

NASA Technical Memorandum 4327

A Knowledge Based Application  
of the Extended Aircraft  
Interrogation and Display System

Richard D. Glover and Richard R. Larson

OCTOBER 1991

**NASA**

NASA Technical Memorandum 4327

A Knowledge Based Application  
of the Extended Aircraft  
Interrogation and Display System

Richard D. Glover and Richard R. Larson  
*Dryden Flight Research Facility*  
*Edwards, California*



National Aeronautics and  
Space Administration  
Office of Management  
Scientific and Technical  
Information Program

1991

# CONTENTS

<b>ABSTRACT</b>	<b>1</b>
<b>NOMENCLATURE</b>	<b>1</b>
<b>INTRODUCTION</b>	<b>2</b>
<b>REQUIREMENTS</b>	<b>3</b>
<b>HARDWARE DESCRIPTION</b>	<b>4</b>
Generic Hardware Description . . . . .	4
F-18 HARV Hardware Description . . . . .	5
<b>SOFTWARE DESCRIPTION</b>	<b>13</b>
Generic Software Overview . . . . .	13
F-18 HARV Extended Aircraft Interrogation and Display System Software Overview . . . . .	13
Rules Source Syntax Description . . . . .	15
FETCH . . . . .	16
EVENT . . . . .	16
ARITH . . . . .	16
COMP . . . . .	17
BOOL . . . . .	17
MSG . . . . .	17
PTR . . . . .	18
Parser . . . . .	18
Rule Base Page . . . . .	19
F-18 HARV Unique Messages Example . . . . .	20
Symbol Table . . . . .	22
Extended Aircraft Interrogation and Display System Parser and Display Page Verification Testing . . . . .	23
<b>F-18 TEST RESULTS</b>	<b>24</b>
<b>FUTURE PLANS</b>	<b>24</b>
<b>CONCLUDING REMARKS</b>	<b>26</b>
<b>REFERENCES</b>	<b>26</b>
<b>APPENDIX A—PARSER ERROR MESSAGES</b>	<b>27</b>
<b>APPENDIX B—DATA RULES SOURCE FILE</b>	<b>29</b>
<b>APPENDIX C—FMET TEST RESULTS FOR CHANNEL 4 AND YAW RATE 2 FAIL</b>	<b>31</b>

## LIST OF FIGURES

Figure 1. F-18 HARV. . . . .	3
Figure 2. XAIDS block diagram. . . . .	5
Figure 3. F-18 XAIDS unit. . . . .	6
Figure 4. F-18 XAIDS master cardcage. . . . .	7
Figure 5. F-18 XAIDS master cardcage complement. . . . .	8
Figure 6. F-18 XAIDS slave cardcage. . . . .	9
Figure 7. F-18 XAIDS slave cardcage complement. . . . .	10
Figure 8. Aircraft interface connection panel. . . . .	10
Figure 9. F-18 XAIDS mapping. . . . .	12
Figure 10. Rule based data acquisition overview. . . . .	15
Figure 11. XAIDS bus connection. . . . .	15
Figure 12. Parser menu. . . . .	19
Figure 13. Parser expansion summary. . . . .	20
Figure 14. Rule base page display. . . . .	21
Figure 15. Symbol table page. . . . .	22
Figure 16. XAIDS status and control panel. . . . .	23

## LIST OF TABLES

Table 1. F-18 HARV XAIDS board complement. . . . .	11
Table 2. Edit options for items 7, 8, and 9. . . . .	23

## ABSTRACT

A family of multiple-processor ground support test equipment developed at the NASA Dryden Flight Research Facility has been used to test digital flight-control systems on high-performance research aircraft. A unit recently built for the F-18 high alpha research vehicle project is the latest model in a series called the extended aircraft interrogation and display system. The primary feature emphasized in this report monitors the aircraft MIL-STD-1553B data busses and provides real-time engineering units displays of flight-control parameters. A customized software package was developed to provide real-time data interpretation based on rules embodied in a highly structured knowledge database. This report describes the configuration of this extended aircraft interrogation and display system briefly, and the evolution of the rule based package and its application to failure modes and effects testing on the F-18 high alpha research vehicle.

## NOMENCLATURE

A/D	analog-to-digital
ADC	airdata computer
AFTI	advanced fighter technology integration
CAST	computer aided systems testing
CCU	computer control unit
CENPRO	central processor
COMM1	communications radio #1
COMM2	communications radio #2
CSC	communication system control
DAC	digital to analog converter
DDI	digital display indicator
DFBW	digital fly-by-wire
DFRF	Dryden Flight Research Facility, Edwards, CA
DMA	direct memory access
EU	engineering units
FCCA	flight control computer A (channels 1, 2)
FCCB	flight control computer B (channels 3, 4)
FF	free-form display page on XAIDS
FMET	failure modes and effects tests
GPIB	general purpose interface bus
GSE	ground support equipment
HARV	High Alpha Research Vehicle
HEX	hexidecimal based number system
HiMAT	Highly Maneuverable Aircraft Technology
INS	inertial navigation set
I/O	input/output

ITF	Integrated Test Facility
Kbyte	kilobyte
LAN	local area network
LSB	least significant bit
Mbyte	megabyte
MAINT	maintenance processor
MSB	most significant bit
MSDR	maintenance signal data recorder
MSS	mass storage subsystem
NDP	numeric data processor
OS	operating system
PERPRO	peripheral processor
PL/M	programming language microprocessor
PROM	programmable read only memory
RAM	random access memory
RAV	remotely augmented vehicle
RB	rule base
RDAS	remote data acquisition subsystems
RT	remote terminal
RTPRO	real-time processor
SCP	status and control panel
TTL	transistor-transistor logic
TVCS	thrust vectoring control system
XAIDS	extended aircraft interrogation and display system

## INTRODUCTION

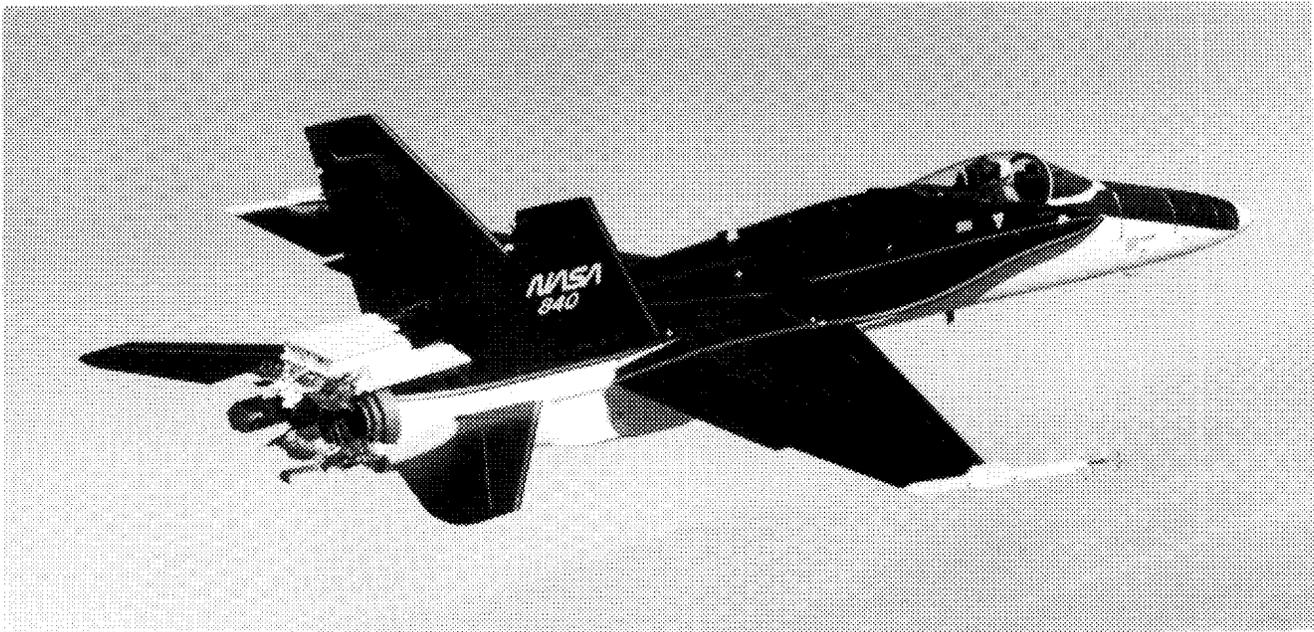
The NASA Dryden Flight Research Facility at Edwards AFB, California, has developed in-house a family of multiple-processor ground support equipment (GSE) used in testing digital flight-control systems on high performance research aircraft (refs. 1–3). A unit built for the F-18 high alpha research vehicle (HARV) project (fig. 1) is the latest model of a series called the extended aircraft interrogation and display system (XAIDS). This GSE monitors the aircraft MIL-STD-1553B data busses and provides real-time displays in engineering units (EU) of flight-control parameters.

At Dryden there is new direction to reduce the time to qualify flight-critical systems by applying automated testing techniques (ref. 4). As part of this initiative it was decided to develop automated data analysis techniques using the XAIDS as the prototype system. The advantages of this type of approach are obvious. Collecting large amounts of data followed by post processing and finally manual analysis is time consuming and unnecessary for certain testing or monitoring environments. Real-time displays containing auto-analyzed data results are all that is required. The previous steps are eliminated.

In this approach the analysis is automated by processing the raw data in real time using a knowledge base containing rules. These rules must be satisfied to trigger a message containing information relative to the event.

The rule knowledge representations are normally constructed by IF-THEN statements. For this application the rule structure was adopted in the form of Boolean expressions which could be nested to provide any level of dependency requirements. Ideally, this database would be constructed by experts who would apply their knowledge of the various parts of the system. For example, a real-time auto-analysis application could be applied to data relative to systems such as data management, redundancy management, executive, periodic built-in-test, flight controls, hydraulics, actuators, engines, airdata, structural loads, and various processors or remote terminal health status. The knowledge of these experts is now made available to any user of this system. The result would be a creative, instant data interpretation and analysis which could be readily supported by few personnel with no detailed knowledge of the system. Another application would be the routine maintenance of a system by technician-level personnel. Sound engineering judgments could be enhanced based on the data presented by the rule based (RB) system. The RB software package was written in programming language microprocessor (PL/M) within the existing XAIDS 8086 based multitasking system and runs in real time.

This paper describes the RB data architecture that was developed for the F-18 HARV program. An excerpt from that database is presented to show the kinds of real-time auto-analysis logic that can be developed. Actual testing experience using this system for the F-18 HARV failure modes and effects tests (FMET) is presented.



EC91 0028-009

Figure 1. F-18 HARV.

## REQUIREMENTS

The goal of the XAIDS family of general purpose GSE has been to provide an aircraft control system research tool which could support many projects with minimum hardware and software reconfiguration. The approach was to provide a baseline host system to which application-dependent hardware and software could be added to meet specific needs. The design objectives included mobility, configurable input/output (I/O), a common core of generic support software, and user-oriented displays.

The F-18 HARV project listed I/O hardware requirements for the XAIDS system to support planned aircraft testing and to provide for future capability. The following aircraft I/O categories were specified

- Maintenance support channels for two AN/AYK-14(V) computers,
- monitoring of three sets of dually redundant MIL-1553B busses,
- bus controller for one general purpose interface bus (GPIB) data bus (IEEE-488),
- analog channels
  - 12 each  $\pm 10.0$  volt outputs
  - 16 each  $\pm 10.0$  volt inputs, and
- discretes channels
  - 16 each 28-volt outputs
  - 16 each 28-volt inputs
  - 32 each transistor-transistor (TTL) outputs
  - 40 each TTL inputs

The software requirements for the first category above stated that the XAIDS must provide an emulation of the computer control unit (CCU). The CCU is GSE which provides hardware testing and software debugging support for the AN/AYK-14(V) processor, two of which serve as mission computers on F-18 aircraft.

