CR 151361

BACKUP FLIGHT CONTROL SYSTEM FUNCTIONAL EVALUATOR SOFTWARE MANUAL

Job Order 34-249

Action Document 249-16

(NASA-CR-151361) BACKUP FLIGHT CONTROL SYSTEM FUNCTIONAL EVALUATOR SOFTWARE MANUAL (Lockheed Electronics Co.) 144 p HC A07/MF A01 CSCL 01C
N77-23128
Unclas
G3/08 26152

Prepared By

Lockheed Electronics Company, Inc.
Systems and Services Division
Houston, Texas

Contract NAS 9-15200

For

DATA SYSTEMS BRANCH

CONTROL SYSTEMS DEVELOPMENT DIVISION

National Aeronautics and Space Administration

LYNDON B. JOHNSON SPACE CENTER

Houston, Texas

April 1977

LEC-10119

BACKUP FLIGHT CONTROL SYSTEM FUNCTIONAL
EVALUATOR SOFTWARE MANUAL

Job Order 34-249

PREPARED BY

Charles A. Helmke

Steve H. Hasara

Frances E. Mount

APPROVED BY

S. D. Person, Job Order Manager
Data Systems Section

W. R. Labby, Operations Manager
Control Systems Development Department

Prepared By

Lockheed Electronics Company, Inc.

For

Data Systems Branch

Control Systems Development Division

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

April 1977

LEC-10119

1. Report No. JSC-12559	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Backup Flight Control System Functional Evaluator Software Manual		5. Report Date April 1977
		6. Performing Organization Code
7. Author(s) Charles A. Helmke, Steve H. Hasara, Frances E. Mount Lockheed Electronics Company, Inc.		8. Performing Organization Report No. LEC-10119
9. Performing Organization Name and Address Lockheed Electronics Company, Inc. 16811 El Camino Real Houston, Texas 77058		10. Work Unit No.
		11. Contract or Grant No. NAS 9-15200
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Lyndon B. Johnson Space Center Houston, Texas 77058 JSC Technical Monitor: Gene T. Rice		13. Type of Report and Period Covered Software Manual
		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

The software for the Backup Flight Control System Functional Evaluator (BFCSFE) on a Data General Corporation Nova 1200 computer consists of three programs: the Ground Support Program, the Operational Flight Program (OFP), and the Ground Pulse Code Modulation (PCM) Program. The Ground PCM Program was written by NASA personnel and is not described in this document. The Nova OFP software is structurally as close as possible to the AP101 code; therefore, this document highlights and describes only those areas of the Nova OFP that are significantly different from the AP101. Since the Ground Support Program was developed to meet BFCSFE requirements and differs considerably from the AP101 code, it is described in detail.

17. Key Words (Suggested by Author(s)) Software, Functional Evaluator, Flight Control, Backup		18. Distribution Statement Unclassified – unlimited	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 144	22. Price*

*For sale by the National Technical Information Service, Springfield, Virginia 22161

JSC Form 1424 (Rev Nov 75)

NASA — JSC

PRECEDING PAGE BLANK NOT FILMED.

CONTENTS

Section Page

PRECEDING PAGE BLANK NOT FILMED.

ACRONYMS

BCE	Bus control element
BFCS	Backup Flight Control System
BFCSFE	Backup Flight Control System Functional Evaluator
BITE	Built-in test equipment
CDU	Command decoder unit
CPU	Central processing unit
CU	Control unit
DGC	Data General Corporation
FC	Flight control
GPC	General purpose computer
I/O	Input/output
IOP	Input/Output Processor
IOPS	Input/Output Processor Simulator
JSC	Lyndon B. Johnson Space Center
JSR	Jump to subroutine
LEC	Lockheed Electronics Company, Inc.
MDM	Multiplexer/demultiplexer
MIA	Multiple interface adapter
MIT	Massachusetts Institute of Technology
MSC	Master sequence controller
NASA	National Aeronautics and Space Administration
OFP	Operational Flight Program
PCI	Program-controlled input
PCM	Pulse code modulation
PCO	Program-controlled output
RDOS	Real Time Disk Operating System
SAIL	Shuttle Avionics Integration Laboratory
SATS	Shuttle Avionics Test System
TCP	Test Control Procedure
WOW	Weight on wheels

1. INTRODUCTION

This manual describes the software written for the Backup Flight Control System Functional Evaluator (BFCSFE) on a Data General Corporation Nova 1200 computer. It was developed for the Data Systems Branch of the National Aeronautics and Space Administration/Lyndon B. Johnson Space Center (NASA/JSC).

The software is referred to interchangeably throughout the document as either the BFCSFE or Nova software. It was written to the requirements specified in section 2.2 of the *Backup Flight Control System Functional Evaluator Implementation Plan* (LEC-9304, Lockheed Electronics Company, Inc., August 1976). It is assumed that the reader has read this document and is also familiar with two other documents: *Backup Flight Control System Flight Program, Program Requirements Document* (Volumes I and II, North American Rockwell, March 29, 1976) and *Backup Flight Control System Flight Program Description Document* (Volumes I and II, Charles Stark Draper Laboratory, MIT, July 15, 1976).

As stated in the implementation plan, the software is divided into three different areas: the Ground Support Program, the Operational Flight Program (OFP), and the Ground Pulse Code Modulation (PCM) Program. The Ground PCM Program was written by NASA personnel and is not described in this document. The Nova OFP software is structurally as close as possible to the AP101 code. Detailed functional descriptions for each OFP routine are not given in this report since this information is readily available in the Draper Laboratory's *Backup Flight Control System Description Document.* The OFP flight control modules, the fault detection routine, and the formatters are essentially a line-by-line conversion. This document highlights and describes only those areas of the Nova OFP that are significantly different from the AP101. Since the Ground Support

Program was developed to meet BFCSFE requirements and differs considerably from the AP101 code, it is described in detail.

The source programs are maintained on a Real Time Disk Operating System (RDOS) disk. LEC personnel will provide source listings and assembly listings on request.

2. SOFTWARE DEVELOPMENT TECHNIQUES

2.1 USE OF DETAILED FLOW CHARTS

The software for the BFCSFE was developed primarily by following the Draper Laboratory's existing detailed flow charts. These flow charts are at an ideal level for assembly language coding in that they are as detailed as possible while remaining machine independent. The flow charts indicate the operations to be performed on the variables without indicating specific machine instructions and registers. The AP101 program listings were used as a backup to the flow charts in developing the Nova code.

2.2 NOVA MACRO CAPABILITY

The converted programs were assembled using the Nova Macro Assembler in the RDOS for the Nova 1200 computer system. The Macro Assembler was beneficial in two major ways. First, it allowed the macros that were used extensively in the flight control modules and the fault detection routine of the AP101 OFP to be implemented in the same manner in the Nova software. Second, it allowed the use of macros to produce code for controlling the Input/Output Processor Simulator (IOPS).

2.3 VARIABLE NAME CONVERSION

The AP101 assembly language allows up to a maximum of eight characters for variable names and instruction labels, and this was often used by Draper Laboratory. The Nova assembly language utilizes only the first five letters of variable names, and thus the names must be unique in the first five characters. The AP101 data base variable names (900 plus) were shortened to five characters, in most cases by truncating to the first five characters. When this procedure yielded duplicate five-character names, other combinations of the original characters were used for the five-character name. A cross reference between the

BFCSFE data base variable name and its corresponding AP101 name is available in the program listing.

2.4 IOPS PROGRAMMING

The IOPS was designed by the Autonetics Division of Rockwell International. Their *SATS II (System 2) IOPS Programming and Users Manual* (October 16, 1974) was used as a reference for developing the Nova code necessary to control the IOPS. Macros were used to develop machine code for the IOPS master sequence controller (MSC) and bus control elements (BCE's).

2.5 RELOCATABLE LOADER

All the programs are relocatable with the exception of most of page zero. The Nova Relocatable Loader is used to link all the programs into a single core image module. With the programs relocatable, assembly updates can be made in each routine, and only a re-execution of the Relocatable Loader is necessary in order to obtain an updated total core image module.

3. SOFTWARE CHECKOUT

3.1 NOVA SYMBOLIC DEBUGGER

The Nova symbolic debugger (DEBUG III) in relocatable binary form is included with the BFCSFE programs in the core image module. DEBUG III was used almost continuously during online checkout of the BFCSFE software because it allows the user to set up to eight breakpoints for suspending program execution. By using DEBUG III, memory locations and system registers can be searched, examined, and modified.

3.2 FLIGHT CONTROL MODULES UNIT TEST

A program independent of BFCSFE software was written to execute all possible paths through the flight control modules and to check the accuracy of all computations. This program is called Unit Test and is an implementation of the *Backup Flight Control System Status Program Test Document* (Charles Stark Draper Laboratory, MIT, April 14, 1975). The results of Unit Test are given in appendix D.

3.3 TEST CONTROL PROCEDURE

In order to verify the overall system operation of BFCSFE, a Test Control Procedure (TCP) based on the Shuttle Avionics Integration Laboratory (SAIL) TCP was run.

4. DETAILED DESCRIPTION OF GROUND SUPPORT PROGRAM

The Ground Support Program serves as a diagnostic for the BFCSFE system. It resides in memory along with the OFP and is usually run prior to the OFP.

4.1 PURPOSE

The BFCSFE Ground Support Program is used to verify the operation of hardware and communication links unique to the BFCSFE. These include the general purpose computer (GPC); the multiplexer/demultiplexers (MDM's) MDMFF5, MDM1, and MDM2; related multiple interface adapters (MIA's); and the busses and wires connecting these elements.

4.2 INITIALIZATION AND MODE CONTROL

Primary mode control of the BFCSFE is accomplished with the GPC MODE switch (HALT, RUN, STANDBY) and the BFCS ENGAGE and RESET switches. From the viewpoint of the GPC hardware, the RUN and STANDBY states are equivalent. They can be differentiated only in software by reading the discrete input registers.

After executing some basic initialization, the ground program moding logic is established. This is essentially a loop that reads the RUN/STANDBY bits and exits to the OFP if the RUN bit is present. That is, the ground program executes only if the STANDBY bit is present. Note that once the OFP has been entered, the only way to return to the ground program is to cycle the MODE switch to the HALT state, then back to STANDBY. The OFP does not examine the RUN/STANDBY condition and has no knowledge of the ground program.

Secondary mode control of the ground program is accomplished with the Backup Flight Control System (BFCS) ENGAGE and DISENGAGE (reset) buttons. In the engaged condition, MDM testing and sum check are performed. The not-engaged state cycles the GPC self-test, IOPS diagnostic, and MDMFF5 testing.

Both primary and secondary mode control is performed in the routine starting at location MODING1. The Input/Output Processor (IOP) discrete input register contains the RUN/STANDBY bits and the three BFCS software ENGAGE bits. When the RUN and STANDBY bits are both on, a hardware error is implied. The routine takes the built-in test equipment (BITE) error exit (MODBITE1), which lights the BFCS FAIL light and enters the WAIT state, waiting for the HALT bit to be present. If the RUN bit is on, the routine jumps to OFP. If the RUN bit is not on, the routine tests the ENGAGE state and calls the appropriate subprogram for BFCS engaged (BENG) or BFCS not engaged (BENGNOT). If the ENGAGE discretes disagree, they are read again after a delay to allow the discretes to settle. If they still disagree, the routine exits to MODBITE1. (If one ENGAGE bit is on, all three should be on. T' disagree means at least one is on, but not all three.)

With each pass, the MODING discretes are tested again and the appropriate action taken. The practical result of this is that either the BENG or BENGNOT loop is repeatedly executed until either the BFCS ENGAGE state changes or the GPC is put into the RUN mode.

4.3 BFCS ENGAGED SUBPROGRAM (BENG)

4.3.1 BENG MODE CONTROL

The BENG subprogram is called from and returns to the moding loop unless a BITE error occurs: a BITE status register (BSR) read not equal to 3400. In this case an exit to MODBITE1 is taken.

Upon entering BENG, the first pass flag (PASSONE) is tested. If the flag is on, then the routine EXECTEST is called to perform first pass initialization for BENG; to execute BITE tests for MDMFF5, MDM1, and MDM2; and to execute MEMCK (memory checksum). PASSONE is set by INIT and by the BENGNOT subprogram, and reset by EXECTEST. Therefore, BENG subprogram initialization, MDM tests, and MEMCK are performed once each time the BFCS is powered into or toggled into the ENGAGED state.

4.3.2 FIRST PASS INITIALIZATION (EXECTEST)

EXECTEST is initialized as follows:

1. The PASSONE flag is set to 1.
2. The BFCS ENGAGE light is turned on.
3. The STFAIL and STATFLAG flags are reset.
4. MDMFF5, MDM2, and MDM1 are tested.
5. MEMCK (memory checksum) is executed if all MDM's pass the BSR read.

If STATFLG is set, this indicates that the BSR read test failed, and the BFCS FAIL light is turned on.

EXECTEST returns to MODING1 unless a BSR error occurs, in which case it jumps to MODBITE1.

4.3.3 FAILURE REPORTING

Error	Flag	Bit number
BSR FAIL MDM5	MDM5FLAG	15
	STATFLAG	15
MEMCK ERROR	—	Light GPC-3 FAIL light

4.4 BFCS NOT ENGAGED SUBPROGRAM (BENGNOT)

The BENGNOT subprogram is called from and returns to the moding loop unless a BITE (BSR read) error occurs. In this case an exit to MODBITE1 is taken.

Upon entering BENGNOT, the BFCS ENGAGED light is turned off, and the PASSONE flag is reset to 0. Then the SELF-TEST FAIL flag (STFAIL) is tested. If the flag is set, then self-test has failed previously and no attempt is made to execute it again while in the BENGNOT mode. The STFAIL flag is reset in INIT and by BENG initialization (EXECTEST), so cycling power or cycling the ENGAGE or HALT state allows self-test to be re-executed.

If self-test is to be performed, execution begins at location GOSLFTST. Certain initialization is done in preparation for self-test execution; then the self-test program is executed. When self-test returns, the GPC has interrupts disabled for the remainder of the BENGNOT operation, and the memory protect limits are reset. GPC self-test is described in more detail in section 4.6.

The STATUS flag indicates the result of the GPC self-test. If there is a GPC self-test failure, the GPC-3 STATUS flag (STFAIL) is set, the GPC-3 FAIL light is set, and the program returns to the MODING1 loop. If there is not a GPC self-test failure, the STATUS flag is reset and MDMFF5 is tested.

The MDM FAIL flag (STATFLAG) is set if MDMFF5 fails and the BFCS FAIL light is set. The program then exits to MODBITE1. If MDMFF5 passes, the program returns to the MODING1 loop.

4.5 GPC SELF-TEST (GOSLFTST)

4.5.1 IMPLEMENTATION

This program consists entirely of the Nova and IOPS diagnostic tests. The tests consist of the following:

1. Reset Test (IOPS)
2. Load IOPS Program Memory Test (IOPS)
3. Load IOPS Output Data Ram Test (IOPS)
4. IOPS Interval Timer Test (IOPS)
5. NO-GO Timer Test (IOPS)
6. BCE Time Out Test (IOPS)
7. Multiply-Divide Test (Nova)
8. Floating Point Test (Nova)
9. Read/Write Memory Test, QENTRY (Nova)

On completion of each test, a FAIL flag (STFAIL) is set if the test has failed. On completion of GOSLFTST, the FAIL flag is checked and the GPC-3 FAIL light is turned on if any one of the tests has failed.

4.5.2 INITIALIZATION BEFORE CALL/AFTER RETURN

The GOSLFTST program is called in the BENGNOT program with a jump to subroutine (JSR).

On returning from self-test, the interrupts are disabled and the IOPS memory protect is reset.

4.5.3 FAILURE REPORTING

Error	Flag	Bit number
SELF-TEST FAIL	STFAIL	15

4.6 MDM TESTING

4.6.1 TESTING CONCEPT

All testing is accomplished using the BITE tests designed into the MDM's. Data values returned to the GPC by a BITE test are compared with expected values. If they agree, the test succeeded. If they do not agree, the test failed.

The MDM tests require that expected values be set up before commanding the BITE (BSR) test. Each test is composed of IOP instructions to initiate the test and execute the data transfers, and central processing unit (CPU) instructions to evaluate the resulting data.

4.6.2 TEST MECHANIZATION

During BENG operation, MDM1, MDM2 (DDU), and MDM5 (MDMFF5) are tested. During BENGNOT operation, only MDM5 is tested.

4.6.3 SUBROUTINES TO EXECUTE TESTS

The subroutines to execute tests on MDM1, MDM2, and MDM5 are called MDM1, MDM2, and MDMFF5. Each test has the same format:

BSR read
Delay
BSR read

If the second BSR read fails, the program exits to FAIL1, FAIL2, and FAIL5 for MDM1, MDM2, or MDM5, respectively. In the FAIL routine the BSR is read again for a maximum of five tries. If at the end of five tries the test still fails, the appropriate MDM FAIL flags are set.

4.6.4 FAILURE REPORTING

Error	Flag	Bit number
BSR FAIL MDM1	MDM1FLAG	15
	STATFLAG	15
BSR FAIL MDM2	MDM2FLAG	15
	STATFLAG	15
BSR FAIL MDM5	MDM5FLAG	15
	STATFLAG	15

4.7 PROGRAMMING CONVENTIONS

4.7.1 MACROS FOR IOPS INSTRUCTIONS

Macros are used to generate the Nova machine code for the IOPS instructions for program-controlled output (PCO), program-controlled input (PCI), MSC, and BCE, with the exception of the IOPS instructions in the self-test. The assembly language statements are assembled on the Nova using Data General Corporation's (DGC's) Macro Assembler along with the Macro Definitions File (FMIOPSAVE). This file was written by Autonetics in the macro language described in DGC Manual 98-81-02, and was updated in January 1976 to permit the assembly of correct instructions for the command decoder unit (CDU).

4.7.2 IOPS INSTRUCTIONS IN GOSLFTST

The PCO, PCI, MSC, and BCE instructions in the self-test (GOSLFTST) use the same names as the macros, but are actually subroutines. The subroutines are part of the SHIOROUTINE file used in the flight program. GOSLFTST has to be assembled separately from the rest of the Ground Support Program, without using the Macro Definitions File (FMIOPSAVE).

4.7.3 IOPS INSTRUCTION VERIFY

After each load of the IOPS Instruction Ram, the subroutine SUBVERIFY is used to verify a correct IOPS load. In the case of a verify error, the program lights the BFCS FAIL light and returns.

5. DETAILED DESCRIPTION OF OFP EXECUTIVE ROUTINE

5.1 POWER-UP INITIALIZATION

The BFCS OFP is self-initializing from its entry point HSPOWER. Upon entering HSPOWER, all Nova interrupts are disabled and all are masked off except those from the IOPS. Then the IOPS interrupts are all masked off. The IOPS interval timer is set high to remove it from consideration during initialization. The interrupt service address is set to HSPOWER, and the interrupts are unmasked. Any pending interrupts cause the entire sequence up to this point to be repeated until all interrupts have been cleared.

Next the data base is zeroed from the variable JXSAD to FNACC. All memory locations other than these are protected from being written by the IOPS. The IOPS is loaded with the predetermined MSC and BCE programs. The IOPS programs for selecting and resetting CDU's 1, 2, and 5 are executed. Then there is a software delay loop of 178 microseconds to allow the CDU's to settle.

Now the input/output (I/O) initialization subroutine (IOPSTART) is called. IOPSTART resets the IOPS, delays with a software loop for 28 microseconds, and then enables transmitters and receivers on BCE's 1, 2, and 5. Then an IOPS program which does a BITE reset for CDU 5 is executed.

The interrupt address is set to HSRUPT in case any interrupts are triggered by the IOPS initialization. The program delays for a short period to allow for any straggler interrupts. If there are none, it executes HSRUPT directly.

At HSRUPT, any interrupts are serviced with a Nova I/O reset. The interrupt service address is set to HSPOWER. Any IOPS interrupts are released, and the IOPS interrupt register is cleared. All interrupts are then enabled, and the FILTER INITIALIZATION flag and the POWER-UP INITIALIZATION flag are set. The S-OUT list and the CAUTION AND WARNING flag are initialized. Interrupts are then disabled, and the interrupt address is set to IEXRUPT. The IOPS interval timer is set for 30 milliseconds, the error interrupt counters are cleared, and interrupts are enabled. An idle loop is established to await the expiration of the interval timer.

5.2 EXECUTIVE (IEXRUPT, EXRUPT, AND EXLOOP)

After power-up initialization, all interrupts are serviced in IEXRUPT. IEXRUPT immediately disables the interrupts and saves the accumulators. It then makes a determination as to whether self-test was in progress at the time of the interrupt. If it was, the saved accumulators and the return address are saved in self-test accumulators. Then a test is made to determine whether the interrupt was caused by a power failure. If this is the case, the IOPS is reset, a HALT instruction is placed at location zero, and the program halts. If power is restored, the program goes immediately to location zero and halts. No recovery is programmed for power failures.

If the interrupt was not caused by a power failure, a test is made to determine whether the interrupt was produced by the IOPS. If the interrupt was produced by any device other than the IOPS, this represents an error. In this case, the interrupt code is set to 12, the subcondition code is set to zero, and a jump is made to the main restart routine EHMAIN.

If the interrupt was generated by the IOPS, then the IOPS interrupt register is read to determine whether the interrupt was caused by the interval timer expiration. If it was, the program goes to EXRUPT. If not, the program goes to the error handler EH5BITE.

EXRUPT resets the IOPS interval timer interrupt and enables interrupts. It then sets bit 1 of the IOPS output discrete to flag the start of OFP execution time. The job table pointer is initialized to the top of the job table. The minor cycle counter (EXMCY) is incremented. The FLIGHT CONTROL COMPLETE flag (QXBFC) and the PCM COMPLETION flag (QXBPCM) are zeroed. The restart phase counter (HTPHASE) is tested to see whether the last four minor cycles have been free of restarts. If so, the persistent restart condition is cleared by zeroing the RESTART FAIL flag (QXBRSTRT) and the temporary restart counter (QTBRSCTR).

The minor cycle initialization is now complete, and the program falls through to the job dispatch routine (EXLOOP). The dispatcher steps down the job table and examines each entry in order. If the conditions for job execution are met, then the dispatcher transfers to that job. If the conditions for job execution are not met, then the dispatcher skips that job and examines the next table entry. The last job is an idle loop (HSIDLE), which simply waits for the interval timer interrupt.

The job table entries are organized as follows:

1. The first word is the address for the job.
2. The second word is an integer which denotes the number of checks (if any) that must be passed before the job can be invoked.
3. For each check that must be passed, two words follow. Thus the job table entries vary in length but are a multiple of two words long. The first word of the pair is either the address of a parameter (a number which would be greater than 3) or a minor cycle counter mask (a number 0, 1, 2, or 3). If the first word is a parameter address, the parameter value is retrieved and compared to the second word of the pair. If they are equal, the check is considered to pass.

If the first word of a pair is a minor cycle counter mask, the mask is "anded" with the minor cycle counter. If the result is equal to the second word of the pair, the check is considered to pass.

5.3 RESTARTS

5.3.1 PHILOSOPHY OF RECOVERY

The only appropriate recovery from most detected hardware errors is to reinitialize the data base and the program state to make this information logically consistent. The amount of initialization necessary to produce the required information consistency after each type of error may vary. Instead of providing multiple recovery schemes to maximize recovery efficiency for different errors, a single reinitialization scheme is used for program simplicity.

The main purpose of the restart program is to recover from transient failures. If a transient failure is defined as a failure that is caused by a single-point-in-time logical discrepancy that does not leave a permanent scar on the BFCS, then reinitialization of the GPC hardware and OFP data base will be sufficient for recovery from all transients.

5.3.2 SPECIFIC OBJECTIVES OF THE RESTART PROGRAM

Certain operational objectives must be met by the restart program to minimize the impact of a failure on normal OFP operations and to maximize the probability of maintaining vehicle control.

1. The integrity of minor cycle timing must be maintained since the flight control algorithm is designed to operate at that rate. The restart program must make appropriate use of the minor cycle clock.

2. Filter initialization must be performed selectively as a function of the frequency of restarts. (For example, if a failure occurred repetitively after flight control was completed and the filters were initialized on every restart, this would result in a manual direct system with no flight control stabilization.) If restarts occur with sufficiently high frequency, only the first restart in a series should call for filter initialization.

3. The integrity of PCM transmissions should be maintained so that the ground can be kept abreast of BFCS operation.

5.3.3 RESTART IMPLEMENTATION

5.3.3.1 IOPS Error Handler (EH5BITE)

EH5BITE is entered as a result of an IOPS interrupt other than the expiration of the interval timer. The indicator EHRCODE is set to 6 to identify this interrupt. The IOPS interrupt register is examined in the following sequence:

1. If bit 1 is on, the interrupt was caused by an overflow in the Input Buffer Ram, and the interrupt subcondition code (EHSCODE) is set to 10.

2. If bit 3 is on, the interrupt was caused when the IOPS Buffer Box hung up, and EHSCODE is set to 13.

3. If bit 7 is a 1, there was an IOPS-to-computer (input) address error, and EHSCODE is set to 14.

4. If bits 1, 3, and 7 are all off, the interrupt is unidentified, and EHSCODE is set to zero.

The program then transfers to EHMAIN.

5.3.3.2 Main Restart Routine (EHMAIN)

The routine consists of four parts:

1. Initialization of the IOPS and OFP variables
2. Fault annunciation and bookkeeping
3. Telemetry maintenance
4. Minor cycle timing management

EHMAIN resets the IOPS and then enables transmitters for BCE's 1, 2, 5, and 8 and receivers for BCE 5. The S-OUT list is initialized. The SELF-TEST FAIL flag (QXBSTEST), the SELF-TEST DONE flag (FDSTDONE), and the weight-on-wheels (WOW) delay filter counter are zeroed.

For fault annunciation and bookkeeping, the RESTART OCCURRED flag (QTBRSTRT) is set. (It is always reset at the start of the next minor cycle.) The permanent restart counter (QXBRSCTR) is incremented by 1 and is just an historical record of restart occurrences. Each restart increments the temporary restart counter (QTBRSCTR). On the third restart, the BFCS FAIL and GPC FAIL lights are lit, the corresponding flags (QXBBFCS and QXBGPC) are set, and the RESTART FAIL condition (QXBRSTRT) is set. QXBRSTRT and QTBRSCTR are zeroed in EXRUPT when four restart-free minor cycles occur. The FLIGHT CONTROL FILTER INITIALIZATION flag (HXBFILT) is set only on the first of a group of persistent restarts. If persistent restarts occur during flight control, HXBFILT is never reset, and the filters are initialized each cycle. If the persistent restarts occur after flight control is completed, the filters are initialized only on the first restart since HXBFILT is reset immediately after flight control in the housekeeping job HSDAP.

The telemetry maintenance section of EHMAIN is responsible for maintaining continuity of PCM operations and for reporting RUPTCODE/SUBCODE information properly. First, the minor cycle counter is checked to see whether this is a cycle when PCM should be sent. If it is, then PCM has either been sent or not. If this is a non-PCM cycle, then TALT formatting has either been done or not. These four conditions are described below:

1. PCM frame, output started (QXBPCM = -1).— If PCM is sent again, information will be lost. In this case a flag (PCODE) is set to indicate that a restart occurred in a PCM cycle after the PCM output was sent. This flag causes the telemetry status formatting subroutine executed during the next minor cycle to set the RESTART LAST CYCLE status bit (bit 15 of TLSTWDA2).

2. PCM frame, output not started (QXBPCM = 0).— The restart code calls the subroutines to do appropriate telemetry formatting for this cycle (frame 1 or frame 2), then calls the subroutine to do the actual output. The output subroutine is not called until at least half of the minor cycle has elapsed so that output from the last cycle is not overwritten in the PCM buffer. This is determined by reading the minor cycle timer. A delay is inserted after the output has been started so that it will complete before the onset of the I/O for the next cycle.

3. Non-PCM frame, status formatting done (QXBPCM = -1).— The RUPTCODE/SUBCODE fields in TLSTWDA1 are replaced, and the RESTART OCCURRED flag in TLSTWDA2 is set. Then RUPTCODE, SUBCODE, and PCODE are zeroed.

4. Non-PCM frame, status formatting not done (QXBPCM = 0).— The subroutine TLSTATA is called to format the status words for this cycle.

Finally, the IOPS interval timer is repetitively read until only 200 microseconds remain in the minor cycle. The IOPS reset at the beginning of EHMAIN destroys the IOPS interval timer reload value and eliminates the interrupt that normally occurs at the timer expiration. Therefore, the interval timer is reloaded with

30 milliseconds and the program assumes that the current minor cycle is complete. It then jumps to the address in EXRUPT that begins a new minor cycle.

5.4 HOUSEKEEPING

Those jobs that do not fall under a major function are considered housekeeping jobs. Some of these jobs pick up loose ends for major functions, while others work in conjunction with the system operations to control the program state.

5.4.1 DIGITAL AUTOPILOT HOUSEKEEPING (HSDAP)

HSDAP zeroes the FLIGHT CONTROL FILTER INITIALIZATION flag (HXBFILT) and the POWER-UP INITIALIZATION flag (HXBPWRUP), and sets the FLIGHT CONTROL COMPLETE flag (QXBFC).

5.4.2 CYCLE HOUSEKEEPING (HSCYCLE)

HSCYCLE resets the DUTY CYCLE discrete (discrete output channel, bit 1) set by the executive at EXRUPT. It also sets the ANALOG COMPUTER ON discrete (discrete output channel, bit 2) if the BFCS is engaged or clears the discrete if the BFCS is not engaged.

5.4.3 ANALOG FETCH FAIL-DELAY (HSDELAY)

If the analog fetch fails, none of the flight control jobs are executed, but the surface commands from the previous minor cycle are still sent by the IOCMDS job. The IOPS normally completes the data transfer before the PCM or display output jobs are called because the filter push jobs and the PCM formatting or display formatting jobs are run in parallel with the IOPS operations. Since the filter push jobs are not executed when the analog fetch fails, it is necessary to insert a delay so the PCM or display output job does not find the IOPS busy and reset it, thereby terminating the transfer of the surface commands.

5.4.4 IDLE JOB (HSIDLE)

HSIDLE is a four-instruction loop which reads the IOPS INPUT discretes. If the HALT bit is set, transfer goes to the ground program. If the HALT bit is not set, HSIDLE continues looping until the start of the next minor cycle interrupt.

5.4.5 ENGAGE/DISENGAGE (HSENG)

HSENG implements the OFP ENGAGE/DISENGAGE logic. It includes controlling the ENGAGE flag (IXBG), the FIRST-PASS ENGAGE flag (HXBEN), and the annunciation of certain discrepancies among the ENGAGE discretes.

HSENG computes a new ENGAGE state based on the current value of the ENGAGE discretes and the current ENGAGE state. ENGAGE state information is contained in the ENGAGE flag (IXBG), the PARTIAL ENGAGE flag (HTBEN), and the disengage error counter (QTBCT). Consequently, these three variables may function as both input and output of HSENG.

There are six distinct cases that result from combinations of the four ENGAGE discretes and the ENGAGE flag. A pointer to the code that handles the specified case is extracted from the ENGAGE state table (ENGTAB), using the ENGAGE discretes and the ENGAGE flag.

