actions to take place as a result of DMS operation. This is done by simply printing out the commands that in field usage would result in actual counteraction initiation.

These hardware items are described in greater detail in Section 5.3, with complete engineering data contained in separate documents.

5.2.2. Software System Overview. There are several individual software components in the overall FDM: the TESS, the firmware used in the adapter units, and the firmware in the crew Display/Control Panel. For the purposes of this section, the description of firmware will be included in the detailed description of the adapter units and the display panel. The major software development for the FDM is in the operational application modules, described in detail in Section 5.5, and the Ada Run Time Operating System development (ARTOS), described in detail in Section 5.4. The graphic description of the overall software family tree is shown in Figure 5-4 (on two pages).

*Original contract requirement. The program is currently downloaded to RAM board (to be converted to EEROM).

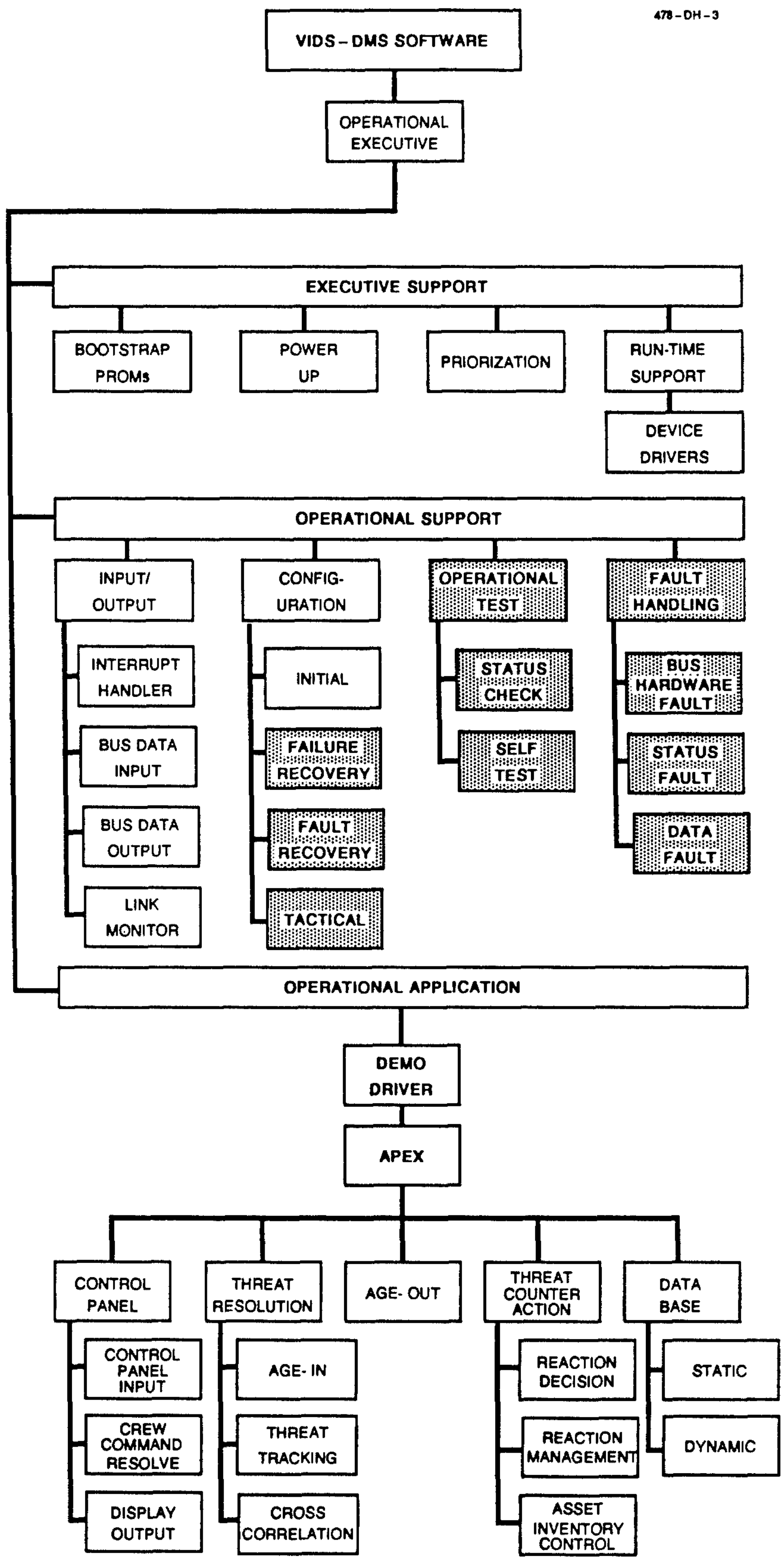


Figure 5-4. FDM Software Configuration

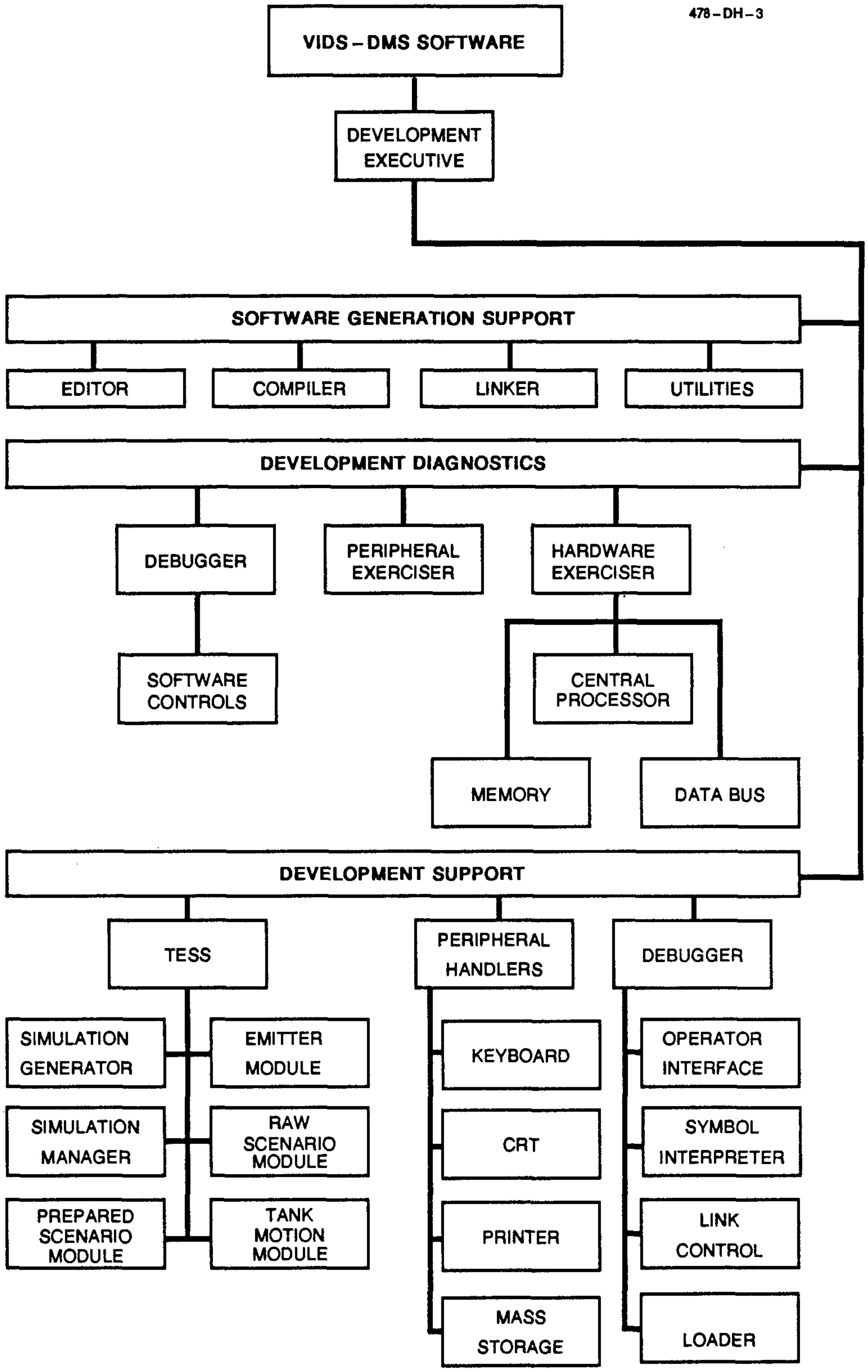


Figure 5-4. FDM Software Configuration (Continued)

This organization distinguishes between the developmental software subsystems and their support tools on the left side of the illustration under the "Development Executive," and the combination of both the operating system and the operational applications packages on the right side under "Operational Executive."

A special run-time executive and real-time operating system has been developed to provide efficient operation of the DMS when implemented as an embedded processor. The developmental software system operates under the UNIX operating system on the Callan Data Systems workstations, while the Ada application packages, which were developed under the UNIX system, are supported on the ARTOS. The ARTOS was designed and coded with a specific embedded system kit to ensure proper support of the Ada packages in run-time on the embedded microcomputer (MC68000).

The DMS functions in the development mode with UNIX as the operating system software. Changing from one operating system to another involves the reinstallation and replacement of two Programmable Read Only Memory (PROM) chips.

Application packages are written in Ada and compiled on the workstation (host), which utilizes the same type of CPU as the DMS (target).

The magnitude of the actual embedded code is approximately 150K bytes of executable code. This includes the skeleton operating system and the complete operational application packages. The combined operating system and applications packages are downloaded from the host workstation disk file into solid-state memory (RAM) by way of a temporary RS-232 data link which can be physically disconnected from the DMS after the program is downloaded.

5.3. Detailed Descriptions of Hardware Subsystems

5.3.1. Sensors. This section describes the interface to each sensor as Dalmo Victor understood it prior to the FDM design freeze (1 September 1984). Additional data may be found in the design documents for the adapter units which are provided with this document.

5.3.1.1. Non-Imaging Sensor (NIS). Although developed with a paralled data interface, the NIS has a MIL-STD-1553B interface with the VIDS DMS. At the time of the design freeze, only preliminary planning for this interface had been started by the NIS contractor. This section reviews the preliminary planning and offers modifications to the preliminary concept.

Current Interface Concept. The NIS samples the environment at intervals of 250 msec or less. During this period, the entire environment is analyzed and data records are generated for the ten highest priority threat sources. The analysis results in ten threat records derived directly from data collected during the sample period. The record for each threat consists of information which can be used to identify and locate the threat. Location data include azimuth, elevation, and range, all relative to the sensor platform. Other data include threat classification (identification) and indications of received frequency and power.

The threat records noted above are assembled in a prioritized table for block transfer to the DMS via a MIL-STD-1553B remote terminal. Table 5-1 illustrates the threat table as it is currently defined. Some problems related to this interface are noted below.

- Word Count: Table 5-1 indicates 64 words. This block is in excess of the 32-word maximum implied by the 5-bit word count field in the MIL-STD-1553B command format. As a minimum, the threat record list must be modified into two 32-word blocks for transfer to the DMS.

- Handshaking: The present plan is to transfer data in two phases. The first phase requires that the DMS interrogate the NIS for valid threat record count (number of groups). Based on the number of valid threat records, the DMS controller would then issue a command requesting either a complete transfer or a transfer only of the number words including active threat records. Two subsequent transfers are required if more than five action threat records exist in the NIS threat table.

- Data Format and Content: The data in Table 5-1 are not favorably organized for the DMS. It is not clear why frequency, power, and count are included in the data field. These would appear to be redundant for VIDS but may be required by some interface.

- Tracking: The NIS does not track records from one sample period to the next. Thus, a threat record may appear in different positions in the table for different analysis cycles. All sample-to-sample tracking must be accomplished in the DMS.

Figure 5-5, when combined with Table 5-2, suggests an alternate and more favorable data organization. This organization will be assumed for the adapter software and sensor simulation. The following are comments relate to Figure 5-5 and Table 5-2:

- Sensor data have been compressed to a total of five 16-bit words for each threat record. The total word count is then 50 to 10 threat records. These 10 threat records should be transferred to the DMS in two blocks, or five threat records each. The threat records and block transfers are handled at the interface in order of decreasing priority so that the highest priority threat is transferred first in each data block.

- All data contained in Table 5-1 are accounted for in Table 5-2 and Figure 5-4. All data will not be used in the DMS, as noted in Table 5-2. Also note that each block of data consists of only 25 words, allowing for seven spare words if the block is expanded to the full 32 words allowed by MIL-STD-1553B.

Table 5-1. NIS Output Data (Planned)

Word	Name	Value	Display	Definition
1	GROUPS	0-A	0-10	Number of Groups
2	BEAR1	0-7FF	0 to 360	Bearing
3	ELEV1	FE00-01FF	-90 to +90	Elevation
4	FREQ1	1-FF	1 to 255	Main Frequency
5	PWR1	1-FF	1 to 255	Power of Main Freq.
6	COUNT1	1-A	1 to 10	Count of Signals
7 L	CLASS1	1-A	TBD	Classification
7 R	RANGE1	0-FF	0 to 25,500 ft.	Estimated Range
8	BEAR2	0-7FF	0 to 360	Bearing
9	ELEV2	FE00-01FF	-90 to +90	Elevation
10	FREQ2	1-FF	1 to 255	Main Frequency
11	PWR2	1-FF	1 to 255	Power of Main Freq.
12	COUNT2	1-A	1 to 10	Count of Signals
13 L	CLASS2	1-A	TBD	Classifications
13 R	RANGE2	0-FF	0 to 25,500 ft.	Estimated Range
.		.	.	.
.		.	.	.
.		.	.	.
.		.	.	.
54	BEAR10	0-7FF	0 to 360	Bearing
57	ELEV10	FE00-01FF	-90 to +90	Elevation
58	FREQ10	1-FF	1 to 255	Main Frequency
59	PWR10	1-FF	1 to 255	Power of Main Freq.
60	COUNT10	1-A	1 to 10	Count of Signals
61 L	CLASS10	1-A	TBD	Classification
61 R	RANGE10	0-FF	0 to 25,500 ft.	Estimated Range
62-64				Spares Estimated

Where

L=left-half word
R=right-half word

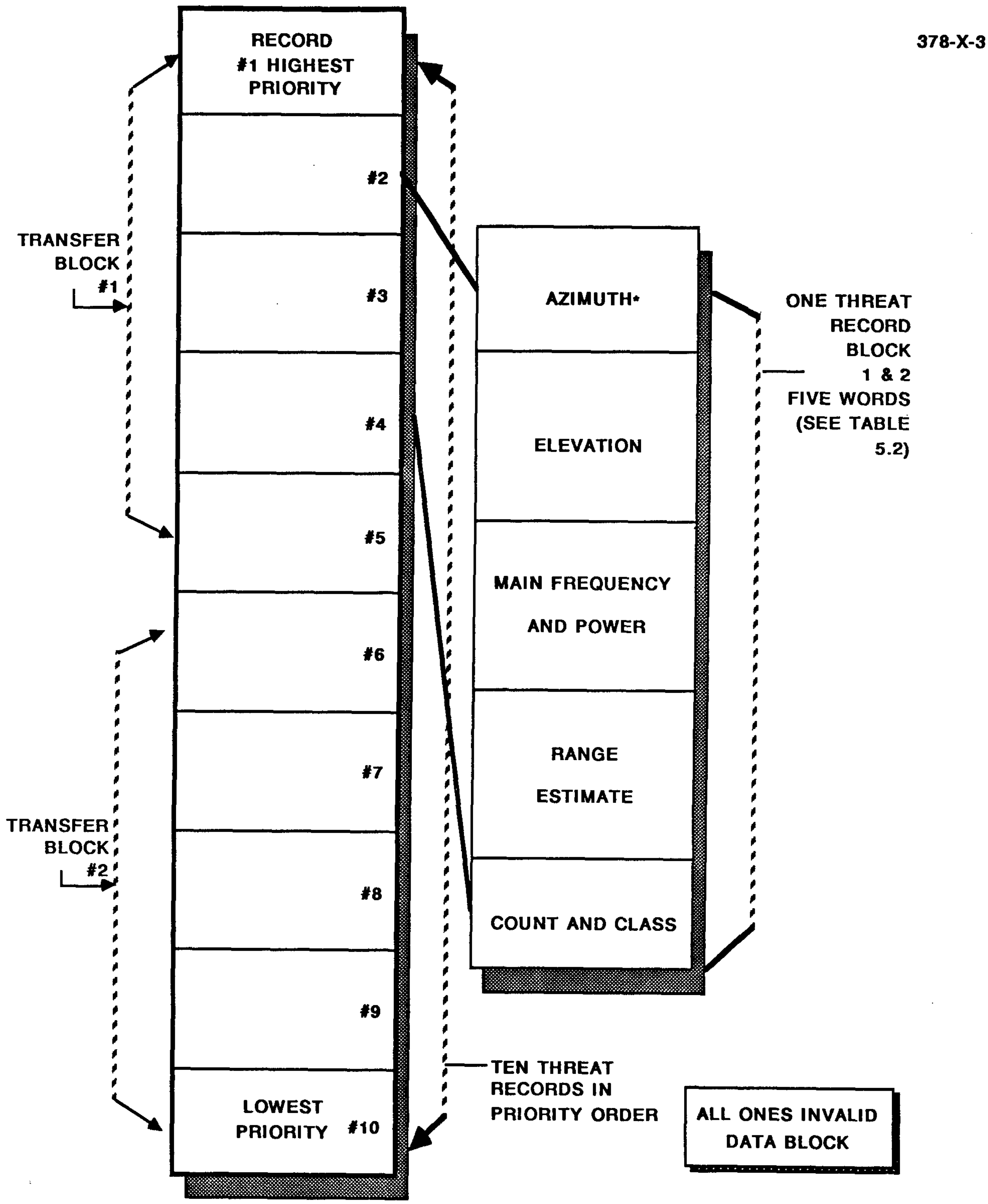


Figure 5-5. Preferred Data Block Structure

Table 5-2. NIS Output Data (Modified)

Word	Name	Value	Display		Definition
1 (a)	BEAR1	0-7FF	0 to 360		Azimuth Bearing
2	ELEV1	FE00-01FF	-90 to +90		Elevation Bearing
3 (b)	FREQ1	1-FF	1 to 255	(e)	Main Frequency
3 (c)	PWR1	1-FF	1 to 255	(e)	Power of Main Freq.
4 (b)	COUNT1	1-A	1 to 10	(e)	Count of Signals
4 (c)	CLASS1	1-A	TBD		Classification
5	RANGE1	0-FF	0 to 25,500 ft.		Estimated Range
6 (a)	BEAR2	0-7FF	0 to 360		Azimuth Bearing
7	ELEV2	FE00-01FF	-90 to +90		Elevation Bearing
8 (b)	FREQ2	1-FF	1 to 255	(e)	Main Frequency
8 (c)	PWR2	1-FF	1 to 255	(e)	Power of Main Freq.
9 (b)	COUNT2	1-A	1 to 10	(e)	Count of Signals
9 (c)	CLASS2	1-A	TBD		Classifications
10	RANGE2	0-FF	0 to 25,500 ft.		Estimated Range
11 (a)	BEAR3	0-7FF	0 to 360		Azimuth Bearing
12	ELEV3	FE00-01FF	-90 to +90		Elevation Bearing
13 (b)	FREQ3	1-FF	1 to 255	(e)	Main Frequency
13 (c)	PWR3	1-FF	1 to 255	(e)	Power of Main Freq.
14 (b)	COUNT3	1-A	1 to 10	(e)	Count of Signals
14 (c)	CLASS3	1-A	TBD		Classification
15	RANGE3	0-FF	0 to 25,500 ft.		Estimated Range
16 (a)	BEAR4	0-7FF	0 to 360		Azimuth Bearing
17	ELEV4	FE00-01FF	-90 to +90		Elevation Bearing
18 (b)	FREQ4	1-FF	1 to 255	(e)	Main Frequency
18 (c)	PWR4	1-FF	1 to 255	(e)	Power of Main Freq.
19 (b)	COUNT4	1-A	1 to 10	(e)	Count of Signals
19 (c)	CLASS4	1-A	TBD		Classification
20	RANGE4	0-FF	0 to 25,500 ft.		Estimated Range
21 (a)	BEAR5	0-7FF	0 to 360		Azimuth Bearing
22	ELEV5	FE00-01FF	-90 to +90		Elevation Bearing
23 (b)	FREQ5	1-FF	1 to 255	(e)	Main Frequency
23 (c)	PWR5	1-FF	1 to 255	(e)	Power of Main Freq.
24 (b)	COUNT5	1-A	1 to 10	(e)	Count of Signals
24 (c)	CLASS5	1-A	TBD		Classifications
25	RANGE5	0-FF	0 to 25,500 ft.		Estimated Range

(a) Threat Record Flag: All 1s = Invalid Threat Data Record
(b) High Byte
(c) Low Byte
(d) One of two blocks: Block #1 (higher priority block) shown
(e) Not used by the DMS

- A flag is used to indicate valid/invalid data in each individual threat record. Since the azimuth data require only 11 bits, the higher-order bits can be used as flags. As defined in Table 5-2, and Figure 5-4, a reading of all 1's in the azimuth field indicate that there are no valid data in the threat record. Since the data sets are arranged in priority order, the "all 1's" flag in the azimuth field implies a limit to the number of valid threat records. Invalid data in any threat record implies invalid data in all lower-priority records. An invalid threat record in Block #1 terminates the need to transfer Block #2.

- If there are no valid data records in either Block #1 or #2 when requested by the DMS, the NIS Remote Terminal will respond with a busy flag in the status word as defined by MIL-STD-1553B.

- A 250-msec sample period will be assumed for NIS-related simulation and adapter software.

5.3.1.2. Optics Sensor (Stingray). The current Optics Sensor (OS) interface is not tailored to the VIDS DMS application. The sensor data are output continuously over a serial link. These data include all information necessary for the VIDS DMS processor. Also included are instrumentation data which are not useful to the DMS processing. In general, the vast majority of data are not of value to the DMS. The current interface is described in the following paragraphs. Modifications will also be discussed.

Current Interface Concept. The current interface continuously outputs over a serial instrumentation link. 2,800 bits are repetitively broadcast at 33.3-msec intervals, which equates to 280 data bytes/33.3 msec. Among these bytes, four to eight may have data which must be stripped out for handoff to the DMS. At the present time, these bytes are not identified by relative position in the data stream. Also, the data content of these bytes are not defined. Figures 5-6, 5-7, and 5-8 illustrate the OS interface. Comments related to these figures are listed below.

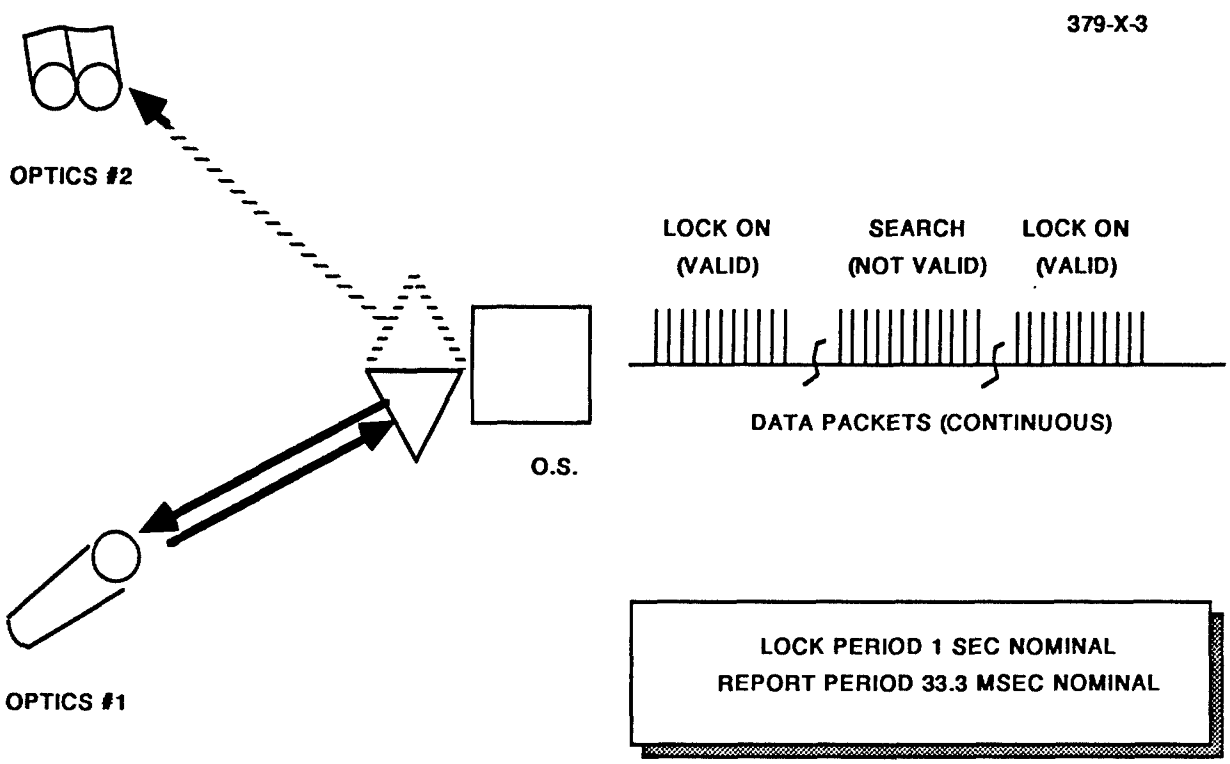


Figure 5-6. Optics Sensor Concept

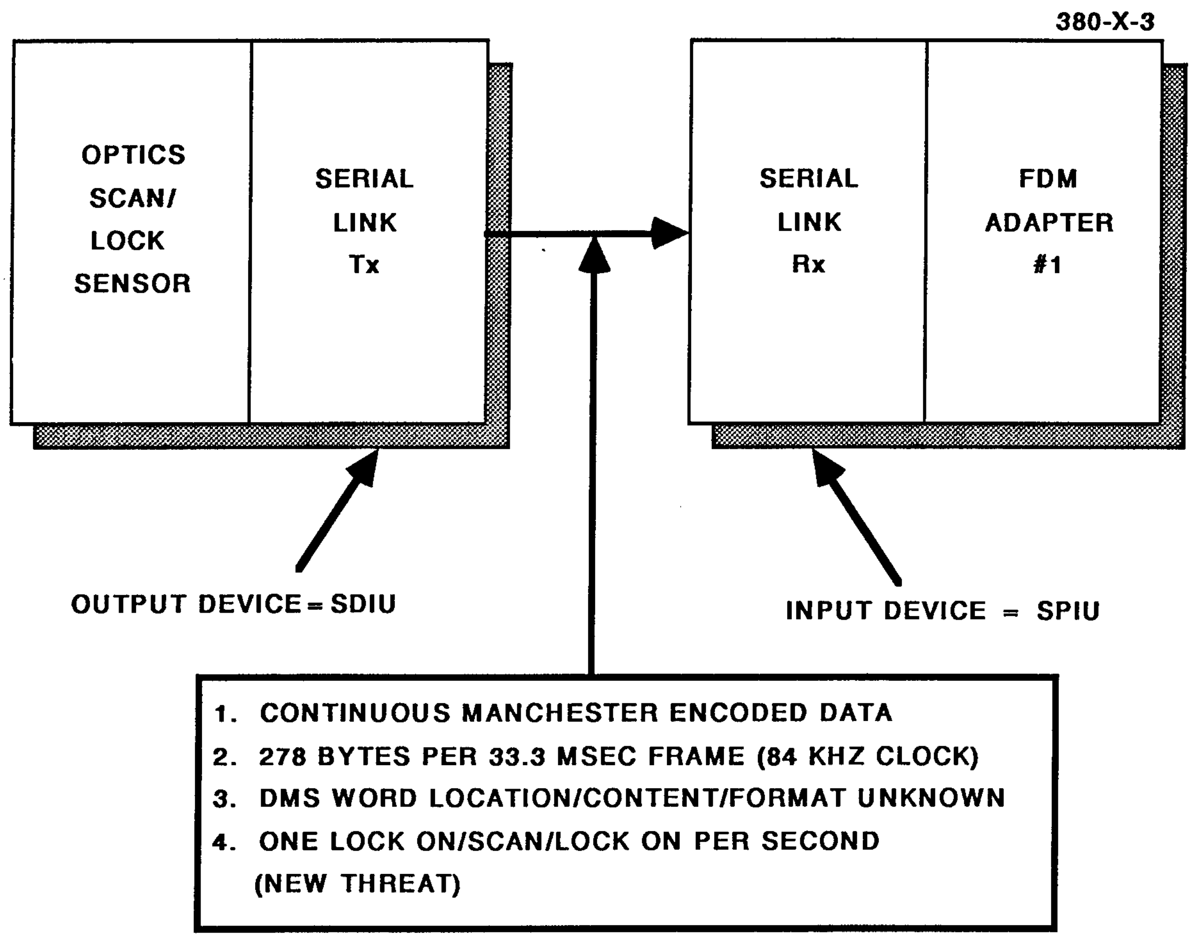


Figure 5-7. Optics Sensor Interface

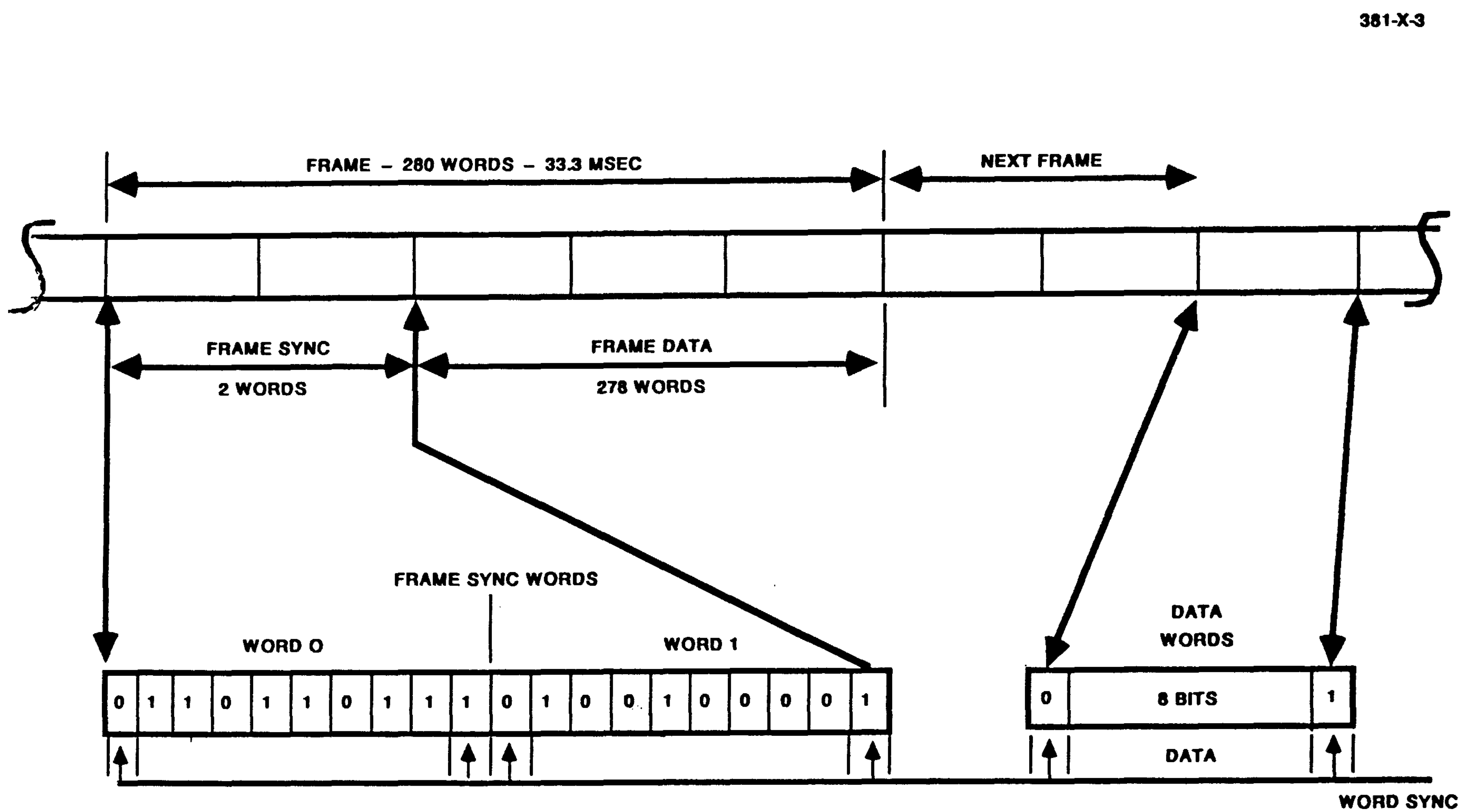


Figure 5-8. Optics Sensor Instrumentation Data Format

- The OS sweeps through a raster scan to detect optical threats. When an optical threat is detected, the system will lock and engage the detected optics for a predetermined period of time. During that time, the OS locates the optics device in azimuth, elevation, and range. The precision of threat location is far greater than that which is necessary for the VIDS DMS. Identification of optics is limited.

- As the OS scans, it also "terrain follows." Embedded in the interface data will be a pointing elevation value which must be added to the threat elevation to determine its true position relative to the vehicle.

- The OS lock-on period is greater than 0.5 seconds. During that time, new threats are not detected. Thus, new threat data will be given to the DMS infrequently (typically one second).

- Also included in the interface data are bits indicating whether the OS is scanning or locked on. During lock-on, only one input is required by the DMS even though the location data are repeated and improved upon during the lock-on period. The improvement is from precise to very precise and therefore is of little additional value to the DMS. The DMS must track transitions between scanning and lock-on to differentiate between old and new threat data.

It was premature in 1984 to suggest modifications to the existing OS interface although, to be consistent with the VIDS, a MIL-STD-1553B Remote Terminal should be provided in the OS. Because of the complexity of the OS and its relationship with other system elements, it is not yet reasonable to define timing or data covenants which might be necessary for the MIL-STD-1553B Remote Terminal. Following are some comments related to a MIL-STD-1553B type interface:

- For the VIDS to be effective, the DMS must be able to call for any new threat data on demand. As a minimum, threat data must indicate location in azimuth, elevation, and range. Repetitive data which does not provide new information should be stripped out so that only one data report is transferred for each detected threat. In other words, given the current scan/lock-on sequence, only one threat would be reported for each lock-on.

- It would be preferable for the OS to calculate the true elevation relative to the platform, thus eliminating the need for the DMS to handle terrain-following data.

- The DMS controller may request new OS threat data at intervals of approximately 200 msec. A new threat would be reported if the sensor has transitioned from a lock-on to a scan to a lock-on since the last controller request. A busy-status flag would be required to indicate no such sequence. A non-busy status would indicate a new threat lock-on with data to follow. In no case would more than one new threat be reported during a response.

- The simulation and adapter software will be based on the current interface.

5.3.1.3. Laser Sensor. The existing interface for the Laser Sensor was established for compatibility with the ALR-69 family of radar warning systems. This interface is not necessarily the best interface for the VIDS DMS. In the following paragraphs, the current interface is outlined and recommended modifications are defined.

Current Interface Concept. The Laser Sensor is analogous to a radar warning system. It senses coherent laser light pulses and collects bearing and time-of-arrival data from these pulses. Pulse data are sorted and analyzed to determine threat type. For each threat detected, a data word containing bearing and type is generated and handed off to the radar warning processor for tracking and display purposes. These data words are generated as often as possible and are repetitive for each threat while that threat is illuminating the sensor platform. The repetition is a function of the sensor environment, the sensor processing time, and the number of pulses required for sensor analysis. For the radar warning system, it is desirable to maximize the number of data words derived from each threat engagement. For the VIDS data management system, maximizing the number of data words is not necessarily desirable. Repetitive data with no new information generated by the current Laser Sensor interface creates an excessive tracking requirement for the DMS and could place an undue burden on the DMS time budget.

Figures 5-9 and 5-10 illustrate the existing interface. Figure 5-11 illustrates the modified interface as applicable to the VIDS DMS. Comments related to the modified interface are as follows:

- The interface timing is based on a sample period during which the sensor collects and analyzes data from the environment. The sensor will save all detected threat data resulting from the analysis until the end of the sample period. At that time, data will be called for by the DMS through the MIL-STD-1553B controller. A block transfer of all detected threat data acquired during each sample period will thus be output to the DMS at the end of each sample period. The sample period will be from 100 to 200 msec depending upon Laser Sensor processing requirements.

- During the sample analysis period, the sensor is expected to compress some of the threat data. Some correlation of data acquired during the period will be accomplished to avoid duplication of threat information. The intent is to achieve only one threat report for each threat detected during the sample period.