The software requirements for the second category above were to monitor bus traffic and display selected data items in engineering units. Production F-18 aircraft have three sets of 1553B busses and the XAIDS was configured for this interface. However, the HARV was subsequently modified so that only two busses are now active. After the XAIDS was placed in service, a significant additional requirement was levied that the XAIDS provide knowledge-based interpretation of bus traffic and to display flight-control system status in plain English message form. Minor additional requirements were added to provide several customized raw input data type handlers and screen display formats. Software requirements for the remaining three categories of I/O were deferred until test plans evolved for their use.

## HARDWARE DESCRIPTION

### Generic Hardware Description

The current baseline XAIDS hardware complement shown in figure 2 has changed little from that of its predecessor built for the X-29A project (ref. 3). The four main processors are the real-time processor (RTPRO), the central processor (CENPRO), the peripheral processor (PERPRO), and the maintenance processor (MAINT). The RTPRO performs front-end processing of data from the user I/O channels, interfaces to the status and control panel (SCP), manages 2 analog output modules (AOM) generating 16 recorder signals, and has an RS-232 serial I/O port. The CENPRO hosts the Intel RMX86<sup>®</sup> operating system (which includes a debugger), handles mass storage I/O, and executes applications software modules (called I/O jobs). The PERPRO provides high-speed servicing of the operator's terminal and generates CRT screen snapshots for printer hardcopy. The MAINT is a stand-alone processor used for hardware checkout. These four processors share an IEEE-796 multibus with the user's I/O channel boards and with the system I/O channels. The XAIDS is designed to be compatible with any IEEE-796 multibus board having 20-bit memory address decoding and 16-bit I/O address decoding. This provides a one megabyte memory space and a 64-Kbyte I/O space.

Two product improvement changes have been incorporated into the baseline hardware suite. A 9-track 0.5-in. tape deck was added which can accept up to 14-in. diameter reels. In addition, an integrated mass storage subsystem (MSS) now provides two 5.25-in. floppy diskette drives, two 25-Mbyte hard disk drives, and a 0.25-in. streaming

---

<sup>®</sup>RMX86 is a registered trademark of Intel Corporation, Santa Clara, CA.

tape cartridge drive. One baseline item which was deleted from the F-18 HARV XAIDS was the local area network (LAN) controller board which can interface to one or more remote data acquisition subsystems (RDAS) by way of a 1.0-MHz serial data bus.

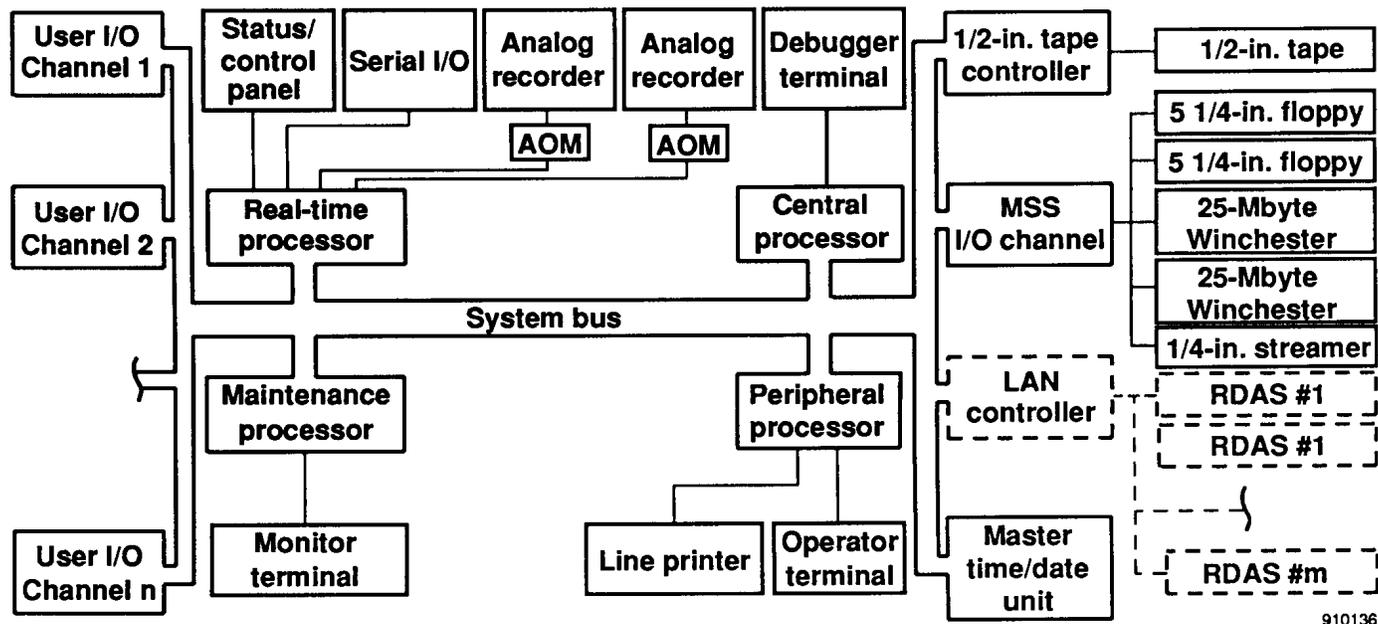


Figure 2. XAIDS block diagram.

## F-18 HARV Hardware Description

The boards selected to meet the F-18 HARV I/O requirements are commercially available except the CCU emulator interface boards. These boards were constructed at Dryden from a design provided by the Naval Air Test Center at Patuxent River, Maryland. The Navy had previously developed a CCU emulator hosted on a VAX<sup>®</sup> computer and offered to allow NASA to use their circuit design and PASCAL software. The implementation of the XAIDS CCU emulator hardware and software is a topic reserved for a later report.

The F-18 XAIDS unit is shown in figure 3. Since the number of user I/O channels for the F-18 HARV XAIDS is large, a pair of 21-slot cardcages was necessary. The cardcage shown in figure 4 is the master cardcage and hosts the board complement shown in figure 5. The cardcage shown in figure 6 is the slave cardcage and hosts the board complement shown in figure 7. The two cardcages are linked by ribbon cables connected to a pair of multibus repeater boards containing the bus interface signal drive circuitry. The aircraft interface connector panel is shown in figure 8.

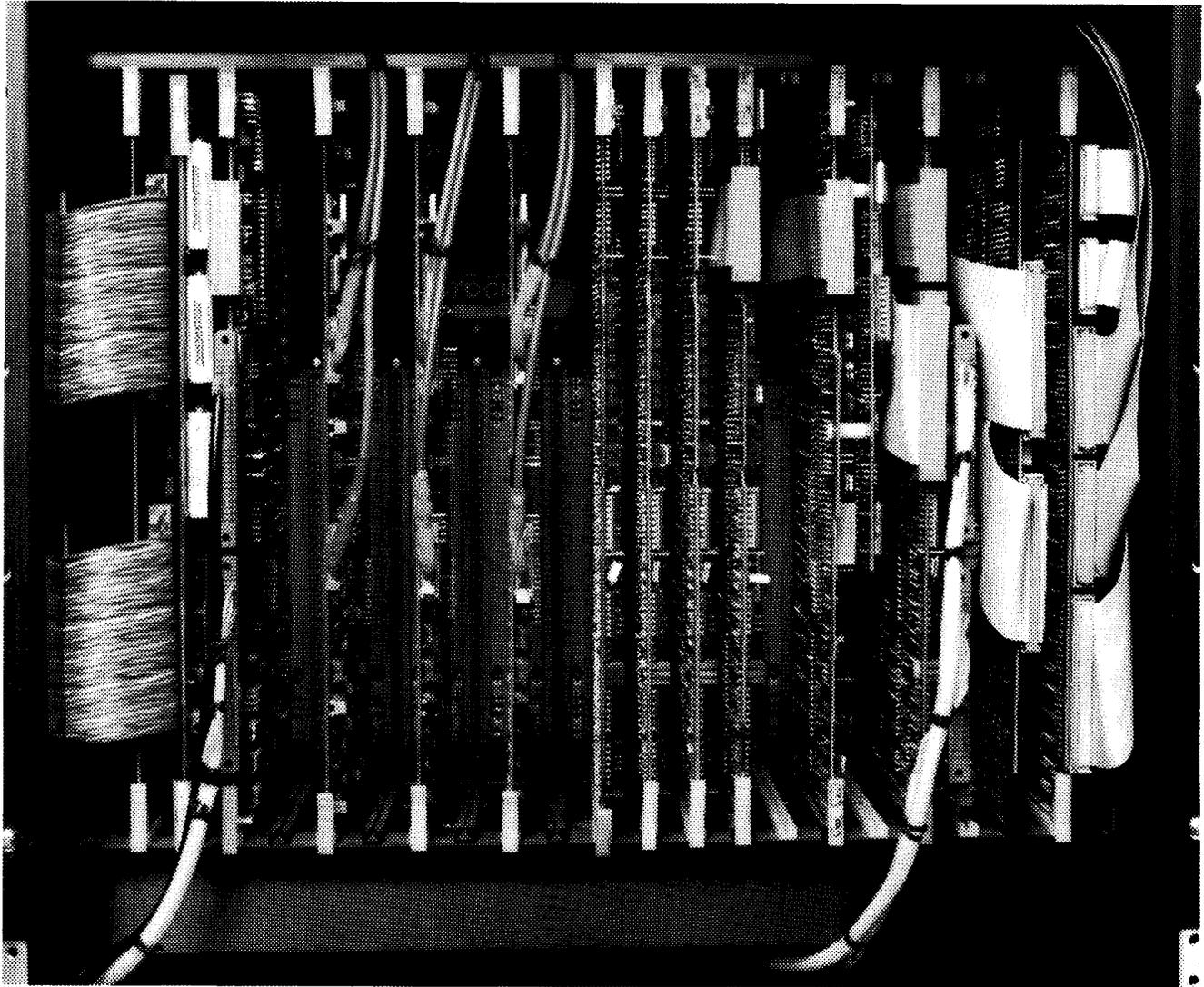
The complete board complement for the F-18 HARV XAIDS is listed in table 1 and the resulting integrated memory and I/O maps are shown in figure 9. Some boards have onboard random access memory (RAM) and control registers (REGS). Each of the three MIL-1553B channels, consisting of a bus interface board and a companion I/O engine board, requires 16 Kbytes of memory address space. An additional 24-Kbyte block for each channel was saved by exposing only the top 8 Kbytes of the I/O engine RAM to view from the multibus. This technique is commonplace in multibus systems and is useful for conserving scarce address space—the savings in this case are 72 Kbytes. The IEEE-488 controller RAM, A/D converter registers, and CCU I/O board registers are also mapped into memory address space, while the remaining user board registers are mapped into the I/O address space as shown (fig. 9).

<sup>®</sup>VAX is a registered trademark of Digital Equipment Corporation, Maynard, MA.



EC90 168-1

Figure 3. F-18 XAIDS unit.



EC90 168-2

Figure 4. F-18 XAIDS master cardcage.