5.4.5.1 Action When Not Engaged

If the BFCS is not currently engaged (IXBG = 0), the 16 combinations of the four ENGAGE discretes fall into three classes:

1. Not engaged (NOTEN).— This is the state when IXBTB is on and IXBG3, IXBG2, and IXBG1 are all off. Any PARTIAL ENGAGE condition (HTBEN) and the CONTROL UNIT (CU) FAIL flag (QXBG1) are cleared.

2. Partial engage (PARTE).– This is the state when IXBTB is off and less than two of the other three are on or when IXBTB is on and less than three of the other three are on. This condition is allowed to persist for one cycle on the assumption that the values have not yet settled. HTBEN is set to indicate a partial engage. If the routine finds HTBEN set, then the PARTIAL ENGAGE condition has persisted for more than one cycle, so the routine turns on the GPC FAIL light and sets the FAIL flags QXBG1 and QXBGP.

3. First pass engaged (FRSTE).– This is the state when IXBTB is off and at least two out of the other three are on or when IXBTB is on and all three of the others are on. These are legitimate ENGAGE conditions. The BFCS ENGAGE discrete (IXBG), the FIRST-PASS ENGAGE flag (HXBEN), and a counter to cause the hydraulics to be reset for 180 milliseconds (HXB18) are set. The PARTIAL ENGAGE flag (HTBEN) and ERROR flags QXBG1, QXBG2, and QXBCT are reset.

5.4.5.2 <u>Action When Engaged</u>

If the BFCS is currently engaged (IXBG = -1), the 16 combinations of IXBTB, IXBG3, IXBG2, and IXBG1 also fall into three classes:

1. Disengage (DISEN).– This is the state when IXBTB is on and all three of the others are off. Once the BFCS has been engaged, it can be disengaged only by passing through this state. The routine resets the ENGAGE flag (IXBG) and the flags HTBEN, QXBG1, QXBG2, and QTBCT.

2. Still engaged (STILE).– This is the state when IXBTB is off regardless of the condition of the other discretes. This condition represents the second and subsequent ENGAGE cycles. The flags HTBEN, QXBG1, QXBG2, and QTBCT are reset.

3. Fail engage (FAILE).– This is the state when IXBTB is on and one or more of the other three are also on. If this condition

persists for four or more consecutive cycles, the routine sets the BFCS FAIL light and the FAIL flags QXBBF and QXBG2. The flag/counter QTBCT is used to count the four cycles. It is set negative on the fourth consecutive cycle and remains so for the duration of the FAIL condition. STILE or DISEN will reset QTBCT if the failure clears.

5.5 SUM CHECK (FDSLF)

The sum check job computes the sum of all static words (instructions and data constants) in the OFP. This value is compared with a predetermined sum. If a discrepancy is found, the GPC FAIL flag (QXBGP) is set, the SUM CHECK FAIL flag (QXBST) is set, and the GPC FAIL light is turned on. If the computed sum agrees with the predetermined value, QXBST is zeroed.

Approximately 80 milliseconds are required to compute the sum of OFP code, so this task cannot be completed in one minor cycle. To resolve this problem, sum check runs as the next to the last job in the minor cycle. It runs until interrupted by the start of the next minor cycle or until it completes, whichever comes first. The EXRUPT routine determines if sum check was executing at the time of the interrupt and, if so, saves the registers and resume address. Sum check restores them the next time it is called and picks up where it left off.

On completion of the sum computation, the appropriate condition is annunciated. If the sum check was valid, the program returns to the executive, which will then transfer to HSIDLE. If the sum check was not valid, RUPTCODE is set to 11, SUBCODE is set to zero, and the program goes to EHMAIN. The next time sum check is called, it starts over from the beginning.

6. DETAILED DESCRIPTION OF OFP I/O ROUTINE

6.1 INTRODUCTION

A table structure is used for input and output. Input is initially stored in the Shuttle Input List (S-IN list) in the format defined by the external source. Once the input is formatted and converted to floating-point quantities, it is stored in the Floating Point and Formatted Input List (F-IN list). Similarly, flight control output is first computed in floating point and stored in the F-OUT list. It is then formatted to conform with the external command formats and stored in the S-OUT list. Display and PCM output does not pass through the F-OUT list. It is directly formatted into the S-OUT list.

The software is organized into six fetch jobs and three output jobs. The fetch jobs are called every minor cycle to update the S-IN list. The FC command output job is called every minor cycle to issue new output from the S-OUT list only if the BFCS is engaged. The display and PCM jobs are called on alternate cycles so that they issue their respective output every other minor cycle. The PCM alternates between two sets of output parameters (frames 1 and 2) so that each frame is issued every fourth minor cycle.

The IOPS control software differs from the rest of the OFP software in that it extends beyond purely CPU code. It also incorporates code which is executed by the IOPS. The IOPS is a programmable I/O device whose electronics include an MSC and BCE's. The MSC and BCE's execute code in parallel with the CPU. All three levels of code (CPU, MSC, and BCE) are required to perform the I/O functions. The CPU code initiates IOPS processing by initiating MSC processing. The MSC in turn initiates I/O operations on the data busses by initiating BCE processing.

The predetermined MSC/BCE programs are loaded as part of the power-up initialization when the OFP is first entered.

Only 56 out of 90 signals are implemented by the BFCSFE hardware. (See appendix E. Zeroes are inserted at all the places marked "NOT USED".)

6.2 ERROR HANDLING

Five of the six fetch jobs call a common routine IOFETCH. IOFETCH has two procedures for detecting a fetch error. The first is to start the job-specified MSC program and wait for the job-specified amount of time for the MSC to go idle. If the MSC does not go idle in this time, an error is indicated. The second test is to read the GO/NOGO status of BCE 5. If it is NOGO, an error is indicated.

In the event that a fetch error is detected, a TEMPORARY FAIL flag relevant to the appropriate fetch job is set. If a fetch job incurs an error in three successive cycles, a HARD FAIL flag relevant to the appropriate fetch job is set. If in a subsequent minor cycle, a previously failed fetch job should fetch error-free information, both the TEMPORARY and HARD FAIL flags are reset. Subroutine IOERROR performs this function for five of the six fetch jobs. The flags are used by other jobs to ascertain the validity of the data in the S-IN list.

6.3 FETCH JOBS

6.3.1 AIR DATA FETCH JOB (IOAIRD)

IOAIRD zeroes the S-IN list and routinely enables the MIA receivers for input. It calls IOFETCH to fetch the 15 air data inputs (JXSAD to JXSAL) defined in appendix E. It then calls IOERROR to maintain the AIR DATA TEMPORARY FAIL flag (QTBAI) and the AIR DATA HARD FAIL flag (QXBAI). If an error does occur, the air data fetch input parameters will consist of an unknown mix of new input and input left over from the previous cycle.

6.3.2 DISCRETE FETCH NUMBER 1 JOB (IO1DISC)

IO1DISC fetches the one discrete word (JXB1), which is defined in appendix E. This word contains the SPEEDBRAKE ENGAGE TAKEOVER discrete in bit 9. It then calls IOERROR to maintain the DISCRETE FETCH NUMBER 1 JOB TEMPORARY FAIL flag (QTBD1) and the DISCRETE FETCH NUMBER 1 JOB HARD FAIL flag (QXBD1).

6.3.3 DISCRETE FETCH NUMBER 2 JOB (IO2DISC)

This job fetches the three discrete words (JXB2, JXB3, and JXB4). JXB2 contains body flap, WOW, and flight control (FC) reset commands. JXB3 is not used. JXB4 contains PANEL TRIM discretes. QTBD2 and QXBD2 are the flags associated with this job.

6.3.4 DISCRETE FETCH NUMBER 3 JOB (IO3DISC)

This job fetches the one discrete word (JXB5), which contains the mode (HALT, STANDBY, or RUN), BFCS engaged, and Input/Output Processor (IOP) TERMINATE B COMMAND discretes. This fetch differs from all the other fetches in that it derives its data from the IOPS discrete input word and not from CDU 5. The IOPS discrete input is accessed using program control. No failure flags are associated with this job.

6.3.5 DISCRETE FETCH NUMBER 4 JOB (IO4DISC)

This job fetches the one discrete word, JXB6. This word contains the RADAR LOCK-ON discrete. QTBD4 and QXBD4 are the flags associated with this job.

6.3.6 ANALOG FETCH JOB (IOANAL)

IOANAL sets the IOPS OUTPUT discrete, bit 10 (analog fetch start) and clears the IOPS OUTPUT discrete, bit 11 (FC output to aft MDM). It then fetches the 20 analog input words (JXCBR to JXSNZ). These words contain the cockpit commands, surface position feed-back, sensed rate, and sensed acceleration input. The flags associated with this job are QTBAN and QXBAN. If the analog FETCH HARD FAIL flag (QXBAN) is set, IOANAL lights the BFCS FAIL light and sets the internal BFCS FAIL flag (QXBBF) for use by the display formatting job. This flag causes the BFCS FAIL discrete also to be issued by the display output job thereafter. If a subsequent analog fetch job is successful, the ANALOG FETCH TEMPORARY and HARD FAIL flags are reset. However, the BFCS FAIL light is not turned off by the analog fetch job. Since the BFCS FAIL light can be turned on by more than one job (the light reflects more than one OFP failure mechanism), the display formatting job determines whether or not the BFCS FAIL light should be turned off.

6.4 OUTPUT JOBS

6.4.1 FC COMMAND OUTPUT JOB (IOCMDS)

IOCMDS routinely enables transmitters on BCE's 1, 2, and 5. It then toggles the IOPS OUTPUT discretes, bit 10 (analog fetch start) and bit 11 (FC output to aft MDM) to the opposite of their condition set by IOANAL. These discretes can be used to measure the transport delay. IOCMDS issues the flight control command output in two steps. First it transmits nine words (OXIRA to O[illegible]LA). Then it waits for the MSC to be idle and transmits two words (OX1A and OX2A). After the MSC becomes idle, the job returns to the executive.

6.4.2 DISPLAY OUTPUT JOB (IODISP)

IODISP routinely enables transmitters on BCE's 1, 2, and 5 and disables the receivers on BCE's 1, 2, and 5. It then outputs the IOPS discretes that are carried in OXB5 and OXB5B; namely, bit 0 (GPC fail), bit 8 (BFCS fail), and bit 9 (GPC ready). Next the nine words (OXDBF to OXDIL) to the surface position indicator (SPI) are output. The MSC is allowed to go idle, and then the 14 words (OXDCD to OXDEY) for the attitude director indicator (ADI) are output. The MSC is allowed to go idle, and then the six words (OXDCV to OXDNZ) for the altitude and vertical velocity indicator (AVVI) are output. Actually this output is meaningless since the current BFCSFE has no AVVI displays. The MSC is allowed to go idle, and then the six words (OXDCM to OXDNX) for the altitude and Mach indicator (AMI) are transmitted. The MSC is allowed to idle, and then two words (OXB3 and OXB4) are transmitted for SPI discretes. The MSC is allowed to idle, and then the program returns to the executive.

6.4.3 PCM OUTPUT JOB (IOPCM)

IOPCM issues the 32 telemetry outputs (PCM01 to PCM32). Two different PCM output frames are issued alternately. IOPCM issues output from the same locations in memory each time it is called by the executive. It is the function of the telemetry formatting jobs (TLFR1, TLFR2, and TLALT) to fill these locations with the properly formatted information. If the MSC is busy (an abnormal condition), the IOP is reset so that the output can be properly issued. When the PCM output is initiated, a PCM flag (QXBPC) is set for use by the restart job.

7. SUMMARY

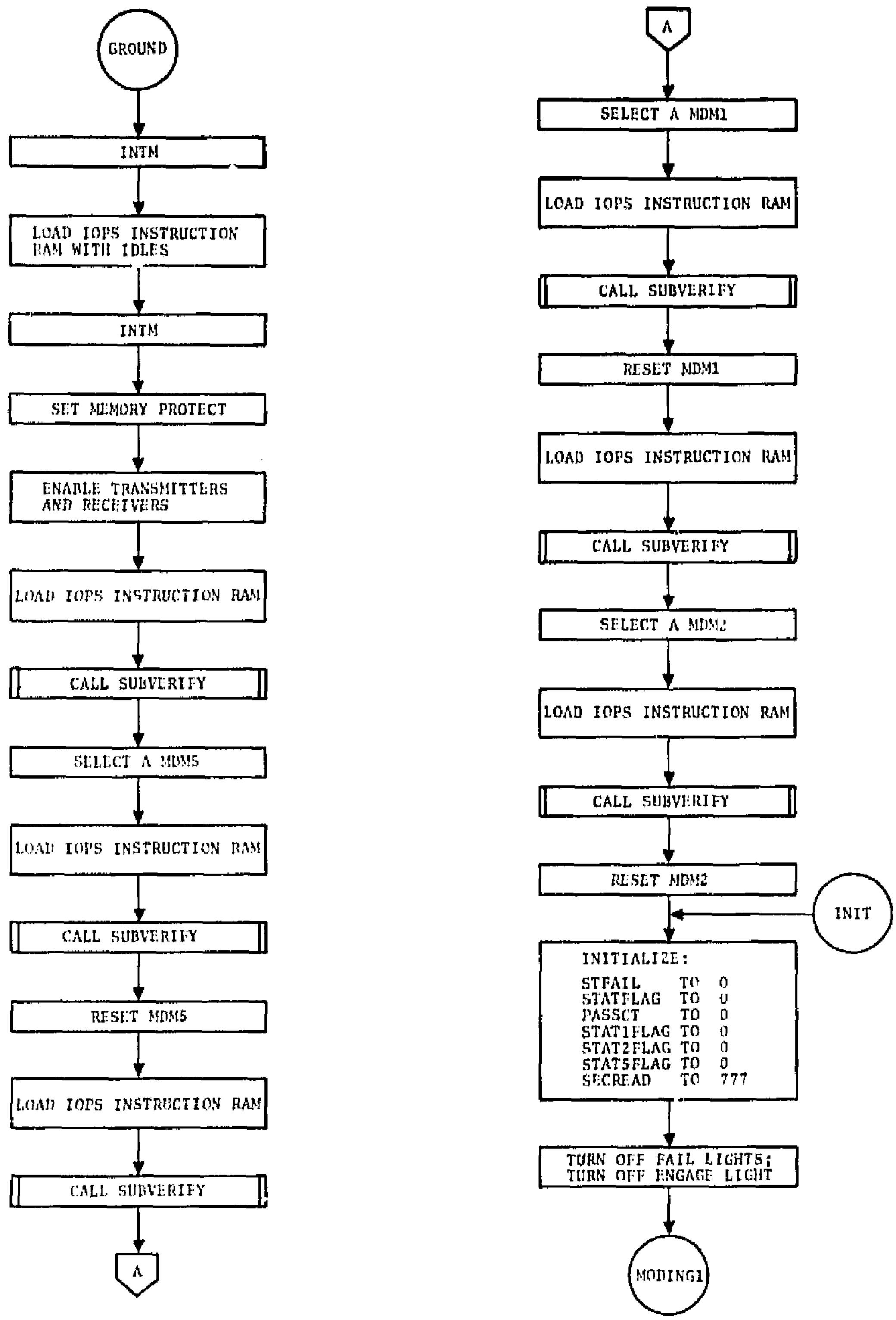
The BFCSFE software conforms as closely as possible (within hardware limitations) to the AP101 software. This document augments the Nova program listing and provides insight into the methods and techniques used to generate and check out the program. In addition, it provides detailed descriptions of the Nova routines that are significantly different from their AP101 counterparts.

