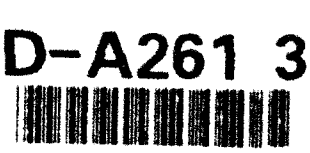


(2)

DOT/FAA/CT-83/32

Hardware Fault Insertion and Instrumentation System (FIIS) Definition Study

D.B. Mulcare, J.W. Benson, R.M. Davis
W.G. Ness, and M.L. Roginsky
Lockheed-Georgia Company
Marietta, Georgia 30063

H.M. Youssef
Lockheed-California Company
Burbank, California 91520

Donald Eldredge
FAA Technical Center
Atlantic City Airport, New Jersey 08405

W.E. Larsen
FAA Technical Field Office
Moffett Field, California 94035

June 1983

US Department of Transportation
Federal Aviation Administration
Technical Center
Atlantic City Airport, N.J. 08405

NOTICE

This document is disseminated under the sponsorship of the U.S. Department of Transportation in the interest of information exchange. The United States Government assumes no liability for the contents or use thereof.

The United States Government does not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the objective of this report.

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
DOT/FAA/CT-83/32		

4. Title and Subtitle
Hardware Fault Insertion and Instrumentation System Definition Study

5. Report Date

6. Performing Organization Code

7. Author(s) D.B. Mulcare, W.G. Ness, H.M. Yosef, J.W. Benson, M.L. Roginsky, D. Eldredge, R.M. Davis, W.E. Larsen

8. Performing Organization Report No.

9. Performing Organization Name and Address
Lockheed-Georgia Company
86 South Cobb Drive
Marietta, Georgia 30063

10. Work Unit No. (TRAIS)

11. Contract or Grant No.
NAS2-11511

12. Sponsoring Agency Name and Address
Federal Aviation Administration
Technical Center
Atlantic City, New Jersey 08405

13. Type of Report and Period Covered
Contractor Report

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 report presents descriptions and critiques of several low-level hardware fault insertion and instrumentation schemes for potential application in a digital flight control system (DFCS) simulator. Representing varying degrees of sophistication, these schemes are tailored to enhance test validity and productivity. Particular attention is therefore directed toward the capabilities offered by the various schemes, as well as their proper utilization. Factors such as fault detection coverage, latency time, and recovery from transients are stressed. Also, the role of the FIU is examined in detail. This tends to emphasize low-level fault injection, such as that on a chip pin level. Such testing should prove valuable despite the trend toward VLSI (Very Large-Scale Integration) circuits because correlation of present chip-versus-card fault observability may be useful in test case definition for VLSI implementations.

Prepared under NASA Contract NAS2-11511, this study has been funded and technically supported by the Federal Aviation Administration.

17. Key Words	18. Distribution Statement
Reliability Fault Monitoring Digital Flight Control Systems Failure Monitoring	Unlimited Subject Category 38

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified		

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

FOREWORD

This report presents descriptions and critiques of several low-level hardware fault insertion and instrumentation system (FIIS) schemes for potential application in a digital flight control system (DFCS) simulator. Representing varying degrees of sophistication, these schemes are tailored to enhance test validity and productivity, especially in assessing DFCS fault detection mechanisms with regard to coverage and latency times. Particular attention is therefore directed toward the capabilities offered by the various schemes, as well as their coordinated utilization to enable overall coverage measures.

Prepared under National Aeronautics and Space Administration (NASA) Contract NAS2-11511, this study has been funded, directed, and technically supported by the Federal Aviation Administration (FAA). Additionally, the ultimate objectives of this study have been significantly fostered by recent simulator test facility enhancements by the NASA-Ames Research Center (ARC). The intent of all study participants has been to address certain vital certification technology issues in a responsive and definitive manner.

This report has also been published as Lockheed-Georgia Company Engineering Report No. LG83ER0087.

TABLE OF CONTENTS

TABLE OF CONTENTS Cont'd

LIST OF FIGURES

LIST OF TABLES

1.0 SUMMARY

Regulatory needs of the FAA have been assessed with regard to fault survivability of critical digital systems, and remedial facilities and investigations have been defined. The assessment is largely based on requirements deriving from FAA Advisory Circular No. (AC) 25.1309-1 (Ref. 1), and the investigations are based on current or projected capabilities in the RDFCS (Reconfigurable Digital Flight Control Systems) Facility at NASA-ARC, especially those of recently installed fault injection unit (FIU).

This study surveys the various types of fault detection mechanisms used in DFCSs to determine the occurrence of a hardware fault, and detailed consideration is given to various test schemes to evaluate their acceptability. Factors such as fault detection coverage, latency time, and recovery from transients are stressed. Also, the role of the FIU is examined in detail. This tends to emphasize low-level fault injection, such as that on a chip-pin level. Such testing should prove valuable despite the trend toward VLSI (very large-scale integrated) circuits because correlation of present chip-versus-card fault observability may be useful in test case definition for VLSI implementations.

In any circumstance, as more definitive and conclusive test results are sought, greater consideration must be accorded to instrumentation to observe the sequences of elemental events issuing from the injected faults(s). Such instrumentation ideally should encompass hardware and software, multiple computer channels, and overall time correlation. Adequate capacity must also exist to assimilate, interpret, and store the associated test data to properly realize the benefits of automated testing.

Although the recently installed FIU adds substantially to the RDFCS facility, the overall low-level test capability is adjudged to lack suitable instrumentation. Several approaches to correcting this deficiency are therefore offered, but all priorities considered, the most prudent course now is to systematically develop and apply the basic capability enabled by the FIU. This is supported and amplified by the investigation plan herein. Three additional levels of facility upgrading, including full FIIS capability, are also defined, along with a description of the additional classes of investigations thereby enabled.

1.1 EXECUTIVE SUMMARY

Although the recently added low-level fault injection capability in the RDFCS laboratory at NASA-ARC is both extensive and usable, certain instrumentaion enhancements are needed to complement the FIU and enable precise investigation of the fault tolerance mechanisms. Also, certain software additions or modifications to the existing RDFCS facility can substantially improve the productivity and quality of investigations. This study, which constitutes the first attempt to address such needs, has resulted in the definition of four levels of FIIS capability based on the composition and constraints of the existing RDFCS facility. The four FIIS configurations are summarized in Section 1.4, and Tables 1 and 2 provide some associated background.

