

et
86
31

COPY 3

DOT/FAA/CT-86/31

FAA TECHNICAL CENTER
Atlantic City International Airport
N.J. 08405

Hardware Fault Insertion and Instrumentation System: Mechanization and Validation

J.W. Benson
Lockheed-Georgia Company
A Division of Lockheed Corporation
Marietta, Georgia 30063

March 1987

Final Report

This document is available to the U.S. public
through the National Technical Information
Service, Springfield, Virginia 22161.



U.S. Department of Transportation
Federal Aviation Administration

1. Report No. DOT/FAA/CT-86/31	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle Hardware Fault Insertion and Instrumentation System: Mechanization and Validation		5. Report Date March 1987	6. Performing Organization Code
7. Author(s) J. W. Benson		8. Performing Organization Report No. DOT/FAA/CT-86/31	10. Work Unit No. NAS2-11853
9. Performing Organization Name and Address Lockheed-Georgia Company Marietta, Georgia 30063		11. Contract or Grant No.	13. Type of Report and Period Covered Contractor Report
12. Sponsoring Agency Name and Address U.S. Department of Transportation Federal Aviation Administration Technical Center Atlantic City Airport, New Jersey 08405		14. Sponsoring Agency Code	
15. Supplementary Notes Point of Contact: W. E. Larsen/MS 210-2 Ames Research Center Moffett Field, CA 94035			
16. Abstract This effort was performed to provide automated test capability for extensive low-level hardware fault insertion testing. The capability developed was applied in a related task that is described in DOT/FAA/CT-86/34. The test capability was used to calibrate fault detection coverage and associated latency times as relevant to projecting overall system reliability. This report describes modifications made to the NASA Ames Reconfigurable Flight Control System (RDFCS) Facility to fully automate the total test loop involving the Draper Laboratories' Fault Injector Unit. The automated capability provided included the application of sequences of simulated low-level hardware faults, the precise measurement of fault latency times, the identification of fault symptoms, and bulk storage of test case results. A PDP-11/60 served as a test coordinator, and a PDP-11/04 as an instrumentation device. The fault injector was controlled by applications test software in the PDP-11/60, rather than by manual commands from a terminal keyboard. The time base was specially developed for this application to use a variety of even signal sources in the system simulator.			
17. Key Words (Suggested by Author(s)) Automated Testing, Command Files, Digital Flight Computers, Fault Handling, Instrumentation, Hardware Faults, Low-Level Testing, Real-Time Testing, Time Measurements		18. Distribution Statement Unlimited Subject Category 38	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages	22. Price*

DOT/FAA/CT-86/31

HARDWARE FAULT INSERTION AND INSTRUMENTATION SYSTEM:
MECHANIZATION AND VALIDATION

J. W. Benson

Prepared for the
Federal Aviation Administration
Under Contract NAS2-11853

Lockheed-Georgia Company Engineering Report LG86ER0161

14 November 1986

LIST OF FIGURES

<u>Number</u>	<u>Title</u>	<u>Page</u>
1	Overall FIIS Block Diagram	2
2	FIIS Software Organization	5
3	Fault Injector Logic	7
4	40-Pin Fault Injector Set-up - AM2901	21
5	18-Pin Fault Injector Set-up - C1, C3, C4, C5, C6, C7, C9, C10, C11, C12, C13, C17, C18	22
6	Non-Standard 20-Pin Fault Injector Set-up - C14, C15, C16	23
7	14-Pin Fault Injector Set-up - C20, C29, C39	24
8	Standard 20-Pin Fault Injector Set-up - P12, P19, P20, C40, C42	25
9	16-Pin Fault Injector Set-up	26

HARDWARE FAULT INSERTION AND INSTRUMENTATION SYSTEM (FIIS)

This system was developed for the FAA for use with the NASA-AMES RDFCS (Reconfiguration Digital Flight Control System) facility utilizing the Draper Fault Injection System with the Collins CAPS-6 flight computers.

The Hardware Fault Injection and Instrumentation System (FIIS) allows the automatic insertion of pre-selected faults into the CAPS-6 computers in the RDFCS system via the Draper Fault Injection System (FIS) (Reference 1). A sequence of predefined fault insertion commands can be set up in command files in the PDP 11/60 to apply faults to specific chips within the CAPS-6 computers. A failure sequence then takes place within the CAPS processor followed by some detectable action such as the disengagement of the autopilot engagementbat handle switch on the Glare Shield panel. This action can then be used to generate an interrupt back to the PDP 11/60 indicating that a failure has occurred. The system can be automated to inject the same fault many times to yield a statistical effect of the failure across the frame time of the CAPS-6 computation, or many different faults can be injected with variations on the failure sequence.

FIIS SYSTEM CONFIGURATION

Four candidate FIIS architectures were proposed in a prior study (Ref. 2). Of these candidate architectures, the FAA chose to proceed with an ehanced version of Option One, which was a minimum effort utilizing the existing RDFCS facility with essentially software modifications only. This FIIS architecture is composed of the existing RDFCS facility supplemented with the Draper fault injection unit. The facility consists of the following equipment arranged in the configuration shown in Figure 1.

PDP 11/60	minicomputer
PDP 11/04	minicomputer
DEC DA11B	DMA interface
DEC DR11C	processor interface
Collins RDFCS	pallet consisting of:
4	- CAPS-6 flight computers
4	- CAPS Test Adapter (CTA) units for controlling the flight computers MDICU (Modular Digital Interface Control Unit) between the PDP 11/60 and the RDFCS pallet Draper Fault Injection Unit.

The hardware is augmented by the following software:

- o Flight software written in the AED programming language resident in the CAPS-6 computers;
- o The six degree of freedom aircraft model written in FORTRAN IV Plus, and executing in the PDP 11/60 in real time;
- o The DA11B interface software, written in MACRO 11, to provide a communications interface between the PDP 11/60, the PDP 11/04, and the CAPS-6 computers;