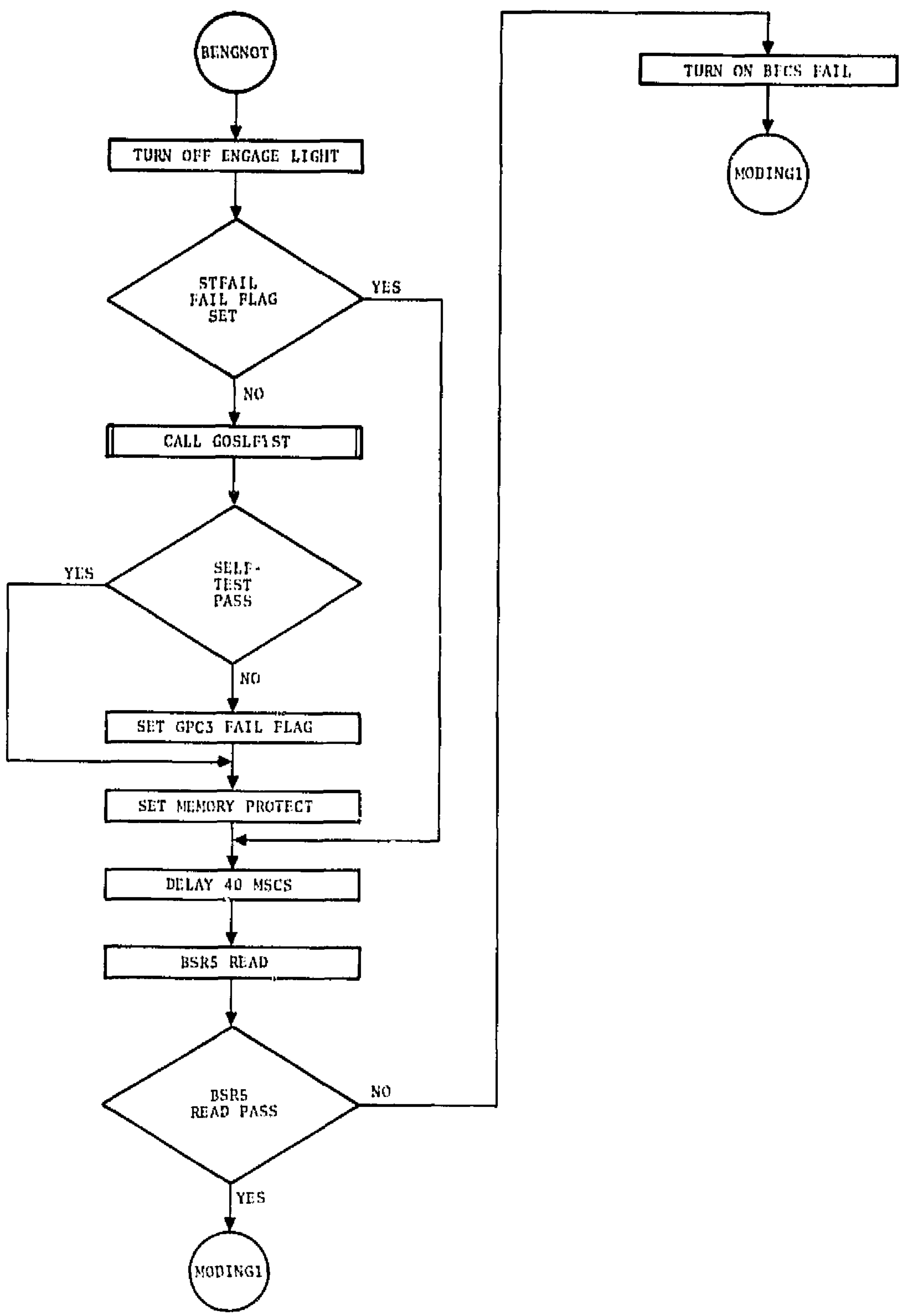
APPENDIX A

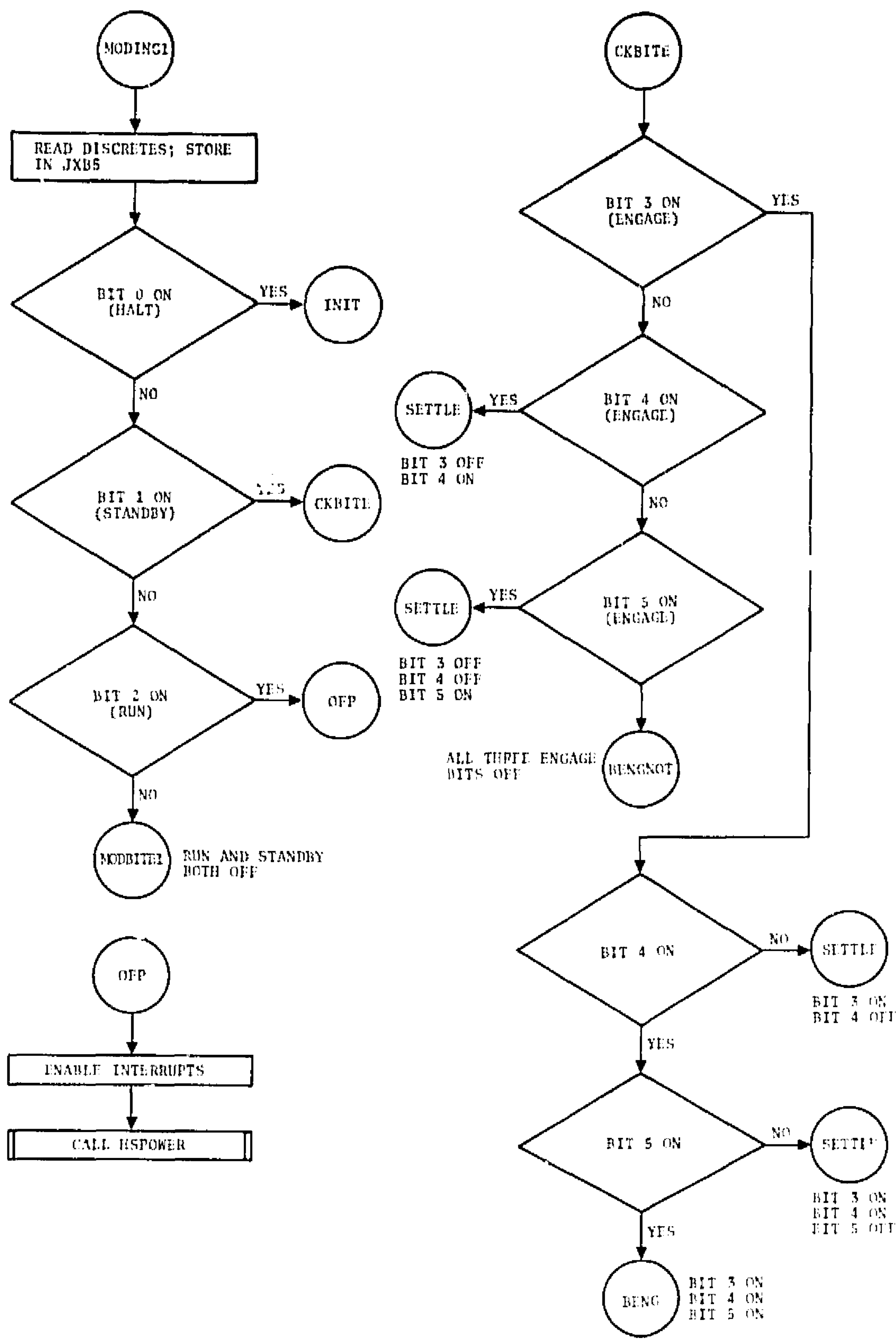
BFCS GROUND SUPPORT PROGRAM
FLOW CHARTS

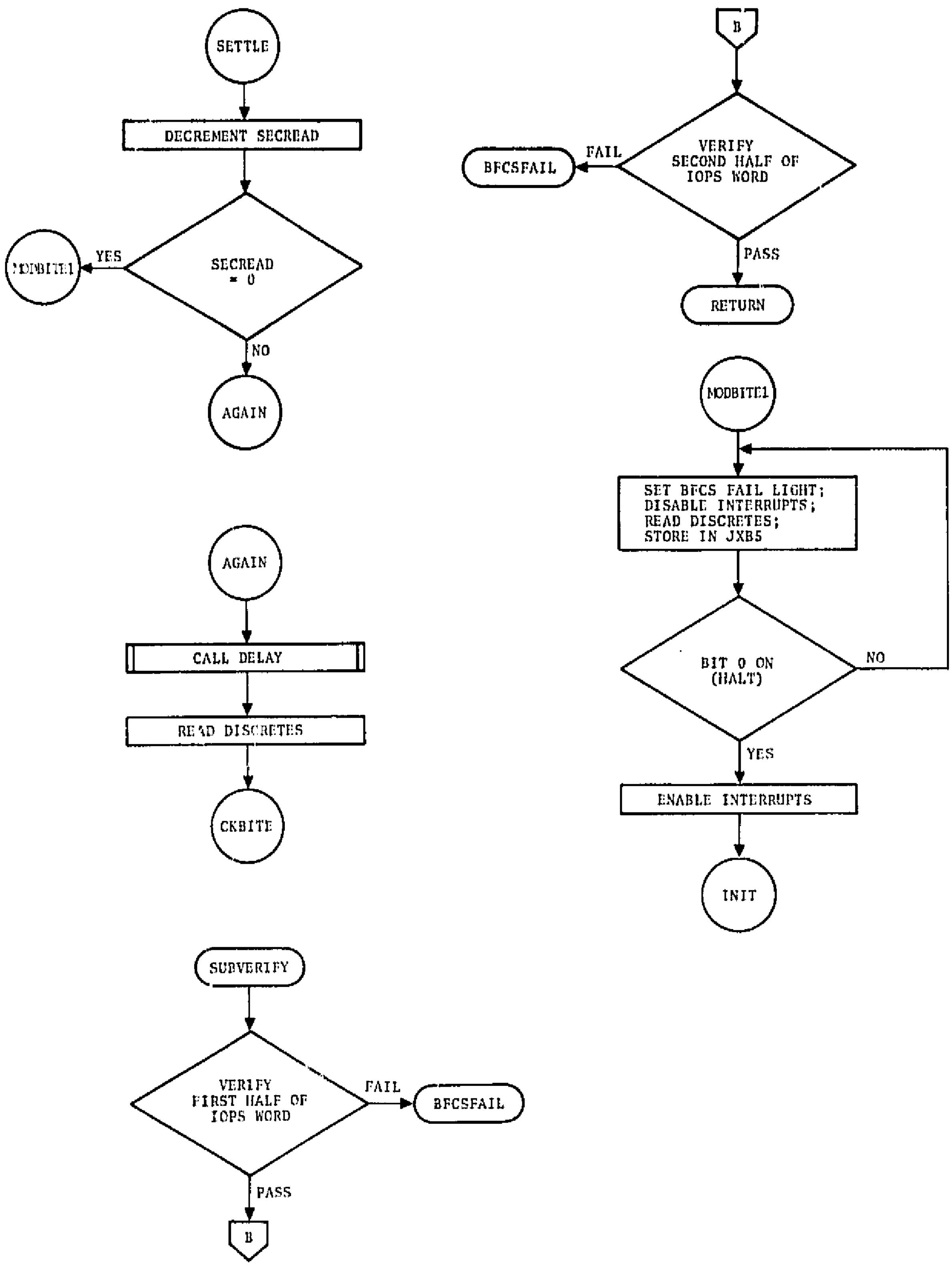
APPENDIX A

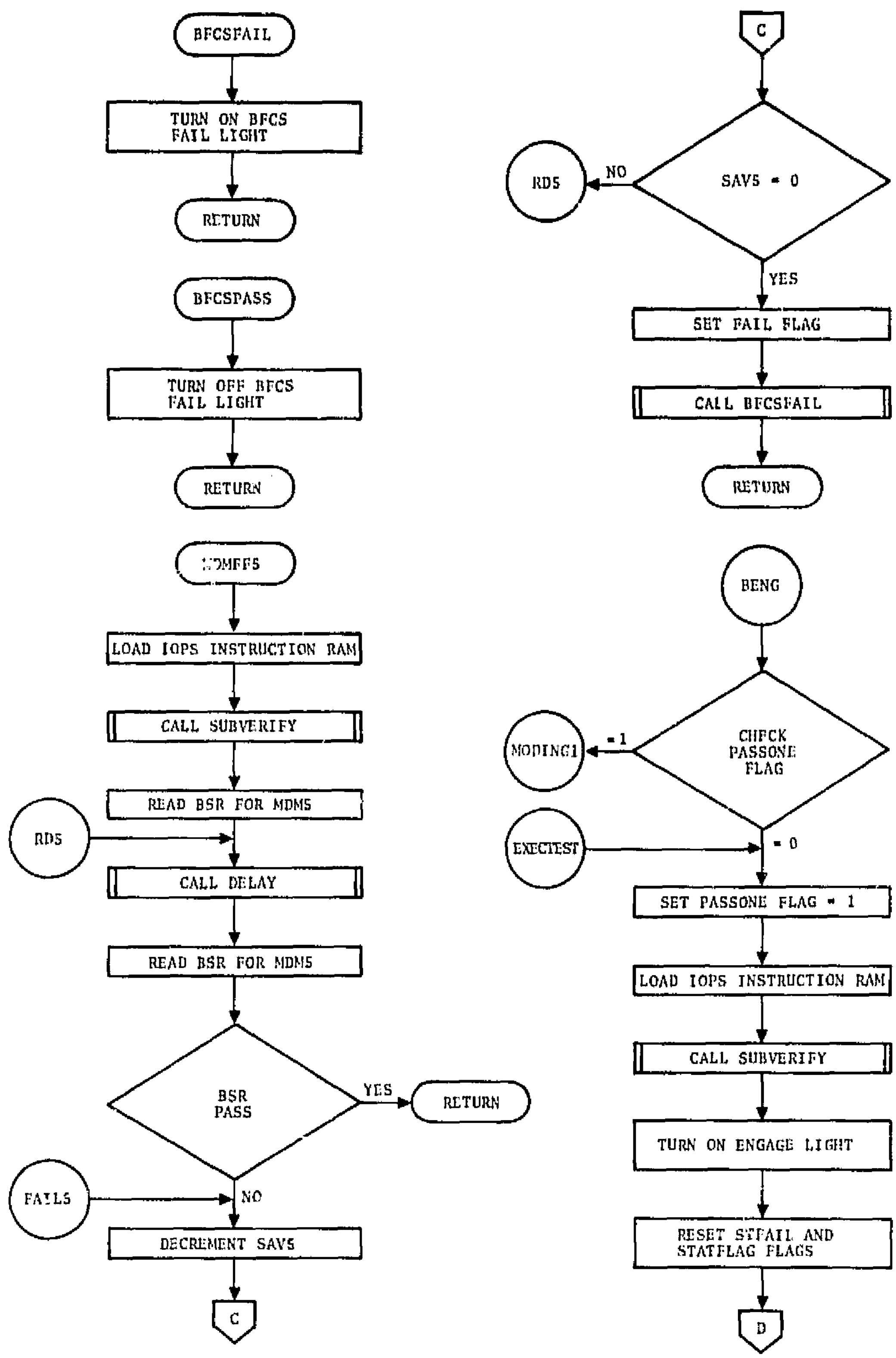
BFCS GROUND SUPPORT PROGRAM FLOW CHARTS

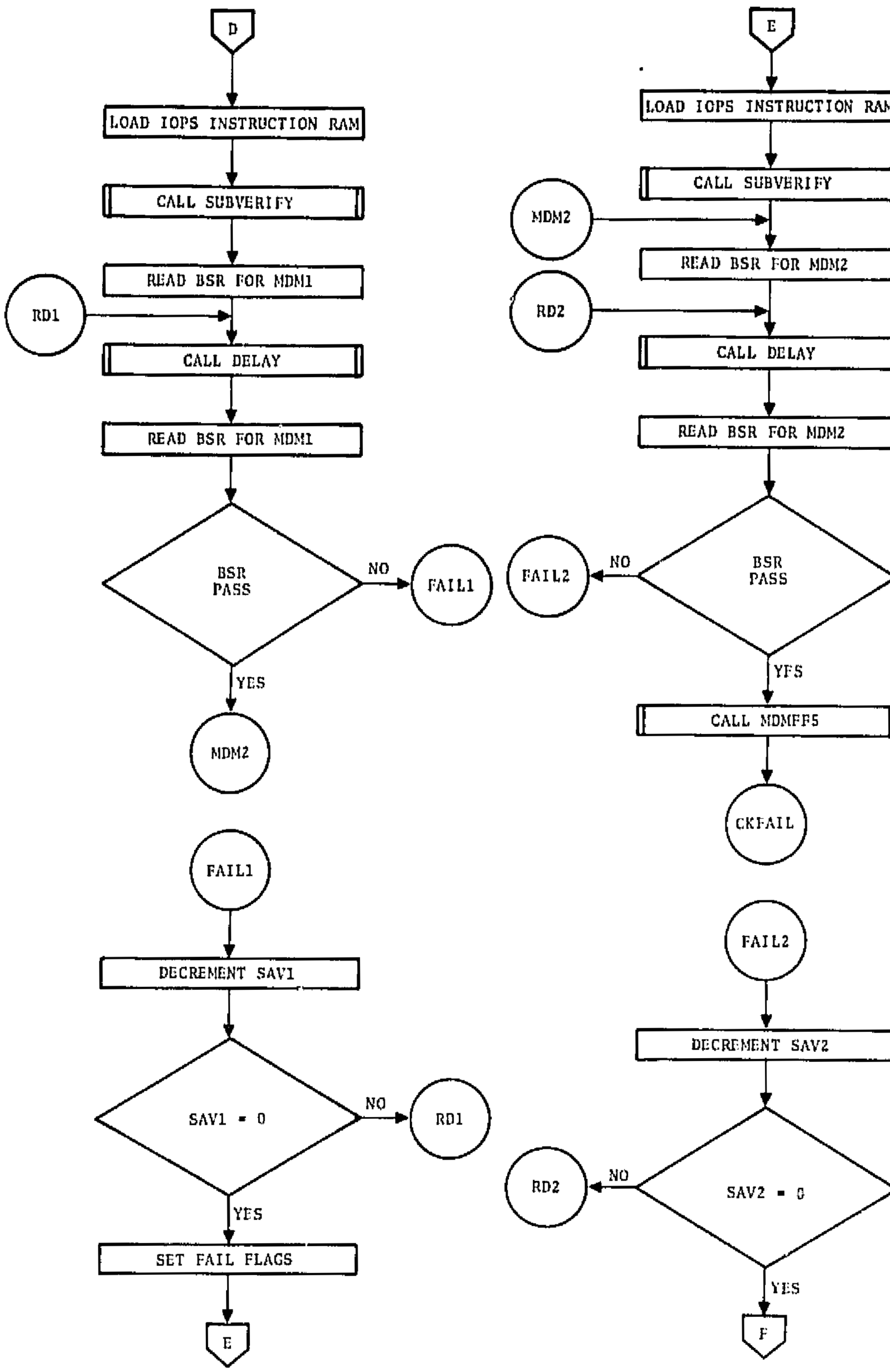


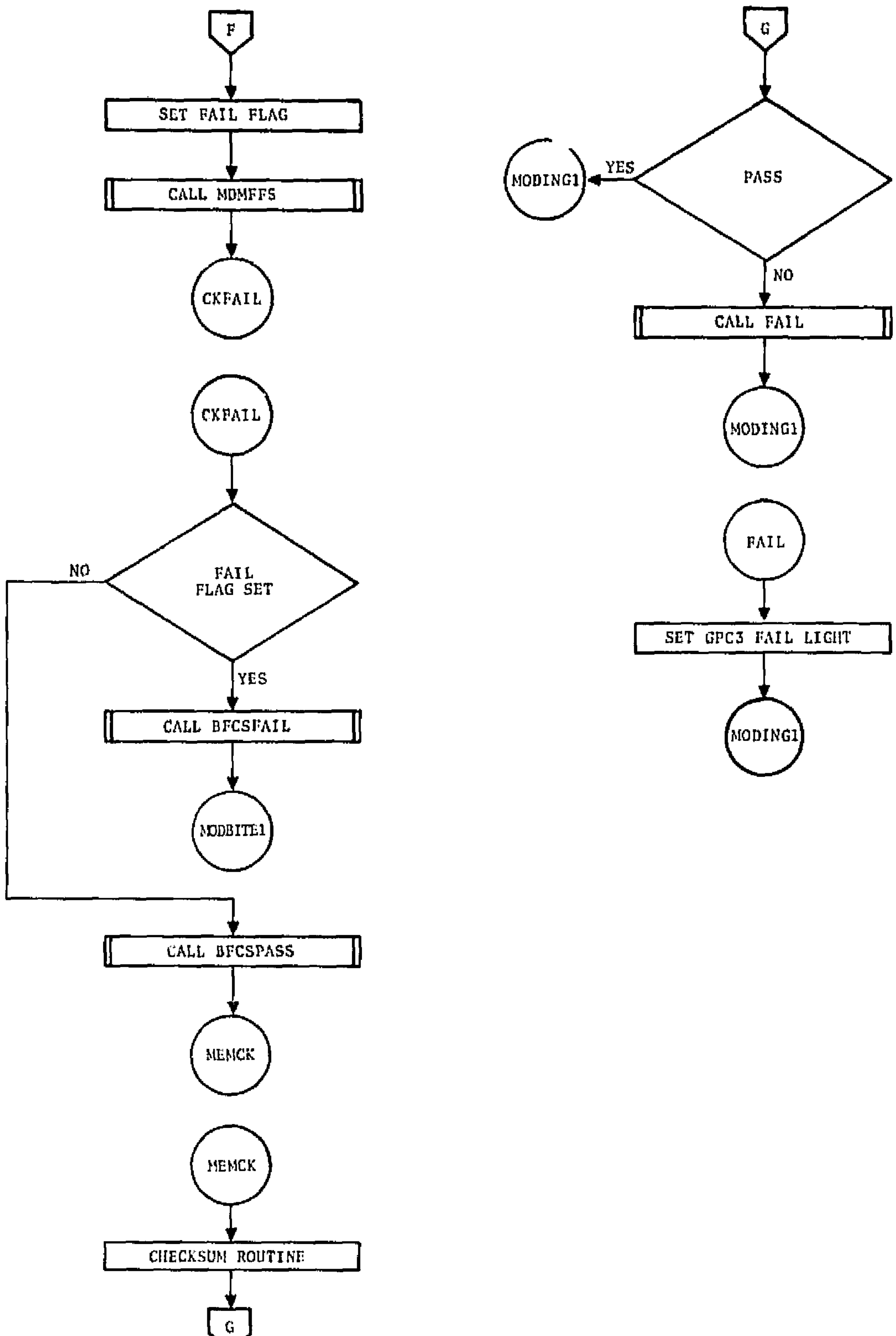

BENGNOT
TURN OFF ENGAGE LIGHT
STFAIL FAIL FLAG SET
YES
NO
CALL GOSLFTST
SELF-TEST PASS
YES
NO
SET GPC3 FAIL FLAG
SET MEMORY PROTECT
DELAY 40 MSCS
BSR5 READ
BSR5 READ PASS
NO
YES
MODING1
TURN ON BFCS FAIL
MODING1

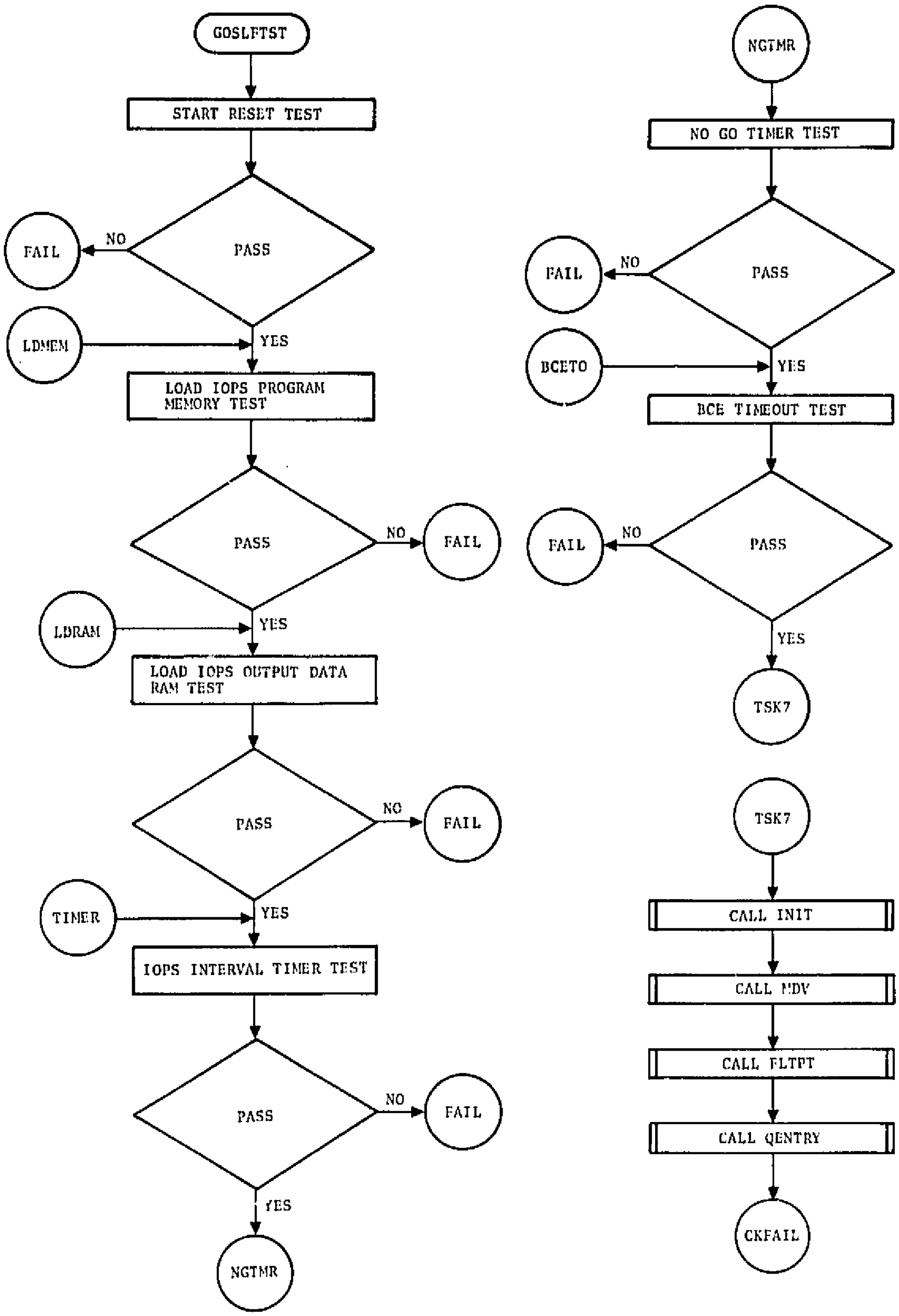

MODING1
READ DISCRETES; STORE IN JXB5
BIT 0 ON (HALT)
YES
INIT
NO
BIT 1 ON (STANDBY)
CKBITE
NO
BIT 2 ON (RUN)
YES
OFP
NO
MODBITE1
RUN AND STANDBY BOTH OFF
OFP
ENABLE INTERRUPTS
CALL HSPOWER
CKBITE
BIT 3 ON (ENGAGE)
YES
NO
BIT 4 ON (ENGAGE)
YES
SETTLE
BIT 3 OFF
BIT 4 ON
NO
BIT 5 ON (ENGAGE)
YES
SETTLE
BIT 3 OFF
BIT 4 OFF
BIT 5 ON
ALL THREE ENGAGE BITS OFF
BENGNOT
BIT 4 ON
NO
SETTLE
BIT 3 ON
BIT 4 OFF
YES
BIT 5 ON
NO
SETTLE
BIT 3 ON
BIT 4 ON
BIT 5 OFF
YES
BENG
BIT 3 ON
BIT 4 ON
BIT 5 ON


SETTLE
DECREMENT SECREAD
SECREAD = 0
YES
MODBITE1
NO
AGAIN
AGAIN
CALL DELAY
READ DISCRETES
CKBITE
SUBVERIFY
VERIFY FIRST HALF OF IOPS WORD
FAIL
BFCSFAIL
PASS
B
B
VERIFY SECOND HALF OF IOPS WORD
FAIL
BFCSFAIL
PASS
RETURN
MODBITE1
SET BFCS FAIL LIGHT;
DISABLE INTERRUPTS;
READ DISCRETES;
STORE IN JXB5
BIT 0 ON (HALT)
NO
YES
ENABLE INTERRUPTS
INIT

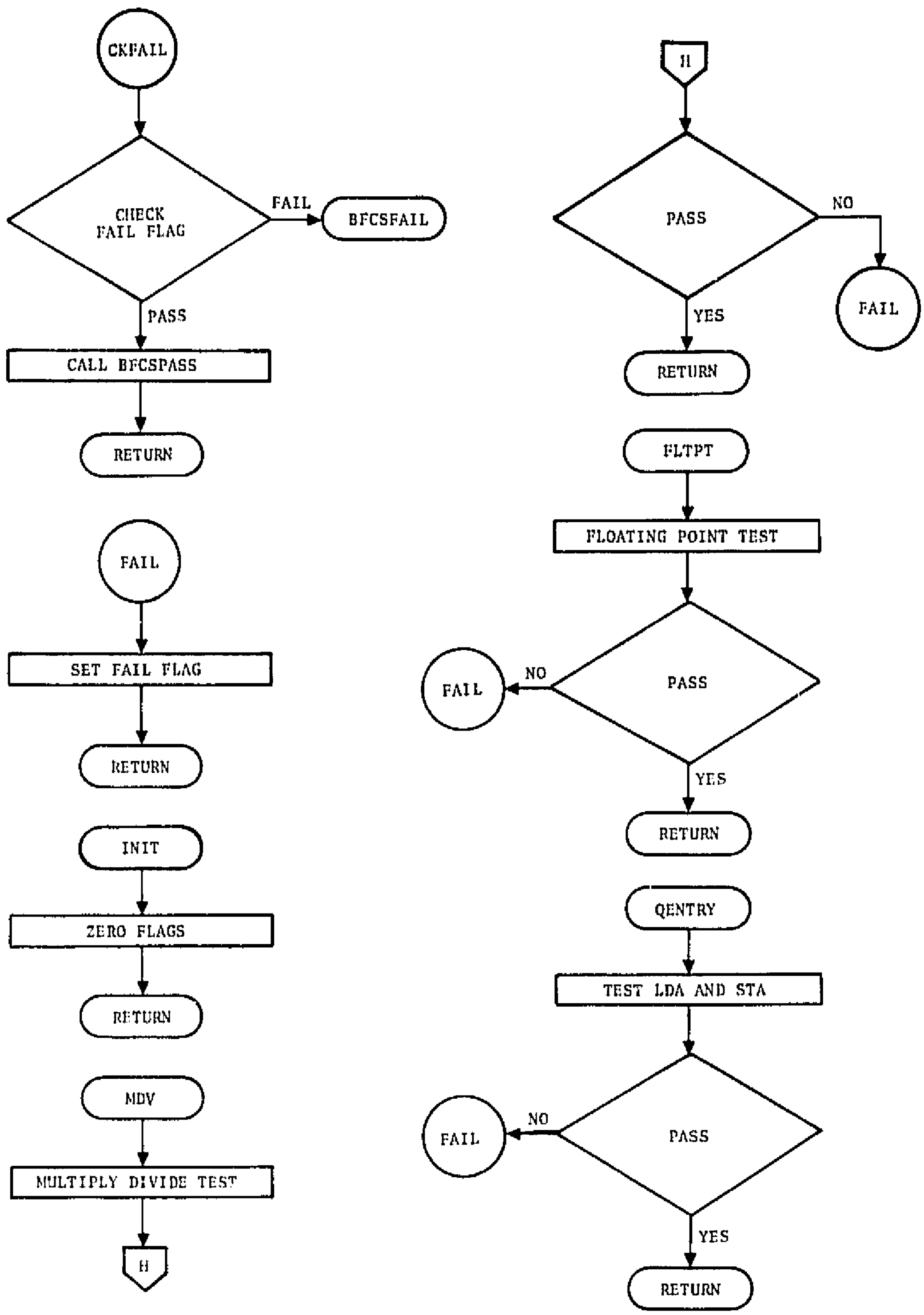

BFCSFAIL
TURN ON BFCS FAIL LIGHT
RETURN
BFCSPASS
TURN OFF BFCS FAIL LIGHT
RETURN
LOAD IOPS INSTRUCTION RAM
CALL SUBVERIFY
READ BSR FOR MDMS
RDS
CALL DELAY
READ BSR FOR MDMS
BSR PASS
YES
RETURN
FAILS
NO
DECREMENT SAVS
C
C
SAVS = 0
NO
RDS
YES
SET FAIL FLAG
CALL BFCSFAIL
RETURN
BENG
CHECK PASSONE FLAG
= 1
MODING1
EXECTEST
= 0
SET PASSONE FLAG = 1
LOAD IOPS INSTRUCTION RAM
CALL SUBVERIFY
TURN ON ENGAGE LIGHT
RESET STFAIL AND STATFLAG FLAGS
D


D
LOAD IOPS INSTRUCTION RAM
CALL SUBVERIFY
READ BSR FOR MDM1
RD1
CALL DELAY
READ BSR FOR MDM1
BSR PASS
NO
FAIL1
YES
MDM2
FAIL1
DECREMENT SAV1
SAV1 = 0
NO
RD1
YES
SET FAIL FLAGS
E
E
LOAD IOPS INSTRUCTION RAM
CALL SUBVERIFY
MDM2
READ BSR FOR MDM2
RD2
CALL DELAY
READ BSR FOR MDM2
BSR PASS
NO
FAIL2
YES
CALL MDMFF5
CKFAIL
FAIL2
DECREMENT SAV2
SAV2 = 0
NO
RD2
YES
F


F
SET FAIL FLAG
CALL MDMFFS
CKFAIL
CKFAIL
FAIL FLAG SET
NO
YES
CALL BFCSFAIL
MODBITE1
CALL BFCSPASS
MEMCK
MEMCK
CHECKSUM ROUTINE
G
G
PASS
YES
MODING1
NO
CALL FAIL
MODING1
FAIL
SET GPC3 FAIL LIGHT
MODING1



NOTE: INIT, FAIL, AND CKFAIL IN GOSLFTST ARE NOT THE SAME AS INIT, FAIL, AND CKFAIL USED PREVIOUSLY. GOSLFTST WAS ASSEMBLED SEPARATELY SINCE IT CONTAINS NAMES IDENTICAL TO THE NORTH AMERICAN ROCKWELL MACROS.



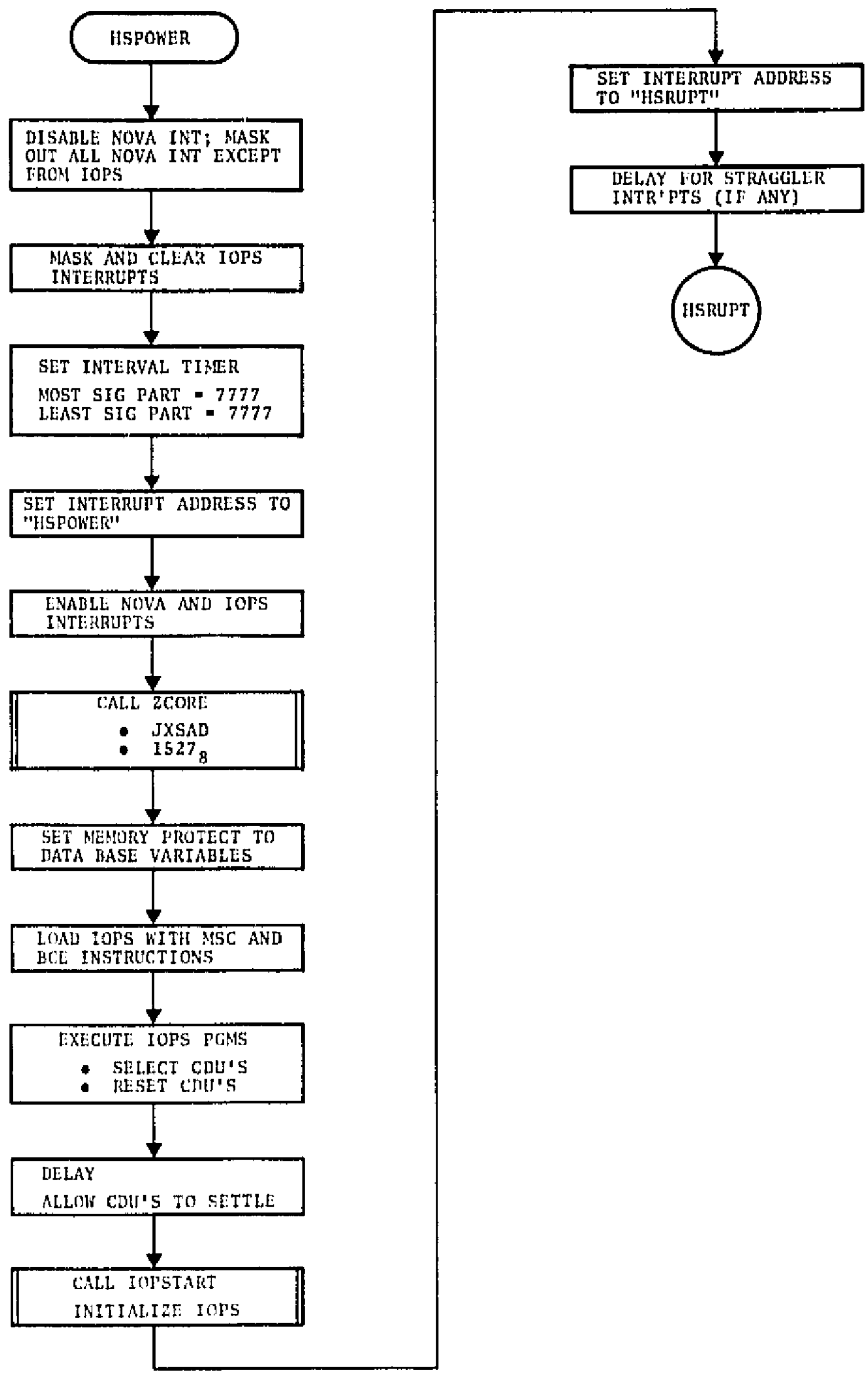
CKFAIL
CHECK FAIL FLAG
FAIL
BFCSFAIL
PASS
CALL BFCSPASS
RETURN
FAIL
SET FAIL FLAG
RETURN
INIT
ZERO FLAGS
RETURN
MDV
MULTIPLY DIVIDE TEST
H
H
PASS
NO
FAIL
YES
RETURN
FLTPT
FLOATING POINT TEST
FAIL
NO
PASS
YES
RETURN
QENTRY
TEST LDA AND STA
FAIL
NO
PASS
YES
RETURN