TABLE 1. FIIS TECHNOLOGY ORIENTATION

LEVEL / ASPECT	OVERALL ASSURANCE	TESTING IN GENERAL	TESTING OF FLIGHT CONTROL COMPUTERS
PROBLEM	HIGH ASSURANCE LEVELS FOR CRITICAL DFCSs	INTRACTABILITY OF THOROUGH TESTING	IMPACT OF ELEMENTAL COMPUTER HARDWARE FAULTS
COMPLICATION	FAULT TOLERANCE COMPOUNDING OF ASSURANCE TASKS	LARGE NUMBER OF FAULT CASES TO BE IDENTIFIED AND APPLIED	SENSITIVITY TO TEST ENVIRONMENT
TECHNOLOGY ISSUE	COMPLEMENTARITY OF ASSURANCE METHODS	TEST CASE DESIGN AND INTERPRETATION	VALID, OBSERVABLE, AND EFFICIENT LOW-LEVEL TESTING
FIIS EMPHASIS	REAL-TIME TEST CONFIRMATION OF FAULT TOLERANCE	DEPENDABLE AND PRODUCTIVE TESTING	EXPLOITATION OF EXISTING FACILITIES (e.g., FIU)

1.2 GENERAL PROBLEM

The general problem addressed in this study is that of defining FIIS configurations that to some useful extent meet the following requirements within the context of the RDFCS facility and the existing FIU:

- Non-interference with real-time RDFCS operation
- Arbitrary automated control of fault insertion/removal
- Low-level hardware and software instrumentation

TABLE 2. FIIS STUDY CONCLUSIONS

LEVEL / ASPECT	OVERALL ASSURANCE	TESTING IN GENERAL	TESTING OF FLIGHT CONTROL COMPUTERS
PROBLEM	IMPROVED ASSURANCE METHODS AND PRACTICES MANDATORY TO CONFIRM HIGH ASSURANCE LEVELS	TEST PRODUCTIVITY VITAL TO APPLICATION OF THOROUGH SET OF TEST CASES	PRESENT FIU LACKS RESOLUTION IN RDFCS INSTALLATION FOR NEEDED INVESTIGATIONS
COMPLICATION	INCREASED EFFORT AND RESOURCES NECESSARY TO COPE WITH INCREASED COMPLEXITY & FAULT CASES	ARBITRARY CONTROL OF FAULT REMOVAL, HEALING, AND INSERTION NOT CURRENTLY AVAILABLE	REAL-TIME SIMULATOR ROLE OR ACCEPTABILITY OF LOW-LEVEL TESTING NOT YET ESTABLISHED
TECHNOLOGY ISSUE	CONCLUSIVENESS OF LOW-LEVEL TESTING USING PRECISE MODELS AS EXECUTION MONITORS	DEFINITION AND IMPLEMENTATION OF COMPREHENSIVE TESTING	RECOMMENDED TEST SPECIMENS AND INSTRUMENTATION ENABLE ADVANCED METHODS AND CAPABILITIES
FIIS EMPHASIS	HIGH FIDELITY FAILURE EFFECTS RESULTS TESTING USING REAL-TIME SIMULATOR ENVIRONMENT	RECOMMENDED FIIS ARCHITECTURES EMPHASIZE AUTOMATED NON-INTERFERENCE TESTING	OPTIMIZATION OF FIIS ARCHITECTURES BASED ON EXISTING FACILITY CONSTRAINTS

- o Multiple RDFCS channel observations
- o Correlated data retrieval/storage
- o Efficient test loop operation.

Since the extent to which these requirements are satisfied is dependent upon resources expended, it is appropriate to delineate several increments of cost/capability. Accordingly, four separate configurations have been defined:

- o Existing system with only software modifications
- o Improved system based on modest hardware modifications and appropriate software changes
- o Advanced system based on extensive modifications
- o Superior system based on the full scope of feasible modifications.

To motivate and substantiate these FIIS configurations, associated simulator investigation plans have also been formulated. These plans serve to indicate how FIIS capability can aid certification technology and to identify costs/benefits tradeoffs in upgrading the present RDFCS facility.

1.3 SPECIFIC PROBLEMS

From the outset certain specific problems have been recognized as highly important to the outcome of the study. These concerns are based on familiarity with testing of digital flight systems in general and the operation of the RDFCS facility in particular. Included among these concerns are the following:

- Autopilot Disconnect - a large number of computer hardware fault insertions result in autopilot disconnect, which owing to the need for manual reset, inhibits automated testing
- Flight Computer Memory Volatility - a significant number of computer hardware fault insertions result in eradication of the flight program in the core memory, thereby necessitating reloading prior to the continuation of testing
- Fault Introduction Phasing - there exists no way to precisely control the introduction of faults relative to the flight software execution runstream
- Analog Data Digitization - some of the essential test data are not available in digitized form for the PDP-11/60 computer
- Massive Test Results Data - efficient and highly observable testing generates real-time test results processing demands to alleviate storage-related problems
- PDP-11/04 Limitations - because of its slowness and lack of flexibility, the PDP-11/04 impedes the full realization of FIIS capability
- Multiple Channel Monitoring - the PDP-11/04 can only access one flight computer channel at a time, an impediment to precise testing that may be aggravated by channel skewing and transport lags
- Time Correlation - there exists no universal time base to correlate events in different channels or various parts of the test loops
- Instrumentation Limitations - many of the foregoing points are among the causes of a fundamentally inadequate instrumentation capability to support certain basic types of low-level investigations.

It should be noted that all of these problems result from trying to use a system simulator for high-resolution, low-level testing, or something

other than what it was actually optimized for. While there is clearly merit in the high fidelity examination of low-level faults as is possible during real-time system operation, the full implications of this were not at issue during the RDFCS development contract. Beginning with the definition of the associated requirements, this study has undertaken to resolve the attendant problems and to maximize FIIS capability.

1.4 RECOMMENDATIONS

A family of four FIIS architectures is summarized in Table 3 along with several miscellaneous recommendations for facility improvement. The four architectures are differentiated by the expense involved in their implementation and by the failure effects investigation capabilities thereby provided. To enable an initial phase of such investigations, the first FIIS architecture is recommended for appreciably extended capability at modest cost. The associated implementation experience would also permit lowered risk realization of the other options. Rather conveniently, the miscellaneous recommendations might be added as desired during any phase.

TABLE 3. FIIS MECHANIZATION RECOMMENDATIONS

IMPACT / OPTION	PDP-11/60	PDP-11/24	FCCs	OTHER
SOFTWARE MODIFIED SYSTEM	• UPGRADE PDP 11/24 LINK • DEVELOP FIIS EXECUTIVE • DEFINE RESULTS PROCESSING	• DEVELOP CAPS MEMORY MONITOR • UPGRADE DAAA LINK	• EXECUTIVE/ MONITOR PROGRAM FOR BACK-GROUND MODE • GENERATE INTERRUPT TO PDP 11/60	• UTILIZE MONITOR TO START/STOP/RESET FCCs
MODEST SYSTEM MODIFICATIONS	• CONTROL PROGRAM FOR BUS MONITOR/ RECORDER UNIT			• ADD BUS MONITOR/ RECORDER UNIT
EXTENSIVE SYSTEM MODIFICATIONS	• INTERFACE TO PDP 11/24 • UPGRADE AIRCRAFT MODEL	• BY-PASS PDP 11/24 WITH PDP 11/24		• ADD PDP 11/24 AND SYSTEM CLOCK • ADD BUS MONITOR/ RECORDER UNIT
FULL-SCOPE SYSTEM MODIFICATIONS	• ADD EMULATOR PROGRAM • INTERFACES TO NEW DEVICES	• BY-PASS PDP 11/24 WITH PDP 11/24	• SYNCHRONIZE CAPS OPERATION	• ADD PARALLEL-I/O UNIT • ADD LOGIC STATE RECORDER
MISCELLANEOUS	• ADD ARINC INTERFACE TO FCCs		• WRITE PROTECT THE FCC CORE MEMORY UNITS • INHIBIT A/P DISCONNECT FOR AUTOMATED TESTING	• ADD "ONE-SHOT" TO CONTROL FIU FOR TRANSIENT TESTING

Basically, the benefits of the various options range from increased test productivity for the initial modifications to extended test significance and resolution for the full-scope modifications. Both aspects are highly important, but the most crucial assurance technology issues focus on the need for responsive high-resolution testing to investigate transient phenomena. As a consequence, it is appropriate to proceed with a multiphase implementation of the Table 3 recommendations, or refinements thereof. Note that these modifications to the existing facility are deceptively difficult to accomplish without close familiarity with the overall RDFCS simulator implementation details.