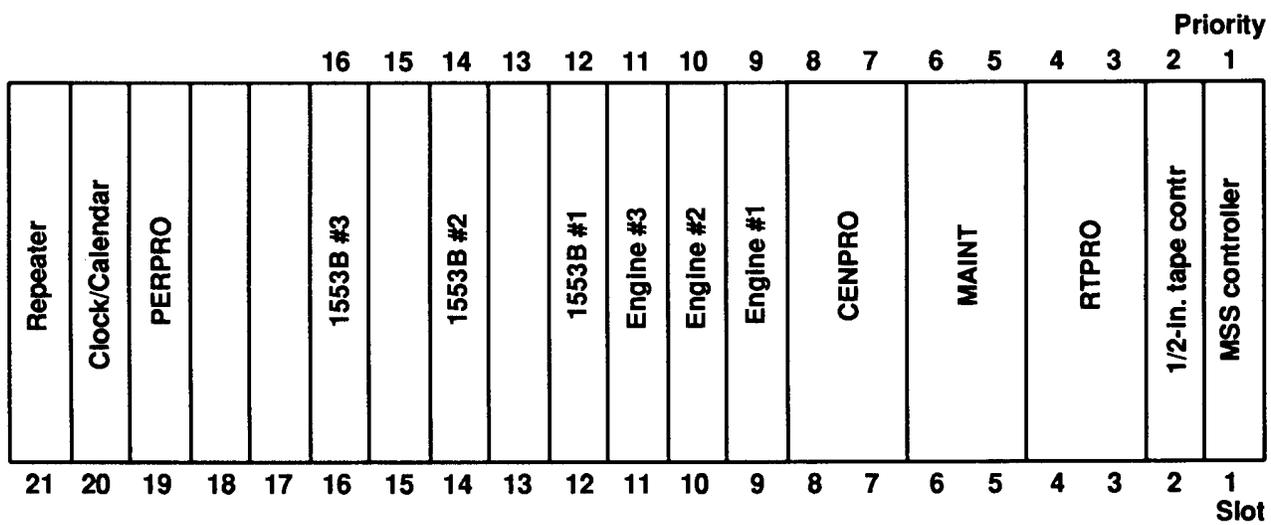


Figure 5. F-18 XAIDS master cardcage complement.

910137



EC90 168-3

Figure 6. F-18 XAIDS slave cardcage.

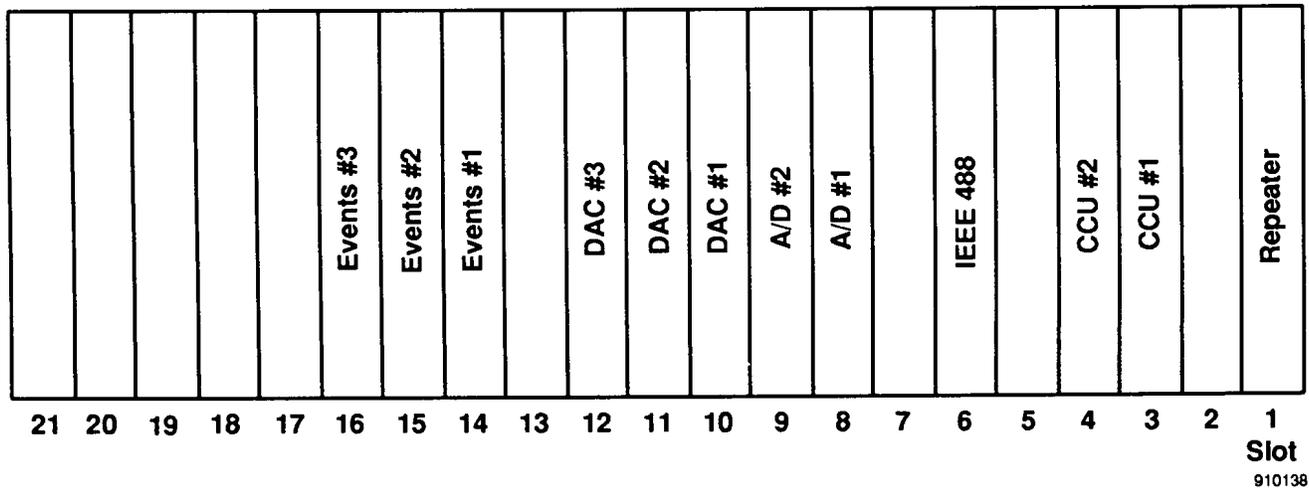
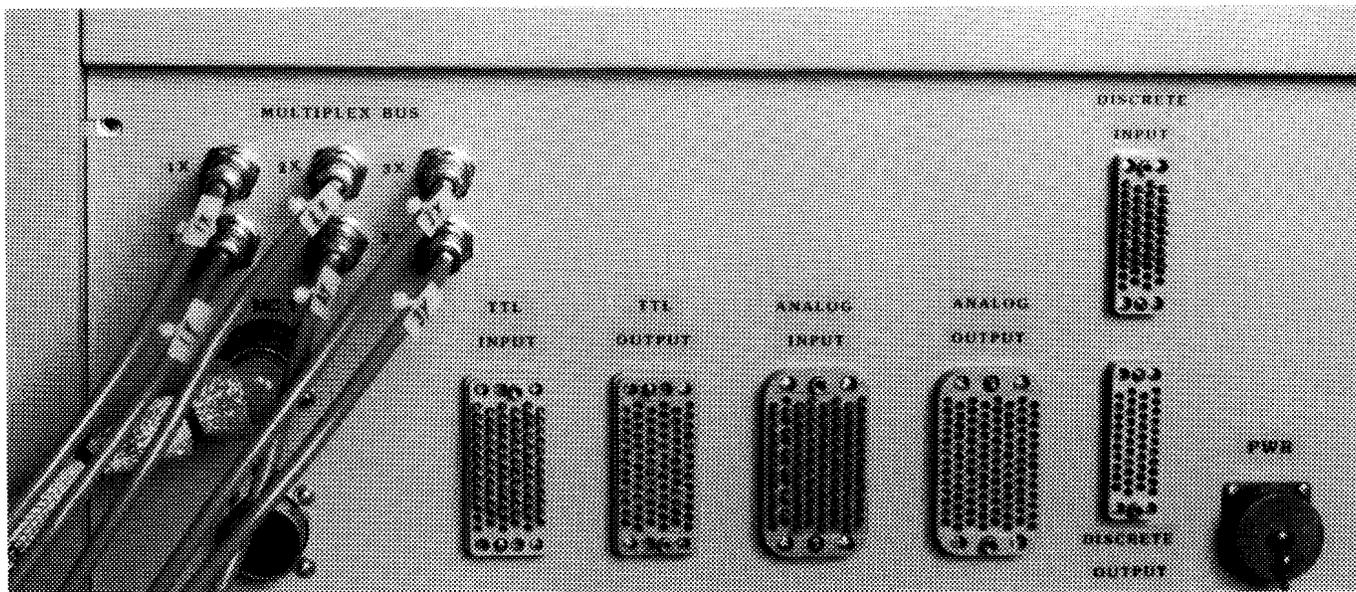


Figure 7. F-18 XAIDS slave cardcage complement.

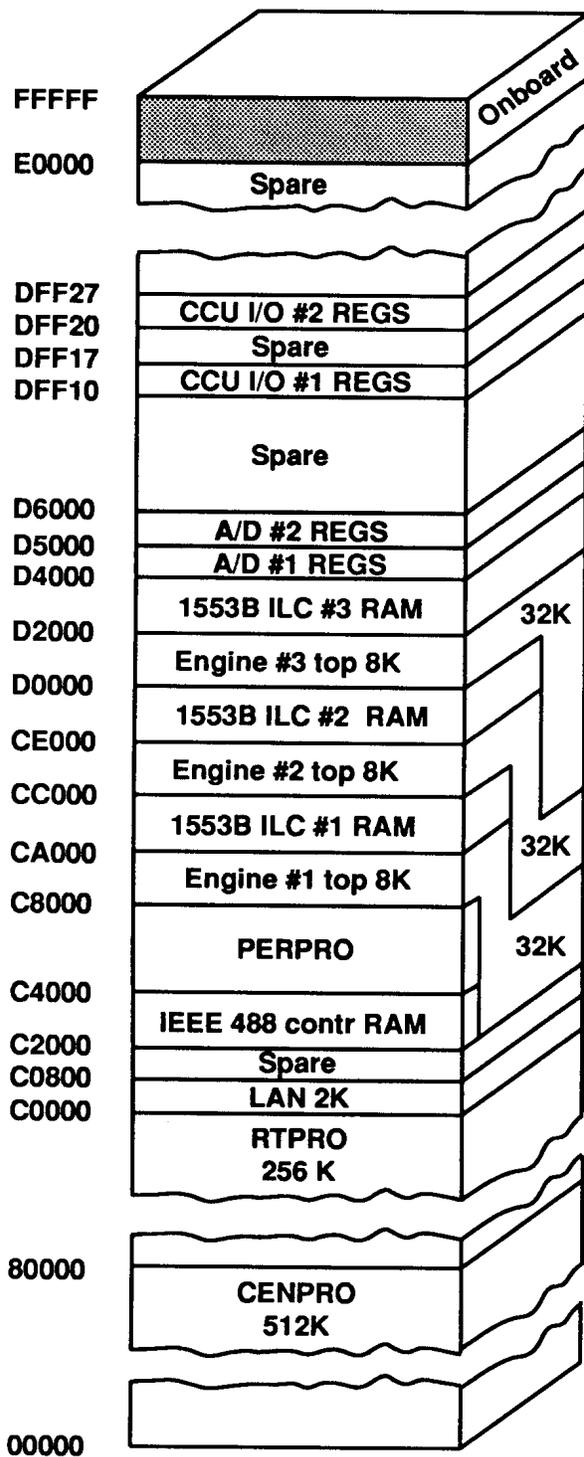


EC90 168-4

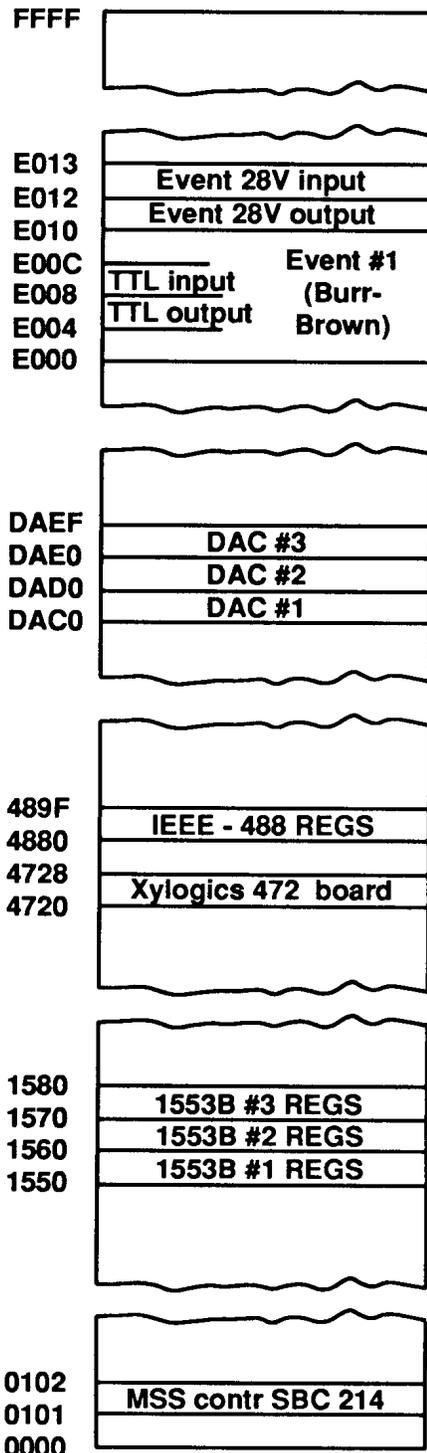
Figure 8. Aircraft interface connection panel.