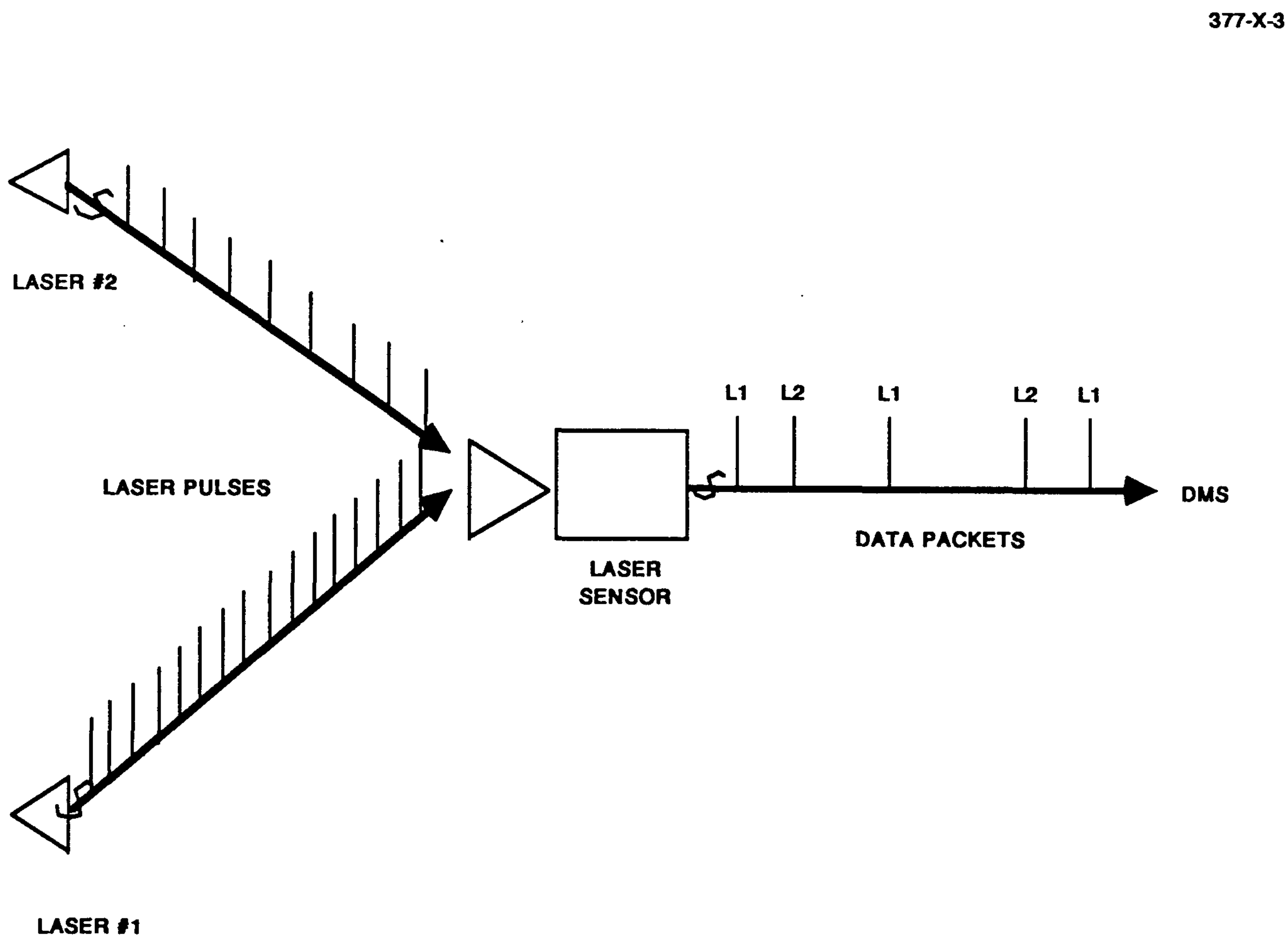
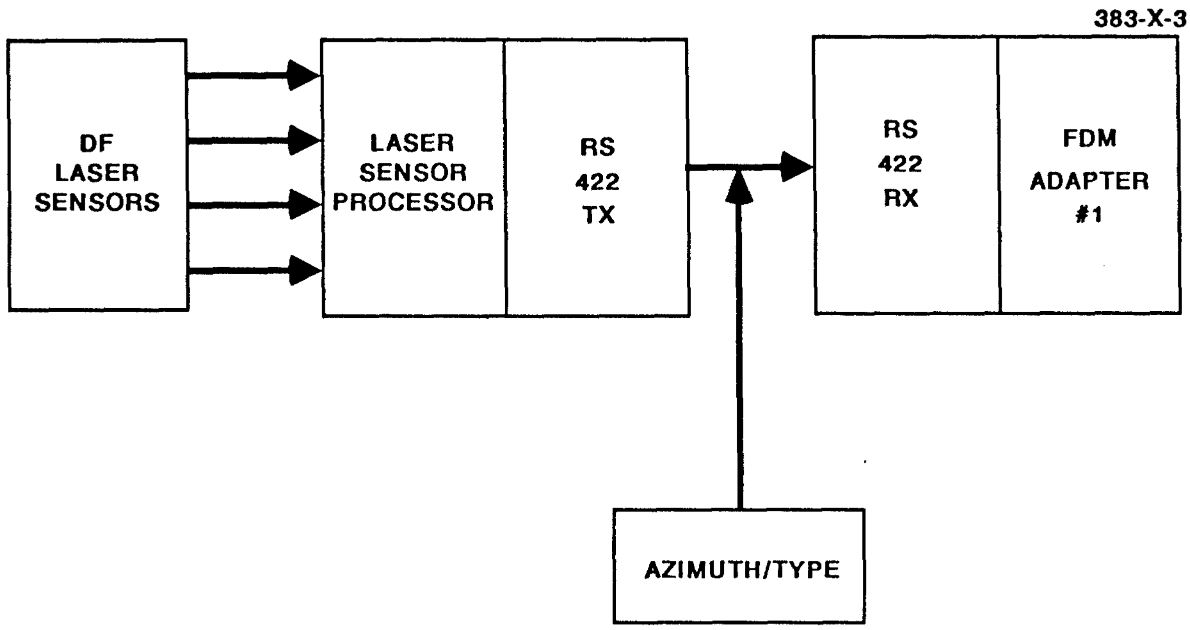


Figure 5-9. Laser Sensor Concept



1. ONE THREAT WORD PER ANALYSIS
2. REPETITIVE THREAT WORDS
3. TIMING DETERMINED BY THREAT RATE AND PROCESSING TIME

Figure 5-10. Laser Sensor Interface (Current)

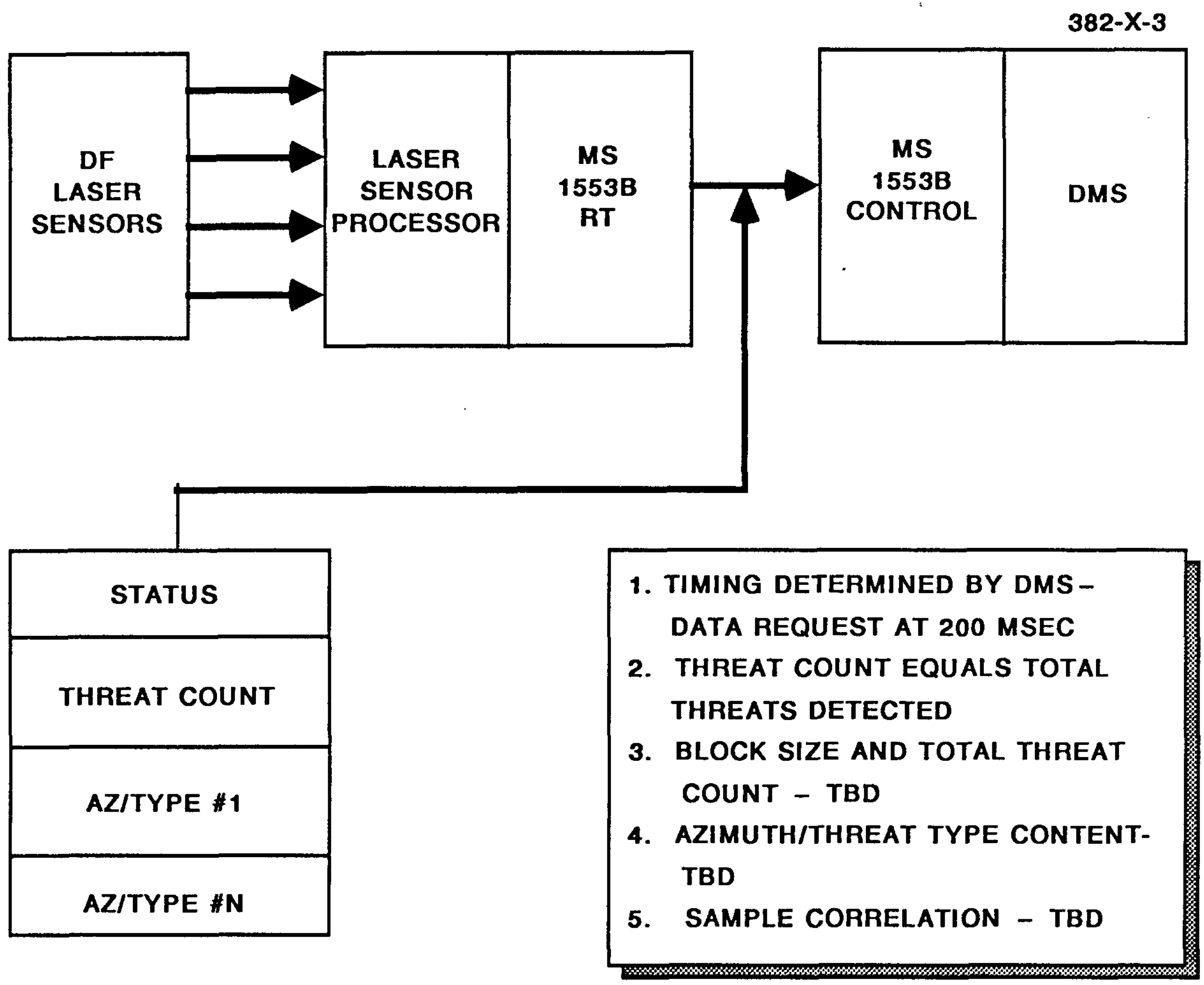


Figure 5-11. Laser Sensor Interface (Future)

- The Laser Sensor will not track threats between sample periods. It is not required to report that a threat has changed mode or position or has terminated its engagement with the sensor platform. All sample-to-sample tracking will be accomplished in the DMS.

- For timing and tracking, the model presented in Figure 5-11 will be assumed for the VIDS FDM simulation adapters. However, the hardware interface will be as indicated in Figure 5-10.

5.3.1.4. Passive missile detector. No information was available at the time of the design freeze.* Assumptions for the purpose of simulation are as follows:

- The interface is a parallel block transfer. A new data block may be transferred every 250 msec. Each block may have data for up to four threats.

- The parallel blocks will be received by Adapter #1 upon demand by the PMD Sensor.

- Each threat report will consist of 4 data bytes, one byte each for azimuth, elevation, range, and threat type.

- Location data will be resolved to 8 bits. Field-of-view assumptions are as follows:

  Azimuth = 0 to 360 degrees
  (zero byte value = 0 degrees)

  Elevation = to +90 degrees
  (zero byte value = -90 degrees)

  Range = 0 to 10,000 meters

- The sensor will not track threats between the 250-msec periods. All tracking will be accomplished by the DMS.

- Each of the four threats reported in each period will be unique. Data will not be duplicated for any threat.

5.3.2. First Adapter Unit (AU No. 31). Adapter Unit No. 31 was originally designed as two separate adapter units, Adapter Unit No. 3 and Adapter Unit No. 1. The purpose of Adapter Unit No. 3 was to accept simulated threat emissions from the TESS via an RS-232 line and to transform the SIPs into realistic sensor detections expected from the VIDS vehicle. Adapter Unit

---

* It was learned prior to publication of this report that the AN/AAR-47 PMD now in development by Honeywell will contain a MIL-STD-1553B interface. Data on the information output by the sensor was not available.

No. 1 was to accept SIPs from Adapter Unit No. 3 as if the SIPs came from real sensors and to buffer the normalized SIPs for transmission to the Bus Controller via the MIL-STD-1553B bus. Because of lack of well defined output formats for the developmental sensors, it was decided that Adapter Unit No. 3 and Adapter Unit No. 1 would be combined into Adapter Unit No. 31.

Adapter Unit No. 31 is designed as a temporary interface between the 1553B Bus Controller and the various sensor devices. It accepts SIPs from the TESS and buffers the IPSs for transmission to the Bus Controller. All the SIPs are normalized into a standard 1553B data format before the SIPs are transmitted to the Bus Controller. A block diagram of this Adapter Unit is shown in Figure 5-12.

The Adapter Unit consists of an Intel SBC 86/12A, a single board computer (SBC), and a separate board for input/output (I/O). The I/O board consists of the following:

- 1.024 x 16-bit of read/write memory
- MIL-STD-1553B dual redundant input/output port
- Four programmable serial ports (RS-232, RS-422, etc.)
- One 16-bit parallel input port
- One 16-bit parallel output port
- General purpose controller for multibus input/output access and peripheral direct access to the onboard read/write memory.

The data within the Adapter Unit flows in only one direction -- from the TESS via the RS-2343 line--and is buffered according to the type of sensor output (serial or parallel).

The SIP from the TESS consists of six 16-bit words and the format is as follows:

- Sync Word
- Threat Identification
- Azimuth (degrees)
- Elevation (degrees)
- Range (degrees)
- Checksum

The high byte is sent serially first before the low byte of each word.

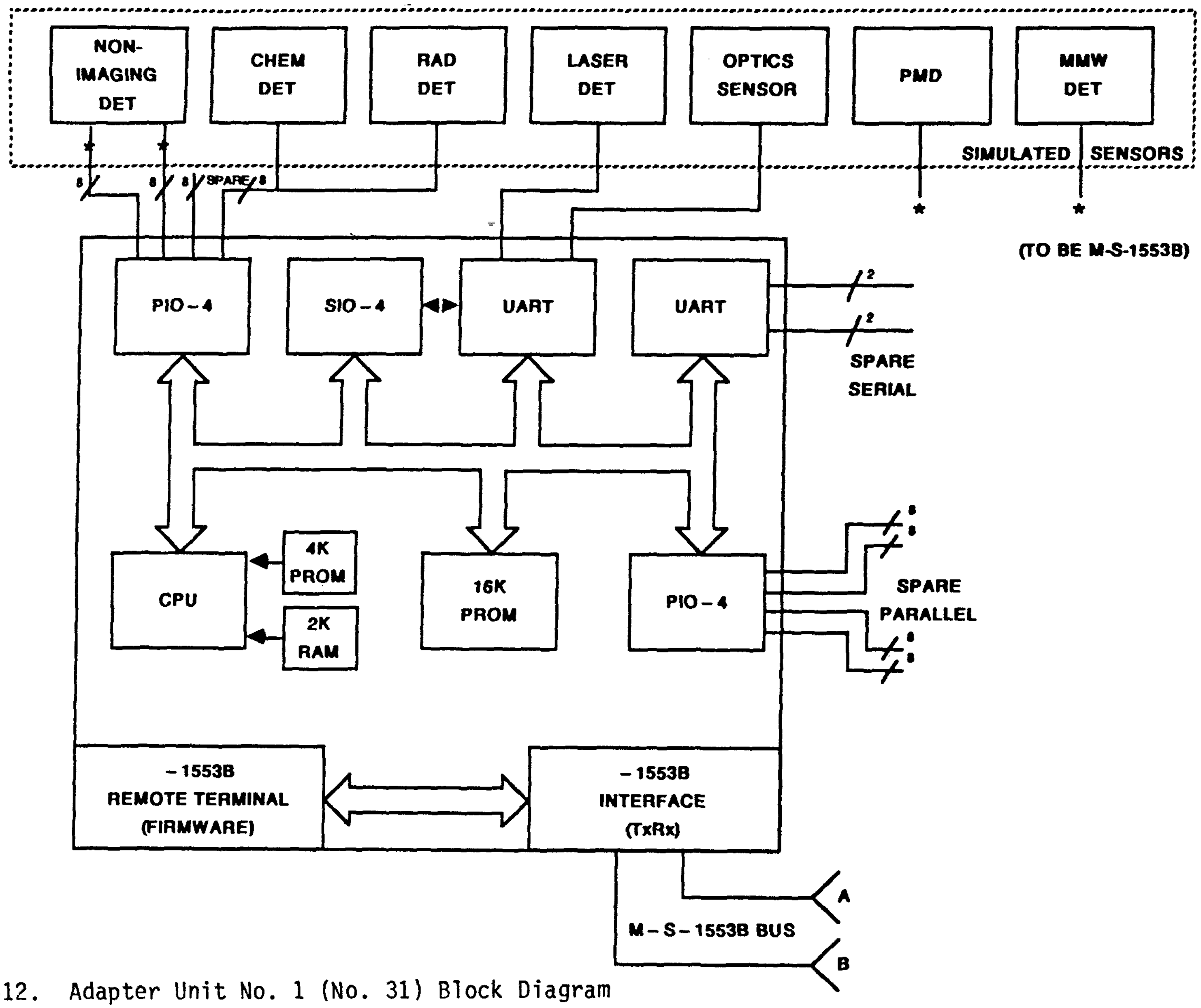


Figure 5-12. Adapter Unit No. 1 (No. 31) Block Diagram

The Adapter Unit removes the sync word and the checksum, and then buffers the SIP according to its sensor type. The Adapter Unit must wait for the Bus Controller to poll that particular sensor before it can transmit the data. The format for transmission for the 1553B is as follows:

Word 1: Command word to 1553 transmitter/receiver chip

Word 2: 1553 receive command transmitted over the bus as command to the remote terminal

Word 3: Undefined

Word 4: 1st data word of output message (device number)

Word 5: 2nd data word of output message (input type)

Word 6: 3rd data word of output message (threat identification)

Word 7: 4th data word of output message (azimuth)

Word 8: 5th data word of output message (elevation)

Word 9: 6th data word of output message (range)

Word 10: Status word transmitted by the remote terminal back to the Adapter Unit No. 31

The device number and input type are predefined as 1 for all data originating from TESS.

The buffers which are used to store the different SIPs for each sensor type are FIFO (first in/first out) queues. Pointers are used to locate the beginning and end of the queues. SIPs are always added to the beginning and end of the queues. SIPs are always added to the beginning of a queue and taken off the end of a queue. Once a SIP is transmitted to the Bus Controller, the pointer to the end of a queue (or last SIP transmitted) is repositioned to point at the next SIP to be transmitted. In the case of a bad transmission, the Bus Controller requests a retransmission. This entails having the end pointer of the last polled sensor repositioned back to the last SIP transmitted. Immediately following the request for retransmission, the Bus Controller must poll the same sensor again. The Bus Controller attempts two retrys before aborting the poll for that sensor.

Details of the hardware and firmware for the adapter units are described in a separate document.

5.3.3. Data Management System (DMS). The VIDS Data Management System for the Feasibility Demonstration is based on a single-board computer utilizing a Motorola MC68000 microprocessor chip. The overall CPU board is an enhancement of the Stanford University Network (SUN) design to provide multibus interface. A small amount of read-only memory (ROM) is on board for CPU management and a large amount of onboard RAM is available for instant ("no wait" state) direct memory access (DMA).

Three additional boards provide one megabyte of RAM for bus DMA, I/O ports for external mass memory software development tools and program store download, and dual MIL-STD-1553B terminal controller for the VIDS peripherals. The present FDM design provides that the DMS run its programs by downloading from a disk file via RS-232 from the host workstation. However, future implementations will employ an internal card containing 512K bytes of programmable ROM (EEROM) to eliminate the need for disk drives, thus enhancing mobility and reliability in the DMS.

The present FDM version of the VID DMS is contained in an enclosed card cage suitable for mounting in a test vehicle. The enclosure is approximately 7 inches by 12 inches by 16 inches, including blower and 28-volt DC power supply. Photographs of the FDM DMS are shown in Figure 5-13, and a block diagram of the DMS is illustrated in Figure 5-14. This shows the overall configuration of the DMS bus architecture, the arrangement of the printed wire boards and their component functions.

A summary of the processor specification is as follows:

Processor Clock Rate	8 MHz (25 MHz, future)
Local Memory Cycle Time (Read)	500 nsec
Electrical Power (One Megabyte RAM):	
+5 V DC $\pm$ 5%	6.4 amp (typical), 6.7 amp (maximum)
+12 V DC $\mp$ 5%	20 milliamps
-5 V DC $\mp$ 5%	50 - 150 milliamps
Operating Temperature	0° to 55°
Storage Temperature	-25° to 70° C
Physical Size	17" long by 10" high by 7 1/2" wide

5.3.3.1. Description of CPU and memory boards. The principal board, the CPU, is a high performance implementation of the Motorola 68000 32-bit microprocessor in an IEE796-compatible configuration. A high-speed local memory bus facilitates an optimum partitioning of CPU and I/O function on one board with local memory on a second board. This permits full-speed operation when accessing local memory since multibus arbitration and synchronization overhead time is not incurred on every memory cycle.

The processor provides a number of I/O and memory enhancements over the SUN family of 68000 IEEE796 SBCs while maintaining upward software compatibility with existing designs, with the exception of the 16-bit parallel input

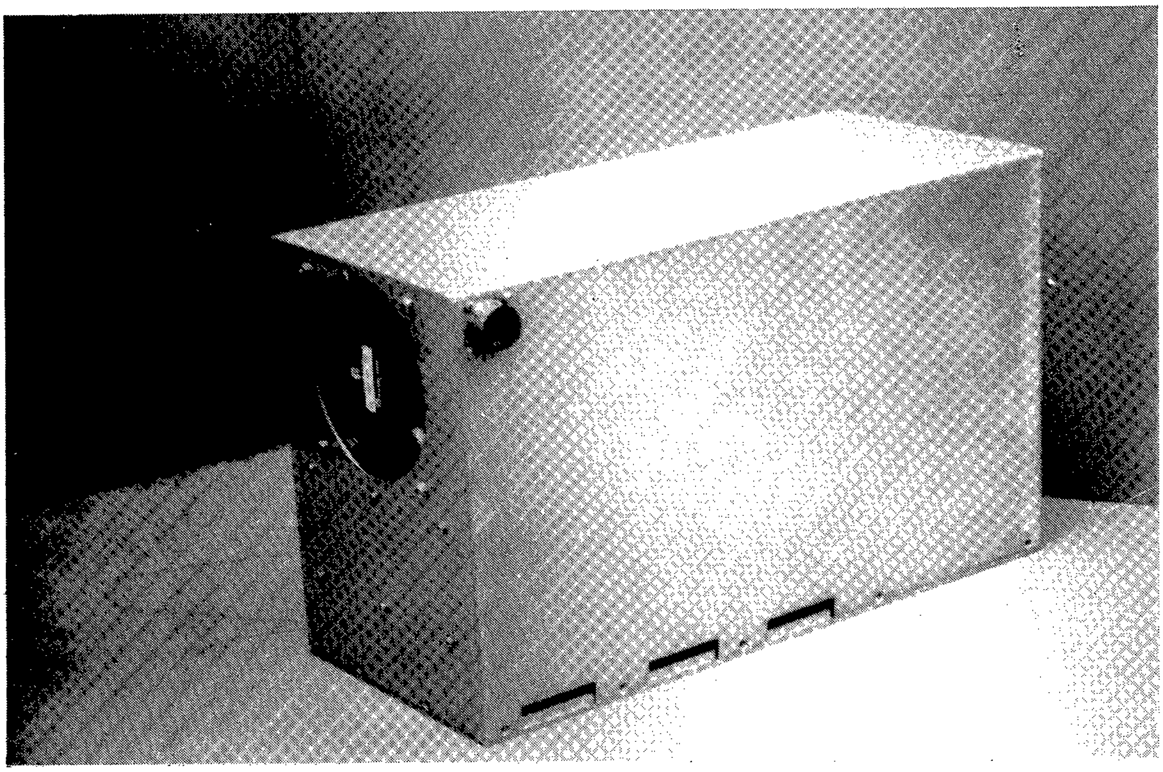
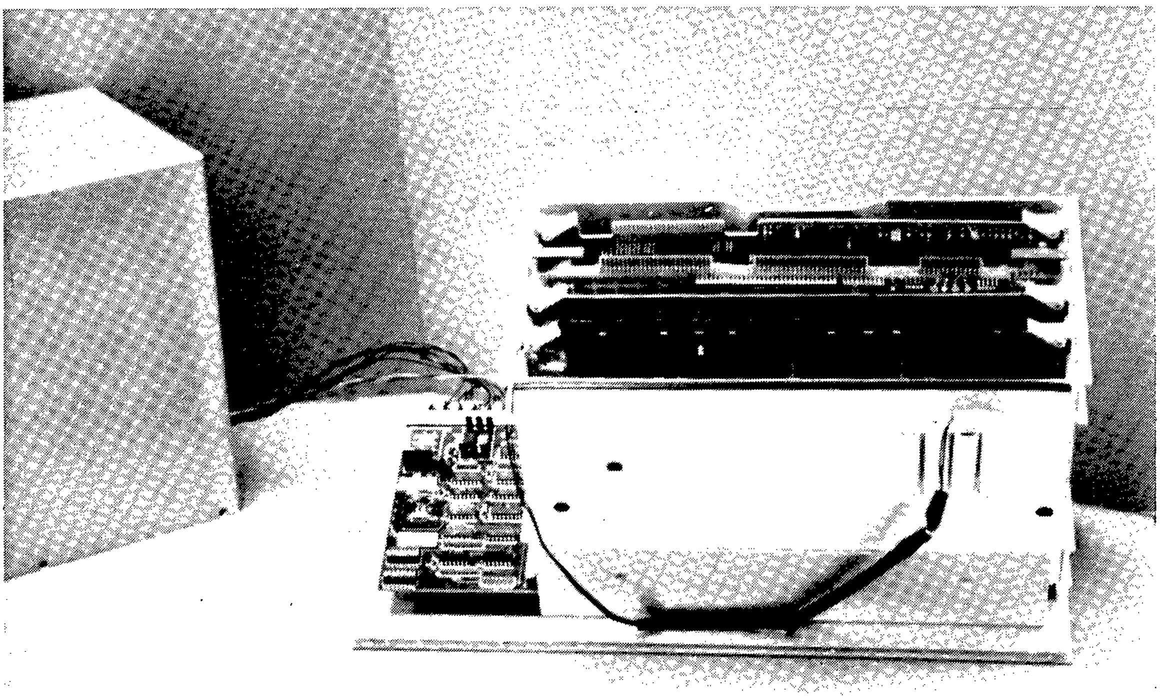


Figure 5-13. VIDS FDM Data Management System
(A) VIDS DMS; (B) DMS Internal Card Cage



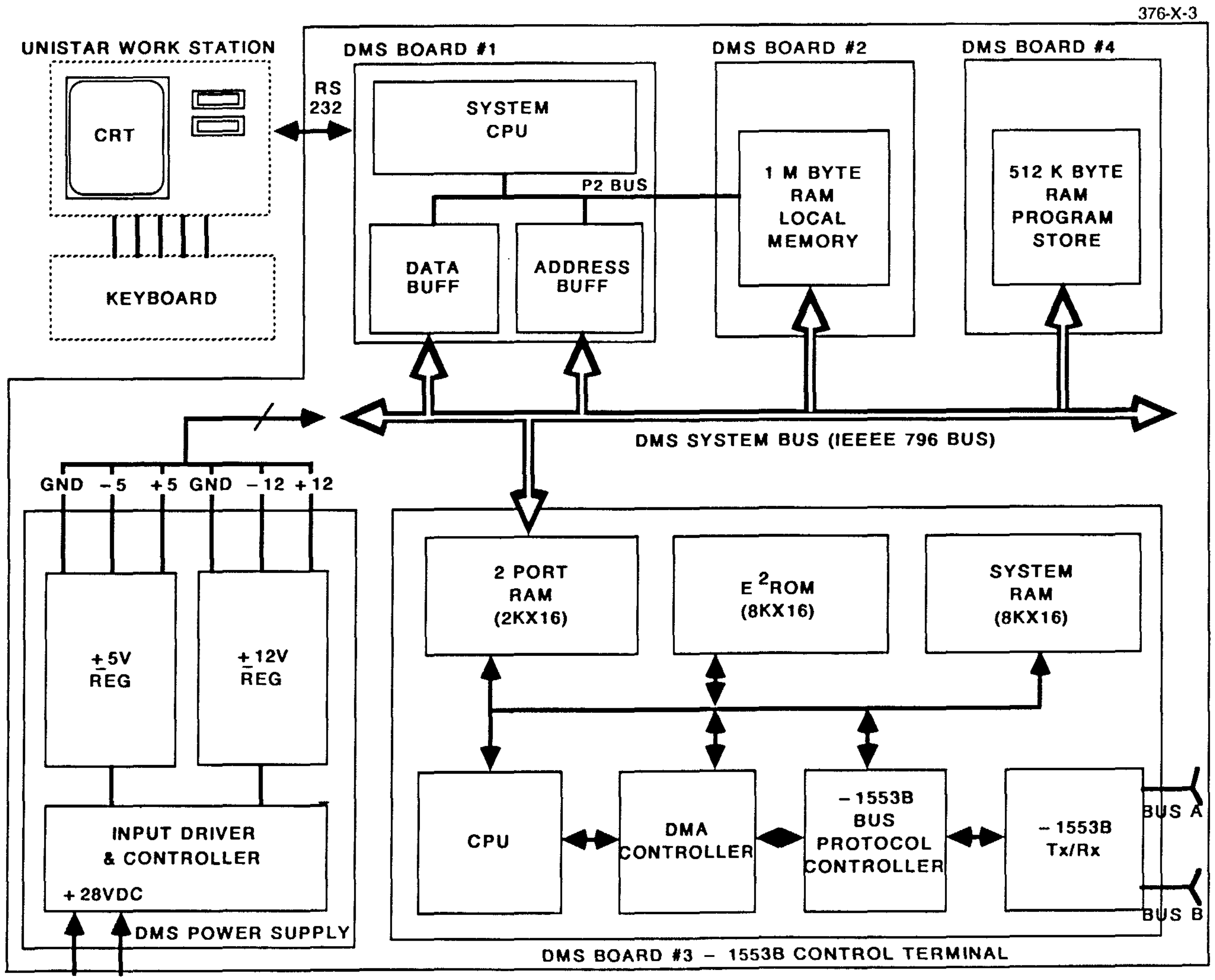


Figure 5-14. VIDS DMS Block Diagram

port. I/O addressing is also compatible using spare I/O address slots of the SUN design. Memory management functions are also completely software-compatible.

Two-level memory management architecture (consisting of separate segment and page maps with a context selection register) allows simultaneous mapping of up to sixteen process contexts. This is an advantage over other architectures which require reinitialization of information on each processing switch. The use of high-speed static RAMs avoids the performance penalty for the memory management hardware function.

The local memory synchronous bus is a private 60-conductor ribbon cable with edge connectors. Local memory can be accessed either directly by the processor or from the IEEE796 bus in a "cycle steal" fashion. This provides dual-port operations for DMA devices or multiprocessor configurations. The interface also proves periodic memory refreshing in the same CPU "cycle steal" fashion.

The bus arbitration logic supports the IEEE796 bus multimaster priority resolution system for both serial daisy chain or parallel schemes. The full 24-bit bus address space (up to 16 megabytes) for addresses generated on board to the bus and for addresses from external devices coming on board is also supported. A PROM-based address decode provides a variety of incoming address recognition schemes.

A simplified block diagram of the central processing unit (CPU) board is shown in Figure 5-15.

The DMS CPU board is designed to function with a companion local memory board by communicating over a high-speed private memory bus. The board utilizes 64K dynamic RAM technology and can be expanded by increments of 128K bytes from 256K to 1 megabyte.

Additional technical information regarding topics of:

- Memory management and protection
- Real-time calendar clock
- System timers
- Interrupts and exceptions
- PROM monitor

are described in the Unistar users' manual, not attached to this report.

5.3.3.2. DMS I/O. There are presently three input/output lines available to the VIDS FDM DMS:

The 1553B bus
RS-232 line from the Unistar workstation
RS-232 line to the cathode ray tube (CRT) screen of the Host workstation

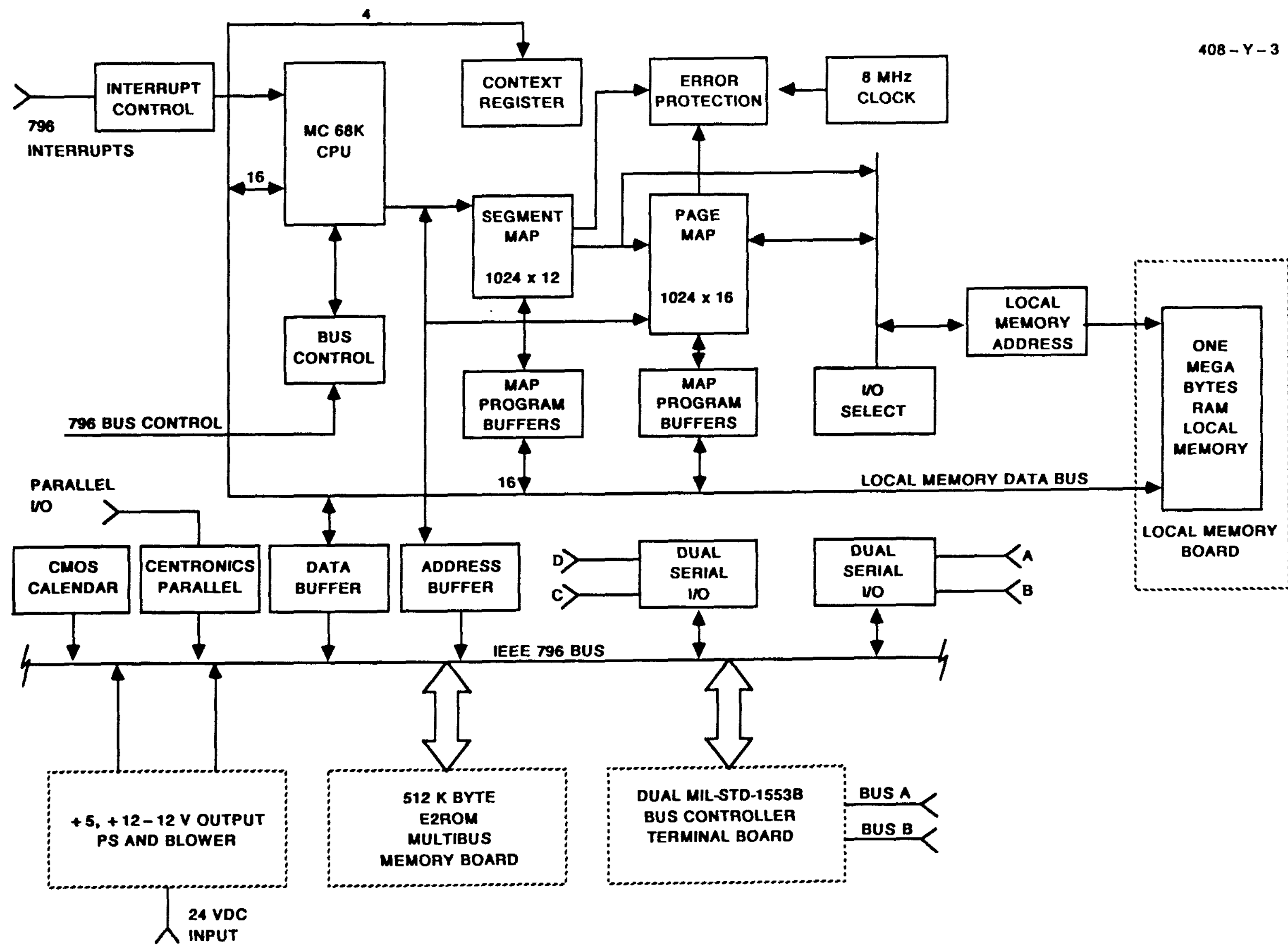


Figure 5-15. VIDS DMS CPU Block Diagram

The 1553B bus is controlled by the 1553B Bus Controller board in the VIDS FDM DMS. The 1553B Bus Controller polls each of the simulated sensors in Adapter Unit No. 31 for sensor data. The 1553B Bus Controller also polls Adapter Unit No. 2 for Control Panel input from the crew. Data is transmitted serially via the 1553B bus to the VIDS FDM DMS for processing. Once the data is processed, display data, audio alert information, counteraction recommendations, and counteraction to be initiated are transmitted serially via the 1553B bus again back to Adapter Unit No. 2 for distribution to the appropriate device. For a description of the 1553B Bus Controller board, see Section 5.3.3.3, below.

The RS-232 line from the Unistar workstation is temporarily used for the download of the VIDS DMS run-time code immediately after power-up. For a complete description of the VIDS DMS download, refer to Section 5.3.3.3., below.

The RS-232 line connected to the CRT is used for debugging only during system integration and test, and can be detached from the VIDS FDM DMS without any ill effect.

5.3.3.3. MIL-STD-1553B Bus Controller Terminal. The VIDS DMS contains a Bus Controller which functions in the management of data transferred between the VIDS peripherals and the DMS via a MIL-STD-1553B data bus. The Bus Controller is a complete subassembly consisting of dual 1553B hybrid bus transceivers, buffer memory for temporary storage of data and address code transferred either to or from the transceivers, and a microprocessor to function as a standard CPU. The Controller is designed with an internal private bus for local memory and buffer access, and is compatible with other CPUs and peripherals by use of IEEE796 multibus. The general specifications of the controller are:

- Programmable operating modes
  - Bus Controller
  - Remote Terminal
- 796 Bus Compliance: Master/Master
- Dual 2K x 16-bit, dual-port buffer for internal message handling
- Single 32K x 16-bit, dual-port RAM "Mail Box" to interface with 796 Bus
- MC68000 Microprocessor, Local CPU
- Multibus Form Factor Card: 12" x 6.75" x 0.5"
- Power Requirements: < 8.0 amp, + 5V DC
< 0.3 amp, +12V DC
< 0.3 amp, -12V DC

Bus Controller Hardware. The Bus Controller provides a MIL-STD-1553B Bus Interface for the IEEE796 Bus (Multibus). It can operate as a 1553B Bus Controller or as a 1553B Remote Terminal. The Controller contains dual 2K x 16-bit buffers (mapped as IEEE796 Bus Memory) for passing data between the host and the 1553B Bus. The buffers are designed so that the host (796 Bus) system has exclusive access to one buffer, while the 1553B protocol logic has exclusive access to the other buffer. The buffers may be swapped between 1553B messages to allow the host system to retrieve data from its buffer without interrupting operation of the 1553B protocol circuitry. Buffer swapping may occur automatically or under program control. Data may be transferred to and from the host system in quantities of 16-bit words.