2.0 BACKGROUND

As far as digital flight system failure modes and effects are concerned, the apprehensions expressed at the Government/Industry Workshop on Methods for Certification of Digital Flight Controls and Avionics in 1976 (Ref. 2) have proven to be largely warranted. This is not a general indictment of digital implementation, but recognition of the tendencies inherent in the increased complexity of digital over analog mechanization. This complexity, which becomes quite evident in fault case definition, tends to mask design and implementation discrepancies.

Much of this complexity relates to software, but in this study only hardware faults, and not software discrepancies, are of direct concern. Since software procedures are often used to detect or isolate hardware faults, attention is ultimately focused on the adequacy of such software. The delineation between hardware and software, moreover, is sometimes barely distinguishable, and this is a particularly significant aspect of digital flight systems. This phenomenon is addressed and reflected by test validity requirements that encourage low-level fault insertion in a high-fidelity environment, or in the case at hand, a real-time system simulator.

2.1 ABBREVIATIONS

AC	Advisory Circular
ADC	Analog-to-Digital Converter
AFCS	Automatic Flight Control System
AIRLAB	Avionics Integration Research Laboratory (at NASA LaRC)
ARC	Ames Research Center (NASA)
AWI	AFCS Warning Indicator
BIT	Built-in Test
BMRU	Bus Monitor/Recorder Unit
CAPS	Collins Adaptive Processor System

CPU	Central Processor Unit
CTA	CAPS Test Adaptor
DAC	Digital-to-Analog Converter
D/D	Digital-to-Discrete
DEC	Digital Equipment Corporation
DFCS	Digital Flight Control System
DMA	Direct Memory Access
FAA	Federal Aviation Administration
FCC	Flight Control Computer
FD	Flight Director
FI	Fault Injector
FIFO	First-in/First-out
FIIS	Fault Insertion and Instrumentation System
FIU	Fault Injection Unit
FTMP	Fault-Tolerant Multiprocessor (Draper)
HZ	Hertz
IRAD	Independent Research and Development
IC	Integrated Circuit
I/O	Input/Output
K	Thousand
LaRC	Langley Research Center (NASA)
LSR	Logic State Recorder
MDICU	Modular Digital Interface Control Unit
msec	Millisecond
NASA	National Aeronautics and Space Administration
NPR	Non-Processor Request

PROM	Programmable Read-Only Memory
RAM	Random Access Memory
RDFCS	Reconfigurable DFCS (at NASA-Ames)
ROM	Read-Only Memory
SAS	Stability Augmentation System
VLSI	Very Large-Scale Integrated
μsec	Microsecond

2.2 TERMINOLOGY

By defining and elaborating on the use of key terms at the outset, it is hoped that the ensuing issues, concepts, and recommendations will be rendered more accessible and meaningful. In addition to the following, other terms defined in AC No. 25.1309-1 (Ref. 1) and the FAA Validation Handbook (Ref. 3) are quite important.

A CRITICAL FUNCTION is one whose availability is necessary to ensure the safe flight and landing of an aircraft. Therefore, failure conditions that can result in the loss or appreciable degradation of a critical function must be extremely improbable, or of an incidence rate of 1.0×10^{-9} per hour of flight or less.

An ESSENTIAL FUNCTION is one whose availability is necessary to ensure the basic safety and flyability of an aircraft under all operating conditions, even the most adverse. Therefore, failure conditions that can result in the loss or significant degradation of an essential function must be improbable, or of an incidence rate of 1.0×10^{-5} per hour of flight or less.

A HARDWARE FAULT is the anomalous behavior resulting from an elemental physical event, which may be due to a transient malfunction or a permanent impairment of hardware. Depending on the implementation, certain faults cannot affect the performance of the system function(s), and these are referred to as "don't cares." Obviously, only those faults that can affect system functions are of consequence. Such faults are said to be distinguishable.

FAULT DETECTION is the recognition and declaration of anomalous behavior by one or more system mechanisms with discretionary capability. Beyond mere detection of faults, it is necessary to isolate or compensate for them to maintain adequate performance of the system function(s).

FAULT LATENCY TIME is the duration from the occurance of a debilitating elemental event until the resultant anomalous behavior is detected. This delay may result from the fact that the effects are not immediately distinguishable, at least within the capabilities of the fault detection mechanisms.

FAULT DETECTION COVERAGE is the composite likelihood of recognizing all distinguishable faults, weighted according to their respective failure rates, by one or more of the fault detection mechanisms. In a representative computer, the identification of the entire set of distinguishable faults and their respective failure rates is clearly a major challenge.

2.3 FAA REGULATORY NEEDS

Certification of critical or essential systems requires an intensive assessment of safety-related implementation aspects. In the case of digital mechanization, the newness of the associated technology along with inherent system complexity tends to complicate the assessment process. One way to inhibit this tendency is through the availability and use of practical, dependable means to conduct the assessment.

Accordingly, the intent of this study has been to review and propose means to aid FAA and industry engineers in demonstrating the acceptability of hardware fault tolerance mechanisms. Demonstration of properties such as CPU (central processor unit) self-test coverage or comparator-monitor coverage are therefore the ultimate end of this study, and associated testing techniques the partial means. To support regulatory needs, some emphasis is also placed on resolution offered by various test levels or methods and on the essential complementarity of different types of assurance methods (see Ref. 4).

2.4 ASSURANCE TECHNOLOGY NEEDS

As described in Ref. 5, system validation is accomplished by the mutually reinforcing contributions of the three classical approaches to assurance: analysis, testing, and inspection. Basically, global confirmation of system acceptability is based upon analysis, which in turn is selectively supported by testing. Scrutiny of these activities is the vital role of inspection. Application of this approach is described in Ref. 4.

The close coupling of analysis and testing is crucial for high assurance levels associated with critical or essential system functions. This coupling is fostered by this definition study in that simulator test investigations have been planned to:

- o Generate empirical data for analysis methods
  - such as transient or fault latency data for reliability and analysis models
- o Calibrate or confirm fault detection coverage for analysis
  - such as needed to comply with AC No. 25.1309-1.
- o Investigate analytically intractable issues
  - such as applications software detection of CPU faults, which is not feasible using many emulators.

All of these represent vital assurance technology needs that transcend testing *per se*.