Table 1. F-18 HARV XAIDS board complement.

Slot	Assignment	Board type	Addon modules	Manufacturer
M1	MSS controller	iSBC214		Intel
M2	Tape controller	Model 742		Xylogics
M3,4	RTPRO	iSBC86/30	iSBX337A NDP iSBX328 AOM # 1 iSBX328 AOM # 2	Intel
M5,6	MAINT	iSBC86/05	iSBX251 MBM	Intel
M7,8	CENPRO	iSBC86/35		Intel
M9	*Engine #1 IOPA	iSBC86/14		Intel
M10	*Engine #2 IOPB	iSBC 86/14		Intel
M11	*Engine #3 IOPC	iSBC 86/14		Intel
M12	*1553B #1	Bus 65509		ILC-DDC
M14	*1553B #2	Bus 65509		ILC-DDC
M16	*1553B #3	Bus 65509		ILC-DDC
M19	PERPRO	iSBC80/30		Intel
M20	Timing control	TCU-410		Digital Pathways
M21	796 Bus repeater	MBH-660		Procise Corp
S1	796 Bus repeater	MBE-660		Procise Corp
S3	*CCU emulator #1	custom		NASA
S4	*CCU emulator #2	custom		NASA
S6	*IEEE 488 I/O	ZT-85/38		Ziatech
S8	*A/D converter #1	ST701		Datel
S9	*A/D converter #2	ST701		Datel
S10	*DAC #1	DT728		Data Translation
S11	*DAC #2	DT728		Data Translation
S12	*DAC #3	DT728		Data Translation
S14	*Events #1 TTL	MP830		Burr-Brown
S15	*Events #2 28v out	1001		Anitech
S16	*Events #3 28v in	1001		Anitech



Memory mapping



Note: XXF0-XXF7 for time/date unit  
I/O mapping

910139

Figure 9. F-18 XAIDS mapping.

# SOFTWARE DESCRIPTION

## Generic Software Overview

As described in reference 3, the software suite for a typical XAIDS consists of three types of components; an operating system (OS), XAIDS generic software, and XAIDS user application software. In general, all three exist as a combination of firmware in programmable read-only memory (PROM) and linked modules loaded from mass storage into RAM. Developing a user application requires optimizing strategies for I/O data management, designing operator displays, and writing custom data handlers and formatters as required. Since the MAINT processor (described in ref. 5) may be removed from the XAIDS and not affect its operation, MAINT firmware is not considered part of the XAIDS software suite.

The XAIDS OS is based upon the Intel RMX86 real-time executive. It is configured for the environment in which it executes: IEEE-796 system bus, CENPRO host (8086 CPU, 8087 NDP), MSS mass storage, 0.5-in. tape drive controller, and PERPRO interface for the operator's terminal and line printer. As much of the OS as possible was programmed into PROM (128 Kbytes total), the remainder is booted from hard disk. The OS provides the operator with a command line interpreter called the human interface with which a wide range of jobs may be invoked. The PERPRO firmware was designed at Dryden to provide high-speed terminal servicing and as such may be considered part of the OS.

Generic XAIDS software consists of an extensive library of modules for the RTPRO and the CENPRO. The baseline RTPRO software provides a data acquisition executive, a set of raw data type handlers, channel handlers for memory mapped RAM, SCP I/O, RS-232C port I/O, and analog output for 16 recorder channels. Within CENPRO, the XAIDS software components include RTPRO interface routines, a set of basic screen displays, a display librarian, a set of data formatting modules, a symbol table manager, and assorted utilities.

A typical user's application software for the RTPRO consists of customized channel handlers for the I/O channel boards and special handlers for unique raw data types. For the CENPRO, the user's software usually consists of customized operator displays and handlers as required for unique display formats. The user's software is linked to the baseline XAIDS components to produce an RTPRO program and a CENPRO job (usually called XAIDS). In a typical application, these two programs work together to fetch data from the user's I/O channel hardware and display it on the operator's CRT screen in real time.

## F-18 HARV Extended Aircraft Interrogation and Display System Software Overview

For this application, these additional components were required:

1. Program for 1553B I/O engines providing
  - Bus interface board management
  - Toggle buffering of specified messages
  - End of frame synchronization logic
2. Custom modules linked to RTPRO program
  - Channel handlers for the I/O engine channels
  - Channel handlers for SCP switches and thumbwheels
  - Raw data type handlers for packed sign and magnitude
3. Stand-alone CENPRO job called parser
  - Knowledge base rules source file
  - Run-time file for RB display page

4. Custom modules linked to CENPRO XAIDS job
  - RB display page create–run package
  - Custom symbol table manager for 1553B bus parameters
  - Custom display formats for unpacking single bits
5. Stand-alone CENPRO job called CCU
  - Computer control unit emulator for the AN/AYK-14(V)

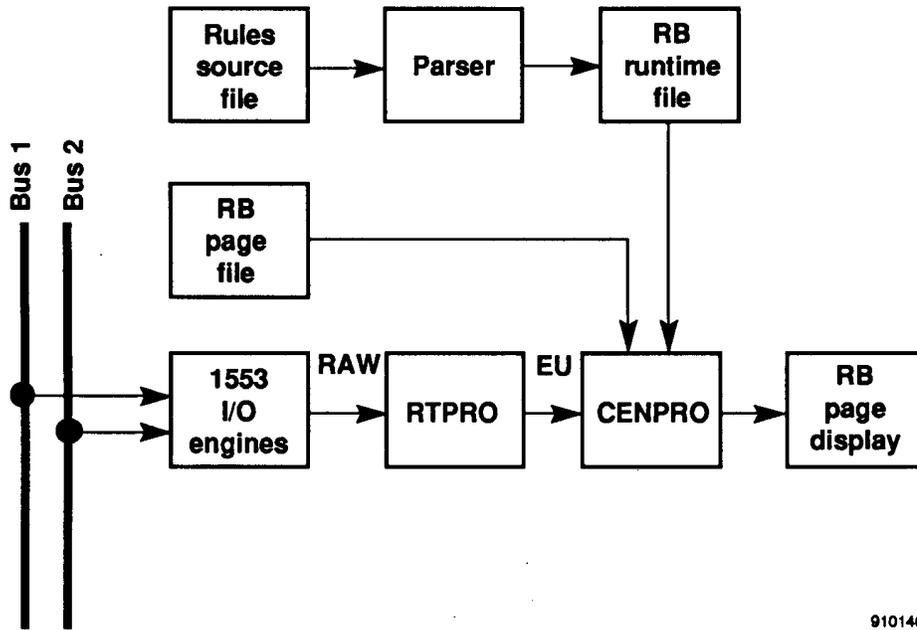
The program for the 1553B I/O engines receives setup information from RTPRO specifying which messages to fetch from a particular bus. It then sets up two identical buffers with each message to be fetched mapped at a particular address. The bus interface board is then commanded to begin sweeping up all messages from the bus into its buffer. The engine scans this buffer, identifies each message, and moves the data into the proper location in one of two buffers. When all messages have been retrieved, a signal is sent to RTPRO specifying which buffer is ready, and begins to fill the other buffer. This process continues until halted by RTPRO when the display is dropped.

To augment the RTPRO program, three channel handlers were written for the three pairs of redundant 1553B busses. The operator identifies which bus the data is to be fetched from by using the channel identifiers M1, M2, and M3. Also, to support testing of the rule based package, eight more “channels” were created and identified as S0 through S7. The operator then edits the symbol table and creates a temporary working file so a given parameter is fetched from one of these eight registers rather than the 1553B bus. Each channel is a 16-bit register in RTPRO which can be manually loaded by the operator using the SCP so that a desired value may be expressed for that parameter for data display testing purposes. An example of this procedure is described in section “XAIDS Parser and Display Page Verification Testing.”

One unusual raw data type is present on the F-18 HARV 1553B busses which could not be processed by any of the baseline raw data type handlers. It consists of 2 bytes packed into a 16-bit word, each half of which is encoded in sign and magnitude format. The two data types are HSMH (half word sign magnitude high) byte and HSML (half word sign magnitude low) byte.

The parser job was created to convert the American Standard Code for Information Interchange (ASCII) rules file (fig. 10) edited by the operator into a run-time control structure for the RB page. The statements in the input file are parsed, error checked, expanded into fundamental statement types, and a run-time structure is created. The operator then saves the structure to hard disk under any desired file name. This file is later loaded if specified as the companion to an RB display page.

To amend the CENPRO XAIDS job, several modules were added and one baseline module was modified. The RB page display permits the operator to display data on the upper half of the page and show derived information in message format on the lower half of the page. When an RB page is created, the operator must specify the name of the companion run-time file created by the PARSER. To support the RB page with its high-density presentation, two new formats for the display of events were created: “U” to unpack a selected bit of a 16-bit discrete word and “N” to negate and unpack a selected bit of that word. This permits a single symbol table entry to be used for all 16 events, with the desired bit specified in the display format. In both cases, the bit is displayed on the screen as either an “x” if true or blank if false. The baseline XAIDS symbol table manager package was modified to permit mapping 1553B parameters (those fetched from M1, M2, or M3) into the proper position in a given bus message as identified by message type, command word, and word sequence number.

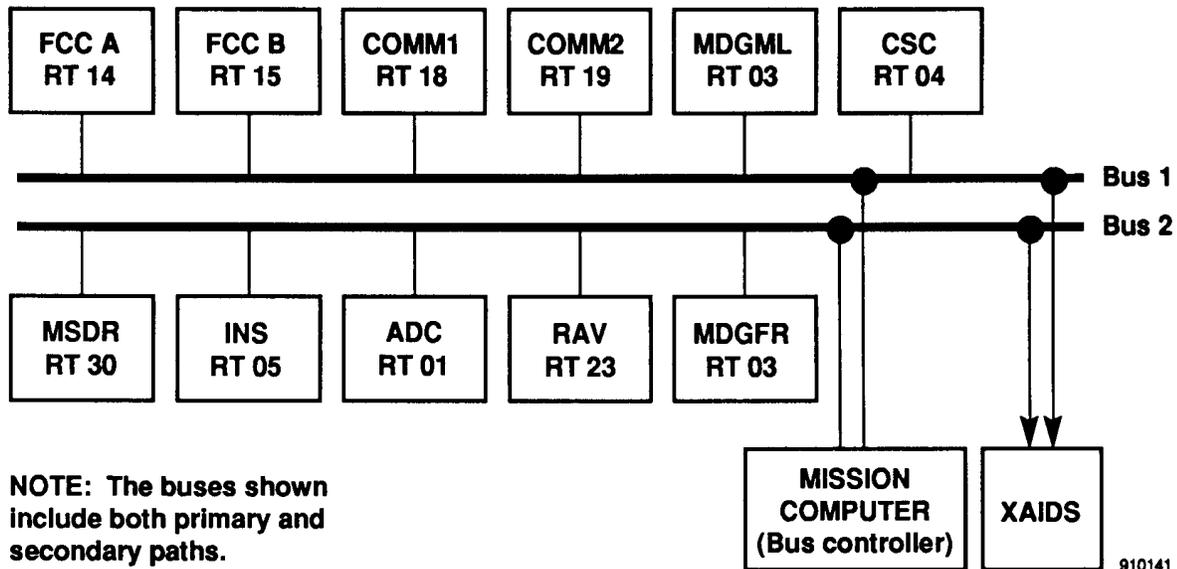


910140

Figure 10. Rule based data acquisition overview.