The Bus Controller also has dual 64 x 16-bit FIFOs which are used to hold status and error information when the Controller is operating as a remote terminal. Each FIFO is paired with a memory buffer so that the current FIFO will contain the current status information. FIFO data is accessed through a 16-bit 796 Bus I/O port, but may also be accessed as two 8-bit bytes. Storing this status information in a FIFO allows easy access to the received command word so that it may be decoded to determine the address where the data associated with the message was stored or retrieved. The message word count may also be determined from the command word.

The Bus Controller provides the capability for two interrupts -- one of which may be programmed to occur at the end-of-message and the other of which may be programmed to occur on one or more error conditions. The Controller contains a RAM array which allows the disabling of the response and end-of-message interrupt when operating as a remote terminal.

The Bus Controller also provides a status register, a control register, and several other special purpose registers. The status register allows the reading of status and error information about 1553B transactions. The control register allows control of buffer swapping and the initiation of 1553B Bus transactions.

In all Bus Controller transactions, the following steps are required to initiate a transaction:

1) A command block must be loaded into the buffer memory.

2) The controller internal address register must be loaded with the starting address of the command block.

3) The host CPU must cause a buffer swap so that the 1553B protocol logic may access the command block.

4) The host CPU must issue a control strobe to alert the 1553B protocol logic that a command block is present.

After these steps have been performed, the 1553B transaction will occur. Any Remote Terminal responses and data words will then be written to the Bus Controller buffer memory. The data will be stored in the buffer memory starting with the first word after the command block. After the data has been stored, an end-of-message interrupt will be generated. If any errors were detected during the transaction, the invalid message bit in the status register will be set. The error register may then be read to determine the type of error that occurred.

The overall architecture of the Bus Controller and its relationship to the complete DMS is shown in Figure 5-16. The 8K x 16-bit, dual-port RAM functions as a family of mailboxes mapped as 796 Bus Memory. These mail boxes are also organized by sensor and counteraction or display message to facilitate storage of sensor data to be processed by the system or host CPU, and also as a message source for instructions to the display and counteraction subsystem (via 1553B bus). The use of this mailbox memory under control of the local Bus Controller CPU eliminates any bus contention or unnecessary processing cycles by the system CPU. Likewise, the 2K x 16-bit, dual-port RAM buffers are used as storage between the mailboxes and the 1553B transceivers. One buffer may transfer information to the 1553B bus while the other is transferring information from the 1553B Bus to the mailboxes. The private bus is the interface between the buffers from the 1553B transceiver to the mailboxes.

The pin connector signal list is compatible with IEEE796 format for the P1 connector. The P2 bus connector is disabled to permit the P2 to remain a private bus only. (The complete P1 connector listing is provided in the separate document on the design of the Bus Controller Terminal.)

The Bus Controller occupies 4K words of memory address space and 16 I/O port locations. The memory base address (hereafter referred to as MEMBASE) is switch-selectable and is capable of up to 24-bit addressing. The I/O port addresses are also relative to a base port address (hereafter referred to as I/OBASE). This address is also switch-selectable and provides the capability of 16-bit addressing. The Controller is capable of 16-bit data transfers for both memory and I/O operations.

Bus Controller Firmware. The following paragraphs describe the function of the firmware within the Bus Controller. Figure 5-17 shows the Bus Controller configuration.

The <u>power-up procedure</u> configures the internal registers of several system components properly and then tests the system for proper operation. The components which must be configured are the system timer, the memory controller (MAC), and the UART (8274-MPSC). For a complete description of the set-up procedures, see following sections.

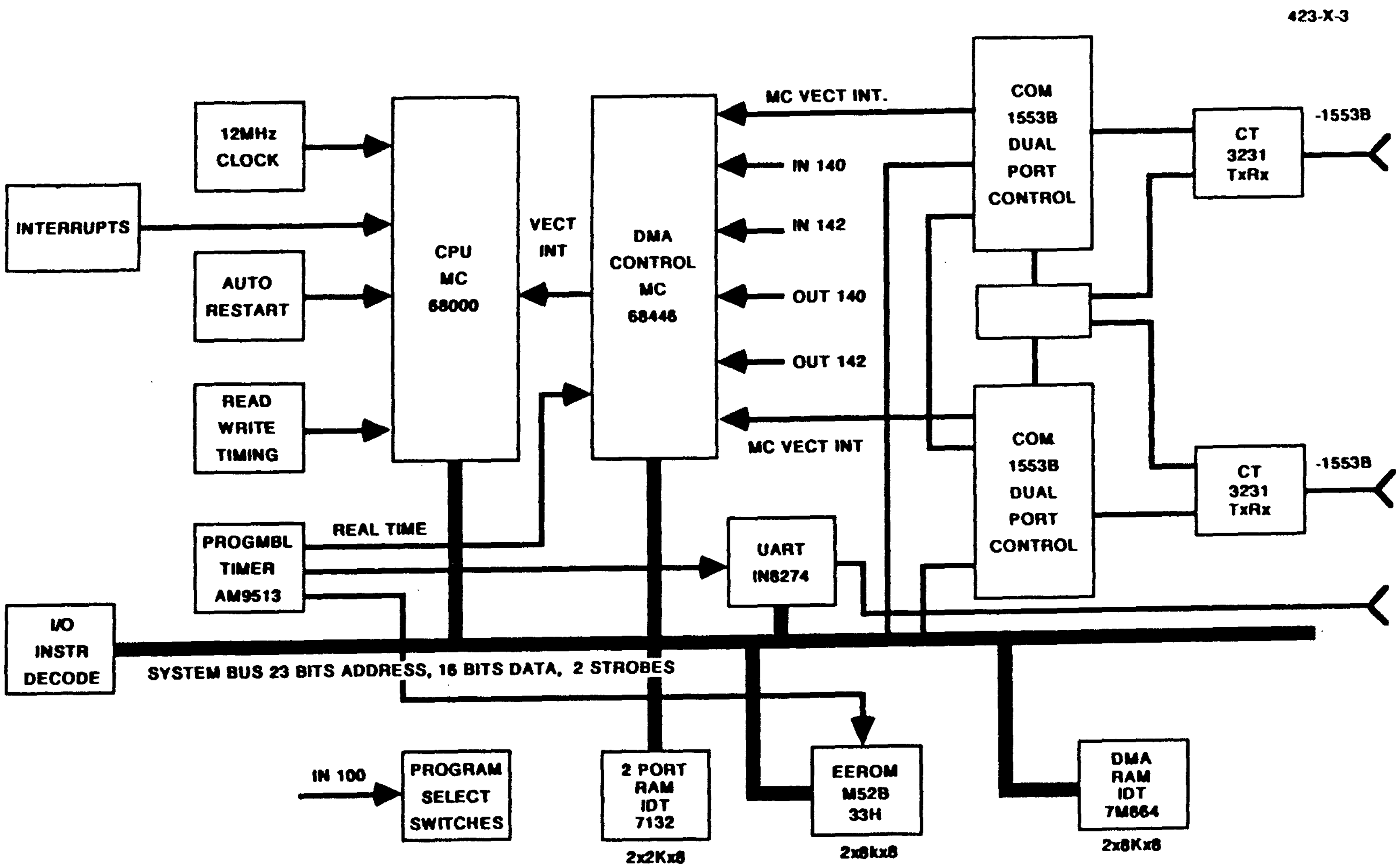


Figure 5-16. DMS Bus Controller Terminal

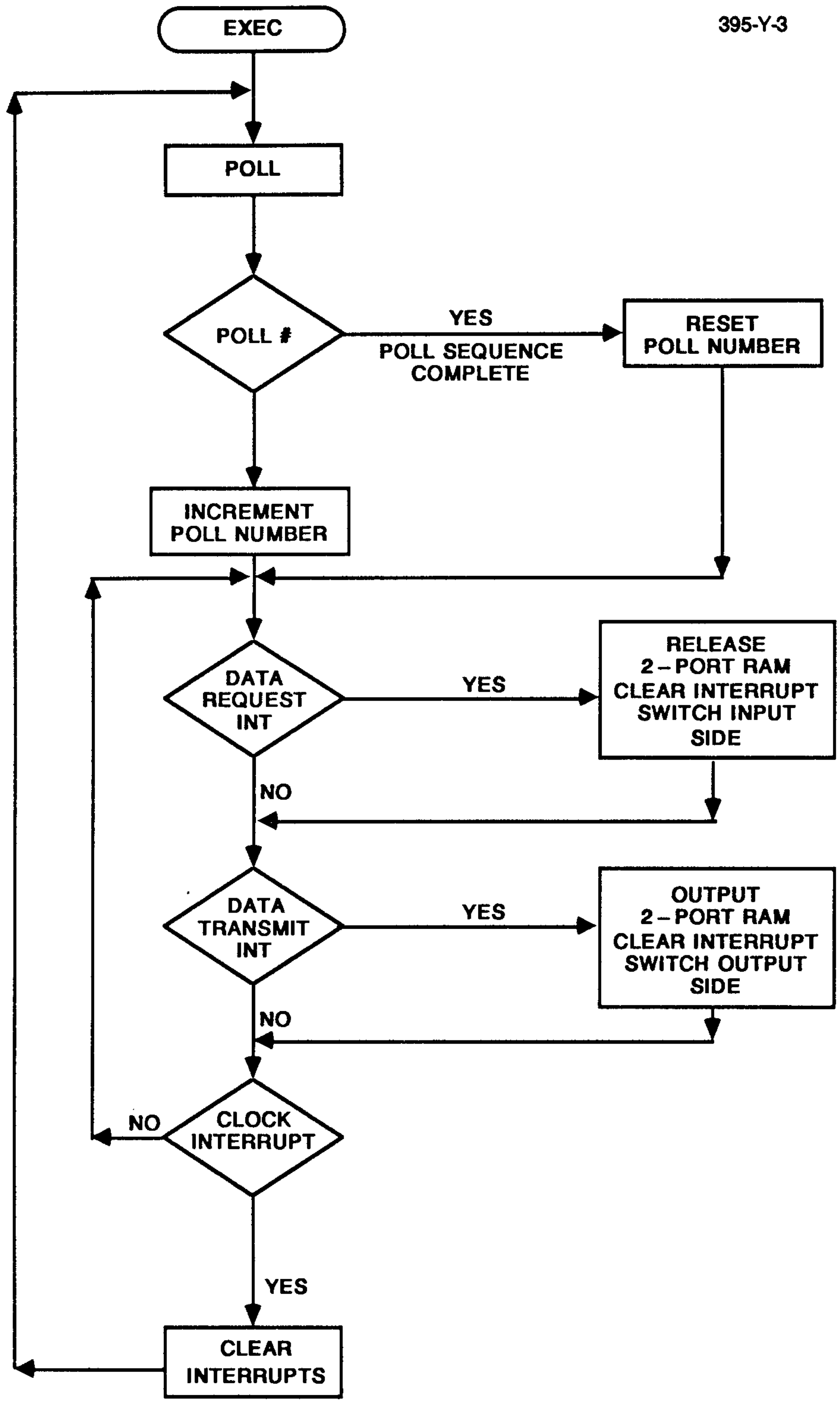


Figure 5-17. Bus Controller Firmware Flow

The order in which these actions are taken is essential. The system timer chip must be initialized immediately after power-up. This is necessary in order to prevent the system from resetting itself before the power-up routine is completed. This would be caused by a hardware reset counter in the system. The hardware reset counter is provided in the cage of a processor lockup. To ensure that the processor is not locked up, the processor must write to CR6 more often than every 650 ms. If this amount of time passes without a write cycle to CR6, the entire system is reset.

The time-critical element in the power-up routine is that before the system timer is properly set up, the reset counter will reset the system much more quickly than 650 ms. When power initially comes on, the reset counter will reset the system in 160 microseconds or 960 machine cycles. To alleviate this problem, the first possible write cycle in the power-up routine should be to CR6. The next task done should be to initialize the system counter, which will end the reset problem. One of the last things done in the power-up routine should be to write to CR6 in order to clear the counter before turning control over to the executive routine.

After the timer is initialized, the processor should mask all maskable interrupts. This is to ensure that the controller is allowed to finish its initialization procedures before attempting to enter normal operation.

Once the system has been completely initialized, the processor should go into a number of self-test routines. Two of these tests are to transmit a message out first over 1553 Bus A, then over Bus B, and to receive that same message over the channel not being used to transmit. The processor may then check to see that the transmitted message matches the message received. The same kind of test also should be run over system UART.

At the end of the power-up procedure, the processor should interrupt the DMS by writing to CR13 in order to signal the completion of the power-up/reset sequence. This signal allows the DMS to synchronize itself to the Bus Controller after the controller has gone through a reset.

The last thing done by the power-up routine is to unmask all interrupts and to set up for normal operation.

Two-way Data Transfer between the Bus Controller and the DMS is accomplished via a 2K x 16 2-port RAM and two pairs of unidirectional interrupt lines (see Figure 5-18). There are two distinct types of data transmitted across the RAM: one originating from bus senders, the other destined for bus receivers.

With two data types and two data flow directions, the first step in defining the dual-port organization is to divide the memory space into two sections. Data traveling to the DMS is loaded into the first section, while data received from the DMS will be loaded into the other section. Each memory section is further divided into two identical memory segments, thus

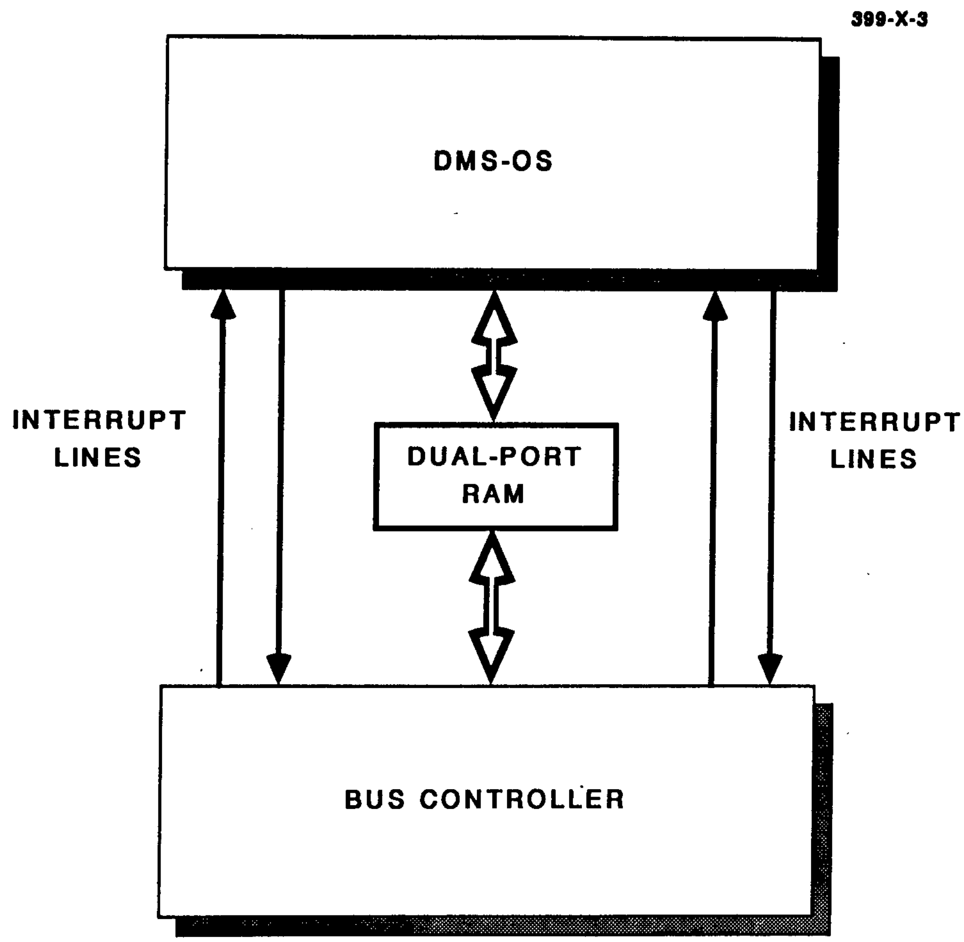


Figure 5-18. Interface Block Diagram

providing four memory segments in total. The purpose of having two identical memory segments is to avoid simultaneous reading and writing from the same word in memory. This is prevented by requiring the Controller to read from or write to one segment of a pair while the DMS is in control of the other segment in the pair.

Under this scheme, the DMS controls data flow via the two DMS-to-Controller interrupt lines. The two Controller-to-DMS lines are used to acknowledge a DMS interrupt.

At any given time, the DMS fully controls either RAM Segment 1 or 2 which contains an entire set of sensor data. When the DMS is through processing that data and requires new data, an interrupt to the Bus Controller occurs. After this interrupt is acknowledged, the DMS takes control of the opposite RAM segment which the Bus Controller has filled with the latest information. The Bus Controller then takes control of the free segment and begins filling it with new sensory data. In this manner, the Controller and DMS toggle between the segments they control.

DMS-to-Controller transfers are handled in the same manner, using RAM Segments 3 and 4 and the two remaining interrupt lines.

The one case in which this transfer methodology algorithm is not followed is when the Bus Controller detects a change in the Control Panel status. When this occurs, the Bus Controller asserts its data request interrupt acknowledge line. Thus, any time this line is asserted before the DMS has sent a data request, the DMS may assume that a change in Control Panel status has occurred.

As stated above, the dual-port RAM is divided into four segments (see Figure 5-19). Segments 1 and 2 are formatted in a mailbox scheme wherein the Controller places each threat report in a RAM position defined as a mailbox slot. Each slot holds information from a specific sensor; sensors with a multiple threat report capability are assigned the necessary number of slots to cover this capability. Other slots will contain information concerning Control Panel status and Bus Controller status (see Table 5-3). The last word in every mailbox slot is a status word for that slot. The first bit in that word serves as a valid data flag. When the Controller fills a mailbox slot with new information, it sets the valid data flag, and when the operating system reads this slot, it should then reset the flag. The flag prevents old data from being reread during subsequent read cycles. Before turning over any data packet to the DMS, the operating system is to place a time tag in the seventh word position.

Also included in Segments 1 and 2 is a word to indicate the current status of data in that segment (see Figure 5-20). This word is used by the operating system in the handling of the data to be transferred to the DMS applications software. The first bit will serve as a flag to the DMS indicating that the Control Panel status has changed and that DMS action may be required.

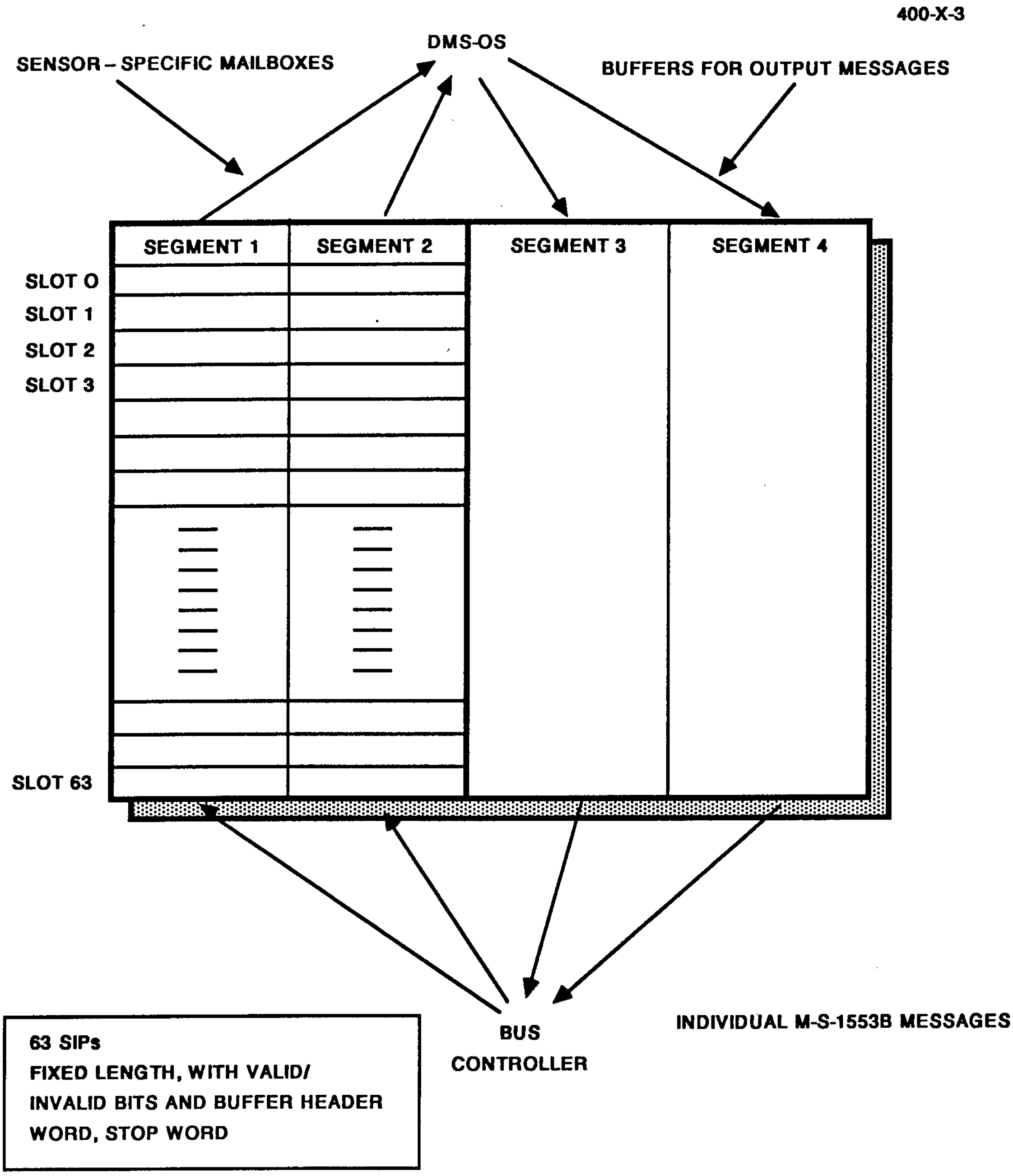


Figure 5-19. Memory Segmentation and Data Flow

Table 5-3. Dual-Port RAM Memory Map

01EFFF	
	Segment four
01EC00	
01EBFF	
	Segment three
01E800	
01E7FF	
	Segment two
01E400	
01E3FF	
	Segment one
01E000	

Table 5-3. Dual-Port RAM Memory Map (Continued)

MEMORY MAP OF SEGMENT ONE*	
01E3FF 1E0	Extra space
1DF 1D0	MMWave sensor - Slot No. 4
1CF 1C0	MMWave sensor - Slot No. 3
1BF 1B0	MMWave sensor - Slot No. 2
1AF 1A0	MMWave sensor - Slot No. 1
19F 190	Control Panel/NBC data
18F 180	PMD data - Slot No. 5
17F 170	PMD data - Slot No. 4
16F 160	PMD data - Slot No. 3
15F 150	PMD data - Slot No. 2
14F 140	PMD Data Slot No. 1
13F 130	Laser Sensor - Slot No. 8
12F 01E120	Laser Sensor - Slot No. 7

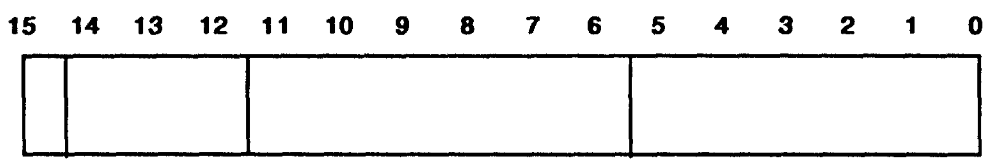
*The map of Segment Two is identical to the map of Segment One but offset by 400 (hex)

Table 5-3. Dual-Port RAM Memory Map (Continued)

01E120 01E11F 110	Laser Sensor - Slot No. 6
10F 100	Laser Sensor - Slot No. 5
0FF E0F0	Laser Sensor - Slot No. 4
E0EF 0E0	Laser Sensor - Slot No. 3
0DF 0D0	Laser Sensor - Slot No. 2
0CF 0C0	Laser Sensor - Slot No. 1
0BF 0B0	Optical Warning Data
0AF 0A0	NIS Data - Slot No. 10
09F 090	NIS data - Slot No. 9
08F 080	NIS data - Slot No. 8
07F 070	NIS data - Slot No. 7
06F 060	NIS data - Slot No. 6
05F 050	NIS data - Slot No. 5
04F 040	NIS data - Slot No. 4
03F 01E030	NIS data - Slot No. 3

Table 5-3. Dual-Port RAM Memory Map (Continued)

01E030	
01E02F 020	NIS data - Slot No. 2
01F 010	NIS data - Slot No. 1
001 01E000	Segment Status Word



BITS 0 - 5:	New Threat Report Counter
BITS 6 - 11:	Not Used
BITS 12 - 14:	Not Used
BITS 15:	Control Panel Status Flag

Figure 5-20. Segment Status Word

A group of bits in this status word is a count of the number of new threat reports which have been placed in that segment since the last segment exchange. After the operating system has discovered and processed that number of new data reports, it should stop searching the dual-port RAM for more data.

Memory Segments 3 and 4 are each used by the DMS OS to hand over a single standard output packet to the Bus Controller. The standard packet received by the operating system from the DMS applications software consists of eight words. When the operating system loads this packet into the dual-port RAM, two blank words must precede the packet, thus giving ten words to load into memory.

There is no wraparound capacity in filling these queues. The operating system must turn a segment over to the Bus Controller at any time up to the point where that segment becomes full. Once the operating system has signaled a data transmit interrupt, it must wait for the acknowledge interrupt before new data can be placed in the free segment.

The two DMS-to-Controller interrupts are non-vectored interrupts which are raised by putting out a specified address on the address bus. To raise a data request interrupt, signaling for new sensory data, the DMS must initiate a write cycle to address 1FF680(H). To raise a data transmit interrupt, which signals to the Bus Controller to take control of an empty new buffer, the DMS must initiate a write cycle to address 1FF700. Included in the hardware of the Bus Controller is an external reset option. The DMS may reset the Bus Controller by writing to address 1FF600.

The DMS OS will receive the Bus Controller acknowledge signals as interrupts 4 and 5. Interrupt 5 serves as the acknowledge to a DMS data request interrupt. This is also the interrupt line used to alert the DMS of a change in Control Panel status. Interrupt 4 serves as the acknowledge to a DMS data transmit interrupt.

Interrupt 3 is a signal to the operating system that the Bus Controller has just completed a power-up or reset sequence, and will come up in its initialized state. If the operating system has an interrupt to the Bus Controller pending, the receipt of this interrupt will signal to the operating system that the pending interrupt has been cleared without service and must be reinitiated.

When handling a Bus Controller-to-DMS interrupt, the OS must also clear that interrupt. To clear a data transmit acknowledge, a write cycle to 1FF480(U) should be initiated. To clear a data request acknowledge, a write cycle to 1FF500(H) should be initiated. To clear a controller reset interrupt, a write cycle to 1FF400(H) should be initiated.

There are two Bus Controller-to-DMS interrupts. The acknowledge interrupt to a DMS data request is raised by initiating a write cycle to $\overline{CR5}$. The acknowledge interrupt to a DMS data transmit interrupt is raised by the Bus Controller initiating a write cycle to $\overline{CR4}$.

The other Bus Controller-to-DMS interrupt is a power-up interrupt. To raise this interrupt, the processor must write to $\overline{CR13}$.

5.3.3.4. Mechanical and environmental. The contract required that the FDM DMS approximate the projected VIDS architecture in terms of size, weight, and power consumption. We packaged the DMS as a standalone unit (LRU) constructed in such a manner that it could be satisfactorily placed within a tank and used for testing as a system for integrating sensors. Although the DMS is not required to be fully MIL-qualified at this time, we have fabricated an enclosure based on a "stiff back" support for this card cage. It is surrounded by a cover of sheet aluminum and deep-brazed for electrical and mechanical integrity.

The envelope dimensions of the overall DMS housing are 7.5 in. wide, 10 in. high and 17 in. long. A dimensioned sketch of the DMS enclosure is shown in Figure 5-21.

Electrical connections to the DMS are via a male chassis connector, MS3147E27-50S. This brings unregulated 28 V DC from the vehicle power bus to the DC/DC converter inside the DMS and also provides termination for the extended MIL-STD-1553B data bus. Program download is presently accomplished via a flat ribbon cable entering one of the slots at the base of the housing. It can also be input via the MIL-STD connector over RS-232, although more time is required for the serial download. The eventual program will be stored on EEROM on the fourth printed wiring board and updated with program changes via RS-232 from a Memory Loader/Verifier at Depot or Intermediate Level Maintenance organizations.

In order to ensure the low cost/low risk technical approach, we demonstrated the proper operation of the software processing on a set of factory proven CPU, memory, and floppy disk driver cards as supplied by the workstation manufacturer, Callan Data Systems. These commercial quality printed wiring boards (PWBs) must be updated with MIL-qualified PWBs in the next generation phase of the FDM project.

Also we note the impracticality of attempting to install the 8-inch (one megabyte) floppy disk in (or on) the instrumented combat vehicle. We have therefore substituted an additional (1/2 megabyte) RAM board on which the software program (application packages and operating system) can be downloaded from the workstation, and then the cables can be detached to provide the independence of a stand-alone embedded microprocessor. This RAM would be replaced by an equivalent PROM (EEROM) PWB program store. This board has already been designed and could be fabricated early in the next phase of the program.

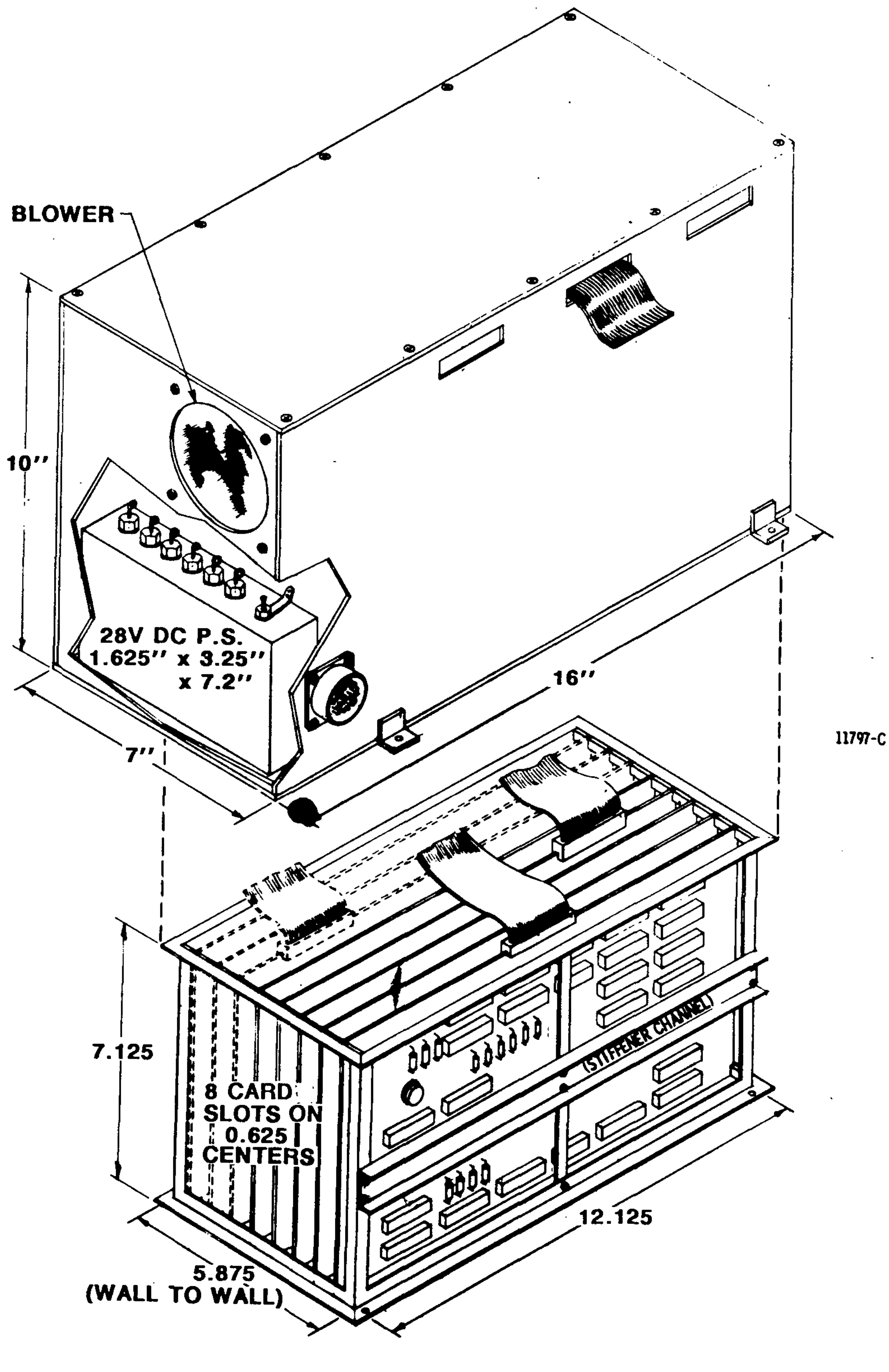


Figure 5-21. FDM DMS Enclosure

These replacement (or next generation) cards will be smaller than those now employed and will be more ruggedly designed, plus they will have greater ability to handle the required thermal dissipation required by the increased density of circuit gates per square centimeter of board area.

The appropriate DC voltages (5 and 12 volts) are supplied internally to the DMS by a self-contained DC/DC converter-regulator. This power supply was specified by Dalmo Victor for the VIDS FDM and was assembled and tested by Arnold Magnetics as a MIL-qualified unit. The electrical specifications of the unit are indicated below:

DC/DC Converter - Regulator
Arnold Magnetics Model DV-4139
Input: 24 - 30V DC
Output: +5V @ 20 Amp
+5V @ 1 Amp
$\pm$12V @ 2 Amp
$\mp$12V @ 1 Amp
Regulation: 2%, Half to Full Load
Maximum Case Temperature : 71°C

To provide adequate cooling air for the DMS, a special, thin-profile 28V DC fan is installed in one end of the enclosure to draw outside air over the power supply and through the card cage to exit at the other end of the enclosure (see Figure 5-21). The fan is Model 4124F, manufactured by PAMOTER.

5.3.4. Adapter Unit No. 2. Adapter Unit No. 2 is designed as a temporary interface between the 1553B Bus Controller and the various peripheral devices. In order to control the cost of development, the adapter unit is used to connect to more economical and commercially available peripherals. It is connected to the VIDS DMS by a 1553B bus and to the Control Panel and the Counteraction Device Simulator (printer) by two RS-232 lines. See Figure 5-22.

The adapter unit consists of a single-board computer (SBC), namely an Intel SBC 86/12A, and a separate board for I/O. The I/O board consists of the following:

- 1,024 x 16 bit of read/write memory
- MIL-STD-1553B dual redundant input/output port
- Four programmable serial ports (RS-232, RS-422, etc.)
- One 16-bit parallel input port
- One 16-bit parallel output port
- General purpose controller for multibus input/output access and peripheral direct access to the onboard read/write memory.

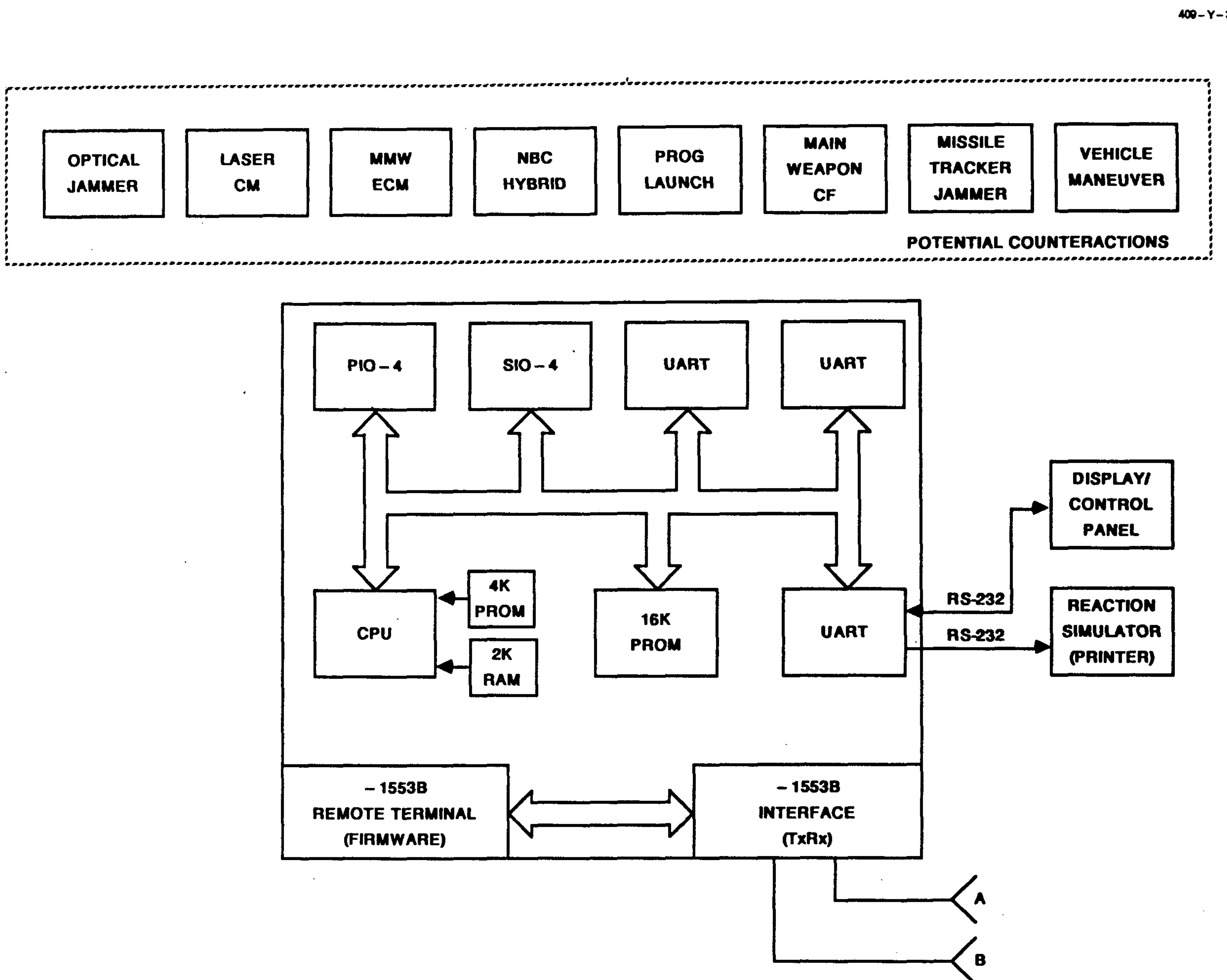


Figure 5-22. Adapter Unit No. 2 Block Diagram

The data within the adapter unit flows in two directions: data that flows from the VIDS DMS box via the 1553B bus can be routed to either the Control Panel or the Counteraction Device Simulator via one of the two RS-232 lines. In the reverse direction, data that flows from the Control Panel by way of the RS-232 line can be passed to the VIDS DMS via the 1553B bus.