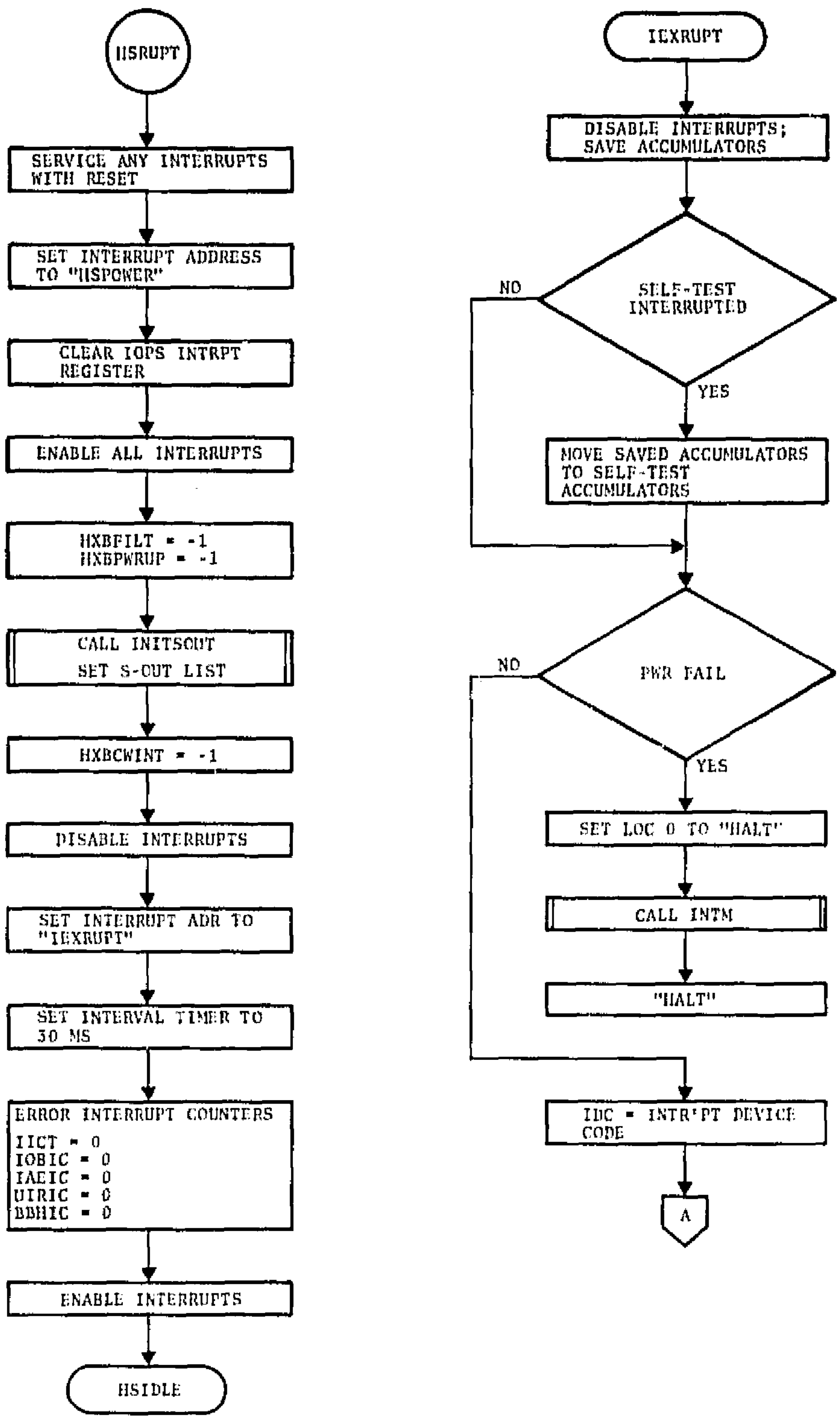
APPENDIX B

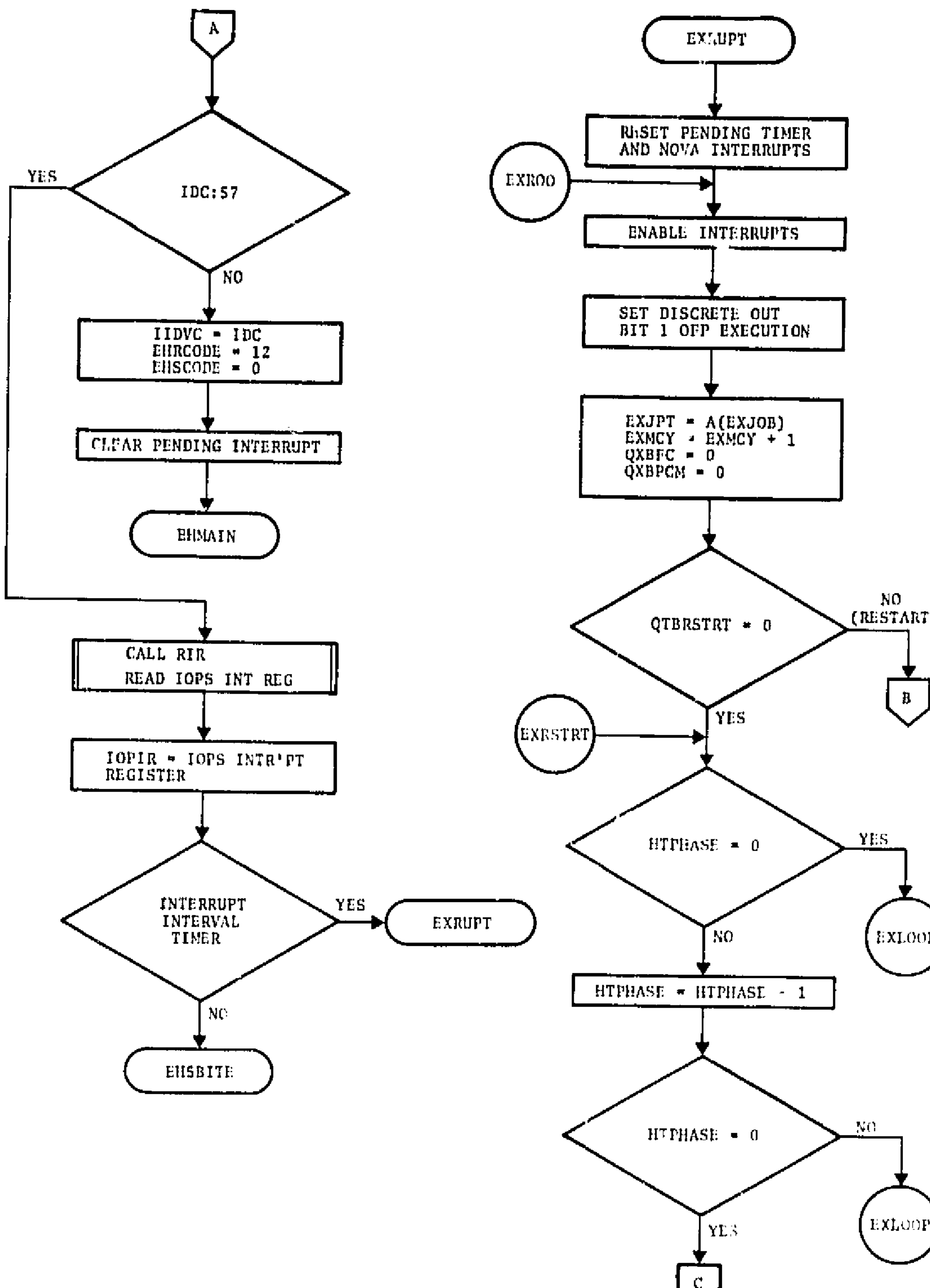
OFP EXECUTIVE FLOW CHARTS

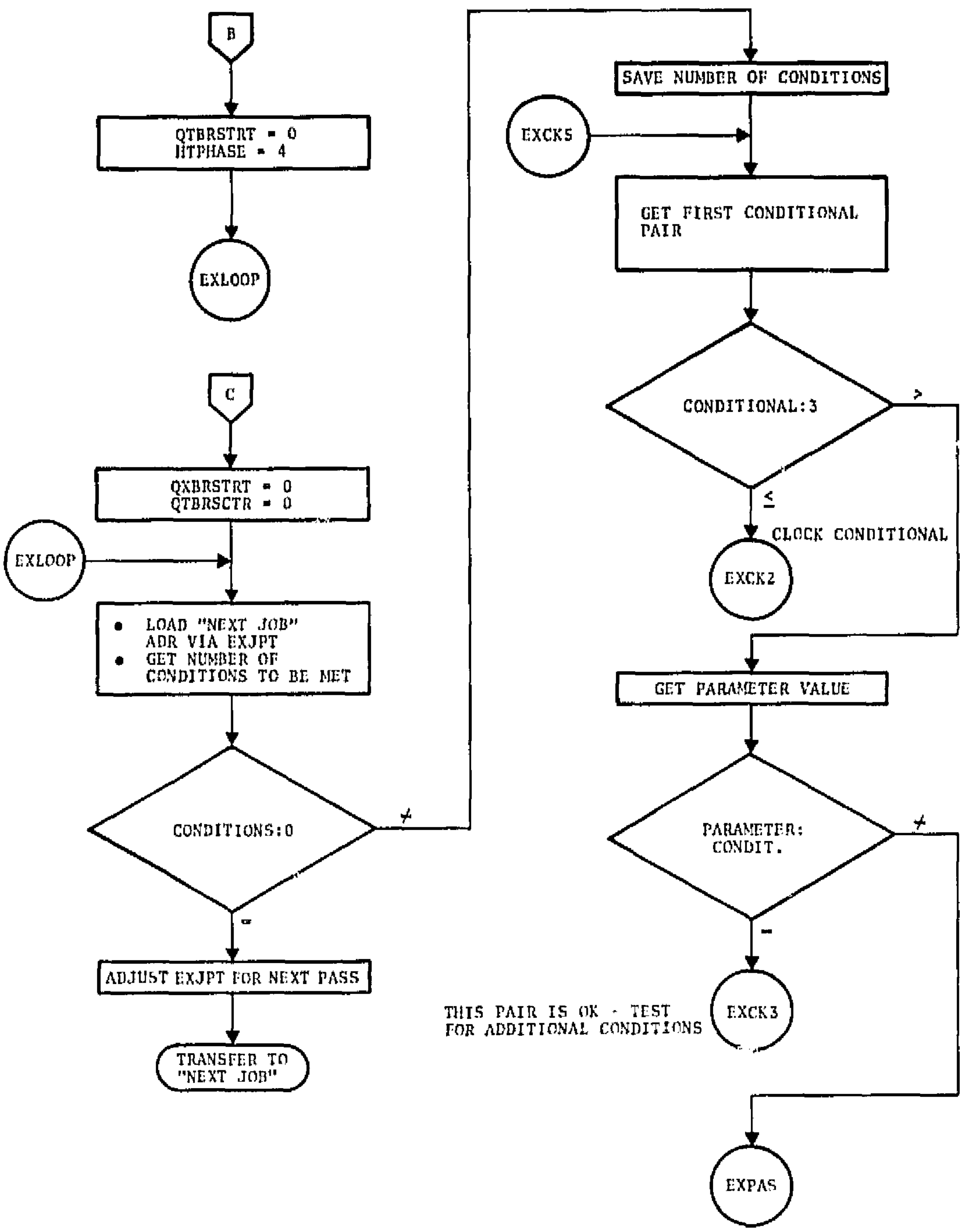
APPENDIX B

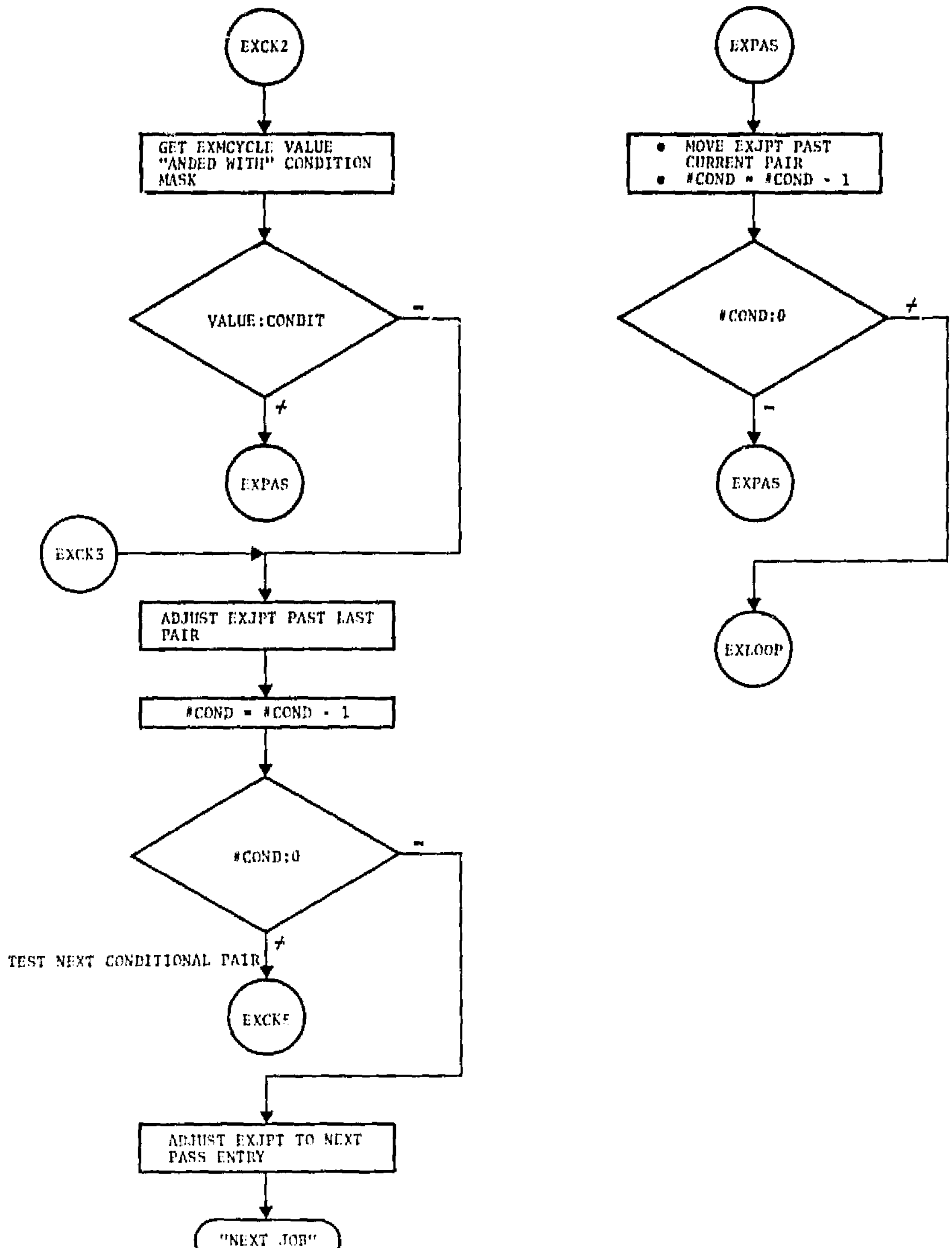
OFP EXECUTIVE FLOW CHARTS

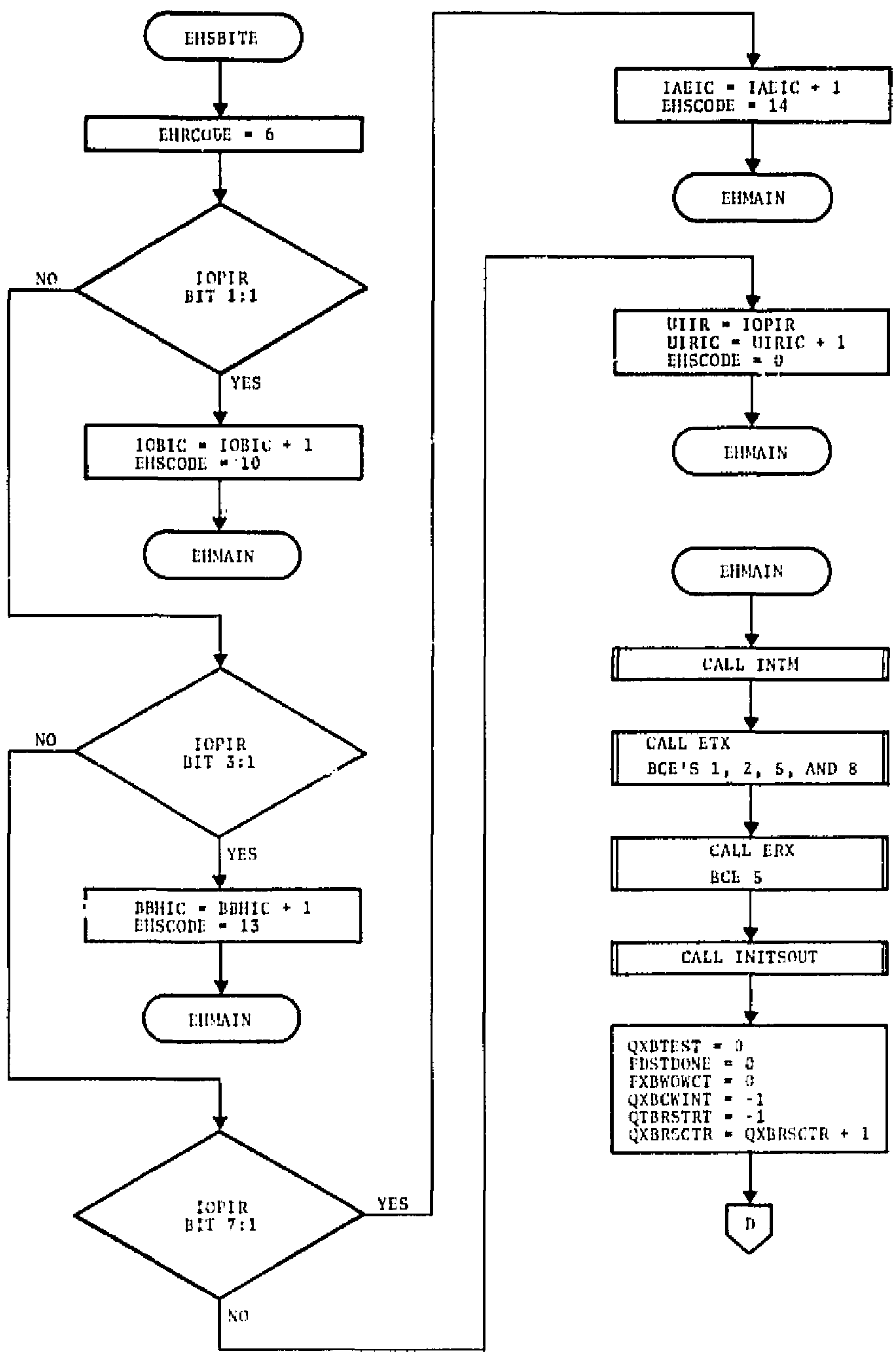


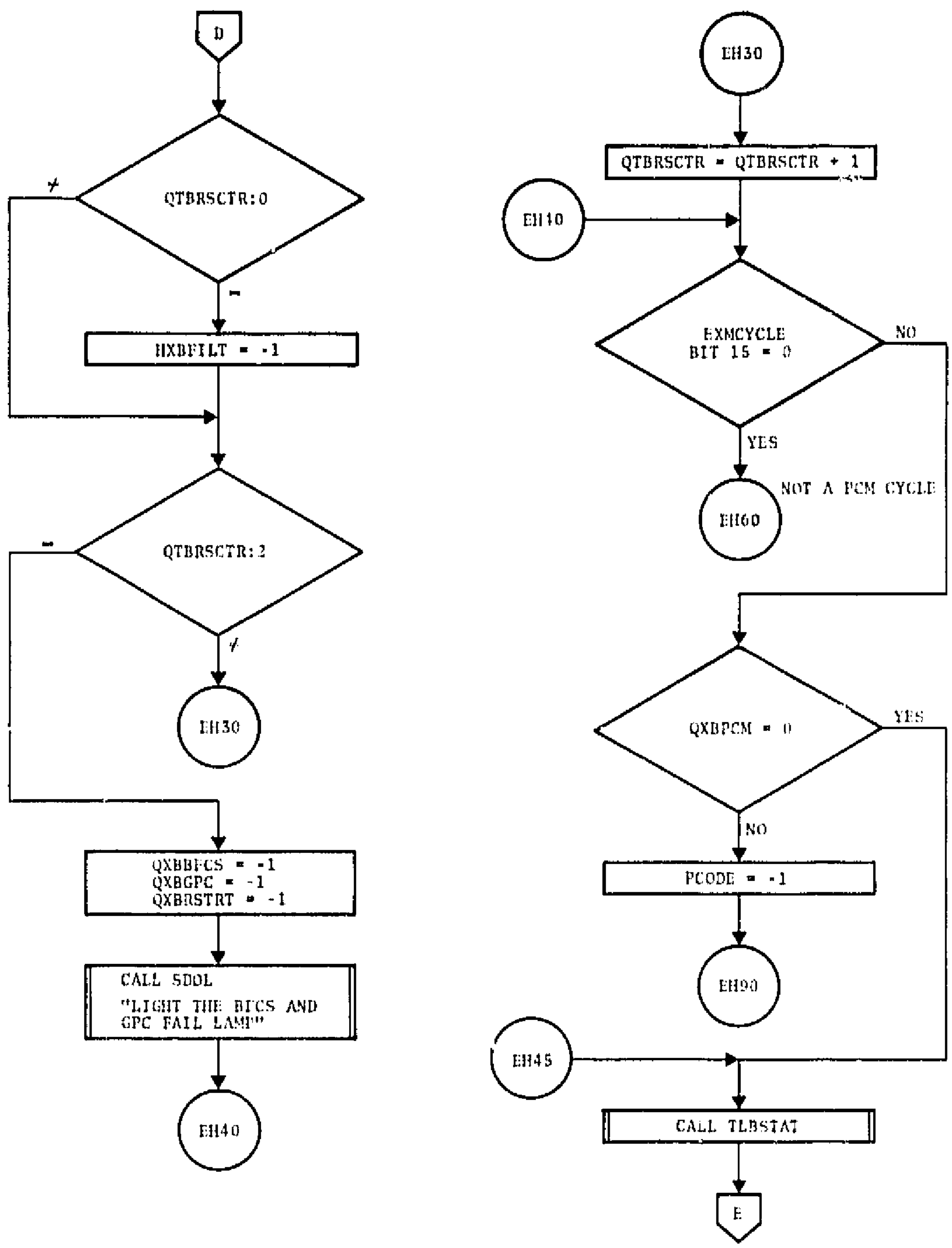

HSRUPT
SERVICE ANY INTERRUPTS WITH RESET
SET INTERRUPT ADDRESS TO "HSPOWER"
CLEAR IOPS INTRPT REGISTER
ENABLE ALL INTERRUPTS
HXBFILT = -1
HXBPWRUP = -1
CALL INITSOUT
SET S-OUT LIST
HXBCWINT = -1
DISABLE INTERRUPTS
SET INTERRUPT ADR TO "IEXRUPT"
SET INTERVAL TIMER TO 30 MS
ERROR INTERRUPT COUNTERS
IICT = 0
IOBIC = 0
IAEIC = 0
UIRIC = 0
BBHIC = 0
ENABLE INTERRUPTS
HSIDLE
IEXRUPT
DISABLE INTERRUPTS; SAVE ACCUMULATORS
SELF-TEST INTERRUPTED
NO
YES
MOVE SAVED ACCUMULATORS TO SELF-TEST ACCUMULATORS
PWR FAIL
NO
YES
SET LOC 0 TO "HALT"
CALL INTM
"HALT"
IDC = INTR'PT DEVICE CODE
A

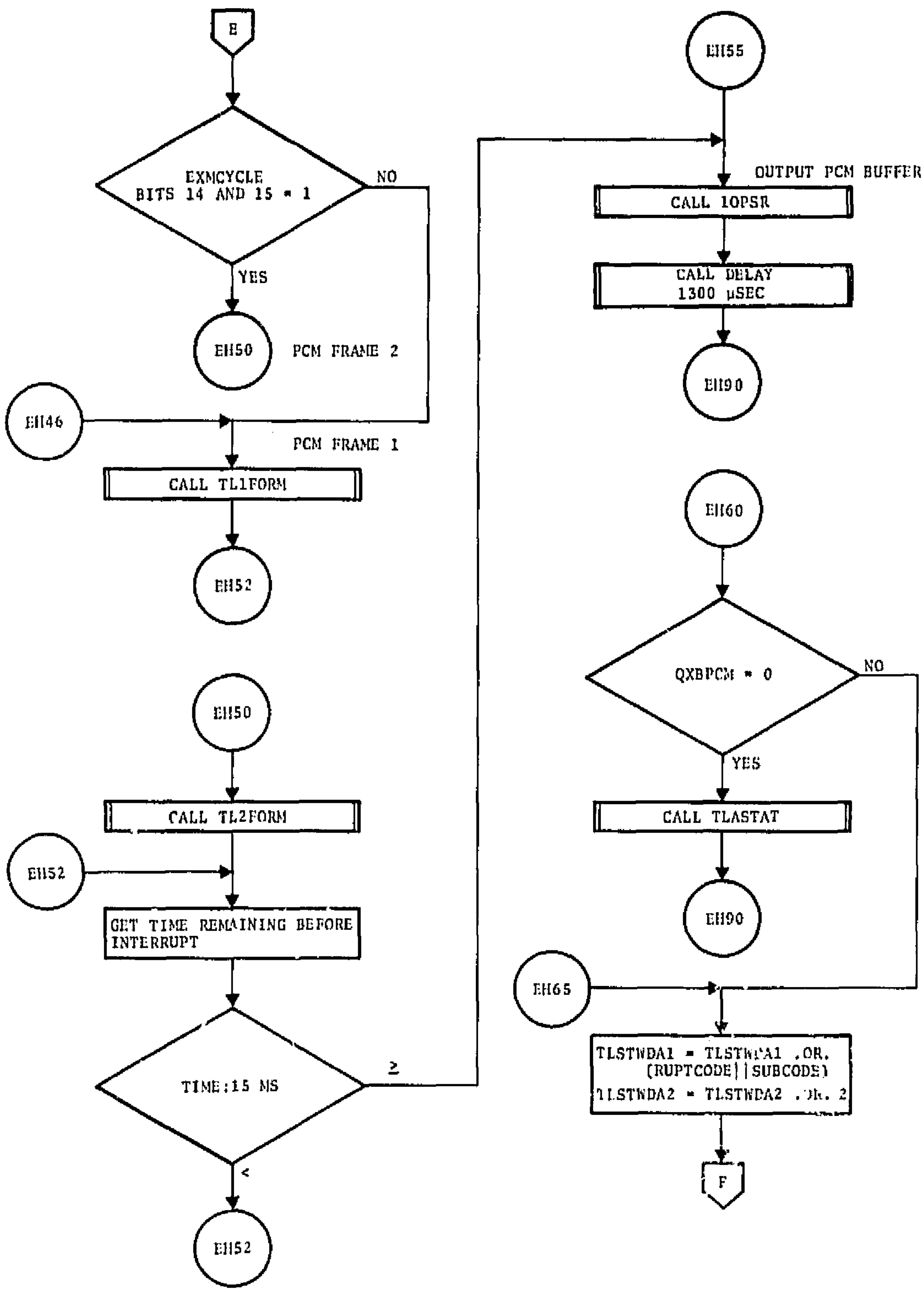

A
IDC:57
YES
NO
IIDVC = IDC
EHRCODE = 12
EHSCODE = 0
CLEAR PENDING INTERRUPT
EHMAIN
CALL RIR
READ IOPS INT REG
IOPIR = IOPS INTR'PT REGISTER
INTERRUPT INTERVAL TIMER
YES
EXRUPT
NO
EHSBITE
EXRUPT
RESET PENDING TIMER AND NOVA INTERRUPTS
EXROO
ENABLE INTERRUPTS
SET DISCRETE OUT BIT 1 OFF EXECUTION
EXJPT = A(EXJOB)
EXMCY = EXMCY + 1
QXBFC = 0
QXBPCM = 0
QTBRSTRT = 0
NO (RESTART)
B
YES
EXRSTRT
HTPHASE = 0
YES
EXLOOP
NO
HTPHASE = HTPHASE - 1
HTPHASE = 0
NO
EXLOOP
YES
C

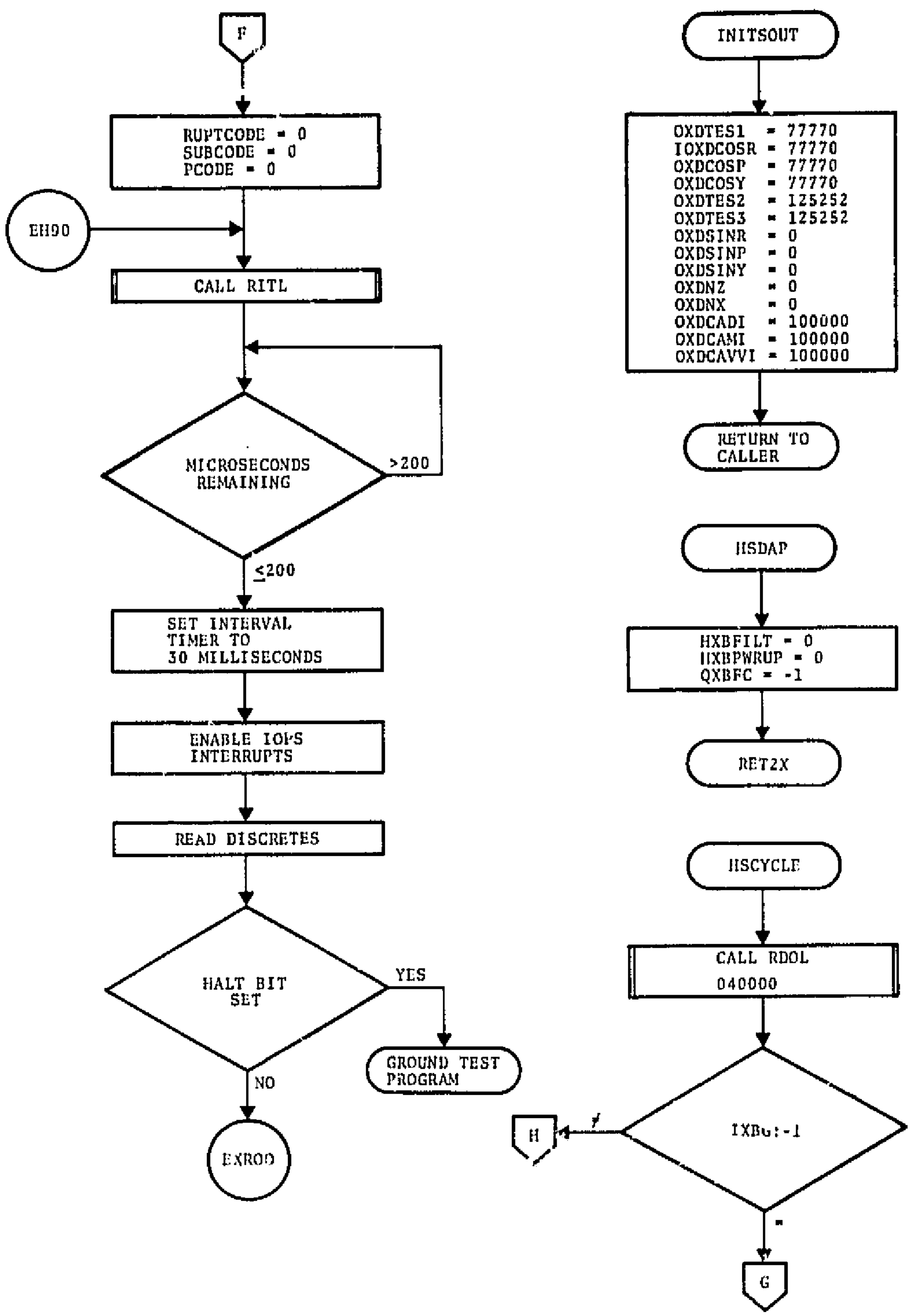

B
QTBRSTRT = 0
HTPHASE = 4
EXLOOP
C
QXBRSTRT = 0
QTBRSCTR = 0
EXLOOP
• LOAD "NEXT JOB" ADR VIA EXJPT
• GET NUMBER OF CONDITIONS TO BE MET
CONDITIONS:0
≠
=
ADJUST EXJPT FOR NEXT PASS
TRANSFER TO "NEXT JOB"
SAVE NUMBER OF CONDITIONS
EXCK5
GET FIRST CONDITIONAL PAIR
CONDITIONAL:3
>
≤
CLOCK CONDITIONAL
EXCK2
GET PARAMETER VALUE
PARAMETER: CONDIT.
≠
=
THIS PAIR IS OK - TEST FOR ADDITIONAL CONDITIONS
EXCK3
EXPAS


EXCK2
GET EXMCYCLE VALUE "ANDED WITH" CONDITION MASK
VALUE:CONDIT
=
≠
EXPAS
EXCK3
ADJUST EXJPT PAST LAST PAIR
#COND = #COND - 1
#COND:0
=
≠
TEST NEXT CONDITIONAL PAIR
EXCK5
ADJUST EXJPT TO NEXT PASS ENTRY
"NEXT JOB"
EXPAS
• MOVE EXJPT PAST CURRENT PAIR
• #COND = #COND - 1
#COND:0
≠
=
EXPAS
EXLOOP

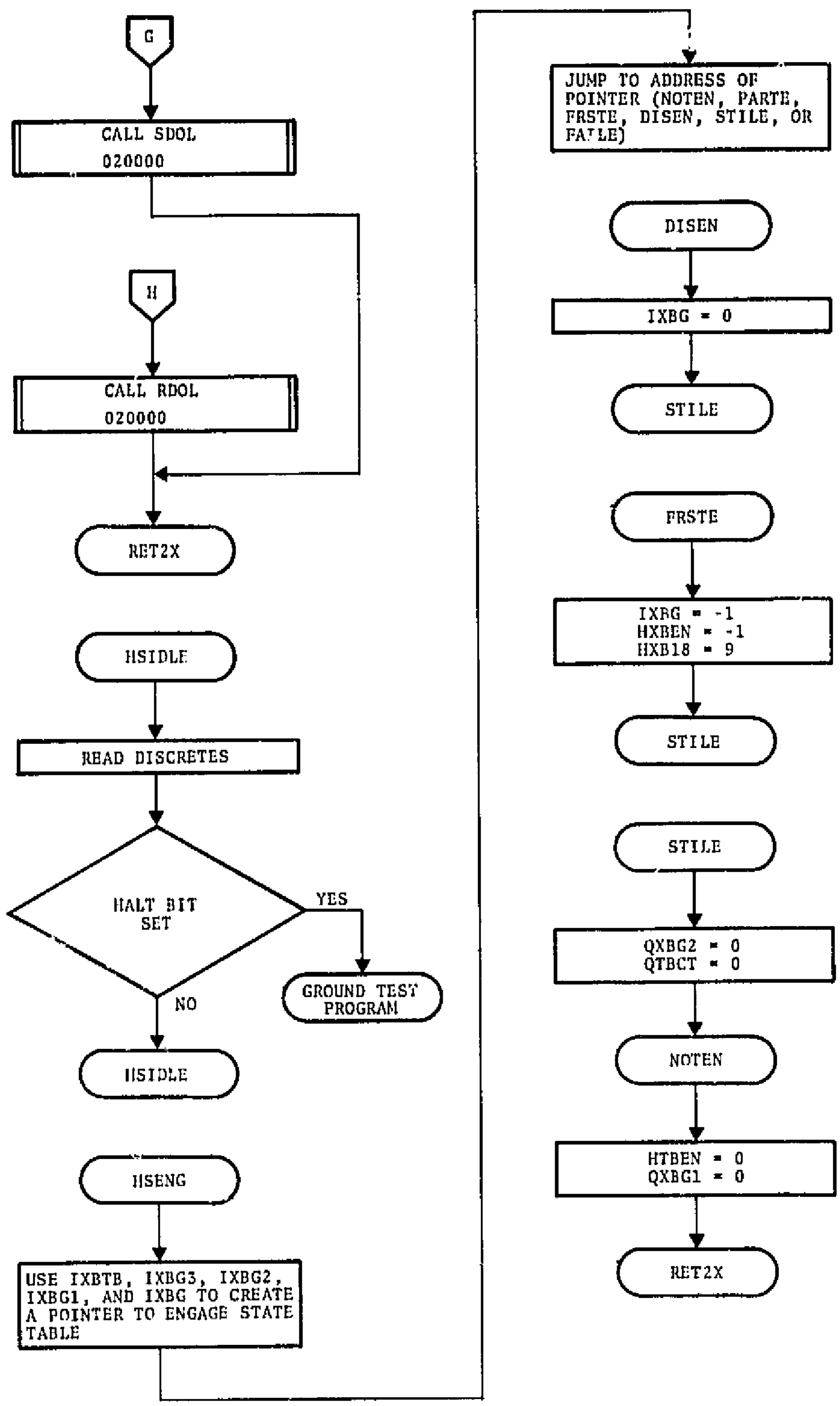

EHSBITE
EHRCODE = 6
IOPIR BIT 1:1
NO
YES
IOBIC = IOBIC + 1
EHSCODE = 10
EHMAIN
IOPIR BIT 3:1
NO
YES
BBHIC = BBHIC + 1
EHSCODE = 13
EHMAIN
IOPIR BIT 7:1
YES
NO
IAEIC = IAEIC + 1
EHSCODE = 14
EHMAIN
UIIR = IOPIR
UIRIC = UIRIC + 1
EHSCODE = 0
EHMAIN
EHMAIN
CALL INTM
CALL ETX
BCE'S 1, 2, 5, AND 8
CALL ERX
BCE 5
CALL INITSOUT
QXBTEST = 0
FDSTDONE = 0
FXBWOWCT = 0
QXBCWINT = -1
QTBRSTRT = -1
QXBRSCTR = QXBRSCTR + 1
D