### Rules Source Syntax Description

The F-18 signal traffic network is highly distributive consisting of numerous remote terminals (RTs) with dually redundant mission computers providing the bus controller function. The simulation bus configuration which was used to develop the rules base application is shown in figure 11. The XAIDS is tied to these two busses, therefore, providing thousands of data words for monitoring. Up to 510 of these words are available to the XAIDS at a given time for testing. The rules source syntax was developed for this particular application, however, this concept should be applicable to other architectures with minimal modifications.



910141

Figure 11. XAIDS bus connection.

The rule base package is a special application whereby a user can create a knowledge base of the system for the purpose of automating the interpretation of the data words from the 1553 bus. This knowledge structure allows for conditional requirements to be satisfied before an interpreted result is generated. A tremendous amount of signal processing and data output combinations is possible from a single display page. This greatly simplifies the real-time monitoring that is necessary for a system. However, the primary advantage is that once the knowledge base has been constructed, the auto analysis is then available to less knowledgeable users of the system. The responsibility of the data interpretation is no longer a function of the individual monitoring the data. The knowledge base structure consists of three parts:

1. reads raw input data words from the 1553 bus (FETCH statement)
2. create logicals to trigger message outputs directly or nested, dependent logicals for multiple message requirements (EVENT, ARITH, COMP, BOOL statements)
3. syntax for three types of message outputs (MSG statement)

Statements following each section keyword are separated by a comma and terminated with a semicolon. A description of how each statement works will now be given.

## **FETCH**

The FETCH command defines a mnemonic name from the symbol table for reference in the RB display application. The particular 1553 bus must also be identified (fig. 11). The format is shown in the following

FETCH: <mnemonic> = <symbol table name> @ <bus (M1 or M2)>;

Example:

FETCH: NZ = ICANAC @ M1;

## **EVENT**

The EVENT statement is used to define a logical parameter from a single bit from the 16-bit word defined in the FETCH statement. The + or – field is used to invert the logical. It has the following format

EVENT: <logical> = <mnemonic (from FETCH)> <+ or –> <bit position>;

Example:

EVENT: OK = STATUS + 4;

The logical OK will be set to true if bit position 4 in the word STATUS which was defined from the FETCH list is asserted. Otherwise the logical OK will be set false. The logical OK could be inverted by the expression OK = STATUS – 4 (OK is asserted if bit 4 is not asserted).

## **ARITH**

The ARITH command is used to create a logical parameter from an arithmetic expression. Its operators are +, –, \*, /, ABS. It has the following format

ARITH: <logical> = <mnemonic> <operator> <mnemonic>;

Example:

ARITH: DELAOA = AOA1 – AOA2;

## COMP

The COMP command is used to create a logical parameter from a comparison expression. Its operators are .LT., .LE., .EQ., .NE., .GE., and .GT.. It has the following format

COMP:<logical> = <mnemonic> <operator> <mnemonic> ; (up to 8 terms)

Examples:

The following example shows 2 equations which must be separated by a comma; the last equation in the section must be delimited by a semicolon.

COMP:

NOTZERO = TEST .NE. 0,  
LOWSPD = VIAS .LT. 140.;

## BOOL

The BOOL command is used to create a logical parameter from a Boolean expression. Its operators are .AND., .NOT., .OR., and .XOR.. It has the following format

BOOL: <logical> = <logical> <operator> <mnemonic> ; (up to 8 terms)

Examples:

BOOL: B1000 = B0 .AND. (.NOT. (B1 .OR. B2 .OR. B3));

In this example B0, B1, B2, and B3 have been previously defined. The variable B1000 will be set if B0 is true and B1, B2, and B3 are all false.

BOOL: B1000SUB = B1000 .AND. (( .ABS. R\_RATE) .GE. 30.0);

In this example R\_RATE is a scalar word which was previously defined in the FETCH list. B1000SUB will only be set if B1000 (previously defined in the Boolean statement is true) and the absolute value of R\_RATE is greater than or equal to 30. A sub message can be created from a top level message which can be used to provide more detail about the higher level message.

## MSG

The MSG command contains the output message for the RB display page. This message can be triggered from the logicals defined from either the EVENT, ARITH, COMP or BOOL. There are three types of message formats: (1) simple logical, (2) embedded logicals (1-5 channels), (3) embedded data display in EU or hexadecimal based number system (HEX).

Formats:

1. MSG: <logical> = '<message string up to 39 characters>';
2. MSG: <logical1 & logical2 & ...logical5> = '<message string up to 39 characters>'#<overwrite message position >;

The overwrite message position identifies the position starting at zero in the message string where numbers 1, 2 ... 5 will be written if logical 1, 2 ... 5 are asserted, otherwise, these positions will be blanked. This logic was developed to indicate which channels are setting a common message so that duplicate messages from each channel could be eliminated.

3. MSG: <logical> <\$(eu),%(hex)> <data mnemonic> = 'message string up to 39 characters including data format field';

The data value is inserted immediately after the last character in the message string. This value will be updated at the message stack rate.

Examples:

MSG:

FFLAPS = 'FULL FLAPS',

QF1 & QF2 & QF3 & QF4 = 'C1234: PITCH RATE FAIL' #1,

The message string will be overwritten by 1234 or blanks starting at the character "1" depending on whether the logicals are true or false, respectively)

LOWSPD \$ VIAS = ' AIRSPEED < 140 = ';

## **PTR**

The PTR is a command to the printer which uses a list of logical flags previously defined using EVENT, ARITH, COMP or BOOL. If any of these logical flags are asserted, an automatic page print command to the RB display page will be generated when the auto print option flag on that page is selected. The auto print works on an "edge trigger" concept, whereby a logical must go from false to true before the auto print will be activated. Only one printout will be generated for each true state. It has the following format

PTR: <logical1, logical2, ... logical100>;

Example:

PTR: FAIL1, FAIL2;

## **Parser**

The parser program takes the rule base source file as input and performs the following functions (1) parses, (2) error checks, (3) builds symbol tables, and (4) creates run-time control data structure as an output file. The parser menu options are shown in figure 12. Messages will appear on the CRT if errors are found by the parser. The source file name is entered using the LOAD command. This executes the parsing program using the source as the input file. The error checker runs two passes through the source file. Messages are generated if any errors are found (refer to appendix A for the parser error messages). The parenthetical elements are expanded and intermediate Boolean terms are generated. At various stages of the parsing messages are written to the CRT. At the completion of a successfully parsed file a complete summary page (fig. 13) of the element expansion is produced. When the <esc> key is pressed, control is returned to the parser menu. The runtime file can be created at this point. The parsed file can also be printed at this time.

To understand how the parsing works, consider the example where the following expression is defined in the BOOL section

M8.18.01.A = M8.18.01 .AND. ((.ABS. R\_RATE) .GE. 301.0)

The parser expands this expression as shown below in the following

ARITH: L00015 = .ABS. R\_RATE;

COMP: L00016 = L00015 .GE. 301.0;

BOOL: M8.18.01.A = M8.18.01 .AND. L00016;

```

RB Parser          EXTENDED AIRCRAFT INTERROGATION & DISPLAY SYSTEM    08:17:43
                   XAIDS BOOLEAN FILE PARSER                          20 SEP 90

COMMANDS
LOAD   (LOAD FILE)          LOAD & PARSE BOOLEAN SOURCE FILE
LIST   (LIST FILE)         LIST PARSED FILE TO CRT OR LP
SAVE   (SAVE FILE)         CREATE RUN-TIME FILE FOR RB PAGE
EXIT   (EXIT PARSER)       RETURN TO RMX86 EXECUTIVE

ENTER COMMAND >

```

910142

Figure 12. Parser menu.

The Lxxxxx terms are created by the parser to form intermediate logicals so that the Boolean expression can be satisfied. The logical operators determine which keyword section is expanded as explained in the rules source file syntax description.

### Rule Base Page

The RB page format (fig. 14) consists of the top half screen (13 lines) reserved for a free form display used for monitoring fixed data while the bottom 7 lines are allotted for the interpreted data in the form of messages. The rule base software can generate up to a 999 message stack which is continually updated at about a 2-Hz rate. However, on the RB page there is only room to display the first 14 messages from the stack (each line is split into a pair of 39-character message fields) which are defaulted on the display page. To view the other messages the window can be moved manually in 7-message increments by using the left and right arrows on the key pad, therefore, it is possible to view any active message on the stack.

The stack will continue to update as newer messages are triggered and pushes the stack down starting at the top left while the 15th message is pushed off the screen at the lower right. Rescinded messages are flashed at 2 Hz for 5 sec and then removed, creating a temporary slot in the stack. The next older messages are then automatically pushed up, filling in the empty slot.

There is an advantage to incorporating the RB format independent of the free-form (FF) display page. The user can remain totally flexible in creating new FF displays and still select the default rule base file to maintain the signal processing test coverage of possible unexpected failures.

## EXTENDED AIRCRAFT INTERROGATION & DISPLAY SYSTEM

```
ENTER COMMAND > load
INPUT BOOLEAN FILENAME > faults
LOADING FILE . . . WAIT
  NUMBER OF BYTES IN INPUT FILE           = 62592 OUT OF 65535
EDITING INPUT FILE . . . WAIT
  NUMBER OF BYTES AFTER EDITING           = 49592 OUT OF 65535
EXPANDING PARENTHETICAL ELEMENTS . . . WAIT
  NUMBER OF BYTES AFTER EXPANSION         = 51909 OUT OF 65535
PERFORMING FIRST PASS . . . WAIT
  NUMBER OF ELEMENTS                       = 1849 OUT OF 5000
  NUMBER OF FETCHED PARAMETERS             = 141 OUT OF 150
  NUMBER OF BOOLEAN PARAMETERS            = 1163 OUT OF 2500
PERFORMING SECOND PASS . . . WAIT
  NUMBER OF EVENT TERMS                    = 790 OUT OF 1000
  NUMBER OF MATH EQUATIONS                 = 36 OUT OF 100
  NUMBER OF COMPARISON EQUATIONS           = 121 OUT OF 200
  NUMBER OF COMPARISON CONSTANTS          = 122 OUT OF 200
  NUMBER OF BOOLEAN EQUATIONS              = 252 OUT OF 400
  NUMBER OF MESSAGES                       = 488 OUT OF 550
  NUMBER OF AUTO PRINT TRIGGERS           = 21 OUT OF 100
WAITING FOR <esc> >
```