2.5 TESTING TECHNOLOGY ISSUES

Basically, the hardware failure effects testing issues are summarized as follows:

- o Low-level fault insertion mechanisms
- o Arbitrary control of fault insertion/removal
- o Non-consequential interference with "normal" real-time operation, faulted or non-faulted

- o Minimized manual intervention in the testing process
- o Selectable low-level hardware and software instrumentation
- o Multiple channel observations
- o Correlated data retrieval/processing/storage
- o Efficient test loop operation.

Note that test case design per se has not been at issue in this study, but rather the means to apply and assess realistic, worthwhile test cases.

2.6 PREDECESSOR R&T ACTIVITIES

It is important to recognize that this study has been constrained by the results of a number of previous programs, and that as a consequence, it has sought to define the best FIIS options under the circumstances. Specific reference applies to Contract NAS2-10270, under which the RDFCS simulator was developed for NASA-ARC and the FAA, and to Contract NAS1-15336, under which the FIU was developed for use with the FTMP (fault-tolerant multiprocessor) at NASA LaRC. These two efforts were not directly related, so some degree of integration engineering remains to be completed after-the-fact.

On a positive note, several subsequent R&T efforts have contributed to the potential realization of FIIS capability. An FAA-sponsored contract, NAS2-11179, investigated low-level RDFCS hardware fault insertion on a limited, manual basis. Then the FIU was installed at NAS-ARC under NAS2-10832, and reportedly was checked out through automated application of the test cases defined under the FAA-sponsored contract. Adding further insight into the use of the RDFCS facility and the characteristics of the flight control computers (FCCs) is the independent research and development (IRAD) work accomplished there by the Lockheed-Georgia Company in mechanizing a quadruplex pitch SAS (stability augmentation system) as described in Ref. 6.

2.7 RDFCS FACILITY CONCERNS/FEATURES

Overall, the concern of this study has been to fully utilize, if not optimize, the current and potential FIIS capabilities of the RDFCS

facility. From the outset, certain aspects of the facility were known to present problems or constraints for the FIIS implementation, and even now, some uncertainties remain that can be resolved only when FIIS development is undertaken. In addition to the problems identified in Section 1.3, it is important to note that:

- The PDP-11/04 is currently indispensable for the use of the facility, but it is a data flow bottleneck for FITS operation
- The PDP-11/60 must iterate airplane simulation equations of motions periodically, and this may be incompatible with test observation time resolution
- The PDP-11/60 overhead associated with disk storage of test data may cause real-time test loop performance problems
- The FIU lacks adequate instrumentation and control features for high resolution/high observability testing
- Failure effects cannot be monitored at the level of insertion (that of the chip), but must be observed at a higher level such as the processor bus lines.

These major concerns have been addressed in this study, and ultimately, they have been among the major determinants in configuring the FIIS options. Another set of determinants has been the existing RDFCS facility features that are supportive of FIIS implementation, as discussed in Section 4.1.2.

2.8 POTENTIAL FIIS BENEFITS

Implementation of FIIS capability can enable certain failure effects investigations that have not (to the knowledge of the authors) been undertaken elsewhere. This results largely from the high fidelity testing afforded by real-time system simulation. There does exist, however, some potential overlap of capability between the proposed FIIS and the FTMP set-up at NASA LaRC (Langley Research Center). To eliminate this, some of the recommendations of this study should actually be targeted for FTMP investigation at NASA LaRC's AIRLAB (Avionics Integration Research Laboratory). In all, the subject investigations are deemed vital to the dependable certification and deployment of critical digital flight systems.

3.0 OBJECTIVES AND SCOPE

Although the scope of this report is purposefully bounded to hardware failure effects investigations suitable for the RDFCS facility, the study has encompassed rather broad objectives in assurance technology. Basically, the overriding concern has been the dependable attainment and assurance of the safety of full-time critical systems. Since the incidences of physical faults or design discrepancies are not negligible, such systems must be capable of preventing associated undesired effects. In the case of hardware faults, this necessitates the timely detection and isolation of defective elements. Such capability involves increased hardware and software to achieve fault tolerance, and this in turn compounds assurance problems. In any case, the issue of fault detection translates ultimately into one of detection coverage, where typically the degree of coverage necessary to meet critical system reliability requirements is extremely high.

Since DFCSs are in general wide bandwidth systems, the allowable time to recognize and isolate a fault is often critically short. Fault latency times are therefore of comparable concern with coverage. In typical digital mechanizations, a large number of faults yield overt manifestations, e.g., complete termination of processing. Many other faults that do not halt processing are readily detectable in a variety of ways such as by hardware monitors or software comparators. The faults of major concern, however, are those that are transient in nature or those that tend to remain undetected for a prolonged duration. The latter class of faults may remain latent until certain input data, runstream instructions, or subsequent faults evoke an anomalous response. Manifestation due to a subsequent fault is highly undesirable because the compound response, which has been neither anticipated nor considered, may well be outside of acceptable limits.

Certification technology is therefore vitally concerned with the definition, implementation, and calibration of fault tolerance provisions. This concern, moreover, focuses largely on the definition and dependable determination of fault detection coverages beyond about 95 per cent. Consequently, the major thrust and objective of this study have been

directed toward testing methods and facilities to enable extensive hardware fault coverage investigations. Since much of the fault detection coverage is dependent on the integrity of the CPU, substantial attention has been oriented toward detection of its failure modes.

3.1 ULTIMATE GOALS

Following the completion of this study, it is hoped that some of its recommendations will be implemented with regard to both system improvements and failure effects investigations. The ultimate goal is that these investigations lead to significant advances in assurance technology, especially with regard to productive test case application and interpretation. As a result it is expected that these technology advances will lead to the earlier and assured certification of full-time flight-critical digital systems.

3.2 PRAGMATIC OBJECTIVES

From a pragmatic standpoint, objectives can be identified on two levels: the results of this study, and the results obtained through carrying out study recommendations. In the case of the study itself, the intent has been to develop FIIS architecture recommendations that provide the best capability for a particular level of expenditure. This has been pursued through in-depth consideration of the existing capabilities and constraints of the RDFCS facility. Further, the determination of what constitutes better capability has been based on assessments of where the most leverage exists to upgrade certification assurances.

Regarding pending FIIS investigation results, the intent has been to foster practitioner confidence in the test methods or interpretations to be applied or developed. Since these are not fully known at this time, there has been an effort to provide an ample margin of FIIS capability. Further, there has been a commitment to pursue development of test methods or mechanisms that can readily be assimilated into industry practice. Last, there has been considerable stress placed upon the capacity for generating clear records of test conditions and events in suitably compact forms.

3.3 STUDY OBJECTIVES

The stated objective of this study was to define FIIS architectures and strategies to enable generic black box-, card-, and chip-level failure effects investigations in the RDFCS facility at NASA-ARC. This definition was to include implementation requirements and design approaches that offered the most attractive cost/benefits features for real-time system simulator investigations of the following:

- o Coverage, latency times, and general effectiveness of various representative DFCS comparator or monitoring schemes
- o Quantification of the extent of failure effects testing achievable or actually achieved
- o Definition of the contributions of the various levels of failure effects testing
- o Development and assessment of failure effects testing methods, with emphasis on transient phenomena, validation coverage, and test productivity
- o Generation of empirical or statistical data for analytical models.

