

AD-A083 114

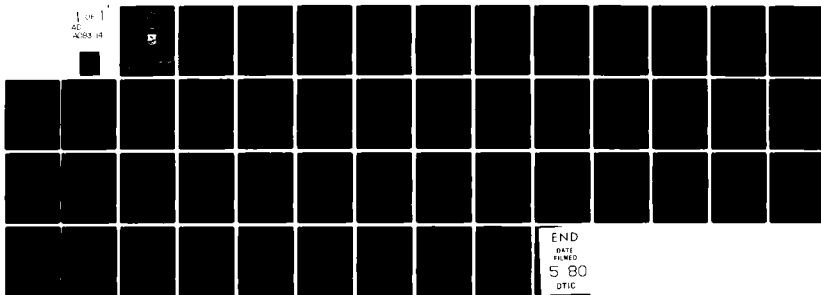
BOEING AEROSPACE CO SEATTLE WA BOEING MILITARY AIRPL--ETC F/6 9/2
COMPUTER PROGRAM DEVELOPMENT SPECIFICATION FOR IDAMST OPERATION--ETC(U)
NOV 76 F33615-76-C-1099
SPEC-58-4043 NL

UNCLASSIFIED

AFAL-TR-76-208-ADD-3

NL

1 UP
AD
A083 114



SPECIFICATION
SB 4043

LEVEL III

COMPUTER PROGRAM DEVELOPMENT SPECIFICATION
FOR IDAIST OPERATIONAL FLIGHT PROGRAMS
ERROR HANDLING AND RECOVERY SOFTWARE

Addendum 3.

ADA083114

Prepared by

✓ THE BOEING AEROSPACE COMPANY
BOEING MILITARY AIRPLANE DEVELOPMENT
SEATTLE, WASHINGTON

11

NOV 76

18 AFAL

19 TR-76-208-ADD-3

14 SPEC-SB-4043



12 49

15 F33615-76-2-2099

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

PREPARED FOR

AIR FORCE AVIONICS LABORATORY
AIR FORCE SYSTEM COMMAND
UNITED STATES AIR FORCE
WRIGHT-PATTERSON AFB, OHIO 45433

DTIC
ELECTE

APR 17 1980

E

DDC FILE COPY

440258 80 4 15 060

FOREWORD

This specification establishes the requirements for design and performance of the IDAMST Operational Flight Program, Error Handling and Recovery.

This specification has been prepared for the Air Force Avionics Laboratory under contract number F33615-76-R-1099, Specification for IDAMST Software. In particular, this specification incorporates guidelines established in the referenced contract, State of Work, Appendix H, Software Management Plan, Section 3.

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

TABLE OF CONTENTS

<u>SECTION</u>	<u>TITLE</u>	<u>PAGE</u>
	FOREWORD	ii
	TABLE OF CONTENTS	iii
	LIST OF FIGURES	vi
	LIST OF TABLES	vii
	ABBREVIATIONS	viii
1.0	SCOPE	1
1.1	IDENTIFICATION	1
1.2	FUNCTIONAL SUMMARY	1
1.2.1	ORGANIZATION	1
2.0	APPLICABLE DOCUMENTS	2
2.1	GOVERNMENT DOCUMENTS	2
3.0	REQUIREMENTS	5
3.1	COMPUTER PROGRAM DEFINITION	5
3.1.1	INTERFACE REQUIREMENTS	7
3.1.1.0.1	MISSION PROCESSOR INTERNAL ERROR EHAR, EQUIPMENT INTERFACE REQUIREMENTS	7
3.1.1.0.2	MULTIPLEX BUS ERRORS EHAR, EQUIPMENT INTERFACE REQUIREMENTS	7
3.1.1.0.2.1	COMPUTER INTERFACE	9
3.1.1.0.3	SUBSYSTEM SENSORS AND CONTROLS AND DISPLAYS ERRORS EHAR, EQUIPMENT INTERFACE REQUIREMENTS	9
3.1.1.1	INTERFACE BLOCK DIAGRAMS	12
3.1.1.1.1	MISSION PROCESSOR INTERNAL ERROR EHAR, INTERFACE BLOCK DIAGRAM	12
3.1.1.1.2	MULTIPLEX BUS ERRORS EHAR INTERFACE BLOCK DIAGRAM, MASTER PROCESSOR	12
3.1.1.1.2.1	MULTIPLEX BUS ERRORS EHAR INTERFACE BLOCK DIAGRAM, REMOTE PROCESSOR	12
3.1.1.1.3	SUBSYSTEM SENSORS, CONTROLS AND DISPLAYS ERROR HANDLING BY OFP APPLICATIONS	12
3.1.1.2	DETAILED INTERFACE DEFINITION	12
3.1.1.2.1	MISSION PROCESSOR INTERNAL ERRORS EHAR FUNCTIONAL RELATIONSHIPS	17
3.1.1.2.2	MULTIPLEX BUS ERRORS EHAR FUNCTIONAL RELATIONSHIPS	17
3.1.1.2.3	SUBSYSTEM SENSORS, CONTROLS AND DISPLAYS EHAR FUNCTIONAL RELATIONSHIPS	18
3.2	DETAILED FUNCTIONAL REQUIREMENTS	22
3.2.1	MISSION PROCESSOR INTERNAL ERRORS EHAR DETAILED REQUIREMENTS	22
3.2.1.1	INPUTS	22
3.2.1.2	PROCESSING	22
3.2.1.3	OUTPUTS	22
3.2.2	MULTIPLEX BUS COMMUNICATION ERRORS EHAR DETAILED REQUIREMENTS	22
3.2.2.1	INPUTS	27
3.2.2.2	PROCESSING, REMOTE AND MONITOR PROCESSOR	27
3.2.2.2.1	PROCESSING, MASTER PROCESSOR	27

TABLE OF CONTENTS (CONTINUED)

<u>SECTION</u>	<u>TITLE</u>	<u>PAGE</u>
3.2.2.3	OUTPUTS, REMOTE AND MONITOR PROCESSOR	28
3.2.2.3.1	OUTPUTS, MASTER PROCESSOR	28
3.2.3	SUBSYSTEM SENSORS, CONTROLS AND DISPLAYS	34
	ERRORS EHAR DETAILED REQUIREMENTS	
3.2.3.0.1	INPUTS AND OUTPUTS, GENERAL	34
3.2.3.0.2	PROCESSING, GENERAL	35
3.2.3.1	DISPLAY PROCESSES	35
3.2.3.1.1	INPUTS	36
3.2.3.1.2	PROCESSING	36
3.2.3.1.3	OUTPUTS	36
3.2.3.2	HANDLERS	37
3.2.3.2.1	INPUTS	37
3.2.3.2.2	PROCESSING	37
3.2.3.2.3	OUTPUTS	37
3.2.3.3	EQUIPS	40
3.2.3.3.1	GENERAL INPUTS AND OUTPUTS	40
3.2.3.3.2	UHF-AM OUTPUT PROCESSING	41
3.2.3.3.3	VHF-AM OUTPUT PROCESSING	41
3.2.3.3.4	VHF-FM OUTPUT PROCESSING	41
3.2.3.3.5	HF/SSB OUTPUT PROCESSING	41
3.2.3.3.6	ICS INPUT PROCESSING	41
3.2.3.3.7	ICS OUTPUT PROCESSING	41
3.2.3.3.8	P.A. INPUT PROCESSING	41
3.2.3.3.9	P.A. OUTPUT PROCESSING	41
3.2.3.3.10	DEK INPUT PROCESSING	41
3.2.3.3.11	DSMU OUTPUT PROCESSING	41
3.2.3.3.12	TACAN INPUT PROCESSING	42
3.2.3.3.13	TACAN OUTPUT PROCESSING	42
3.2.3.3.14	HCU INPUT PROCESSING	42
3.2.3.3.15	OMEGA INPUT PROCESSING	42
3.2.3.3.16	OMEGA OUTPUT PROCESSING	42
3.2.3.3.17	FCS INPUT PROCESSING	42
3.2.3.3.18	FCS OUTPUT PROCESSING	42
3.2.3.3.19	INS INPUT PROCESSING	42
3.2.3.3.20	INS OUTPUT PROCESSING	43
3.2.3.3.21	SKE/ZM INPUT PROCESSING	43
3.2.3.3.22	SKE/ZM OUTPUT PROCESSING	43
3.2.3.3.23	LF/ADF INPUT PROCESSING	43
3.2.3.3.24	LF/ADF OUTPUT PROCESSING	43
3.2.3.3.25	UHF/ADF INPUT PROCESSING	43
3.2.3.3.26	UHF/ADF OUTPUT PROCESSING	43
3.2.3.3.27	RADAR ALTIMETER INPUT PROCESSING	43
3.2.3.3.28	RADAR ALTIMETER OUTPUT PROCESSING	44
3.2.3.3.29	ILS INPUT PROCESSING	44
3.2.3.3.30	ILS OUTPUT PROCESSING	44
3.2.3.3.31	COMPASS INPUT PROCESSING	44
3.2.3.3.32	COMPASS OUTPUT PROCESSING	44
3.2.3.3.33	LR RADAR OUTPUT PROCESSING	44
3.2.3.3.34	FLIGHT SURFACES PROCESSING	44
3.2.3.3.35	AIRCRAFT SENSORS PROCESSING	44

TABLE OF CONTENTS (CONTINUED)

<u>SECTION</u>	<u>TITLE</u>	<u>PAGE</u>
3.2.4	SPECIAL REQUIREMENTS	49
3.2.4.1	JOVIAL J73	49
3.2.4.2	STRUCTURED PROGRAMMING	49
3.3	ADAPTATION	50
3.3.1	GENERAL ENVIRONMENT	50
3.3.1.1	IDAMST CORE ELEMENTS	50
3.3.1.2	OTHER IDAMST INTEGRATED HARDWARE	50
3.3.2	SYSTEM PARAMETERS	50
3.3.3	SYSTEM CAPACITIES	50
4.0	QUALITY ASSURANCE PROVISIONS	53
4.1	INTRODUCTION	53
4.2	COMPUTER PROGRAM VERIFICATION	54
4.2.1	PROGRAM ELEMENT TESTS	54
4.2.2	CPCI INTEGRATION TESTS	55
4.2.3	FORMAL SOFTWARE TESTING	55

Accession For	
NTIS GRA&I	☒
DDC TAB	☐
Unannounced	☐
Justification	
By _____	
Distribution/	
Classification Codes	
and/or	
Date	
A	

LIST OF FIGURES

<u>FIGURE NO.</u>	<u>TITLE</u>	<u>Page</u>
3.1-1	IDAMST ERROR DETECTION MECHANISMS	6
3.1-2	EHAR PROCESSOR INTERNAL ERROR HANDLING BY OPERATIONAL FLIGHT PROGRAM	13
3.1-3	EHAR MASTER BCIU/PROCESSOR ERROR HANDLING BY MASTER EXECUTIVE BCIU CONTROLLER	14
3.1-4	EHAR REMOTE/MONITOR BCIU & PROCESSOR ERROR HANDLING BY LOCAL EXECUTIVE BCIU MINOR CYCLE SETUP	15
3.1-5	EHAR SUBSYSTEM SENSORS, CONTROLS AND DISPLAYS ERROR HANDLING BY OFP APPLICATIONS SOFTWARE.	16
3.2-1A	EHAR PROCESSOR INTERNAL ERRORS	23
3.2-1B	EHAR BUS COMMUNICATION ERRORS (MASTER EXECUTIVE)	24
3.2-1B	EHAR BUS COMMUNICATION ERRORS (MASTER EXECUTIVE) (CONT.)	25
3.2-1C	EHAR BUS COMMUNICATION ERRORS (LOCAL AND MONITOR EXECUTIVE)	26