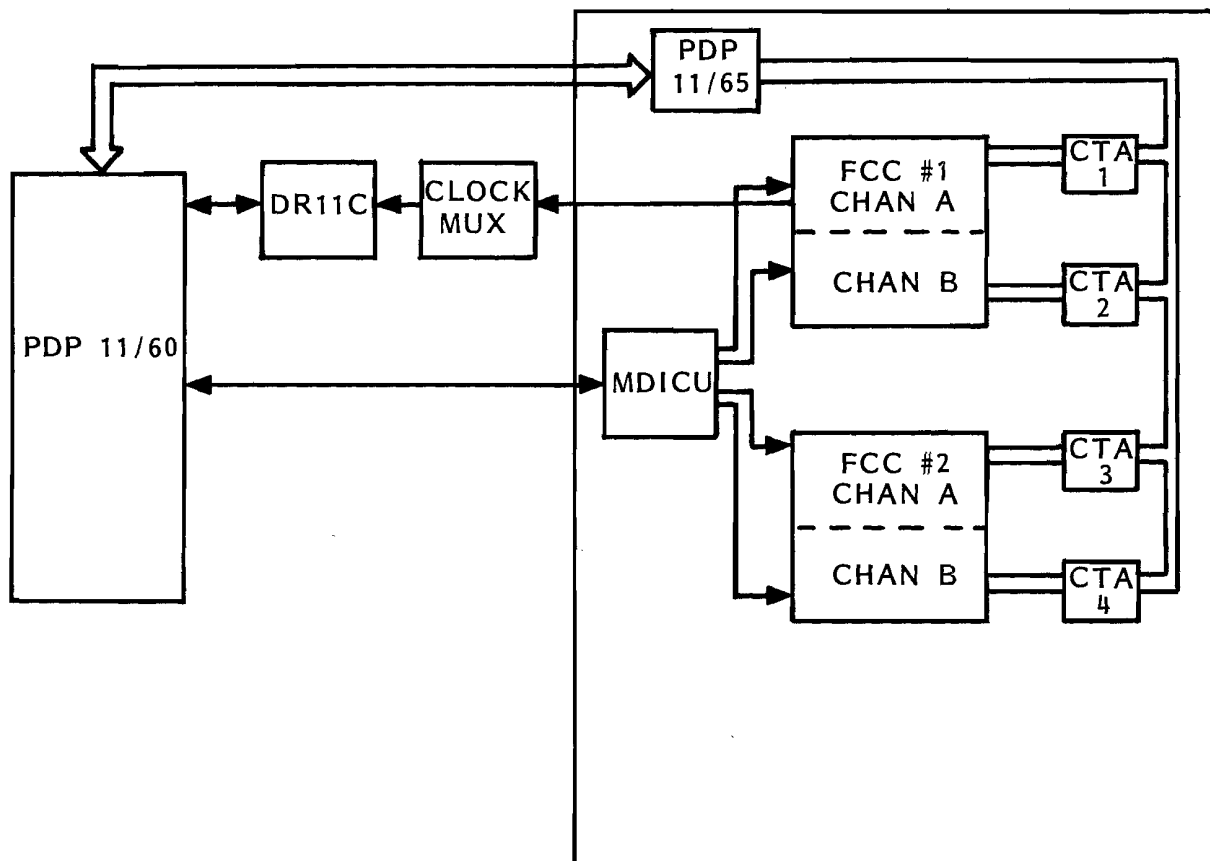


Figure 1

- o The MDICU software to provide real time signal transfer between the aircraft model in the PDP 11/60 and the flight computers.

The capabilities of the existing RDFCS and FIS hardware and software were utilized as much as possible to minimize the modification and additions to the system.

While the option selected by the FAA for implementation was a software modified system with no additional hardware, it was considered expedient by Lockheed to include a minimal amount of additional hardware to provide a real-time external clock and a forty-eight channel multiplexer to monitor comparator signals from the breakout panel on the RDFCS pallet. The input lines into the forty-eight multiplexer channels are terminated with RS232 receivers and can handle voltages originating at the breakout panel up to plus or minus fifteen volts, DC. The RS232 receivers then convert these voltages to standard TTL levels.

The combination of DR11C interface and multiplexer allows:

- o Detection of interrupts from the pallet indicating that a fault has been detected,
- o Determination of the incremental time between fault injection and fault detection using the external clock running at 1.2 khz.,
- o Monitoring of up to forty eight points on either the breakout panel or the back connector. This additional hardware greatly extended the data gathering capability of the system. It allows greater precision in time measurements of the interval between fault injection and fault detection. It also allows the detection of the states of a number of CAPS-6 hardware comparators from the breakout panel.

FIIS SYSTEM CAPABILITIES

The Draper FIS software has been augmented with additional software supplied by Lockheed Georgia to provide the capabilities for automating the fault injection and instrumentation system. The total system is now capable of:

- o Reading and passing commands to the FIIS subsystem
- o Initializing and running the CAPS processor
- o EXECUTING THE FIS COMMANDS
- o Reading the set-up configuration of the FIS and writing this information to a data file
- o Initializing and executing the software monitor in the PDP 11/04 to detect changes in the CAPS-6 software comparators
- o Halting the entire system in the event of a detected failure

- o Reading the incremental time between fault injection and fault detection from the external clock
- o Dumping all comparator values to the output data file

A number of software modifications were made to the existing RDFCS system to accommodate the gathering of data from the software comparators in the CAPS-6 computers. The PDP 11/04 downloaded task CNDL04 was extended and renamed FIS04. This new task performs all of the functions of CNDL04, and in addition provides for the monitoring of changes in the CAPS-6 software comparators. The initialization, start and stop of the comparator monitor is controlled from the PDP 11/60 via the subroutine "SET04" within the file "DRFIS". If any monitored CAPS-6 comparator changes state, the new state value and the iteration cycle in which the change occurred is recorded in a PDP 11/04 buffer. At the end of a failure sequence, this buffer can be transferred to the PDP 11/60. This comparator detection function requires that the PDP 11/04 continuously monitor the CTA (windows) into the CAPS-6 memories. Consequently, the standard PDP 11/04 keyboard monitor is no longer available when this activity is taking place.

The PDP 11/60 interrupt routine called "CONNECT" has been modified to accommodate the PDP 11/04 comparator monitor function, and has been renamed "CONFIS." CONFIS has primarily been changed to provide verification that the PDP 11/04 has accepted and responded to the extended requests from the PDP 11/60. CONFIS is an interrupt routine which remains in the wait state until a PDP 11/04 command is activated or a data transfer is required.

The interface call subroutine "DR11B" has been extended as "DR11BF" to handle the transfer of data between the PDP 11/60 and the PDP 11/04. The operation of this interface system is described in detail in Reference 3.

FIIS OPERATION

In addition to the above interface routines which provide communications between the PDP 11/60 and the PDP 11/04, three additional programs control the FIIS operation. The interaction of these three programs is shown in Figure 2.

- o FISCMD is a command file which controls the sequence of operations.
- o DR11C is an interface driver which accepts interrupts indicating that a failure has occurred and controls the timing of the external clock and the external multiplexer.
- o FISMON is a FORTRAN program which provides most of the initialization for the system and dumps both the setup and the results data to the output file on disk.