3.4 STUDY ORIENTATION

Originally, this study was to have considered a minimum of three distinct FIIS architectures. It was presumed that appreciably different cost/benefits would be present, so that the study would have focused on the selection of one architecture for development and optimization. With the acquisition of the FIU, the orientation of the study shifted to its best utilization and to compensating for its inadequacies in the RDFCS simulator.

While none of this has altered the foregoing objectives, it has significantly changed the study tasks. As a result, the emphasis has been on a family of FIIS architectures that represent a logical progression of additional facility development and incremental capability. Each architectural option is therefore optimized within the constraints of its allotted resources. Selection of an option is in some respects a matter of the extent of failure effects investigation capability that can be afforded.

3.5 SCOPE OF THE STUDY

As indicated in Figure 1, the thrust of this study is to establish FIIS design requirements that provide desired failure effects investigation capabilities. Furthermore, the rationale as to the types of faults to be investigated and the nature of feasible, worthwhile RDFCS facility modifications are to be described. It remains a follow-on task to actually implement the basic FIIS capability, whether in the form of new system software or additional hardware.

Once such capability is provided, low-level hardware failure effects investigations can be performed through the development and use of applications test software. Again, the present study must anticipate the associated investigator needs, and specify their realization within the constraints of the existing RDFCS facility.

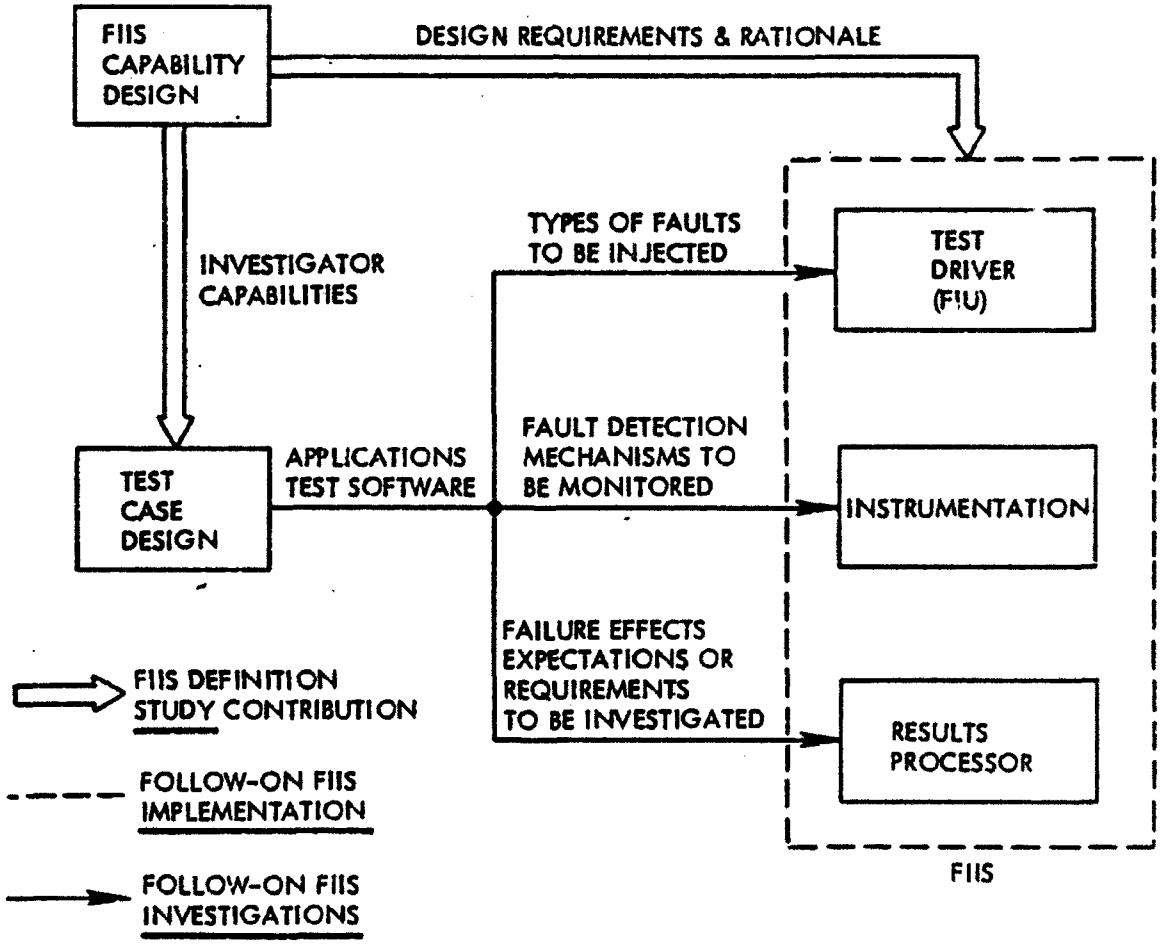


Figure 1. Multi-Stage FIIS Development and Utilization

4.0 TASK RESULTS

4.1 PROBLEM ANALYSIS AND POTENTIAL SOLUTIONS

The investigations formulated and proposed in this study are directed toward establishing the effects of a significant array of faults within an FCC processor. Of special concern is the relationship of particular faults to their means of detection, latency times, and effects on the system. These are paramount assurance issues for flight-critical digital systems because they ultimately determine if system reliability requirements can be met. The approach taken may be summarized as follows:

- Address FAA concerns relative to digital system validation:
  - monitor coverage
  - latency times
  - test conclusiveness
- Pursue generic value, especially at the chip and processor levels
  - Results applicable to other digital systems
- Ensure conclusiveness
  - repeatable, encompassing, documented results
- Extend and better definitize results obtained under Contract NAS2-11179
  - more faults inserted
  - ample, meaningful data recording
  - relevant assessment of results
- Assure effectiveness and compatibility of FIIS options
  - worthwhile results with simplest option
  - extended results with more sophisticated options

A large number of faults have effects that can be easily determined analytically, so these are not pursued here. Included in this group are permanent chip faults such as to enable pins, ground pins, and power pins.

These faults result in a totally inoperative chip, which in most instances causes either a completely inoperable processor or loss of a major computer function.

4.1.1 FIIS Motivation and Requirements

A large percentage of the processor pins can be readily analyzed for the effect of permanent (stuck-high or stuck-low) faults, to the extent that it can be confidently stated that the processor will either produce obviously erroneous outputs or not function at all. Typically, such faults include those that cause erroneous data or addresses (either data, machine instructions, or microcode), or that prevent proper execution of the micro-instructions. The detailed process by which the obvious effect is manifested is often dependent on when the fault is inserted relative to the flight software iteration cycle. Nonetheless the analytically identified overt effect, or variation thereof, will ultimately occur if the fault persists. In the recommended investigations, persistent faults are included not to establish or confirm that processor failure ensues, but to identify, document, and illustrate the fault propagation process.

The second type of fault recommended in the proposed investigations is the transient type. These are recommended for insertion at random points in the flight software, with careful recording of results. Transient faults should be inserted for the minimum duration possible, and hence they tend to necessitate a full-scope FIIS option. This type fault also simulates pattern-sensitive faults that may remain latent for a period of time and then cause erroneous chip output for one or more cycles when certain input patterns are present.