LIST OF TABLES

<u>TABLE NO.</u>	<u>TITLE</u>	<u>PAGE</u>
3.1-1	IDAMST MISSION PROCESSOR INTERNAL ERROR DETECTION	8
3.1-2	IDAMST OFF APPLICATIONS PROCESSES INCORPORATING EHAR FUNCTIONS	11
3.2-1	IDAMST SUBSYSTEMS CONTAINING BIT	34
3.3-1	IDAMST MISSION PROCESSORS CAPACITY	51

ABBREVIATIONS

ADF	Automatic Direction Finder
AFAL	Air Force Avionics Laboratory
AMST	Advanced Medium STOL Transport
BCIU	Bus Controller Interface Unit
BIT	Built In Test circuitry
CARP	Computed Air Release Point
CCA	Column Control Assembly
CCIP	Continually Computed Impact Point
CCT	Combat Control Team
CRT	Cathode Ray Tube
DAIS	Digital Avionics Information System
DEK	Data Entry Keyboard
DITS	Digital Integrated Test System
DS/MU	Display Switch/Memory Unit
ECM	Electronic Counter Measure
EFCS	Electronic Flight Control System
EHARS	Error Handling and Recovery Software
FCS	Flight Control System
GMT	Greenwich Mean Time
IDAMST	Integrated Digital Avionics for a Medium STOL Transport
IFF/SIF	Identification Friend or Foe/Selective Identification Feature
ILS	Instrument Landing System
HCU	Hand-Controller Unit
HF/SSB	High Frequency/Single Side Band
HSD	Horizontal Situation Display
HUD	Head-Up Display
IMK	Integrated Multifunction Keyboard
INS	Inertial Navigation System
LAPES	Low Altitude Parachute Extraction System
MCL	Master Caution Light
MDSC	Modular Digital Scan Converter

MFDC	Multi-Function Display Controls
MMK	Master Mode Keyboard
MMU	Mass Memory Unit
MPD	Multi-Purpose Display
MPDG	Modular Programmable Display Generator
OPF	Operational Flight Program (Software)
OPS	Operational Sequencer
RF	Radio Frequency
FTU	Remote Terminal Unit
SCP	Sensor Control Panel
SKE	Station Keeping Equipment
STOL	Short Take-Off and Landing
TACAN	Tactical Air Navigation
T/R	Transmit/Receive
T/R+G	Transmit/Receive Plus Guard
UHF	Ultra High Frequency
VHF	Very High Frequency
VLF	Very Low Frequency
ZM	Zone Marker

1.0 SCOPE

1.1 IDENTIFICATION

This part of this specification establishes the requirements for performance, design, test and qualification of computer programs identified as IDAMST Operational Flight Program, Error Handling and Recovery, IDAMST-OFPEHAR, CPCI number TBD. This CPCI is used to provide the IDAMST system with a pre-planned response to anomalies that occur within the system.

1.2 FUNCTIONAL SUMMARY

The purpose of this specification is to establish the functional and design requirements for the IDAMST Error Handling and Recovery computer programs. This document is intended to compliment the IDAMST OFP Executive Specification, IDAMST specification SB-4041, July 1976. Error detection and error recovery strategy has been treated, in this document, separate from the OFP requirements for functional capability.

IDAMST OFP EHARS is specified for three major functional areas: Response for errors occurring internal to the IDAMST mission processors, including computer program induced errors, response for errors occurring in the multiplex connecting subsystems of IDAMST, and response for errors occurring in subsystem sensors, actuators, and crew station controls and displays. Within each functional area, the EHARS software shall accomplish the functions of: Determining that an error of sufficient severity to degrade system performance has occurred, isolate the suspect subsystem, notify the flight crew/IDAMST top level control program where appropriate to configure the faulty subsystem/function out of service.

1.2.1 Organization

This specification is organized in such manner as to address IDAMST system responses to three major categories of errors. These categories of errors are Mission Processor internal errors, multiplex bus core element errors and subsystem sensor and controls and displays errors. Each major paragraph of the document is organized in three subparagraphs, each addressing software requirements for system response to one of the three major categories of errors.

- 2.0 APPLICABLE DOCUMENTS
- 2.1 GOVERNMENT DOCUMENTS
 - 2.1.1.a Appendices to Contract F33615-76-C-1099 Statement of Work (SOW).
 - 2.1.1.b Appendix A - "AMST Mission Profile and Scenario (Updated)"
 - 2.1.1.c Appendix C - "System Architecture".
 - 2.1.1.d Appendix E - "DAIS Mission Software, OFP Applications (SA-201-303)", 17 June 76.
 - 2.1.1.e Appendix F - "DAIS Mission Software, Executive (SA-201-320)", 26 Dec. 75.
 - 2.1.1.f Appendix H - "Software Management Plan"
 - 2.1.1.g Appendix M - "TRW System Backup and Recovery Strategy (TRW 6404-5-6-06)", Sept. 75.
 - 2.1.2 DAIS Documents (Reference)
 - 2.1.2.a ICD - Mission Operation Sequence: Pilot/Controls and Displays/Interface with Application Software (SA-803-200), 15 Mar. 76
 - 2.1.2.b Mission Software/Controls and Displays Interface (SA-802-301), 12 Mar. 76.
 - 2.1.2.c DAIS System Control Procedures (SA-100-101 Appendix A), 7 Nov. 75.
 - 2.1.3 IDAMST Documents (Program Generated)
 - 2.1.3.a Computer Program Development Specification, IDAMST OFP Executive (SB 4041), July 76.
 - 2.1.3.b Computer Program Development Specification, IDAMST OFP Applications (SB 4043), July 76.
 - 2.1.3.c Computer Program Development Specifications, IDAMST Operational Test Program (SB 4044), July 76.
 - 2.1.4 IDAMST Documents (Reference)

The following documents because of release dates server only as reference documentation for this specification; however, are considered prime to further definition of the IDAMST system design.

 - 2.1.4.a System Specification for IDAMST, Type A (S1-101-), June 76.

- 2.1.4.b Prime Item Development Specification IDAMST Processor,
Type B1 (SI 4030), June 76.
- 2.1.4.c System Segment Specification, IDAMST Control/Display Subsystem,
Type A (SI 5020), June 76.
- 2.1.4.d System Specification, IDAMST Information Transfer System, Type A
(SS 3020), May 76.
- 2.1.4.e Prime Item Development Specification, IDAMST Remote Terminal,
Type B1 (SS 3130), May 76.
- 2.1.4.f Prime Item Development Specification, IDAMST Bus Control Interface,
Type B1 (SS 3230), May 76.

This sheet intentionally left blank.

3.0 Requirements

The performance and design requirements of the IDAMST, Operational Flight Program (OFP) Error Handling and Recovery (EHAR) software are described in this section.

Paragraph 3.1 identifies equipment and computer program interfaces and relationship to IDAMST OFP architecture.

Paragraph 3.2 describes detailed requirements of the EHAR computer program.

3.1 Computer Program Definition

Figure 3.1-1 indicates the nature of error detection mechanisms within the IDAMST system. An idealized subsystem sensor containing built-in test circuitry, core element remote terminal, BCIU, mission processor, operational flight program and idealized display subsystem are shown. Associated with each system element are noted the type of error detection implemented in that class of elements. At the bottom of the figure, the OFP program functional area responsible for handling the detected errors is indicated.

Code implemented within the Operational Flight program applications programs are responsible for detection of data reasonableness and life errors, improper subsystem sensor sequences and operating modes. The OFP applications code is also tasked with making comparisons between results obtained from redundant and similar function (e.g., Omega and INS derived position) subsystems and is tasked with reasonable rate and response time tests of data where applicable.

EHAR processing within the OFP Local Executive is tasked with handling of mission processor and memory BIT detected errors. Cooperating EHAR code is required in the Master/Monitor executive to respond to processor internal errors in other (remote) processors. EHAR response to processor internal errors is to cause the processor in which the error occurred to become passive on the multiplex bus. Cooperating EHAR response in the Master/Monitor processor must be able to respond to error response from the processor that experiences the internal error.

EHAR processing within the BCIU Controller of the Master Executive is tasked with handling of errors detected by the multiplex core element BIT. Communication errors detected by the multiplex units include data transmission errors, multiplex bus message content or sequence errors, BCIU or Remote terminal failures and incorrect operations. Also certain BIT error signals from subsystems are reflected into the multiplex as error signals at the Master BCIU.

Errors detected in the subsystem sensors are handled by the OFP applications software when those errors are not reflected into the multiplex error signals. When reflected into the multiplex error signals (i.e. RT status word BIT error bits) EHARS code within the Master Executive BCIU controller must take action to allow continued operation of unaffected portions of the system until the error detected signal is resolved.

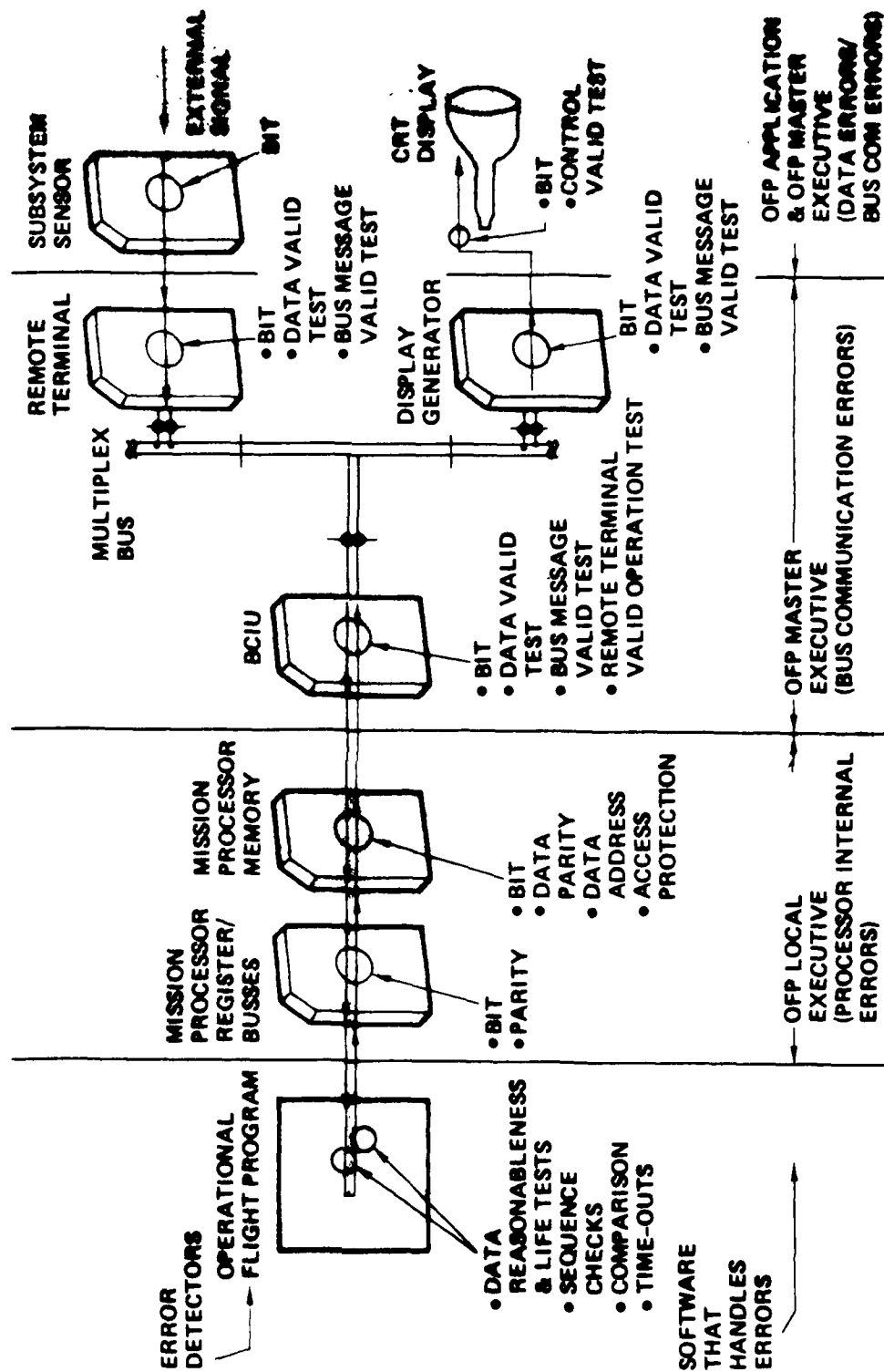


Figure 3.1-1 IDAMST Error Detection Mechanisms

3.1.1 Interface Requirements

The three functional areas within the IDAMST EHARS software, mission processor internal errors, multiplex bus errors and subsystem sensors/controls and displays each have unique equipment interface requirements. This paragraph describes interface requirements for each of the Functional areas.