The following sequence of events describe the operation of the system.

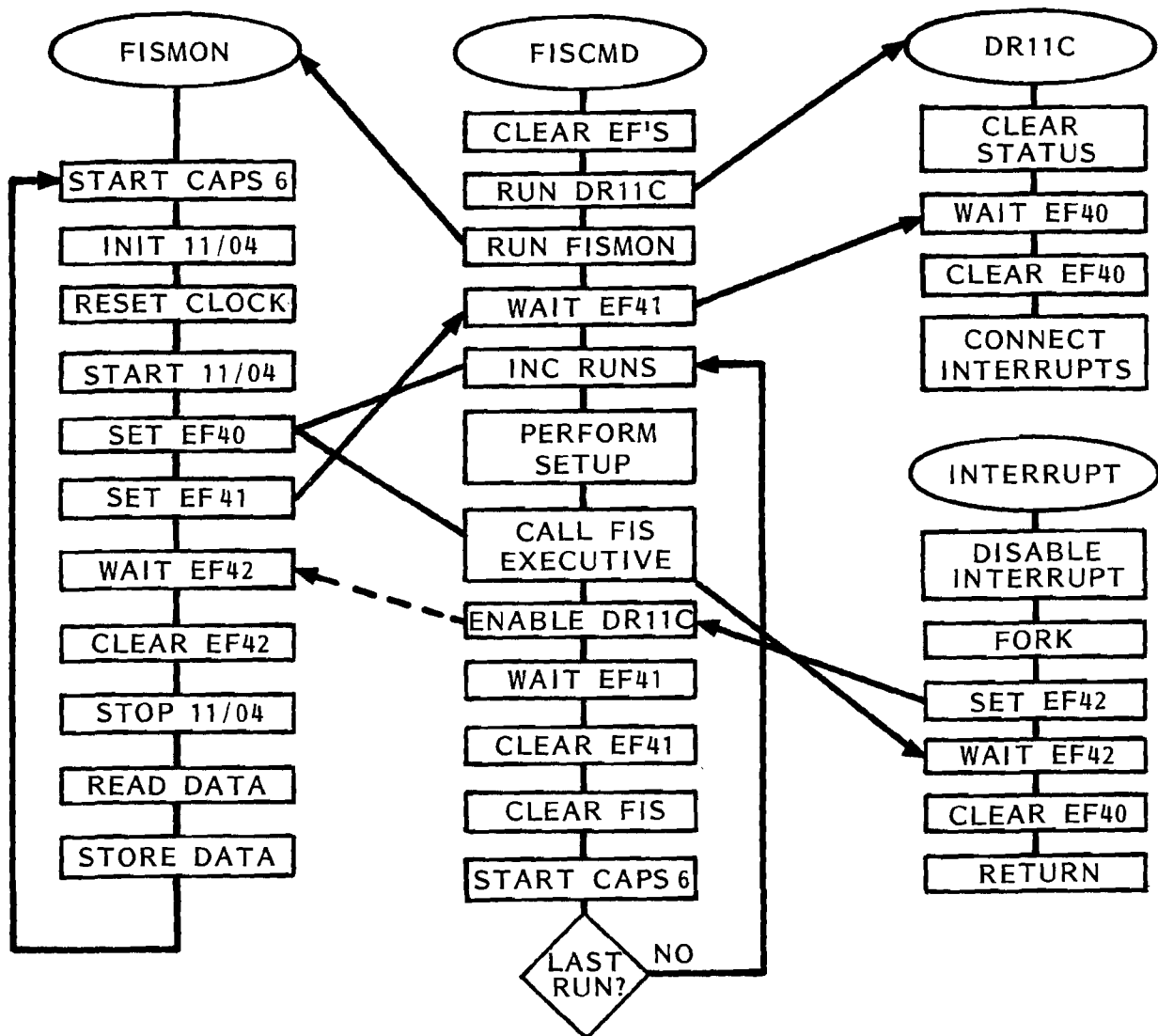


Figure 2

- 1) The FIIS system is started by executing FISCMD. FISCMD clears the Draper Fault Insertion Unit. It makes all appropriate connections so that the chip under test is in a normal operating mode. It then clears all of the event flags in the system. The event flags along with the external interrupt are used to synchronize the operation of the various software and hardware components. FISCMD then installs and runs Programs DR11C and FISMON and then goes into a wait state, waiting on Event Flag 41 to be set.
- 2) Program DR11C begins execution, clears the DR11C interface status register and goes into the wait state, waiting on Event Flag 40.
- 3) Program FISMON begins execution, reads the output file name, starts the CAPS-6 processor, starts the PDP 11/04 software comparator program, sets Event Flags 40 and 41 and then enters the wait state, waiting on Event Flag 42.
- 4) Program DR11C receives Event Flag 40 and enters the active state, connects the interrupt service routine to the DR11C interface external interrupt and again enters the wait state.
- 5) Program FISCMD receives Event Flag 41 and executes the Draper Fault Injection file "FIS." FIS performs the necessary actions to initialize the hardware fault insertion unit with the fault setup and then dumps the setup configuration to a temp disk file on "DMO". The fault is injected, the external clock started, and the DR11C interrupt enabled. FISCMD again goes into the wait state and waits on Event Flag 41.
- 6) As the fault propagates through the CAPS-6 processor, the DR11C interrupt service routine is activated in response to the failure detection in the CAPS-6, setting Event Flag 42.
- 7) FISMON receives Event Flag 42, stops the PDP 11/04 comparator monitor program, reads the comparator values from the PDP 11/04, reads the external clock and multiplexer values via the DR11C interface, and dumps the data to disk in the selected output file. FISMON then reinitializes and starts the CAPS-6 processor and PDP 11/04 software comparator programs, sets Event Flag 40 and 41, and goes into the wait state, waiting on Event Flag 42.
- 8) FISCMD receives Event Flag 41 and recycles at Step 5.

FIIS COMMANDS

When the Draper FIS program is executed from the FISCMD command file, it calls a user defined command file "FLT'N'.CMD" which designates the fault configuration for the fault injector unit. 'N' designates the sequence of command files i.e. 1, 2, 3,....., etc. The fault injector unit and the fault commands are described in detail in Reference 1.

However, these command functions will also be described here for clarification and ease of use. It will be helpful to refer to Figure 3, of Reference 1 to relate the commands to the fault insertion unit logic.