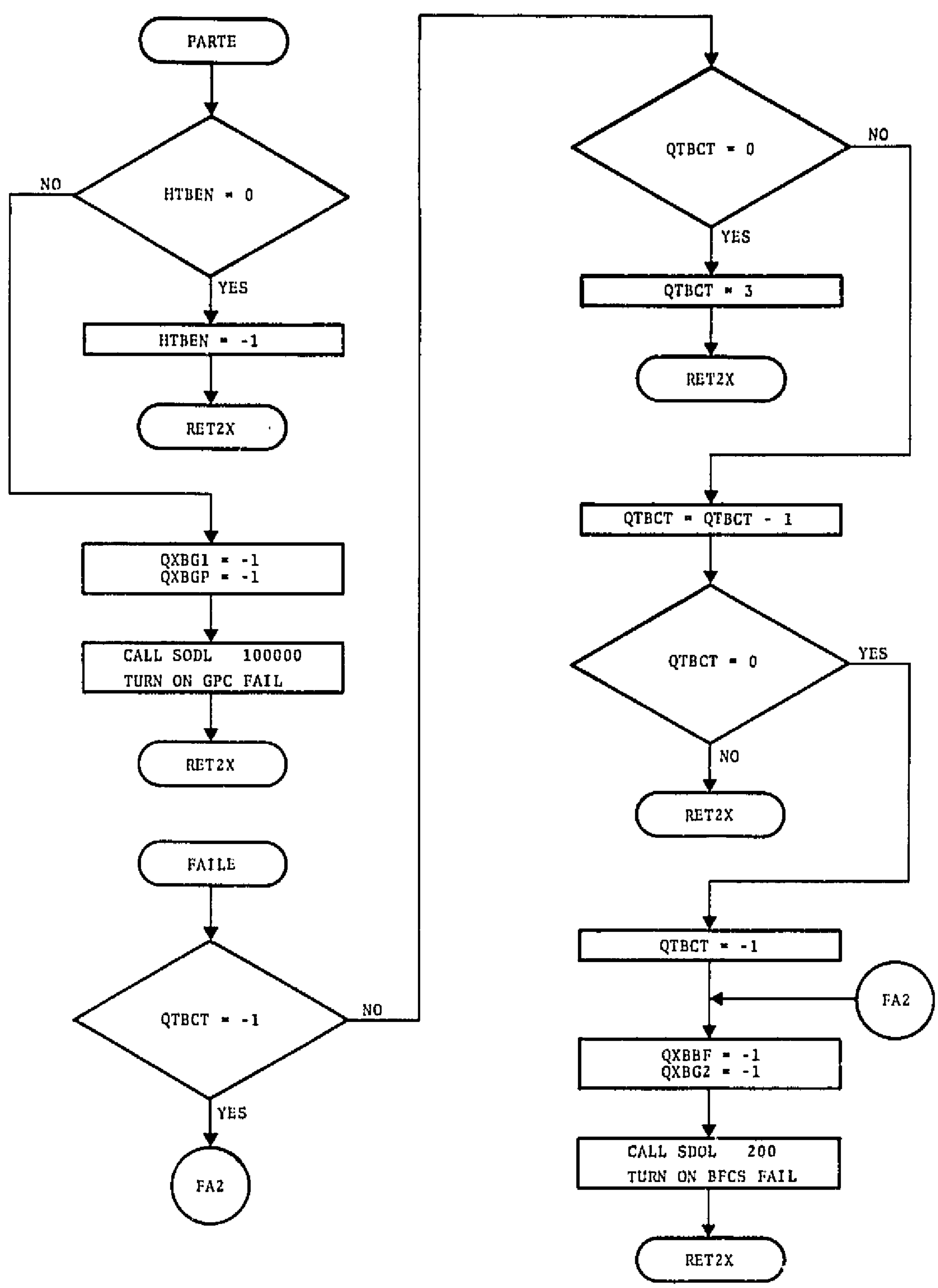

D
QTBRSCTR:0
≠
=
HXBFILT = -1
QTBRSCTR:2
=
≠
EH30
QXBBFCS = -1
QXBGPC = -1
QXBRSTRT = -1
CALL SDDL
"LIGHT THE BFCS AND GPC FAIL LAMP"
EH40
EH30
QTBRSCTR = QTBRSCTR + 1
EH10
EXMCYCLE BIT 15 = 0
NO
YES
EH60
NOT A PCM CYCLE
QXBPCM = 0
YES
NO
PCODE = -1
EH90
EH45
CALL TLBSTAT
E

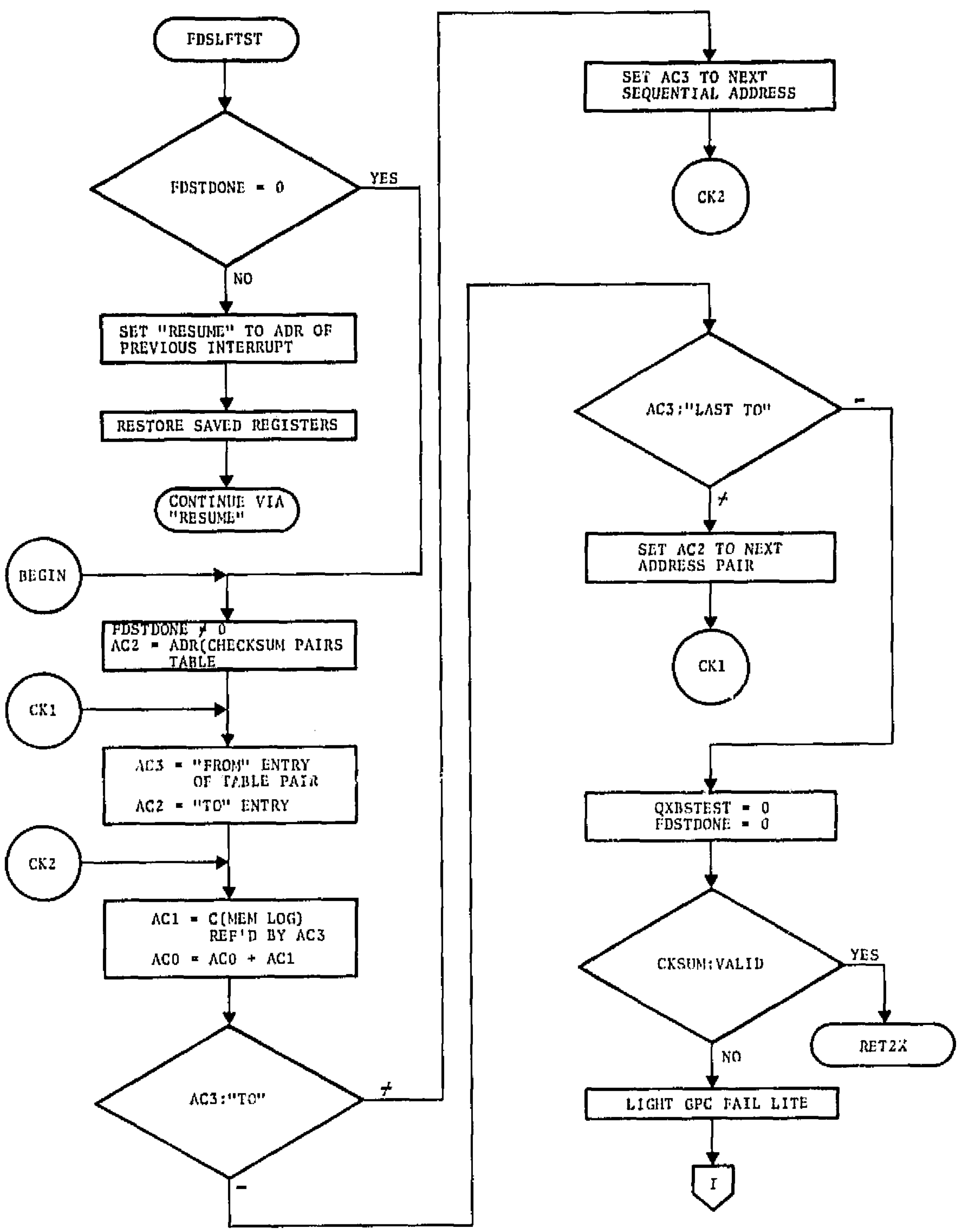

E
EXMCYCLE BITS 14 AND 15 = 1
NO
YES
EH50
PCM FRAME 2
EH46
PCM FRAME 1
CALL TL1FORM
EH52
EH50
CALL TL2FORM
EH52
GET TIME REMAINING BEFORE INTERRUPT
TIME:15 MS
≥
<
EH52
EH55
OUTPUT PCM BUFFER
CALL IOPSR
CALL DELAY 1300 µSEC
EH90
EH60
QXBPCM = 0
NO
YES
CALL TLASTAT
EH90
EH65
TLSTWDA1 = TLSTWDA1 .OR. (RUPTCODE||SUBCODE)
TLSTWDA2 = TLSTWDA2 .OR. 2
F



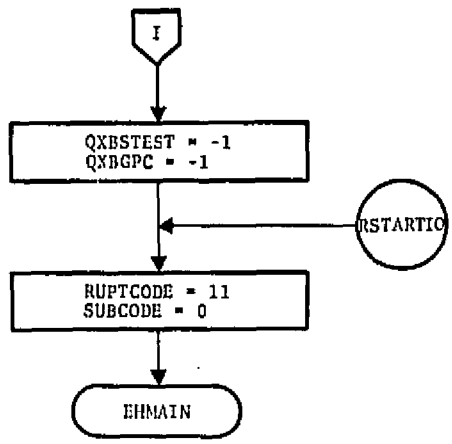
F
RUPTCODE = 0
SUBCODE = 0
PCODE = 0
EH90
CALL RITL
MICROSECONDS REMAINING
>200
≤200
SET INTERVAL TIMER TO 30 MILLISECONDS
ENABLE IOPS INTERRUPTS
READ DISCRETES
HALT BIT SET
YES
GROUND TEST PROGRAM
NO
EXR00
INITSOUT
OXDTES1 = 77770
IOXDCOSR = 77770
OXDCOSP = 77770
OXDCOSY = 77770
OXDTES2 = 125252
OXDTES3 = 125252
OXDSINR = 0
OXDSINP = 0
OXDSINY = 0
OXDNZ = 0
OXDNX = 0
OXDCADI = 100000
OXDCAMI = 100000
OXDCAVVI = 100000
RETURN TO CALLER
HSDAP
HXBFILT = 0
HXBPWRUP = 0
QXBFC = -1
RET2X
HSCYCLE
CALL RDOL
040000
IXBG:-1
≠
H
=
G


G
CALL SDOL
020000
H
CALL RDOL
020000
RET2X
HSIDLE
READ DISCRETES
HALT BIT SET
YES
NO
GROUND TEST PROGRAM
HSIDLE
HSENG
USE IXBTB, IXBG3, IXBG2, IXBG1, AND IXBG TO CREATE A POINTER TO ENGAGE STATE TABLE
JUMP TO ADDRESS OF POINTER (NOTEN, PARTE, FRSTE, DISEN, STILE, OR FATLE)
DISEN
IXBG = 0
STILE
FRSTE
IXBG = -1
HXBEN = -1
HXB18 = 9
STILE
STILE
QXBG2 = 0
QTBCT = 0
NOTEN
HTBEN = 0
QXBG1 = 0
RET2X


PARTE
HTBEN = 0
NO
YES
HTBEN = -1
RET2X
QXBG1 = -1
QXBGP = -1
CALL SODL 100000
TURN ON GPC FAIL
RET2X
FAILE
QTBCT = -1
NO
YES
FA2
QTBCT = 0
NO
YES
QTBCT = 3
RET2X
QTBCT = QTBCT - 1
QTBCT = 0
YES
NO
RET2X
QTBCT = -1
FA2
QXBBF = -1
QXBG2 = -1
CALL SDOL 200
TURN ON BFCS FAIL
RET2X



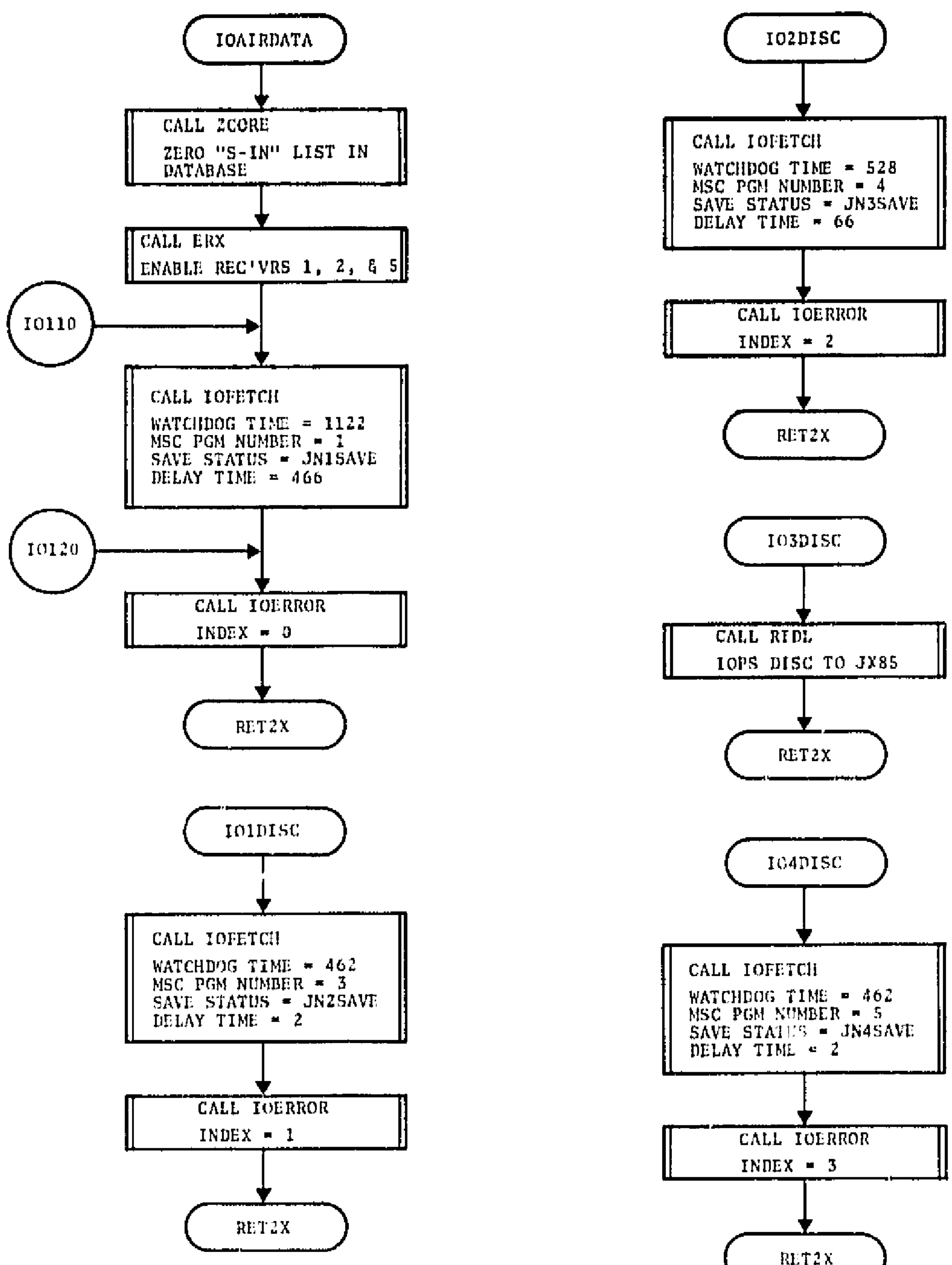
FDSLFTST
FDSTDONE = 0
YES
NO
SET "RESUME" TO ADR OF PREVIOUS INTERRUPT
RESTORE SAVED REGISTERS
CONTINUE VIA "RESUME"
BEGIN
FDSTDONE ≠ 0
AC2 = ADR(CHECKSUM PAIRS TABLE
CK1
AC3 = "FROM" ENTRY OF TABLE PAIR
AC2 = "TO" ENTRY
CK2
AC1 = C(MEM LOG) REF'D BY AC3
AC0 = AC0 + AC1
AC3:"TO"
≠
=
SET AC3 TO NEXT SEQUENTIAL ADDRESS
CK2
AC3:"LAST TO"
=
≠
SET AC2 TO NEXT ADDRESS PAIR
CK1
QXBSTEST = 0
FDSTDONE = 0
CKSUM:VALID
YES
RET2X
NO
LIGHT GPC FAIL LITE
I

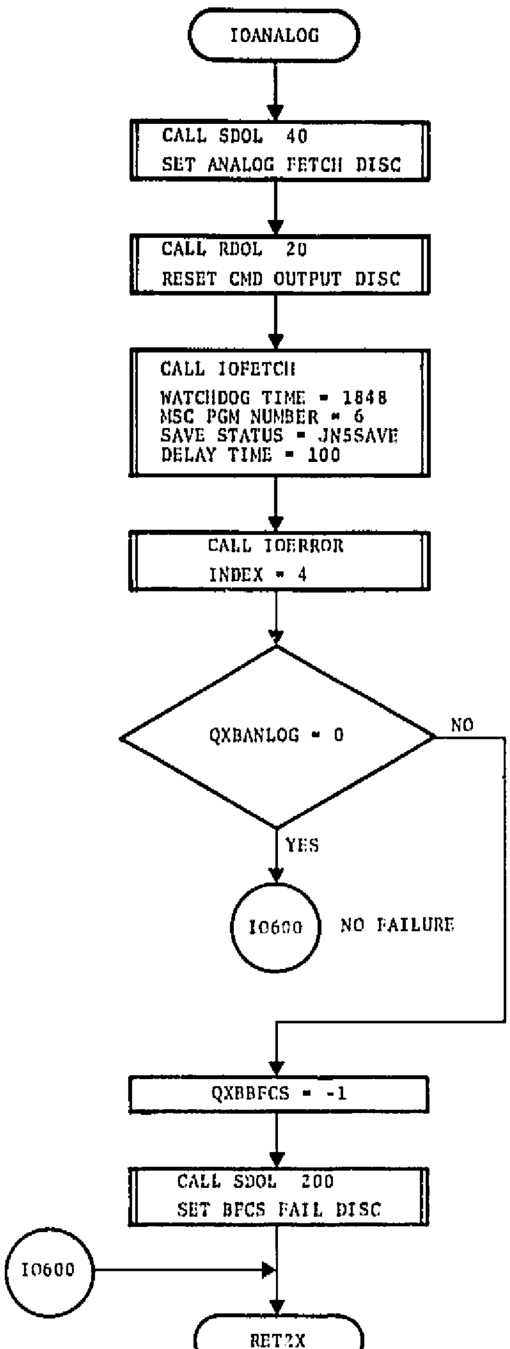

I
QXBSTEST = -1
QXBGPC = -1
RSTARTIO
RUPTCODE = 11
SUBCODE = 0
EHMAIN

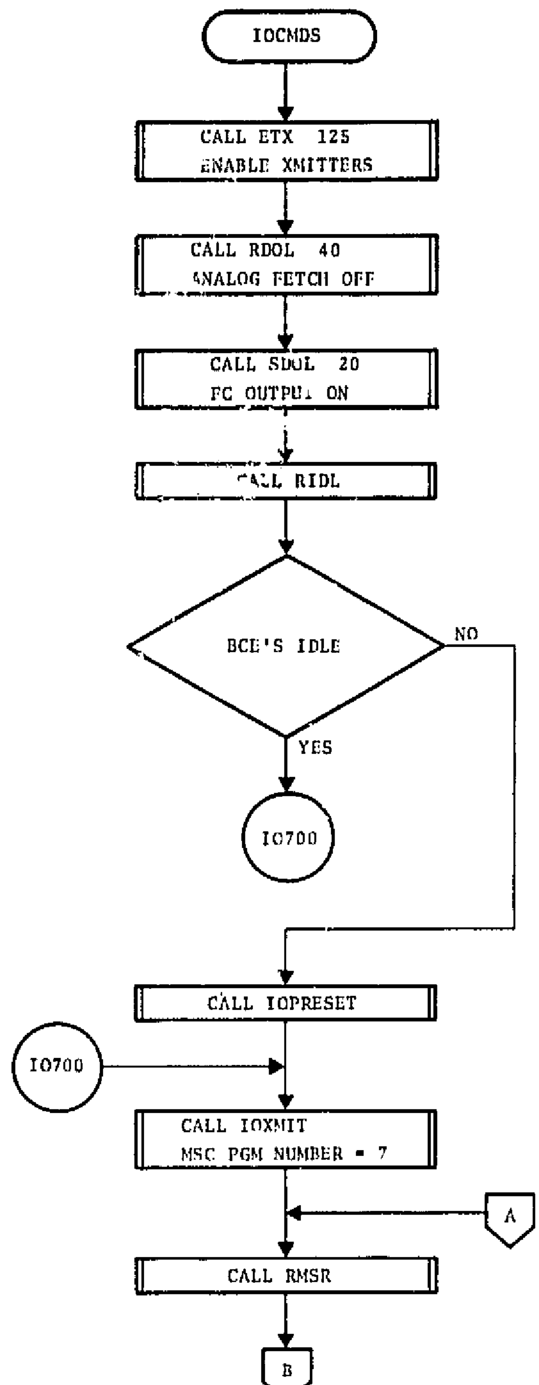
APPENDIX C

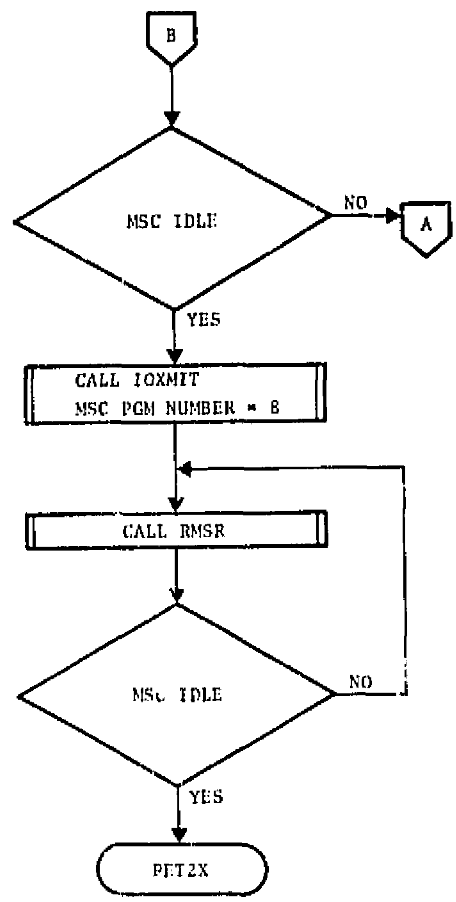
INPUT/OUTPUT JOB FLOW CHARTS

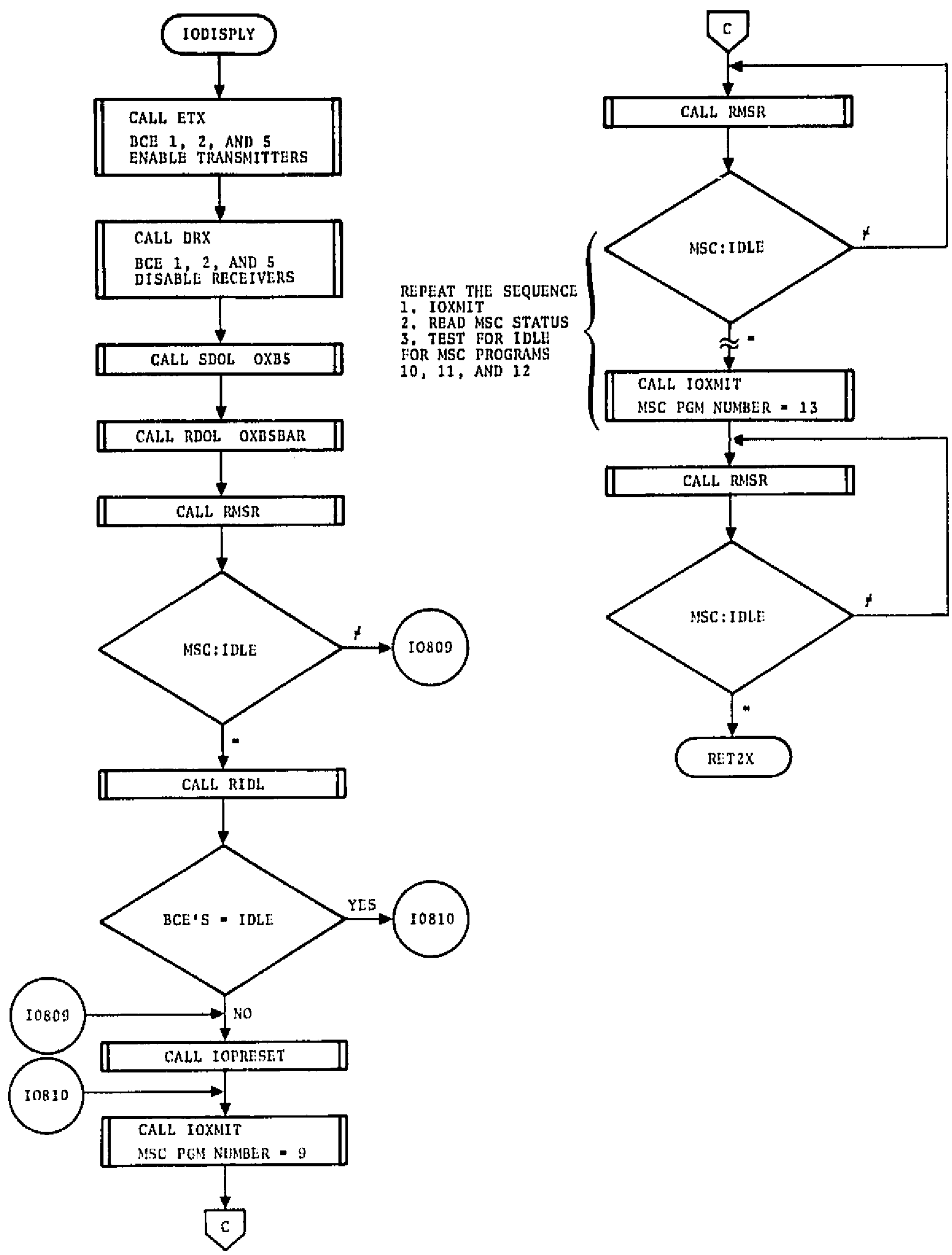
APPENDIX C
INPUT/OUTPUT JOB FLOW CHARTS

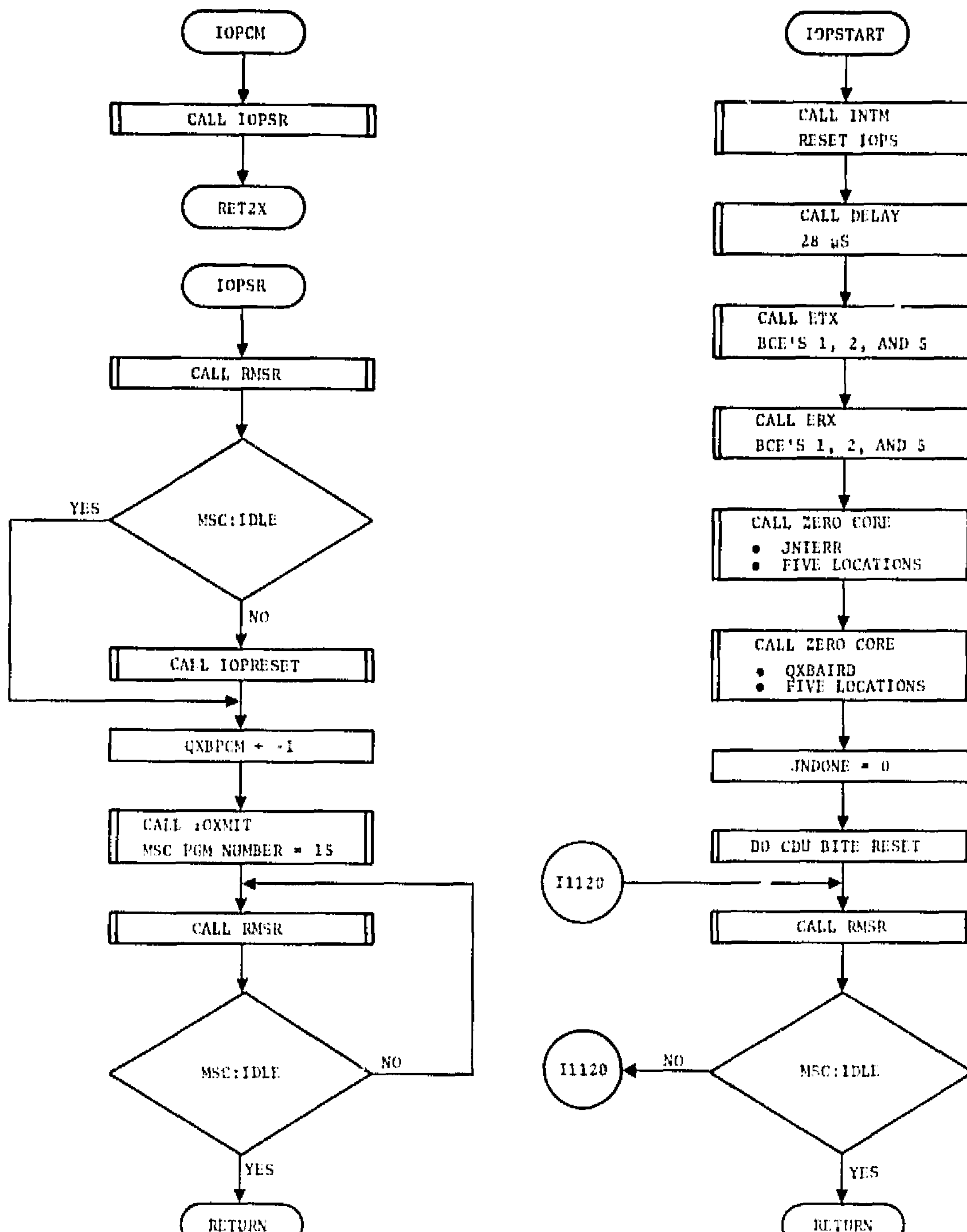


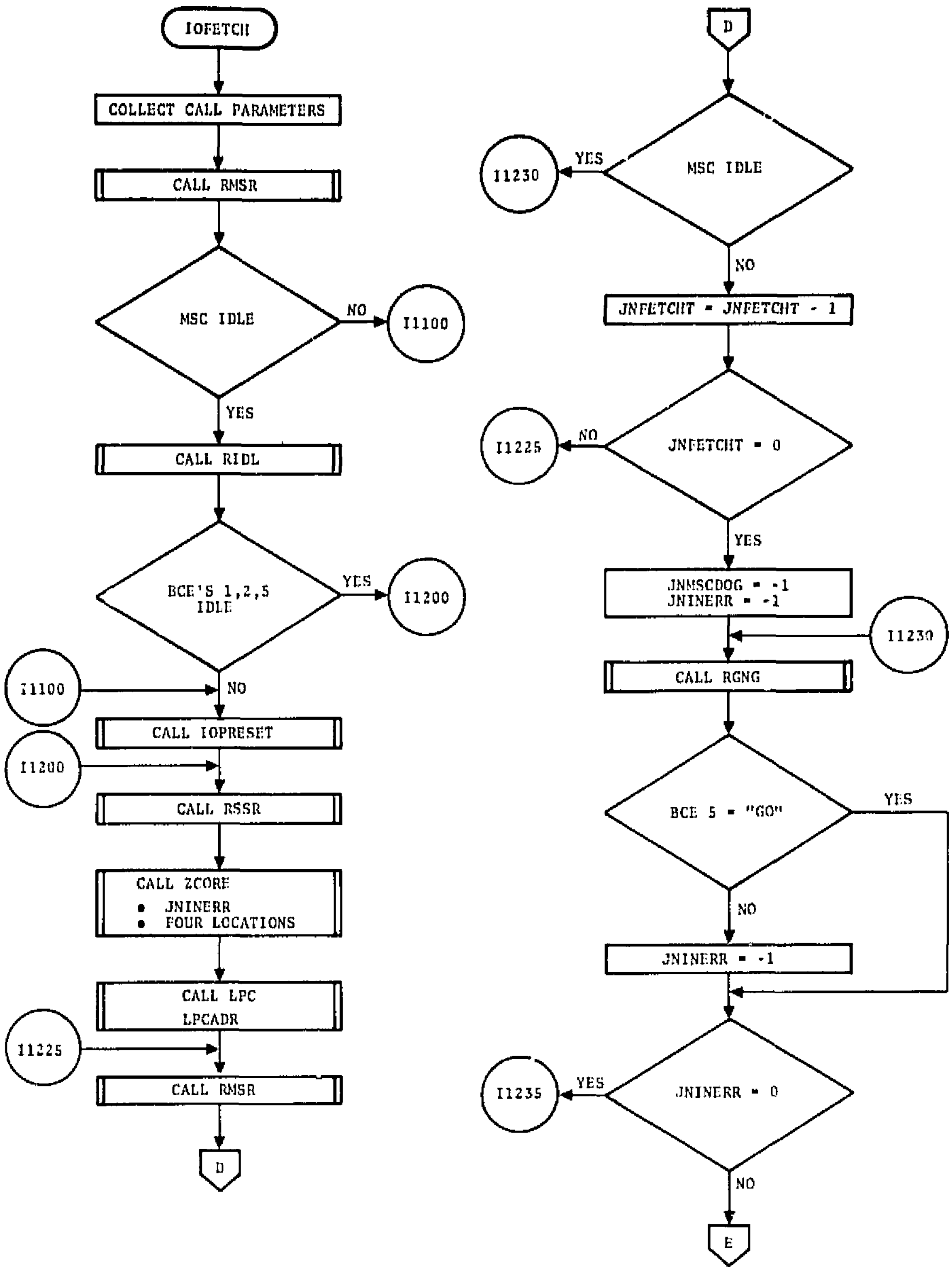

IOANALOG
CALL SDOL 40
SET ANALOG FETCH DISC
CALL RDOL 20
RESET CMD OUTPUT DISC
CALL IOFETCH
WATCHDOG TIME = 1848
MSC PGM NUMBER = 6
SAVE STATUS = JN5SAVE
DELAY TIME = 100
CALL IOERROR
INDEX = 4
QXBANLOG = 0
NO
YES
IO600
NO FAILURE
QXBBFCS = -1
CALL SDOL 200
SET BFCS FAIL DISC
IO600
RET2X