3.1.1.0.1 Mission Processor Internal Error EHAR, Equipment Interface Requirements

Built-in test (BIT) circuitry within the IDAMST mission processor provides hardware error detection mechanism for errors occurring internal to the mission processors. Notification of error detection is communicated to the IDAMST OFP software by activation of the mission processor interrupt signals. These interrupt signals cause suspension of executing programs immediately upon completion of an executing machine operation code. Occurrence of an interrupt automatically causes start of execution of the OFP Local Executive Interrupt handler.

Table 3.1-1 lists the error detection within the mission processor and the priority level of the resulting interrupt signal. Highest priority is level sixteen. All interrupts except level sixteen can be disabled by execution of the processor interrupt disable instruction. Interrupts can be selectively disabled (except for level sixteen) by execution of the processor set interrupt control (SIC) instruction.

Those interrupt signals marked with an asterisk (*) represent detection of a processor internal error.

Hardware interval timers, interval timer A and B, are used in conjunction with OFP Local and Master executive code to provide timeout tests.

3.1.1.0.2 Multiplex Bus Errors EHAR, Equipment Interface Requirements

The multiplex bus equipment within the IDAMST system includes the following equipment:

- o Dual-channel twisted-pair communications medium (MIL-STD-1553A Multiplex Bus)
- o Three Bus Control Units interfacing one mission processor each to the dual channel multiplex (IDAMST Specification SS-3020 June 1976)
- o Eleven remote terminal units interfacing 130 subsystem interface modules (via approximately 200 subaddresses) to the dual channel multiplex (IDAMST Specification No. SS-3130, May 1976)

Table 3.1-1

IDAMST Mission Processor Internal Error Detection

<u>Interrupt Priority Level</u>	<u>Description</u>
1	BCI interrupt #8
2	spare
3	BCI interrupt #7
4	* illegal operation code
5	BCI interrupt #6
6	* boundary alignment error
7	BCI interrupt #5
8	interval timer B
9	BCI interrupt #4
10	interval timer A
11	BCI interrupt #3
12	* processor parity error
13	BCI interrupt #2
14	* processor memory protect
15	BCI interrupt #1
16	* power down

Note: *designates mission processor internal errors.

3.1.1.0.2.1 Computer Interface

IDAMST OFP EHARS software interfaces to the multiplex equipment via the Bus Control Interface Unit (BCIU) and via status data placed into data compools by the BCIU. EHARS software receives control on occurrence of certain BCIU generated processor interrupt signals. EHARS software can effect operation of the BCIU by either directly loading information into BCIU internal registers or by placing BCIU commands into the Bus Message Lists of the OFP Master Executive Bus Control task.

EHARS software can affect operation of the remote terminal (RT) by placing RT mode operation commands into the Bus Message Lists of the OFP Master Executive. EHARS software can determine status of the RT's by obtaining the bus message status word (any message involving the RT) from the Master BCIU or by placing RT mode commands into the Bus Message List of the Master BCIU. Mode Commands that allow access to RT status are:

- o Transmit RT status word
- o Transmit RT Built-In Test (BIT) word
- o Transmit RT Last Command Register
- o Interrogate Activity Register
- o Interrogate Interface Module Error Register

Execution of these RT mode commands by the Master BCIU causes the indicated RT status information to be placed in a data compool in the Master Mission processor memory (or a BCIU error interrupt to the master mission processor if an anomaly or fault exists)

IDAMST specification number SS-3130, May 1976 details RT mode command and status information. IDAMST specification number SS-3020, June 1976 details BCIU to mission processor interfaces. MIL-STD-1553A details multiplex bus operation.

3.1.1.0.3 Subsystem Sensors and Controls and Displays Errors EHARS Equipment Interface Requirements

IDAMST OFP EHARS responsible for handling errors occurring in subsystem sensors, controls and displays is an integral part of the IDAMST OFP Applications programs. Those programs of the OFP Applications programs that interface with the IDAMST equipment are the EQUIPMENT processes (EQUIPS), DISPLAY processes (DISPS), and HANDLER processes (HANDLERS). These equipment interface processes of the OFP Applications software incorporate EHARS code. OFP Applications EHARS is responsible for life and reasonableness tests of mission processor input and output data, comparison of redundant subsystems data and comparison of similar function subsystems data.

OFP Applications EHARS interfaces to the IDAMST subsystems equipment through the OFP Applications EQUIPS, DISPS and HANDLERS. IDAMST OFP Applications specification, Document number SB 4043, July 1976, Paragraph 3.1.1 is incorporated herein by reference to establish interface requirements for the OFP Applications EHARS.

Table 3.1-2 lists the OFP Applications processes that incorporate EHARS functions.

TABLE 3.1-2
IDAMST OFP APPLICATIONS PROCESSES
INCORPORATING EHAR FUNCTIONS

DISPLAY PROCESSES (DISP'S)

HUD
HSD
Instruments
Lights
MPD Checklist
MPD Par/Status
MPD Error/Warn
IMK Fixed Text
IMK Status
DEK Mark

HANDLER PROCESSES

IMK Handler
MPD Handler

EQUIPMENT PROCESSES (EQUIP'S)

UHF-AM
VHF-AM
VHF-FM
HF/SSB
ICS Input
ICS Output
P.A. Input
P.A. Output
DEK Input
DSMU Output
TACAN Input
TACAN Output
HCU Input
OMEGA Input
OMEGA Output
FCS Input
FCS Output

INS Input
INS Output
SKE/ZM Input
SKE/ZM Output
LF/ADF Input
LF/ADF Output
UHF/ADF Input
UHF/ADF Output
Radar Altimeter Input
Radar Altimeter Output
ILS Input
ILS Output
Compass Input
Compass Output
LR Radar Output
Flight Surfaces
Aircraft Sensors

3.1.1.1 Interface Block Diagram

The functional interfaces of the IDAMST OFP EHAR software is graphically portrayed in this paragraph. The interfaces for the three functional areas of EHARS, processor internal errors, multiplex errors and subsystem sensors/controls and displays are portrayed in separate subparagraphs.

3.1.1.1.1 Mission Processor Internal Error EHAR Interface Block Diagram

Figure 3.1-2 portrays the major interfaces to Mission Processor internal errors EHAR. EHAR receives control from the OFP Local Executive interrupt handler as a result of a mission processor BIT hardware error detection. EHARS receives control from the OFP Master Executive on occurrence of watchdog time-out. EHAR sets the BCIU Status Register (SR), bits 0-9 to the terminal fail state via the OFP Master Executive BCIU Interface (Master Processor) or via the OFP Local Executive Minor Cycle Set-up (Remote Processor). EHAR also ensures that the Go bit of the BCIU Processor Control Register (PCR) is set to zero.

3.1.1.1.2 Multiplex Bus Errors EHAR Interface Block Diagram, Master Processor

Figure 3.1-3 portrays the interfaces to multiplex bus errors EHAR in the Master Processor. In the Master Processor, EHAR interfaces to the BCIU via the Master Executive BCIU interface. EHAR performs alternate bus operation, retrieves status from remote devices or effects message retry via the Master Executive BCIU Command List Handler. EHAR receives control on occurrence of certain BCIU hardware interrupt signals.

3.1.1.1.2.1 Multiplex Bus Errors EHAR Interface Block Diagram, Remote Processor

Figure 3.1-4 portrays the interfaces to multiplex bus errors EHAR in the remote or monitor mission processors. EHAR receives control on occurrence of certain BCIU hardware interrupt signals. EHAR sets terminal failure indication in a remote mode BCIU via the program input/output BCIU interface of the local executive minor cycle setup.

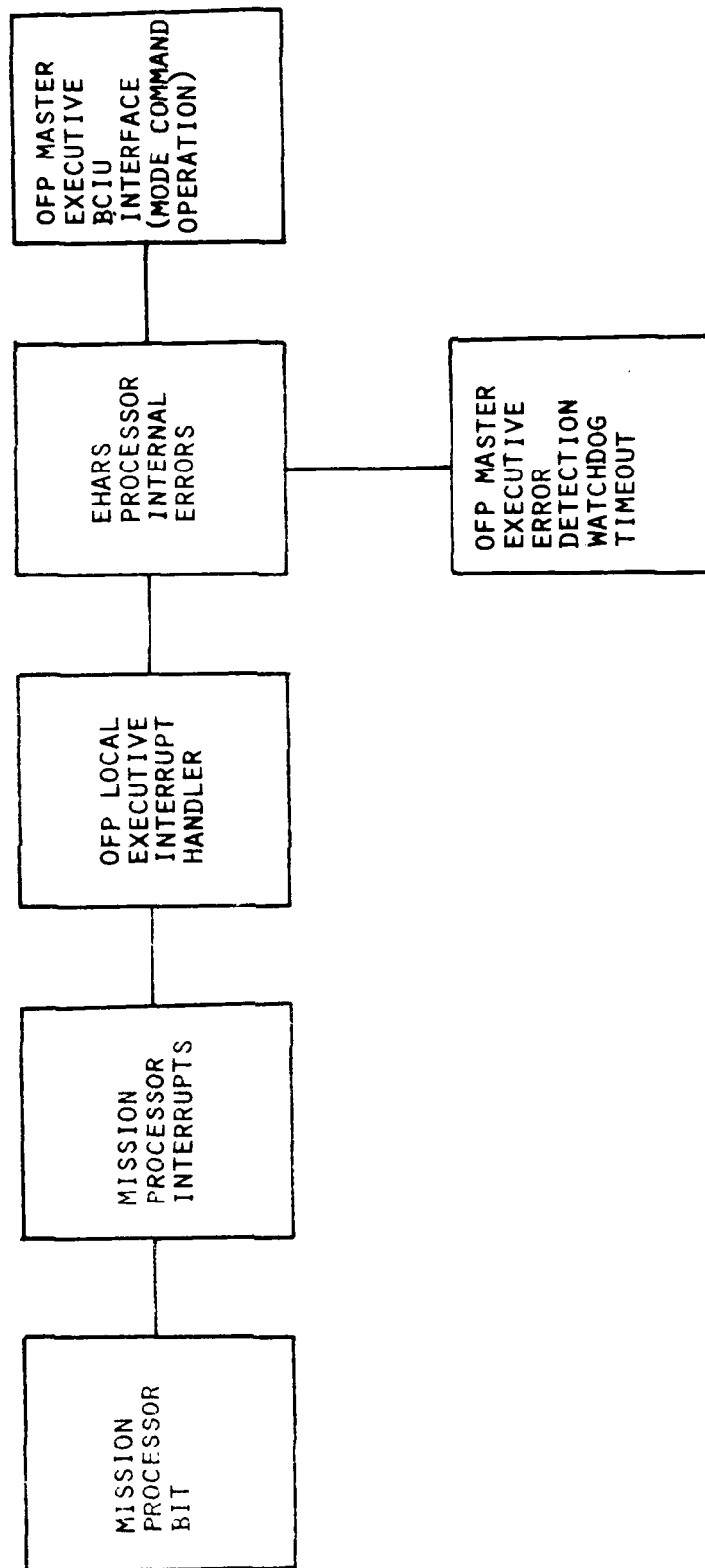
3.1.1.1.3 Subsystem Sensors, Controls and Displays Errors EHAR Interface Block Diagram

Figure 3.1-5 portrays the interfaces to subsystem sensors, controls and displays errors EHAR. EHAR functions are an integral part of the OFP Applications equipment processes, (EQUIPS) display processes (DISPS) and HANDLERS. As an integral part of the OFP Applications, EHAR interfaces are shared with the Applications processes, EHAR interfaces with hardware subsystems and other IDAMST software subsystems via data compools. The OFP Applications processes are activated by the OFP Applications Configurator. Applications processes commence execution based upon predetermined event conditions.