910143

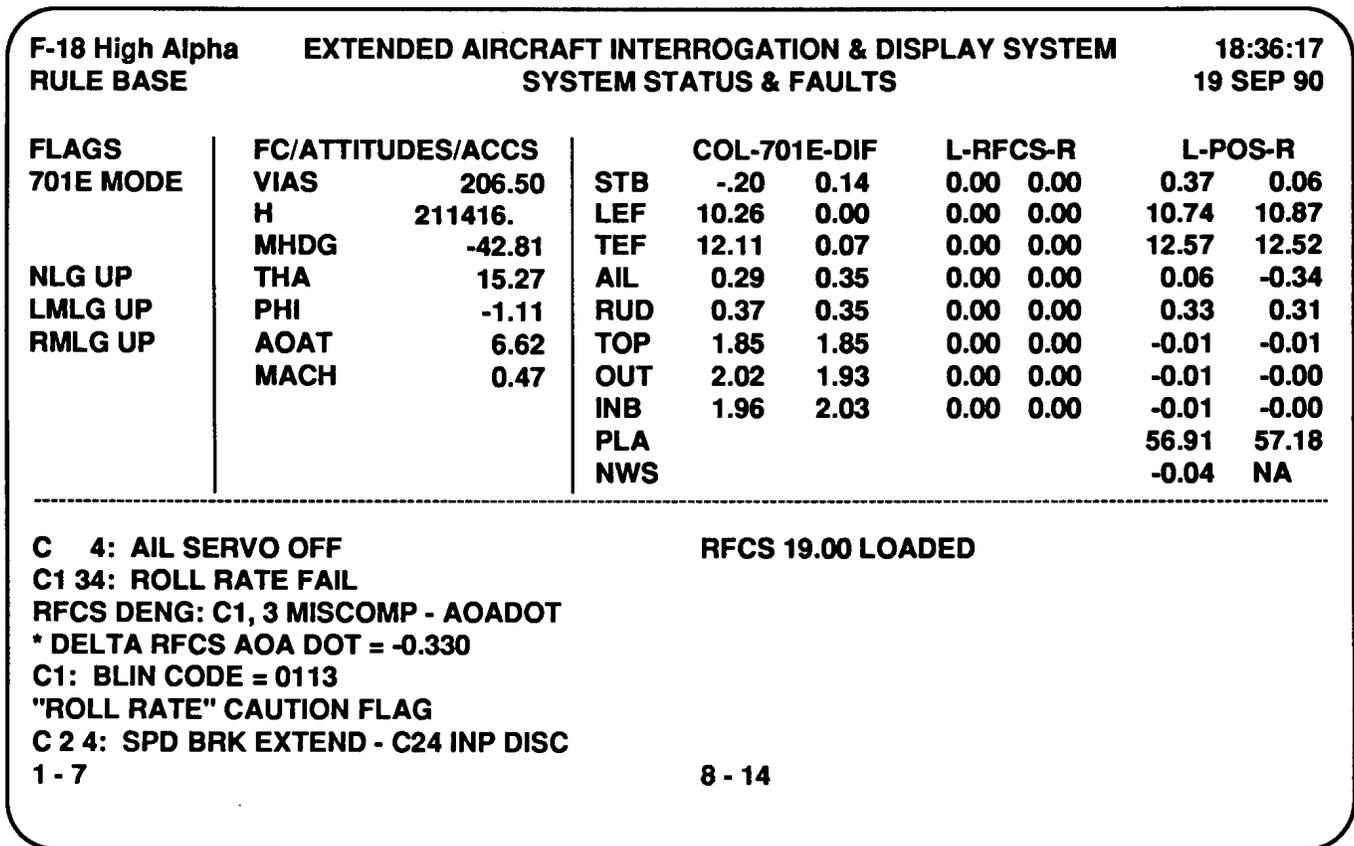
Figure 13. Parser expansion summary.

### F-18 HARV Unique Messages Example

Figure 14 is a typical display of the RB page application developed for the F-18 HARV program. The top two thirds of the page is in a FF format which shows a continuous real-time monitor of a particular set of data. Starting at the left the data display consists of discretes (up to ten character description); flight condition, attitudes, and accelerations; collective and differential commands; and finally, positions. The lower one third is the RB page application where the knowledge base has been applied and the output is in the form of messages.

The messages shown are typical examples of the different types that can be generated. To understand the logic, an edited version of the F-18 HARV knowledge base source file which was used to generate these messages is shown in appendix B. The source file was constructed in a way so that the analysis of the bus data is done for certain cases by grouping sets of test conditions before a message is generated. The rule base knowledge base consists primarily of system health status, modes, and configuration. If a fault indication is generated, the lowest level source causing that problem is automatically generated. Also, if a flight-control mode is disabled or disengaged, the reason for that problem is automatically generated with a message(s).

The rule base source file is in appendix B. This shows the minimum logic required to generate the eight message examples shown in figure 14. The first message, C 4: AIL SERVO OFF, indicates the aileron servo is off in channel 4. The logical M12.23.1 is the only one that is asserted since the "4" is the only channel which appears in the figure. The "123" are blanked out because the other logicals are not asserted. The logicals are defined in the EVENT section. The logical name refers to the source of the signal. For example M12.23.0 defines a logical parameter from message 12, word 23, bit 0. The ICAC23 is the name defined for this word from the symbol table



910144

Figure 14. Rule base page display.

which is identified in the FETCH section. Comment fields are delimited by /\* and \*/. The next message, C1 34: ROLL RATE FAIL, is defined similarly as the SERVO message. The logical flag corresponding to channels 1, 3, and 4 is asserted.

The RFCS DENG message is generated by decoding 4 bits from two different words (A8.18.456-latched and M17.26.456-updated) and OR'ing the result to form a signal flag (M8.18.456) as defined in the BOOL section. This flag is also used to trigger a submessage to provide more information relative to the higher level message. (A submessage is arbitrarily identified by a leading "\*" character.) In this case an angle-of-attack rate miscompare occurred between channels 1 and 3. The delta between these channels (DELRFCSAOAR) is computed in the ARITH section. Whenever the higher level flag is set for this miscompare, a submessage containing the delta is also output if the conditions for the submessage are satisfied.

A Bit Light INspect (BLIN) code embedded data message is shown in the next example. This flag (M9.1) is defined in the COMP section when a 16-bit word is not equal to zero. An output in hex was desired, therefore, the message logical and data value was separated by a % symbol as shown in the MSG section.

The ROLL RATE message is triggered by a single logical which is defined in the EVENT section. Bit decoding is not required since this is triggered by a single discrete. This format is shown in the MSG section as a simple logical according to the format (1) syntax.

The SPD BRK message comes from a generic message (memory inspect) which is used to inspect any FCC address. This logic is found in the COMP section. A particular address is tested in message 28 to set a flag (MI.09B3). Message flags for each channel are computed if the address matches and the 0 bit positions in message 29 are asserted. For this particular word the bits are inverted, using the NOT operator in the BOOL section. Since this particular



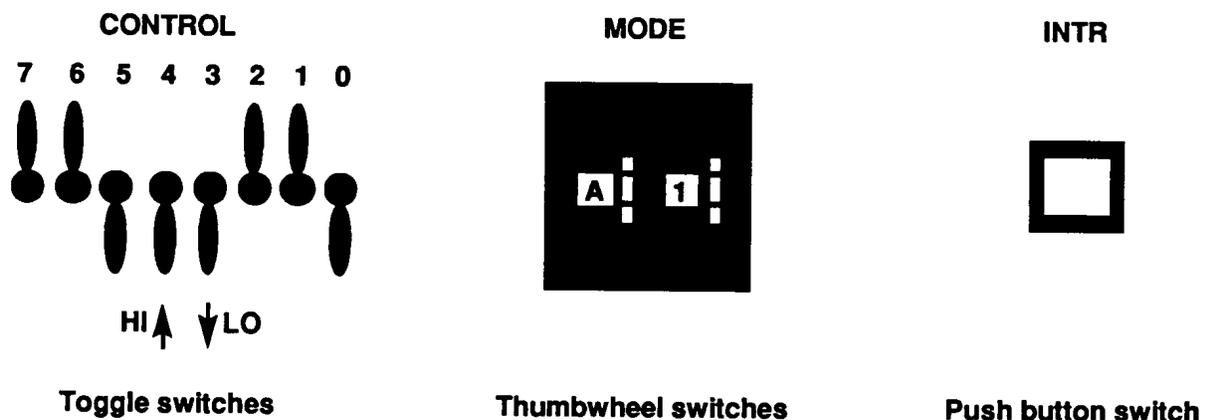
Table 2. Edit options for items 7, 8, and 9.

Item	Options
7	MSG SYNCH MODE (0 = NON-SYNCH 1 = BEGIN FRAME 2 = END FRAME)
8	MSG TYPE (1 = RCV 2 = XMT 3 = RT-RT 4 = MODE 5 = MODE XMT 6 = MODE RCV) ENTER RT NUMBER (1-31) DOES RT RECEIVE OR TRANSMIT ? (0 = RCV 1 = XMT) ENTER MESSAGE NUMBER (0 = 31) ENTER MESSAGE LENGTH (1-32)
9	ENTER WORD SEQUENCE # (1-32)

### Extended Aircraft Interrogation and Display System Parser and Display Page Verification Testing

Considerable software was developed for the parser, rule base constructs, and display page signal referencing. To verify this software, an important software testing tool was developed so that all 16 bits of any given 1553B bus signal could be manipulated. Modifications were made in the SYMBOL table and the SCP which provides an external input (fig. 16) so that up to eight 1553B bus signals could be controlled. This procedure will now be described.

All page displays such as the one shown in figure 14 were verified by temporarily reassigning the RELATED CHANNELS for pertinent 1553B bus signals defined in the SYMBOL TABLE (fig. 15) to S0, S1, . . . S7. Control of up to 8 16-bit words is possible by the use of 8 toggle and 2 thumbwheel (0-F) test switches on the XAIDS SCP as shown in figure 16. The most significant bit (MSB) of the word on the bus is referenced by toggle switch 7; the least significant bit (LSB) is the right hand thumbwheel position 1. Therefore, the switch setting shown in the figure in hex is C6A1. To actually substitute this value for a symbol table mnemonic with a bus referenced to S0, the INTR button is pressed. The switch reference can be changed to S1 by setting all the toggle switches down with number 1 set in the 'up' position, then set both thumbwheel switches to FF and press INTR. Values can now be entered for an S1 defined mnemonic. This procedure can be repeated similarly for S2 to 7. As the data values are entered on these switches the appropriate data on the page display can be verified.



910148

Figure 16. XAIDS status and control panel.

## **F-18 TEST RESULTS**

The test results presented used the entire rule base package developed for the F-18 HARV program, which currently has about 490 unique messages available out of 550 possible. The 4-channel embedded discrete messages can generate 15 unique combinations, but are counted as a single message. The system fault status from each channel along with configuration, modes, and research flight-control system (RFCS) mode engage-disengage logic has been programmed into the knowledge database. A number of special messages for particular addresses from the four memory inspect words have also been programmed.

It was not possible for the XAIDS equipment to perform data computations for the real-time display at the 80-Hz frame rate. The RB format updates data to the screen at an approximate rate of 2 Hz depending on how many messages are in the stack. With this in mind, the data that could be monitored by the XAIDS was not as dynamic. For dynamically changing conditions other strategy would have to be used which is discussed in the FUTURE PLANS section.

The rule base application in the XAIDS was used with great success during the F-18 HARV FMET. Test results were immediately verified from a single page display which provided the interpreted results for the entire activity. Off line data processing other than to generate overlay plots was totally eliminated. The RB pages were printed and included directly into the system test report without any additional interpretation required. The report was written shortly after the completion of the tests because the degree of the automation by the rule base analysis application. During the FMET activity, additional engineers were required who did not have an in-depth understanding of the F-18 HARV system; however, because the rule base application was available to all users the test results were easily verified.

The rule base test results for an FMET channel 4 and yaw rate fail in channel 2 are shown in appendix C. Many fail flags from channel 4 are set because of the channel 4 fail. The fail flags that are generated as a result of a channel fail could be eliminated by creating a knowledge rule to suppress their messages if that particular channel fail is present. Only the FCC FAIL message would be output. The yaw rate fail in channel 2 is shown as C 2 4: YAW RATE FAIL. The C4 was already set because of the channel 4 fail. The BLIN CODE message in channel 3 is output in hex (001B). This code description is X CHANNEL POWER OFF (channel 3 to pointing to a channel 4 power condition). This information could have been included in the rule base knowledge database; however, since there are several hundred of these codes, there was not enough memory in the XAIDS for this application. It is possible to create a separate rule base file for only the BLIN codes. A FF page output was also printed to provide a more pictorial display of some of the failures which looks more like the FCS STATUS display page on the F-18 digital display indicator (DDI).