There are two types of data that can come from the VIDS DMS: data for the Control Panel or for the Counteraction Device Simulator. The low byte of the second word of the data packet determines which peripheral the data is directed toward. If it is a 4, then the data is decoded and the ASCII representation is sent to the Counteraction Device Simulator to be printed. The data packet from the VIDS DMS to the Counteraction Device Simulator is seven 16-bit words and the format is as follows:

WORD #	DESCRIPTION	
1	DEVICE NUMBER (2)	
2	THREAT ID (1 .. 64) / MESSAGE TYPE (4)	
3	AUTOMATIC COUNTERACTION	(1 .. 14)
4	MANUAL COUNTERACTION	(1 .. 14)
5	AZIMUTH	(0 .. 360)
6	RANGE	(0 .. 25,500)
7	UNDEFINED	

For word numbers 3 and 4, only one of the words is defined as the counteraction message and the other word is set to zero.

The counteraction messages are numbered as follows:

COUNTERACTION #	COUNTERACTION PRINTOUT
1	flare
2	smoke
3	move
4	mwcf ___ degree(s)
5	millimeter wave aerosol
6	missile tracker jammer
7	laser decoy
8	millimeter wave jammer
9	optic jammer
10	cue optical warning
11	hybrid activity
12	surrender
13	cover
14	shoot machine guns ___ degree(s)

If the counteraction number appears in word number 3, the Counteraction Device Simulator will first print "auto" before the counteraction printout. For example, if word 3 contains the number 14, the counteraction recorder would print "auto shoot machine guns 45 degree(s)". The angle is taken from word number 5, the azimuth, which is then converted to the ASCII equivalent and sent serially to the printer.

If the counteraction number appears in word number 4, the Counteraction Device Simulator will first print "--->" before the counteraction printout. For example, if word 3 contains the number 4, the counteraction recorder would print "---> mwcf 345 degree(s)".

The second type of data which comes from the VIDS DMS is data destined for the Control Panel. Any data packets which are not for the Counteraction Device Simulator are considered Control Panel data. The first word of the 7-word data packet is removed and words 2 to 7 are transmitted along with a sync word and a checksum across the RS-232 line to the Control Panel. The data format to the Control Panel appears as follows:

FFFF	-	Sync Word
Word #2	-	Data from VIDS DMS
Word #3	-	Data from VIDS DMS
Word #4	-	Data from VIDS DMS
Word #5	-	Data from VIDS DMS
Word #6	-	Data from VIDS DMS
Word #7	-	Data from VIDS DMS
Checksum	-	-1* (FFFF + Word #2 + Word #3 + ... + Word #7)

The high byte is sent serially first before the low byte of the word.

Data that is transmitted from the Control Panel to Adapter Unit No. 2 is in a 5-word data packet along with a sync word and a checksum.

The data format to the adapter unit appears as follows:

FFFF (Hex)	-	Sync Word
Crew Data #1	-	Data from Control Panel
Crew Data #2	-	Data from Control Panel
Crew Data #3	-	Data from Control Panel
Crew Data #4	-	Data from Control Panel
Crew Data #5	-	Data from Control Panel
Checksum	-	-1* (FFFF + Crew Data #1 + Word #2 + ... + Crew Data #5)

The adapter unit removes the sync word and the checksum and then appends a device number before the 5-word data packet and transmits the data across the 1553B bus back to the VIDS DMS.

5.3.5. Display/Control Panel (Overall Description). The contract requires a means of displaying threat information to the crew for alerting and counteraction recommendation. Dalmo Victor designed, fabricated, and tested a flat panel, thin-film transistor display with a touch-sensitive screen for designation of threat. Several pushbuttons were incorporated for activation or enabling of specific counteractions and mode selections. We also designed and coded a synthetic speech generator within the display/control panel even though the contract offered to provide for interfacing with a Government-furnished equipment (GFE) voice unit.

Approximately 512 English words have been coded into phonemes and burned into PROM for use in the display. Also, a selection of ten threat symbols have been designed and programmed into PROM plus priority designators and soft keys for counteraction selection via the screen rather than via the hardwired pushbuttons.

A photograph of the Display/Control Panel developed for the VIDS FDM demonstration is illustrated in Figure 5-23. This Control Panel measures 3-1/2 inches deep, 8 inches high and 12-1/4 inches wide (not including connectors). The actual display area measures 3.5 inches x 4.75 inches. The importance of a small, flat panel display was forecast in our Phase I study and emphasized in subsequent discussions with users. This is illustrated in Figure 5-24 which is a direct representation of the Gunner/Commander crew stations in the M1 Main Battle tank.

The crew Display/Control Panel was developed initially for specific requirements of the VIDS DMS FDM. Its design, however, has taken into consideration its possible utilization on other programs where a tactical vehicular application might require similar features and operational characteristics. The following paragraphs describe the functional capabilities of this Control Panel as it is used on the VIDS FDM. Because the panel includes a completely self-contained single-board computer, it is expected to be sufficiently flexible to provide adaptability to several other applications of this nature.

The primary functions are to provide a compact means of:

- Alerting the crew with synthetic voice warnings
- Displaying the type and relative position of the threat in a visual (plain view) relation to own vehicle
- Providing means for positive selection/control of countermeasure/targeting/C^2 modes of operation of the crew Control/Display Panel itself.

The method of providing these functions is described in the following subsections for each primary function.

5.3.5.1. Synthetic voice generator. The first function of the Display Control Panel is provided by the utilization of a speech processor chip and audio amplifier on Board No. 4 of the assembly. Programming of the chip is self-contained at the allophone level, and words/phrases are accumulated in 4K bytes of electrically programmable read only memory (EPROM) on an adjacent microcontroller chip. These phrases are called out of local memory

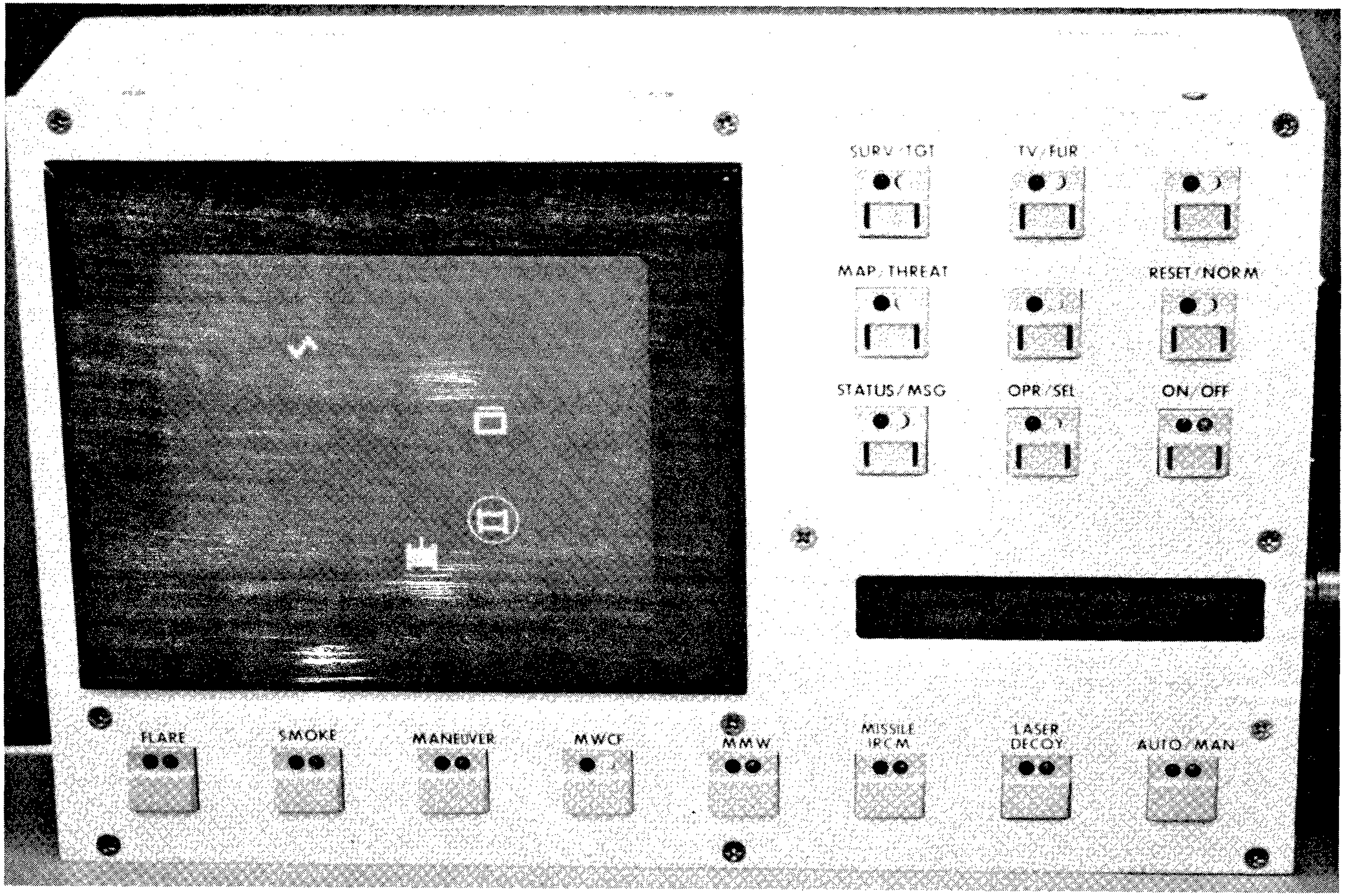


Figure 5-23. Display/Control Panel

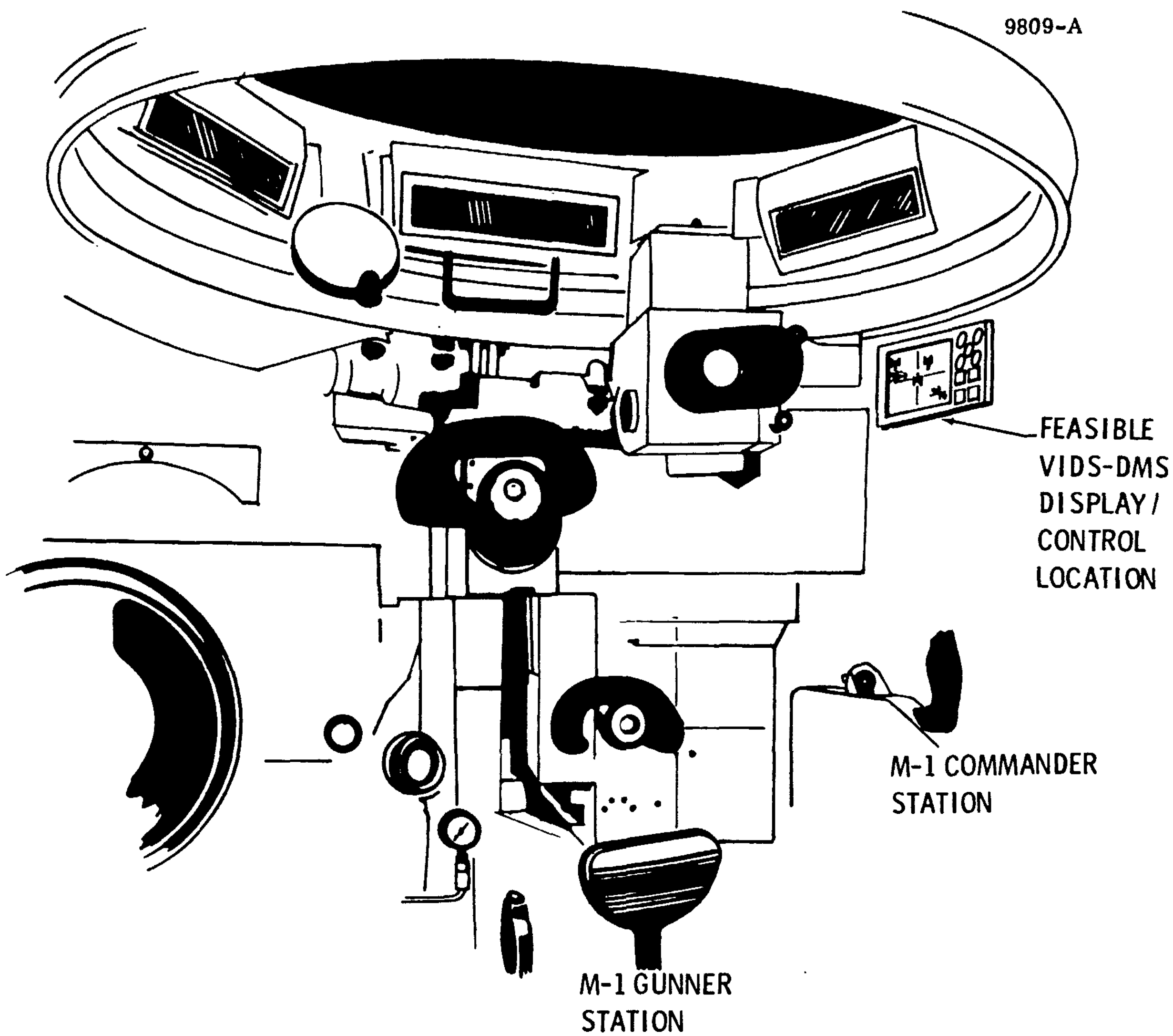


Figure 5-24. Candidate Location for Display in M1

by the DMS via the SBC of the Control Panel. The output of the audio amplifier can drive a speaker or the vehicle intercom net. Details of the Voice Generator Operator follow.

The speech processor selected for the synthetic audio alert is the SPO-256-AL2. This chip uses linear predictive coding, which is more memory-efficient than digitized voice. We have written a program that allows a single microcontroller chip, the Intel 8751, to manage the generation of our entire vocabulary of 512 individual English words by storing them in memory (PROM). The 8751 converts the word number it receives from the 8086 SBC (CCA #3) into the corresponding allophone numbers. The speech chip then converts the allophone numbers into audible voice sounds.

Allophones are basic elements of phonetics which are used to form spoken words. An example of allophones are taken from the word "scout" as in scout helicopter, which is used in our VIDS demonstration. The allophone "SS" (037H) represents the first sound of the word. This is followed in turn by:

A 30-ms Pause (01H), "KK" (02AH), "AW" (020H)
A 30-ms Pause (01H), "TT" (011H)
A 50-ms Pause between words (02H) and STOP (00H).

By running a string of these allophones together, we form a word and record it in PROM for rapid callup. A phrase is enunciated by running a string of words together with reasonable pauses between words.

Overview of Speech Software

Gaining an audio output is a 3-step process. First, the whole phrase is quickly stored in memory. Second, the first word is converted into allophone numbers. Third, these allophone numbers are concatenated to the speech chip to form an audible word. This process is repeated until the whole phrase is enunciated. Each of these steps in discussed in more detail below.

Step #1 - Storing the Phrase:

The 8086 sends a phrase message to the 8651. Since the words in this phrase can not be enunciated as quickly as they arrive, they are saved in a buffer. Each word is identified by its word number which refers to an address space in hexidecimal. This 16-bit number is received in two 8-bit bytes: low byte first, then high byte.

Step #2 - Converting Word Numbers into Allophone Numbers:

The starting address for the allophone number which makes up each word are located at even 10 Hex address. The 8751 multiplies the word number it receives by 10 Hex. This product is the starting address in memory for the string of allophone numbers of the word.

The 8751 sequentially feeds allophone numbers to the speech chip, waiting until the last allophone is enunciated before sending the next allophone number. The 8751 concatenates all of the allophone numbers for all of the words until the whole phrase is transmitted.

Step #3 - Converting Allophone Numbers into Audible Words:

The speech chip converts each allophone number into an allophone sound. Run together, these sounds are heard as words. The 8751 continues to feed the speech chip allophones until the whole phrase is enunicated.

Details of Software for Synthetic Voice Module

The module receives data essentially from a weapon system embedded processor via a serial line to the SBC/30 on an RS-232-C cable. The Intel CPU (8086) checks the integrity of the incoming data, identifies the message as a voice command, processes the data in the message, and, outputting to an I/O port, sends the data to an offboard 8751 voice chip.

The actual processing of the data in the voice command message that is done by the 8086 basically consists of storing the message in a buffer before it can be sent to the 8751. After the CPU (8086) confirms that the data received is valid, it is ready to process the data that is in the form of word numbers. Until no more word numbers remain, or until the end of the voice command message is reached, each word number is placed into a buffer with a comma-word number inserted between word numbers. A comma-word number is a number that corresponds to the sound of a short pause (100 msec of silence). When there are no more word numbers, and the end of the message hasn't been reached, commas are put into the remaining buffer message block. Then, at the end of each message block, a longer pause (1 second of silence) is placed at the end of the message block in the buffer.

Voice Command input data from the DMS contains the following:

Word Numbers - Each word to be "spoken" has a corresponding hexadecimal number. This number is a 16-bit integer greater than 0. If the word number is zero, then it is interpreted as a flag to signal that there are no more words in the message.

Start Word - To indicate the beginning of a message, a 16-bit integer is input = OFFFFH.

Checksum - At the end of each message, the checksum is used to verify the integrity of the data transmitted. The checksum is a 16-bit integer which equals the negative value of the sum of all the 16-bit data words in the entire message, including the start words.

The start word and the checksum are generated by software in an adapter unit which converts M-S-1553B messages to RS-232-C format.

Each Voice Message to the SBC/30 has the following format and order of transmission:

Word 1	Start Word	= OFFFFH
Word 2	Type Number	= Prioritized Track File ID Number; 1 (Audio)
Word 3	Word Number	= Type of threat (i.e, Missile, Tank, etc.)
Word 4	Word Number	= O'clock position of the threat (1..12)
Word 5	Word Number	= Reaction 1
Word 6	Word Number	= Reaction 2 (i.e., Cover, Smoke, Jam)
Word 7	Word Number	= Reaction 3
Word 8	Checksum	= Negate the Sum

The device to handle the data input is an Intel 8251A, Programmable Communications Interface Chip. This device is programmed to receive at 1200 baud, asynchronous transmission, parity disabled. For every 8 bits of data received, the 8251A indirectly interrupts the CPU (via an interrupt controller, 8259) to service the device and incoming data.

Each Voice Message is output to the 8751, Voice Controller Chip. It is sent in a series of bytes in the following format and order of transmission:

Bytes 1, 2	Type of threat	(Low Byte, High Byte)
Bytes 3, 4	Comma	(Low Byte, High Byte)
Bytes 5, 6	Position	(Low Byte, High Byte)
Bytes 6, 8	Comma	(Low Byte, High Byte)
Bytes 9, 10	O'Clock	(Low Byte, High Byte)
Bytes 11, 12	Comma	(Low Byte, High Byte)
Bytes 13, 14	Reaction 1	(Low Byte, High Byte)
Bytes 15, 16	Comma	(Low Byte, High Byte)
Bytes 17, 18	Reaction	(Low Byte, High Byte)
Bytes 19, 20	Comma	(Low Byte, High Byte)
Bytes 21, 22	Reaction 3	(Low Byte, High Byte)
Bytes 23, 24	Pause	(Low Byte, High Byte)

Position, o'clock, and any reaction can be replaced with a comma.

The 8751 is sent a byte whenever it is ready to receive one. It interrupts the 8086 CPU (via an interrupt controller, 8259) to request a byte. Each byte sent by the 8086 to the 8751 is written to a latch and is strobed in by executing a successive write to an 8651 interrupt line. Thus, two back-to-back writes send one byte to the 8651.

The voice module has two limitations:

- If a message received from the weapon system is incorrectly formatted, then it will be ignored and its retransmission not requested. The adapter unit stores several messages in a buffer whose input and output rates vary. Thus, the adapter cannot determine which message to retransmit.

- If a voice message sent by the weapon system contains any undefined word numbers, then, since no current list of valid word numbers is kept, this undefined word number is passed on to the 8751. The resulting audio output is stuttering and incoherent, and will terminate either by itself or by pressing a Voice Reset Switch.

5.3.5.2. Flat panel display. The second function of the Display/Control Panel is the display of symbols and graphic imagery for use in:

- Threat warning and location
- Map presentations for C^2
- Message presentation for C^2
- Maintenance and training programs.

To minimize the depth required for the display, the technology utilized is that of thin-film transistor (or electroluminescence) which provides a black on orange (or orange on black) image. The image area is 3.5 inches by 4.75 inches which is sufficient for viewing at extremely close range. ($\approx$ 14 inches on the M1). See Figure 5-25.

The flat panel display is driven by row and column electrode drivers built on to the display PWB. A Driver board is attached to the Display Panel which locates the characters or symbols on the display. The Driver board is controlled by the internal SBC or, as an option, could be operated from an external source. The specific programming of the display is performed externally to the unit and can be fed to the Control Panel via RS-232 (or MIL-STD-1553B in the future). In the VIDS FDM configuration, the symbols or graphics are called up from messages arriving from the DMS via MIL-STD-1553B (to Adapter Unit No. 2) and via RS-232 to the display CPU (SBC).

An additional display area is provided on the Control Panel which is completely independent of the thin-film transistor electroluminescent (TFT-EL) panel. This is a message cueing display comprised of segmented light-emitting diodes (LEDs) that can exhibit up to 16 alphanumeric characters. This is intended for use as a cue to the operator alerting him visually to a need for subsequent action on the availability of additional information without using space on the main screen. Examples might include:

- "MSG WAITS" — Indicating that if the commander pushes the "message" switch, a string of words or symbols will be printed on the screen

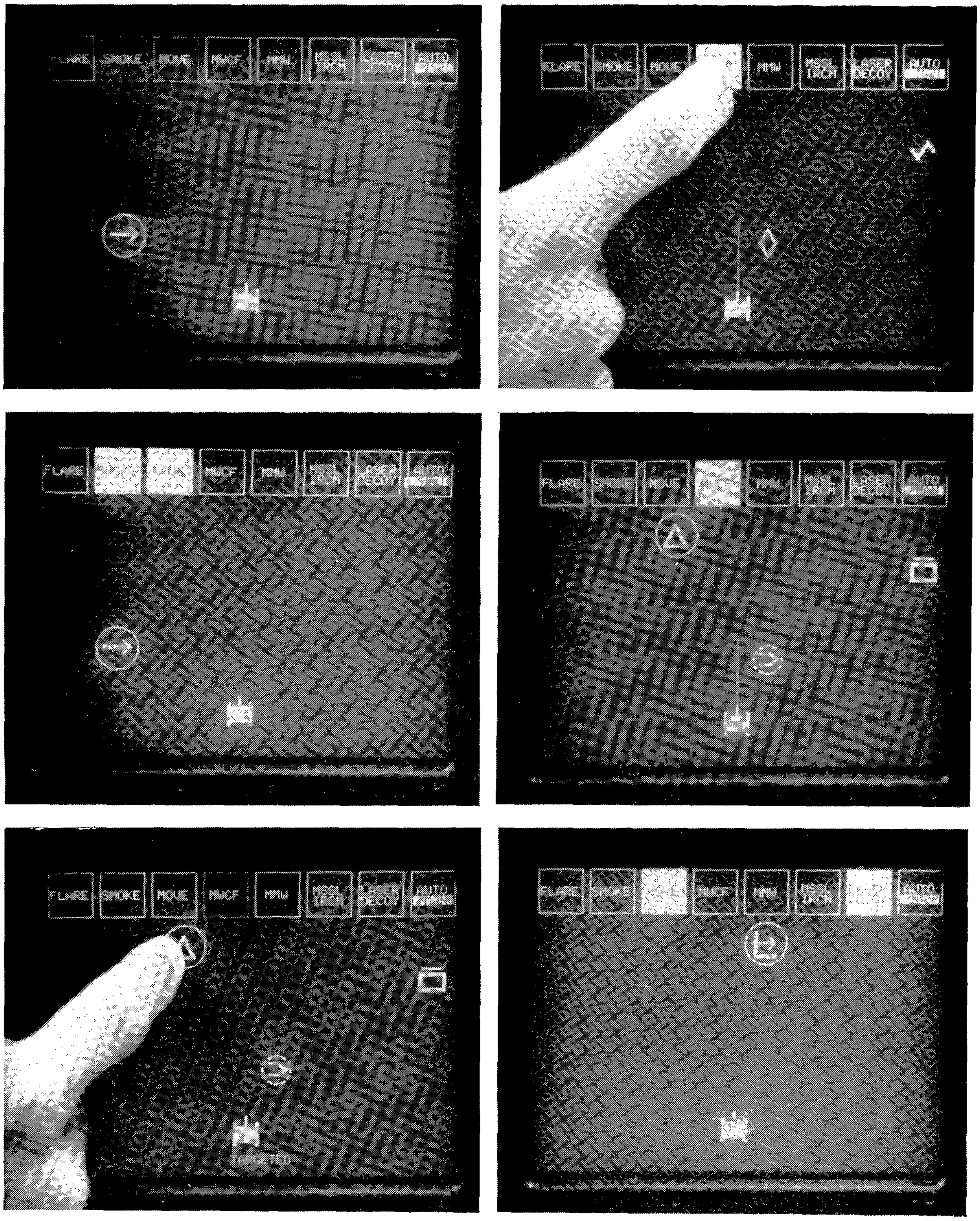


Figure 5-25. Flat Panel Display

- "CALL PL" Indicating a need to communicate
- "CHECK MAP" Indicating a change of some sort on topographical map
- "MAINTENANCE" Indicating the need to put the panel in the maintenance mode for instructions regarding some field repair or correction called for.

The rationale for the methodology of the presentation of symbols on the flat panel display is described in the following paragraphs.

A table of the symbols of emitters (for example, a BMP, is represented by the Optics Detection Symbol, a ◊ or a Δ) has been defined and is included in the VIDS Threat ID/Reaction Chart or Table 5-4. The rules for symbolic display are as follows:

- There will be a maximum of two symbols drawn to display a particular threat, instead of three of four symbols.
- When there are two symbols, they will be alternately superimposed upon one another to produce a "mippling" effect.
- A circle is placed around the symbol by the specific threat designated to be the target for counteraction.
- Because of sensor limitations, range will remain undefined. Instead of arbitrarily assigning an optimum, constant range value, symbols will be located according to relative lethality with highest (1) closest and lowest (5) furthest away.

A table in firmware defines all the combinations of different types of symbols in accordance with the data shown in Table 5-4. The software table is indexed by the threat type number from the DMS which in turn fetches the address of the symbol(s) to be displayed.

When a threat is detected, symbols will appear on the screen at the given azimuth relative to our own vehicle. Also, if the multiple sensors detect signals from the same angle (for example, laser range finder and optics), then both symbols will aternately blink on and off, over each other, producing a "mippling" effect. Furthermore, if the DMS detects a missile, and it has just been launched from, for example, a BMP, then a missile symbol will be "mippled" with a BMP symbol. To indicate the launch of the missile, the symbol of the missile will be drawn with a dotted circle surrounding it.

Table 5-4. VIDS DMS Platform Identification and Reaction Analysis

	A	B		C_1	D_1	E_1		C_2	D_2	E_2		C_3	D_3	E_3
	PLATFORM	POSSIBLE OBSERVABLES (EMITTERS)		DETECTION SENSOR 1	PANEL SYMBOL	REACTION		DETECTION SENSOR 1&2	PANEL SYMBOL	REACTION		DETECTION SENSOR 1 AND 3	PANEL SYMBOL	REACTION
1	MBT	LASER RF OPTICS LARGE SIZE	1 (3)	LASER RF		MWCF	16 (3)	LASER RF OW (LARGE)		MWCF OJ	31			
2	BMP-ATGM	OPTICS LARGE SIZE MISSILE PLUME	2 (5)	OW (LARGE)		OJ	17 (1)	OW (LARGE) PMD		OJ MWCF MTJ	32			
3	RPG	PLUME	(3)	PMD		M SHOOT	18				33			
4	SAGGER PORTABLE ATGM	OPTICS SMALL SIZE PLUME	4 (5)	OW (SMALL)		OJ	19 (1)	OW (SMALL) PMD		OJ MWCF MTJ	34			
5	ATTACK HELO #1	A LASER RF/ LASER DESIG OPTICAL SIGHT PLUME	5 (2)	NIS + LASER DESIG		LD C MWCF	20 (1)	NIS PLUS LASER DESIG PMD		LD C MWCF	35 (1)	NIS PLUS OW (LARGE) PLUS PMD		OJ MTJ C
6	ATTACK HELO #2	A MMW PLUME	6 (5)	NIS		C	21 (2)	NIS PLUS PMD		FLARE C	36 (1)	NIS PLUS MMW_2 PLUS PMD		C MMA
7	SCOUT HELO	A OPTICS	7 (5)	NIS		C	22 (3)	NIS PLUS OW (LARGE)		OJ C	37			
8	LASER BEAM-RIDER (GROUND)	LASER BEAM OPTICS PLUME	8 (3)	LASER DET CULS OW		S OJ M	23 (2)	LASER DET OW (SMALL)		MANEUVER OJ SMOKE	38 (1)	LASER BEAM-RIDER PMD		OJ SHOOT MG
9	LASER DE-SIGNATOR (GROUND)	LASER BEAM OPTICS PLUME	9 (3)	LASER		LD M	24 (2)	LASER DET OW (SMALL)		OJ, LD, MWCF M	39 (1)	LASER DESIGNA-TOR PMD		LD M
10	MILLIMETER WAVE (GROUND)	MILLIMETER WAVE PULSES	10 (5)	MMW_1		MWCF, MMJ	25				40			
11	MILLIMETER WAVE ATTACK AIRCRAFT	MILLIMETER WAVE PULSES	11 (4)	MMW_2		MMJ C	26				41			
12	MILLIMETER ATTACK HELO #1	MILLIMETER WAVE PULSES	12 (4)	MMW_2 NIS		MMA C	27				42			
13	NBC	NBC	13 (1)	NUC	NUC	HYBRID	28 (1)	CHEM	GAS	HYBRID	43			
			14				29				44			
			15				30				45			

○ = LETHALITY INDEX

1 = HI 5 = LOW

OJ = OPTICAL JAMMING MTJ = MISSILE TRACKER JAMMER MMJ = MILLIMETER WAVE JAMMER MMA = MMW AEROSOL

C = COVER M = MANEUVER MWCF = MAIN WEAPON COUNTER FIRE LD = LASER DECOY

The symbol size has been selected so that they enclose at least four and as many as nine contact points of the touch-sensitive screen. Similarly, the EL display area has been subdivided into 66 discrete location for the symbols to be displayed. The availability of specific radials (azimuth) thus depends on the ratio of the symbol size to screen size. The values assigned to ranges and radials (azimuth) thus depends on the ratio of the symbol size to screen size. The values assigned to ranges and radials are as follows:

- There are four levels of lethality for which a symbol can be displayed: #1, #2, #3, and #4/5 where lethality #1 is the most lethal and lethality #3 is less lethal. The approximate radii are:

  - Lethality #1 has a radii of 18 mm
  - Lethality #2 has a radii of 3 mm
  - Lethality #3 has a radii of 54
  - Lethality #4/5 has a radii of 70 mm

  These given radii are the actual distances on the screen from the symbol representing the VIDS vehicle.

- For each different lethality, there will be a different number of radial sectors. The sectors will be of varying width at various ranges. These radials are defined as follows:

  - For Lethality #1

    Radial #1 = 0-14 degrees and 346-360 degrees
    Radial #2 = 15-44 degrees and 316-345 degrees
    Radial #3 = 45-74 degrees and 286-315 degrees
    Radial #4 = 75-105 degrees and 255-285 degrees

  - For Lethality #2

    Radial #1 = 0-7 degrees and 353-360 degrees
    Radial #2 = 8-22 degrees and 338-352 degrees
    Radial #3 = 23-37 degrees and 323-337 degrees
    Radial #4 = 38-52 degrees and 30-322 degrees
    Radial #5 = 53-67 degrees and 293-307 degrees
    Radial #6 = 68-82 degrees and 278-292 degrees
    Radial #7 = 83-105 degrees and 255-277 degrees

  - For Lethality #3

    Radial #1 = 0-4 degrees and 356-306
    Radial #2 = 5-14 degrees and 346-355 degrees
    Radial #3 = 15-24 degrees and 336-345 degrees
    Radial #4 = 25-34 degrees and 326-335 degrees
    Radial #5 = 35-44 degrees and 316-325 degrees

Radial #6 = 45-54 degrees and 306-315 degrees
Radial #7 = 55-64 degrees and 296-305 degrees
Radial #8 = 65-74 degrees and 286-295 degrees
Radial #9 = 75-84 degrees and 276-285 degrees
Radial #10 = 85-105 degrees and 255-275 degrees

- For Lethality #4 and #5

Radial #1 = 0-3 degrees and 356-360 degrees
Radial #2 = 4-11 degrees and 349-358 degrees
Radial #3 = 12-18 degrees and 349-358 degrees
Radial #4 = 19-26 degrees and 334-341 degrees
Radial #5 = 27-33 degrees and 327-333 degrees
Radial #6 = 34-41 degrees and 319-326 degrees
Radial #7 = 42-48 degrees and 312-381 degrees
Radial #8 = 49-56 degrees and 304-311 degrees
Radial #9 = 57-63 degrees and 297-303 degrees
Radial #10 = 64-71 degrees and 289-296 degrees
Radial #11 = 72-78 degrees and 282-288 degrees
Radial #12 = 79-86 degrees and 274-281 degrees
Radial #12 = 87-105 degrees and 255-273 degrees

- Figure 5-26 illustrates the display area and three representative symbols. Also shown is the actual map of the discrete locations available to these symbols relative to each other on the display.

When the threats move relative to the VIDS vehicle, their corresponding symbols will also move on the display. These symbols will appear to move by erasing them from their previous locations and redrawing them at their new locations. The fact that a threat is moving will be indicated in the content of the data sent from the DMS.

The following paragraphs briefly describe the **software modules** that produce the symbols on the flat panel display screen. When data received from the DMS is identified as a symbology message, all its corresponding information is then recognized and stored. Next, the message is determined to be one of three types:

- A command to draw a new symbol
- A command to move an already existing symbol to a new location
- A command to erase a symbol altogether from the screen.

For each of these three types of situations, a different line of action is taken.

The first type of command is to draw a new symbol on the screen. First, the threat type that was given by the DMS is looked up in a Type Table. Corresponding to that threat type is a character code number. This character code corresponds to an address in memory. (Similarly, an ASCII

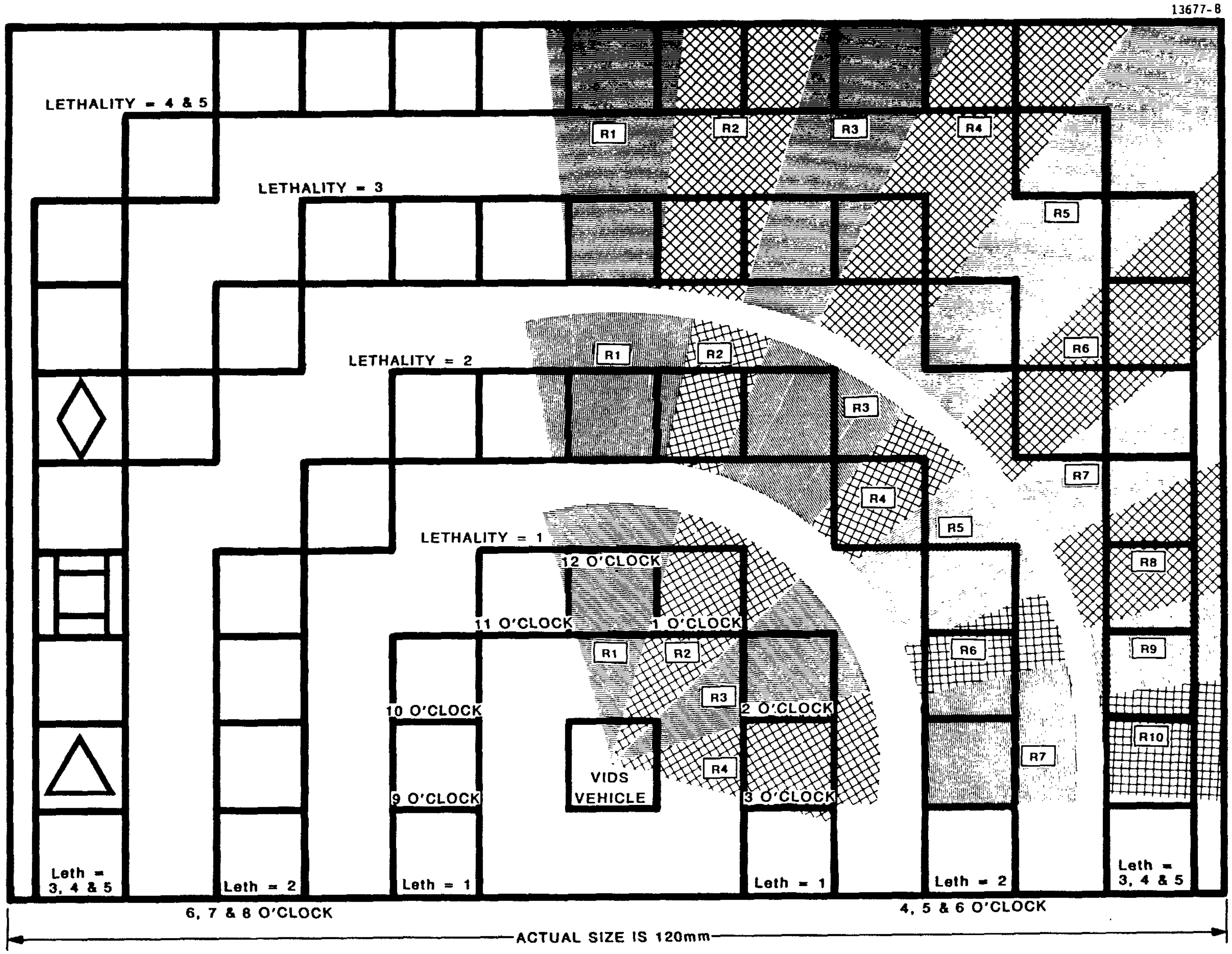


Figure 5-26. Radial Locations on Display

code corresponds to an address that stores this ASCII character to be drawn.) This address in memory is actually a programmable character generator.