3.1.1.2 Detailed Interface Definition

The functional relationships of the IDAMST EHAR to IDAMST hardware and computer programs is specified in this paragraph. The three functional areas of mission processor internal errors, multiplex bus errors and subsystem sensors and controls

Figure 3.1-2 EHARS PROCESSOR INTERNAL ERROR HANDLING
BY OPERATIONAL FLIGHT EHARS



ENHANCED MASTER BCIU/PROCESSOR

Figure 3.1-3

ERROR HANDLING BY MASTER EXECUTIVE BCIU CONTROLLER

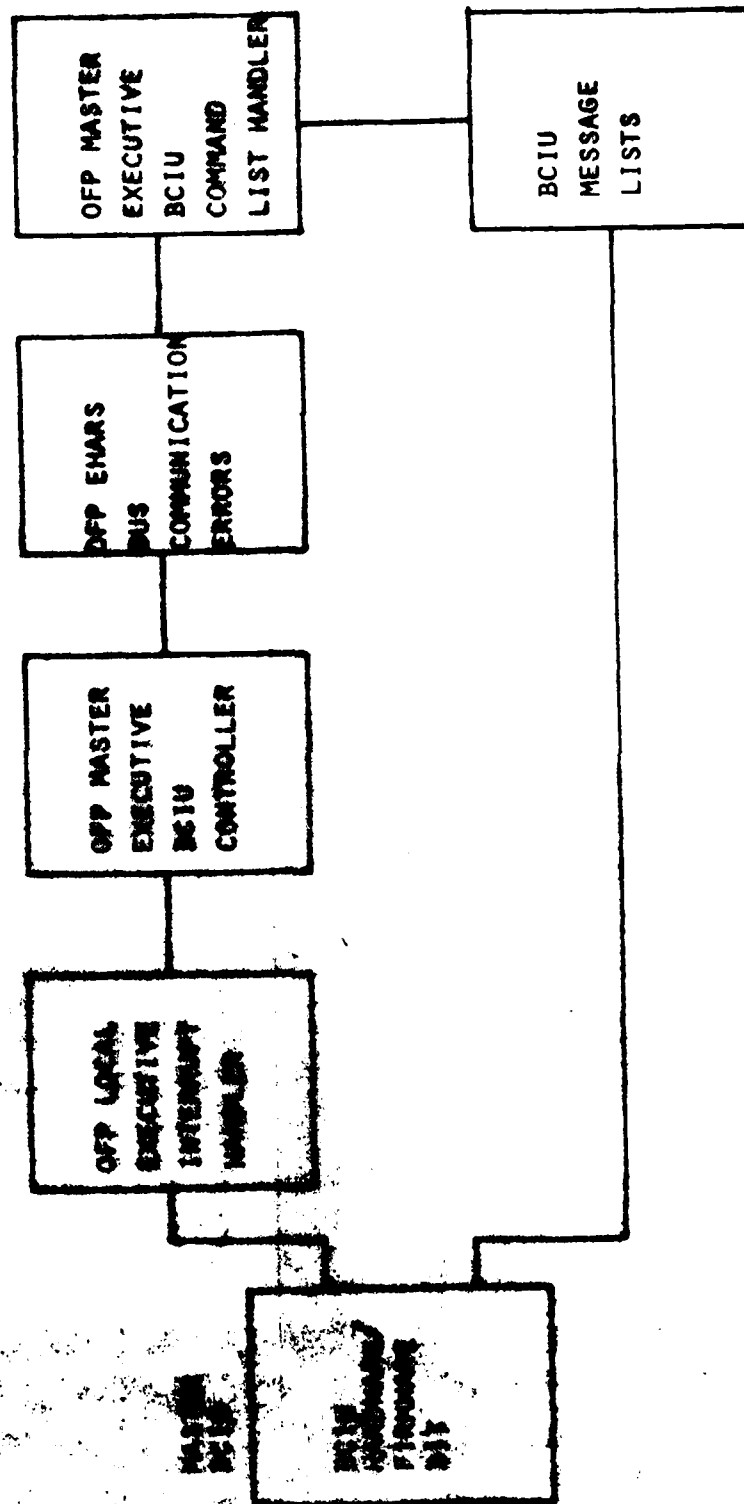
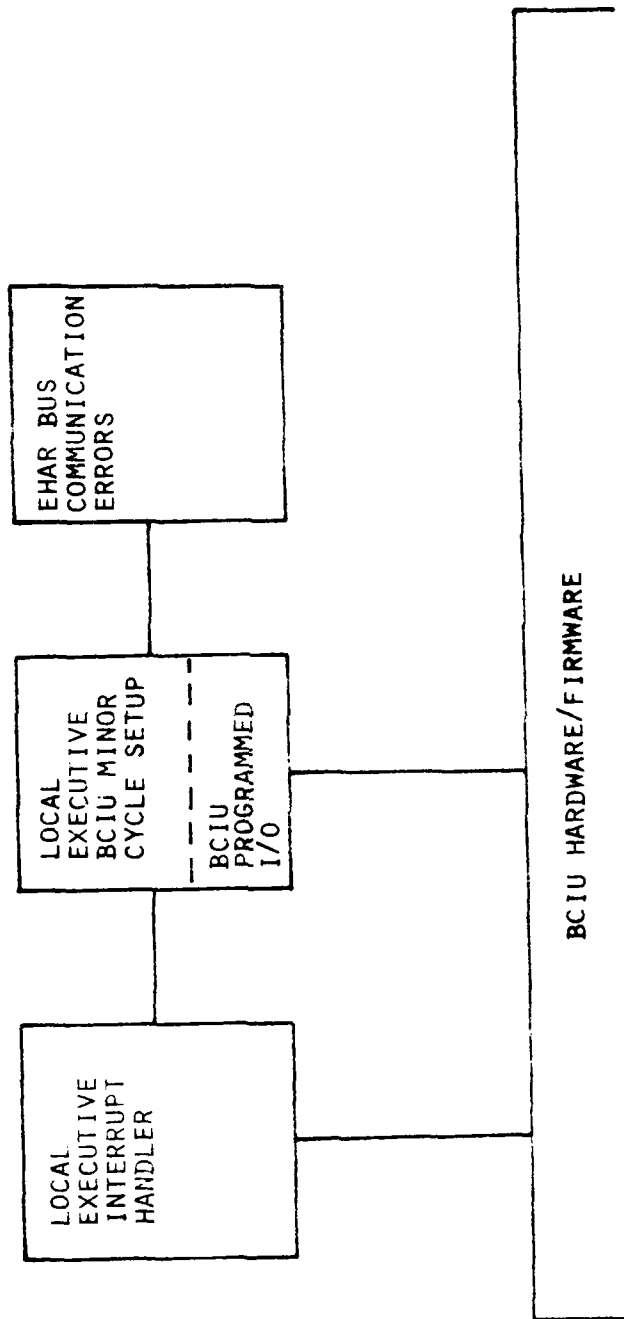


Figure 3.1-4 Error Handling by Local Executive BCIU Minor Cycle Setup
REMOTE/MONITOR BCIU & PROCESSOR



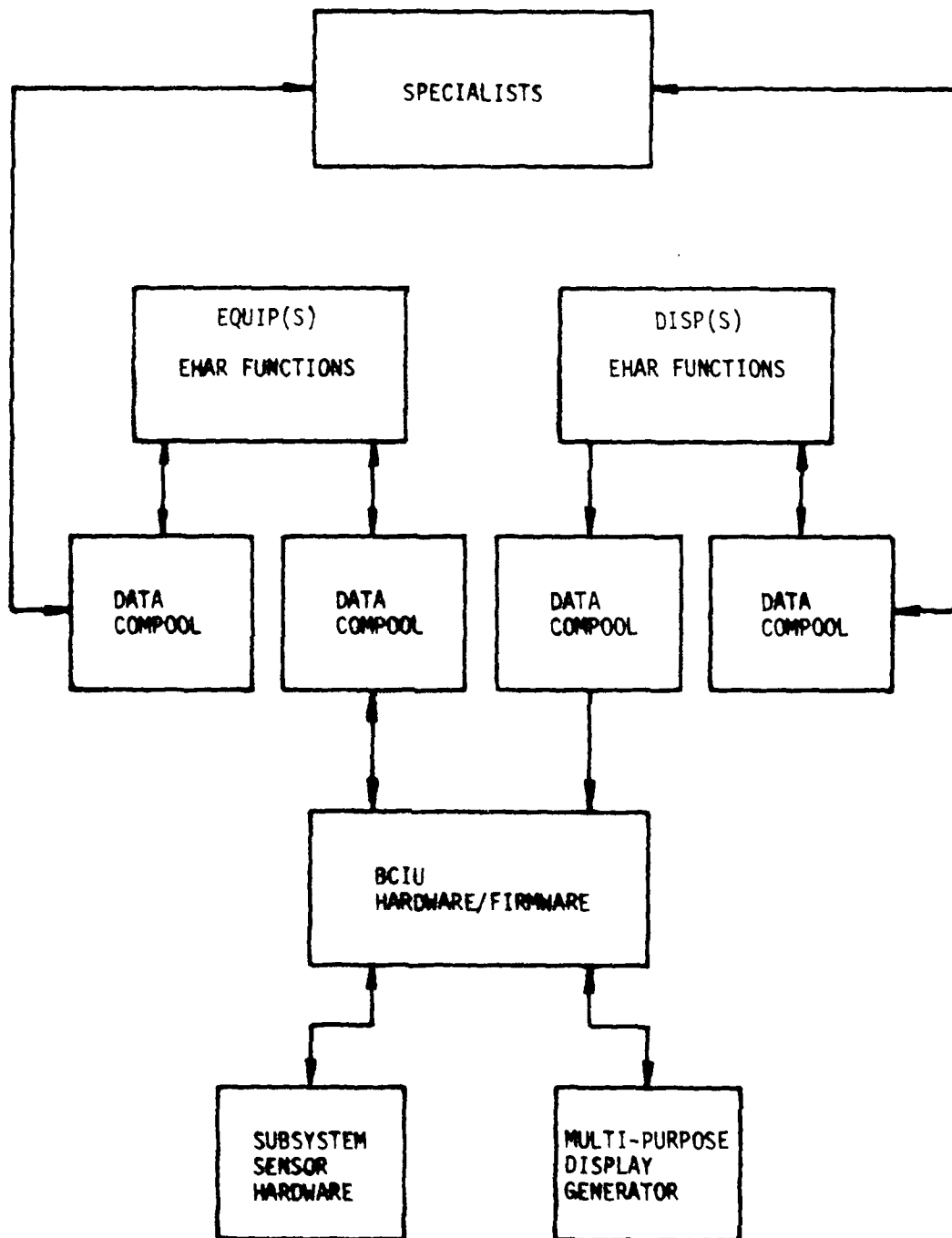


FIGURE 3.1-5. EHR Subsystem Sensors, Controls and Displays Error Handling, by OFP Applications Software

and displays errors are specified in subparagraphs 3.1.1.2.1 through 3.1.1.2.3 respectively.