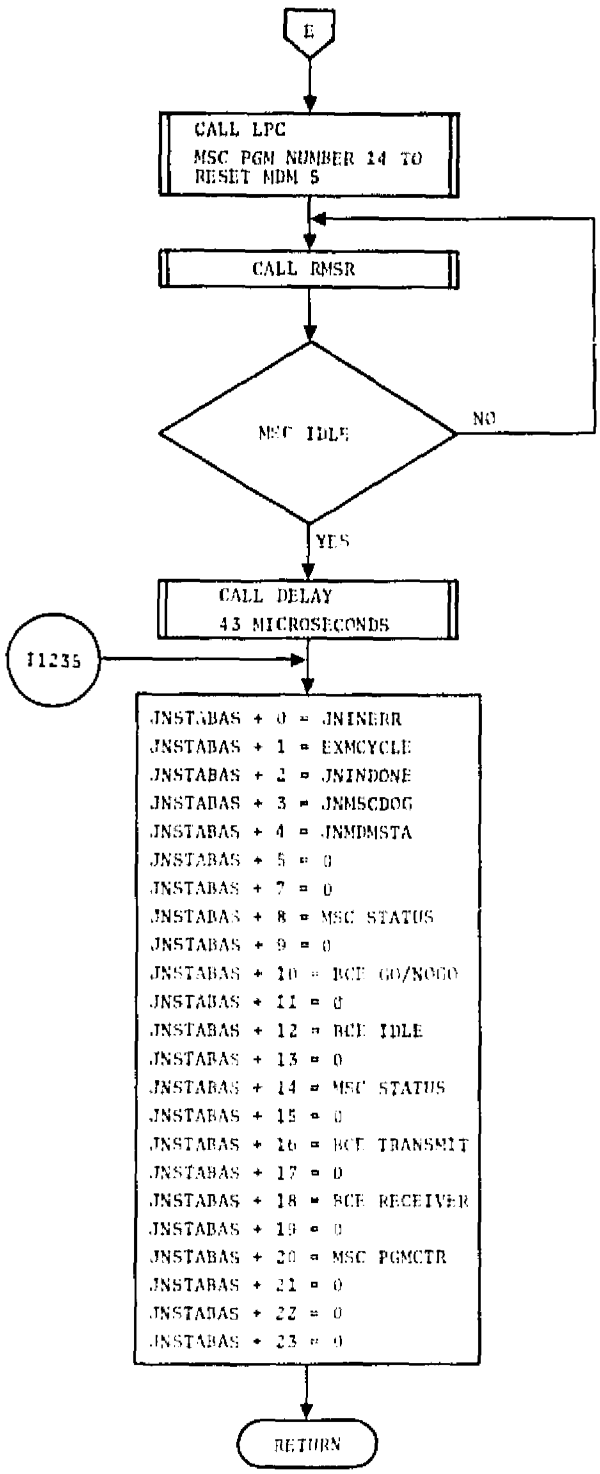

IOCMDS
CALL ETX 125
ENABLE XMITTERS
CALL RDOL 40
ANALOG FETCH OFF
CALL SDOL 20
FC OUTPUT ON
CALL RIDL
BCE'S IDLE
NO
YES
IO700
CALL IOPRESET
IO700
CALL IOXMIT
MSC PGM NUMBER = 7
A
CALL RMSR
B

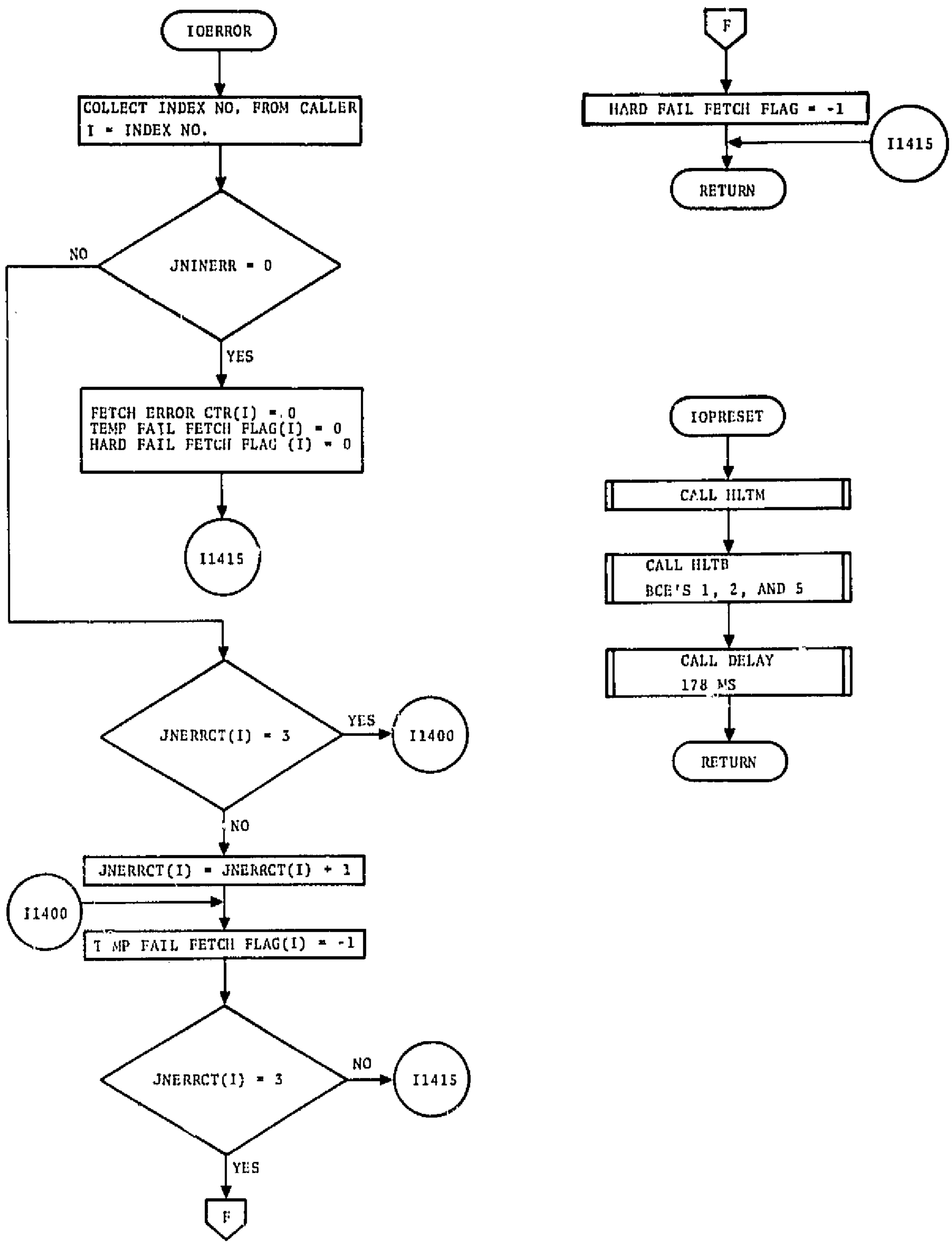

B
MSC IDLE
NO
A
YES
CALL IOXMIT
MSC PGM NUMBER = 8
CALL RMSR
MSC IDLE
NO
YES
PET2X

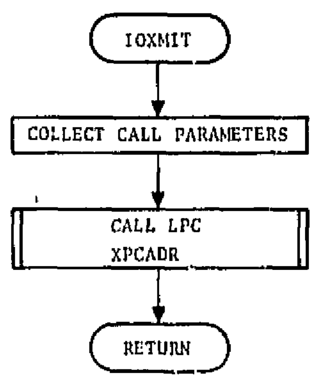

IODISPLY
CALL ETX
BCE 1, 2, AND 5
ENABLE TRANSMITTERS
CALL DRX
BCE 1, 2, AND 5
DISABLE RECEIVERS
CALL SDOL OXB5
CALL RDOL OXB5BAR
CALL RMSR
MSC:IDLE
≠
=
IO809
CALL RIDL
BCE'S = IDLE
YES
NO
IO810
IO809
CALL IOPRESET
IO810
CALL IOXMIT
MSC PGM NUMBER = 9
C
C
CALL RMSR
MSC:IDLE
≠
=
REPEAT THE SEQUENCE
1. IOXMIT
2. READ MSC STATUS
3. TEST FOR IDLE
FOR MSC PROGRAMS
10, 11, AND 12
CALL IOXMIT
MSC PGM NUMBER = 13
CALL RMSR
MSC:IDLE
≠
=
RET2X

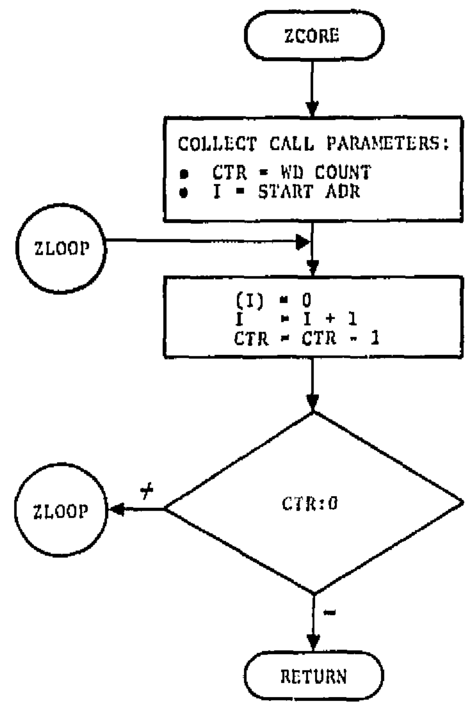

IOPCM
CALL IOPSR
RET2X
IOPSR
CALL RMSR
MSC:IDLE
YES
NO
CALL IOPRESET
QXBPCM ← -1
CALL IOXMIT
MSC PGM NUMBER = 15
CALL RMSR
MSC:IDLE
NO
YES
RETURN
IOPSTART
CALL INTM
RESET IOPS
CALL DELAY
28 μS
CALL ETX
BCE'S 1, 2, AND 5
CALL ERX
BCE'S 1, 2, AND 5
CALL ZERO CORE
• JNIERR
• FIVE LOCATIONS
CALL ZERO CORE
• QXBAIRD
• FIVE LOCATIONS
JNDONE = 0
DO CDU BITE RESET
I1120
CALL RMSR
MSC:IDLE
NO
I1120
YES
RETURN

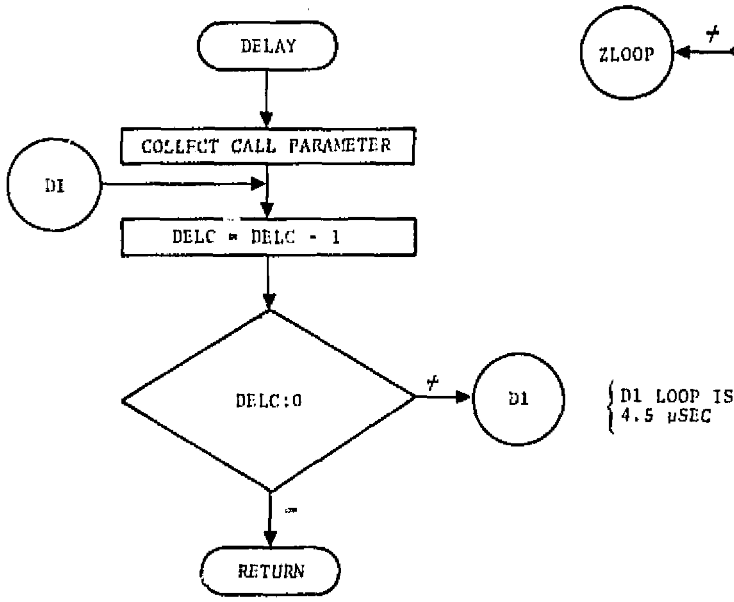

IOFETCH
COLLECT CALL PARAMETERS
CALL RMSR
MSC IDLE
NO
I1100
YES
CALL RIDL
BCE'S 1,2,5 IDLE
YES
I1200
NO
I1100
CALL IOPRESET
I1200
CALL RSSR
CALL ZCORE
• JNINERR
• FOUR LOCATIONS
CALL LPC
LPCADR
I1225
CALL RMSR
D
D
YES
I1230
MSC IDLE
NO
JNFETCHT = JNFETCHT - 1
I1225
NO
JNFETCHT = 0
YES
JNMSCDOG = -1
JNINERR = -1
I1230
CALL RGNG
BCE 5 = "GO"
YES
NO
JNINERR = -1
I1235
YES
JNINERR = 0
NO
E


E
CALL LPC
MSC PGM NUMBER 14 TO
RESET MDM 5
CALL RMSR
MSC IDLE
NO
YES
CALL DELAY
43 MICROSECONDS
I1235
JNSTABAS + 0 = JNINERR
JNSTABAS + 1 = EXMCYCLE
JNSTABAS + 2 = JNINDONE
JNSTABAS + 3 = JNMSCDOG
JNSTABAS + 4 = JNMDMSTA
JNSTABAS + 5 = 0
JNSTABAS + 7 = 0
JNSTABAS + 8 = MSC STATUS
JNSTABAS + 9 = 0
JNSTABAS + 10 = BCE GO/NOGO
JNSTABAS + 11 = 0
JNSTABAS + 12 = BCE IDLE
JNSTABAS + 13 = 0
JNSTABAS + 14 = MSC STATUS
JNSTABAS + 15 = 0
JNSTABAS + 16 = BCE TRANSMIT
JNSTABAS + 17 = 0
JNSTABAS + 18 = BCE RECEIVER
JNSTABAS + 19 = 0
JNSTABAS + 20 = MSC PGMCTR
JNSTABAS + 21 = 0
JNSTABAS + 22 = 0
JNSTABAS + 23 = 0
RETURN


IOERROR
COLLECT INDEX NO. FROM CALLER
I = INDEX NO.
JNINERR = 0
NO
YES
FETCH ERROR CTR(I) = 0
TEMP FAIL FETCH FLAG(I) = 0
HARD FAIL FETCH FLAG (I) = 0
I1415
JNERRCT(I) = 3
YES
I1400
NO
JNERRCT(I) = JNERRCT(I) + 1
I1400
TEMP FAIL FETCH FLAG(I) = -1
JNERRCT(I) = 3
NO
I1415
YES
F
F
HARD FAIL FETCH FLAG = -1
I1415
RETURN
IOPRESET
CALL HLTM
CALL HLTB
BCE'S 1, 2, AND 5
CALL DELAY
178 MS
RETURN


IOXMIT
COLLECT CALL PARAMETERS
CALL LPC
XPCADR
RETURN


ZCORE
COLLECT CALL PARAMETERS:
• CTR = WD COUNT
• I = START ADR
ZLOOP
(I) = 0
I = I + 1
CTR = CTR - 1
CTR:0
ZLOOP
≠
=
RETURN


DELAY
COLLECT CALL PARAMETER
D1
DELC = DELC - 1
DELC:0
≠
D1
=
RETURN
D1 LOOP IS
4.5 μSEC

APPENDIX D

UNIT TEST RESULTS

APPENDIX D

UNIT TEST RESULTS

CONTENTS

Section | Page

Section | Page

1. ACTUATOR INPUT PROCESSING MODULE (FC01AIPM) TESTING

1.1 COMBINED LEFT ELEVON ACTUATOR INPUT COMPUTATION (ASPM01) SUBMODULE TESTING

The ASPM01 was tested for the single set of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.1.1.1

	Data Set
Inputs:	
IXSOEL =	-10.0
IXSIEL =	-10.0
Constants:	
AC01 =	+0.5
Expected Output:	
ANSEL =	-10.0
Actual Output:	
ANSEL =	-10.0

1.2 COMBINED RIGHT ELEVON ACTUATOR INPUT COMPUTATION (ASPM02) SUBMODULE TESTING

The ASPM02 was tested for the single set of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.2.2.1

	Data Set
Inputs:	
IXSOER	+2.0
IXSIEL =	+2.0
Constants:	
AC02 =	+0.5
Expected Output:	
ANSER =	+2.0
Actual Output:	
ANSER =	+2.0

1.3 SENSED ELEVATOR POSITION COMPUTATION (ASPM03) SUBMODULE TESTING

The ASPM03 was tested for the single set of input values given. The actual results compared with the expected results as listed in the specifications.

Test # 2.1.3.1

	Data Set
Inputs:	
ANSEL =	-10.0
ANSER =	+2.0159
Constants:	
AC03 =	+0.5
Expected Output:	
AXSEVT =	+4.0
Actual Output:	
AXSEVT =	+4.0

1.4 SENSED AILERON POSITION COMPUTATION (ASPM04) SUBMODULE TESTING

The ASPM04 was tested for the single set of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.1.4.1

	Data Set
Inputs:	
ANSEL =	-10.0
ANSER =	+2.0
Constants:	
AC04 =	+0.5
Expected Output:	
ANSAIL =	-6.0
Actual Output:	
ANSAIL =	-6.0

1.5 AILERON POSITION DISPLAY LIMITER (ASPM17) SUBMODULE TESTING

The ASPM17 was tested for the single set of input values given. The actual results compared with the expected results as listed in the specification.

Test #2.1.5.1

	Data Set
Inputs:	
ANSAIL =	-6.0
Constants:	
AL17U	+5.0
AL17L	-5.0
Expected Output:	
AXSAIL =	-5.0
Actual Output:	
AXSAIL =	-5.0

2. ROLL AXIS CONTROL MODULE (FC02RACM) TESTING

2.1 LEFT RHC ROLL DEADBANDING (RACM01) SUBMODULE TESTING

The RACM01 was tested for the single set of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.2.1.1

	Data Set
Inputs:	
IXCRCL =	-5.0
Constants:	
RB01DBL =	+1.15
Expected Output:	
RX01RCL =	-3.85
Actual Output:	
RX01RCL =	-3.85

2.2 RIGHT RHC ROLL DEADBANDING (RACM02) SUBMODULE TESTING

The RACM02 was tested for the single set of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.2.2.1

	Data Set
Inputs:	
IXCRCR =	+5.0
Constants:	
RB02DBR =	+1.15
Expected Output:	
RX02RCR =	+3.849999
Actual Output:	
RX02RCR =	+3.85

2.3 COMBINED ROLL COMMAND SHAPING AND LIMITING (RACM03) SUBMODULE THEORY

The RACM03 was tested for the two sets of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.2.3.1

	Data Set 1	Data Set 2
Inputs:		
RX01RCL =	+20.0	-20.0
RX02RCR =	+80.0	+10.0
Constants:		
RC03PS1 =	+0.08	+0.08
RC03PS2 =	+0.0636	+0.636
RL03U =	+23.0	+23.0
RL03L =	+23.0	+23.0
Expected Output:		
RX03RC =	23.0	-7.16
Actual Output:		
RX03RC =	23.0	-7.159994

2.4 ROLL COMMAND FILTER (RACM04) SUBMODULE TESTING

The RACM04 was tested for the two sets of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.2.4.1

	Data Set 1	Data Set 2
Inputs:		
HXBFILT =	-0001	0000
RX03RC =	+4.0	+5.0
RN04FI =	+4.0	+5.0
RF04NI =	+2.0	+2.0
Constants:		
RK04IO =	+0.047619048	+0.047619048
RG04 =	+1.0	+1.0
RG04DMP =	+1.14	+1.14

Test #2.2.5.1 (continued)

	Data Set 1	Data Set 2
Expected Outputs:		
RX04RCF =	+4.559999	+2.551428
RN04FO =	+4.0	+2.238094
RN04FI =	+4.0	+5.0
Actual Outputs:		
RX04RCF =	+4.559997	+2.551428
RN04FO =	+4.0	+2.238094
RN04FI =	+4.0	+5.0

2.5 ROLL RATE GYRO NOISE FILTER (RACM05) SUBMODULE TESTING

The RACM05 was tested for the two sets of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.2.5.1

	Data Set 1	Data Set 2
Inputs:		
HXBFILT =	-0001	0000
IXSRGR =	+6.0	+6.0
RF05N1 =	+4.0	+4.0
Constants:		
RG05 =	+1.0	+1.0
RK05IO =	+0.25	+0.25
Expected Outputs:		
RXRGRF =	+6.0	+5.50
RN05FO =	+6.0	+5.50
RN05FI =	+6.0	+5.0
Actual Outputs:		
RXRGRF =	+6.0	+5.50
RN05FO =	+6.0	+5.50
FN05FI =	+6.0	+6.0

2.6 ROLL RATE COMMAND ERROR COMPUTATION (RACM06) SUBMODULE TESTING

The RACM06 was tested for the three sets of input values given. The actual results compared with the expected results as listed in the specifications. This includes the intermediate values.

Test #2.2.6.1

	Data Set 1	Data Set 2	Data Set 3
Inputs:			
IXSQBAR =	+1.0	+0.00499999	+137.2
RX04RCF =	+2.0	+5.0	+0.0562
RXRGRF =	+4.0	+1.0	+11.34999
Constants:			
RL06QBAR =	+0.01	+0.01	+0.01
RC06KROL =	+85.0	+85.0	+85.0
RL06GU	+1.0	+1.0	+1.0
RL06GL =	+0.2	+0.2	+0.2
RG06PP =	+1.0	+1.0	+1.0
Expected Output:			
RX06ERR =	-2.0	+4.0	-6.996887
Actual Output:			
RX06ERR =	-2.0	+4.0	-6.996887
Expected Intermediate:			
RT06QBAR =	+1.0	+0.01	+137.2
RT06GDAC =	+1.0	+1.0	+0.6195335
Actual Intermediate:			
RT06QBAR =	+1.0	+0.01	+137.2
RT06GDAC =	+1.0	+1.0	+0.6195335

2.7 ROLL BENDING FILTER NO. 1 (RACM07) SUBMODULE TESTING

The RACM07 was tested for the two sets of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.2.7.1

	Data Set 1	Data Set 2
Inputs:		
HXBFILT =	-0001	0000
RX06ERR =	+2.0	+2.0
RN07FI =	+2.0	+2.0
RF07N1 =	+4.0	+4.0
Constants:		
RK08IO =	+0.090078	+0.090078
RG07 =	+1.0	+1.0
Expected Output:		
RN07FI =	+2.0	+2.0
RN07FO =	+2.0	+4.180155
Actual Output:		
RN07FI =	+2.0	+2.0
FN07FO =	+2.0	+4.180155

2.8 ROLL BENDING FILTER NO. 2 (RACM08) SUBMODULE TESTING

The RACM08 was tested for the two sets of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.2.8.1

	Data Set 1	Data Set 2
Inputs:		
HXBFILT =	-0001	0000
RN08FO =	+2.0	+2.0
RF08N1 =	+8.0	+8.0
Constants:		
RK08IO =	+0.987804	+0.987804
RG08 =	+1.0	+1.0

Test #2.2.8.1 (continued)

	Data Set 1	Data Set 2
Expected Outputs:		
RN08FI =	+2.0	+2.0
RN08FO =	+2.0	+9.975609
Actual Outputs:		
RN08FI =	+2.0	+2.0
RN08FO =	+2.0	+9.975609

2.9 MANUAL ROLL TRIM RECTANGULAR INTEGRATION (RACM09) SUBMODULE TESTING

The RACM09 was tested for the four sets of input values given. The actual results compared with the expected results as listed in the specifications. This includes the intermediate values.

Test #2.2.9.1

	Data Set 1	Data Set 2	Data Set 3	Data Set 4
Inputs:				
IXBENG =	-0001	-0001	-0001	0000
FXBRTP =	-0001	0000	0000	0000
FXBRTN =	-0001	-0001	0000	0000
RX09RT =	-2.0	-1.0	+12.0	+2.0
Constants:				
RG09RT =	+0.910	+0.910	+0.910	+0.910
RC09DT =	+0.020	+0.020	+0.020	+0.020
RL09U =	+10.0	+10.0	+10.0	+10.0
RL09L =	-10.0	-10.0	-10.0	-10.0
Expected Output:				
RX09RT =	-1.981799	-1.0182	-10.0	+2.0
Actual Output:				
RX09RT =	-1.981799	-1.0182	-10.0	+2.0
Expected Intermediate:				
RT09RTR =	+0.91	-0.91	0.0	0.0
Actual Intermediate:				
RT09RTR =	+0.91	-0.91	0.0	0.0

2.10 AILERON POSITION COMMAND COMPUTATION (RACM10) SUBMODULE TESTING

The RACM10 was tested for the single set of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.2.10.1

	Data Set
Inputs:	
RX09RT =	+3.0
RN09FO =	+2.0
Expected Output:	
RXCAIL =	+5.0
Actual Output:	
RXCAIL =	+5.0

3. YAW AXIS CONTROL MODULE (FC03YACM) TESTING

3.1 MANUAL YAW TRIM RECTANGULAR INTEGRATION (YACM01) SUBMODULE TESTING

Test No. 2.3.1.1

The YACM01 was tested for the eight sets of input values given. The actual results compared with the expected results and this included the value of the intermediate variables as listed in the specifications.

The test results are as follows:

	Data Set 1	Data Set 2	Data Set 3	Data Set 4
Inputs:				
IXBENG =	-0001	-0001	-0001	0000
FXBYP =	-0001	0000	0000	-0001
FXBYN =	0000	-0001	0000	-0001
YX01YT =	-2.0	-1.0	+23.0	-2.0
Constants:				
YG01YT =	+3.0	+3.0	+3.0	+3.0
YC01DT =	+0.020	+0.020	+0.020	+0.020
YL01U =	+22.5	+22.5	+22.5	+22.5
YL01L =	-22.5	-22.5	-22.5	-22.5
Expected Output:				
YX01YT =	-1.94	-1.06	+22.5	-2.0
Actual Outputs:				
YX01YT =	-1.94	-1.06	+22.5	-2.0
Actual Intermediate:				
YT01YTR =	+3.0	-3.0	0.0	0.0
Expected Intermediate:				
YT01YTR =	+3.0	-3.0	0.0	0.0

	Data Set 5	Data Set 6	Data Set 7	Data Set 8
Inputs:				
IXBENG =	-0001	0000	0000	0000
FXBYP =	-0001	0000	-0001	000[illegible]
FXBYN =	-0001	0000	0000	-0001
YX01YT =	-2.0	-2.0	-2.0	-2.0

Test No. 2.3.1.1 (continued)

Constants:	Data Set 5	Data Set 6	Data Set 7	Data Set 8
YG01YT =	+3.0	+3.0	+3.0	+3.0
YC01DT =	+0.020	+0.020	+0.02	+0.020
YL01U =	+22.5	+22.5	+22.5	+22.5
YL01L =	-22.5	-22.5	-22.5	-22.5
Expected Output:				
YX01YT =	-1.94	-2.0	-2.0	-2.0
Actual Output:				
YX01YT =	-1.94	-2.0	-2.0	-2.0
Actual Intermediate:				
YT01YTR =	+3.0	0.0	0.0	0.0
Expected Intermediate:				
YT01YTR =	+3.0	0.0	0.0	0.0

3.2 RUDDER PEDAL COMMAND SHAPING (YACM02) SUBMODULE TESTING

Test No. 2.3.2.1

The YACM02 was tested for the two sets of input values given. The actual results compared with the expected results and this included the value of the intermediate variables as listed in the specifications.

The test results are as follows:

	Data Set 1	Data Set 2
Inputs:		
IXCRU =	+20.0	+10.0
Constants:		
YB02DB =	+1.125	+1.125
YC02PS1 =	+0.9	+0.9
YC02PS2 =	+0.0194	+0.0194
YL02U =	+22.5	+22.5
YL02L =	-22.5	-22.5
Expected Output:		
YX02CRU =	+22.5	+9.515553

Test No. 2.3.2.1 (continued)

	Data Set 1	Data Set 2
Actual Output:		
YX02CRU =	+22.5	+9.515553
Expected Intermediates:		
YT02DRU =	+18.875	+8.875
YT02PRU =	+23.89904	+9.515553
Actual Intermediates:		
YT02DRU =	+18.875	+8.875
YT02PRU =	+23.89904	+9.515553

3.3 LATERAL G' LIMITER (YACM03) SUBMODULE TESTING

Test No. 2.3.3.1

The YACM03 was tested for the given set of input values. The actual results compared with the expected results and this included the value of the intermediate variables as listed in the specifications.

The test results are as follows:

	Data Set 1
Inputs:	
YX01YT =	+10.0
YX02CRU =	+20.0
Constants:	
YG03DRM =	+0.0664
YL03U =	+0.5
YL03L =	-0.5
Expected Output:	
YX03CNY =	+0.5
Actual Output:	
YX03CNY =	+0.5
Expected Intermediate:	
YT03NY =	+1.992
Actual Intermediate:	
YT03NY =	+1.991999

3.4 RADAR ALTITUDE NOISE FILTER (YACM04) SUBMODULE TESTING

Test 2.3.4.1

The YACM04 was tested for the two sets of input values given. The actual results compared with the expected results and this included the value of the intermediate variables as listed in the specifications.

The test results are as follows:

	Data Set 1	Data Set 2
Inputs:		
HXBFILT =	-0001	0000
FXSRAD =	+10.0	+10.0
YF04N1 =	+2.0	+2.0
Constants:		
YK04I0 =	+0.25	+0.25
YG04 =	+1.0	+1.0
Expected Outputs:		
YXSRADF =	+10.0	+4.5
YN04FO =	+10.0	+4.5
YN04FI =	+10.0	+10.0
Actual Outputs:		
YXSRADF =	+10.0	+4.5
YN04FO =	+10.0	+4.5
YN04FI =	+10.0	+10.0

3.5 LATERAL ACCELERATION FILTER (YACM06) SUBMODULE TESTING

Test 2.3.6.1

The YACM06 was tested for the two sets of input values given. The actual results compared with the expected results as listed in the specifications.