The intent to maximize the generic value of FIIS investigations motivates the emphasis on faulting pins of the microprocessors, the interrupt controller, and the control store programmable read-only memories (PROMs). The reasoning behind this is developed in the next section.

The investigations proposed here would not be redundant to the noteworthy results obtained by McGough and Swern (Ref. 7). The referenced work investigated the fault detection coverage afforded by explicit built-in-test (BIT) routines for a specific avionics processor. The results were

obtained sing parallel gate-level emulations of the subject processor. Each gate- and pin-level fault investigated was inserted in one emulator, with a non-faulted emulator providing a reference against which fault effects could be determined.

The BIT procedure investigated in Reference 7 included simple test problems with correct answers stored, a watchdog timer similar to the RDFCS interation monitor, memory sum tests, parity tests, and others. As each fault was emulated, the time to detection and means of detection were recorded, or the fault was classified as non-detectable.

In the investigations proposed here, the system context, including realistic complete flight software, is provided by the RDFCS. The injected fault may therefore be detected by any of the full set of comparators (e.g., servo coil current, active mode) or by any of the intra-channel fault detection provisions (e.g., bus timeout, iteration monitor, illegal opcode). The results of the proposed investigations then will extend rather than duplicate the Reference 7 results.

Generic Fault Effect Considerations - Figure 2 shows in cursory form the major functional elements of one channel of a representative DFCS. The data producers consist of sensors, other channels of the DFCS, and other aircraft subsystems. The term sensor is used in a very broad sense here to include the inputs from control panel switches and control knobs, as well as from accelerometers and gyros.

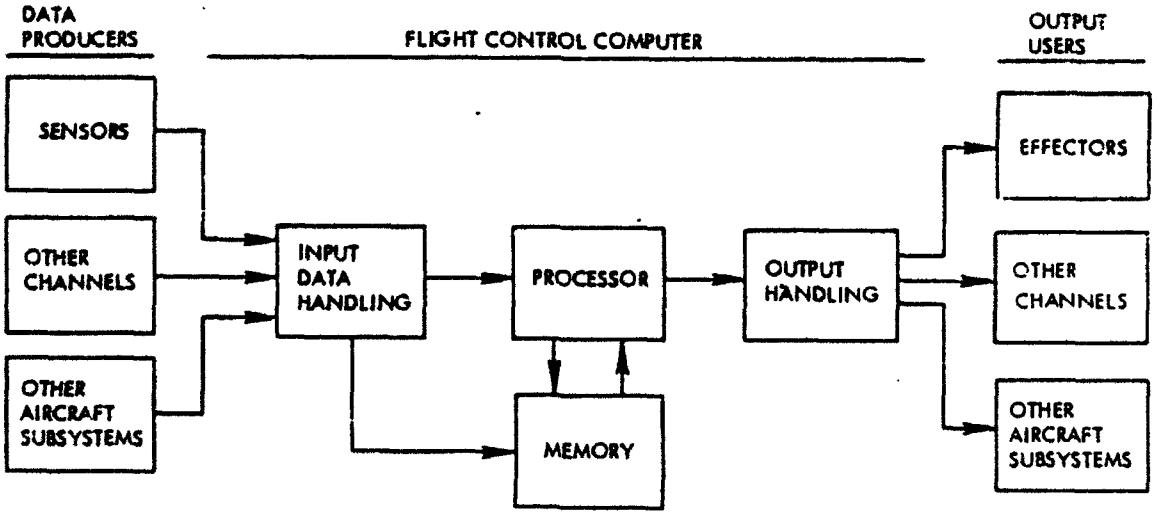


Figure 2. Data Flow in a Representative FCC Channel

The FCC channel shown in Figure 2 assumes autonomous input handling, i.e., the processor is not involved in acquiring the incoming data. This assumption is made since autonomous input (and output) is prevalent in flight controls computer design and will likely remain so in the foreseeable future. The input data handling circuitry also transmits a copy of the sensor data to other channels.

Figure 3 expands somewhat the input data handling function of Figure 2. The ports shown may vary considerably, and each may be fairly complex. Ports for analog inputs may include hardware-implemented pre-filters, signal scaling, and circuits to convert an alternating current signal to direct current. Ports for digital inputs may also be complex, with reformatting, validity bit interpretation, or other built-in functions.

A significant amount of fan-in occurs in the block in Figure 3 labelled MULTIPLEXING. Failures in this block (which may include analog-to-digital signal conversion) may affect several incoming signals, whereas a failure in a discrete port typically affects only a single signal. An exception is that of a port receiving data from other channels, in which a failure could cause the inputs from several sensors to be lost in the

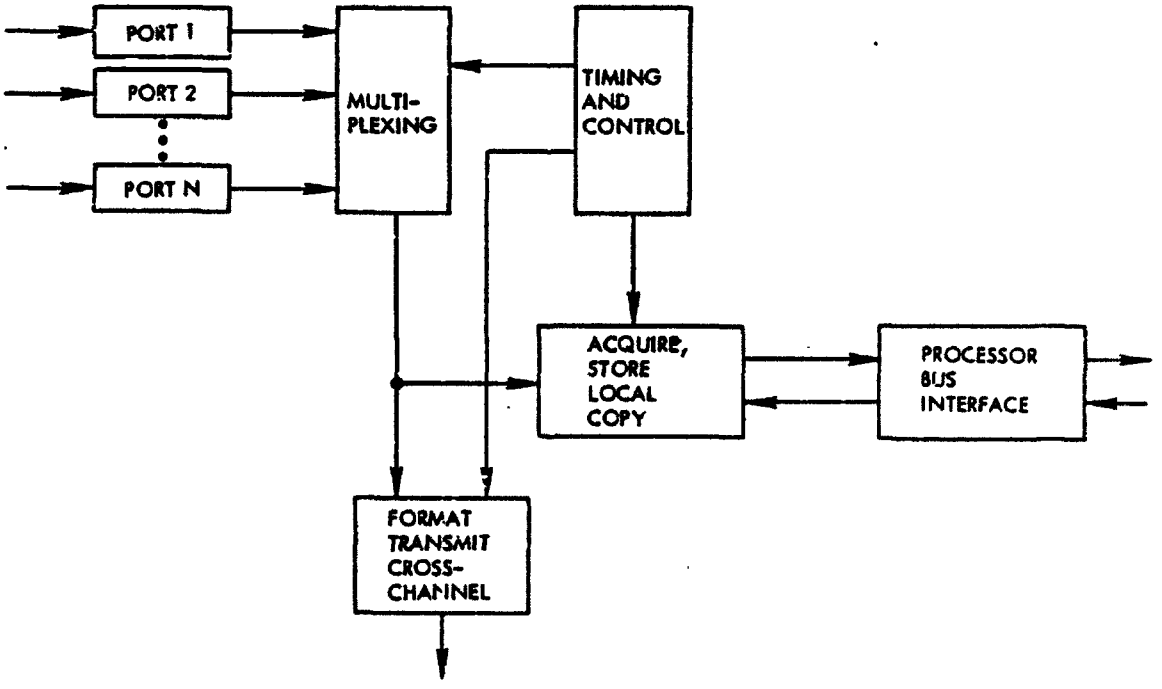


Figure 3. Input Data Handling in a Representative FCC Channel

receiving channel. Depending on the particular implementation, some percentage of faults in the other blocks of Figure 3 would result in data from several data producers being lost to the channel under consideration. The investigation of such faults using the FIIS is of limited interest, since the large variety of ways of implementing the input function diminishes the generic value. Also, the fault detection mechanism of comparison monitoring that is commonly used in voting planes is well understood, so this further lessens the motivation to pursue FIIS investigation of this functional area.