3.1.1.2.1 Mission Processor Internal Errors EHAR Functional Relationships

On occurrence of mission processor BIT hardware error detection, a mission processor is removed from service in a passive manner. A suspect failed processor does not attempt bus communication on detection of internal error. This approach is taken for the IDAMST system to prevent a failed processor from propagating errant information into other elements of the system. Mission Processor internal error EHAR receives control directly from the OFP Local Executive interrupt handler. The interrupt handler vectors mission processor internal BIT interrupt signals to EHAR. EHAR effects a passive shutdown of the suspect processor by stopping processor participation in multiplex bus communication. This is effected by setting a terminal failed code into the BCIU Status Register (SR) and by resetting Go in the BCIU Processor Control Register (PCR).

If the errant processor is the multiplex Master processor, two minor cycles time-out in the monitor processor will cause the Monitor to take over as multiplex master. If the errant processor is a remote processor, passive shutdown will cause Multiplex Bus errors EHAR to execute in the Master Processor. A terminal failed code is returned to the master BCIU on the first message directed to the errant mission processor/BCIU.

Error detection BIT, timing of interrupt signals and the mission processor, software interfaces are specified in IDAMST specification number SI 4030, Prime Item Development Specification for IDAMST Processor, Type B1, June 1976, which is incorporated by reference as a portion of this specification.

3.1.1.2.2 Multiplex Bus Errors EHAR Functional Relationships

On failure to successfully complete a bus message, including failure of automatic retries, the BCIU generates a hardware interrupt signal to its associated mission processor. The OFP Local Executive Interrupt handler vectors BCIU interrupts to the OFP Master Executive, BCIU Controller in the case of the Master BCIU/processor. The OFP Local Executive Interrupt handler vectors BCIU interrupts to the OFP Local Executive, Minor Cycle Setup in the case of the monitor or remote BCIU/processor. OFP Master Executive BCIU controller via programmed input/output interface retrieves the BCIU internal status register to assist in interpretation of the interrupt cause. The OFP Local Executive Minor Cycle Setup retrieves only Minor Cycle number so that the remote and monitor processor bus error EHAR has responsibility for retrieving internal status register data. Bus errors EHAR interprets status data and interrupt signal identity to determine a possible cause for failure to complete the errant bus message. EHAR chooses an option of retry of the original message via the alternate bus, testing of the BCIU to mission processor interface or of declaring the BCIU faulty and causing a passive shutdown of the BCIU/processor. EHAR in the Master processor can resolve ambiguous situations (e.g., BCIU indicates no bus message status word returned) by requesting additional information from remote terminals/BCIU's. EHAR retrieves additional information by specifying mode command message operation in the BCIU controller. Mode command operations are treated as immediate asynchronous messages.

The processor to BCIU programmed input/output interface, direct memory access interface, mode command operation and status information available to EHAR in the BCIU internal registers are specified in specification SS-3200, Prime Item

Development Specification for IDAMST Bus Control Interface Unit (BCIU), Type B1, March 1976, which is included by reference as a portion of this specification. Bus communication errors EHAR specifies error isolation to a subsystem, Remote Terminal or BCIU/mission processor by interface with the Subsystem Status Monitor. EHAR specifies the identity of the declared failed unit to the Subsystem Status Monitor. The Subsystem Status Monitor has the system level information necessary to correlate errors to localize failed units, (e.g., the Subsystem Status Monitor can correlate that multiple errors are occurring in communication with several subsystems connected to the same remote terminal). The Subsystem Status Monitor effects system/subsystem reconfiguration by interface to the reconfiguration routines. Subsystem Status Monitor determines that error thresholds have been exceeded, formats mission recorder entries for errors, signals crew warnings/cautions via the OFP Applications DISPS, and if reconfiguration is indicated, invokes the task reconfiguration logic of the configurator.

3.1.1.2.3 Subsystem Sensors, Controls and Displays EHAR Functional Relationships

EHAR functions for subsystem sensors, controls and displays are incorporated in the OFP Applications EQUIPS, DISPS and Handlers. The OFP Applications processes provide the EHAR to IDAMST interfaces. The OFP Applications interfaces and functional relationships are specified in document number SS-4043 IDAMST OFP Applications Specification.

EHAR functions require an interface between the OFP Applications processes and the Subsystem Status Monitor. On determination of a subsystem error by EHAR functions, the Subsystem Status Monitor is notified of the error occurrence and suspect subsystem identity. The Subsystem Status Monitor formats a mission recorder entry, signals crew warnings/cautions via the OFP Applications DISPS where appropriate, and effects subsystem reconfiguration via the task configuration logic of the configurator.

3.2 DETAILED FUNCTIONAL REQUIREMENTS

Figure 3.2-1 depicts the functional block diagram of the IDAMST EHAR functions portrayed in structured design representation. The different functional areas of EHAR are represented in three separate diagrams. Figure 3.2-1A depicts the EHAR functions associated with mission processor internal errors. Figure 3.2-1B depicts the EHAR functions associated with multiplex bus communications errors detected in the Master BCIU/processor. Figure 3.2-1C depicts the EHAR functions associated with multiplex bus or BCIU errors detected in a remote BCIU/processor.

3.2.1 Mission Processor Internal Errors EHAR Detailed Requirements

Mission Processor Internal Error EHAR is responsible for providing the IDAMST system with a preplanned response to anomalies that affect the internal operation of the IDAMST mission processor. After detection of a processor internal error by the processor BIT hardware, EHAR ensures that the processor is isolated from the rest of the IDAMST elements to prevent propagation of suspect data or control signals throughout the system.

3.2.1.1 Inputs

Processor control is vectored directly to EHAR mission processor internal errors on occurrence of a mission processor BIT failure by the OFP Local Executive Interrupt handler. EHAR receives the identity of the interrupt signal. A list of interrupt signals is given in Table 3.1-1.

3.2.1.2 Processing

EHAR sets control and status bits in the BCIU internal registers via the BCIU programmed input/output interface. EHAR must determine if the BCIU is a remote mode BCIU. If remote, EHAR must busy out the BCIU to prevent ambiguous status indication at the master BCIU. EHAR sets the Busy Bit in the BCIU Processor Control Register, waits 700 microseconds (length of longest message that could currently be in progress) and then sets the terminal failed code in the BCIU Internal Status Register and resets the Go and Busy bits in the Processor Control Register.

3.2.1.3 Outputs

EHAR directly sets the contents of the BCIU internal registers. EHAR controls the Processor control Register bits Busy and Go. These bits are used internally by the BCIU to control BCIU actions. EHAR sets the Terminal Failed status code in the BCIU internal status register (bits 00-09). The BCIU internal status register is used by Multiplex Bus errors EHAR in the Master Processor to determine why communication with the failed processor has ceased.

3.2.2 Multiplex Bus Communication Errors EHAR Detailed Requirements

Multiplex bus communication errors EHAR is responsible, in cooperation with the BCIU/RT hardware and firmware, for providing the IDAMST system with a preplanned response to anomalies that effect the operation of message communication on the MIL-STD-1553/multiplex bus. After detection of a bus message error by the multiplex hardware BIT, EHAR first attempts to complete transmission of the message and if unsuccessful isolates the suspect unit from the system to prevent degradation of system performance and propagation of errant data through the system.