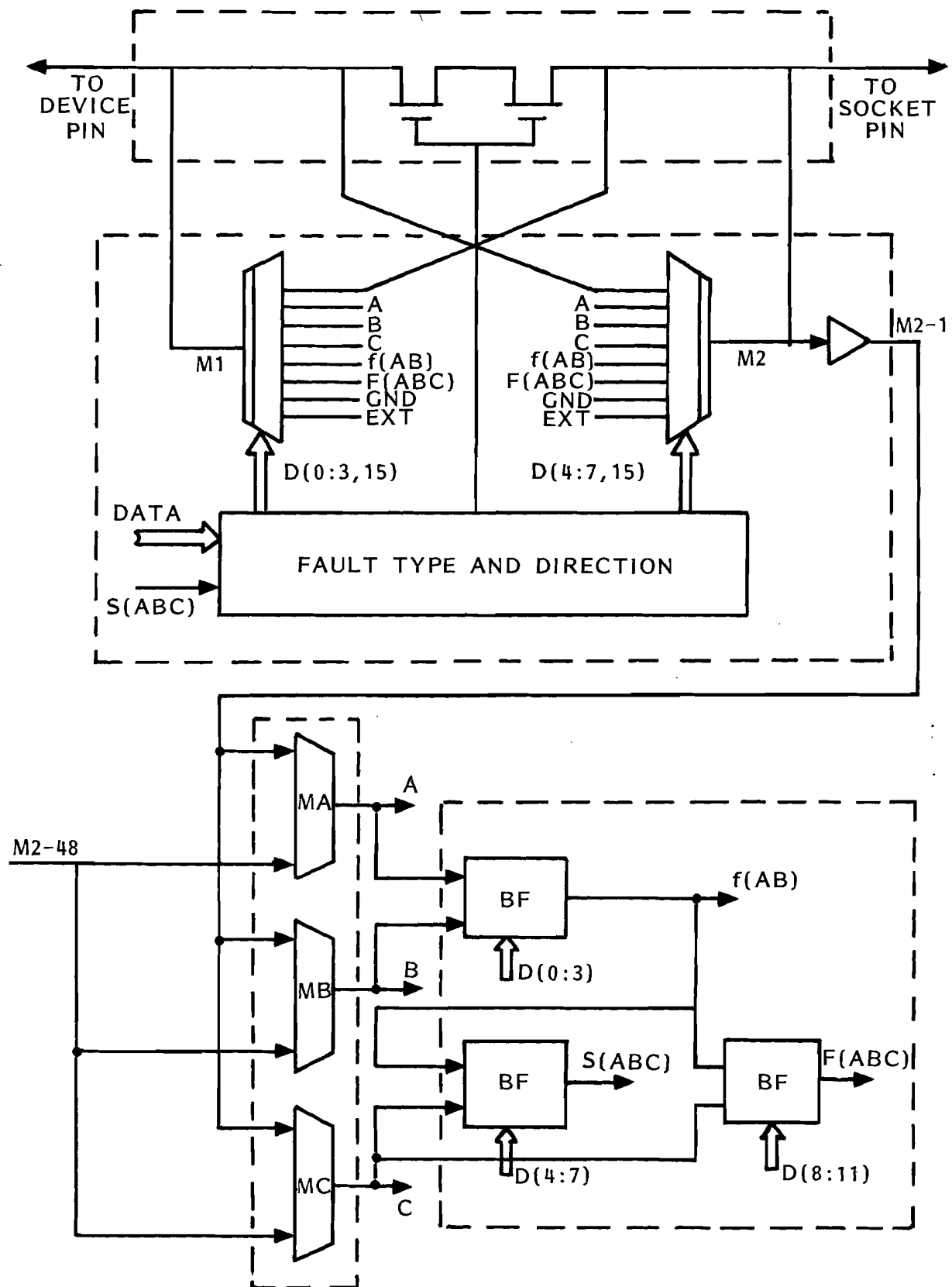


Figure 3 Fault Injector Logic

CLEAR clears the fault injector unit

DEFINE Unn m defines an m pin IC package at location Unn on the circuit board. The last defined package becomes the "active" package.

Ex: DEFINE U13 16 defines a 16 pin IC chip at location U13 on the board.

SELECT Unn if a number of IC packages are defined, SELECT can be used to name the "active" package.

MAP n z m l maps pin n of the active IC package to pin m of the zth implant segment. 1-1 subsequent IC package pins are mapped to 1-1 subsequent implant segment pins. z can be segments A, B, C, D, E, F.
Ex: MAP 9 B 1 4 maps pins 9, 10, 11, 12, of the active IC to pins B1,B2,B3,B4 of implant segment B

DESCRIBE n abcd defines the fault to be injected into pin n of the active package. abcd is a 16-bit hexadecimal number defining the fault as shown in the following format.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

1	E	0	0	1		X				Y			Z		

Field Z bits 0-3 Type of fault to device pin

Field Y bits 4-7 Type of fault to socket pin

Field X bits 8-11 direction of fault (device or socket)

The three fields are defined as:

X	Fault Direction	Y/Z	FAULT SIGNAL
0	To socket	0	Inverted signal
1	S(A,B,C)	1	F(A,B,C)
2	A	2	A
3	B	3	B
4	C	4	C
5	f(A,B)	5	f(A,B)
6	not used	6	1
7	not used	7	Ext.
8	To device	8	Original signal
9	S(A,B,C)	9	F(A,B,C)
A	A	A	A
B	B	B	B
C	C	C	C
D	f(A,B)	D	f(A,B)
E	not used	E	0
F	not used	F	Ext

When X is equal to 0, the Y field is used for fault designation and sends the fault to the socket.

When X is equal to 8, the Z field is used for fault designation and sends the fault to the IC device.

For faults that use the A, B, and C multiplexers or the boolean functions f(A,B), S(A,B,C) or F(A,B,C), one or more multiplexers and boolean function generators must be defined.

MUX n Unn m selects a pin m (1 to 48) on IC package Unn to drive multiplexer MA, MB, MC designated as n equal 1, 2, or 3 respectively. The IC package pin has already been designated by the MAP command.

FUNC abcd designates a boolean function of the format:

15	12 11	8 7	4 3	0
unused	F(A,B,C)	S(A,B,C)	f(A,B)	

where the boolean functions can be defined as:

DATA	f(A,B)	S(A,B,C)/F(A,B,C)
0	0	0
1	$\overline{A} \cdot \overline{B}$	$\overline{f(AB)} \cdot \overline{C}$
2	$A \cdot \overline{B}$	$f(AB) \cdot \overline{C}$
3	$\overline{B}$	$\overline{C}$
4	$\overline{A} \cdot B$	$\overline{f(AB)} \cdot C$
5	$\overline{A}$	$\overline{f(AB)}$
6	$A + B$	$f(AB) + C$
7	$\overline{A} + \overline{B}$	$\overline{f(AB)} + \overline{C}$
8	AB	$f(AB) \cdot C$
9	$\overline{A + B}$	$\overline{f(AB) + C}$
A	A	f(AB)
B	$A + \overline{B}$	$f(AB) + \overline{C}$
C	B	C
D	$\overline{A} + B$	$\overline{f(AB)} + C$
E	$A + B$	$f(AB) + C$
F	1	1

F(A,B,C) defines a boolean fault condition which can be applied to any of the mapped pins designated by the DESCRIBE command Y and Z fields.