The test results are as follows:

	Data Set 1	Data Set 2
Inputs:		
HXBILT =	-0001	+0000
IXSNY =	+10.0	+10.0
YF06n1 =	+2.0	+2.0
YX03CNY =	+3.0	+3.0

Test 2.3.6.1 (continuned)

	Data Set 1	Data Set 2
Constants:		
YK06I0 =	+0.02380952	+0.02380952
YG06 =	+0.5	+0.5
Expected Outputs:		
YX06NYE =	+2.0	-0.7619047
YN06FO =	+5.0	+2.238094
YN06FI =	+10.0	+10.0
Actual Outputs:		
YX06NYE =	+2.0	-0.7619047
YN06FO =	+5.0	+2.238094
YN06FI =	+10.0	+10.0

3.7 LATERAL ACCELERATION COMMAND FILTER (YACM07) SUBMODULE TESTING

Test No. 2.3.7.1

The YACM07 was tested for the two sets of input values given. The actual results compared with the expected results as listed in the specifications.

The test results are as follows:

	Data Set 1	Data Set 2
Inputs:		
HXBFILT =	-0001	0000
YX06NYE =	+10.0	+10.0
YN07FI =	+10.0	+10.0
YF07N1 =	+2.0	+2.0
Constants:		
YK07I0 =	+0.2	+0.2
YG07 =	+1.0	+1.0
YG07RAY =	+20.0	+20.0
Expected Outputs:		
YX07NYEF =	+200.0	+80.0
YN07FO =	+10.0	+4.0
YN07FI =	+10.0	+10.0

Test No. 2.3.7.1 (continued)

	Data Set 1	Data Set 2
Actual Outputs:		
YX07NYEF =	+200.0	+79.999997
YN07FO =	+10.0	+3.999999
YN07FI =	+10.0	

3.7 YAW RATE GYRO NOISE FILTER (YACM08) SUBMODULE TESTING

Test 2.3.8.1

The YACM08 was tested for the two sets of input values given. The actual results compared with the expected results as listed in the specifications.

The test results are as follows:

	Data Set 1	Data Set 2
Inputs:		
HXFFILT =	-0001	0000
IXSRGY =	+4.0	+4.0
YF08N1 =	+5.0	+5.0
Constants:		
YK08IO =	+0.25	+0.25
YG08 =	+1.0	+1.0
Expected Outputs:		
YXRGYF =	+4.0	+6.0
YN08FO =	+4.0	+6.0
YN08FI =	+4.0	+4.0
Actual Outputs:		
YXRGYF =	+4.0	+6.0
YN08FO =	+4.0	+6.0
YN08FI =	+4.0	+4.0

3.8 ROLL/YAW RATE COMPENSATOR (YACM09) SUBMODULE TESTING

Test No. 2.3.9.1

The YACM09 was tested for the two sets of input values given. The actual results compared with the expected results as listed in the specifications.

Test 2.3.9.1 (continued)

The test results are as follows:

	Data Set 1	Data Set 2
Inputs:		
HXBFILT =	-0001	0000
IXSALPHA =	+2.0	+2.0
IXSRGR =	+5.0	+5.0
YF09N1 =	+3.0	+3.0
Constants:		
YC09DTR =	+0.01745200	+0.01745200
YK09IO =	+0.16666666	+0.16666666
YG09 =	+1.0	+1.0
YG09RPA =	+1.0	+1.0
Expected Outputs:		
YX09RYC =	+0.1745200	+3.029086
YN09FO =	+0.1745200	+3.029086
YN09FI =	+0.1745200	+0.1745200
Actual Outputs:		
YX09RYC =	+0.1745200	+3.029086
YN09FO =	+0.1745200	+3.029086
YN09FI =	+0.1745200	+0.1745200

3.9 YAW AXIS CONTROL ERROR COMPUTATION (YACM10) SUBMODULE TESTING

Test No. 2.3.10.1

The YACM10 was tested for the input values given. The actual results compared with the expected results as listed in the specifications.

The test results are as follows:

	Data Set 1
Inputs:	
YX07NYEF =	5.0
YXRGYF =	7.0
YX09RYC =	2.0
Constants:	
YG10RR =	1.0

Test 2.3.10.1 (continued)

	Data Set 1
Expected Output:	
YX10ERR =	10.0
Actual Output:	
YX10ERR =	10.0

3.10 YAW RATE COMMAND ERROR COMPUTATION (YACM11) SUBMODULE TESTING

Test No. 2.3.11.1

The YACM11 was tested for two sets of input values given. The actual results compared with the expected results as listed in the specifications.

The test results are as follows:

	Data Set 1	Data Set 2
Inputs:		
IXSQBAR =	+0.004999999	+100.0
YX10ERR =	+2.0	+2.0
Constants:		
YL11QBAR =	+0.01	+0.01
YC11KYAW =	+500.0	+500.0
YL11GU =	+6.0	+6.0
YL11GL =	+1.0	+1.0
Expected Output:		
YX11ERR =	+12.0	+10.0
Actual Output:		
YX11ERR =	+12.0	+10.0
Expected Intermediates:		
YT11QBAR =	+0.01	+100.0
YT11GRDC =	+6.0	+5.0
Actual Intermediates:		
YT11QBAR =	+0.01	+100.0
YT11GRDC =	+6.0	+5.0

3.11 YAW BENDING FILTER NO. 1 (YACM12) SUBMODULE TESTING

Test No. 2.3.12.1

The YACM12 was tested for the two sets of input values given. The actual results compared with the expected results as listed in the specifications.

The test results are as follows:

	Data Set 1	Data Set 2
Inputs:		
HXBILT =	-0001	+0000
YX11ERR =	+10.0	+10.0
YN12FI =	+10.0	+10.0
YF12N1 =	+2.0	+2.0
Constants:		
YK12IO =	+0.09007835	+0.09007835
YG12 =	+1.0	+1.0
Expected Outputs:		
YN12FO =	+10.0	+2.900782
YN12FI =	+10.0	+10.0
Actual Outputs:		
YN12FO =	+10.0	+2.900782
YN12FI =	+10.0	+10.0

3.12 YAW BENDING FILTER NO. 2 (YACM13) SUBMODULE TESTING

Test No. 2.3.13.1

The YACM13 was tested for the two sets of input values given. The actual results compare with the expected results as listed in the specifications.

The test results are as follows:

	Data Set 1	Data Set 2
Inputs:		
HXBFILT =	-0001	0000
YN12FO =	+10.0	+10.0
YF13N1 =	+2.0	+2.0

Test No. 2.3.13.1 (continued)

	Data Set 1	Data Set 2
Constants:		
YK13IO =	+0.9878048	+0.9878048
YG13 =	+1.0	+1.0
Expected Outputs:		
YXCRU =	+10.0	+11.87804
YN13FO =	+10.0	+11.87804
YN13FI =	+10.0	+10.0
Actual Outputs:		
YXCRU =	+10.0	+11.87804
YN13FO =	+10.0	+11.87804
YN13FI =	+10.0	+10.0

4. SPEEDBRAKE/RUDDER ACTUATOR COMMAND MODULE (FC04SRAM) TESTING

4.1 SPEEDBRAKE POSITION COMMAND SELECTION LOGIC SUBMODULE (PACM17)

The PACM17 was tested for the two sets of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.4.1.1

	Data Set 1	Data Set 2
Inputs:		
FXBSBR =	0000	-0001
IXCSBR =	+35.0	+35.0
IXCSBL =	-25.0	-25.0
Expected Output:		
PXCSBS =	-25.0	+35.0
Actual Output:		
PXCSBS =	-25.0	+35.0

4.2 SPEEDBRAKE ENABLE INITIATE LOGIC (PACM18)

The PACM18 was tested for the eight sets of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.4.2.1

	Data Set 1	Data Set 2	Data Set 3	Data Set 4
Inputs:				
IXBENG =	0000	0000	0000	0000
FXBSL =	0000	0000	-0001	-0001
FXBSBR =	0000	-0001	0000	-0001
Expected Output:				
PXBSBEL =	0000	0000	0000	0000
Actual Output:				
PXBSBEL =	0000	0000	0000	0000

Test #2.4.2.1 (continued)

	Data Set 5	Data Set 6	Data Set 7	Data Set 8
Inputs:				
IXBENG =	-0001	-0001	-0001	-0001
FXBSL =	0000	0000	-0001	-0001
FXBSBR =	0000	-0001	0000	-0001
Expected Output:				
PXBSBEL =	0000	-0001	-0001	-0001
Actual Output:				
PXBSBEL =	0000	-0001	-0001	-0001

4.3 RUDDER POSITION LIMITER SUBMODULE (ASPM11)

The ASPM11 was tested for the three sets of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.4.3.1

	Data Set 1	Data Set 2	Data Set 3
Inputs:			
YXCRU =	-30.0	+1.0	+30.0
Constants:			
AL11U =	+27.1	+27.1	+27.1
AL11L =	-27.1	-27.1	-27.1
Expected Output:			
ANCRU =	-27.09999	+1.0	+27.09999
Actual Output:			
ANCRU =	-27.09999	+1.0	+27.09999

4.4 RUDDER R/SB PANEL POSITION LIMITER SUBMODULE (ASPM12)

Test #2.4.4.1-4

The ASPM12 was tested for the four sets of input values given. The actual results compared with the expected results as listed in the specifications.

	Data Set 1	Data Set 2	Data Set 3	Data Set 4
Inputs:				
IXSSB =	+98.59999	+98.59999	+98.59999	0.0
ANCRU =	+27.09999	-27.09999	+1.0	-27.09999

Test #2.4.4.1-4 (continued)

	Data Set 1	Data Set 2	Data Set 3	Data Set 4
Constants:				
AC12 =	+0.5	+0.5	+0.5	+0.5
AL12 =	+60.0	+60.0	+60.0	+60.0
Expected Outputs:				
AVL12U =	+10.7	+10.7	+10.7	+60.0
AVL12L =	-10.7	-10.7	-10.7	-60.0
AN12CRU =	+10.7	-10.7	+1.0	-27.09999
Actual Outputs:				
AVL12U =	+10.70001	+10.70001	+10.70001	+60.0
AVL12L =	-10.70001	-10.70001	-10.70001	-60.0
AN12CRU =	+10.70001	-10.70001	+1.0	-27.09999

4.5 RUDDER POSITION RATE LIMITER SUBMODULE (ASPM13)

Test #2.4.5.1-4

The ASPM13 was tested for the four sets of input values given. The actual results compared with the expected results as listed in the specifications.

	Data Set 1	Data Set 2	Data Set 3	Data Set 4
Inputs:				
HXBFILT =	0000 (False)	-0001	0000	-0001
HXBENG =	0000 (False)	0000	-0001	-0001
AXCRU =	+10.5	+10.0	+10.0	+10.0
AN12CRU =	+8.0	+10.1	+12.0	+12.0
Constants:				
AL13U =	+0.242	+0.242	+0.242	+0.242
AL13L =	-0.242	-0.242	-0.242	-0.242
Expected Outputs:				
AT13CRUP =	+10.5	+10.0	+10.0	+10.0
AT13DRU =	-0.242	+0.10	+0.242	+0.242
AXCRU =	+10.258	+10.1	+10.242	+10.242
Actual Output:				
AT13CRUP =	+10.5	+10.0	+10.0	+10.0
AT13DRU =	-0.242	+0.09999948	+0.242	+0.242
AXCRU =	+10.258	+10.1	+10.242	+10.242

4.6 SPEEDBRAKE POSITION LIMITING AND HOLD FUNCTION SUBMODULE (ASPM14)

The ASPM14 was tested for the four sets of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.4.6.1-4

	Data Set 1	Data Set 2	Data Set 3	Data Set 4
Inputs:				
HXBFILT =	0000	-0001	0000	-0001
IXBENG =	0000	0000	-0001	-0001
IXSSB =	+99.0	+60.0	+60.0	+60.0
AN14CSBH =	+35.0	+35.0	+35.0	+35.0
PXBSBEI =	+0000	-0001	-0001	-0001
BXCSBS =	+10.0	+10.0	-1.0	+10.0
Constants:				
AL14U =	+98.6	+98.6	+98.6	+98.6
AL14L =	0.0	0.0	0.0	0.0
Expected Output:				
AN14CSBH =	+99.0	+60.0	+35.0	+60.0
ANCSB =	+98.59999	+10.0	0.0	+10.0
Actual Output:				
AN14CSBH =	+99.0	+60.0	+35.0	+60.0
ANCSB =	+98.59999	+10.0	0.0	+10.0

4.7 SPEEDBRAKE R/SB PANEL POSITION LIMITER SUBMODULE (ASPM15)

The ASPM15 was tested for the four sets of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.4.7.1-4

	Data Set 1	Data Set 2	Data Set 3	Data Set 4
Inputs:				
ANCRU =	20.0	0.0	-20.0	-20.0
ANCSB =	98.6	98.59999	98.6	-10.0
Constants:				
AT15CRUA =	2.0	2.0	2.0	2.0
AVL15U =	60.0	60.0	60.0	60.0
AN15CSB =	0.0	0.0	0.0	0.0

Test #2.4.7.1-4 (continued)

	Data Set 1	Data Set 2	Data Set 3	Data Set 4
Expected Outputs:				
AT15CRUA =	20.0	0.0	20.0	20.0
AVL15U =	80.0	120.0	80.0	80.0
AN15CSB =	80.0	98.59999	80.0	0.0
Actual Outputs:				
AT15CRUA =	20.0	0.0	20.0	20.0
AVL15U =	80.0	120.0	80.0	80.0
AN15CSB =	80.0	98.59999	80.0	0.0

4.8 SPEEDBRAKE POSITION RATE LIMITER SUBMODULE (ASPM16)

The ASPM16 was tested for the four sets of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.4.8.1-4

	Data Set 1	Data Set 2	Data Set 3	Data Set 4
Inputs:				
HXBFILT =	0000	-0001	0000	-0001
HXBENG =	0000	0000	-0001	-0001
AXCSB =	+10.5	+10.0	+10.0	+10.0
AN15CSB =	+8.0	+10.1	+12.0	+12.0
Constants:				
AL16U =	+0.122	+0.122	+0.122	+0.122
AL16L =	-0.122	-0.122	-0.122	-0.122
Expected Output:				
AT16CSBP =	+10.5	+10.0	+10.0	+10.0
AT16DSB =	-0.122	+0.1	+0.122	+0.122
AXCSB =	+10.378	+10.1	+10.122	+10.122
Actual Output:				
AT16CSBP =	+10.5	+10.0	+10.0	+10.0
AT16DSB =	-0.122	+0.09999948	+0.122	+0.122
AXCSB =	+10.378	+10.1	+10.122	+10.122

5. PITCH AXIS CONTROL MODULE (FC05PACM) TESTING

5.1 PITCH AUTO TRIM COMPENSATOR (PACM03) SUBMODULE TESTING

The PACM03 was tested for the two sets of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.5.3.1

	Data Set 1	Data Set 2
Inputs:		
HXBFILT =	-0001	0
AXSEVT =	+3.0	-12.0
PF03N1 =	+5.0	+7.0
Constants:		
PK03IO =	+0.0477832	+0.01477832
PG03 =	+1.0	+1.0
PG03EF =	+1.0	+1.0
Expected Outputs:		
PX03PAT =	+3.0	+6.822659
PN03FO =	+3.0	+6.822659
PN03PI =	+3.0	-12.0
Actual Outputs:		
PX03PAT =	+3.0	+6.822659
PN03FO =	+3.0	+6.822659
PN03FI =	+3.0	-12.0

5.2 LEFT RHC PITCH COMMAND DEADBANDING (PACM04) SUBMODULE TESTING

The PACM04 was tested for the single set of input values given. The actual results compared with the expected results as listed in the specifications.

Test # 2.5.4.1

	Data Set
Inputs:	
IXCPCL =	-1.20
Constant:	
PB04DBL =	+1.15

Test # 2.5.4.1 (continued)

Data Set

Expected Output:
PX04PCL = -0.05

Actual Output:
PX04PCL = -0.05

5.3 RIGHT RHC PITCH COMMAND DEADBANDING (PACM05) SUBMODULE TESTING

The PACM05 was tested for the single set of input values given. The actual results compared with the expected results as listed in the specifications.

Test # 2.5.5.1

Data Set

Inputs:
IXCPCR = +2.0

Constant:
PB05DBR = +1.15

Expected Output:
PX05PCR = +0.8500000

Actual Output:
PX05PCR = +0.8500004

5.4 COMBINED RHC PITCH COMMAND SHAPING AND LIMITING (PACM06) SUBMODULE TESTING

The PACM06 was tested for the single set of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.5.6.1

Data Set

Inputs:
PX04PCL = -130.0
PX05PCR = +40.0

Test #2.5.6.1 (continued)

	Data Set
Constants:	
PC06PS1 =	+0.36
PC06PS2 =	+0.0484
PL06U =	+23.0
PL06L =	-23.0
Expected Output:	
PX06PC =	-23.0
Actual Output:	
PX06PC =	-23.0
Expected Intermediates:	
PT06PCS =	-90.0
PT06PCPS =	-424.4399
Actual Intermediates:	
PT06PCS =	-90.0
PT06PCPS =	-424.4399

5.5 PITCH COMMAND FILTER (PACM07) SUBMODULE TESTING

The PACM07 was tested for the two sets of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.5.7.1

	Data Set 1	Data Set 2
Inputs:		
HXBFILT =	-0001	0000
PX06PC =	+3.0	+2.0
PN07FI =	+3.0	+2.0
PF07N1 =	+9.0	+9.0
Constants:		
PK07IO =	+1.980391	+1.980391
PG07 =	+1.0	+1.0
PG07DS =	+0.4	+0.4

Test # 2.5.7.1 (continued)

	Data Set 1	Data Set 2
Expected Outputs:		
PX07PCF =	+1.20	+5.184313
PN07FO =	+3.0	+12.96078
PN07FI =	+3.0	+2.0
Actual Outputs:		
PX07PCF =	+1.20	+5.184313
PN07FO =	+3.0	+12.96078
PN07FI =	+3.0	+2.0

5.6 MANUAL PITCH TRIM RECTANGULAR INTEGRATION (PACM08) SUBMODULE TESTING

The PACM08 was tested for the eight sets of input values given. The actual results compared with the expected results and this included the value of the intermediate variables as listed in the specifications.

Test #2.5.8.1

	Data Set 1	Data Set 2	Data Set 3	Data Set 4
Inputs:				
FXBPTP =	-0001	-0001	0000	0000
FXBPTN =	-0001	-0001	-0001	0000
IXBENG =	0000	-0001	-0001	-0001
PX08PT =	-4.009999	-4.0	+2.0	-4.0
Constants:				
PG08PT =	+0.0375	+0.0375	+0.0375	+0.0375
PC08DT =	+0.020	+0.020	+0.020	+0.020
PL08U =	+4.0	+4.0	+4.0	+4.0
PL08L =	-4.0	-4.0	-4.0	-4.0
Expected Output:				
PX08PT =	-4.0	-3.999249	+1.999250	+2.009999
Actual Output:				
PX08PT =	-4.0	-3.999249	+1.999250	+2.009999
Expected Intermediate:				
PT08PTR =	0.0	+0.0375	-0.0375	0.0

Test # 2.5.8.1 (continued)

	Data Set 1	Data Set 2	Data Set 3	Data Set 4
Actual Intermediate:				
PT08PTR =	0.0	+0.0375	-0.0375	0.0

	Data Set 5	Data Set 6	Data Set 7	Data Set 8
Inputs:				
FXBPTP =	0000	0000	-0001	-0001
FXBPTN =	0000	-0001	0000	0000
IXBENG =	0000	0000	0000	-0001
PX08PT =	+2.0	+2.0	+2.0	-4.0
Constants:				
PG08PT =	+0.0375	+0.0385	+0.0375	+0.0375
PC08DT =	+0.020	+0.020	+0.020	+0.020
PL08U =	+4.0	+4.0	+4.0	+4.0
PL08L =	-4.0	-4.0	-4.0	-4.0
Expected Output:				
PX08PT	+2.0	+2.0	+2.0	-3.999249
Actual Output:				
PX08PT	+2.0	+2.0	+2.0	-3.999249
Expected Intermediate:				
PT08PRT	0.0	0.0	0.0	+0.0375
Actual Intermediate:				
PT08PTR	0.0	0.0	0.0	+0.0375

5.7 TOTAL PITCH RATE COMMAND COMPUTATION (PACM09) SUBMODULE TESTING

The PACM09 was tested for the single set of input values given. The actual results compared with the expected results as listed in the specifications.

Test # 2.5.9.1

	Data Set
Inputs:	
PX07PCF =	+9.5
PX08PT =	+10.0

Test # 2.5.9.1 (continued)

	Data Set
Expected Output:	
PX09PRC =	-19.5
Actual Output:	
PX09PRC =	-19.5

5.8 PITCH RATE GYRO NOISE FILTER (PACM11) SUBMODULE TESTING

The PACM11 was tested for the two sets of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.5.11.1

	Data Set 1	Data Set 2
Inputs:		
HXBFILT =	-0001	0000
IXSRGP =	-2.0	+2.0
PF11N1 =	-5.0	+5.0
Constants:		
PK11IO =	+0.25	+0.25
PG11 =	+1.0	+1.0
Expected Outputs:		
PXRGPF =	-2.0	+5.5
PN11FO =	-2.0	+5.5
PN11FI =	-2.0	+2.0
Actual Outputs:		
PXRGPF =	-2.0	+5.5
PN11FO =	-2.0	+5.5
PN11FI =	-2.0	+2.0

5.9 PITCH RATE COMPENSATOR (PACM12) SUBMODULE TESTING

The PACM12 was tested for the two sets of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.5.12.1

	Data Set 1	Data Set 2
Inputs:		
HXBFILT =	-0001	0000
PXRGPF =	+12.0	+11.0
PF12N1 =	+3.0	+5.0
Constants:		
PK12I0 =	+1.980391	+1.980391
PG12 =	+1.0	+1.0
Expected Outputs:		
PX12RGPC =	+12.0	+26.78430
PN12FO =	+12.0	+26.78430
PN12FI =	+12.0	+11.0
Actual Outputs:		
PX12RGPC =	+12.0	+26.78430
PN12FO =	+12.0	+26.78430
PN12FI =	+12.0	+11.0

5.10 PITCH RATE COMMANDS ERROR COMPUTATION (PACM13) SUBMODULE TESTING

The PACM13 was tested for the three sets of input values given. The actual results compared with the expected results as listed and this included the intermediate values.

Test #2.5.13.1

	Data Set 1	Data Set 2	Data Set 3
Inputs:			
IXSQBAR =	+0.01	+110.0	-2.0
PX09PRC =	-4.0	+5.0	+1.2
PX12RGPC =	-3.0	-25.39999	+0.8

Test # 2.5.13.1 (continued)

	Data Set 1	Data Set 2	Data Set 3
Constants:			
PL13QBAR =	+0.01	+0.01	+0.01
PC13KPIT =	+220.0	+220.0	+220.0
PL13GU =	+6.0	+6.0	+6.0
PL13GL =	+1.0	+1.0	+1.0
PL13U =	+16.0	+16.0	+16.0
PL13L =	-41.0	-41.0	-41.0
Expected Output:			
PX13ERR =	-41.0	-40.79999	+12.0
Actual Output:			
PX13ERR =	-41.0	-40.79999	+11.9999
Expected Intermediates:			
PT13QBAR =	+0.01	+110.0	+0.01
PT13GDQ =	+6.0	+2.0	+6.0
Actual Intermediates:			
PT13QBAR =	+0.01	+110.0	+0.01
PT13GDQ =	+6.0	+2.0	+6.0

5.11 PITCH BENDING FILTER NO. 1 (PACM14) SUBMODULE TESTING

The PACM14 was tested for the two sets of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.5.14.1

	Data Set 1	Data Set 2
Inputs:		
HXPFILT =	-0001	0000
PX13ERR =	+2.0	+10.0
PN14FI =	+2.0	+10.0
PF14N1 =	+3.0	+4.0
Constants:		
PK14IO =	+0.09007835	+0.09007835
PG14 =	+1.0	+1.0

Test # 2.5.14.1 (continued)

	Data Set 1	Data Set 2
Expected Outputs:		
PN14FO =	+2.0	+4.900782
PN14FI =	+2.0	+10.0
Actual Outputs:		
PN14FO =	+2.0	+4.900782
PN14FI =	+2.0	+10.0

5.12 PITCH BENDING FILTER NO. 2 (PACM15) SUBMODULE TESTING

The PACM15 was tested for the two sets of input values given. The actual results compared with the expected results as listed in the specifications.

Test # 2.5.15.1

	Data Set 1	Data Set 2
Inputs:		
HXBFILT =	-0001	0000
PN14FO =	+3.0	+100.0
PF15N1 =	+5.0	-8.0
Constants:		
PK15IO =	+0.9878048	+0.9878048
PG15 =	+1.0	+1.0
Expected Outputs:		
PN15FI =	+3.0	+100.0
PN15FO =	+3.0	+90.78047
Actual Outputs:		
PN15FI =	+3.0	+100.0
PN15FO =	+3.0	+90.78047

5.13 ELEVON POSITION COMMAND COMPUTATION (PACM16) SUBMODULE TESTING

The PACM16 was tested for the eight sets of input values given. The actual results compared with the expected results as listed in the specifications.

Test # 2.5.16.1

	Data Set 1	Data Set 2	Data Set 3	Data Set 4
Inputs:				
HXBFILT =	-0001	0000	-0001	0000
IXBWOW =	-0001	0000	0000	-0001
IXBENG =	-0001	0000	0000	+0000
PX03PAT =	+3.0	+4.0	+5.0	+3.0
PN15FO =	+12.0	+3.0	+2.0	+2.0
PN16MET =	+200.0	+200.0	+200.0	+200.0
Expected Outputs:				
PXCEVT =	+15.0	+7.0	+7.0	+5.0
PN16MET =	+3.0	+4.0	+5.0	+3.0
Actual Outputs:				
PXCEVT =	+15.0	+7.0	+7.0	+5.0
PN16MET =	+3.0	+4.0	+5.0	+3.0

	Data Set 5	Data Set 6	Data Set 7	Data Set 8
Inputs:				
HXBFILT =	0000	0000	-0001	-0001
IXBWOW =	0000	-0001	-0001	0000
IXBENG =	-0001	-0001	0000	-0001
PX03PAT =	+2.0	+3.0	+5.0	+10.0
PN15FO =	+3.0	+2.0	+6.0	+2.0
PN16MET =	+200.0	+200.0	+200.0	+200.0
Expected Outputs:				
PXCEVT =	+5.0	+202.0	+11.0	+12.0
PN16MET =	+2.0	+200.0	+5.0	+10.0
Actual Outputs:				
PXCEVT =	+5.0	+202.0	+11.0	+12.0
PN16MET =	+2.0	+200.0	+5.0	+10.0

5.14 BODY FLAP COMMAND COMPUTATION (PACM20) SUBMODULE TESTING

The PACM20 was tested for the nine sets of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.5.7.1

	Data Set 1	Data Set 2	Data Set 3	Data Set 4
Inputs:				
FXBBFU =	-0001	0000	-0001	0000
FXBBFD =	-0001	0000	0000	-0001
IXSBF =	+22.5	+50.0	+22.48999	-11.7
Constants:				
PL20U =	+22.5	+22.5	+22.5	+22.5
PL20L =	-11.7	-11.7	-11.7	-11.7
Expected Outputs:				
PXBBFU =	0000	0000	-0001	0000
PXBBFD =	-0001	0000	0000	0000
PXBBFE =	-0001	0000	-0001	0000
Actual Outputs:				
PXBBFU =	0000	0000	-0001	0000
PXBBFD =	-0001	0000	0000	0000
PXBBFE =	-0001	0000	-0001	0000

	Data Set 5	Data Set 6	Data Set 7	Data Set 8
Inputs:				
FXBBFU =	-0001	-0001	-0001	0000
FXBBRD =	-0001	-0001	0000	0000
IXSBF =	-11.69	-18.1	+50.0	0.0
Constants:				
PL20U =	+22.5	+22.5	+22.5	+22.5
PL20L =	-11.7	-11.7	-11.7	-11.7
Expected Outputs:				
PXBBFU =	-0001	-0001	0000	0000
PXBBFD =	-0001	000	0000	0000
PXBBFE =	-0001	-0001	0000	0000

Test #2.5.7.1 (continued)

	Data Set 5	Data Set 6	Data Set 7	Data Set 8
Actual Outputs:				
PXBBFU =	-0001	-0001	0000	0000
PXBBFD =	0000	0000	0000	0000
PXBBFE =	-0001	-0001	0000	0000

	Data Set 9
Inputs:	
FXBBFU =	0000
FXBBFD =	0000
IXSBF =	-50.0
Constants:	
PL20U =	+22.5
PL20L =	-11.7
Expected Outputs:	
PXBBFU =	0000
PXBBFD =	0000
PXBBFE =	0000
Actual Outputs:	
PXBBFU =	0000
PXBBFD =	0000
PXBBFE =	0000

6. ELEVON ACTUATOR COMMAND MODULE (FC06EACM) TESTING

6.1 ELEVON POSITION LIMITER SUBMODULE (ASPM05).

Test # 2.6.1.1

	Data Set 1	Data Set 2	Data Set 3
Inputs:			
PXCEVT =	-40.0	+1.0	+30.0
Constants:			
AL05U =	+20.0	+20.0	+20.0
AL05L =	-35.0	-35.0	-35.0
Expected Output:			
AXCEVT =	-35.0	+1.0	+20.0
Actual Output:			
AXCEVT =	-35.0	+1.0	+20.0

6.2 AILERON POSITION LIMITER SUBMODULE (ASPM06).

The ASPM06 was tested for the three sets of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.6.2.1

	Data Set 1	Data Set 2	Data Set 3
Inputs:			
RXCAIL =	-15.0	+1.0	+15.0
Constants:			
AL06U =	+10.0	+10.0	+10.0
AL06L =	-10.0	-10.0	-10.0
Expected:			
AXCAIL =	-10.0	+1.0	+10.0
Actual Results:			
AXCAIL =	-10.0	+1.0	+10.0

6.3 LEFT ELEVON POSITION LIMITER SUBMODULE (ASPM07)

The ASPM07 was tested for the three sets of input values given. The actual results compared with the expected results as listed in the specifications.

Test # 2.6.3.1

	Data Set 1	Data Set 2	Data Set 3
Inputs:			
AXCEVT =	-25.0	+10.0	+15.0
AXCAIL =	-15.0	-9.0	+10.0
Constants:			
AL07U =	+20.0	+20.0	+20.0
AL07L =	-35.0	-35.0	-35.0
Expected Output:			
AN07CEL =	-35.0	+1.0	+20.0
Actual Output:			
AN07CEL =	-35.0	+1.0	+20.0

6.4 LEFT ELEVON POSITION RATE LIMITER (ASPM08)

The ASPM08 was tested for the four sets of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.6.4.1-4.