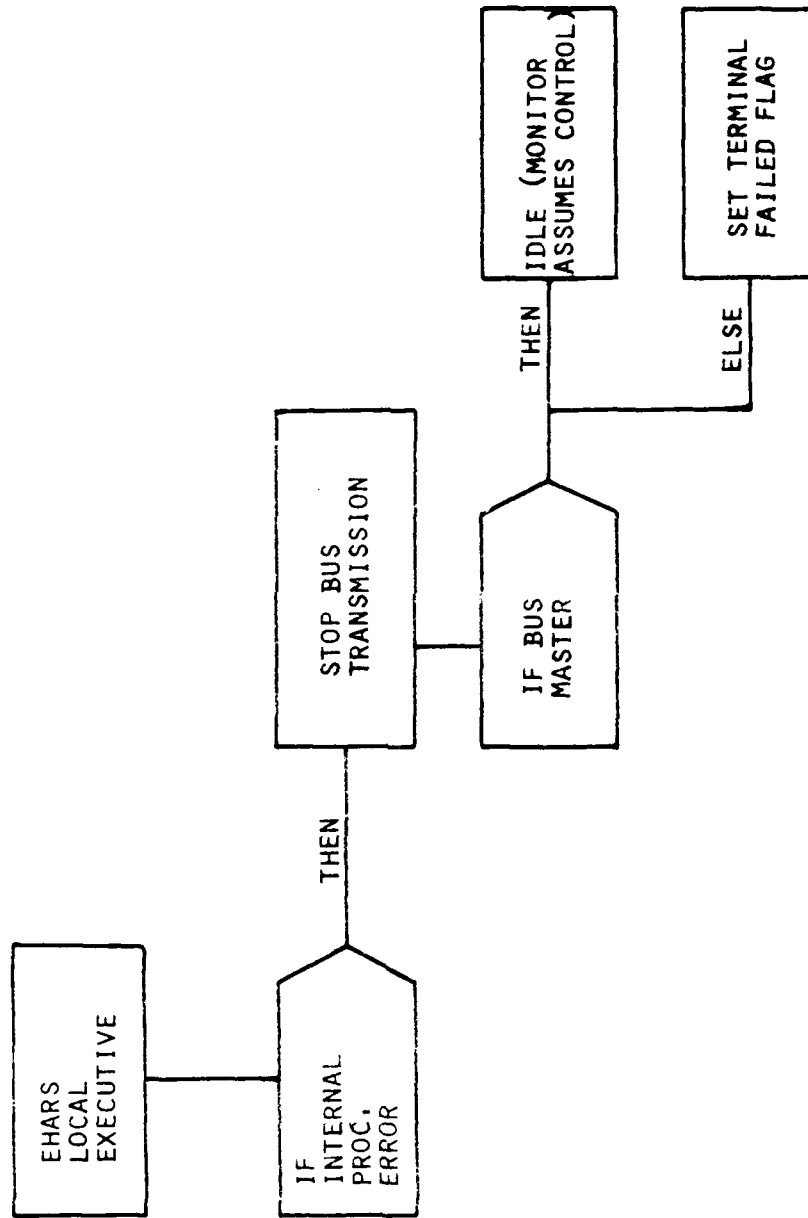


Figure 3.2-1A EHAR PROCESSOR INTERNAL ERRORS

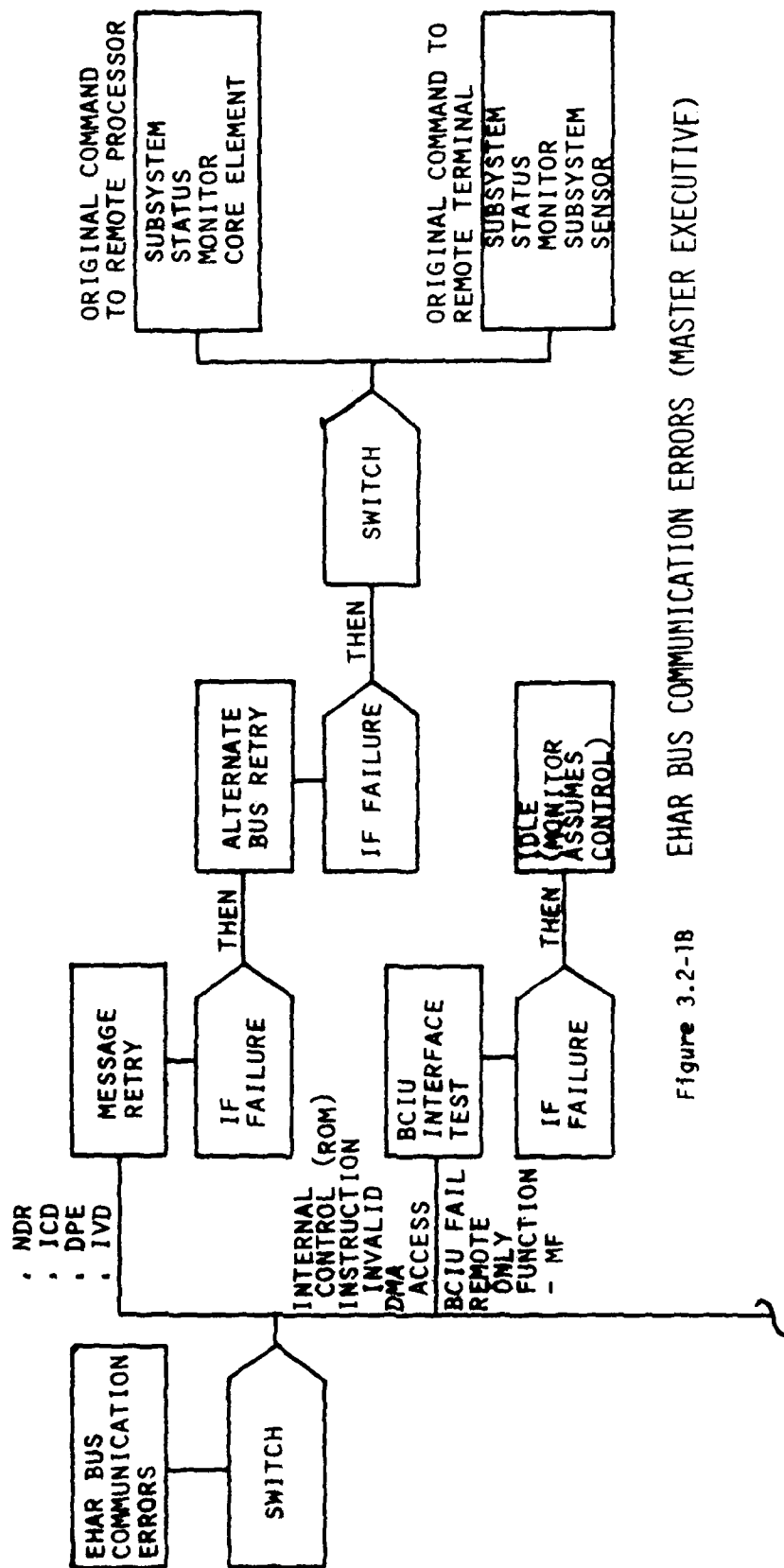
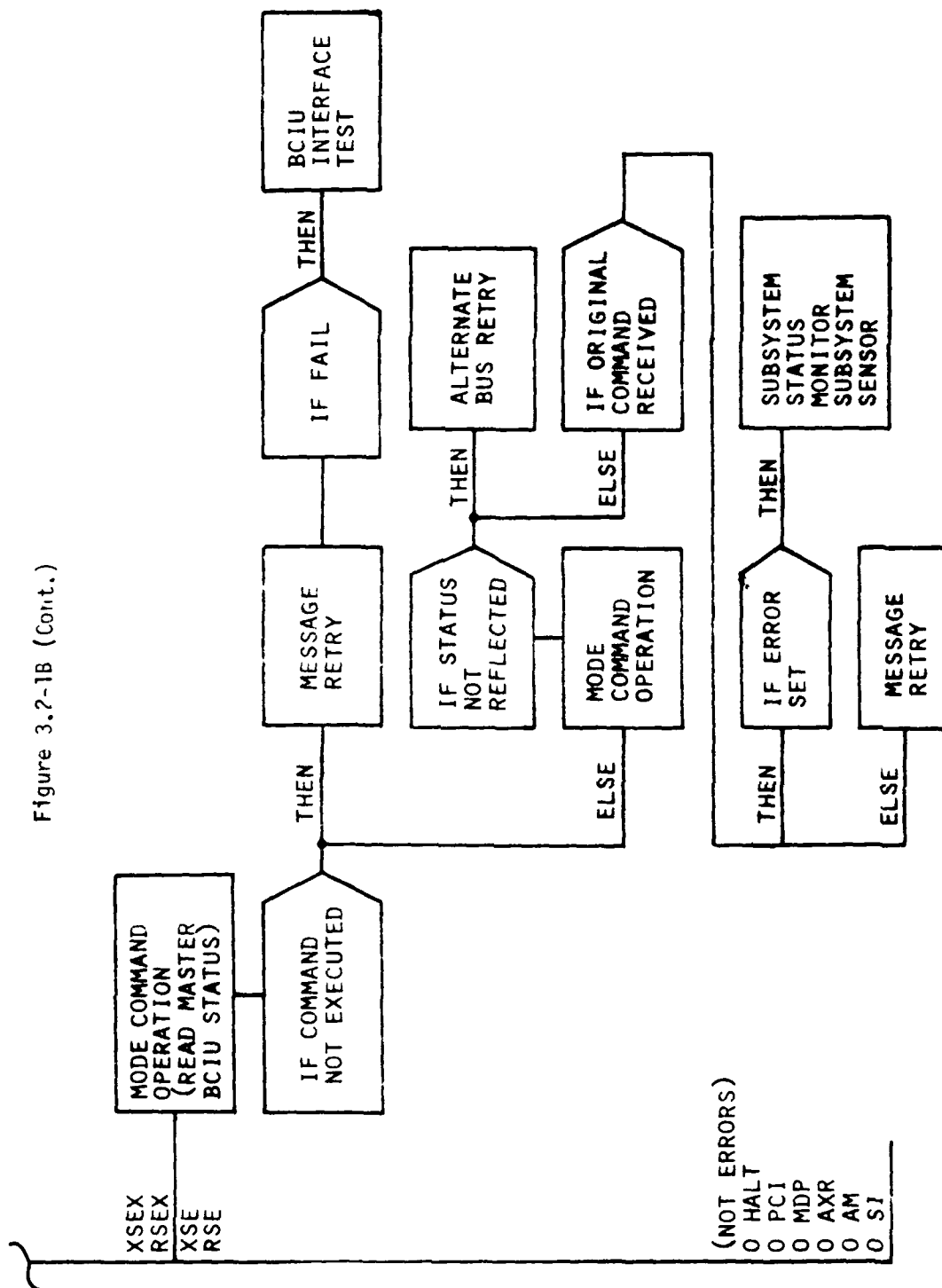


Figure 3.2-1B EHAR BUS COMMUNICATION ERRORS (MASTER EXECUTIVE)

Figure 3.2-1B (Cont.)



3.2.2.1 Inputs

BCIU Internal status, bus message status word, master BCIU processor command words (2 words from the BCIU command register), minor cycle number, BCIU processor interrupt identity are accessed by EHAR from the OFP Master Executive BCIU controller. Format and timing are specified in IDAMST Specification SS-3230, Prime Item Development Specification for IDAMST Bus Control Interface Unit (BCIU) which is incorporated by reference as a portion of this specification. In addition EHAR may request the status of any piece of equipment in the system from the Subsystem Status Monitor. EHAR receives an equipment identity number and equipment status code from the Subsystem Status Monitor. Equipment status code values are:

- . On line
- . Software out of service (software busy for maintenance)
- . Out of service (failed)
- . Idle

In addition EHAR accesses the Device Table of the Master Executive BCIU command list handler, via Local Executive service compool read. EHAR uses Device Table data to translate from BCIU command number to device identity.

3.2.2.2 Processing, Remote and Monitor Processors

EHAR receives control from the OFP Local Executive Interrupt Handler after occurrence of a BCIU generated processor interrupt. In event that the interrupt cause is one of the interrupt functions reserved for Master BCIU operation only, EHARS idles the processor/BCIU in the same manner as in the case of a mission processor internal error. Refer to Paragraph 3.2.1.2. The Processor/BCIU is idled to prevent propagation of errant data as a master mode interrupt from a remote mode BCIU is presumed to be caused by faultily operation of the BCIU. The BCIU is idled until diagnostic routines can verify correct operation. Diagnostics are not run until after reconfiguration has occurred. In event that the interrupt signal indicates a BCIU failure, the BCIU is idled as above. Refer to Paragraph 3.2.1.2. In event that the interrupt signal indicates a minor cycle message has been received from the Master BCIU, EHAR determines if the new minor cycle number is a valid number (new major cycle or one greater than previous minor cycle number). If minor cycle number is valid, EHAR relinquishes control to the OFP Local Executive minor cycle setup routine. If minor cycle is not valid, EHAR will format a mission recorder error entry. On second occurrence of invalid cycle number within a major cycle EHAR relinquishes control to reconfiguration. If monitor processor, reconfiguration takes control of the bus.

3.2.2.2.1 Processing, Master Processor

EHAR receives control from the OFP Master Executive BCIU controller on occurrence of BCIU processor interrupt signals. Certain interrupt signals are used by the Master Executive BCIU controller for system synchronization and control. Only those interrupt signals indicating errors or ambiguous situations are directed to EHARS. The BCIU internal status register bits that require action by EHAR when the BCIU generates a processor interrupt are:

- | | |
|-------|--------|
| . OMA | . RSE |
| . IVD | . XSK |
| . DPE | . RSEX |
| . ICC | . XSEX |
| . VDR | . IVI |

In addition the BCIU Fail and ROM parity error processor interrupts are handled by EHAR. In event of BCIU Fail, ROM parity interrupt or the instruction invalid or master function bits being set in the BCIU internal status register, (ISR), EHAR idles the BCIU by resetting PCR GO and executes a nominal BCIU/processor interface wrap-around or echo test. This test eliminates the possibility of intermittent error. If the error condition persists through the interface test, the BCIU/processor pair are idled as in the case of a processor internal error. Refer to Paragraph 3.2.1.2.

In event of message data error detection by the BCIU, (ISR bits NDR, ICP, DPE or IVD set), EHAR checks the BCIU command to see if BCIU automatic retries were specified for the message. If BCIU automatic retry has been attempted (as indicated by the interrupt signal) the error persists. EHAR modes the BCIU to the alternate bus and retries the message via the alternate bus. If alternate bus is not available because the subsystem status monitor tables indicate a previous failure or the alternate bus retry is unsuccessful, EHAR relinquishes control to reconfiguration (via the Subsystem Status Monitor) to reconfigure the subsystems accessed by the bus message subaddress out of the system. If alternate bus retry is successful, an RT and subsystem error record are logged in the subsystem status monitor tables. In addition, the original multiplex bus is marked as unavailable for future use. The unavailable multiplex bus can be restored to service by periodic or crew requested testing.

In event of message status word error detection by the BCIU, (ISR bits XSEX, RSEX, XSE, RSE) EHAR determines if the bus message command word was received by the remote device. EHAR uses mode command operation to determine if the bus command word was correctly received by the remote device. If the command word was correctly received and no message data errors were detected by the master BCIU or remote device, an error record is logged in the subsystem status monitor. If the command word was not received or is received in error, a message retry and alternate bus retry are attempted. If the error persists, the master BCIU is idled and the monitor BCIU assumes control. If the remote device also indicates a message error or device failure status, a message retry is attempted. If the error persists, the subsystems associated with the bus command subaddress are reconfigured out of service.

3.2.2.3 Outputs, Remote and Monitor Processor

To idle the processor/BCIU, EHAR directly controls the GO and Busy bits in the BCIU processor control register and sets the terminal failed code into the BCIU internal status register via the processor/BCIU programmed input/output interface. EHAR returns minor cycle number to the OFP Local Executive Minor Cycle Setup after verifying a valid minor cycle number. If minor cycle number is invalid, EHAR passes the received minor cycle number, the local (expected) minor cycle number to the subsystem status monitor for logging the error on the mission recorder.

Entry to reconfiguration will cause the monitor to assume control automatically and the remote to begin time out of the monitor to take over control, no parameters are passed.