During and after the FMET, updates to the XAIDS software were made to add and to correct messages. Some messages were corrected because the asserted states were inverted on the bus. Additional messages were added for use with the memory inspect message. All changes were made and verified during off-testing hours so that the XAIDS could be used full time by the project.

This system is used daily for all levels of testing including general health status monitoring. The formal research control law verification tests have not begun but a significant amount of logic for the control law arming, abort, and engage-disengage logic has been programmed and tested. This system should continue to be an aid to the F-18 HARV for the duration of the program.

## **FUTURE PLANS**

A version of the current XAIDS rule base package will be included as part of the computer aided systems testing (CAST) being developed at Dryden (ref. 4). The CAST will use a new generation of auto test strategy using workstations in the Dryden Integrated Test Facility (ITF). This conversion will require recoding using the

C programming language which will then run on a Unix<sup>®</sup> operating system (ref. 6). This software should then become portable to any popular Unix platform.

Because of the hardware and software limitations using the XAIDS, several features for the rule base system were not included in the package. These items are planned for the CAST and are shown in the following paragraphs.

1. **Sample rate** – Data should be monitored and recorded at the highest sample rate message on the bus so that dynamically changing data is not lost. In the case of the F-18 HARV this would be 80 Hz. It takes 5 sec for any message to be rescinded from the display stack, therefore, if a message was triggered for only 1 sample iteration at 80 Hz it would still be easily observed. The screen refresh rate should not be confused with the data sample. The screen refresh rate is done at 2 Hz.
2. **Compute max, min** – A maximum and minimum function for any signals in the rule base database should be done. The real-time values could then be displayed using the embedded data message format. Peak values of signals would be available if desired. The max/min signals could be cleared when the display is not in the refresh mode.
3. **Generate message log file with time tag** – For every message generated and rescinded, there needs to be a message log file with a time tag written. This would provide a time history of all the messages generated during a data collection period.
4. **Record/playback data into rule base database** – All data referenced by the symbol table needs to be saved into a file which is available for playback into the rule base package at a “non-real-time” rate. This rate would be selectable by the user at some multiple of real time (for example 1/10 of real time). This file could also be edited for the purpose of creating raw word combinations for the purpose of verifying new display pages.
5. **Read 1553 STATUS bus words** – There is a STATUS word from each RT which follows the data portion of the 1553 transmission. It is often hard to tell if live data from a particular message is being received for a given RT. The bits from this word contain error information which could be decoded to determine if there are any problems.
6. **Logic analyzer “stop” function** – It would be helpful for troubleshooting purposes to check for a stop condition by testing for a particular bit(s) pattern in some predetermined word using the typical Boolean operator set. While this test is being done, a selected set of words is being written to a buffer file in which the last n seconds are saved. When the stop condition is satisfied, the buffer write stops. This file can then be inspected or played into some formatted display page.
7. **Message “latch”** – This would be used to freeze a particular message as a latched message if it should be set. The user would then be reminded to look at the message log file to determine what caused this message to be set. For example an FCC out of sync word might be monitored by the rule base system which should never be set. The user may want to create a latch for this message if it should be set.
8. **Message counter** – A particular message counter would be displayed in the data field of the message to indicate the number of times a message has been asserted and rescinded. For example, a frame overrun word from the executive might be monitored. The number of overruns may be related to the engagement of a particular mode. This information may be helpful to understand the problem.
9. **Direct memory access (DMA) interface at minor frame rate** – The F-18 does not provide a DMA interface for the GSE. This limitation requires that all data words for monitoring must be on the bus. Unfortunately, most internal words that need to be monitored are not on the bus. The memory inspect function from the

---

<sup>®</sup>Unix is a registered trademark of AT&T Bell Laboratories, Whippany, NJ, 07981.

built-in-test (BIT) in the case of the F-18 must be used. However, this is only a 4-word, 10-Hz message which is a severe limitation. Future digital flight-control systems should have this DMA interface. The GSE then could read any internal address at the minor frame rate which would be 6.25 msec for the F-18.

## CONCLUDING REMARKS

This report describes the extended aircraft interrogation and display system design and test experience gained using a rule base application which was developed in-house for the F-18 high alpha research vehicle program. A knowledge base was generated containing system health status, modes, flight-control moding condition logic, and configuration. The interpretation of the raw data signals was displayed in real time in the form of messages. The rule base software was written in programming language microprocessor within the existing extended aircraft software package. Several design iterations were made as a result of the test experience gained using the F-18 high alpha research vehicle systems.

The automated analysis concept proved to save considerable testing, data processing, and analysis time during the F-18 high alpha research vehicle failure modes and effects tests activities. Much of the normal, manual interpretation of the test results was eliminated. The rule base pages were printed and inserted directly into system test reports reducing documentation writing time because of the easy interpretation of the test results.

The extended aircraft interrogation and display system has proven to be a valuable tool for the tests described in this paper and for follow on testing phases in support of the F-18 high alpha research vehicle program. The concept of organizing the data presentation by combining a real-time display of fixed data (free form display) with a message stack format (rule base display) has proven very useful for these tests. A single, general-purpose display which can generate up to 550 messages in the lower 7 lines of the screen plus 13 lines fixed data formatting has the effect of replacing numerous free form displays which could only be monitored individually and not self interpreting.

## REFERENCES

1. Glover, Richard D., *Aircraft Interrogation and Display System: A Ground Support Equipment for Digital Flight Systems*, NASA TM-81370, 1982.
2. Glover, Richard D., *Application Experience With the NASA Aircraft Interrogation and Display System: A Ground Support Equipment for Digital Flight Systems*, Proc. IEEE/AIAA 5th Digital Avionics Systems Conference, Seattle, WA, Oct. 31–Nov. 3, 1983, pp. 17.3.1 to 17.3.10.
3. Glover, Richard D., *Design and Initial Application of the Extended Aircraft Interrogation and Display System: Multiprocessing Ground Support Equipment for Digital Flight Systems*, NASA TM-86740, 1987.
4. Sitz, Joel R. and Todd H. Vernon, *Flight Control System Design Factors for Applying Automated Testing Techniques*, NASA TM-4242, 1990.
5. Glover, Richard D., *Concept of a Programmable Maintenance Processor Applicable to Multiprocessing Systems*, NASA TM-100406, 1988.
6. Kernighan Brian W. and Rob Pike, "The UNIX Programming Environment," Prentice-Hall, Inc., Englewood Cliffs, NJ, 1984.

## APPENDIX A PARSER ERROR MESSAGES

The following is a list of all possible error messages that can be generated during the first and second passes of the parser.

### 1ST PASS ERROR CHECKING

#### COMMENT

ILLEGAL COMMENT FIELD

#### SYNTAX EXPANSION

UNTERMINATED LITERAL FIELD

CONCATENATED DECIMAL POINTS - UNRESOLVABLE

EDITED BUFFER STARTS WITH ILLEGAL CHARACTER

MISMATCHED PARENTHESES

TOO MANY SUBFIELDS IN AN EXPRESSION

TABLE OVERFLOW - 1024 ELEMENTS PARSED

NO ELEMENTS PARSED

SUB-ELEMENT ERROR(S) FOUND IN EXPRESSION

#### FETCH:

ILLEGAL <FETCH:> EXPRESSION

NO <FETCH:> ELEMENTS

<FETCH:> ERROR

#### EVENT:

ILLEGAL (EVENT) EXPRESSION

ILLEGAL <EVENT:> ASSIGNMENT

<EVENT:> ERROR

#### ARITH:

ILLEGAL <ARITH:> ASSIGNMENT

<ARITH:> ERROR

#### COMP:

ILLEGAL COMPARISON ASSIGNMENT

<COMP:> ERROR

#### BOOL:

ILLEGAL <BOOL:> ASSIGNMENT

INVALID BOOLEAN OPERATOR

<BOOL:> ERROR

#### MSG:

ILLEGAL <MSG:> ELEMENT

NO <MSG:> ELEMENTS

<MSG:> ERROR

#### PTR:

<PTR:> ERROR

ILLEGAL <PTR:> EXPRESSION

## 2ND PASS ERROR CHECKING

### GENERIC

ILLEGAL ELEMENT TYPE -  
SUB-ELEMENT ERROR -  
INVALID OPERATOR

### FETCH:

ILLEGAL FETCH ASSIGNMENT

### EVENT:

EVENT TERMS OVERFLOW  
ILLEGAL EVENT ASSIGNMENT  
ILLEGAL BIT NUMBER

### ARITH:

# MATH OPERATIONS OVERFLOW  
ILLEGAL # SUBFIELDS IN ARITHMETIC EXPRESSION  
ILLEGAL ARITHMETIC ASSIGNMENT  
INVALID ARITHMETIC OPERATOR

### COMP:

# COMPARISONS OVERFLOW  
ILLEGAL COMPARISON ASSIGNMENT  
INVALID COMPARISON OPERATOR

### BOOL:

# EQUATIONS OVERFLOW  
ILLEGAL BOOLEAN ASSIGNMENT

### MSG:

MESSAGE TABLE OVERFLOW  
ILLEGAL OPERATOR  
MISSING FIELD POSITION  
FIELD POSITION WITH SINGLE PARAMETER  
ILLEGAL FIELD POSITION  
LAST OPERATOR NOT = OR #  
BAD NTERMS OR BAD MESSAGE FIELD  
OPERATOR SEPARATING TERMS NOT &  
SECOND TERM NOT EMBEDDED DATA  
ILLEGAL MESSAGE ELEMENT

### PTR:

PRINTER TABLE OVERFLOW

## APPENDIX B DATA RULES SOURCE FILE

The following rule base source file is an edited version for illustration purposes of the one developed for the F-18 HARV program.

### FETCH:

```
/* RT 14; MSG 8 */
ICAV18 = ICAV18 @ M1,

/* RT 14; MSG 11 */
RFCSAOAR=OCAM1105 @ M1,

/* RT 15; MSG 11 */
RFCSAOARB=OCBM1105 @ M1,

/* RT 14; MSG 9 (BLINS) */
ICA1B1=ICA1B1 @ M1, ICA2B1=ICA2B1 @ M1, ICA3B1=ICA3B1 @ M1,
ICA4B1=ICA4B1 @ M1,

/* RT 14 (FCC 1,2); MSG 12 (FAULTS) */
ICAC01=ICAC01 @ M1, ICAC05=ICAC05 @ M1, ICAC09=ICAC09 @ M1,
ICAC13=ICAC13 @ M1, ICAC23=ICAC23 @ M1,

/* RT 14; MSG 17 */
ICAX26 = ICAX26 @ M1,

/* RT 14; MSG 28 */
OCAADD = OCAADD @ M1,

/* RT 14; MSG 29 */
ICAM01=ICAM01 @ M1, ICAM02=ICAM02 @ M1, ICAM03=ICAM03 @ M1,
ICAM04=ICAM04 @ M1,

/* RT 14; MSG 31*/
ICARON=ICARON @ M1;
```

### EVENT:

```
/* MSG 12 BIT UNPACKING */
M12.1.13 =ICAC01+13, M12.5.13 =ICAC05+13, M12.9.13 =ICAC09+13,
M12.13.13 =ICAC13+13, M12.23.0 =ICAC23+ 0, M12.23.1 =ICAC23+ 1,
M12.23.8 =ICAC23+ 8, M12.23.9 =ICAC23+ 9,

/* RFCS FLAG WORD - MSG 17; WORD 26 (UPDATED) */
W17.26.4 =ICAX26 + 4, W17.26.5= ICAX26 + 5, W17.26.6 = ICAX26 + 6,
W17.26.7 =ICAX26 + 7,

/* MSG 29; WORDS 1,2,3,4 (MEMORY INPECT) */
Y29.1.0 = ICAM01 +0, Y29.2.0 = ICAM02 +0, Y29.3.0 = ICAM03 +0,
Y29.4.0 = ICAM04 +0;
```