S(A,B,C) defines a boolean control condition which can be applied to any of the mapped pins designated by the DESCRIBE command X field.

ENABLE n enables pin n of the active IC package.

DISABLE n disables pin n of the active IC package.

DUMP writes the injector test configuration to the terminal and also to a temporary disk file on DMO for later inclusion in the output file.

EXEC injects faults into enabled pins of the IC package.

EXIT exits from the FIS program.

As a synopsis of the logic setup commands,

"DESCRIBE" controls multiplexers M1 and M2;

"MUX" controls multiplexers MA, MB, MC;

"FUNC" controls the logic of the boolean function generators.

A typical command file can take the form:

```
CLEAR
DEFINE U17 40
MAP 1 A 1 8
MAP 9 B 1 4
MAP 17 B 5 4
MAP 21 C 1 8
MAP 33 D 1 8
DESCR 7 180E
DESCR 21 10E0
ENABLE 7
```

This command file yields a setup configuration of the form

A1 0000 U17	1	B1 0000 U17	9	C1 10E0 U17	21
A2 0000 U17	2	B2 0000 U17	10	C2 0000 U17	22
A3 0000 U17	3	B3 0000 U17	11	C3 0000 U17	23
A4 0000 U17	4	B4 0000 U17	12	C4 0000 U17	24
A5 0000 U17	5	B5 0000 U17	17	C5 0000 U17	25
A6 0000 U17	6	B6 0000 U17	18	C6 0000 U17	26
*A7 180E U17	7	B7 0000 U17	19	C7 0000 U17	27
A8 0000 U17	8	B8 0000 U17	20	C8 0000 U17	28
D1 0000 U17	33	E1 0000		F1 0000	
D2 0000 U17	34	E2 0000		F2 0000	
D3 0000 U17	35	E3 0000		F3 0000	
D4 0000 U17	36	E4 0000		F4 0000	
D5 0000 U17	37	E5 0000		F5 0000	
D6 0000 U17	38	E6 0000		F6 0000	
D7 0000 U17	39	E7 0000		F7 0000	
D8 0000 U17	40	E8 0000		F8 0000	

The IC package is a 40 pin unit located at position U17 on the circuit board. Pins 1 through 8 of implant segment "A" of the fault insertion unit are assigned to pins 1 through 8 of the IC. Pins 1 through 4 of implant segment "B" are assigned to pins 9 through 12 of the IC. Pins 5 through 8 of implant segment "B" are assigned to pins 17 through 20 of the IC. Pins 1 through 8 of implant segment "C" are assigned to pins 21 through 28 of the IC, pins 1 through 8 of implant segment "D" are assigned to pins 33 through 40 of the IC. Two faults are assigned; pin 7 to the IC is opened by the DESCR 7 180E command and pin 21 to the socket is opened by the DESCR 21 10E0 command. In this case the open fault to pin 7 is the only one enabled.

The fault can now be injected into the FIIS hardware by the "EXEC" command.

The fault initialization configuration and results are stored on DM0: as a file of the form FLT"m".DAT;"n". For repetitions of a given fault, "n" is incremented with each cycle. The following listing is an example FORTRAN program with data formats to recover the fault result data for further processing. The fault configuration is read into the six by eight by twenty two character buffer labeled "TABLE". The comparator values and associated count times in the PDP 11/04 are read into the sixteen word buffers MEMVAL, LCNT, HCNT. The multiplexed inputs from the breakout panel through the DR11C interface are read into MUX2, MUX3, and MUX4. The incremental clock count determined at the interrupt time is read into CLOCK. For formatting clarity, the program is written expanded with no attempt at combining any of the I/O functions

```

C      EXAMPLE PROGRAM FOR READING FIIS OUTPUT DATA FILE
C
      LOGICAL*1 TABLE(6,8,22)
      BYTE      FISNAM(40)
      BYTE      TMPCHR(60)
      INTEGER    CLOCK,MUX2,MUX3,MUX4
      INTEGER    MEMVAL,LCNT,HCNT
C
      DIMENSION MEMVAL(16)
      DIMENSION LCNT(16)
      DIMENSION HCNT(16)
C
      TYPE 1000
1000    FORMAT(X,'TYPE TWELVE CHARACTER FILENAME FOR OUTPUT',/,
1      X,' OF FORM XXXXXX.DAT;Y')
      ACCEPT 1010,FISNAM
1010    FORMAT(40A1)
      FISNAM(40) = 0
      OPEN(UNIT=9,NAME=FISNAM,TYPE='OLD',FORM='FORMATTED')
      READ(9,1100) TMPCHR
1100    FORMAT(X,60A1)
      WRITE(5,1100) TMPCHR
      READ(9,1100) TMPCHR
      WRITE(5,1100) TMPCHR
C
C
C      *****
C      *
C      * READ THE FIIS SETUP TABLE FROM THE OUTPUT DATA *
C      * FILE AND PLACE IN TABLE(6,8,22) *
C      *
C      *****
C
C
      DO 1110 IPIN = 1,8
      READ(9,1120)
1      (TABLE(1,IPIN,I1),I1=1,22),
2      (TABLE(2,IPIN,I1),I1=1,22),
3      (TABLE(3,IPIN,I1),I1=1,22)
1110    WRITE(5,1120)
1      (TABLE(1,IPIN,I1),I1=1,22),
2      (TABLE(2,IPIN,I1),I1=1,22),
3      (TABLE(3,IPIN,I1),I1=1,22)
C
C
1120    FORMAT(3(22A1))
C
C