3.2.2.3.1 Outputs, Master Processor

To idle the processor/BCIU, EHAR directly controls the GO and Busy bits in the BCIU processor control register and sets the terminal failed code into the BCIU internal status register via the programmed input/output interface.

To execute the BCIU/processor interface wrap-around or echo test, EHAR provides dummy BCIU interrupt vector locations to field interrupts generated by the BCIU in the course of testing. To use these dummy interrupt locations EHARS provides six sixteen-bit addresses to the OFP local Executive Interrupt Handler. EHAR directly loads and reads registers in the BCIU via the programmed input/output interface to execute the echo test. Refer to Prime Item Development Specification, IDAMST Bus Control Interface, Type B' (SS-3230), May 1976 for detailed description of BCIU internal registers. EHAR modes the BCIU to the alternate bus by placing asynchronous mode commands into the BCIU command list via the OFP Master Executive command list handler.

EHAR passes a 16-bit error flag, minor cycle number, message number, retry count, retry on alternate bus flag (if attempted) and up to four 16-bit bus message status words and four 16-bit BCIU internal status words to the Subsystem Status Monitor in the event that a subsystem sensor fails error recovery procedures. The status words and cycle/message identity are recorded on the mission recorder for post-flight analysis. The message identity is used to access the device table to identify the faulty piece of equipment. If alternate bus retry is successful, EHAR passes a unique 16-bit error code to the Subsystem Status Monitor along with message identity and bus message and BCIU status words.

3.2.3 Subsystem Sensors, Controls and Displays Errors EHAR Detailed Requirements

3.2.3.0.1 Input and Outputs, General

IDAMST subsystem sensors and controls and displays EHAR software is embedded code within the IDAMST OFP Applications. As an integral part of the OFP Applications, EHARS code interfaces to avionics equipment via IDAMST OFP Local Executive service requests (e.g., READ compool, WRITE compool) and data contained in data compools. Data compools are used for exchange of avionics subsystem control parameters and status data with the applications and error handling programs. Compoools in the IDAMST are transmitted on a real-time synchronized basis.

Input and Output specifications shall be as specified in Computer Program Development Specification, IDAMST OFP Applications (SB-4043), July 1976, Paragraph 3.2 and subparagraphs. Where additional requirements are imposed by EHARS requirements, the additions are noted in the following subparagraphs.

Table 3.1-2 lists the IDAMST OFP applications tasks that interface with the IDAMST subsystem sensors/actuators and controls and displays. EHARS code incorporated in each of these tasks is responsible for data life and reasonableness test, comparison with similar/like sensors where appropriate and with exercising BIT and interrogating BIT results and is responsible for determining that subsystems are in correct operating modes/states. Failure of any tests or checks performed by the EHARS code results in an error report being formatted and error notification being made to the subsystem status monitor of the OFP applications program. A unique 16-bit error code is assigned to each subsystem within the IDAMST system.

Table 3.2-1 lists those IDAMST subsystems (excluding multiplex and processors) having self-contained BIT capability. EHARS code within the OFP applications tasks that interface to these subsystems, is responsible for exercising BIT and interrogating BIT results.

EHARS provides the capability to exercise BIT on notification from other applications tasks. EHARS within the applications tasks however runs on a periodic basis and communicates with subsystems via synchronous, periodic messages.

- o HF/SSB
- o IFF
- o OMEGA (RECEIVER)
- o TACAN
- o IMS
- o WEATHER RADAR
- o RADAR ALTIMETER
- o STATION KEEPING EQUIPMENT
- o FLIGHT CONTROLS SYSTEM

Table 3.2-1 IDAMST Subsystems Containing BIT

3.2.3.0.2 Processing, General

IDAMST subsystem sensors and controls and displays EHAR code within the OFP Applications software is responsible for detection of errors in addition to recovery procedures. Techniques used in the EHAR code for detection of errors are:

- o Redundant data comparison
- o Similar function processed data comparison.
- o Life test.
- o Limit or reasonableness tests.

These tests are selectively applied to subsystems, dependent upon the nature of subsystem data. Where redundant subsystems are identical in function, EHAR directly compares data. Tolerance for slight discrepancies is incorporated. Exact values of tolerance is TBD by operational/mission requirements and by hardware performance specifications. Tolerance values are mechanized as system parameters (constants) that are updated by modification to the OFP load tape.

Where subsystems are similar in function, processed data derived from subsystem raw data is compared. Greater tolerances must be allowed for discrepancies than with redundant subsystem data because of discrepancies in external signals and because of differing accuracies of similar function subsystems.

Life tests for continuing operation of subsystems are applied where a minimum rate of data change is expected in normal operation of the subsystem. Where minimum rates are not ensured, life tests are augmented by hardware BIT incorporated by subsystems.

Limit or reasonableness tests of subsystem data are checks for data within fixed, reasonable limits or more typically checks for data rate changes within fixed, reasonable limits.

Recovery from errors requires isolation of failed subsystems to prevent propagation of errant data and degradation of system performance. EHAR reports errors to the Subsystem Status Monitor.

3.2.3.1 Display Processes

Display processes include the following OFP Applications tasks:

- HUD (R)
- HUD (R)
- INSTRUMENTS (R)
- LIGHTS
- MPD CHECKOUT
- MPD PARAMETERS/STATUS
- MPD ERROR/WARNING
- IMV FIXED TEXT
- IMV STATUS
- DEK MARK

Those tasks that are implemented redundantly in two mission processors are marked (R) above. Redundant display processes receive independent inputs

from separate applications tasks and output display data to the pilot and copilot, separate display devices.

3.2.3.1.1 Inputs

Refer to Computer Program Development Specification, IDAMST OFP Applications, SB-4043, July 1976. In addition to normal data inputs EHARS requires redundant copies of output data from the redundant display process (DISP). EHAR reads subsystem status from the OFP Applications Subsystem Status Monitor subsystem status tables.

3.2.3.1.2 Processing

Redundant display processes, DISP's incorporate direct comparison of output data and reasonableness tests. Reasonableness tests shall include maximum rate (derived from aircraft maximum rates) tests and absolute maximum (out of view) value tests.

Errors detected shall result in unique 16-bit error codes being passed to the subsystem status monitor. A unique error code is required for each display parameter as parameters are derived from different sensor data and are generated in different specialists applications tasks.

Non-redundant display processes incorporate life and reasonableness tests. All non-redundant tasks also display non-graphics, textual or discrete information. Life and reasonableness tests are limited to logical sequences/combinations of data.

3.2.3.1.3 Outputs

Refer to OFP Applications Specifications, SB-4043, July 1976. In addition, EHARS outputs unique 16-bit error codes to the OFP Applications Subsystems Status Monitor. Error codes are unique to the display page (IMK, MPD) or to the display device (LIGHTS, DEK MARK) were applicable.

3.2.3.2 Handlers

Two handlers are included in the IDAMST system to provide interactive software interface to the crew input/output devices:

IMK HANDLER
MPD HANDLER

3.2.3.2.1 Inputs

Refer to OFP Applications Specification, SB-4043, July 1976.

3.2.3.2.2. Processing

The display functions of HANDLERS incorporates the same logical reasonableness test of textual data that are used in display processes (Paragraph 3.2.3.1.2). The keyboard inputs functions of HANDLERS, incorporate logical reasonableness tests of sequences, timing and amount of input data.

Unique error codes for each page of output data and each function of keyboard input are passed to the OFP Applications Subsystem Status Monitor on detection of errors.

3.2.3.2.3 Outputs

Refer to OFP Applications Specification, SB-4043, July 1976.

3.2.3.3

EQUIPS

EQUIPS provide the IDAMST subsystem to software interfaces. EQUIPS provide subsystem equipment design dependent features, mode control of subsystems, and translation and normalization/scaling of subsystem data where required. EHARS is incorporated into the EQUIPS to control/retrieve subsystem BIT capability and to detect errors in data generated by subsystems. Subsystem data errors are herein considered separately from communications errors occurring in the communication channel between subsystem and mission processors. Communications errors are typified by single bit errors arising from transmission noise on the multiplex bus. Subsystem data errors are typified by a data parameter of all zeros or all ones arising from failure of a subsystem LRU or separation of a subsystem internal connector. Subsystem data errors are not detected by the multiplex bus BIT and where these errors are not detected by subsystem internal BIT, EHARS software in the mission processors is responsible for protection of the system from such errors.

3.2.3.3.1

General, Inputs and Outputs

For subsequent subparagraphs, inputs and outputs are as specified in Computer Program Development Specification, IDAMST OFP Applications, SB-4043, July 1976, Paragraph 3.2.5 and Subparagraphs which are incorporated herein by reference. Inputs and outputs specifications are not duplicated herein except where EHAR functions impose special requirements. In general and applicable to all EQUIPS, the EHAR function will output to the Subsystem Status Monitor a one word error code on detection of errors. The error code shall be a one word unique code specifying the subsystem identity and if input or output data was found to be in error.

3.2.3.3.2 UHF-AM Output Processing

EHAR tests that control commands are one of the set of commands valid for the radio and that numeric values are valid within the operational capability of the radio. EHAR checks for valid radio identity in the outgoing message compool.

3.2.3.3.3 VHF-AM Output Processing

EHAR tests that control commands are one of the set of the four commands valid for the radio set.

3.2.3.3.4 VHF-FM Output Processing

EHAR tests that control commands are one of the set of the nine commands valid for the radio set.

3.2.3.3.5 HF/SSB Output Processing

EHAR tests that control commands are one of the set of the ten commands valid for the radio set. EHAR exercises and monitors HF/SSB BIT.

3.2.3.3.6 ICS Input Processing

EHAR checks that control panel inputs are within valid ranges of input control variables.

3.2.3.3.7 ICS Output Processing

EHAR checks that control commands are one of the six valid commands for the intercom set. EHAR checks that station selection values are within the range of stations equipped in the aircraft.

3.2.3.3.8 P.A. Input Processing

EHAR checks that P.A. control panel inputs are within valid ranges of the input control variables.

3.2.3.3.9 P.A. Output Processing

EHAR checks that control commands are one of the four commands valid for the P.A. system. EHAR checks that speaker selection values are within the range of speakers equipped in the aircraft.

3.2.3.3.10 DEK Input Processing

EHAR shall only determine if one of the sixteen valid DEK input characters was input.

3.2.3.3.11 DSMU Output Processing

EHAR checks that DSMU switching commands are within the valid set of video source/memory location/display device combinations valid for the display system design.

EHAR provides capability to perform diagnostic test routines of the DSMU on demand from the flight crew. The nature of these tests are TBD based on the detailed design of the DSMU.

3.2.3.3.12 TACAN Input Processing

EHAR performs data life and reasonableness tests on TACAN input data. Due to system external considerations (distance, weather conditions, interference) EHAR tests may be disabled to allow degraded or intermittent TACAN operation. Reasonableness tests include range and bearing rate within acceptable limits, course deviation rate within acceptable limits and rate change of to-from indicator. EHAR shall periodically or on demand exercise the TACAN BIT capability and shall monitor TACAN status for abnormal conditions.

3.2.3.3.13 TACAN Output Processing

EHAR checks only that a TACAN control output is one of the valid types allowed and that channel or course select values are within acceptable limits.

3.2.3.3.14 HCU Input Processing

EHAR checks that HCU position variables X, Y and position rates $\Delta X, \Delta Y$ are within maximum allowable values.

3.2.3.3.15 OMEGA Input Processing

EHAR performs life and reasonableness tests on OMEGA input data. Due to system external considerations (distance to station, weather, interference) EHAR test may be disabled to allow degraded or intermittent OMEGA operation. Life and reasonableness tests check that pulse timing rates are within maximum acceptable limits. Station drop-out for excessive lengths of time are tested. Drop-out can result in channel re-selection or degraded operation. EHAR monitors OMEGA BIT data.