Instead of drawing a symbol line by line for each Draw command, preprogrammed symbols are quickly fetched from memory and placed on the screen. This is done by using a character generator. This generator is essentially PROM containing pictures of all possible symbols. The character code for each threat type corresponds to an address in PROM where the dot sequence of the symbol has already been stored.

In addition, for each character code, there corresponds another address where another symbol can be stored. This second address is termed "Page 2". Thus, two symbols can be programmed for each character code and, by switching continuously from "Page 1" to "Page 2", both symbols will alternately be displayed. This results in a "mippled" image.

Once the character code is obtained for a particular threat, a location is determined where the symbol will be displayed on the screen. Given an azimuth reading from the DMS, the threat is assigned at a particular o'clock position (radial). Then, using the lethality index, also given by the DMS, the threat is located at a specific lethality ring from the VIDS tank. At this point, an x and y coordinate are calculated at which the symbol will be drawn.

A simple graphics package built into the screen display enables ASCII characters to be input to its processor and interpreted as graphic commands. To draw a symbol on the screen, the processor is first commanded to move the cursor to the x, y coordinates and then to draw the symbol that corresponds to the character code. Since a symbol is larger then a single text character, it is actually made up of nine consecutive character codes arranged in a square. For simplicity, a symbol is referred to by only one character code.

Once the symbol is drawn on the screen, information about the symbol (x-, y- coordinates, threat ID number, etc.) are stored in a list of all ongoing threats. Recommendations by the DMS are kept updated in another table by priority, where the threat currently designated to take action against is stored at the top of the table. Highly lethal threats are inserted at the top when the DMS or the tank crew designates a threat to be the most lethal. Precedence between the crew recommendation and the DMS recommendation is determined by the mode of operation. In manual mode, the crew's designation of the most lethal threat overrides the DMS designation. Conversely, in auto mode, the DMS has full control in deciding the order in which it will react to the threats.

The second type of command is to move an already existing threat to a new location on the screen. First, the threat ID number given by the DMS is searched for in the list of ongoing threats. When the threat is found in the list, its x-y coordinates, which had been previously stored during the

Draw Command, are fetched. At this location, blank characters are drawn to erase the symbol from the screen. The threat is also removed from the list of ongoing threats. For the remainder of the Move command, the process is executed as if it were a Draw command. A character code and x-y coordinates are determined and the symbol is redrawn on the screen at a new location. Updated information is stored in the list, and the threat's lethality status is also updated.

The third type of command is to erase a symbol altogether from the display screen. This command is similar to the Move command. It searches the list for the threat and uses its x-y coordinates to find and erase the symbol from the screen. All information pertaining to the threat is then erased form the list of ongoing threats and from the table that prioritizes the threat's lethality.

5.3.5.3. Single board computer and control interface boards. The Display/Control Panel includes an SBC. For economy and risk reduction, the commercial model Intel iSBC 86/30 was selected. This board is multibus-compatible and contains the 16-bit 8086-2 microprocessor operating at 8 MHz clock rate. It has 128K bytes of dynamic RAM, up to 64K bytes of EPROM, a serial communications port providing an RS-232-C interface, three parallel I/O ports providing 24 individual I/O lines, and two iSBX Bus connectors which provide interface to the control interface board which is described in the following paragraph as a "piggyback" board plugged into the 86/30 SBC. A block diagram of the Interface Adapter board is shown in Figure 5-27.

The Interface Control Board is designed to plug into the single board computer (CCA #3), hence its designation.

It is composed of output and input section. The output section is driven by the SBC. The data flow is output from the SBC via the interface circuits to the following:

- Threat/Response advisory LEDs (yellow)
- Threat/Response return switches/LED clear
- Litronix display
- 8751 Microcomputer controlling "voice".

The input section feeds data to the SBC. The data flow is in the SBC via the interface circuits, from threat/response switches, mode/status switches, and touch panel inputs.

The output control section is composed of buffers from the SBC to the interface section of 16 data lines (U15 and U16), four control lines (U38), and one return control line (U38) from the interface section to the SBC. A 4-section digital delay line (U56) is used to generate the wait states to the SBC, the 8751 Microcomputer interrupts, and the digital display (Litronix) Clear and Write timings.

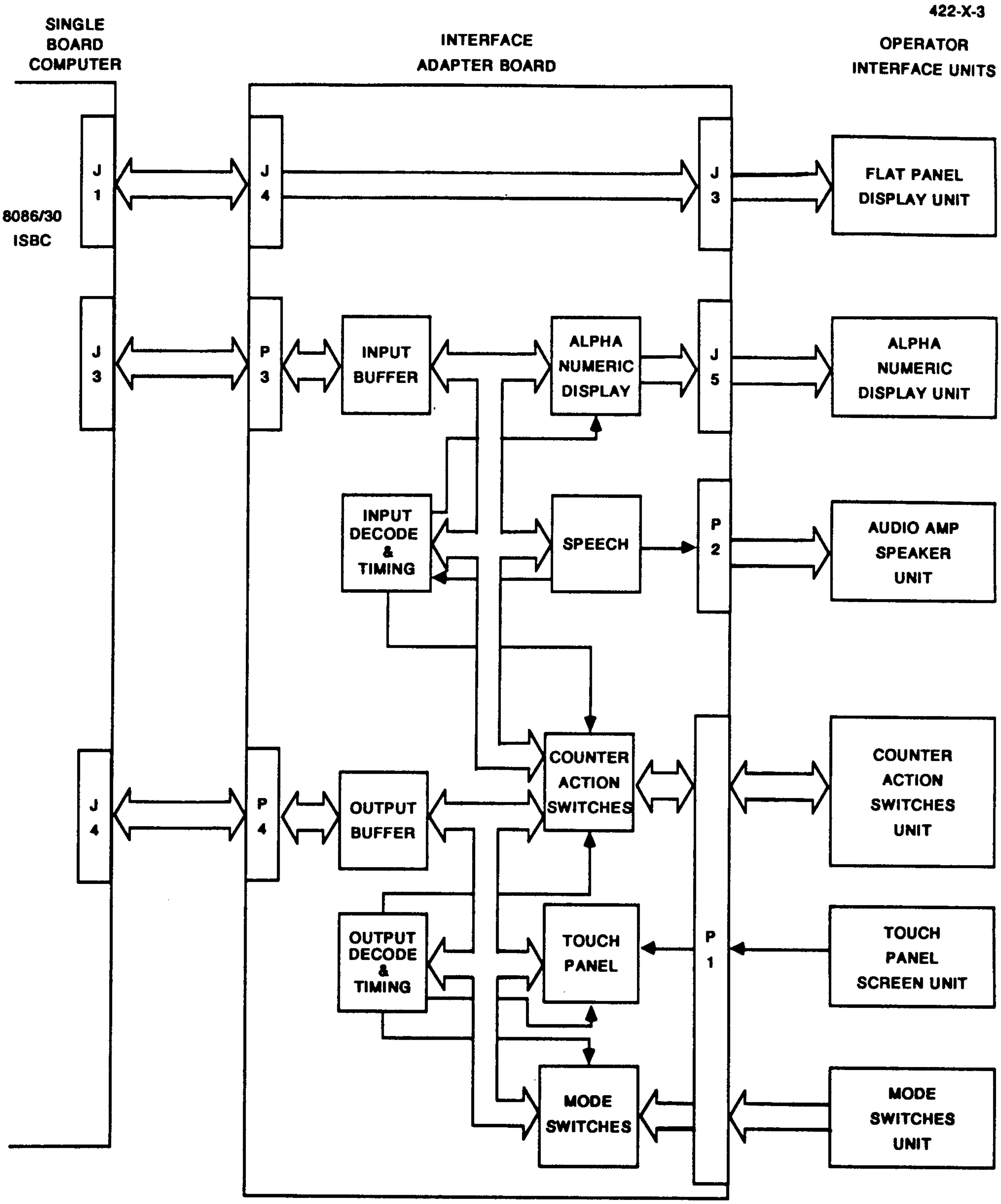


Figure 5-27. Display/Control Panel Block Diagram

All control signals (Enable, Clear, Write, and Interrupt) and the Wait signal that return to the SBC are generated in a programmable array logic (PAL) (U29). The PAL equations are documented elsewhere.

The interface to the threat/response advisory LEDs is a static storage register (U13). This register is set by a specific SBC command/data set and reset by either the Master Reset (MR) or another specific SBC Command/Data Set.

The LED line set is pulsed with a 0.2 microseconds "LOW" every 0.8 milliseconds. This is used to generate a detectable LOW (pulsed so that no light is seen) on the response switches, even though the advisory LED is not lit and is generated in a counter (U25, U24, and U34) and a buffer (U23).

The Litronix display interface has output buffers (U17 and U18) that pass the data from the SBC (d0 through d7) to Port 1 of the 8751. Two control signals from the PAL and a clock of about 4.5 MHz are buffered (U38) to the 8751. Our internal document gives a flow diagram of the interface control program.

The input control circuits consist of a decoder (PAL2)(U28) and a wait state generator (U47). The output of the decoder enables the various data inputs:

- Threat/Response switches
- Status/Mode switches
- Touch Panel data.

The decoder also enables the data transmission to the SBC and puts up the wait generator signal.

The Touch Panel input and the Threat/Response switch input generate interrupt signals to signal the SBC that data is ready. The Status/Mode switch input must be sampled on a periodic basis by the SBC.

5.3.5.4. Touch-sensitive screen and push button controls. The third function of the Display/Control Panel is positive control. This is provided through a combination of a touch-sensitive transparent screen placed over the display and interconnected to the SBC via a serial line, and two sets of positive action single pole double throw (SPDT) illuminated switches located on the front panel. The VIDS utilization of these touch-sensitive circuit activators is best described by the following sequence of actions for a typical engagement while in the manual mode.

1) Vehicle sensors input threat data to the DMS by way of MIL-STD-1553B data bus.

2) The DMS analyzes the multiple sensor inputs, performs the necessary evaluation via look-up tables, and recommends activity.

3) A specific message is sent from the DMS to the display to generate a specific command phrase (string of words) and the synthetic audio gives voice alert.

4) The display shows position of threat(s) and lights the yellow LED on the pushbutton for the indicated counteraction (recommended by DMS computer).

5) The commander puts a finger over the critical threat symbol on the display screen to select the particular threat he wishes to address.

6) The touch panel identifies the selected threat (by x-y locations) and designates it to DMS by way of the onboard CPU and serial data link to/from the DMS.

7) The computer (DMS) now has a registration between its recommended action and the commander's selected target (threat). The computer is ready to take whatever action the commander demands or it will automatically take its own recommended action.

8) The commander presses the indicated pushbutton to initiate appropriate counteraction against a selected threat. (The computers convert the command instructions into the proper interface signals to the specific counteraction circuitry.)

Note: When in the automatic mode, the commander may cancel the computer recommendation and the semiautomatic initiation of the recommended counteraction by pushing the auto/man button to return to the manual mode.

The touch-sensitive panel is a transparent pair of conductive panels with dimpled contact in a 0.25 in. x 0.25 in. array. When an area is pressed with the fingers or pencil eraser, an average of the 2 to 4 contacts are evaluated by measurement of effective resistance along the vertical and horizontal axis. This is converted to relative voltages which are digitized and transmitted as several words (12 bits for "x" value and 12 bits for "y") to the interface board for action by this SBC.

The Touch Panel input is received from the Touch Panel controller in four 10-bit (quasi RS-232) words. The words are sent in a single group, and the first start bit is assumed to start after a long (100 ms) "0" time. The leading edge of the first start bit is detected (U39 and U59) and used to start a sampling clock generator consisting of a divide by 16 clock generated shifter (U22), and a bit clock generator (U11 and U21)("BCLK"). The first start bit is sampled approximately 1.6 μs after the leading edge in the serial-to-parallel converter (U55, U51,, U54, U52 and U53) and each "BCLK" is about 1.6 μs additionally delayed from the leading edge so that at the fortieth bit time (last stop bit) the sample clock is about 66 μs after the leading edge. The "BCLK" strobes each bit (start, synch, synch, 6 data, and stop for each of our words) through the serial-to-parallel converter and is counted in the "40 counter" (U31 and U32). At the fortieth strobe, the "OVER" signal (U58) disables any further inputs, causes an interrupt to be sent to the SBC, and resets/presents all timings. "OVER is removed when the "E3 and "E4" signals generated by decoding the SBC read commands, used to gate data onto the output bus, are enabled. The quasi RS-232 data is transformed into two 14-bit words of "X Pos." (2 synch bits and 12 data bits) and "Y Pos." (same as X pos.) by wiring.

Two types of pushbuttons are installed for positive control. The mode switches (Threat/Map, Surveillance/Targeting, TV/FLIR, ON/OFF, etc.) are latching, alternate-action SPDT illuminated pushbuttons while the counteraction switches *(Flare, Smoke, MWCF, etc.) are momentary-action SPDT illuminated pushbuttons.

The Status/Mode switches are buffered through the interface (U41) with no logical operations done. The SBC samples this input by "reading" the lines.

The Threat/Response switches are each monitored by an edge detector (U73, U74, and U75 or U63, U84, and U85). The input (while the switch is held) is either a level of the "Yellow" LED was lit (response is the suggested action) or a pulse (duty cycle 1:4000) if the "Yellow" LED was unlit (response is not a suggested action). The detected edge latches the buffer of switch status (U71), disables further unsuggested action *until data is read/reset,), drives the "Red" LED on *U72), sends an interrupt to the SBC (58 and U35) and when the results are to be read, buffers the data onto the output bus (U61).

5.3.5.5. Power Supply Requirements. The display and control box is supplied with power through a single cable. The part number for the female connector on the cable is M8151106ED03S1. The part number for the male connector on the display and control box is M81511/01ED03P1.

The power requirements for the display and control box are as follows;

SUPPLY VOLTAGE (Volts)	CURRENT (Amps)	POWER (Watts)
+15 $\pm$ 5%	0.650	9.7
+12 $\mp$ 5%	0.025	0.3
+ 5 $\mp$ 5%	11.760	58.8
-12 $\mp$ 5%	0.025	0.3
	Total Power Requirement	69.1

*Subsequent to initial demonstration of the display, the counteraction switch functions have been transferred to "soft keys" that show on the display and are actuated by means of the touch panel and new firmware instead of the hardwired pushbutton switches.

5.3.5.6. Summary and specifications. The Display/Control Panel enclosure has been carefully designed to occupy as little volume as possible, consistent with the needs for a rugged, mobile, small-screen display while utilizing existing (commercially available) subassemblies. To accomplish this, the Control Panel is assembled with only four circuit boards averaging 3/4-inch in thickness plus spacing between boards. Switches are of a new waterproof type that are installed on printed wiring boards at the second level of assembly. The touch sensitive transparent signal screen is attached at the front surface of the display, its interconnect wires enclosed by a thin bezel assembly on the front surface of the control panel. Neither touch panel nor switch component extend more than 1/4-inch above the front surface of the panel.

Following are a few pertinent specifications for the crew Display/Control Panel:

Synthetic Voice

- Allophones created by linear predictive coding (LPC) techniques and stored in ROM onboard speech processor chip.
- Pulsewidth modulation converts D to A which is amplified and filtered for external speaker on vehicle intercom.
- Present VIDS vocabulary consists of 250 words concatenated into 25 phrases stored in μV PROM. Words and phrases may be varied to suit particular applications. The standard vocabulary is available for inspection on request.

Visual Display

• Primary display screen	Thin film transistor electroluminescence
• Area of display	4-3/4 inches wide by 3-1/2 inches high
• Resolution of Graphic Display	0.015 inches or 66 lines per inch
• Display Color	Orange-Yellow (5859 °A)
• Brightness	25 foot lamberts
• Frame frequency	60 Hz

Positive Control

• 6 mode switches	SPDT alternate action
• 3 Function switches	SPDT alternate action
• 8 counteraction switches	SPDT momentary contact*
• illumination	Yellow LED "Ready"; Red LED "Activated"
• Touch-Sensitive Screen	
Thickness	0.125 inches; transparent
Resolution	0.25 inch, (10 M sec conversion time)

Auxiliary Display

• Cue	16-character segmented LEDs
• Size	1/2 inch x 4 inches display area
• Access	Serial Link (16 bit) from internal CPU (SBC)

Size

- 12-3/8 inches wide
- 8 inches high*
- 3-1/2 inches deep

Weight

- 5 pounds

Power Requirements

- +5 volts at 5 amps
- +15 volts at 0.8 amps

*Hardwired pushbutton counteraction control switches have been replaced with "soft keys" on screen. Panel height may be reduced to 6 1/2" by removal of the bottom row of pushbuttons.

5.3.6. Counteraction Device Simulator. Obviously, in the laboratory, it is not practical to hook up a flare dispenser or a 120-mm tube and turret to see whether the selected counteraction takes place. As an alternative, in place of the counteraction device we have connected a reaction recorder. The reaction recorder prints a brief description of what counteraction has been initiated by the DMS.

The reaction recorder is an Alphacom 42 thermal printer which accepts 4 1/2-inch thermal paper. It uses an Alphacom 1842 Plug-In Interface Module for RS-232-Serial. The 1842 Interface Module is set for a baud rate of 1200, with incoming data on RD-pin 3 and busy signal line on DTR-pin 20.

Typically, the description of what counteraction has taken place begins with either "--->" or "auto". The "--->" indicates that the crew has selected a counteraction. The "auto" indicates that the VIDS DMS has automatically selected a counteraction. After the "--->" or the "auto", the description of the counteraction follows immediately. There are 14 possible counteractions:

COUNTERACTION #	COUNTERACTION PRINTOUT
1	flare
2	smoke
3	move
4	mwcf ___ degree(s)
5	millimeter wave aerosol
6	missile tracker jammer
7	laser decoy
8	millimeter wave jammer
9	optical jammer
10	cue optical warning
11	hybrid activity
12	surrender
13	cover
14	shoot machine guns ___ degree(s)

The counteraction printout with "___" needs the azimuth angle to be filled in. Adapter Unit No. 2 extracts the azimuth from the data packet sent by the VIDS DMS. Adapter Unit No. 2 then takes the azimuth and forms the ASCII representation of the azimuth and stores the data in the counteraction printout message. Finally, the counteraction printout message is sent to the 1842 RS-232 interface module for printout.

For example, a printout may look as follows:

```
---> mwcf 45 degree(s)
auto optical jammer
```

5.4. Ada Run Time Operating System (ARTOS)

It is necessary to provide software that supports the operation of the application software package in Run Time on the embedded microcomputer. An illustration of the relationship between the shell of the Operating System, the computer, and the applications that run on it is shown in Figure 5-28.

The ARTOS was designed and coded by Bell Technical Operations (presently known as Systems and Software Engineering, Department of Dalmo Victor), in Tucson, Arizona. The ARTOS was developed on a Pixel 100, 68000 host computer, using the Telesoft 1.5 Ada compiler. The ARTOS Top Level Design Specification (provided to TACOM in a separate document) describes a complete operating system with many utilities and system services.

Because of the semi-efficient Embedded System Kit which we were using to develop the ARTOS, several functions became extremely difficult and time consuming to code. In the interest of economy, we elected to code and test only those functions of the ARTOS that were essential to the demonstration of the FDM. The resulting "Skeleton Operating System" is described in Section 5.4.4.

Nearly all of the ARTOS was written in Ada with the exception of the interrupt handlers and the terminal I/O drivers, which were written in 68000 assembly.

5.4.1. Overview of ARTOS Design. The ARTOS provides general system functions and procedures for the VIDS DMS operational application software. The following functions are performed by the ARTOS:

- Allocation of the 68000 central processor to individual software processes based on priority
- Management of physical memory among individual software processes
- Low-level communications between the 68000 central processor and the MIL-STD-1553BB Bus Controller
- Management of all data transferred between operational application software and the Bus Controller
- Exception handling
- Interrupt handling
- InterProcess Communication (IPC)
- Real-time clock management.

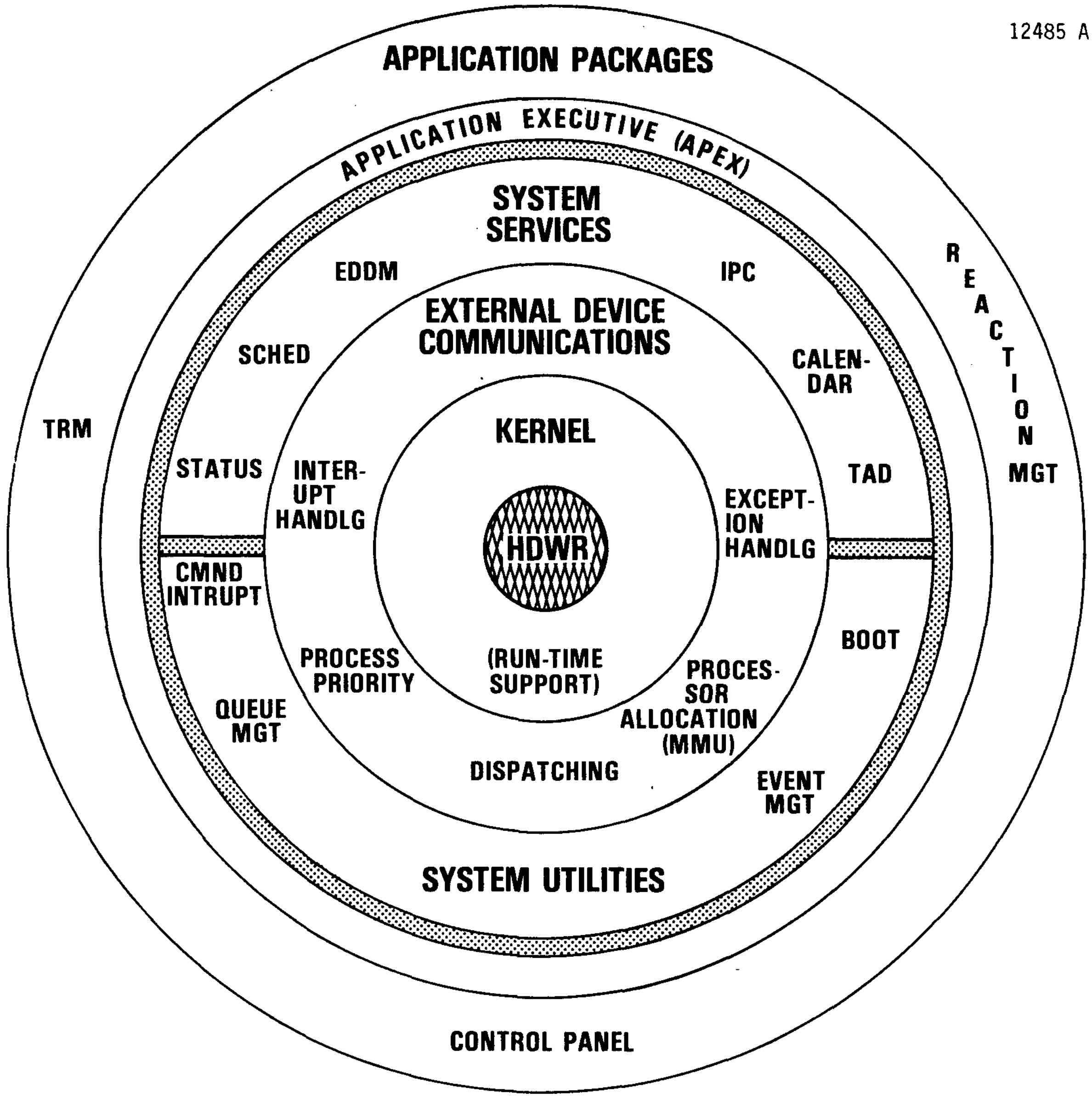


Figure 5-28. ARTOS Shell Supporting Specific Ada Application Packages

The following additional functions are provided to facilitate system development and evaluation:

- System generation services
- Bootstrap loader
- Command Language Interpreter (CLI)
- Trace and audit services.

The run-time subsystem provides service and support for the application programs on the target machine. It consists of 18 packages which are organized in a top-down structure as shown in Figure 5-29.

5.4.2. System Generation. To generate run-time program to download to the VIDS DMS, both the source code for the application programs and the ARTOS must be compiled on the host system (Callan 200) using the Telesoft 1.5 Ada compiler. For example:

```
cd /usr/esk/skeleton  -- Directory where source code is located.
ada sys_serv          -- Start recompiling.
```

Next, the code files are then moved to the directory where the run-time program is to be generated.

```
mv *.c* /usr/esk
```

In the directory where the run-time program is to be generated are the Run-Time Kernal, linker command file, bind file, and other file necessary creating the runtime program.

```
cd /usr/esk
```

If the interrupt handlers are to be modified, they can be found in "rtk.traps.text". Once "rtk.traps.text" is modified, it must be reassembled and relinked.

```
asm +list rtk.traps.text
link ln.68ktest
```

"ln.68ktest.text" is the linker command file which specifies the heap and stack memory areas and includes any user-written drivers. If the heap or the stack memory area are modified, "ln.68ktest.text" must be relinked. (The stack can be set to the highest available memory location and the heap can be set to the first address after the run-time program.)

It is important to check whether the location of "rtk.rminit" has changed after relinking. To verify if the location has changed, check at the address for "rtk.rminit" in the "mp.text" file; if the address equals 712E hex when 5,000 hex is added to it, then the location initial start address for the run-time program must be set to this new value plus 5000 hex.

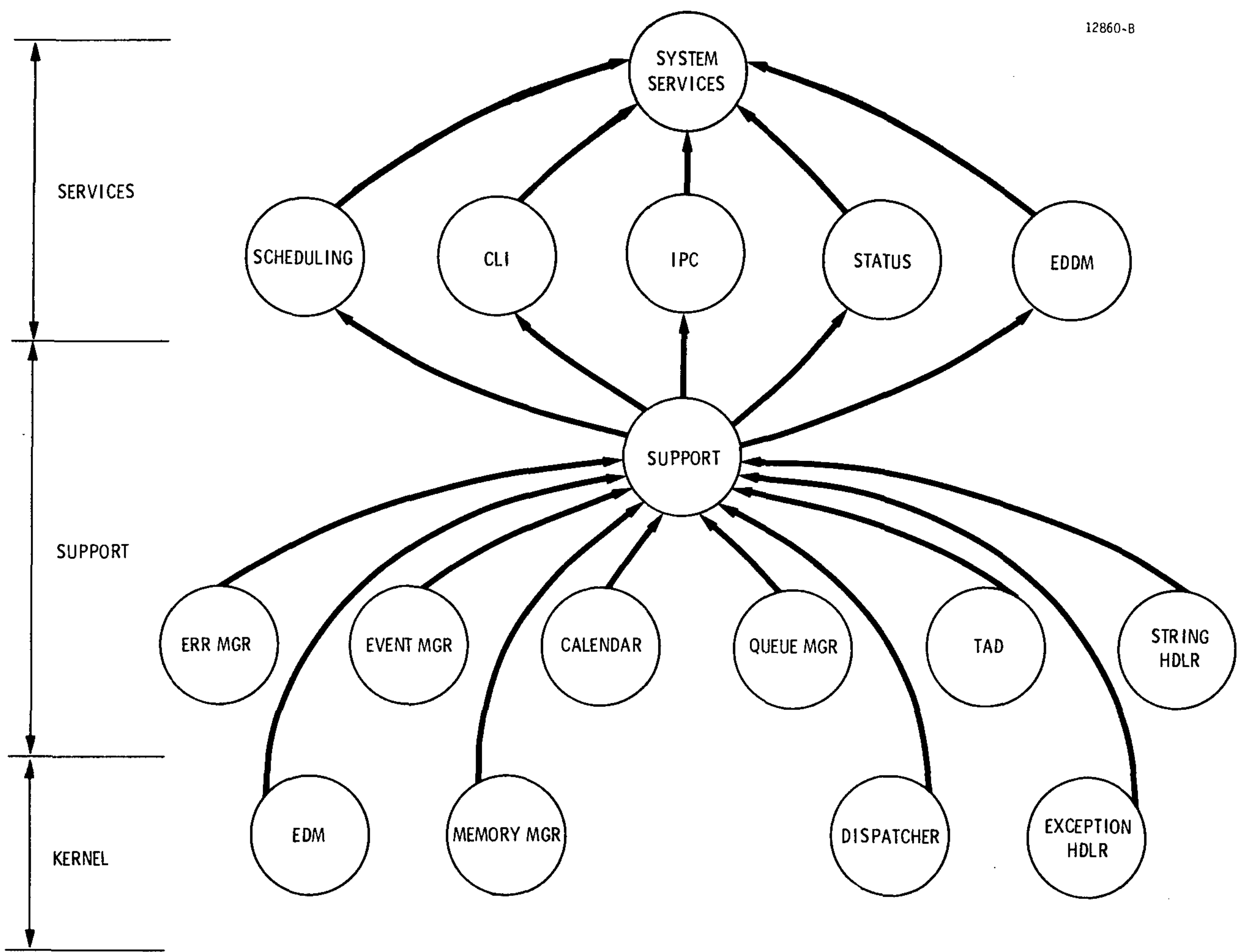


Figure 5-29. ARTOS Package Structure

Next, the bind program must be checked if any code files are to be added or deleted from the final run-time program. If there is a change, recompile the bind file. The bind program binds system components and user-written application code to produce a bind file (or run-time program).

```
emacs bind_xxx.text  -- The "xxx" should be replaced by the real
ada bind_xxx         -- file name.
```

Remove the old bind files and maps, and then run the bind program.

```
rm b.file b.map
run bind+xxx +r b.file b.map
```

The new run-time program is b.file which can be downloaded to the VIDS DMS via the RS-232 port of the host computer.

```
cat b.file > /dev/ttyf
```

For more information on how to generate the run-time program, refer to the Embedded Systems Kit user's manual.

5.4.3. Boot and Program Download. The boot program is burned into EEROMs which reside on the 68000 Single Board Computer. When power is turned on and the reset switch is pressed, the boot program is executed.

The function of the boot program is to load the run-time program (the ARTOS and the VIDS DMS) into 512KB RAM board (which is to be replaced by 512KB EEROM board). When the code is loaded into the 512KB RAM, the boot program will copy the entire 512KB into the 1MB local RAM where the run-time program will execute.

In more detail, the boot program remaps memory so that address 100000 hex through 17FFFE hex are mapped to multibus address 280000 hex through 2FFFFE hex located on the 512KB RAM board. Once memory is remapped, the boot program reads the board's write enable switch (at location 17FFFE hex). If the write enable is a 1, the boot program reads the downloaded run-time program from the UART and then writes into the 512KB RAM (or EEROM) in 64-byte blocks. The boot program assumes that the download is completed two seconds after the last byte of data is read from the UART. When download is completed, the write enable is set to a zero on the 512KB RAM board and then proceeds to copy the entire 512KB of RAM (or EEROM) to the 1M local RAM board. The run-time program is loaded starting at location 5,000 hex on the local RAM board. When it is copied to the local RAM, execution is transferred to location 712E hex on the local RAM board. Before the run-time program can be executed, the boot program must replace the old memory map and bus error vector.

To perform a software reset, press the reset switch. A software reset allows the boot program to recopy the 512KB RAM to the local RAM board. And since the power is not turned off, the run-time program residing in the 512KB RAM is not erased so the run-time program does not need to be re-downloaded from the host computer.

As mentioned earlier, the VIDS DMS run-time program is downloaded via an RS-232 line from the Callan 200 to the VIDS DMS. When the VIDS DMS is initially powered up, control is transferred to the boot program on the VIDS DMS CPU. The boot program waits for the VIDS DMS run-time program to be downloaded. The downloaded program is first stored on the National Semiconductor Multibus 512KB RAM board. Once the run-time program is completed, the boot program copies the run-time program from the multibus RAM to the 1MB local RAM board. After the run-time program is copied into local RAM, controller is transferred to the run-time program. If the reset switch connected to the VIDS DMS were pressed, the boot program would recopy the run-time program in the multibus RAM to the local RAM and the run-time program would be restarted.

The National Semiconductor Multibus 512KB RAM board is temporarily being used in place of an EEROM board. Obviously, if power to the multibus board were to be turned off, the run-time program would be lost and as a result the run-time program would have to be re-downloaded again. In the next phase of the development, a multibus 512KB EEROM board is planned to replace the 512KB RAM board. The EEROM board would offer a non-volatile form of memory which can be easily reprogrammed to meet the needs of ever changing requirements.

5.4.4. Skeleton Operating System Implementation. The Dalmo Victor ARTOS is a fully defined Ada Run Time Operating System. It was found that only certain portions of the ARTOS are needed for a limited demonstration of the FDM. In the interest of time and cost, only a functional ARTOS containing the critical component was implemented. The functions which are not provided are:

- Command Language Interpreter (CLI)
- Trace and audit services
- Allocation of the 68000 central processor to individual software processes based on priority
- Real-time clock management.

A satellite 68000 emulator was used in place of the CLI and the trace and audit services. The emulator was used to examine memory, modify memory, and to trace program flow. It was especially useful in debugging the interface between the Bus Controller board and the VIDS DMS CPU board.

Allocation of the 68000 CPU would have greatly improved the performance of the VIDS FDM. It would have allowed high priority tasks to have larger allocations of the CPU time. Currently both high and low priority tasks have an equal chance for the CPU.

By also having a real-time clock, threat information could have been aged out more accurately. Currently, time is being simulated.

5.4.5. Utilities and Services. The skeleton ARTOS contains the following processes and utilities:

D_QUE_MGR (Data Queue Manager) provides two queues for management of the data handled between EDDM (External Device Data Manager) and EDM (External Device Manager). The first queue is for data traffic from EDDM to EDM and the second is for data traffic in the opposite direction. Data packets can be inserted or removed from either queue. A process must request to insert or remove a data packet from a queue.

EDM (External Device Manager) manages the transfer of data packets between the dual-port memory and the D_QUE_MGR (Data Queue Manager). The EDM continually polls the 1553B Bus Controller for data. If the Bus Controller acknowledges that data is available, the interrupt handler for the input will copy the data from the dual-port memory to a local buffer for the EDM to process. For data being transmitted to the 1553B Bus Controller, the EDM stores the data directly in the dual-port memory and interrupts the Bus Controller to transmit the data.

EDDM (External Device Data Manager) manages the transfer of data packets between SYS_SERV (System Services) and D_QUE_MGR (Data Queue Manager). It responds to requests from the application processes through SYS_SERV to read data coming from the 1553B Bus Controller. It also responds to requests from application processes thru SYS_SERV to write data to the 1553B Bus Controller.

SYS_DEF (System Definitions) contains the ARTOS parameters defined at system generation and also contains the return status definitions.

TERM_MGR (Terminal I/O Manager) responds to read or write requests by calling the terminal handler to receive or display character strings.

TERM_DRVR (Terminal I/O Driver) acts as an interrupt handler for the CRT terminal so that processes may request to read and write character strings to and from the terminal. This package is written in 68000 assembly.

IPC (InterProcess Communication) provides a message-passing facility. A process wishing to send a message specifies its own identify, the identity of the receiver and the message. A process wishing to receive a message specifies its own identity, the identity of the sender and a receptacle for the message. A bounded buffering facility is available to hold messages sent but not yet received.

STORAGE provides the memory for the messages passing via IPC. It also provides the functions necessary to send and retrieve a message.

MEMORY (Memory Manager) provides a method of allocating and deallocating memory to a process.

SYS_SERV (System Services) is the primary interface to the application software. Every ARTOS function which is available to the application software is specified in this package. The following is a list of functions and procedure and a description of what they do:

WHO_AM_I identifies the currently executing process.

ALLOCATE allocates memory to the requesting process.

FREE deallocates the dedicated memory.

SEND sends a message to another process.

RECEIVE receives a message from another process.

READ_DATA reads data packets from the EDDM.

WRITE_DATA passes data packets to the EDDM.

DELAAY suspends the calling process for a specified duration.

5.5. Ada Operational Application Modules (Software Packages)

The Ada operational application modules are all coded by Telesoft's 1.5 compiler which compiles a subset of the Ada language (see Figure 5-30). All of the current application codes, with few exceptions, will also compile on a fully validated Ada compiler.