```

```

DO 1130 IPIN=1,8
READ(9,1120)
1  (TABLE(4,IPIN,I1),I1=1,22),
2  (TABLE(5,IPIN,I1),I1=1,22),
3  (TABLE(6,IPIN,I1),I1=1,22)
1130 WRITE(5,1120)
1  (TABLE(4,IPIN,I1),I1=1,22),
2  (TABLE(5,IPIN,I1),I1=1,22),
3  (TABLE(6,IPIN,I1),I1=1,22)

C
C
C *****
C *
C * START READING FAULT INJECTOR TEST RESULTS *
C *
C * PLACE PDP 11/04 COMPARITOR VALUES IN MEMVAL *
C *
C * PLACE COMPARITOR COUNT IN LCNT, HCNT *
C *
C * PLACE DR11C MUX VALUES IN CLOCK, MUX2, MUX3 *
C * AND MUX4 *
C *
C *****
C
C READ 'FAULT INJECTOR TEST RESULTS' HEADING
C
C READ(9,1100) TMPCHR
C WRITE(5,1100) TMPCHR
C
C READ 'PDP 11/04 - CAPS COMPARATOR VALUES' HEADING
C
C READ(9,1100) TMPCHR
C WRITE(5,1100) TMPCHR
C
C READ PDP 11/04 - COMPARATOR BUFFER
C
C READ(9,1150) MEMVAL
C WRITE(5,1100) MEMVAL
1150 FORMAT(4012)
C
C READ ' PDP 11/04 LOW COUNTERS' HEADING
C
C READ(9,1150) LCNT
C WRITE(5,1150) LCNT
C
C READ 'PDP 11/04 HIGH COUNTERS' HEADING
C
C READ(9,1100) TMPCHR
C WRITE(5,1100) TMPCHR
C
C READ PDP 11/04 HIGH COUNTER VALUES
C

```

```

READ(9,1150) HCNT
WRITE(5,1150) HCNT

C
C   READ DR11C 'EXTERNAL CLOCK VALUE' HEADING
C
  READ(9,1100) TMPCHR
  WRITE(5,1100) TMPCHR

C
C   READ EXTERNAL CLOCK VALUE
C
  READ(9,1150) CLOCK
  WRITE(5,1150) CLOCK

C
C   READ DR11C 'EXTERNAL MUX CHANNEL 2' HEADING
C
  READ(9,1100) TMPCHR
  WRITE(5,1100) TMPCHR

C
C   READ DR11C MUX CHANNEL 2 VALUE
C
  READ(9,1150) MUX2
  WRITE(5,1150) MUX2

C
C   READ DR11C 'EXTERNAL MUX CHANNEL 3' HEADING
C
  READ(9,1100) TMPCHR
  WRITE(5,1100) TMPCHR

C
C   READ DR11C MUX CHANNEL 3 VALUE
C
  READ(9,1150) MUX3
  WRITE(5,1150) MUX3

C
C   READ DR11C 'EXTERNAL MUX CHANNEL 4' HEADING
C
  READ(9,1100) TMPCHR
  WRITE(5,1100) TMPCHR

C
C   READ DR11C MUX CHANNEL 4 VALUE
C
  READ(9,1150) MUX4
  WRITE(5,1150) MUX4
  CLOSE(9)
  CALL EXIT
  END

```

OPERATIONAL PROCEDURES FOR RUNNING FIIS WITH THE PALLET

SETUP:

Interrupt:

Connect a wire from P2A-PIN 61 on FCC 1 & 2 Breakout Panel. This wire goes through a diode/resistor network to PIN 24 - (CB0-CB15) on the multiplexer box.

Connect a wire from the low end of the resistor network to PIN 17 (CB0-cb15) for a ground.

Bat Handle:

Connect a wire from P1-PIN U of the glare shield breakout panel to PIN 1 (CB32) of the multiplexer box.

0 volts indicates Bat Handle down.

24 volts indicates Bat Handle up.

Connect a wire from the frame timer DAC to PIN 25 (CB00-CB15).
Set the CAPS TEST ADAPTER keyboard value to "OF6F".

Set the CAPS TEST ADAPTER address value to "OF76".
Set the CAPS TEST ADAPTER toggle switch to the SET/CLEAR mode.

Connect the following wires from the Breakout panel to the multiplexer box.

P2-A21	--	PIN 1 CB0	Split Mode Annunciator
P2-A51	--	PIN 2 CB1	Autopilot Disconnect Annunciator
P2-A49	--	PIN 3 CB2	No-Align Annunciator
P2-A26	--	PIN 4 CB3	No-Dual Annunciator

PDP 11/04-MDICU-FIS SETUP:

- 1) Turn on 60 cycle power , 400 cycle power , and 28 volts
- 2) Boot PDP 11/04
- 3) Turn on FCC 1 & 2 AC and DC power.
- 4) Turn on all memory.
- 5) Reset and start CAPS.
- 6) Turn rotary switch to MDICU.

7) Type M , GO , IC , EX ,MO for turning on MDICU.
or

Type M , ST , MO for turning off MDICU.

8) Turn rotary switch to PDP 11/04.

TYPE L 2000
S

9) Turn on FIS power switch

10) Turn on Multiplexer power switch.PDP 11/60 - PDP 11/04
DATA LINK:

From UIC [10,300] on the PDP 11/60 type

INS FIS04
FIX FIS04
RUN FIS04

Follow directions in setting up PDP 11/04, then type

ABO FIS04
REM FIS04
INS CONFIS
FIX CONFIS
RUN CONFIS

This establishes the data link between the PDP 11/60 and the PDP 11/04
via the DA11B interface.FIIS FILES AND TASK IMAGES:

FIIS files are in UIC [10,300]

Three standalone FIIS command files exist.

FISCMD.CMD - allows operation without the airplane so that only the
flight computers are working.

FISCE3.CMD - includes an L1011 6 DOF aircraft in the Approach
configuration.

FISCF6.CMD - includes an L1011 6 DOF aircraft in the Landing
configuration.

FIS command files are of the form FLT'N'.CMD

AIRCRAFT SOURCE FILES:

UIC [10,301] contains aircraft source files.

Because of incompatibility in the BLK1 Common Block, any changes to the aircraft should be performed on the Serial #1571567 disk titled FAA TRANSITION MODEL in UIC [10,301], and then transferred to the FIIS disk UIC [10,301].

Task images can then be transferred to UIC [10,300] for execution.
FIIS OPERATION:

Set address 6720H into CTA1-DAC 1.

Set the J Curve Pot on the Servo Simulator Panel to 0.76.

Start the FIIS program by typing the command file.

@FISCMD - for Autopilot only.

@FISCE3 - for the approach L1011 airplane.

@FISCF6 - for the landing L1011 airplane.

Wait for 7FFFH to be displayed on DAC 1 then raise the Bat Handle for FCC 1.

If the airplane is being used, the SAS switches on the MDICU Switch Panel should be switched off and back on before raising the Bat Handle.

The system will now cycle with the Bat Handle raised manually for each cycle.