3.2.3.3.16 OMEGA Output Processing

EHAR tests that control commands are one of the three valid commands for the OMEGA set. EHAR tests that control values are within reasonable maximum values. EHAR exercises and monitors OMEGA BIT data.

3.2.3.3.17 FCS Input Processing

Air data andpositional/attitude data is received into the IDAMST system from the flight controls system (FCS) over three independent paths (three remote terminals). This data is independently processed in two mission processors and resulting position and attitude displays are independently presented to the pilot and co-pilot. The Flight Controls EQUIP accepts input data from the FCS interface triple channels. Flight Controls EQUIP EHARS in each of two computers independently compares the input data. Discrepancies exceeding a predetermined threshold result in an error indication being generated by the Flight Controls EQUIP. EHARS also checks that rates of attitude and air data are within maximum allowable values (determined from air frame maximum rates).

3.2.3.3.18 FCS Output Processing

Outputs from the IDAMST system to the FCS (e.g. autopilot steering commands, flight crew inputs) are generated in two IDAMST mission processors. A third channel of flight control output data is copied from the flight control data generated in the Master IDAMST mission processor. Flight Control EQUIP, EHARS code compares the two flight controls data. Discrepancies exceeding a predetermined threshold value results in an error indication and no new data being

sent. In case of discrepancies, the Flight Controls EQUIP will substitute previous values. Because two channels of flight control data are generated from one IDAMST source and the third channel of flight controls data is generated from a second IDAMST source, one channel of the flight controls system could be declared failed by voters in the FCS if differing data were sent to the FCS.

In addition EHAR tests that steering signals and signal rates are within acceptable maximum values. Errors in signals or signal rates results in previous values being substituted for errant data. EHAR exercises and monitors FCS BIT.

3.2.3.3.19 INS Input Processing

EHAR checks that altitude, attitude and position and velocity rates are within reasonable limit values. EHAR compares air data derived velocity (generated by the Instrument Display Process) and FCS attitude data compare within acceptable tolerance. EHAR monitors INS BIT data.

3.2.3.3.20 INS Output Processing

EHAR checks that output commands are one of the four valid commands for the INS system. EHARS checks format of waypoint data entered into the INS system. EHAR exercises and monitors INS BIT data.

3.2.3.3.21 SKE/ZM Input Processing

EHAR checks that range and bearing rates are within acceptable limits. EHAR exercises and monitors SKE BIT.

3.2.3.3.22 SKE/ZM Output Processing

EHARS tests that the output data type is correctly identified as one of the acceptable types and that values are within acceptable limits.

3.2.3.3.23 LF/ADF Inputs Processing

EHARS tests that bearing rate is within acceptable limits.

3.2.3.3.24 LF/ADF Output Processing

EHARS tests that control command outputs are identified by one of the valid five types and that frequency, test and volume data values are within limits valid for the LF/ADF radio set.

3.2.3.3.25 UHF/ADF Inputs Processing

EHARS tests that bearing rate is within acceptable limits.

3.2.3.3.26 UHF/ADF Outputs Processing

EHARS tests that control command outputs are identified by one of the valid five types and that frequency, test and volume data values are within limits valid for the UHF/ADF radio set.

3.2.3.3.27 Radar Altimeter Input Processing

3.2.4 Special Requirements

This section contains special requirements imposed on EHAR Software development.

3.2.4.1 Structured Programming

Top-down, structured programming concepts will be used throughout EHAR Software development. Software elements will be established which correspond to functions defined in this document.

3.2.4.2 JOVIAL J73

All EHAR Software will be coded in the JOVIAL J73 higher order language.

PRECEDING PAGE BLANK-NOT FILLED

3.3 Adaptation

This section summarizes the EHAR Software requirements with respect to the operating facility, system parameters, and system capacities.

3.3.1 General Environment

Further definition of the IDAMST system design is required prior to completing this portion of the specification. Pending definition the following assumptions are made.

3.3.1.1 IDAMST Core Elements

IDAMST core element hardware including the core element control/displays are assumed to be identical in all AMST aircraft and require no modification or variations in software to adapt the IDAMST OFP and OTP software.

3.3.1.2 Other IDAMST Integrated Hardware

Variations in AMST equipment complement associated with the IDAMST system is expected. It is assumed that the IDAMST OFP and OTP software will be automatically adaptable to hardware variations in the AMST. This will be accomplished through the use of an equipment status word from the IDAMST avionic hardware which identifies the existing hardware configuration. The OFP and OTP software will subsequently adapt to the actual configuration by omitting software functions associated with non-existent avionics hardware elements. The OFP and OTP software will compile a list of active and installed avionic equipment hardware and display list upon command and also write list on DITS recorder for a maintenance record.

3.3.2 System Parameters

Constants and other data pertaining to the particular mission must be available at load time for the EHAR Software to function at full capability.

3.3.3 System Capacities

Estimated capacity requirements of the Mission OFP Software is summarized in Table 3.3-1. These estimates are related to an IDAMST processor like that described in applicable document 2.1.4.b, "Prime Item Product Fabrication Specification for DAIS Processor", and allow a 25% growth margin.

Table 3.3-1 IDAMST MISSION PROCESSORS CAPACITY

	<u>INST</u>	<u>DATA</u>	<u>NOM EXEC.</u>
<u>IDAMST MASTER PROCESSOR</u>			
Top Level Control	7800	5854	304.4
Flight & Propulsion	1900	495	16.0
Communication	1515	339	1.6
Nav. & Guidance	4511	835	6.8
Aircraft System	3580	2200	52.9
Defense	230	29	
Misc.	1500	300	
	<u>21136</u>	<u>10053</u>	<u>281.7</u>
<u>IDAMST MONITOR PROCESSOR</u>			
Top Level Control	7800	5470	210.0
Flight & Propulsion	1880	492	16.0
Communication	1515	339	1.6
Nav. & Guidance	4511	835	6.8
Aircraft System	3580	2200	52.9
Defense	230	29	
Misc.	1500	300	
	<u>21016</u>	<u>9665</u>	<u>277.3</u>
<u>IDAMST REMOTE PROCESSOR</u>			
Top Level Control	2500	2419	108.0
Flight & Propulsion	75	16	3.2
Nav & Guidance	5210	431	14.0
Aircraft Systems	2700	326	7.1
Defense	25	6	
Misc.	400	100	
	<u>10910</u>	<u>3298</u>	<u>132.3</u>

4.0 QUALITY ASSURANCE PROVISIONS

This section identifies the basic method for accomplishing software verification.

4.1 Introduction

IDAMST CPCIs will incorporate top-down, structured concepts, described briefly below:

Structured Program

A structured program is a computer program constructed of a basic set of control logic figures which provide at least the following: Sequence of two or more operations, conditional branch to one of two operations and return repetition of an operation. A structured program has only one entry and one exit point. A path will exist from the entry to each node and from each node to the exit. In addition, certain practices are associated, such as indentation of source code to represent logic levels, use of intelligent data names and descriptive commentary.

Top-Down Programming

Top-down programming is the concept of performing in hierarchical sequence a detailed design, code, integration and test as concurrent operations.

Top-Down Structured Programs

A top-down structured program is a structured program with the additional characteristics of the source code being logically but not physically segmented in a hierarchical manner and only dependent on code already written. Control of execution between segments is restricted to transfers between vertically adjacent hierarchical segments.

Top-down coding and verification is an ordering of system development which allows for continual integration of the system parts as they are developed and provides for interfaces prior to the parts being developed. At each stage, the code already tested drives the new code, and only external data is required.

In top-down programming, the system is organized into a tree structure of segments. The top segments contain the highest level of control logic and decisions within the program, and either passes control to the next level segments or identifies the next level segments for in-line inclusions. The next level may include stubs. Stubs which are to be replaced eventually with running code may contain a "no operation" instruction or possibly a display statement to the effect that control has been received. The process at replacement of successively lower level stubs with operational code continues until all functions within a system are coded and verified.

In top-down coding and verification, the highest level element is coded first. Coding, checkout, and integration proceed down the hierarchy until the lowest levels have been integrated. This does not imply that all elements at a given level are developed in parallel. Some branches will intentionally be

developed early, e.g., to permit early training and early development of critical functions or hardware/software integration.

Many systems interfaces occur through the data base definition in addition to calling sequence parameters. Top-down programming requires that sufficient data definition statements be coded and that data records be generated before exercising any segment which references them. Ideally, this leads to a single set of definitions serving all the programs in a given application.

This approach provides the ability to evolve the product in a manner that maintains the characteristic of always being operable, extremely modular and always available for successive levels of testing that accompany the corresponding levels of implementation. Exception to the top-down coding and integration approach will be considered on a case-by-case basis.

Each computer program will be coded in a higher order language. Use of assembly or machine language will be restricted to coding of certain executive functions where the higher order language cannot be used.

Real Time Structured Programs

An additional complexity in the IDAMST system is the Real Time, asynchronous communication of structured programs as tasks. Tasks are also organized as a hierarchy. Each task has a Controller Task which is the only task permitted to schedule or cancel the lower level task. However, any task is permitted to activate any other task in IDAMST.

4.2 Computer Program Verification

Computer program verification is the process of determining whether the results of executing a computer program in a test environment agree with the specification requirements. Verification is usually only concerned with the logical correctness of the computer program (i.e., satisfying the functional/performance requirements) and may be a manual or a computer-based process (i.e., testing software by executing it on a computer).

The use of top-down structured programming techniques provide certain program characteristics that may lead to a simplification of the computer program verification process. Top-down integration of the program elements in a CPCI minimizes the use of complex driver routines and replaces them with actual program elements and simple program stubs. It also provides a system in which the computer program is continually being tested as successively lower levels of program elements are integrated and the interfaces between program elements are verified prior to the integration of the next lower level.

4.2.1 Program Element Tests

Program elements are coded in the sequence required for top-down integration. When coding and code review are completed, each program element shall be functionally tested in a stand-alone configuration by the programmer to assure that the element can be executed and that the specified functions are performed. Since program elements are small and are restricted to one entry point and one exit point, the test environment is relatively simple.

4.2.2 CPCI Integration Tests

Following successful completion of the Program Element Tests, the program elements are entered into the Computer Program Library where they are subjected to configuration control procedures. Controlled program elements are compiled/assembled, link-edited and the current CPCI version is made available for integration testing. Integration tests are dynamic tests designed to verify program functions and interfaces between program elements and with the data base. The result is a complete CPCI for which all design features have been verified.

The integration of program elements or tasks into the complete computer program shall be accomplished in a top-down sequence. The highest level elements which contain the highest level controller tasks shall be tested and integrated first. These tasks are the Master Sequencer, Configurator, Request Processor, and Subsystem Status Monitor. Testing and integration shall proceed down the hierarchy until all program elements (e.g., equipment interface functions), have been integrated and the design completely verified.

An important aspect of integration testing of IDAMST will be the invocation and synchronization of the tasks, since these functions do not fall under the structured programming rules.

4.2.3 Formal Software Testing

The purpose of formal testing is to confirm that the computer program performs the functions and satisfies the performance requirement contained in the software requirements specification. Formal testing consists of Preliminary Qualification Tests (PQT) and Formal Qualification Tests (FQT), and are conducted in accordance with Air Force approved test plans.

Pre-Qualification Testing (PQT)

PQT is an incremental process which provides visibility and control of the CPC2 development during the time period between the Critical Design Review and Formal Qualification Testing.

PQT consists of functional level tests, conducted at the development facility, and using Air Force approved test plans. These tests will use documented procedures, completed by the contractor, and submitted to the Air Force Sufficiently in advance of the scheduled test session to permit review and analysis. They will typically use controlled inputs specifically prepared for the test purpose.

A Pre-Qualification test will generally be conducted for each CPCI function. If a test's cost or time consumption estimates are significantly high, the test will be deferred to FQT unless it is time-critical or performance-critical to the development of the CPCI.