The 1.5 compiler was selected because Telesoft had an embedded system kit for the 1.5 compiler but not for the fully validated 2.1 compiler. The embedded system kit provides utilities which allow the application software to execute on an embedded 68000 system.

The Ada language was chosen not only because it was a DOD-sponsored programming language but also because of the reduced life cycle costs of maintaining the code over a period of several years. As the VIDS DMS software evolves, maintainability becomes more critical; the cost of maintaining the software could rapidly exceed the initial development costs. When the Ada language was designed, a great deal of attention was given to keeping life cycle costs under control and reducing those costs from present levels.

5.5.1. Overview of Software Capability. The purpose of the VIDS DMS is to enhance the vehicle's survival capabilities by taking much of the burden of

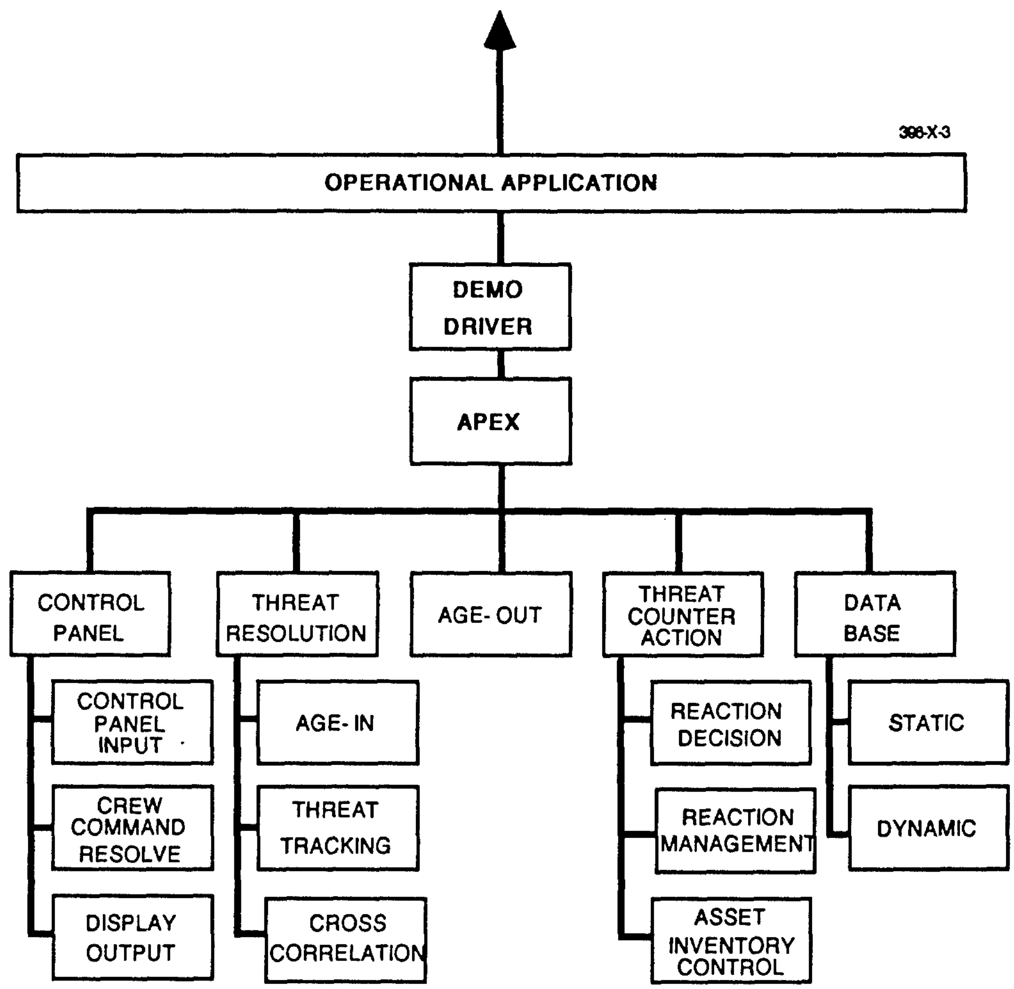


Figure 5-30. Ada Operational Application Software Modules

interpreting sensor data away from the crew so that the crew can devote all of their attention to the immediate threat. In order to accomplish this, the VIDS DMS software must be capable of:

- Automatic crew alert (audio and visual)
- Semiautomatic counteraction response and control
- Real-time embedded processing of threat detection, tracking, correlation, identification, prioritization, and reaction recommendation.

5.5.2. Databases. In order to satisfy the above requirements, the database must be structured in such a way which allows for fast and efficient access to the data. In an effort to compartmentalize the database, it is divided into two Ada packages, the STATIC_DATABASE and the DYNAMIC_DATABASE.

The STATIC_DATABASE contains data from which sensors can be correlated and also information on lethality, recommended counteractions, audio alert messages, and other static data.

The DYNAMIC_DATABASE provides the data structure for the Threat Track File (TTF), the Threat Correlation File (TCF), and the Priority Threat List (PTL). The TTF contains records with the raw sensor data; the TCF contains links (or references) to the correlated TTF records; and the PTL contains links (or references) to TTF records or TCF records. The DYNAMIC_DATABASE also provides the functions/procedures required to manipulate the links within the data structures. Even though the DYNAMIC_DATABASE package provides most of the rudimentary functions/procedure, the more complex operations are given their own package, in DB_TOOLS.

In a real-time environment, the database can be accessed simultaneously by different tasks trying to read or write to the database. For the STATIC_DATABASE, the data is static and is never modified during execution so there is no need to have protection for the STATIC_DATABASE. But in the case of the DYNAMIC_DATABASE, there exists the possibility of two different tasks trying to modify the same data at the same time. To ensure that this will never happen, a traffic controller (REALTIME_DBMS) is used to restrict access to the DYNAMIC_DATABASE so that concurrently running tasks would not collide or interfere with each other's operations. In the current version of the VIDS FDM, the REALTIME_DBMS monitor is not used because time slicing is not available with the current ARTOS. The current ARTOS uses a round-robin technique of scheduling each task. A task can keep control until it decides to give it up by making rendezvous with another task. The round-robin technique of scheduling tends to slow down system performance because background tasks are given as equal a chance to use the CPU as critical tasks.

Time slicing allows for a task to be interrupted in the middle of execution when its time allotment is up. The task is first suspended and the next task which is ready to execute is started. At the time when the first task is interrupted, it may have been in the middle of the modifying part of the database and the second task takes over and starts to read the half-modified data. That is when the conflict can occur and when the REALTIME_DBMS monitor would be required to protect the integrity of the data.

5.5.3. Operations and Processes. The most straightforward way to describe the VIDS DMS is to list sequentially the flow of data from input to output. Keep in mind that all VIDS DMS processes are running asynchronistly and it does not process data sequentially. The VIDS DMS processing is as follows:

- Immediately after a SIP is received by the VIDS DMS software, the type of input is determined and the data is buffered until the appropriate task requests for the data.

- If the input is from a specific sensor, the sensor data is compared with existing TTF records from the same sensor. If the sensor data does not match any of the existing records within a given tolerance, a new TTF record is created with the new sensor data. If the sensor data does match, the old TTF record is updated.

- The new sensor (TTF record) is then compared with other sensors to determine if there is a correlation between two sensors. If the correlated sensor has a TCF record the new TTF is added to the existing TCF record or else a new TCF record is created with references to the two TTF records. If there is no correlation, a new TCF record is created with a reference to the TTF record.

- The new TCF records are placed on the PTL by creating new PTL records with reference to the TCF record. This process is known as "aging-in" a threat.

- The PTL records are periodically scanned to see if the data to the display panel needs to be updated, or if the audio alert needs to be sounded, or if any automatic counteraction devices need to be activated.

- Crew input is handled differently from sensor input. Crew input is also buffered until the appropriate task requests for the data. The first type of crew input is from the touch screen. Touch-screen input indicates to the VIDS DMS that the crew has selected a threat to react to and the VIDS DMS will respond by recommending the appropriate counteractions for that threat. The second type of crew input is from the pushbutton controls. Button input indicates to the VIDS DMS to initiate the corresponding counteraction for the button pressed.

- All outputs are buffered for output until the ARTOS is ready to accept the data.

- In the background, old threats are continuously being purged from the database.

5.5.4. Major Tasks. The BIM (Bus Input Module) reads in data from the ARTOS which includes data from different sensors and commands from the crew. The data is then copied into different FIFO queues. The type of data determines which queue the data belong in. When the data is requested, the data is taken off the appropriate queue and passed to the process requesting the data.

There are six different TAM (Track And Match) processes, one for each sensor. Each process requests for sensor data from the BIM. If data is available, the TAM process calls upon the procedure MATCH_AND_UPDATE in the DB_TOOLS package. If a match cannot be found, a new TTF record is created with the new input, otherwise the old data is updated in the existing TTF record. New TTF records which can be correlated with others are handed off to the CORRELATION process else the uncorrelatible TTF records are handed off to the AGE_IN process.

The CORRELATION process takes data handed off to it from the TAM processes and performs cross-correlation with other sensors' TTF records. If a correlation is made, in the case where the correlated TTF records do not have an associated TCF record, a new TCF record is created with references to the correlated TTF records or in the case where one of the correlated TTF records has an associated TCF record, reference to the new TTF is added to the existing correlated TTF's TCF record. If no correlation is made, a new TCF record is created with a single reference to the new TTF record. All new TCF records are handed off to the AGE_IN process.

The AGE_IN process takes data handed off to it from the TAM processes and from the CORRELATION process and creates a PTL record for the threat.

The BOM (Bus Output Module) accepts data from both the Threat Counteraction Module (TCM) process and the CONTROL_PANEL processes and buffers all outgoing data in a FIFO queue and calls upon the ARTOS to pass the data to the Bus Controller to be transmitted over the 1553B bus to the Adapter Unit No. 2.

The CONTROL_PANEL process is responsible for scanning all the PTL records to check if a threat has moved or if it is to be erased or if a new threat has appeared in the database. The CONTROL_PANEL takes actions to send display information to the control panel. Also, if necessary, it will send audio alert information to the control panel, threat counteraction recommendations for the buttons to the control panel, and automatic counteractions to the printer, thereby simulating the operation and counteraction command input.

The TCM reacts to input from the control panel. The TCM functions in two modes of operation, automatic and manual. In automatic operation, the TCM will not respond to any button inputs except the AUTO/MAN button. If the AUTO/MAN is pressed while the TCM is in automatic operation, the TCM will revert back to manual operation. While in automatic operation, the TCM will select the most lethal threat and display the recommended counteraction. After a few seconds, the TCM will initiate the counteractions automatically. The crew has the option of cancelling the automatic operation during the few seconds before the counter-actions take place. Once the counteractions are initiated, the TCM will select the next most lethal threat to act on.

In manual operation, the crew selects the threat to react on by touching the corresponding symbol on the control panel screen. Immediately, the TCM will display the counteraction recommendations. The crew can press any of the counteraction buttons to initiate that counteraction. The counteraction button pressed need not be recommended; by pressing a non-recommended button, the crew indicated that they wish to override the VIDS DMS recommendations. To change mode of operation, the AUTO/MAN button can be pressed to put the TCM into automatic operation. In automatic or manual, once a counteraction is started, it can not be stopped.

The AGE_OUT process periodically scans the database for outdated information to be removed from the database. If the threat is currently being displayed on the control panel screen, the AGE_OUT process flags the threat so that the CONTROL_PANEL process will erase it from the screen. Once the threat is removed from the screen, the CONTROL_PANEL will flag the threat so that when the AGE_OUT process scans the database again, it will delete that threat from the database.

5.5.5. Development Effort Summary. All the initial development of the VIDS DMS code was developed on the Callan/Unistar 200 stand-alone workstation. Initial design and code of the database took several iterations before the data structures were finalized. Due to restrictions imposed by the Telesoft compiler, variant records were replaced by fixed record types.

After the database was completed, the BIM was the first major task to be coded. Because ARTOS was not available at the time, a temporary operating system to fake the function of the ARTOS was used. SIPs were simulated by reading a data file with predefined data. (All of the inputs were defined from the Tactical Engagement Scenarios of simulated battlefield examples.) To debug the BIM process, output to the CRT terminal was used to display what the BIM process was receiving from our dummy operating system.

Next came the TAM processes; there are six TAM processes, one for each sensor. Each of the TAM processes was added to the VIDS DMS software one at a time. Each time a new TAM process was added, the TAM process was checked to determine that the data was transferred correctly from the BIM process.

The next two processes to be added to the VIDS DMS software were the AGE_IN process and the CORRELATION process. After a threat was aged in by the AGE_IN process, the CORRELATION process was used to verify the sensor was aged in.

After they were debugged, the CONTROL_PANEL and the BOM was added. The display output for the CONTROL_PANEL was developed first. The display data output from the CONTROL_PANEL process to the BOM process was used to verify that the correlation and tracking was taking place. Based upon the display output data from the BOM, the track and match algorithm and the correlation algorithm were refined and corrected. The next function to be added was the counteraction recommendation.

To test the output, in an intermediary hardware configuration, the Callan 200 system was hooked up serially to the control panel. The fake operating system would accept data from the BOM and reformat the data into a format which the control panel expects and would output the data over the RS-232 line. By cutting out the adapter units, bus controller, ARTOS, and other VIDS FDM hardware, the Callan 200 with the VIDS DMS could interface directly with the control panel, thus resolving any communication problems early on.

The TCM was next to be added to allow for simulated button input to be integrated into the system to see if the system would respond with the correct countermeasure.

Once that was accomplished successfully, the AGE_OUT process was added to age out threats after an arbitrary length of time.

Further details of the application software are found in the enhanced specifications and the actual listings of the code.

5.6. Integration

While we note that the principle thrust of the VIDS DMS FDM development was software intensive, the engineering discipline was system integration. The extensive use of digital processing in the resultant "Expert System" required that not only was it essential to design a well integrated set of hardware subsystems but was also essential that all of the software in the system (firmware code as well as disk file programs) first be integrated within itself, and then subsequently integrated (embedded) with the hardware as a complete system.

Software was integrated and tested on the host workstations (UNISTAR 100) where sufficient edit/assembly and debug tools were available. Following satisfactory performance on the host, the DMS software was downloaded to RAM space in the DMS hardware and then retested in the now embedded program residing within this standalone DMS. This standalone DMS can be completely detached from the workstation.

In this section we first describe the functional relationships and integration interfaces for the hardware system, and then describe the software system integration.

5.6.1. Hardware Integration for FDM: Functional Relationships. The overall configuration of the hardware comprising the FDM was shown previously in Figure 5-3. An electrical interface diagram is provided for reference in this section as Figure 5-31 on the following page. This diagram represents the entire FDM and generally flows from left to right. A summary of these functions and the transfer relationship between hardware subassemblies of the FDM are described in the following paragraphs.

5.6.1.1. TESS to Adapter Unit No. 31: The TESS creates a file containing the proposed scenario of the simulated equipment. The data is read from the file, one situation at a time, and loaded into the input buffers of Adapter Unit No. 31 over an RS-232 bus. The data buffers of Adapter Unit No. 31 simulate the receiving of emissions by the absent individual sensors. The output circuits of Adapter Unit No. 31 emulate the processed data words that would be sent from the sensors if they were realized in hardware and were stimulated by the emitters which we simulate in the output of the TESS.

Summary Statement: TESS stimulates the emulation which puts out TTL levels as if the actual sensors were present and operating.

5.6.1.2. Adapter Unit No. 3 to Adapter Unit No. 1. The output of Adapter Unit No. 3 "emulates" the physical sensors as if they were present. Since our FDM contract does not include the actual use of real sensors, we emulate their combined existence in the hardware of Adapter Unit No. 3. Some sensors output their data on serial lines, some on parallel lines. These diverse hardware interfaces are replicated and input to Adapter Unit No. 1 as if the actual sensors were present and receiving emissions from the various platforms they are intended to detect.

Adapter Unit No. 1 has two principal functions:

- It accepts input data from the several diverse hardware interfaces and buffers it into storage suitable for polling by the external 1553B Bus Controller.

- It provides a 1553B Remote Terminal Unit (RTU) which can be made to appear like a 1553B standard bus termination for each of the virtual sensors emulated by Adapter Unit No. 3. The common RTU then commutates the data from the individual virtual sensors, as that data is called for by the Bus Controller, and communicates it to the DMS via the Bus Controller.

Summary Statement: Adapter Unit No. 3 outputs TTL data as if the virtual sensors were present and transmits the data to Adapter Unit No. 1 where it is buffered, commutated, and made available for retransmission on a standard bus interface to the DMS.

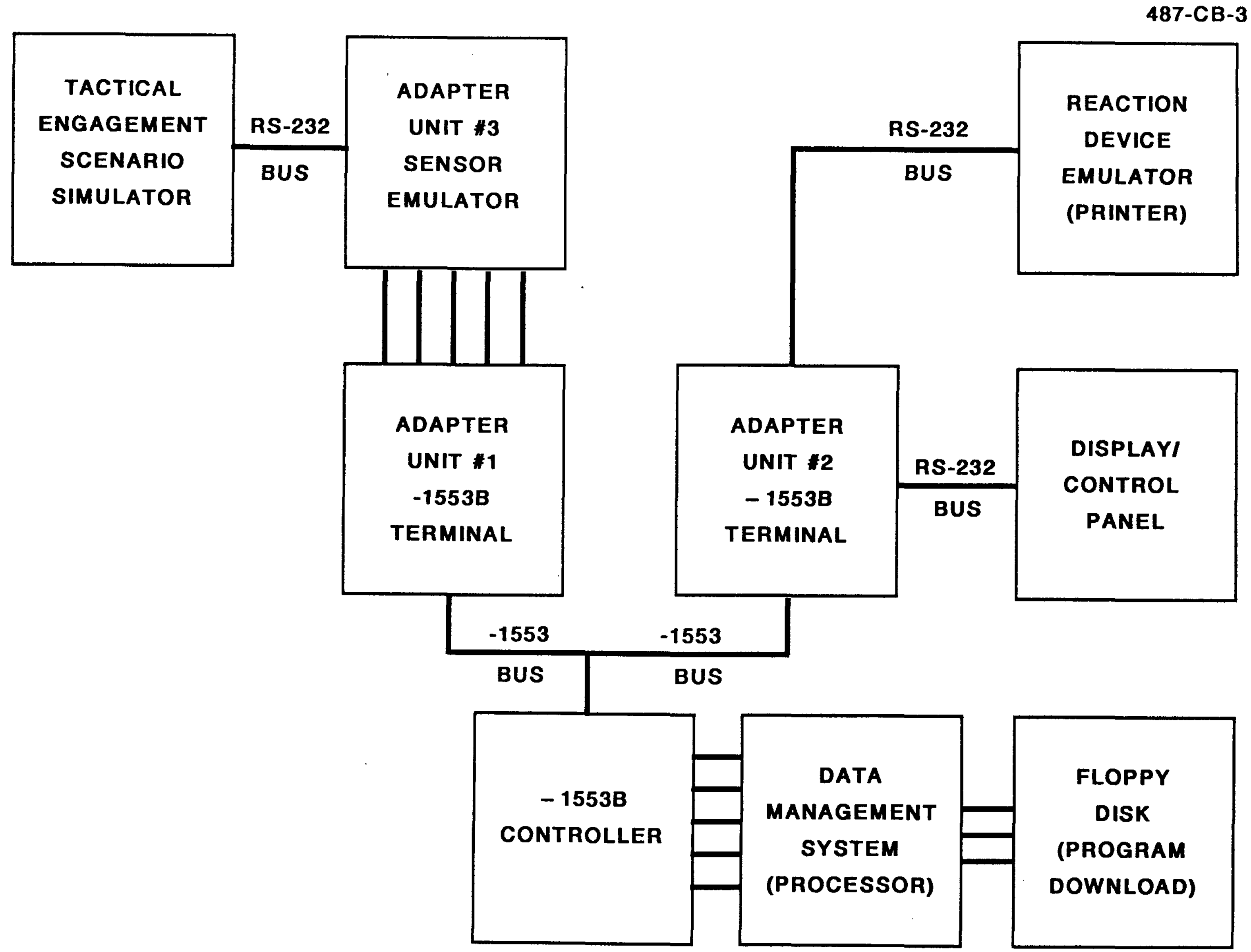


Figure 5-31. FDM Interface Diagram

5.6.1.3. Adapter Unit No. 1 to Bus Controller. As indicated above, Adapter Unit No. 1 commutates the multiple sensor data and makes it available for retransmission. A MIL-STD-1553B Remote Terminal Unit resides within Adapter Unit No. 1 to interface with the actual 1553 bus. When data is requested by the DMS, the 1553B Bus Controller board polls the RTUs for data. When they have data, it is communicated to the Bus Controller in accordance with 1553 protocol. When no data is present or is in the process of accumulation, the RTU signals that it is busy. When data is ready for transmission, it is communicated in standard words to the Bus Controller as described earlier in Section 5.3.3.3.

Summary Statement: Adapter Unit No. 1 accomplishes one of the primary design requirements of the contract by providing a single standard interface to the DMS for the sensors. This is done via MIL-STD-1553B. A 1553B Bus Controller board is contained within the DMS to manage this communication function from 2 (expandable to 32) remote terminals.

5.6.1.4. Bus Controller to and from DMS Processor. The Bus Controller contains 2K-by-16 dual-port RAM on the board. This is physically connected to the DMS processor board (main CPU) by way of IEEE 796 (multibus). The dual-ported RAM acts as four sets of mailboxes. Two of these mailboxes can be written into from the 1553 bus receivers. These same two mailboxes alternately are read out to the DMS RAM board under the control of the special ARTOS EDM software module.

The other two mailboxes are read into from the DMS RAM space, and can be read out to the 1553 for transmission out onto the bus for use by the remote terminals. The writing and reading of these mailboxes is under the control of the special ARTOS (EDDM). The complex interface between these dual-ported RAMs and the DMS is described in greater detail in Section 5.3.3.3.

Summary Statement: The Bus Controller polls the sensor data from the RTU and loads it into alternate action mailboxes which are read into DMS memory space under control of the ARTOS. Outward bound messages for either the sensors or the reaction devices and Display/Control Panel are likewise communicated from DMS RAM space via the dual-ported RAM on the 1553 controller board for subsequent transmission on the 1553B bus.

5.6.1.5. Bus Controller to Adapter Unit No. 2: The function of Adapter Unit No. 2 is to provide a standard interface to the DMS (1553B) by the counteraction devices and the Display/Control Panel, which is a fundamental requirement of the contract specification for the FDM. It also provides a secondary standard bus (RS-232) interface to the counteraction device emulator (which is a printer to record counteraction messages) and the Display/Control Panel CPU board. The reason for transferring messages from the 1553B to the RS-232 is simply that existing printers (recorders) and displays are available off the shelf with an RS-232 interface built in. Since these devices are not intended to duplicate the devices they represent but merely to emulate them, the use of commercial equipment and data interfaces is acceptable for the FDM.

When the DMS determines a threat symbol to be displayed and a voice alert message to be created, it sends the appropriate message to the external device first over the 1553B bus and eventually over the RS-232 by way of the Bus Controller and Adapter Unit No. 2. Also, counteraction command messages are sent to the recorder via Adapter Unit No. 2 where they are recoded into ASCII for printout to confirm the validity of the response as planned.

Soldier-Machine Interface (SMI) is provided by simple pushbuttons on the control panel and by a touch-sensitive screen over the display image area. Signals form these circuits are transmitted first to Adapter Unit No. 2 as RS-232 words, then to the DMS via the 1553B bus from Adapter Unit No. 2. A detailed description of the messages between the DMS and the external devices is provided in Section 8.

Summary Statement: The 1553B Bus Controller sends command message via the Adapter Unit No. 2 to the counteraction device emulator for recording on paper tape and to the Display/Control Panel for threat symbol display and voice alert. Touch-screen response messages are communicated from the Display/Control Panel to the DMS in reverse order of the above (RS-232 to 1553B).

5.6.2. Integration Procedure. Integration of the FDM takes place in four steps. These steps are organized as logical packages of hardware with their embedded firmware and software downloaded to RAM. These four integration steps are defined in the following paragraphs.

5.6.2.1. TESS/Adapter Unit No. 3/Adapter Unit No. 1. SIPs are created in the TESS, which is an Ada-based file generator. The SIPs are communicated via RS-233 to Adapter Unit No. 3 where they are integrated and buffered. The first check is to see that the SIPs are read properly. They are then converted to various serial and parallel outputs at TTL. These outputs from Adapter Unit No. 3 are monitored with a logic analyzer and input to Adapter Unit No. 1.

Inputs here are buffered for commutation to the front end of Adapter Unit No. 1. The TTL signals are converted to words suitable for use by a 1553B RTU. These sensor outputs are stored in "channels" representing the sensor name and made available for callout by the RTU. These words are checked with a logic analyzer. The RTU is interfaced with a 1553B bus analyzer (SBR100A) and tested for end-to-end validity of data. This completes the first step of integration.

5.6.2.2. Bus Controller and ARTOS. The protocol software for the 1553B controller board is written and coded on a UNISTAR 200 workstation in assembly language for the MC68000 (ASR68K). It is then loaded into the controller board via an in-circuit emulator for debug and test. The controller board is then tested with the SBR100A bus tester.

The ARTOS has been developed, coded, and debugged on the UNISTAR 200. Special ARTOS boot PROM chips have been burned and inserted in the main CPU board in the UNISTAR 200. The ARTOS is now downloaded from 5 1/4-inch floppy disk into the UNISTAR RAM space and tested with special test routines developed to support ARTOS.

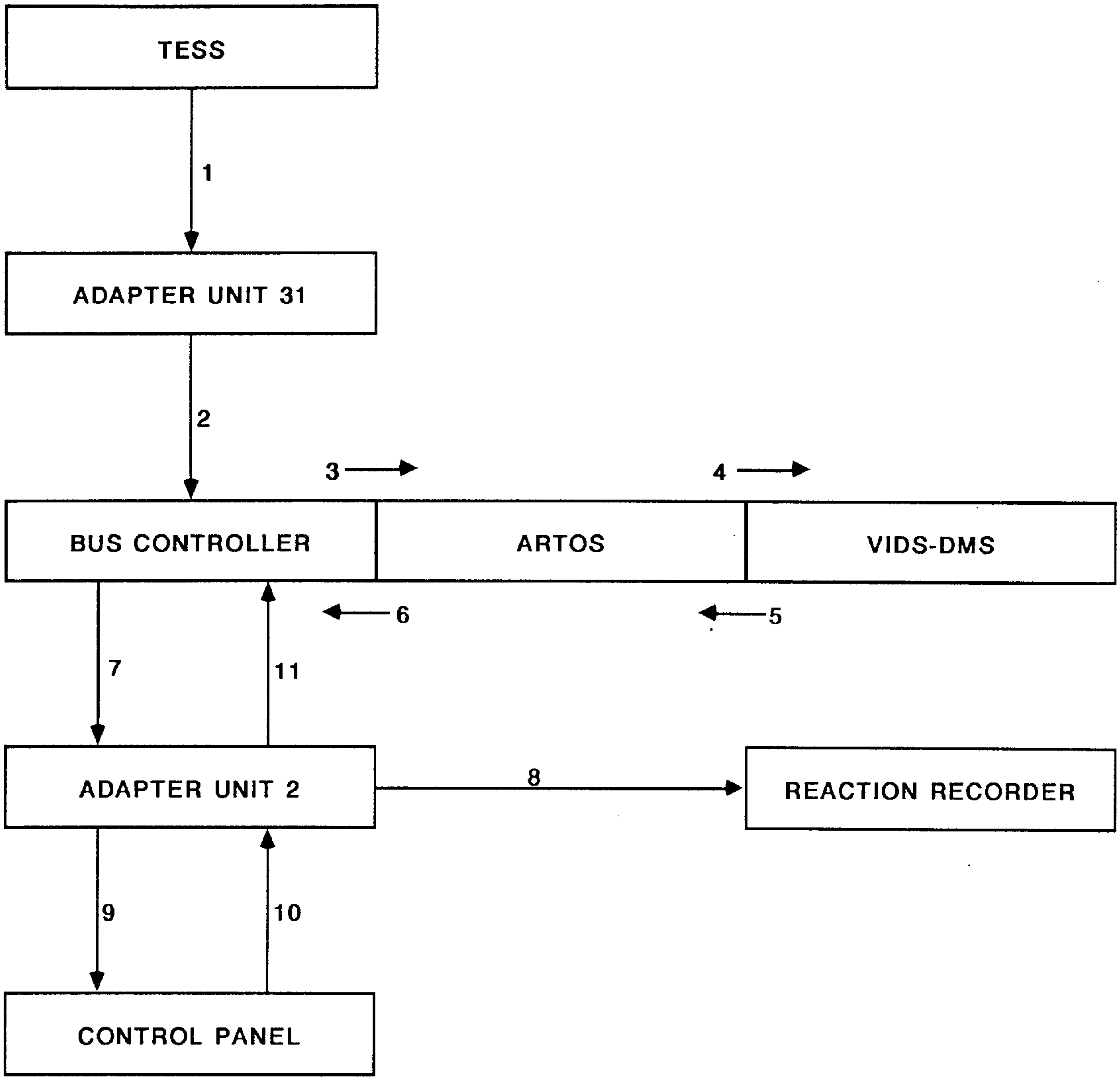
When this test is complete, the Bus Controller is installed in the UNISTAR and tested for addressing reception and transmission of data from the controller board dual-port RAM to the UNISTAR system bus (IEEE 796) at the command of ARTOS. Messages (data) from the DMS to the controller board are also called by the ARTOS via the dual-port RAM and read out to the controller board also by the system bus (multibus). When this is shown to be properly operating, the Bus Controller board is interfaced with the SBR100A for test of messages from the 1553B bus to the DMS and for messages from the DMS to the 1553B bus.

The integration step is completed when the DMS application software is downloaded from 5 1/4-inch floppy disk to the UNISTAR 200 RAM space and shown to run in response to input/output frp, the Bus Controller as supported by ARTOS.

5.6.2.3. Bus Controller/Adapter Unit No. 2/Display-Control Panel/ Counteraction Emulator (Printer). Messages are simulated in the 1553B bus tester (SBR100A) which input Adapter Unit No. 2 as 1553B formatted words. These messages are decoded, translated into RS-232 format, and forwarded to the drivers of one of two RS-232 serial ports in Adapter Unit No. 2. Display messages result in symbology code for the display and voice alert messages. These are decoded with the CPU board within the Display/Control Panel and result in symbols being placed on the screen and synthetic voice alert sounds being generated within the Control Panel. Touch panel and pushbutton control messages are transmitted backward to the 1553B Bus Controller via Adapter Unit No. 2, which converts RS-232 type segments to 1553B messages and places them in the RTU of Adapter Unit No. 2.

Similarly, counteraction messages are sent from the 1553B bus tester to Adapter Unit No. 2 where they are converted to ASCII code for display on the strip printer.

5.6.3. Software Integration. The VIDS FDM is composed of several different pieces of software modules executing asynchronously and is networked together by several hardware and software interfaces. In order to allow for software development to proceed in parallel, the data format and the software protocol between the major software modules were defined early in the program. After defining the data format and the software protocol, each software module could be developed independently from the rest of the system and could also be debugged and tested separately. As a software module matures, it is added incrementally to the system. In Figure 5-32, the major software modules of the system are divided up and the arrows indicate the direction of the data flow between them:



1. TESS & Adapter Unit No. 31
2. Adapter Unit No. 31 & Bus Controller
3. Bus Controller & ARTOS
4. ARTOS & VIDS DMS
5. VIDS DMS & ARTOS
6. ARTOS & Bus Controller
7. Bus Controller & Adapter Unit No. 2 Controller
8. Adapter Unit No. 2 & Reaction Recorder
9. Adapter Unit No. 2 & Control Panel
10. Control Panel and Adapter Unit No. 2
11. Adapter Unit No. 2 & Bus Controller

Figure 5-32. Data Flow Between Hardware Subassemblies

1. TESS and Adapter Unit No. 31. Data is transmitted serially from the TESS to Adapter Unit No. 31 via the RS-232 line. Each SIP is composed of 6 x 16 bit words, where the high byte is transmitted before the low byte. Each Sensor Input Packet is defined as follows:

1.	Sync Word	(-1)
2.	Threat Identification	(0 .. 12)
3.	Azimuth (degrees)	(0 .. 360)
4.	Elevation (degrees)	(-90 .. +90)
5.	Range (meters)	(0 .. 10,000)
6.	Checksum	(Sum of words 1 to 5, negated)

Where the Threat Identification is defined as follows:

Index	Emitter	Sensor
0	Laser Range Finder	Laser
1	Laser Designator	Laser
2	Beam Rider	Laser
3	Optics (Large Platform)	Optical Warning
4	Optics (Small Platform)	Optical Warning
5	Attack Helo Type 1	NIS
6	Attack Helo Type	NIS
7	Scout Helo	NIS
8	Missile	PMD
9	Millimeter Wave #1	MMW
10	Millimeter Wave #2	MMW
11	Nuclear	NBC
12	Chemical	NBC

there is no protocol between the two modules so it is up to the firmware in Adapter Unit No. 31 to keep up with the data transmission.

2. Adapter Unit No. 31 and Bus Controller. After receiving a complete SIP, the Adapter Unit No. 31 strips off the sync word and the checksum and buffers the data by sensor type in FIFO queue. When the Bus Controller polls the adapter unit for data from a particular sensor, the adapter unit removes the SIP from the appropriate queue, adds the device number and input type before the SIP, and reformats the data for transmission from Adapter Unit No. 31 to the Bus Controller:

1553B OUTPUT MESSAGE FORMAT ON THE ADAPTER UNIT NO. 31 SIDE:

Word 1: Command word to 1553B transmitter/receiver chip
2: 1553B Receive command transmitted over the bus as command to the remote terminal
3: Undefined
4: 1st data word of output message (device number)
5: 2nd data word of output message (input type)
6: 3rd data word of output message (threat identification)
7: 4th data word of output message (azimuth)
8: 5th data word of output message (elevation)
9: 6th data word of output message (range)
11: Status word transmitted by the remote terminal back to the Bus Controller

1553B POLLING MESSAGE FORMAT ON THE BUS CONTROLLER SIDE:

Word 1: Command word to 1553B transmitter/receiver chip
2: 1553B Receive command transmitted over the bus as command to the remote terminal
3: Status word transmitted by the remote terminal back to the Bus Controller
4: Undefined
5: 1st data word of input message
6: 2nd data word of input message
7: 3rd data word of input message
8: 4th data word of input message
9: 5th data word of input message
10: 6th data word of input message

3. Bus Controller and ARTOS. The Bus Controller polls for data from both the sensors and the Control Panel. Upon receiving the message "Complete Interrupt," the Bus Controller must check the busy bit in the status word. If the busy bit is not set, then that implies that there is data available in words 5 to 10. If the busy bit is set, then there is no data available. If data is available, the six data words are transferred to one of the eight word slots in Segment 1 or Segment 2 of the dual-port RAM. Upon receiving a request for data interrupt from the ARTOS, the Bus Controller sends an "acknowledge interrupt" to ARTOS, releases the segment it is working on to ARTOS and begins to fill the other segment. The ARTOS transfers the available segment to local memory and them searches each slot for valid data. To determine if a slot has valid data, the last word in the slot must be greater than zero; once data is found, the ARTOS buffers the data.

4. ARTOS and VIDS DMS. The ARTOS buffers the 8-word packet in a FIFO queue. In the last two words of the packet, ARTOS puts a simulated time stamp. The VIDS DMS requests for data from the ARTOS by way of SYS-SERV (system services). Because of a bug in the Telesoft compiler, it is necessary to perform two rendezvous in order to perform the data transfer between the ARTOS and the VIDS DMS. All data transferred are the standard length of eight words. The data formats are shown in Figures 5-33 and 5-35.