FILE GLOSSARY:

BATHAN	Wait for bat handle on glare shield panel to be raised
CLEFnn	Clears Event Flag nn
ENDRIB	Enable DR11C interrupt B
FISACT (FISINJ)	FIS program modified to set CSR1 (Bit 1) in DR11C Status register
FISEXX	FIS program modified to enable Int B in DR11C
FISTIM	Command file to determine latency time of FIS EXECUTE command
RSETRY	Set Location 6720H in CAPS6 #1 to 0
RCSR1	Reset CSR1 (Bit 1) in DR11C Status register
SCSR1	Set CSR1 (Bit 1) in DR11C Status register
SEFnn	Set Event Flag nn

SETFIS Allows control of the PDP 11/04 FIS Monitor from the PDP 11/60
 SETRY Set Location 6720H in CAPS6 #1 to 7FFFH
 STARTC Resets and starts CAPS6 #1
 ST04FI Stop monitoring the FIS comparators in PDP 11/04
 TIMOUT Monitor external clock to timeout and set EF 43 if fault
 interrupt does not occur
 WTEFnn Wait on Event Flag nn

FISMON/DRFIS

GFLNAM Get name for output file
 INICAP Reset CAPS6 #1 and put into RUN mode
 INITCK Initialize the external clock using CSR0 (Bit 0)
 in the DR11C Status register at location 767770
 OUTDM Read fault injector test configuration and test results ,
 then dump to disk
 RDFI04 Read the buffers MEMVAL, LCNT, HCNT in the PDP 11/04
 RDMUX Read four channels of DR11C mux box and place values in
 FISBUF 49 through 52
 SET04 Control FIIS monitoring in the PDP 11/04
 STOPCK Stop the external clock using CSR1 (Bit 1) = 0 in the
 DR11C Status register at location 767770
 SETFST Sets Software Monitor in the PDP 11/04 to RUN.

INSTRUCTION SELF TEST

The Special Instruction Self Test program was run without the autopilot. Only CAPS #1 was used.

A monitor wire was run from a CTA #1 DAC set to location "40F7" into
 CB15(PIN25) of the DR11C multiplexer. CAPS location "40F7" yielded a "1"
 if INSTRUCTION SELF TEST (IST) was executing, or a "0" if in setup.
 This bit was then monitored through the DR11C interface by program
 TIMIST.

FAULT INJECTION UNIT PIN MAPPING

The Fault Injection Unit was mapped using the following chip configurations. For each chip type, the socket layout is shown along with the Fault Injector mapping commands

A total of 4992 fault cases were run at NASA Ames during the FIIS study. These can be subdivided into the following:

Open loop	2736
Self test	1728
Closed loop	528

The closed loopcases can be further subdivided into eighty eight fault runs in each of the following catagories

- 1) Cruise altitude hold condition
- 2) Cruise climb condition
- 3) Cruise turn condition
- 4) Localizer capture condition
- 5) Glideslope capture condition
- 6) Glideslope deviation condition

These fault files were then sorted, interpreted and combined into 2728 fault files for analysis. Program SORTFS was used for this sorting task. SORTFS opened each individual failure file, determined its recorded characteristics, and combined it with similar files of the same fault. The combined data were then written as records to files F0001.dat through F2728.dat corresponding to each fault.

Where faults were detected by the idle loop in the CAPS-6, they were written to a file containing 41 records. When the fault was undetected, closed loop runs were made recording data for each of the six closed loop flight cases for a total of 215 records.

Using the sorted raw data, histograms were run on both the open loop and closed loop conditions to determine a spread of fault detection times at 50 millisecond intervals between 0 and 15 seconds. The histogram data was run using programs HISGRM.FTN, HISCCC.FTN, HISCAC.FTN, HISCTC.FTN, HISLCC.FTN, HISGSC.FTN, HISGSD.FTN .

Fault counts for the conditions listed in the FIIS FAULT COUNT table were obtained with program FAULT.FTN .

Programs used for the analysis of the FIIS data:

CHIPFL.DAT - data file containing chip characteristics generated by program CHIPFM.FTN.

CHIPFM.FTN - generates a table describing the chip characteristics for the FIIS data reduction.