### ARITH:

```
DELRFCSAOAR=RFCSAOAR-RFCSAOARB;/* COMPUTE AOA DOT DELTA*/
```

### BOOL:

```
/* SET FLAG IF BITS 4,5,6,7 ARE 1110 IN MSG 8; WORD 18*/
A8.18.456=W8.18.4 .AND. W8.18.5 .AND. W8.18.6 .AND. (.NOT.W8.18.7),
```

/\* SET FLAG IF BITS 4,5,6,7 ARE 1110 IN MSG 17; WORD 26\*/  
M17.26.456=W17.26.4 .AND. W17.26.5 .AND. W17.26.6 .AND.  
(.NOT.W17.26.7),

/\* SET FLAG IF EITHER A8.18.456 OR M17.26.456 IS SET\*/  
M8.18.456=A8.18.456 .OR. M17.26.456,

/\* CREATE A SUBMESSAGE FLAG IF DELTA RFCS AOA DOT NE 0)  
M8.18.456.A=M8.18.456 .AND. (DELRFCSAOAR.NE.0);

COMP:

FALSE = 0 .EQ. 1 /\* CREATE DUMMY FALSE FLAG FOR MESSAGES\*/

/\* MSG 9; BLIN CODES\*/

M9.1=ICA1B1.NE.0,

/\* MSG 31; OFP S/W CONFIGURATION\*/

M31.19.1=ICARON.EQ.609,

/\* MSG 28; MEMORY INSPECT WORDS \*/

MI.09B3 = OCAADD .EQ. 2483, /\* 2483=09B3 HEX \*/

BOOL:

/\* MSG 29; COMPUTE MESSAGE FLAGS FOR MEMORY INSPECT WORDS  
IF ADDRESS (09B3) WAS FOUND IN OCAADD \*/

M29.2.0.09B3 = MI.09B3 .AND. (.NOT.Y29.2.0),

M29.4.0.09B3 = MI.09B3 .AND. (.NOT.Y29.4.0);

MSG:

/\* MESSAGE 8,17;RFCS DISCRETE WORDS\*/

M8.18.456\$DELRFCSAOAR='\* DELTA RFCS AOA DOT =',  
M8.18.456='RFCS DENG; C1,3 MISCOMP - RFCS AOADOT',

/\* MESSAGE 9; BLIN CODE WORDS\*/

M9.1%ICAAB1='C1: BLIN CODE =',

/\* MESSAGE 12 WORDS; FAULTS, STATUS WORDS \*/

M12.1.13 & M12.5.13 & M12.9.13 & M12.13.13 =

'C1234: ROLL RATE FAIL'#1,

M12.18.4 = "'ROLL RATE" CAUTION FLAG',

M12.23.0&M12.23.8&M12.23.9&M12.23.1='C1234: AIL SERVO OFF'#1,

/\* MEMORY INSPECT MESSAGES\*/

FALSE & M29.2.0.09B3 & FALSE & M29.4.0.09B3 =

C1234: SPD BRK EXTEND - C24 INP DISC'#1,

/\* MESSAGE 31 RFCS OFP CONFIGURATION WORD \*/

M31.19.1='RFCS 19.1 LOADED';

PTR:

M8.18.456;

**APPENDIX C**  
**FMET TEST RESULTS FOR CHANNEL 4 AND YAW RATE 2 FAIL**

The following pages from the RB real-time displays were printed after the failures were inserted. There were 72 messages, therefore, the message stack was shifted by 14 messages for each subsequent printout so a record of all messages was possible.

FLAGS	FC/ATTITUDES/ACCS	COL-701E-DIF	L-RFCS-R	L-POS-R
701E MODE	VIAS 215.69	STB 0.00 0.14	0.00 0.00	1.16 0.06
	H 25000.	LEF 8.08 0.00	0.00 0.00	8.11 8.06
	MHDG 15.94	TEF 8.55 0.07	0.00 0.00	8.57 8.59
NLG UP	THA 6.07	AIL -0.04L 0.04R	0.00 0.00	-0.18 0.02
LMLG UP	PHI 0.00	RUD 0.00L 0.00R	0.00 0.00	0.00 -0.04
RMLG UP	AOAT 6.09	TOP	0.00 0.00	-0.01 -0.01
	MACH 0.53	OUT	0.00 0.00	-0.01 -0.01
		INB	0.00 0.00	-0.01 0.00
		PLA		56.91 56.91
		NWS		0.00 NA

C3: BLIN CODE = 001B  
 C 4: RT STAB SERVO FAIL  
 C 4: LT STAB SERVO FAIL  
 C 4: RDR SERVO OFF  
 C 4: AIL SERVO OFF  
 C 4: DEGRADED  
 C 4: PROCESSOR OFF

C 4: BADSA DATA FAIL  
 C 4: AOA FAIL  
 C 4: RUDDER PEDAL FAIL  
 C 4: STICK FAIL  
 C 2 4: YAW CAS FAIL  
 C 4: ROLL CAS FAIL  
 C 4: PITCH CAS FAIL

1 - 7

8 - 14

72

File = F18.LARSON/001

Directory = F18 High Alpha Program Test Displays

FLAGS	FC/ATTITUDES/ACCS	COL-701E-DIF	L-RFCS-R	L-POS-R
701E MODE	VIAS 215.69	STB 0.00 0.00	0.00 0.00	1.16 1.12
	H 25000.	LEF 8.06 0.00	0.00 0.00	8.11 8.06
	MHDG 15.94	TEF 8.52 0.00	0.00 0.00	8.50 8.59
NLG UP	THA 6.06	AIL -0.09L 0.09R	0.00 0.00	-0.23 -0.01
LMLG UP	PHI 0.00	RUD 0.04L 0.02R	0.00 0.00	0.00 0.00
RMLG UP	AOAT 6.09	TOP	0.00 0.00	-0.01 -0.01
	MACH 0.53	OUT	0.00 0.00	-0.02 0.00
		INB	0.00 0.00	-0.02 -0.01
		PLA		56.91 57.00
		NWS		0.00 NA

C 4: RT TEF SERVO FAIL  
 C 4: LEF SERVO FAIL  
 C 4: LT TEF SERVO FAIL  
 "FCS" CAUTION FLAG  
 "NWS FAIL" CAUTION FLAG  
 C ---4: LT APC FAIL  
 C 4: NWS FAIL

C 4: LT RDR DEL FAIL  
 C1 4: LT AIL DEL FAIL  
 C 4: PITCH TRIM FAIL  
 C 4: ROLL TRIM FAIL  
 C 4: L STAB SOV 2 OPEN  
 C 4: R STAB SOV 2 OPEN  
 C 4: R STAB SOV 1 OPEN

15 - 21

22 - 28

72

File = F18.LARSON/001

Directory = F18 High Alpha Program Test Displays



FLAGS	FC/ATTITUDES/ACCS	COL-701E-DIF	L-RFCS-R	L-POS-R
701E MODE	VIAS 215.69	STB 0.00 0.00	0.00 0.00	1.23 1.18
	H 25500.	LEF 8.04 0.00	0.00 0.00	8.06 8.02
	MHDG 15.94	TEF 8.50 0.00	0.00 0.00	8.50 8.59
NLG UP	THA 6.05	AIL -0.04L 0.09R	0.00 0.00	-0.07 0.02
LMLG UP	PHI 0.00	RUD 0.00L 0.00R	0.00 0.00	-0.04 0.00
RMLG UP	AOAT 6.08	TOP	0.00 0.00	-0.01 -0.01
	MACH 0.53	OUT	0.00 0.00	-0.02 0.00
		INB	0.00 0.00	-0.01 0.00
		PLA		56.91 57.00
		NWS		-0.09 NA

C 4: ROLL STICK POSN FAIL  
 C 4: RDR PDL FORCE FAIL  
 C 4: YAW TRIM SIGNAL FAIL  
 C 4: LT STAB RAM POS FAIL  
 C 4: LT STAB CAS POS FAIL  
 C 4: RT STAB RAM POS FAIL  
 C 4: RT STAB CAS POS FAIL  
 57 - 63  
 File = F18.LARSON/001

C 4: DISC WD 1 FAIL  
 C 4: DISC WD 2 FAIL  
 LAUNCH BAR DOWN  
 CAUTION RESET  
 LT PWR LEVER ANGLE INVALID  
 RFCS 18.1 LOADED  
 "ROLL RATE" CAUTION FLAG  
 64 - 70  
 Directory = F18 High Alpha Program Test Displays

72

FLAGS	FC/ATTITUDES/ACCS	COL-701E-DIF	L-RFCS-R	L-POS-R
701E MODE	VIAS 215.69	STB 0.00 0.00	0.00 0.00	1.23 1.23
	H 25500.	LEF 8.04 0.00	0.00 0.00	8.11 8.04
	MHDG 15.94	TEF 8.50 0.00	0.00 0.00	8.52 8.63
NLG UP	THA 6.05	AIL -0.09L 0.09R	0.00 0.00	-0.18 -0.01
LMLG UP	PHI 0.00	RUD 0.04L 0.04R	0.00 0.00	0.02 -0.09
RMLG UP	AOAT 6.06	TOP	0.00 0.00	-0.01 -0.01
	MACH 0.53	OUT	0.00 0.00	-0.01 0.00
		INB	0.00 0.00	-0.02 0.00
		PLA		56.91 57.00
		NWS		0.00 NA

C 4: "G LIMITER OFF" CAUTION FLAG  
 C 4: RT ENG BLEED DOOR CLOSED

71 - 77  
 File = F18.LARSON/001

78 - 84  
 Directory = F18 High Alpha Program Test Displays

72

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE October 1991	3. REPORT TYPE AND DATES COVERED Technical Memorandum	
4. TITLE AND SUBTITLE A Knowledge Based Application of the Extended Aircraft Interrogation and Display System			5. FUNDING NUMBERS  WU-	
6. AUTHOR(S)  Richard D. Glover and Richard R. Larson				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)  NASA Dryden Flight Research Facility P.O. Box 273 Edwards, California 93523-0273			8. PERFORMING ORGANIZATION REPORT NUMBER  H-1706	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)  National Aeronautics and Space Administration Washington, DC 20546-0001			10. SPONSORING/MONITORING AGENCY REPORT NUMBER  NASA TM-4327	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT  Unclassified — Unlimited Subject Category 62			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words)  A family of multiple-processor ground support test equipment developed at the NASA Dryden Flight Research Facility has been used to test digital flight-control systems on high-performance research aircraft. A unit recently built for the F-18 high alpha research vehicle project is the latest model in a series called the extended aircraft interrogation and display system. The primary feature emphasized in this report monitors the aircraft MIL-STD-1553B data busses and provides real-time engineering units displays of flight-control parameters. A customized software package was developed to provide real-time data interpretation based on rules embodied in a highly structured knowledge database. This report describes the configuration of this extended aircraft interrogation and display system briefly, and the evolution of the rule based package and its application to failure modes and effects testing on the F-18 high alpha research vehicle.				
14. SUBJECT TERMS Aircraft test equipment Real-time auto analysis Rules knowledge database			15. NUMBER OF PAGES 40	
			16. PRICE CODE A03	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT Unlimited	