The processor section is that functional area of the FCC offering the most promise for FIIS application. The basic functional content of a hypothetical, microprogrammed, bit-slice processor is shown in Figure 4. The interconnections between functional areas are not shown, since most blocks connect to every other block.

In Figure 4, the microprocessors perform arithmetic and logic operations in conjunction with certain supporting circuits, such as carry look-ahead logic. An interrupt controller receives incoming requests for

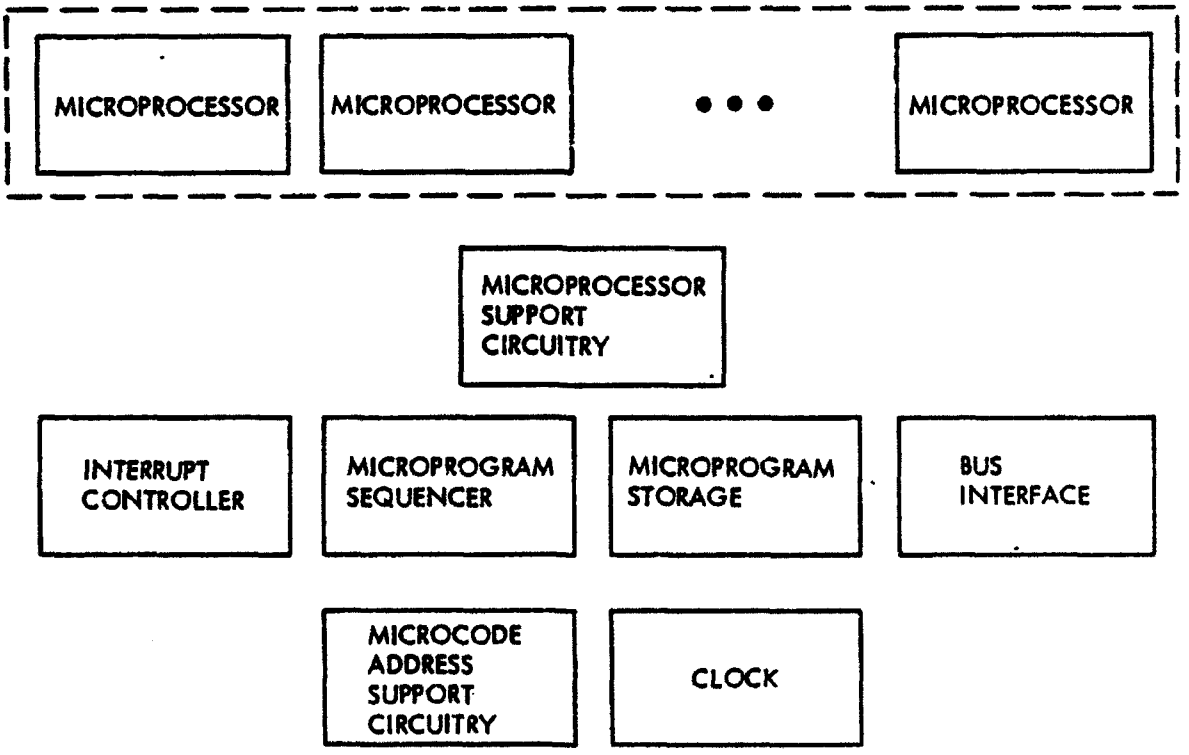


Figure 4. Basic Functional Content of a FCC Processor

service and manages these cooperatively with the microprocessors. The microprogram sequencer generates the required sequences of addresses needed to retrieve the microinstruction words from microprogram storage. Some supporting circuits, such as registers and logic gates, are also involved in selection of the next micro-instruction address, usually as the result of a preceding computational step. Bus interface circuitry connects the processor to the address and data lines of the computer. The clock produces a square wave which provides a timing reference for the other circuits.

Three areas of the processor are of particular interest then from a fault-insertion perspective: the microprocessors, the interrupt controller, and the microprogram storage. The microprocessors are of interest because of their centrality to the computation process, as well as their complexity. The interrupt controller is of interest because of its internal complexity. Note that the nature of the interrupt controller function is generic, even though the details of its function may vary considerably from processor to processor.

Similarly, the contents of the microprogram memory vary among processors depending on the machine-level instruction set, the details of the processor design, and the choice of microcode algorithms. Nevertheless, this memory is of high generic interest because the same algorithms would probably be used for basic arithmetic and logical test operations in a different processor.

Preference of the three identified functional areas also results from the fact that almost any fault anywhere in the processor can be mimicked by some particular fault in the microprocessors, interrupt controller, or microprogram storage. Consequently, faults in other processor areas are de-emphasized, but some shift/rotate multiplexer faults have been included as representative of faults in the microprocessor support circuits.

No faults have been included for the output handling section of the FCC. This section has significant signal fan-out, somewhat the reverse of the fan-in of the input data handling section. Faults in this area tend to be more amenable to analysis than in the processor, and of more predictable consequence.

Memory faults have not specifically been included for evaluation here. This results from the fact that memory faults which cause many individual

data or instruction words to be in error are manageable in that they are easily detected.

Data memory faults affecting only a single data word are represented by the faults as subsequently identified in this report for the microprocessor data and output pins. Faults in program memory which affect only a single instruction are easily detected if they produce an invalid op-code (machine instruction operation code). Those resulting in a valid but wrong op-code can be more difficult to detect. These faults can be most easily simulated by altering the address during an instruction fetch. Momentarily-faulted Am2901 processor chip output pins as called out in Table 4 can produce such faults. They can be explicitly produced if a suitable means can be found to trigger the fault during only a single instruction fetch operation.

The following sections present details of the recommended fault investigations. It should be noted that the value derived from actually inserting such faults increases according to the sophistication of the FIIS option used. The greater results recording capability of the more expansive options allows more meaningful evaluation of the effects of permanent faults. The limited duration faults are of the greatest value if the 250 nanosecond one-shot multivibrator and the parallel chip unit (to be described later) are both available.

Microprocessor Faults - The microprocessors are of particular interest because of their centrality to the execution of the flight software. Additionally, the microprocessors used in the Collins Adaptive Processor System (CAPS) are Advanced Micro Devices Am2901s, which are popular for airborne minicomputers, so consideration of their failure effects is of nigh generic value. The actual extent to which the effects would differ for some other processor depends on the overall processor organization, the processor architecture, and the microcode algorithms. Assuming commonly used microcode algorithms for numerical computations, the Am2901 output pins can be expected to produce similar outputs in any processor, so that the effect of faults affecting only numerical computations would be the same in any processor.