Word 1	DEVICE NUMBER		
Word 2	INPUT TYPE		
Word 3	TYPE	BUTTON NUMBER/ BUTTON STATUS	PTL ID
Word 4	AZIMUTH		
Word 5	ELEVATION		
Word 6	RANGE		
Word 7	TIME TAG FROM ARTOS		
Word 8			

Figure 5-33. Standard VIDS Input Data Format

SENSOR INPUT:

Word 1:	Device number	(1)
2:	Input type	(1)
3:	Type of sensor input	(0 .. 12)
4:	Azimuth	(0 .. 360)
5:	Elevation	(-90 .. 90)
6:	Range	(0 .. 10,000)
7:	Time tag from ARTOS	(High word)
8:	Time tag from ARTOS	(Low word)

BUTTON INPUT: (Figure 5-34 showns button placement on panel)

Word 1:	Device number	(1)
2:	Input type	(2)
3:	Button number/status	(1 .. 17 / 0 .. 3)
4:	Undefined	
5:	Undefined	
6:	Undefined	
7:	Time tag from ARTOS	(High word)
8:	Time tag from ARTOS	(Low word)

TOUCH SCREEN INPUT:

Word 1:	Device number	(1)
2:	Input type	(3)
3:	PTL ID number	(1 .. 64)
4:	Undefined	
5:	Undefined	
6:	Undefined	
7:	Time tag from ARTOS	(High word)
8:	Time tag form ARTOS	(Low word)

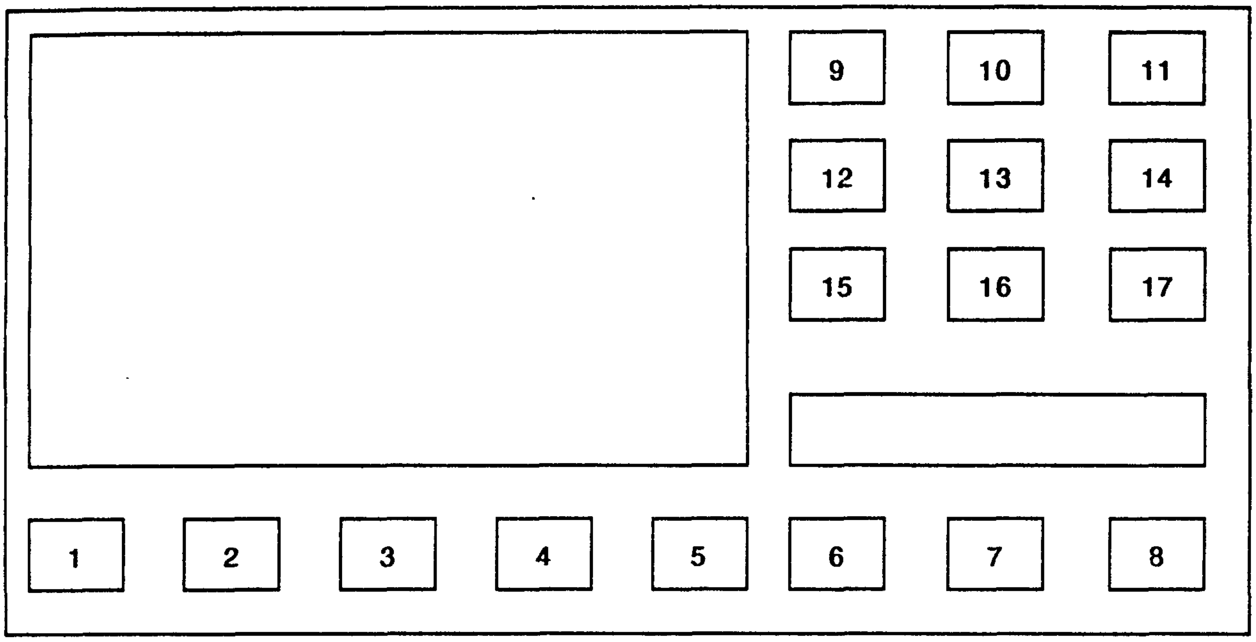


Figure 5-34. Control Panel Button Placement

The button status in word number 3 of the button input is defined as follows:

0 - Both red and yellow LEDs are off
1 - Only the red LED is on
2 - Only the yellow LED is on
3 - Both red and yellow LEDs are on

5. VIDS DMS and ARTOS. After processing the input data, the VIDS DMS may output the following data to the ARTOS:

Word 1	DEVICE NUMBER			
Word 2	PTL ID FOR SYMBOL	(HIGH BYTE / OUTPUT TYPE (LOW BYTE)		
Word 3	AUDIO ALERT FOR THREAT	BUTTON STATUS / NUMBER	DRAW COMMAND / LETHALITY	AUTO REACTION
Word 4	O'CLOCK-	BUTTON STATUS / NUMBER	SYMBOL TYPE	MANUAL REACTION
Word 5	COUNTERACTION RECOMMENDED	BUTTON STATUS / NUMBER	AZIMUTH	AZIMUTH
Word 6	COUNTERACTION RECOMMENDED	BUTTON STATUS / NUMBER	RANGE	RANGE
Word 7	COUNTERACTION RECOMMENDED	BUTTON STATUS / NUMBER		
Word 8	RESERVED FOR OPERATING SYSTEM USE			

Figure 5-35. Standard VIDS Output Data Format

AUDIO OUTPUT:

Word 1:	Device number	(2)
2:	PTL ID / Output type	(0 .. 64 / 1)
3:	Audio alert for threat	(1 .. 367)
4:	O'Clock position	(1 .. 12)
5:	Counteraction recommended	(0 .. 367)
6:	Counteraction recommended	(0 .. 367)
7:	Counteraction recommended	(0 .. 367)
8:	Reserved for operating system use	

(Note: If there is a zero in word 5, 6, or 7, then that indicates that there are no further audio counteractions recommended. Thus, if word 6 contained a zero, the audio alert message would contain only one counteraction recommended from word 5; word 7 would be ignored.)

BUTTON OUTPUT:

Word 1:	Device number	(2)
2:	PTL ID / Output type	(0 .. 64 / 2)
3:	Button status / number	(0 .. 3 / 0 .. 17)
4:	Button status / number	(0 .. 3 / 0 .. 17)
5:	Button status / number	(0 .. 3 / 0 .. 17)
6:	button status / number	(0 .. 3 / 0 .. 17)
7:	Button status / number	(0 .. 3 / 0 .. 17)
8:	Reserved for operating system use	

(Note: If there is a zero in word 4, 5, 6, or 7, that indicates that there are no further button commands. Hence, if word 5 contained a zero, the status for the buttons indicated in words 3 and 4 would be displayed and words 6 and 7 would be ignored.)

SYMBOL OUTPUT:

Word 1:	Device number	(2)
2:	PTL ID / Output type	(0 .. 64 / 3)
3:	Draw Command / Lethality	(1 .. 3 / 1 .. 5)
4:	Symbol type	(1 .. 39)
5:	Azimuth	(0 .. 360)
6:	Range	(0 .. 10,000)
7:	Undefined	
8:	Reserved for operating system use	

(Note: For the draw command:
1 - Erase the symbol
2 - Move the symbol
3 - Create a new symbol
For the symbol type, the numbering of the symbols corresponds to the numbering on the "VIDS DMS Platform Identification and Reaction Analysis" Chart.)

REACTION OUTPUT:

Word 1:	Device number	(4)
2:	PTL ID / Output type	(0 .. 64 / 4)
3:	Automatic reaction	(0 .. 14)
4:	Manual reaction	(0 .. 14)
5:	Azimuth	(0 .. 360)
6:	Range	(0 .. 10,000)
7:	Undefined	
8:	Reserved for operating system use	

6. The ARTOS buffers all outputs from the VIDS DMS in a FIFO queue and waits until the Bus Controller is ready to transmit more data. The Bus Controller sends an interrupt to the ARTOS to acknowledge that it is ready to transmit; the ARTOS removes the data from the queue and copies the data into segments 3 and 4 of the two-port RAM. (There are three words reserved for the ARTOS use prior to 8-word data packet from the VIDS DMS in segments 3 and 4.) The ARTOS then interrupts the Bus Controller to transmit the data.

7. The Bus Controller formats the data for 1553B transmission to Adapter Unit No. 2 as follows:

1553B OUTPUT MESSAGE FORMAT ON THE BUS CONTROLLER SIDE:

Word 1: Command word to 1553B transmitter/receiver chip
2: 1553B Receive command transmitted over the bus as command to the remote terminal
3: Undefined
4: 1st data word of output message
5: 2nd data word of output message
6: 3rd data word of output message
7: 4th data word of output message
8: 5th data word of output message
9: 6th data word of output message
10: 7th data word of output message
11: Status word transmitted by the remote terminal back to the Bus Controller

1553B INPUT MESSAGE FORMAT ON THE ADAPTER UNIT NO. 2 SIDE:

Word 1: Command word to 1553B transmitter/receiver chip
2: 1553B Receive command transmitted over the bus as command to the remote terminal
3: Status word transmitted by the remote terminal back to the Bus Controller
4: Undefined
5: 1st data word of output message
6: 2nd data word of output message
7: 3rd data word of output message
8: 4th data word of output message
9: 5th data word of output message
10: 6th data word of output message

8. If the data is a counteraction, Adapter Unit No. 2 forms the appropriate reaction message and transmits the ASCII message serially to the Reaction Recorder. There is no protocol and, because the Reaction Recorder has no buffering capability, it is up to Adapter Unit No. 2 to transmit at a rate slow enough so that the recorder does not lose any character.

9. If the data is for the Control Panel, Adapter Unit No. 2 removes the device number and transmits the data serially to the Control Panel in the following format (Figure 5-36):

Word 1	SYNC WORD: -1		
Word 2	PTL ID (HIGH BYTE / OUTPUT TYPE (LOW BYTE)		
Word 3	AUDIO ALERT FOR THREAT	BUTTON STATUS / NUMBER	DRAW COMMAND / LETHALITY
Word 4	O'CLOCK-	BUTTON STATUS / NUMBER	SYMBOL TYPE
Word 5	COUNTERACTION RECOMMENDED	BUTTON STATUS / NUMBER	AZIMUTH
Word 6	COUNTERACTION RECOMMENDED	BUTTON STATUS / NUMBER	RANGE
Word 7	COUNTERACTION RECOMMENDED	BUTTON STATUS / NUMBER	
Word 8	CHECKSUM: (-1*(Word 1 + ... + Word 7)		

Figure 5-36. Control Panel Input Data Format

10. If the Control Panel receives input from the crew, there are only two possible data formats which are transmitted serially from the Control Panel to Adapter Unit No. 2. The crew input can be encoded as one of the following (Figure 5-37):

Word 1	SYNC WORD: -1	
Word 2	INPUT TYPE	
Word 3	BUTTON STATUS / NUMBER	PTL ID
Word 4 Word 5 Word 6		
Word 7	CHECKSUM: (-1*(Word 1 + ... + Word 7)	

Figure 5-37. Crew Input Data Format

11. Adapter Unit No. 2 strips both the checksum and sync word from the data received from the Control Panel. It then appends a device number to form a 6-word data packet. The data format from Adapter Unit No. 2 to the Bus Controller is as follows:

1553B OUTPUT MESSAGE FORMAT ON THE ADAPTER UNIT NO. 2 SIDE:

Word 1: Command word to 1553B transmitter/receiver chip
2: 1553B Receive command transmitted over the bus as command to the remote terminal
3: Undefined
4: 1st data word of output message
5: 2nd data word of output message
6: 3rd data word of output message
7: 4th data word of output message
8: 5th data word of output message
9: 6th data word of output message
11: Status word transmitted by the remote terminal back to the Bus Controller

1553B POLLING MESSAGE FORMAT ON THE BUS CONTROLLER SIDE:

Word 1: Command word to 1553B transmitter/receiver chip
2: 1553B Receive command transmitted over the bus as command to the remote terminal
3: Status word transmitted by the remote terminal back to the Bus Controller
4: Undefined
5: 1st data word of input message
6: 2nd data word of input message
7: 3rd data word of input message
8: 4th data word of input message
9: 5th data word of input message
10: 6th data word of input message

Summary Statement: Each of the major software modules was tested to the greatest extent possible without the rest of the VIDS FDM hardware. The VIDS DMS was first integrated with the ARTOS on the Callan workstation by downloading from the floppy drive. Next, the 1553B Bus Controller was added to the Callan work-station to test the communication between the Bus Controller and the ARTOS. Once the VIDS DMS, ARTOS, and Bus Controller were working on Callan, all the boards were transferred to the embedded MC68000 system. In parallel, the TESS was integrated with Adapter Unit No. 31, Adapter Unit No. 2 with the Control Panel and the Reaction Recorder, and the Bus Controller with both adapter units. After system integration was completed, the floppy disk and the floppy disk controller was replaced by 512K Byte RAM board which allowed the VIDS DMS and ARTOS to be downloaded directly from its workstation via the RS-232 line.

Lessons Learned: Much of the software integration went very smoothly except for the stack and heap size which had to be increased for the VIDS DMS and the ARTOS. Also, the interrupt handlers in the ARTOS would not work when they were written in Ada so they had to be rewritten in assembly code.

Unfortunately, a symbolic debugger for the Ada code was not available. If it had been available, debugging would have been much simpler.

5.6.4. System Integration. The eventual demonstration testing of the FDM results from the final stage of integration, when each preceding step of integration is successfully concluded and all hardware is connected together for end-to-end tests. TESS is actuated to generate SIPs for the test scenario, and eventual report to the DMS by way of Adapter Unit No. 3, Adapter Unit No. 1, and the Controller Board.

The DMS application software responds to the SIPs (as detailed in the software functional descriptions and annotated source listings) and sends the appropriate reaction recommendations, display, and voice alert commands out to the bus. The test is complete when appropriate symbols are displayed, proper voice sounds are generated, correct counteractions are recorded on the printer, proper pushbutton and touch-screen responses are processed by the DMS and returned to the Control Panel and counter-action recorder as supervisory signals to the colored LEDs in the push-buttons, and ASCII code and words on the printer.

5.7. TESS

The initial VIDS Phase I study recognized that actual sensors would not be in place for the Feasibility Demonstration. We therefore undertook (on IR&D funding) to develop a software model in Ada to simulate the data expected from the sensors. We did this by first creating a topography-based threat scenario which resulted in the generation of SIPs. These SIPs represented "sighting" of threat observables at a given location relative to our tank. We then converted these "sightings" into the actual TTL-level sensor data output that would eventually be placed on the 1553B data bus to the DMS.

5.7.1. Overview of TESS. The program for the VIDS FDM demonstration scenario executes on the Dalmo Victor TESS. It runs on a Callan Data System UNISTAR 100 workstation (terminal) from a prepared floppy disk. Communication between the TESS environment model and the Adapter Unit No. 3 software is performed by means of a standard RS-232 interface at a rate of 1200 baud. All data is sent to a serial RS-232 port.

Individual modules are defined as the following:

Emitter	Creates static database of platform information
Raw Scenario Module	Collects input data used to formulate an operational scenario
Prepared Scenario Module	Processes inputs to create the operational scenario (story)
Tank Motion Module	Inputs position of own tank
Scenario Generator	Covers scenario data to simultated sensor inputs
Scenario Manager	Real-time simulator
Display Transmitter	Drivers to send data to subsystem destinations
VIDS Transmitter	i.e., ESSM display and sensor emulator (or Adapter Unit No. 3)

The relationship of these modules is shown in Figure 5-38.

5.7.2. Software Formats. Data for a Single Threat: A threat report will have one 16-bit word for each field as specified in Table 5-5. The parameter window or restrictions on the data are also specified in Table 5-5.

The Scenario Generator creates sensor data for every 250-ms interval of the test scenario. The reporting of sensor data is repeated as long as the scenario calls for continuous presence of the emitter. A graphic representation of a sample scenario is presented in the following paragraphs. Figure 5-39 illustrates placement of platforms at a given time.

The appearance and disappearance of platforms in the environment is independent of the presence of other platforms and of the time they can overlap as illustrated in the following time line drawing, Figure 5-40.

In this scenario, Platform P2 is "turned on" at 100 ms. At 200 ms, Platform P1 is "turned on." The scenario generator will report two active platforms, P1 and P2, at 250 ms and 500 ms. Platform P2 is turned off at 750 ms. The scenario generator will report three active platforms at 750 ms. The time lag between platforms P4 and P5 represents a break between situations.

Any platform that has been active during the reporting period is included in the SIP report. If a platform is turned on at the start of a reporting period (as illustrated by Platfor P4), it is not included in the report for the previous period. For example, Platform P4 is not reoprted at 2,000 ms but is reported at 2,250 ms. It is also possible to have a reporting period in which no platforms are reported. In this case, no data is sent out to the Adapter Unit and the passage of time is simulated.

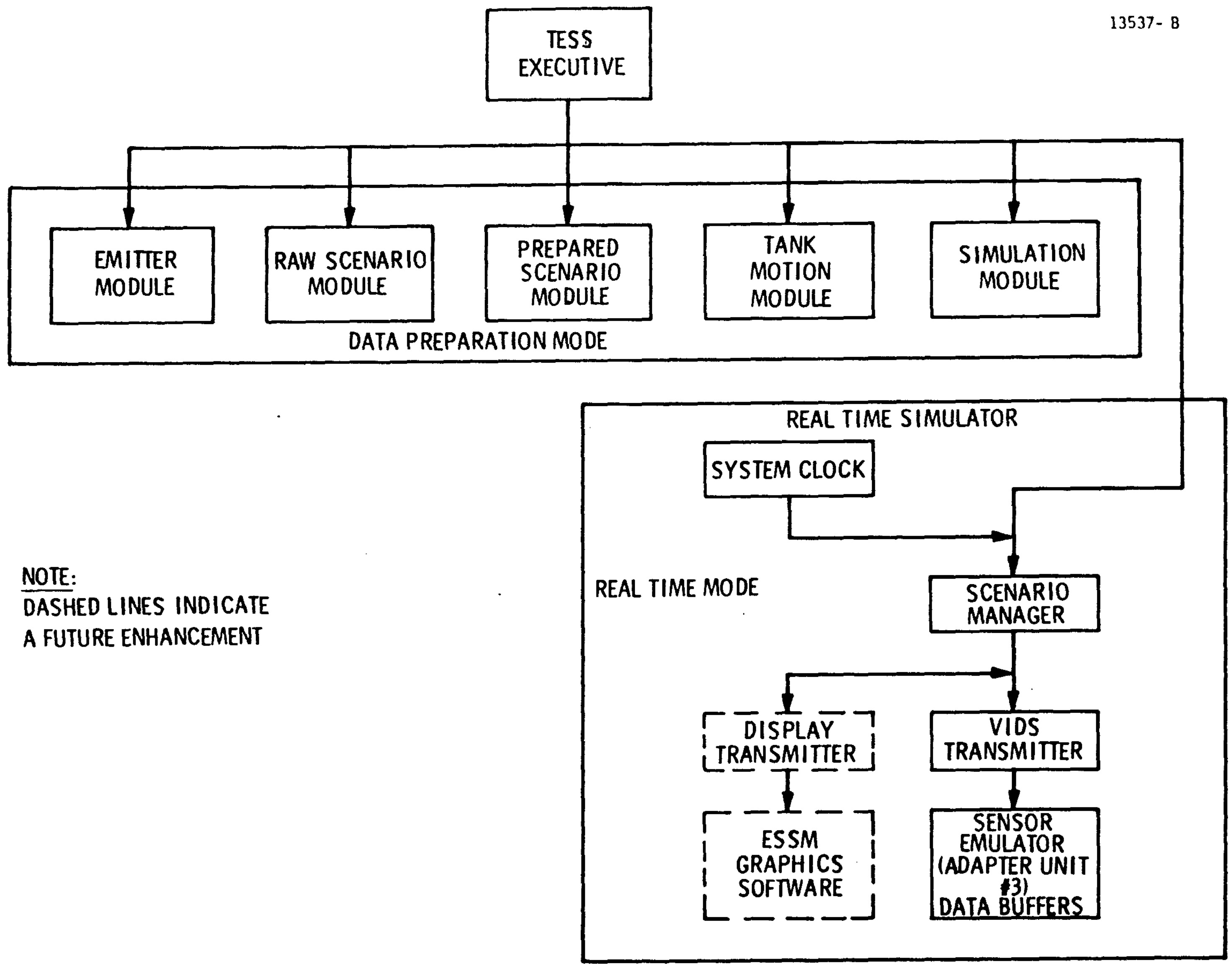
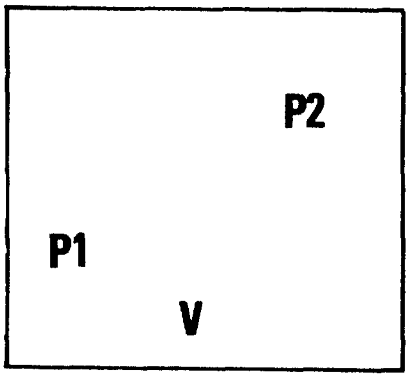


Figure 5-38. TESS Functional Structure

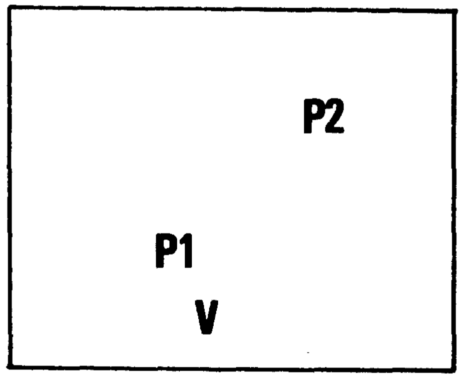
Table 5-5. Threat Report Format and Restrictions

COMPONENT	DATA RESTRICTIONS	REPRESENTATION
START	DNA	--
THREAT ID	0 TO 12	INTEGER
AZIMUTH	0° TO 360°	INTEGER
ELEVATION	-90° TO +90°	INTEGER
RANGE	0 TO 10,000 METERS	INTEGER
STOP	DNA	--



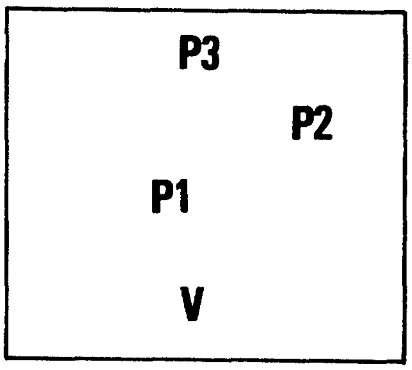
t = 250 ms

Two platforms, P1 and P2,



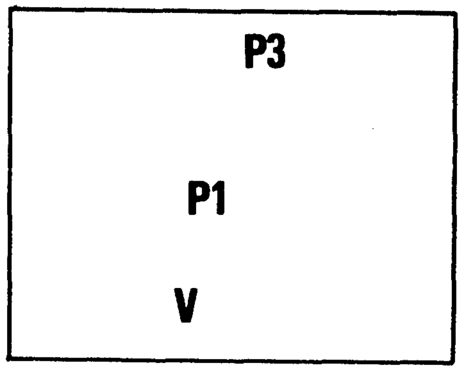
t = 500 ms

Platform P1 has moved closer



t - 750 ms

Platform P1 is closer; a third platform, P3, has appeared



t = 1000 mx

Platform P1 and P3 have moved closer. Platform P2 has been eliminated

Figure 5-39. Placement of Platforms

429-X-3

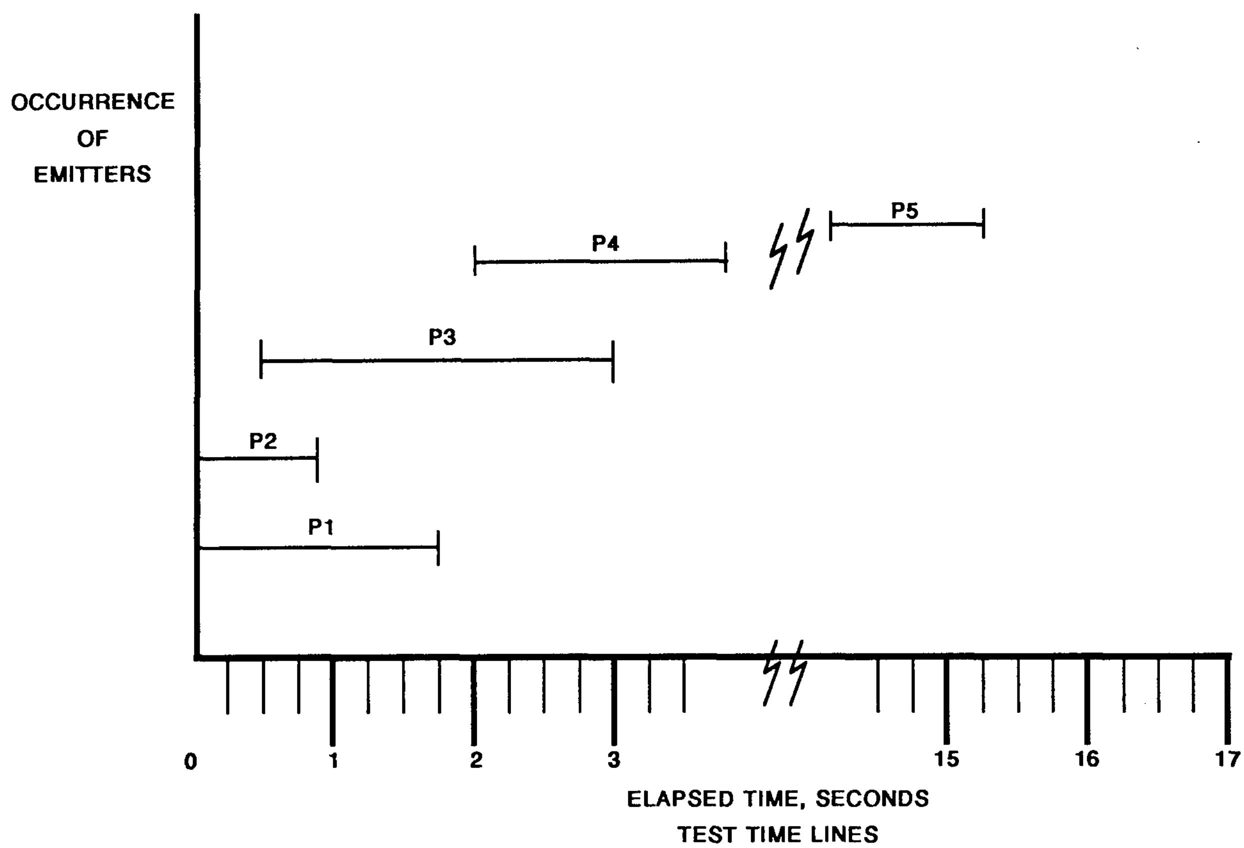


Figure 5-40. Time Line of Emitters

Data reports are continuous. There are no pauses in data transmission (unless there is no data to send). The passage of time is simulated to send reports at approximately 250-ms intervals. All data entry and file creation occurs offline in the non-real-time portion of TESS. The real-time portion of the TESS scenario manager fills a local buffer with data from the disk file. The real-time module will access the buffer to send the data transmission until the buffer is empty. At that time, the real-time module will refill the buffer and continue to process the data.

A sampling period will be approximately 250 msec. This will vary from one sampling period to the next (i.e., the first sampling period may be 255 msec; the second 250 msec). Without access to the system clock on the UNISTAR 100, sampling periods cannot be more uniform.

The data for a sampling period will be transmitted sequentially in a large block. All data groups will be output without interruption until complete. The data in the block will be sorted by sensor; i.e., all information from a given sensor will be sent together. In a single data block, there may be several threat reports that reference the same type of threat indicated in the Threat Identification table but are seen to emanate from separate platforms or individual emitters of the same type at different locations.

5.7.3. Sample Simulation. A sample data block is illustrated in Figure 5-41. In this sample there are four threats: three laser threats and one NIS threat. Threats 1 and 3 reference the same type of "emitter".

Additional Information on the overall design and operation of the TESS, including a user's manual with examples of operation, is found in the separate document on the TESS.

5.8. Demonstration/Test Procedure

This section describes the effort and results in determining the means of testing this VIDS FDM. The contract requires that a "time-phased scenario" be provided for use in the acceptance testing of the FDM. The scenario must consist of at least 25 threats, in which two or more are moving relative to one another, and at least 3 threats have data from multiple sensors. The development of this test scenario involved several comprehensive steps based on a series of pseudo-realistic engagement situations and the logic of presumed counteraction possibilities which form this expert rule-based system constituting the DMS software. This development is summarized in the following steps:

1) Threats of Counteractions: We established the eight principal sensor categories (optical, laser RF, laser design, Non-Imaging Sensor (NIS), Passive Missile Detector (PMD), chemical, radiological, and radar), and assigned a basic lethality reference (1-5). For each category, we listed all weapon systems that could possibly

Threat Type:	1	
Azimuth:	45	Threat Laser Designator
Elevation:	10	
Range:	2,000	
Threat Type:	2	Threat #2 Laser Beamrider
Azimuth:	60	
Elevation:	0	
Range:	7,000	
Threat Type:	1	Threat #3 Laser Designator
Azimuth:	30	
Elevation:	70	
Range:	8,000	
Threat Type:	5	Threat #4 NIS Attack Helicopter
Azimuth:	20	
Elevation:	15	
Range:	1,500	

Figure 5-41. Sample Simulation Data Block

be detected by this sensor and its possible correlation(s) with detection of the same weapon by other sensors. A higher lethality index was assigned in the case of correlations. Symbology representing each emitter type and available reactions were listed with the recommended decisions for counteractions. This basic data is illustrated in Table 5-6, VIDS Threat Resolution/Reaction Management Data.

2) FDM Combinations. The next step was to generate a table of the possible observables from each platform by taking one emitter at a time, then two emitters (pre-correlation) and finally three emitters from this same platform (two correlations). We assigned a display symbol (alternating a "mippling" in some cases) and assigned a preferred or recommended counteraction. The basic "threat table" was illustrated in Table 5-4, VIDS DMS Platform Identification and Reaction Analysis Chart.

3) Groundrules. An extensive listing of all assumptions for possible correlation, display conditions, and operations of counteraction devices was prepared. These groundrules were used in the coding of the data bases for the operational application packages. The list is set forth in Table 5-7, Groundrules and Assumptions.

4) Generic Situation. We next established a list of 39 threat situations for which a specified platform, means of detection (sensor type), and display symbol were listed. The exact voice alert message and the counteraction recommendation were added to complete this situation. This is illustrated in Table 5-8, Platforms/Scenario Support Data. We also listed the specific Voice Alert Messages and Counteractions in Table 5-9 and 5-10.

5) Scenario. For each of the 39 generic situations, we generated a model for the TESS by assigning a location for the vehicle (X, Y, Z) and a location for the threat(s) (X_2, Y_2, Z_2) and the relative time of sensing (t_1, t_2, t_3). A military topographic map of a typical maneuver area near Ft. Knox was used for the exact location of our vehicle at a given time and for the prediction of time of sight and therefore the specific time of sensing a threat emitter. Each of these situations was then written up to describe the engagement and a data sheet of pertinent information was printed out. An example of the description and data sheet for situation No. 1 is provided in this section. Copies of all situations programmed in the TESS scenario are attached as a separated document to this report and were presented in November as the basis for the FDM demonstration.

Table 5-6. VIDS Threat Resolution/Reaction Management

Threat Information	Possible Threat Source (Weapon System)	Possible Correlation(s)	Lethality Index (1-5)*	Control Panel Display (ID)**	Available Reactions*** Priority (1-6) Recommended Automatic Vs. Manual (A; M)	Recommended Decision
SENSOR: OPTICAL 5						
Azimuth, Angle of Sight	MBT-Gunner's Sight	Laser (RF)	3		OJ(1A); LD(2M); MWC(3A); S(4M); C(5M)	OJ; LD; MWC; S; C
	BPM-ATGM	IR (Missile)	1		OJ(1A); MTJ(2A); F(3M); MWC(4M); M(5M); S(6M)	OJ; MTJ: MCW; M
	Ground Mount-ATGM	IR (Missile)	1		OJ(1A); MTJ(2A); F(3M); MWC(4M); M(5M); S(6M)	OJ, MTJ; MWC; M
	Ground Mount-ATGM	Millimeter Wave	4		MJ(1A); OJ(2A); MWC(3A); M(4M)	MJ; OJ; MCW; M
	Ground Mount-ATGM	Laser Designator	2		OJ(1A); LD(2M); MWC(3M); M(4M)	OJ; LD; MWC; M
	Attack Helicopter	NIS, Laser (RF)	3		OJ(1A); LD(2M); C(3M)	OJ; LD; C
	Attack Helicopter	Millimeter Wave; NIS	4		OJ(1A); MJ(2A); C(3M); S(4M)	OJ; MJ; C
	Attack Helicopter	IR (Missile); NIS	1		OJ(1A); MTJ(2A); C(3M)	OJ; MTJ; C
	Attack Helicopter	NIS; (Laser Designator)	1		OJ(1A); LD(2M); M(3M); F(4M); S(5M)	OJ; LD; M
	Attack Helicopter	NIS (Acoustic Only)	3		OJ(1A); C(2M)	OJ; C
		No Correlation	5		OJ(1A); C(2M)	
SENSOR: LASER (RANGE FINDER) 4						
Azimuth, Angle of Sight	Main Battle Tank	Optic (Gunners Sight)	3		OJ(1A); LD(2M); MWC(3A); S(4M); C(5M)	OJ; LD; MWC; S; C
	Attack Helicopter	NIS	3		LD(1A); C(2M)	LD; C
		No Correlation	4		MWC(1A); C(2M); S(3M)	MWC; C; S
SENSOR: LASER (DESIGNATOR) 3						
Azimuth, Angle of Sight	Ground Mount (Target Acquisition)	IR (Missile)	1		MTJ(1A); LD(2M); S(3M); M(4M); MWC(5M)	MTJ; LD; S; M; MCW
	Ground Mount (Target Acquisition)	Optics (Acquisition)	2		OJ(1A); LD(2M); MWC(3M); M(4M); S(5M)	OJ; LD; MCW; M; S
	Ground Mount (Acquisition)	Millimeter Wave	2		MJ(1A); LD(2M); S(3M); M(4M); MWC(5M)	MJ; LD; S; M; MWC
	Vehicle Mount (BMP, etc.)	IR (Missile)	1		MTJ(1A); LD(2M); S(3M); M(4M); MWC(5M)	MTJ; LD; S; M; MWC
	Vehicle Mount (BMP, etc.)	Optics (GPS)	2		OJ(1A); LD(2M); MWC(3M); M(4M); S(5M)	OJ; LD; MWC, M; S
	Attack Helicopter	Millimeter Wave	2		LD(1M); MJ(2A); C(3M); S(4M)	LD; MJ; C; S
	Attack Helicopter	IR (Missile)	1		MTJ(1A); LD(2M); S(3M); M(4M)	MTJ; LD; S; M
	Attack Helicopter	Acoustic (NIS) Only	2		LD(1M); S(2M); M(3M)	LD; S; M
		No Correlation	3		LD(1M); S(2M); M(3M)	LD; S; M

Table 5-6. VIDS Threat Resolution/Reaction Management (Continued)

Threat Information	Possible Threat Source (Weapon System)	Possible Correlation(s)	Lethality Index (1-5)*	Control Panel Display (ID)**	Available Reactions*** Priority (1-6) Recommended Automatic Vs. Manual (A; M)	Recommended Decision
SENSOR: NON-IMAGING SENSOR (NIS) (ACOUSTIC)				5		
Azimuth, Angle of Sight, Range, Number, Type	Attack Helicopter	Millimeter Wave	4		MJ(1A); C(2M); S(3M)	MJ; C; S
	Attack Helicopter	Laser (Designator)	2		LD(1M); S(2M); M(3M)	LD; S; M
	Attack Helicopter	IR (Missile)	2	⊂	MTJ(1A); OJ(2A); M(3M)	MTJ; OJ; M
	Attack Helicopter	Optics (ATGM)	3	Δ	OJ(1A); S(2M); M(3M)	OJ; S; M
	Attack Helicopter	Laser (RF)	3	L	LD(1M); C(2M)	LD; C
		No Correlation	5		C(1M); S(2M)	C; S
SENSOR: IR-PASSIVE MISSILE DETECTOR (MISSILE PLUME)				3		
Sector of Flight	ATGM (Ground Mount)	Optics (ATGM)	1	⊂ Δ	OJ(1A); MTJ(2A); F(3M); MWC(4M); M(5M)	OJ; MTJ; F; MWC; M
	ATGM (BMP)	Optics (ATGM)	1	⊂ Δ	OJ(1A); MTJ(2A); F(3M); MWC(4M); M(5M)	OJ; MTJ; F; MWC; M
	Attack Helicopter	Millimeter Wave	2	⊂	OJ(1A); MTJ(2A); M(3M); MJ(4A); C(5M)	OJ; MTJ; M; MJ; C
	Attack Helicopter	Optics (ATGM)	1	⊂ Δ	OJ(1A); MTJ(2A); C(3M); M(4M)	OJ; MTJ; C; M
	Attack Helicopter	Acoustic (NIS)	2	⊂	MTS(1A); OJ(2A); M(3M)	MTS; OJ; M
	Attack Helicopter	Laser (Designator)	1	⊂	MTJ(1A); LD(2M); S(3M); M(4M)	MTJ; LD; S; M
	RPG (Shoulder Fired)	None	2	⊂	MTJ(1A); M(2M); MWC(3M)	MTJ; M; MWC
	Ground Mount (Laser Desig'tr)	Laser (Designator)	1	⊂	MTJ(1A); LD(2M); S(3M); M(4M); MWC(5M)	MTJ; LD; S; M; MWC
	Ground Mount (Acquisition Radar)	Millimeter Wave	3	⊂	MTJ(1A); MJ(2A); S(3M); M(4M); MWC(5M)	MTJ; MJ; S; M; MWC
		No Correlation	3	⊂	MTJ(1A); M(2M)	MTJ; M
SENSOR: CHEMICAL 1						
Chemical Detection	Chemical Munitions	N/A	1		BMB	BMB
SENSOR: RADIOLOGICAL 1						
High Radiation Dosage	Radiation Source	N/A	1		BMB	BMB

Table 5-6. VIDS Threat Resolution/Reaction Management (Continued)

Threat Information	Possible Threat Source (Weapon System)	Possible Correlation(s)	Lethality Index (1-5)*	Control Panel Display (ID)**	Available Reactions*** Priority (1-6) Recommended Automatic Vs. Manual (A; M)	Recommended Decision
SENSOR: MILLIMETER WAVE RADAR			5			
Azimuth, Angle of Sight, Type	Attack Helicopter	Optical (ATGM)	4		OJ(1A); MJ(2A); C(3M); S(4M)	OJ; MJ; C; S
	Attack Helicopter	Laser (Designator)	2		LD(1M); MJ(2A); C(3M); S(4M)	LD; MJ; C; S
	Attack Helicopter	Acoustic Only	4		MJ(1A); C(2M); S(3M)	MJ; C; S
	Attack Helicopter	IR (Missile)	2		OJ(1A); MTJ(2A); M(3M); MJ(4A); C(5M)	OJ; MTJ; M; MJ; C
	Attack Helicopter	Laser (Designator)	2		MJ(1A); LD(2M); S(3M); C(4M)	MJ; LD; S; C
	Attack Aircraft	IR (Missile)	2		MTJ(1A); MJ(2A); S(3M); C(4M)	MTJ; MJ; S; C
	Ground Station	Laser (Designator)	2		MJ(1A); LD(2M); S(3M); M(4M); MWC(5M)	MJ; LD; S; M MWC
	Ground Station	Optics (ATGM)	4		MJ(1A); OJ(2A); MWC(3M); M(4M)	MJ; OJ; MWC, M
	Ground Station	IR (Missile)	3		MTJ(1A); MJ(2A); S(3M); M(4M); MWC(5M)	MTJ; MJ; S; M; MWC
		No Correlation	5		MJ(1A); M(2M)	MJ; M

NOTES:

*Lethality Index

1. Attack Imminent
2. Attack in Progress
3. Tracking/Danger
4. Acquisition
5. Threat Presence (Searching)

**Control Panel Display

Optical:

Gnd Laser (RF):

Laser (Designator):

Acoustic (NIS):

IR(PMD):

MMW:

Laser (RF:)

Chemical: Chemical Alarm

Radiological: Radiation Alarm

***Reaction Abbreviations

OJ = Optical Jamming
LD = Laser Decoy
MTJ = Missile Tracker Jammer
MJ = Millimeter Wave Jammer
MWC = Main Weapon Counterfire
F = Flare
S = Smoke
C = Take Cover
M = Maneuver
BMB = Button, Mask, Blow

Table 5-7. VIDS FDM Reaction Management Groundrules and Assumptions

GROUNDRULES
• The FDM will react with multiple options and decisions. • Maneuver changes vehicle position to disrupt targeting. • Cover also includes concealment to elude attacker and protect vehicle. • All helicopter identifications require NIS. • Completion of reaction decision removes threats in question. • All survivability actions will be successful. • FDM software will handle up to two correlations, i.e., three different emitters from the same platform. • Selection of maneuver option includes cover and concealment. • Automatic maneuver is not implemented in the FDM. • Smoke is an immediately effective screen (assumption). • All sensings are considered "new guys" by sensors. VIDS sensors cannot track. Crew must make judgement. DMS discerns "new guys" if tracks exceed certain limits of position variation tolerance. • Screen graphics remain for five seconds after last reported emission intercept to allow for crew reaction time. • Sensors cannot detect movement. Only sequential NIS sensing can be regarded as likely movement. • Each sensing (except NIS) replies on short pulses of emitters and are of short duration. • Nuclear and chemical alarms are reacted to in same manner (audio and hybrid). • The crew can only see and hear within the capability of the VIDS sensors. • Crew reactions and countermeasures are limited to DMS recommendations in automatic mode. Alternative selections can be made in manual mode. • Ground laser rangefinder designates a Main Battle Tank (MBT). • Optical Warning (OW) (large) can be a MBT or BMP. • OW (small) is a ground-mount SAGGER, or Laser Beam Rider. • OW (large) from air with NIS is attack helo with ATGM. • Laser RF from air with NIS is attack helo. • Millimeter wave from ground is search radar (non-lethal). • Millimeter wave from air with NIS is attack helo (search or lethal). • Lasers can be differentiated as rangefinder, designator or beam-rider. • Millimeter wave countermeasure (aerosol) is activated by <u>MMW</u> button. • Millimeter Wave Jammer (optional) is automatic only. • OJ (optical jamming) is automatic only. • Operator can change sequence of counteraction by pressing alternate symbol, or pushbutton.