	Data Set 1	Data Set 2	Data Set 3	Data Set 4
Inputs:				
HXBFILT =	0000	-0001	0000	-0001
HXBENG =	0000	0000	-0001	-0001
ANSEL =	+10.0	+10.0	+10.0	+10.0
AN07CEL =	+8.0	+10.1	+12.0	+12.0
Constants:				
AL08U =	+0.40	+0.40	+0.40	+0.40
AL08L =	-0.40	-0.40	-0.40	-0.40
Expected Outputs:				
AT08DEL =	-0.40	+0.10	+0.40	+0.40
AXCEL =	+9.599999	+10.1	+10.4	+10.4
Actual Output:				
AT08DEL =	-0.40	+0.09999948	+0.40	+0.40
AXCEL =	+9.599999	+10.1	+10.4	+10.4

6.5 RIGHT ELEVON POSITION LIMITER SUBMODULE (ASPM09)

The ASPM09 was tested for the three sets of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.6.5.1

	Data Set 1	Data Set 2	Data Set 3
Inputs:			
AXCEVT =	-25.0	+10.0	+15.0
AXCAIL =	+15.0	+9.0	-10.0
Constants:			
AL09U =	+20.0	+20.0	+20.0
AL09L =	-35.0	-35.0	-35.0
Expected Output:			
AN09CER =	-35.0	+1.0	+20.0
Actual Output:			
AN09CER =	-35.0	+1.0	+20.0

6.6 RIGHT ELEVON POSITION RATE LIMITER (ASPM10)

The ASPM10 was tested for the four sets of input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.6.6.1-4

	Data Set 1	Data Set 2	Data Set 3	Data Set 4
Inputs:				
HXBFILT =	0000	-0001	0000	-0001
HXBENG =	0000	0000	-0001	-0001
AXCER =	+10.0	+10.0	+10.0	+10.0
AN09CER =	+8.0	+10.1	+12.0	+12.0
Constants:				
AL10U =	+0.4	+0.4	+0.4	+0.4
AL10L =	-0.4	-0.4	-0.4	-0.4
Expected Output:				
AT10DER =	-0.4	+0.1	+0.4	+0.4
AXCER =	+9.599999	+10.1	+10.4	+10.4

Test #2.6.6.1-4 (continued)

	Data Set 1	Data Set 2	Data Set 3	Data Set 4
Actual Output:				
AT1ODER =	-0.4	+.9000048	+0.4	+0.4
AXCER =	+9.599999	+10.1	+10.4	+10.4

7. FIRST ORDER FILTER NODE UPDATE MODULE (FC07FOUM) TESTING

7.1 SPECIFICATIONS: The test is designed to check the initialization of the first order filters, which occurs when the flag, HXBFILT, is 'TRUE'. On the first pass the output of the filter (RN05FO) is set equal to the input to the filter (RN05FI) times the DC gain of the filter (RG05). On subsequent passes, HXBFILT is set 'FALSE' signalling the end of filter initialization. However, if the input remains unchanged, the output of the filter (RN05FO) will also remain unchanged when the filter is updated. This mechanization results in minimum transients upon initialization."

7.2 FIRST ORDER FILTER INITIALIZATION AND STEP RESPONSE

Objective: To demonstrate necessary interactions between the C FILTER macro (filter computations routine) and CFPUSH1 macro (the first order filter node update computations routine). Only one filter was simulated. This was the RACM05 (Roll Rate Gyro Noise Filter). The test specifications and related data are presented herein.

Test #2.7.0.1

The first order filter initialization was tested for the single set of input values given. The actual results compared with the expected results as listed in the specifications.

	Data Set
Inputs:	
HXBFILT =	-0001
RN05FI =	+1.0
RN05FO =	0.0
Constants:	
RG05 =	1.0
RK05I0 =	+0.25
RK05I1 =	+0.25
RK05O1 =	-0.5

Test #2.7.0.1 (continued)

Pass no.	Expected filter output (RN05FO)	Actual filter output (RN05FO)	Expected nodal value (RF05N1)	Actual nodal value (RF05N1)
1	1.0	1.0	0.75	0.75
2	1.0	1.0	0.75	0.75
3	1.0	1.0	0.75	0.75
4	1.0	1.0	0.75	0.75
5	1.0	1.0	0.75	0.75
6	1.0	1.0	0.75	0.75
7	1.0	1.0	0.75	0.75
8	1.0	1.0	0.75	0.75
9	1.0	1.0	0.75	0.75
10	1.0	1.0	0.75	0.75

Test #2.7.0.2

This test is designed to check the transient response characteristics of the first order filter. The filter, RACM05, is initialized with the input values specified in the accompanying table. For the first and subsequent passes, HXBFILT = 'False'. The input to the filter is set to 1.0 and the filter node is initialized to 0.0. The resultant of these conditions is unit step response of the filter.

Data Set

Inputs:

HXBFILT = 0000

RN05FI = +1.0

RF05N1 = +0.0

Constant:

RG05 = +1.0

RK05I0 = +0.25

RK05I1 = +0.25

RK05O1 = -0.5

Pass no.	Expected filter output (RN05FO)	Actual filter output (RN05FO)	Expected nodal value (RF05N1)	Actual nodal value (RF05N1)
1	0.25	0.25	0.375	0.375
2	0.625	0.625	0.5625	0.5625
3	0.8125	0.8125	0.65625	0.65625
4	0.90625	0.90625	0.703125	0.703125
5	0.953125	0.953125	0.7265625	0.7265625
6	0.9765625	0.9765625	0.7382813	0.7382813

Test #2.7.0.2 (continued)

Pass no.	Expected filter output (RN05FO)	Actual filter output (RN05FO)	Expected nodal value (RF05N1)	Actual nodal value (RF05N1)
7	0.9882813	0.9882813	0.7441406	0.7441406
8	0.9941406	0.9941406	0.7470703	0.7470703
9	0.9970703	0.9970703	0.7485352	0.7485351
10	0.9985352	0.9985351	0.7492676	0.7492675

The FC07FOUM was tested for the sets of input values given. The actual results compared with the expected results as listed in the specifications.

Roll Command Filter

	Data Set
Inputs:	
RN04FI =	+1.0
RN04FO =	+2.0
RK04I1 =	+0.04761905
RK04O1 =	-0.9047619
Expected Output:	
RF04N1 =	+1.857143
Actual Output:	
RF04N1 =	+1.857143

Roll Rate Gyro Noise Filter

	Data Set
Inputs:	
RN05FI =	+3.0
RN05FO =	+4.0
RK05I1 =	+0.25
RK05O1 =	-0.5
Expected Output:	
RF05N1 =	+2.75
Actual Output:	
RF05N1 =	+2.75

Test #2.7.0.2 (continued)

Radar Altitude Noise Filter

	Data Set
Inputs:	
YN04FI =	+5.0
YN04FO =	+6.0
YN04I1 =	+0.25
YN04O1 =	-0.5
Expected Output:	
YF04N1 =	+4.25
Actual Output:	
YF04N1 =	+4.25

Lateral Acceleration Filter

	Data Set
Inputs:	
YN06FI =	+7.0
YN06FO =	+8.0
YK06I1 =	+0.02380952
YK06O1 =	-0.9047619
Expected Output:	
YF06N1 =	+7.404761
Actual Output:	
YF06N1 =	+7.404761

Lateral Acceleration Command Error Filter

	Data Set
Inputs:	
YN07FI =	+10.0
YN07FO =	+11.0
YK07I1 =	+0.2
YK07O1 =	-0.6
Expected Output:	
YF07N1 =	+8.599999
Actual Output:	
YF07N1 =	+8.599999

Test #2.7.0.2 (continued)

Yaw Rate Gyro Noise Filter

	Data Set
Inputs:	
YN08FI =	+12.0
YN08FO =	+13.0
YK08I1 =	+0.25
YK08O1 =	-0.5
Expected Output:	
YF08N1 =	+9.5
Actual Output:	
YF08N1 =	+9.5

Roll/Yaw Rate Compensator

	Data Set
Inputs:	
YN09FI =	+14.0
YN09FO =	+15.0
YK09I1 =	+0.1666666
YK09O1 =	-0.6666666
Expected Output:	
YF09N1 =	+12.33333
Actual Output:	
YF09N1 =	+12.33333

Pitch Auto Trim Compensator

	Data Set
Inputs:	
PN03FI =	+16.0
PN03FO =	+17.0
PK03I1 =	+0.01477832
PK03O1 =	-0.97044335
Expected Output:	
PF03N1 =	+16.73398
Actual Output:	
PF03N1 =	+16.73398

Test #2.7.0.2 (continued)

Pitch Command Filter

	Data Set
Inputs:	
PN07FI =	+18.0
PN07FO =	+19.0
PK07I1 =	-1.941176
PK07O1 =	-0.9607843
Expected Output:	
PF07N1 =	-16.68626
Actual Output:	
PF07N1 =	-16.68625

Pitch Rate Gyro Noise Filter

	Data Set
Inputs:	
PN11FI =	+20.0
PN11FO =	+21.0
PK11I1 =	+0.25
PK11O1 =	-0.5
Expected Output:	
PF11N1 =	+15.5
Actual Output:	
PF11N1 =	+15.5

Pitch Rate Compensator

	Data Set
Inputs:	
PN12FI =	+22.0
PN12FO =	+23.0
PK12I1 =	-1.941176
PK12O1 =	-0.9607843
Expected Output:	
PF12N1 =	-20.60783
Actual Output:	
PF12N1 =	-20.60782

8. SECOND ORDER FILTER NODE INITIALIZATION MODULE (FC08SOIM) TESTING

8.1 CASCADED BENDING FILTER INITIALIZATION TEST

Objective: To demonstrate necessary interactions between the CFILTER macro (filter computations routine) and the CINIT2 macro (second order filter node initialization computations). For this purpose, two cascaded second order bending filters (RACM07 — Roll Bending Filter No. 1 and RACM08 — Roll Bending Filter No. 2) were simulated.

Test #2.8.0.1

	Data Set		Data Set
Constants:			
RG07 =	+1.0	RG08 =	+1.0
RK07I0 =	+9.09007835	RK08I0 =	+0.9878048
RK07I1 =	+0.02349870	RK08I1 =	-0.7317072
RK07I2 =	-0.06657963	RK08I2 =	+0.2317073
RK07O1 =	-1.608355	RK08O1 =	-0.7317072
RK07O2 =	+0.6553525	RK08O2 =	+0.2195122
Inputs:			
HXBFILT =	-001 (true)		
RN07FI =	1.0		
RF07N1 =	0.0		
RF07N2 =	0.0		
RF08N1 =	0.0		
RF08N2 =	0.0		

Pass No.	Expected Output for 1st Bending Filter (RN07FO)	Actual Output for 1st Bending Filter (RN07FO)	Expected Output for 2nd Bending Filter (RN08FO)	Actual Output for 2nd Bending Filter (RN08FO)
1	1.0	1.0	1.0	1.0
2	1.0	1.0	1.0	0.9999999
3	1.0	1.0	1.0	0.9999999
4	1.0	1.0	1.0	0.9999998
5	1.0	1.0	1.0	0.9999998
6	1.0	1.0	1.0	0.9999998
7	1.0	1.0	1.0	0.9999998
8	1.0	1.0	1.0	0.9999998
9	1.0	1.0	1.0	0.9999998
10	1.0	1.0	1.0	0.9999998

TEST #2.8.1

The FC08SOIM was tested for the input values given. The actual results compared with the expected results as listed in the specifications.

Roll Bending Filter No. 1	Data Set	Roll Bending Filter No. 2	Data Set
Inputs:		Inputs:	
RN07FI =	+10.0	RN08FI =	+10.0
RN07FO =	+10.0	RN08FO =	+10.0
RK07I1 =	+0.02349869	RN08I1 =	-1.282051
RK07I2 =	-0.06657963	RK08I2 =	+0.4700855
RK07O1 =	-1.608355	RK08O1 =	-1.282051
RK07O2 =	+0.6553524	RK08O2 =	+0.4529914
Expected Output:		Expected Output:	
RF07N2 =	-7.219320	RF08N2 =	+0.1709402
RF07N1 =	+9.099217	RF08N1 =	+0.1709402
Actual Output:		Actual Output:	
RF07N2 =	-7.219319	RF08N2 =	+0.1709402
RF07N1 =	+9.099212	RF08N1 =	+0.1709402

Yaw Bending Filter No. 1

Data Set

Inputs:

YN12FI =	20.0
YN12FO =	20.0
YK12I1 =	0.02349869
YK12I2 =	-0.06657963
YK12O1 =	-1.608355
YK12O2 =	0.6553524

Expected Output:

YF12N2 =	-14.43864
YF12N1 =	+18.19843

Actual Output:

YF12N2 =	-14.43864
YF12N1 =	+18.19843

Yaw Bending Filter No. 2

Data Set

Inputs:

YN13FI =	+20.0
YN13FO =	+20.0
YK13I1 =	-1.282051
YK13I2 =	+0.4700855
YK13O1 =	-1.282051
YK13O2 =	+0.4529914

Expected Output:

YF13N2 =	+0.3418804
YF13N1 =	+0.3418804

Actual Output:

YF13N2 =	+0.3418803
YF13N1 =	+0.3418803

Pitch Bending Filter No. 1

Data Set

Inputs:

PN14FI =	+30.0
PN14FO =	+30.0
PK14I1 =	+0.02349869
PK14I2 =	-0.06657963
PK14O1 =	-1.608355
PK14O2 =	+0.6553524

Expected Output:

PF14N2 =	-21.65796
PF14N1 =	+27.29763

Actual Output:

PF14N2 =	-21.65796
PF14N1 =	+27.29763

Pitch Bending Filter No. 2

Data Set

Inputs:

PN14FI =	+30.0
PN15FO =	+30.0
PK15I1 =	-1.282051
PK15I2 =	+0.4700855
PK15O1 =	-1.282051
PK15O2 =	+0.4529914

Expected Output:

PF15N2 =	+0.5128206
PF15N1 =	+0.5128206

Actual Output:

PF15N2 =	+0.512805
PF15N1 =	+0.5128205

9. SECOND ORDER FILTER NODE UPDATE MODULE (RC09SOUM) TESTING

9.1 CASCADED BENDING FILTER STEP RESPONSE

Objective: To demonstrate necessary interactions between the CFILTER macro (filter computation) and the CFPUSH2 macro (the second order filter node update computations routine). Two cascaded second order bending filters (RACM07 and RACM08) were simulated.

Test # 2.9.0.1

	Data Set		Data Set
Constants:			
RG07 =	+1.0	RG08 =	+1.0
RK07I0 =	+0.09007835	RK08I0 =	+0.9878048
RK07I1 =	+0.02349870	RK08I1 =	-0.7317072
RK07I2 =	-0.06657963	RK08I2 =	+0.2317073
RK0701 =	-1.608355	RK0801 =	-0.7317072
RK0702 =	+0.6553525	RK0802 =	+0.2195122
Inputs:			
HXBILT =	0000 (False)	RF07N2 =	0.0
RN07FILT =	+1.0	RF08N1 =	0.0
RF07N1 =	0.0	RF08N2 =	0.0

Pass No.	Expected Output for 1st Bending Filter (RN07FO)	Actual Output for 1st Bending Filter (RN07FO)	Expected Output for 2nd Bending Filter (RN08FO)	Actual Output for 2nd Bending Filter (RN08FO)
1	0.09007835	0.09007835	0.08897984	0.08897978
2	0.2584549	0.2584549	0.2544993	0.2544991
3	0.4036517	0.4036517	0.3971743	0.3971743
4	0.5268335	0.5268338	0.5196894	0.5196896
5	0.6297986	0.6297994	0.6232352	0.6232358
6	0.7146756	0.7146768	0.7091505	0.7091516
7	0.7837094	0.7837108	0.7792304	0.7792318
8	0.8391159	0.8391177	0.8355339	0.8355355
9	0.8829879	0.8829894	0.8801394	0.8801407
10	0.9172388	0.9172397	0.9149881	0.9149889

9.2 SECOND ORDER FILTER UPDATE MODULE (FC09SOUM) TESTING

The FC09SOUM was tested for the input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.9.1

Roll Bending Filter No. 1	Data Set	Roll Bending Filter No. 2	Data Set
Inputs:			
RN07FI =	+10.0	RN08FI =	+10.0
RN07FO =	+20.0	RN08FO =	+20.0
RF07N2 =	+30.0	RF08N2 =	+30.0
RK07I1 =	+0.02349869	RK08I1 =	-1.282051
RK07I2 =	+0.06657963	RK08I2 =	+0.4700855
RK07O1 =	-1.608355	RK08O1 =	-1.282051
RK07O2 =	+0.6553524	RK08O2 =	+0.4529914
Expected Output:			
RF07N1 =	+62.40208	RF08N1 =	+42.82051
RF07N2 =	-13.77285	RF08N2 =	-4.358974
Actual Output:			
RF07N1 =	+62.40206	RF08N1 =	+42.82051
RF07N2 =	-13.77284	RF08N2 =	-4.358974

Yaw Bending Filter No. 1	Data Set	Yaw Bending Filter No. 2	Data Set
Inputs:			
YN12FI =	+20.0	YN13FI =	+20.0
YN13FO =	+40.0	YN13FO =	+40.0
YF12N2 =	+60.0	YF13N2 =	+60.0
YK12I1 =	+0.02349869	YK13I1 =	-1.282051
YK12I2 =	-0.06657963	YK13I2 =	+0.4700855
YK12O1 =	-1.608355	YK13O1 =	-1.282051
YK12O2 =	+0.6553524	YK13O2 =	+0.4529914
Expected Output:			
YF12N1 =	+124.8041	YF13N1 =	+85.64102
YF12N2 =	-27.54569	YF13N2 =	-8.717948
Actual Output:			
YF12N1 =	+124.8041	YF13N1 =	+85.64102
YF12N2 =	-27.54569	YF13N2 =	-8.717948

Pitch Bending Filter No. 1		Pitch Bending Filter No. 2	
Inputs:			
PN14FI =	+30.0	PN15FI =	+30.0
PN14FO =	+60.0	PN15FO =	+60.0
PF14N2 =	+90.0	PN15N2 =	+90.0
PK14I1 =	+0.02349869	PK15I1 =	-1.282051
PK14I2 =	-0.06657963	PK15I2 =	+0.4700855
PK14O1 =	-1.608355	PK14O1 =	-1.282051
PK14O2 =	+0.65535[illegible]1	PK15O2 =	+0.4529914
Expected Output:			
PF14N1 =	+187.2062	PF15N1 =	+128.4615
PF14N2 =	-41.31852	PF15N2 =	-13.07692
Actual Outputs:			
PF14N1 =	+187.2062	PF15N1 =	+128.4615
PF14N2 =	-41.31852	PF15N2 =	-13.07692

10. COMMON ROUTINE MODULE TESTING

10.1 LIMITER ROUTINE (CLIMITER) TESTING

The Limiter Routine (CLIMITER) was tested for the input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.10.1.1

Data Set

Constants:

ULIMIT = +10.0 (upper limit)

LLIMIT = -5.0 (lower limit)

Inputs (X):	Expected Results (Y')	Actual Results (Y)
-15.0	-5.0	-5.0
-5.99999	-5.0	-5.0
-5.0	-5.0	-5.0
-4.9	-4.9	-4.9
-2.5	-2.5	-2.5
0.0	0.0	0.0
+2.5	+2.5	+2.5
+9.9	+9.9	+9.9
+10.0	+10.0	+10.0
+10.1	+10.0	+10.0
+15.0	+10.0	+10.0

10.2 DEADBAND ROUTINE (CDBAND) TESTING

The Deadband Routine (CDBAND) was tested for the input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.10.2.1

Constants: DB = 1.61 (deadband value)

Input (X):	Expected Output (Y')	Actual Output (Y)
-5.0	-3.389999	-3.39
-1.62	-0.01	-0.01000023
-1.61	0.0	0.0
-1.599999	0.0	0.0
-1.0	0.0	0.0
0.0	0.0	0.0

Test #2.10.2.1 (continued)

Input (X):	Expected Output (Y')	Actual Output (Y)
+1.0	0.0	0.0
+1.599999	0.0	0.0
+1.61	0.0	0.0
+1.62	+0.01	+0.01000023
+5.0	+3.389999	+3.39

10.3 PARABOLIC SHAPER (CPSHAPER) TESTING

The Parabolic Shaper Routine (CPSHAPER) was tested for the input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.10.3.1

Constants:
PS1 = 0.08 (First Order Parabolic Shaping Coef.)
PS2 = 0.051 (Second Order Parabolic Shaping Coef.)

Inputs (X)	Expected Output (Y')	Actual Output (Y)
-20.0	-22.0	-21.99998
-15.0	-12.675	-12.675
-10.0	-5.899999	-5.899999
-5.0	-1.674999	-1.674999
0.0	0.0	0.0
+5.0	+1.674999	+1.674999
+10.0	+5.899999	+5.899999
+15.0	+12.675	+12.675
+20.0	+22.0	+21.99998

10.4 FILTER ROUTINE (DFILTER) TESTING

The normal pass of the Filter Routine (CFILTER) was tested for the input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.10.4.1

Data Set

Inputs:
BFILT = 0000 (FLAG) (Filter Initialization Flag)
FI = +2.609999 (Filter input)

Test #2.10.4.1 (continued)

Data Set

N1 =	-7.71 (Filter Node No. 1)
KIO =	+1.12 (Filter N=O Input Coef.)
DCGAIN =	+3.5 (Filter DC Gain)

Expected Output:

FO = 4.786799 (Filter Output)

Actual Output:

FO = 4.786799 (Filter Output)

10.5 FILTER ROUTINE (CFILTER) TESTING

The initialization pass of the Filter Routine (CFILTER) was tested for the input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.10.4.3

Data Set

Inputs:

BFILT =	-0001 (Filter Initialization Flag)
FI =	+2.609999 (Filter Input)
N1 =	-7.71 (Filter Node No. 1)
KIO =	+1.12 (Filter N=O Input Coef.)
DCGAIN =	+3.50 (Filter DC Gain)

Expected Output:

FO = +9.134999 (Filter Output)

Actual Output:

FO = +9.134998 (Filter Output)

10.6 FIRST ORDER FILTER PUSHDOWN ROUTINE (CF1PUSH) TESTING

Test the First Order Filter Pushdown Routine (CF1PUSH) was tested for the input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.10.5.1

Data Set

Inputs:

FI =	+2.609999
FO =	+6.50

Test #2.10.5.1 (continued)

	Data Set
KI1 =	+4.12
IO1 =	+6.58
Expected Output:	
N1 =	-32.01678
Actual Output:	
N1 =	-32.01680

10.7 SECOND ORDER FILTER PUSHDOWN ROUTINE (CF2PUSH) TESTING

The Second Order Filter Pushdown Routine (CF2PUSH) was tested for the input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.10.6.1

	Data Set
Inputs:	
FI =	-5.56
FO =	+7.97
N2 =	-1.339999
KI1 =	+4.1
KI2 =	+6.55
KO1 =	+2.37
KO2 =	-1.86
Expected Output:	
N1 =	-43.19170
N2 =	-21.59379
Actual Output:	
N1 =	-43.19168
N2 =	-21.59379

10.8 SECOND ORDER FILTER NODE INITIALIZATION ROUTINE (CN2INIT) TESTING

The Second Order Filter Node Initialization Routine (CN2INIT) was tested for the input values given. The actual results compared with the expected results as listed in the specifications.

Test #2.10.7.1

	Data Set
Inputs:	
FI =	-5.559999
FO =	+7.97
KI1 =	+4.13
KI2 =	+6.55
KO1 =	+2.37
KO2 =	-1.86
Expected Output:	
N1 =	-63.44549
N2 =	-21.59379
Actual Output:	
N1 =	-63.44549
N2 =	-21.59379

APPENDIX E

BFCS SIGNALS IMPLEMENTED ON BFCSFE

AIR DATA INPUT FROM TR48

WORD NO.	PARAMETER	MDM NO.	EPROM ADDRESS	COMMENT
1*	JXSADC		0	NOT USED
2*	JXSQC		1	NOT USED
3*	JXSQTOTC		2	NOT USED
4*	JXSTAT		3	NOT USED
5	JXSALPHU	5	4	ALPHA UNC
6*	JXSBETAU		5	NOT USED
7	JXSEAS	5	6	EQUIL. AIR SPEED
8	JXSMACH	5	7	MACH NUMBER
9*	JXSPALT		8	NOT USED
10	JXSVV	5	9	ALT. RATE
11*	JXSQBAR		10	NOT USED
12*	JXSQU		11	NOT USED
13	JXSBARO	5	12	BARO-CORR. ALT
14*	JXSQTOTU		13	NOT USED
15	JXSALPHA	5	14	ALPHA CORR.
16	JXSMDM5	5	15	MDM STATUS (BITE)

*NOT USED

FLIGHT CONTROL COMMANDS OUTPUT TO TR48

WORD NO.	PARAMETER	UNIT	MDM NO.	EPROM ADDRESS	COMMENT
1	OXCIERA	TR48	1	128	RT. IN-BOARD ELEV. RESET
2*	OXCRGA		1	129	NOT USED
3	OXCIELA	TR48	1	130	LT. IN-BOARD ELEV. RESET
4*	OXCPGA		1	131	NOT USED
5*	OXCYGA		1	132	NOT USED
6	OXCSBA	TR48	1	133	SPEEDBRAKE A
7	OXCRUA	TR48	1	134	RUDDER A
8*	OXCOERA			135	NOT USED
9*	OXCOELA			136	NOT USED
	DISCRETES				
1	OXB1A	TR48	1	69	BIT 3 OXBBFD1 - BODY FLAP DOWN CMD BIT 10 OXBBFU1 - BODY FLAP UP CMD
2*	OXBIABAR				NOT USED (AND NOT TRANSMITTED)
3	OXB2A	TR48	1	70	OXBBFE1 - BODY FLAP ENABLE BIT 0 DISCRETE
4*	OXB2ABAR				NOT USED (AND NOT TRANSMITTED)

*NOT USED

ANALOG INPUT

WORD NO.	PARAMETER	MDM NO.	EPROM ADDRESS	COMMENT
				HAND CONTROL
1*	JXCSBR		16	NOT USED
2	JXCPCL	5	17	PITCH CMD + RHC LEFT
3	JXCRU	5	18	YAW CMD (RUDDER PEDAL)
4*	JXCPCR		19	NOT USED
5	JXCSBL	5	20	SPEEDBRAKE LEFT HC
6	JXCRCL	5	21	ROLL CMD (RHC) LEFT
7*	JXCRCR		22	NOT USED
				ACTUATOR
8	JXSBF	5	23	BODY FLAPS
9	JXSIEL	5	24	IN-BOARD ELEV. LT.
10	JXSSB	5	25	SPEEDBRAKE
11*	JXSOEL		26	NOT USED
12	JXSRU	5	27	RUDDER PEDAL
13	JXSIER	5	28	IN-BOARD ELEV. RT.
14*	JXSOER		29	NOT USED
				TR48
15	JXSNY	5	30	ACC LATERAL
16	JXSRGR	5	31	ROLL DATE GYRO
17	JXSRAD	5	32	RADAR ALT
18	JXSRGP	5	33	PITCH RATE GYRO
19	JXSRGY	5	34	YAW RATE GYRO
20	JXSNZ	5	35	ACC. NORM

*NOT USED

INPUT DISCRETES FROM IOPS

WORD NO.	PARAMETER	MDM NO.	EPROM ADDRESS	COMMENT
	DISC 1			
1	JXB1	5	64	SBTC ENGAGE, BIT 9 TAKEOVER
	DISC 2			
1	JXB2	5	65	†
2	JXB3	5	66	NOT USED
3	JXB4	5	67	‡
	DISC 3			
1	JXB5	IOPS	DISCRETES→	HALT MODE CMD (BIT 0) STBY MODE CMD (BIT 1) RUN MODE CMD (BIT 2) BFCS ENGAGE 1 (BIT 3) BFCS ENGAGE 2 (BIT 4) BFCS ENGAGE 4 (BIT 5) TERM CMD (BIT 6)
	DISC 4			
1	JXB6	5	68	RADAR ALT. LOCK ON BIT 6

†BIT 0 - LH BODY FLAP DOWN CMDBFCS
BIT 1 - LH BODY FLAP UP CMDBFCS
BIT 3 - FCS MON CH A RES/OVRD A
BIT 4 - FCS MON CH B RES/OVRD A
BIT 5 - FCS MON CH D RES/OVRD A
BIT 9 - NO WOW
BIT 14 - FCS MON CH C RES/OVRD A

‡BIT 4 - LH + PITCH TRIM UP
BIT 5 - LH - PITCH TRIM DOWN
BIT 6 - LH + ROLL TRIM LEFT
BIT 7 - LH - ROLL TRIM RIGHT
BIT 10 - LH + YAW TRIM LEFT
BIT 11 - LH - YAW TRIM RIGHT
BIT 14 - LH ROLL TRIM DISABLE
BIT 15 - LH PITCH TRIM DISABLE

OUTPUT TO SPI DISPLAY

WORD NO.	PARAMETER	MDM NO.	EPROM ADDRESS	COMMENT
1	OXDBF	5	128	BODY FLAP SPI
2	OXDSB	5	129	SPEED BRAKE
3	OXDSBC	5	130	SPEED BRAKE CMD
4	OXDSAIL	5	131	ACCELERATE POSITION X
5	OXDOER	5	132	OUT-BOARD ELEVON RT
6	OXDIER	5	133	IN-BOARD ELEVON RT (SPI)
7	OXDOEL	5	134	OUT-BOARD ELEVON LT (SPI)
8	OXDRU	5	135	RUDDER (SPI)
9	OXDIEL	5	136	IN-BOARD ELEVON LEFT

*NOT USED

OUTPUT TO ADI DISPLAY

WORD NO.	PARAMETER	MDM NO.	EPROM ADDRESS	COMMENT
1*	OXDCADI		120	NOT USED
2*	OXDTES1		121	NOT USED
3*	OXDSINR		122	NOT USED
4*	OXDCOSR		123	NOT USED
5*	OXDSINP		124	NOT USED
6*	OXDCOSP		125	NOT USED
7*	OXDSINY		126	NOT USED
8*	OXDCOSY		127	NOT USED
9	OXDRGR	2	128	ROLL RATE
10	OXDRGP	2	129	PITCH RATE
11	OXDRGY	2	130	YAW RATE
12	OXDER	2	131	ROLL ERROR
13	OXDEP	2	132	PITCH ERROR
14	OXDEY	2	133	YAW ERROR

*NOT USED

OUTPUT TO AMI DISPLAY

WORD NO.	PARAMETER	MDM NO.	EPROM ADDRESS	COMMENT
1*	OXDCAMI		134	NOT USED
2*	OXDTES3		135	NOT USED
3	OXDMACH	2	136	MACH NO. INDICATOR
4	OXDALPHA	2	137	ALPHA INDICATOR
5	OXDEAS	2	138	CAS INDICATOR
6	OXDNX	2	139	AIR SPEED COMMAND

*NOT USED

OUTPUT TO SPI DISCRETES

WORD NO.	PARAMETER	MDM NO.	EPROM ADDRESS	COMMENT
1	OXB3	5	69	BFCS CAUTION BIT 0, SPI VALID BIT 2
2*	OXB3BAR			NOT USED (AND NOT TRANSMITTED)
1	OXB4	5	70	BFCS ENGAGE LAMP BIT 0
2*	OXB4BAR			NOT USED (AND NOT TRANSMITTED)
1	OXB5		TO IOPS DIRECT	BIT 0 = GPC-IOP 3 FAIL (S/W) BIT 1 = OFP EXECUTION TIME BIT 2 = ANALOG COMP ON BIT 8 = BFCS FAIL BIT 9 = GPC READY IND BIT 10 = ANALOG FETCH START BIT 11 = OUTPUT TO AFT MDM(1) NOTE: ONLY BITS 0, 8, AND 9 ARE MAINTAINED IN OXB5. BITS 1, 2, 10, AND 11 ARE DYNAMICALLY SET AND RESET
2*	OXB5BAR			NOT USED (AND NOT TRANSMITTED)

*NOT USED