8		1	8	E	5	8	C
40	D	33	32		29	28	21
AM2901							
1		8	9		12	13	20
1	A	8	1	E	4	1	B 8

MAP 1 A 1 8

MAP 9 E 1 4

MAP 13 B 1 8

MAP 21 C 1 8

MAP 29 E 5 4

MAP 33 D 1 8

40 PIN FAULT INJECTOR SETUP

8		B		1	
18	17			10	
C1, C3, C4, C5, C6, C7, C9					
C10, C11, C12, C13, C17, C18					
1				8	9
1	A		8		

MAP 1 A 1 8

MAP 10 B 1 8

18 PIN FAULT INJECTOR SETUP

6	C	5	8	B	1
20	19	18			11
			C14, C15, C16		
1	V _{CC}	2		7	Gnd
				8	10
2	A			8	1
					C
					3

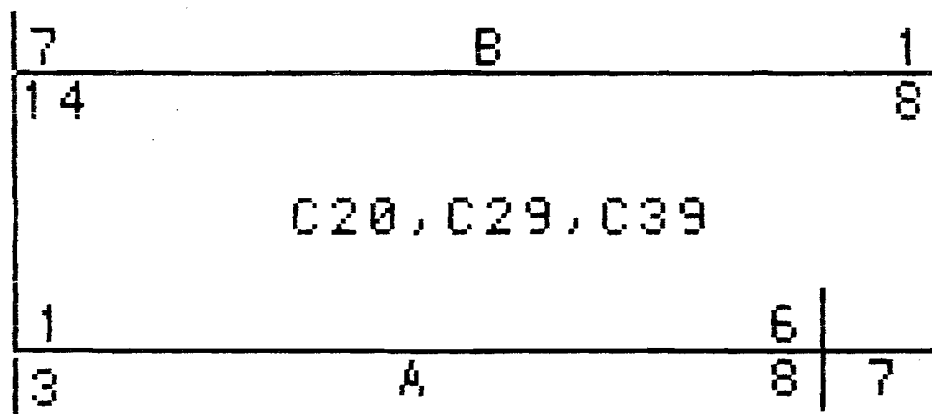
MAP 1 A 2 7

MAP 8 C 1 3

MAP 11 B 1 8

MAP 19 C 5 2

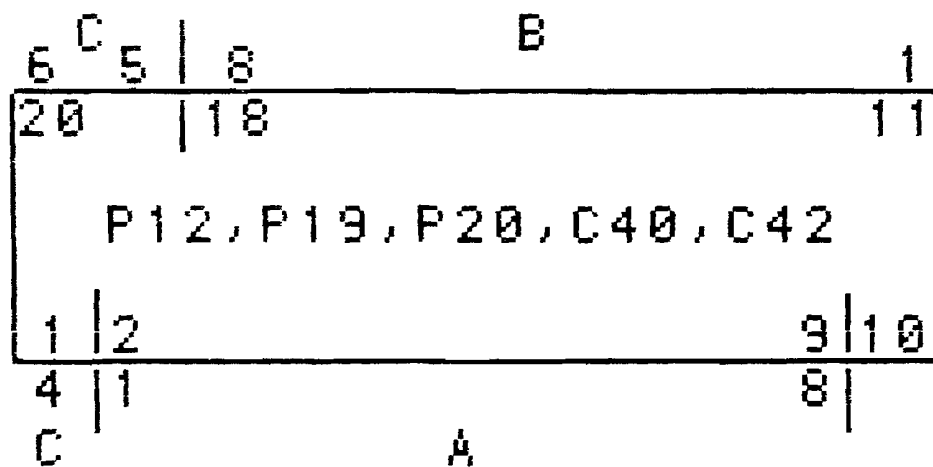
NON-STANDARD 20 PIN
FAULT INJECTOR SETUP



MAP 1 A 3 6

MAP 8 B 1 7

14 PIN FAULT INJECTOR SETUP



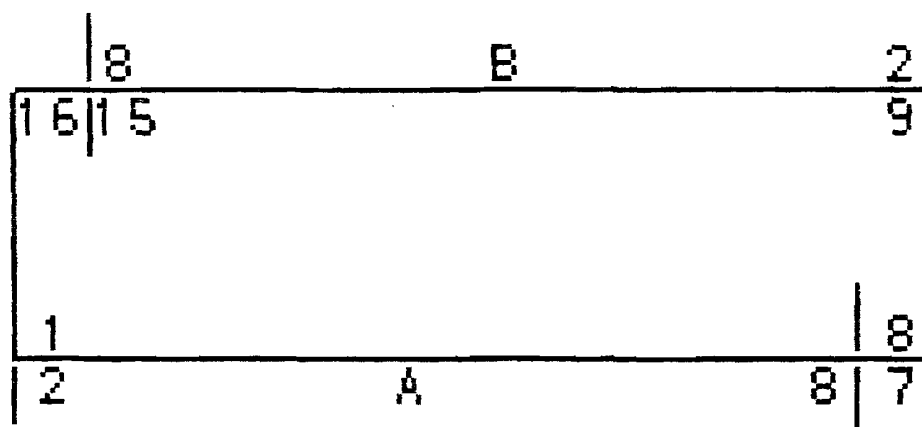
MAP 1 C 4 1

MAP 2 A 1 8

MAP 11 B 1 8

MAP 19 C 5 2

STANDARD 20 PIN
FAULT INJECTOR SETUP



MAP 1 A 2 7

MAP 9 B 2 7

16 PIN FAULT INJECTOR SETUP

FAULTS.FTN - uses the FIIS fault data to determine the following conditions for both weighted and unweighted faults:

- 1A) Number of total faults detected by the baseline CAPS-6 program (Idle Loop)
- 1b) Number of total faults detected by the CAPS-6 Self Test
- 2A) Number of total faults not detected by the baseline CAPS-6 program (Idle Loop), but detected by the CAPS-6 closed loop program running with the aircraft model on the PDP-11/60
- 2B) Number of faults detected by the CAPS-6 closed loop program running with the aircraft model, and also detected by the CAPS-6 Self Test.
- 3A) Number of total faults not detected by the baseline CAPS-6 program (Idle Loop), and not detected by the CAPS-6 closed loop program running with the aircraft model.
- 3B) Number of faults from 3A detected by the CAPS-6 Self Test.

FISFUN.FTN - a subroutine called by SORTFS.FTN, used to locate the following FIIS function characteristics in the data table , i.e.

- Pin Identification
- Component Identification
- FIIS Function Designation
- Fault Direction
- Fault Type

GETCL.FTN - transfers selected closed loop raw data files from tape to disk 1, UIC [10,303].

GETCMP.FTN - subroutine called by SORTFS, breaks out bits from the comparator at CAPS address FB\$B, from MUX2, from the clock and from the Iteration Monitor to determine the state of discrete inputs and the time between fault insertion and fault detection.

HISGRM.FTN - generates latency histograms for the total faults, i.e., the proportion of faults verses time to detect, with the time axis divided into 50 millisecond intervals.

HISCAC.FTN - generates histograms for the closed loop, cruise altitude hold condition.

HISCCC.FTN - generates histograms for the closed loop, cruise climb condition.

HISCTC.FTN - generates histograms for the closed loop, cruise turn condition.

HISGSC.FTN - generates histograms for the closed loop glide slope capture condition.

HISGSD.FTN - generates histograms for the closed loop glide slope deviation condition.

HISLCC.FTN - generates histograms for the closed loop, localizer capture condition.

RDATST.FTN - subroutine called by SORTFS to open the self test data files and get the raw self test data recorded at AMES.

RDCLP.FTN - subroutine called by SORTFS to form the closed loop file address with the designation (A, C, T, L, G, H) from the variable RUNCHR and the file name in AUXFIL.

RDDIR.FTN - open and read the directory file for DM!:[10,303] containing the closed loop FIIS data recorded at AMES.

RDATFL.FTN - subroutine called by SORTFS to open Ames Data files and return setup and basic data.

SOROUT.FTN - subroutine called by SORTFS to output sorted FIIS data to a direct, sequential, formatted, fixed record type file named Fnnnn.DAT corresponding to each fault type. A record number is designated for each record.

SORTFS.FTN - the main sort program, sorts the FIIS data and places each fault condition into a separate file with the identification Fnnnn.DAT .

STNAM.FTN - subroutine called by SORTFS to form the Self Test file name from variable AUXFIL and return the name in AUXFIL.

REFERENCES

1. Lala, J., and T.B. Smith: "Development and Evaluation of a Fault-Tolerant Multiprocessor (FTMP) Computer - Volume III, FTMP Test and Evaluation," CSDL-R-1602, November 1982.
2. Mulcare, D.B., et al.: "Hardware Fault Insertion and Instrumentation System (FIIS) Definition Study," DOT/FAA/CT-83/32, June 1983.
3. Benson, J.W. Benson: "PDP 11/60 - CAPS Test Adapter Data Link," Contract Report on NAS2-10270, June 1980.