Table 5-8. Platform/Scenario Support Data

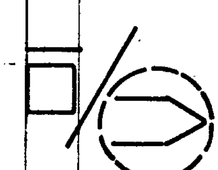
SITUATION SEQUENCE	REF.	PLATFORM	DETECTION	SYMBOL ON SCREEN
1	(7)	SCOUT HELO	NIS Only	
VOICE ALERT MESSAGE:	SCOUT, SCOUT, 2 O'CLOCK, COVER			
COUNTERACTION RECOMMENDATION:	COVER			
2A	(1)	MBT	LASER RF	
VOICE ALERT MESSAGE:	"TANK, TANK, 12 O'CLOCK, SHOOT, 12 O'CLOCK, SHOOT"			
COUNTERACTION RECOMMENDATION:	MWCF 0°			
2B	(2)	BMP	OW (LARGE)	
VOICE ALERT MESSAGE:	"BMP, BMP, 11 O'CLOCK"			
COUNTERACTION RECOMMENDATION:	AUTO OJ			
3	(17)	BMP	OW, LARGE, PLUME	
VOICE ALERT MESSAGE:	"MISSILE, MISSILE, 12 O'CLOCK, SHOOT"			
COUNTERACTION RECOMMENDATION:	MWCF 0° AUTO OJ, MTJ			
4	(12)	ATTACK HELO #1	MMW DET (+ NIS)	
VOICE ALERT MESSAGE:	"HELICOPTER, 2 O'CLOCK, CHAFF, 2 O'CLOCK, CHAFF, COVER			
COUNTERACTION RECOMMENDATION:	MMW EXPENDABLE, COVER			
5A	(22)	SCOUT HELO	NIS + OW	
VOICE ALERT MESSAGE:	"SCOUT 11 O'CLOCK, COVER			
COUNTERACTION RECOMMENDATION:	AUTO OJ, COVER			
5B	(21)	ATTACK HELO #2	NIS + PLUME	
VOICE ALERT MESSAGE:	"HELICOPTER MISSILE, 1 O'CLOCK, FLARE COVER"			
COUNTERACTION RECOMMENDATION:	FLARE, COVER			

Table 5-8. Platform/Scenario Support Data (Continued)

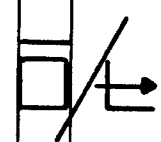
SITUATION SEQUENCE	REF.	PLATFORM	DETECTION	SYMBOL ON SCREEN
6	(11)	ATTACK ACFT	MMW DET	
VOICE ALERT MESSAGE: "AIRCRAFT 11 O'CLOCK, JAM, JAM, COVER"				
COUNTERACTION RECOMMENDATION: MMW JAM, COVER				
7	(5)	ATTACK HELO #1	NIS + LASER DESIG	
VOICE ALERT MESSAGE: "HELICOPTER 1 O'CLOCK, DECOY, 1 O'CLOCK, DECOY, COVER, COVER"				
COUNTERACTION RECOMMENDATION: LASER DECOY/COVER				
8	(16)	MBT	LASER RF, OW (LARGE)	
VOICE ALERT MESSAGE: "TANK, TANK, 1 O'CLOCK, SHOOT, 1 O'CLOCK, SHOOT"				
COUNTERACTION RECOMMENDATION: MWCF 30°, AUTO OJ				
9	(8)	LASER BEAMRIDER GROUND LAUNCHER	LASER DET	
VOICE ALERT MESSAGE: "MISSILE, MISSILE, 3 O'CLOCK, SMOKE, SMOKE, MOVE"				
COUNTERACTION RECOMMENDATION: CUE OW, SMOKE, MOVE				
10	(10)	MMW GROUND	MMW DET	
VOICE ALERT MESSAGE: "RADAR 11 O'CLOCK, SHOOT, JAM"				
COUNTERACTION RECOMMENDATION: MWCF, 330°, JAM				
11	(6)	ATTACK HELO #2	NIS Only	
VOICE ALERT MESSAGE: "HELICOPTER, 2 O'CLOCK, COVER"				
COUNTERACTION RECOMMENDATION: COVER				

Table 5-8. Platform/Scenario Support Data (Continued)

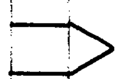
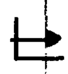
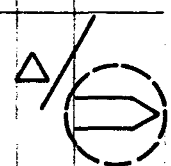
SITUATION SEQUENCE	REF.	PLATFORM	DETECTION	SYMBOL ON SCREEN
12	(28)	CHEMICAL	NBC	GAS
VOICE ALERT MESSAGE: "GAS, GAS, GAS", "GAS, GAS, GAS"				
COUNTERACTION RECOMMENDATION: HYBRID ACTIVITY				
13	(13)	NBC	NUC	NUC
VOICE ALERT MESSAGE: "NUKE, NUKE, NUKE -- NUKE, NUKE, NUKE"				
COUNTERACTION RECOMMENDATION: HYBRID ACTION				
	(3)	RPG	PLUME	
VOICE ALERT MESSAGE: "MISSILE LAUNCH ___ O'CLOCK, SHOOT, ___ O'CLOCK, SHOOT.				
COUNTERACTION RECOMMENDATION: TURN TO ___ SHOOT MG				
	(4)	SAGGER	OW (SMALL)	
VOICE ALERT MESSAGE: "OPTICS, OPTICS, ___ O'CLOCK, ___ O'CLOCK"				
COUNTERACTION RECOMMENDATION: AUTO OJ				
	(9)	LASER DESIGNATOR	LASER DET	
VOICE ALERT MESSAGE: "DESIGNATOR ___ O'CLOCK, ___ O'CLOCK, DECOY, DECOY, MOVE"				
COUNTERACTION RECOMMENDATION: LASER DECOY				
	(19)	SAGGER	OW (SMALL) + PLUME	
VOICE ALERT MESSAGE: "MISSILE, MISSILE ___ O'CLOCK, SHOOT, JAM"				
COUNTERACTION RECOMMENDATION: AUTO OJ, MWCF, MTJ				

Table 5-8. Platform/Scenario Support Data (Continued)

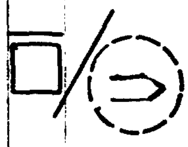
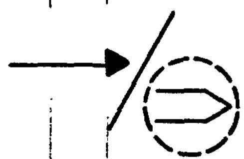
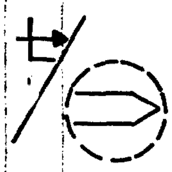
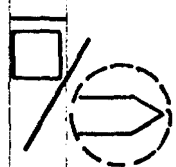
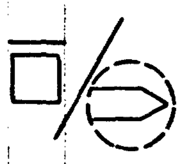
SITUATION SEQUENCE	REF.	PLATFORM	DETECTION	SYMBOL ON SCREEN
	(20)	ATTACK HELO #1	NIS + LASER + PMD	
VOICE ALERT MESSAGE:	"HELICOPTER MISSILE, ___ O'CLOCK, DECOY, COVER"			
COUNTERACTION RECOMMENDATION:	DECOY, COVER			
	(23)	LASER BEAMRIDER GROUND LAUNCH	LASER DETECT OW (SMALL)	
VOICE ALERT MESSAGE:	"LASER MISSILE ___ O'CLOCK, SMOKE"			
COUNTERACTION RECOMMENDATION:	AUTO OJ, SMOKE			
	(24)	LASER DESIGNATOR (GROUND)	LASER DETECT OW (SMALL)	
VOICE ALERT MESSAGE:	"DESIGNATOR ___ O'CLOCK, DECOY ___ O'CLOCK, SHOOT, ___ O'CLOCK, MOVE"			
COUNTERACTION RECOMMENDATION:	AUTO OJ, LASER DECOY, MWCF ___°, RANGE ___ M, MOVE			
	(35)	ATTACK HELO #1	NIS + OW (LARGE, PLUME	
VOICE ALERT MESSAGE:	"HELICOPTER MISSILE, ___ O'CLOCK, JAM, COVER"			
COUNTERACTION RECOMMENDATION:	AUTO OJ, MTJ, COVER			
	(36)	ATTACK HELO #2	NIS + MMW + PLUME	
VOICE ALERT MESSAGE:	"HELICOPTER, RADAR MISSILE ___ O'CLOCK, JAM, COVER"			
COUNTERACTION RECOMMENDATION:	MMW JAM, COVER, MMW AEROSOL			

Table 5-8. Platform/Scenario Support Data (Continued)

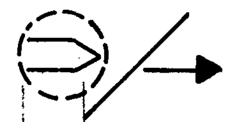
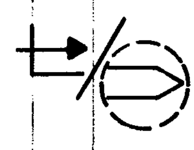
SITUATION SEQUENCE	REF.	PLATFORM	DETECTION	SYMBOL ON SCREEN
	(38)	LASER BEAMRIDER (GROUND)	LASER DET, PLUME	
VOICE ALERT MESSAGE:	"LASER MISSILE, ___ O'CLOCK, MOVE, SHOOT, MACHINE GUNS ___ O'CLOCK"			
COUNTERACTION RECOMMENDATION:	AUTO OJ, TURN ___ SHOOT MG			
	(39)	LASER DESIGNATOR (GROUND)	LASER DET, PLUME	
VOICE ALERT MESSAGE:	"LASER MISSILE, ___ O'CLOCK, DECOY, ___ O'CLOCK, MOVE"			
COUNTERACTION RECOMMENDATION:	LASER DECOY, TURN ___			

Table 5-9. Voice Alert Messages

PLATFORM	PHRASE
1.	"TANK, TANK, ___ O'CLOCK, SHOOT"
2.	"BMP, BMP, ___ O'CLOCK"
3.	"MISSILE, MISSILE, ___ O'CLOCK, MOVE, SHOOT, MACHINE GUNS"
4.	"OPTIC, OPTIC, ___ O'CLOCK
5.	"HELICOPTER, HELICOPTER, ___ O'CLOCK, DECOY, COVER, SHOOT"
6.	"HELICOPTER, HELICOPTER, ___ O'CLOCK, COVER"
7.	"SCOUT, SCOUT, ___ O'CLOCK, COVER"
8.	"MISSILE, MISSILE, ___ O'CLOCK, SMOKE, MOVE"
9.	"DESIGNATOR, DESIGNATOR, ___ O'CLOCK, DECOY, MOVE"
10.	"RADAR, RADAR, ___ O'CLOCK, JAM, SHOOT"
11.	"AIRCRAFT, AIRCRAFT, ___ O'CLOCK, JAM, COVER"
12.	"HELICOPTER, HELICOPTER, ___ O'CLOCK, CHAFF, COVER"
13.	"NUKE, NUKE"
14.	BLANK
15.	BLANK
16.	"TANK, TANK, ___ O'CLOCK, SHOOT"
17.	"MISSILE, MISSILE, ___ O'CLOCK, SHOOT, JAM"
18.	BLANK
19.	"MISSILE, MISSILE ___ O'CLOCK, SHOOT, JAM"
20.	"HELO MISSILE, HELO MISSILE, ___ O'CLOCK, DECOY, COVER, SHOOT"
21.	"HELO MISSILE, HELO MISSILE, ___ O'CLOCK, FLARE, COVER"
22.	"SCOUT, SCOUT, ___ O'CLOCK, COVER"
23.	"LARGE MISSILE, LARGE MISSILE, ___ O'CLOCK, SMOKE, MOVE"

Table 5-9. Voice Alert Messages (Continued)

PLATFORM	PHRASE
24.	"DESIGNATOR, DESIGNATOR, ___ O'CLOCK, DECOY, SHOOT, MOVE"
25.	BLANK
26.	BLANK
27.	BLANK
28.	"GAS, GAS"
29.	BLANK
30.	BLANK
31.	BLANK
32.	BLANK
33.	BLANK
34.	BLANK
35.	"HELO MISSILE, HELO MISSILE, ___ O'CLOCK, JAM, COVER"
36.	"HELO RADAR MISSILE, HELO RADAR MISSILE, ___ O'CLOCK, CHAFF, COVER"
37.	BLANK
38.	"LASER MISSILE, LASER MISSILE, ___ O'CLOCK, SHOOT MACHINE GUNS"
39.	"LASER MISSILE, LASER MISSILE, ___ O'CLOCK, DECOY, MOVE"
40	BLANK
41.	BLANK
42.	BLANK

Table 5-10. VIDS DMS FDM Counteraction Printouts

	SITUATION	PRINT
1.	(TANK)	"MWCF ___° RANGE ___M
2.	(BMP-ATGM)	"AUTO OJ"
3.	(RPG)	"TURN TO ___ SHOOT MG"
4.	(PORTABLE ATGM)	"AUTO OJ"
5.	(ATTACK HELO #1)	"LASER DECOY/COVER"
6.	(ATTACK HELO #2)	"COVER"
7.	(SCOUT HELO)	"COVER"
8.	(LASER BEAMRIDER)	"CUE OW, SMOKE, MOVE"
9.	(LASER DESIGNATOR)	"LASER DECOY, MOVE"
10.	(MMW-GROUND)	"MWCF ___° RANGE ___M, JAM"
11.	(ATTACK AIRCRAFT)	"MMW JAM, COVER"
12.	(ATTACK HELO #1)	"MMW EXPENDABLE, COVER"
13.	(NUCLEAR)	"HYBRID ACTION"
14.	--	BLANK
15.	--	BLANK
16.	(TANK)	"MWCF ___°, AUTO OJ"
17.	(BMP)	"MWCF ___°, AUTO OJ, AUTO MTJ"
18.	--	BLANK
19.	(PORTABLE ATGM)	"MWCF ___° RANGE ___M AUTO OJ, MTG"
20.	(ATTACK HELO 1)	"LASER DECOY AND COVER"
21.	(ATTACK HELO 2)	"FLARE, COVER"
22.	(SCOUT HELO)	"TURN TO ___, AUTO OJ, COVER"

Table 5-10. VIDS DMS FDM Counteraction Printouts (Continued)

	SITUATION	PRINT
23.	(LASER BEAMRIDER)	"AUTO OJ, SMOKE, TURN ___"
24.	(LASER DESIGNATOR)	"AUTO OJ, LASER DECOY, MWCF ___°, RANGE ___M, MOVE"
25.	--	BLANK
26.	--	BLANK
27.	--	BLANK
28.	(CHEMICAL)	"HYBRID ACTIVITY"
29.	--	BLANK
30.	--	BLANK
31.	--	BLANK
32.	--	BLANK
33.	--	BLANK
34.	--	BLANK
35.	(ATTACK HELO #1)	"AUTO OJ, MTJ, COVER"
36.	(ATTACK HELO #2)	"MMW JAM, COVER, MMW AEROSOL"
37.	--	BLANK
38.	(LASER BEAMRIDER)	"AUTO OJ, TURN ___, SHOOT MG"
39.	(LASER DESIGNATOR)	"LASER DECOY, TURN ___"

5.8.1. Sample Situation Description and Data Sheet. A realistic engagement scenario is based on references to an actual topographic map on which are "played" several typical situations for our own vehicle and a particular threat weapon system. A literal description of the first situation, in which a scout helicopter is detected and tracked using NIS only, is provided in the following principle. The attendant data for Situation No. 1 follows this description as Table 5-11.

Situation No. 1 (Reference Platform No. 7 - Scout Helicopter - NIS Only)

The time period of this situation will be a total of 52 seconds. During this time, the helicopter will move from a position at our 2 o'clock across the front of the tank and then, changing directions, will continue to cross to the left past our 11 o'clock. At the first sensing, the helicopter symbol will appear in the non-lethal range at our 2 o'clock and the audio alert will be "scout, scout, 2 o'clock, cover".

At approximately 16 seconds, the threat will have moved sufficiently that the symbol should move to our 1 o'clock position on the display and the audio alert will be "scout, scout, one o'clock, cover". At approximately 25 seconds, the helicopter will have moved to a position directly in front of us, at which time the symbol should move on the display to the 12 o'clock position, still in the non-lethal range and the audio alert should sound "scout, scout, 12 o'clock, cover". The helicopter changes direction slightly at approximately 28 seconds, and at time of approximately 49 seconds, it has moved into a location to our left, which would cause the symbol to move into an eleven o'clock position in the non-lethal range and the alert would be "scout, scout, eleven o'clock, cover." Shortly after first detection, the recorder should print out the word "cover."

5.8.2. Sequence of Demonstration. This section describes the functions performed by the VIDS DMS FDM in the required demonstration of the system. Other listings have been prepared for the purpose of describing pushbutton operation and the interface commands for the flat panel display. These lists, combined with the scenario on the TESS described in the preceding section and the data flow described in this section, constitute the overall description of the operation of the FDM during demonstration and test.

As an aid to the following discussion, the block diagram for the overall VIDS FDM is repeated for reference as Figure 5-42. The sequence of events is taken as they logically occur. A flow chart of functions (Figure 5-43) shows the sequential relationships. This entire process is initiated by the orchestrated scenario previously prepared and instrumented by the TESS. This scenario has been described to contain a listing of several threats to appear at various locations on a tactical map as our own tank moves on the same map. The timing of the threat emitter "sightings" is based on predetermined velocities of threat and our vehicle, and on calculated Line of Sight (LOS) conditions for each situation of the scenario.

Table 5-11. VIDS DMS Demonstration Scenario

SITUATION NO 1: VEHICLE 058,817 (M), 230 M ELEVATION
270 DEG. HULL HEADING,

* = LETHALITY
** = MANUAL/AUTO

TIME OF SENSINGS (SEC)	THREAT X, Y (OOM) (ELEV)	PLAT-FORM, TYPE	EMITTER SENSING	AUDIO ALERT(S) AND SYMBOL	*	COUNTER-ACTION DE-CISION	**	PUSH-BUTTON SYMBOL-OGY
1.0 - 3.7	030838 400 M	SCOUT HELO	NIS	SCOUT, SCOUT, 1 O'CLOCK, COVER	5			
4.0 - 6.7	030826							
7.0 - 9.7	030824	(REF. 7)						
10.0 - 12.7	030823							
13.0 - 15.7	030821			SCOUT, SCOUT, 12 O'CLOCK, COVER	5	COVER/ CONCEAL-MENT	M	MANEU-VER
16.0 - 18.7	030820							
19.0 - 21.7	030818							
22.0 - 24.7	030817							
25.0 - 27.7	030815							
28.0 - 30.7	030814							
.0 - 33.7	031812							
.0 - 36.7	032811							
37.0 - 39.7	033810			SCOUT, SCOUT, 11 O'CLOCK, COVER	5			
40.0 - 42.7	033809							
43.0 - 45.7	034807							
46.0 - 48.7	035806							
49.0 - 51.7	035804							
52.0 - AGEOUT								

NOTE: This data is used in the TESS to create Sensor Input Pockets (SIPs) to exercise the FDM in representation of this situation.

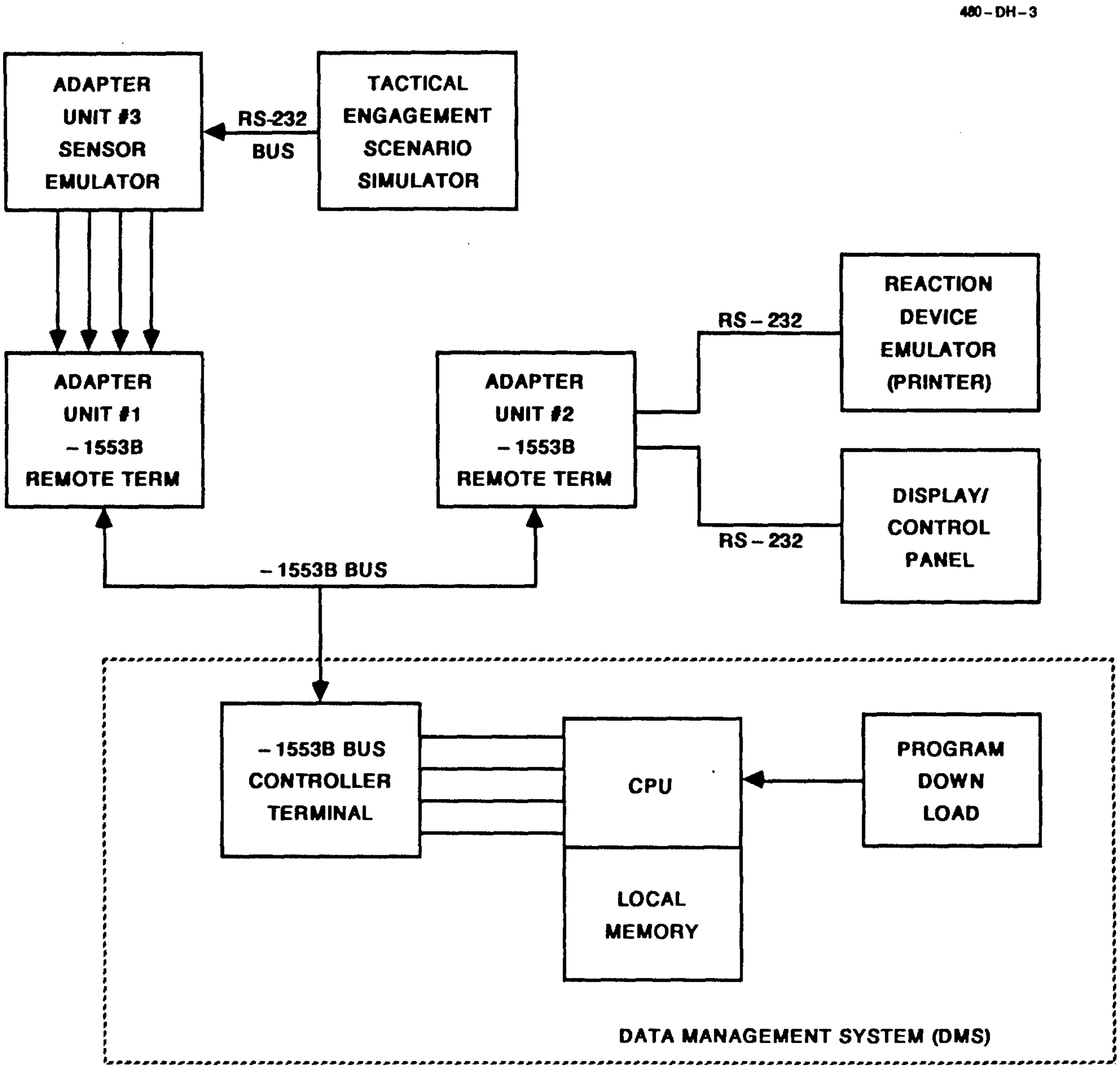


Figure 5-42. VIDS Feasibility Demonstration Model

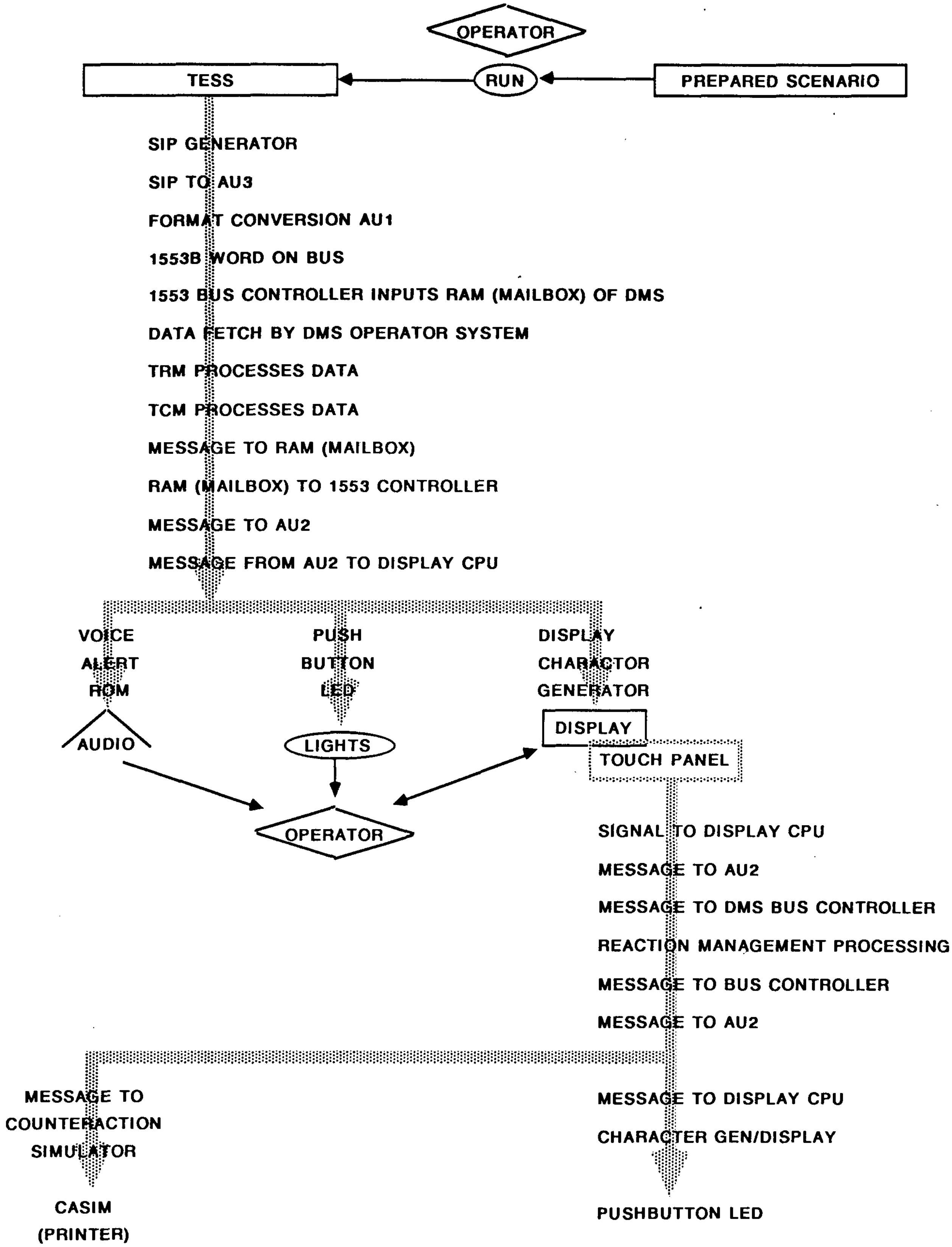


Figure 5-43. VIDS FDM Functions Flow Chart

The following listing will refer to events that occur in the VIDS DMS operation and the resulting display of its logical decisions. References in the following functions assume understanding of the performance of various subsystems that have described elsewhere. The purpose of these listings is to show the interrelationship of the functions in some sort of logical sequence as they should occur during the demonstration.

Sequence of Demonstration

1) Select scenario.

2) Run scenario on TESS. Observe operation. Modify scenario if desired by test manager or observer.

3) When the preliminaries have been completed satisfactorily, the operator will "run the scenario."

4) Let us assume that the first threat is a scout helicopter detected by NIS only. TESS will now send a message of SIPs to the sensor emulator indicating that the NIS of the VIDS should detect a scout helicopter at a certain angle. The sensor emulator interprets the SIP, converts it into sensor output at TTL levels and sends it to Adapter Unit No. 1, which indicates that the NIS has detected a scout helicopter at a certain angle and range. This angle reference is from the hull of the tank and also includes elevation and range plus type of platform. Adapter Unit No. 1 converts this message into a 1553B format and makes it available to the bus. The Bus Controller in its polling sequence calls up the information from Adapter Unit No. 1. It is transmitted over the 1553B Bus Controller to the dual-port RAM in the Bus Controller of the DMS. The DMS CPU now processes this information in its TRM and initiates a message through the display panel software module. The message is now communicated back to the Bus Controller and subsequently onto the 1553B bus with an address for the flat panel display (and voice alert) which is routed by way of Adapter Unit No. 2. Adapter Unit No. 2 converts the 1553B format into an RS-2343 (because this is a convenient, commercially available hardware interface with the CPU of the flat panel display which may eventually be a 1553B interface), and this message is thereby distributed to the CPU (8086/35) of the display unit. The message is accepted by the flat panel CPU and splits apart into two subsequent controls. One sends a message to the voice board which might say "scout at 2 o'clock" and a message also to the flat panel that puts a symbol for the helicopter upon the flat panel at approximately 2 o'clock from our own vehicle.

5) Now a Main Battle Tank (MBT) and a Personnel Carrier (BMP) are called up at two different locations in front of our tank. The MBT is detected only by the Laser Range finder and is given a high lethality rating, which places its symbol in the inside range of our flat panel display. Since our counteraction recommendation is to "SHOOT", the following sequence takes place when operating in the manual mode.*

6) The operator reaches up and touches the flat panel display by placing the tip of his finger over the symbol representing the MBT. This touch panel contact then sends a serial coded word to the 8086 CPU and back through the Adapter Unit No. 2 into the DMS. This identifies the location of that particular threat and designates the location within the DMS software (reaction management module) of that particular threat for subsequent actions. (Note that with only one threat on the panel, this designation may be unnecessary, but if there were a half-dozen or more, it would be required in order to cause the subsequent actions to take place against the proper threat.

7) The operator now pushes the pushbutton on the control panel marked with the recommended counteraction and indicated by the illumination of a yellow LED. The closure of that pushbutton sends a message back through the network to the DMS indicating to the Control Panel software module that the threat previously designated should be engaged by the recommended counteraction, in this case the main weapon. The DMS now sends the appropriate information as a series of commands through the Bus Controller, through the 1553B, to Adapter Unit No. 2. These commands are addressed for the counteraction asset simulator over the RS-232 line and the printer reads out a message to the effect that the turret is slewed 30 degrees counterclockwise, the tube is elevated 2 degrees, and the target is engaged. The balance of the fire control operation will be handled by the gunner in the conventional manner.

8) As our tank continues to move along the scenario, it encounters the next set of threats which, for example, now covers a BMP at 11 o'clock and a helicopter moving from right to left at 2 o'clock. The TESS sends commands to the sensor emulator to simulate a signal for an optic (large) at 11 o'clock and also an acoustic sensor report at 2 o'clock. These two commands are converted into TTL outputs for Adapter Unit No. 1. There they are converted into 1553B messages and communicated to the Bus Controller and its dual-port RAM. They are then picked up sequentially by the DMS CPU and analyzed by the TRM software module.

---

*If operating in the automatic mode, the reaction will take place automatically after two seconds unless the commander stops the action manually.

9) Since the optics are considered more lethal than the acoustic signature of a helicopter alone, the Threat Resolution Module (TRM) reaction logic initiates a message that goes out to Adapter Unit No. 2 and subsequently to the Control Panel CPU where the message is again split and a slice command alerts "BMP 11 o'clock" and also paints a diamond symbol for the BMP at 11 o'clock. The second message may be sent immediately following which has no voice alert (because of the priority of the BMP) but also puts up a symbol for a helicopter at 2 o'clock.

10) Now the commander pushes the symbol for the BMP designating it to the CPU as the threat upon which to take action and also pushes the pushbutton marked "main weapon counterfire." These messages are coordinated by the DMS CPU and the reaction management software module sends a message to Adapter Unit No. 2 and subsequently over the RS-2343 to the counteraction assets simulator which again rotates the turret 30 degrees to the left and hands off to the gunner for dispatch of a round against the BMP. As soon as the gunner or the commander confirms a hit, (manual, not software) they may then elect to touch the symbol for the helicopter which by now has shown by way of sensor detection that a laser designator is illuminating our tank. The commander pushes the sensitive screen over the symbol for the helicopter which communicates that message back to the DMS that is now the location of the threat of concern, and the commander then may push the button marked "Laser Decoy" which initiates a message for the DMS that is the requested counteraction.

11) The DMS interprets these two commands and sends a message to the counteraction assets simulator to initiate laser decoy action in the direction of the threat designated previously by the touch panel.

Additional scenarios can be visualized from the tables of possible combinations found at the end of this document. The exact sequence of the 39 engagement situations used in the FDM demonstration is found in its referenced document.

LIST OF ABBREVIATIONS, ACRONYMS, AND SYMBOLS

ARTOS	Ada Run Time Operating System
ASCII	American Standard Code for Information Interchange
BIM	Bus Input Module
BMP	A Russian Personnel Carrier
BMS	Battle Management System
BOM	Bus Output Module
CLI	Command Language Interpreter
CPU	Central Processing Unit
CRT	Cathode Ray Tube
DMA	Direct Memory Access
DMS	Data Management System
EDDM	External Device Data Manager
EDM	External Device Manager
EEROM	Electronically Erasable Read Only Memory
EL	Electro Luminescent
EOB	Electronic Order of Battle
EPROM	Erasable Programmable Read Only Memory
ESSM	Enhanced Scenario Simulator Module
ETAS	Elevated Target Acquisition System
FAADS	Forward Area Air Defense System
FDM	Feasibility Demonstration Model
FIFO	First In/First Out
FSED	Full-Scale Engineering Development

LIST OF ABBREVIATIONS, ACRONYMS, AND SYMBOLS (Continued)

GFE	Government-Furnished Equipment
HEX	Hexidecimal
HOL	Higher Order Language
I/O	Input/Output
LED	Light Emitting Diode
LOS	Line of Sight
LPC	Linear Predictive Coding
LRU	Line Replaceable Unit
MBT	Main Battle Tank
MMW	Millimeter Wave
MS	Military Standard (or MILSPEC)
MWCF	Main Weapon Counter Fire
NBC	Nuclear, Biological, Chemical
NIS	Non-Imaging Sensor
OJ	Optical Jamming
OS	Operating System
OW	Optical Warning
PAL	Progammable Array Logic
PDL	Program Design Language
PDW	Pulse Descriptor Word
PMD	Passive Missile Detector

LIST OF ABBREVIATIONS, ACRONYMS, AND SYMBOLS (Continued)

PROM	Programmable Read Only Memory
PTL	Priority Threat List
PWB	Printed Wiring Board
RAM	Read/Write Random Access Memory
ROM	Read Only Memory
RTU	Remote Terminal Unit
SBC	Single-Board Computer
SIP	Sensor Input Packet
SMI	Soldier-Machine Interface
SPDT	Single Pole Double Throw
SUN	Stanford University Network (CPU board design)
SYS-SERV	System Serve
TAM	Track And Map
TCF	Threat Correlation File
TCM	Threat Counteraction Module
TESS	Tactical Engagement Scenario Simulator
TFT-EL	Thin Film Transistor-Electro Luminescent
TRM	Threat Resolution Module
TTF	Threat Track File
TTL	Transistor-Transistor Logic
UART	Universal Asychronous Receiver Transmitter
VIDS	Vehicle Integrated Defense System
VISTA	Very Intelligent Surveillance Target Acquisition

THIS PAGE INTENTIONALLY LEFT BLANK

DISTRIBUTION LIST

	COPIES
Commander U.S. Army Tank-Automotive Command ATTN: AMSTA - ZSC Warren, MI 48397-5000	20
Commander U.S. Army Tank-Automotive Command ATTN: AMSTA - RSC Warren, MI 48397-5000	2
Commander U.S. Army Tank-Automotive Command ATTN: AMSTA - TSL Warren, MI 48397-5000	14
Commander U.S. Army Tank-Automotive Command ATTN: AMSTA - TSE Warren, MI 48397-5000	1