Microcoded special functions, tailored for the needs of the specific application, would tend to differ among various processor designs. The

effect of a microprocessor fault on the Am2901 output pins would therefore be of limited generic value.

At the processor level, the effect of numerical computations on the flight control laws being wrong has high generic interest. The effect of wrong microprocessor output on other processor functions, such as addressing machine-level instructions or responding to interrupts, is very dependent on the number of such operations affected as well as on the processor organization and architecture. Therefore, the areas of similarity and difference between the RDFCS and other processors must be carefully assessed before using FIIS test results on other processors.

Table 4 shows the faults which have been identified for insertion in the microprocessors. The results from Fault Set I will relate specific monitoring features to particular faults. Fault Set II is representative of transient and intermittent faults, and the effect of each may depend on when in the flight software execution cycle the faults occur. The results from the application of these fault sets can enable a preliminary statistical estimate of the coverage afforded by the monitoring mechanisms of the types used in the RDFCS.

While a significant number of the permanent faults of Table 4 were manually inserted as part of Contract NAS2-11179, they are called out for repetition here so that the superior data recording capability of the FIIS can be used to identify more details of the fault effects and to better relate the faults to particular detection methods and times.

Shift-Rotate Multiplexer Faults - The two shift-rotate multiplexer chips can also be of significant generic interest, although less than the microprocessors, in that similar functions can be expected in other processors using Am2901s. The internal logic of these two circuits is straightforward, in contrast to the Am2901s. Their interaction with other circuits, however, can be a source of non-trival complexity. The exact use of these multiplexers is dependent on the algorithms used for arithmetic operations, data shifts, and special microcoded functions, some of which may be quite different in the processors used in other DFCSs. Thus, the results obtained on the RDFCS should be related to another processor only after analytical comparison of the multiplexer functions and the

TABLE 4. MICROPROCESSOR FAULTS

			FAULT SET 1		FAULT SET 2		TEST CONDITIONS	
CIRCUIT	FUNCTION	PINS	TYPE	DURATION	TYPE	DURATION	RESULTS RECORDING	REPETITIONS
AM 2901-(4 EA)	MICROPROCESSOR	F3 CN CN+4 P G QO Q3 RAMO RAM3 AO-A3 BO-B3 DO-D3 YO-Y3 IO-I8 OVER F=O	OPEN	PERMANENT	OPEN	MINIMUM	ALL SOFTWARE-IMPLEMENTED MONITORS. ITERATION MONITOR RECORDED BUT DISABLED. AWB COMMANDS. BUS TIME-OUT RECORDED BUT MASKED OVERFLOW INTERRUPT RECORDED BUT MASKED	FAULT SET 1: 1 EACH FAULT SET 2: 5 EACH
		F3 CN CN+4 P G QO Q3 RAMO RAM3 AO-A3 BO-B3 DO-D3 YO-Y3 IO-I8 F=O	HIGH	PERMANENT	HIGH	MINIMUM	NOTES: 1. UNUSED PINS NEED NOT BE TESTED. UNUSED PINS ARE: F3 (U14, U17, U18) G (U15) CN+4 (U14, U17, U18) P (U15) OVR (U14, U17, U18). 2. RAMO, RAM3, QO, Q3 MUST BE TESTED SEPARATELY IN INPUT AND OUTPUT MODES.	
		CN CN+4 P G QO Q3 RAMO RAM3 AO-A3 BO-B3 DO-D3 YO-Y3 IO-I8 F=O F3			INVERT	MINIMUM		

implementing circuitry in the two processors. Table 5 lists specific faults recommended for these chips.

Interrupt Controller Faults - The Am2914 interrupt controller used in the CAPS processors is a complex integrated circuit with several levels of logic between the input and output pins. It includes internal registers whose contents affect the output produced from a particular input. Hence this circuit has the potential to display pattern-sensitive failure modes. These are of more interest than the pin-level permanent faults manually inserted under Contract NAS2-11179. The most appropriate approach to simulating the presence of such faults is to invert input bits for a minimum length of time, per Table 6.

It is anticipated that some of the faults in Table 6 would cause interrupts that trigger an error routine in the flight software. As currently implemented, this error routine traps the processor in an infinite loop if a bus time-out or overflow error occurs. This software must be modified to eliminate this trap in order to enable productive, automated fault insertion investigations.

Control Store Faults - The 40 output pins of the control store PROMs can be faulted momentarily for a variety of effects. The functions of these pins are shown in Figure 5. A large percentage of the faults that could occur elsewhere in the processor have the same effect as a control store fault, since the control store output is directly involved in almost every function within the processor. The set of faults recommended in Table 7 includes such cases.

It may be noticed that output pins 0-8 have been excluded in Table 7. This is because these pins produce the instruction bits to the four Am2901s, with all four receiving the same bit pattern. More subtle effects are judged possible if the bit pattern to only one Am2901 is disrupted, as specified in Table 4.

The control store pins corresponding to bits 26-35 and 20-23 should be separately faulted depending on the usage of the pins at the time of the fault. For example, bits 32-35 are a direct "A" port address for the Am2901s if bit 15 is 1 (see Figure 5), and bits 28-31 are a direct "B" port address if bits 16-17 are 11. However, when the output 08 (pin 9) of the

TABLE 5. SHIFT-ROTATE REGISTER FAULTS

			FAULT SET 1		FAULT SET 2		TEST CONDITIONS	
CIRCUIT	FUNCTION	PINS	TYPE	DURATION	TYPE	DURATION	RESULTS RECORDING	REPETITIONS
DUAL 4-INPUT MULTIPLEXER U2	GENERATE Q3, RAM3 INPUTS TO 2901 U15	1C0 1C1 1C2 1C3 2C0 2C1 2C1 2C3 1C0 1C1 1C2 1C3 2C0 2C1 2C2 2C3	LOW (CHIP SIDE HIGH (CHIP SIDE)	PERMANENT PERMANENT	INVERT (CHIP SIDE	MINIMUM	ALL SOFTWARE-IMPLEMENTED MONITORS. ITERATION MONITOR RECORDED BUT MASKED. AWI COMMANDS, BUS TIME-OUT RECORDED BUT MASKED.	FAULT SET 1: 1 EACH FAULT SET 2: 5 EACH
DUAL 4-INPUT MULTIPLEXER U3	GENERATE Q0, RAM0 INPUTS TO 2901 U17	1C0 1C1 1C2 2C0 2C1 2C2 2C3 1C0 1C1 1C2 1C3 2C0 2C1 2C2 2C3	LOW (CHIP SIDE) HIGH (CHIP SIDE)	PERMANENT PERMANENT	INVERT (CHIP SIDE)	MINIMUM		

TABLE 6. INTERRUPT CONTROLLER FAULTS

			FAULT SET 1		FAULT SET 2		TEST CONDITIONS	
CIRCUIT	FUNCTION	PINS	TYPE	DURATION	TYPE	DURATION	RESULTS RECORDING	REPETITIONS
AM 2914 INTERRUPT CONTROLLER	MANAGE INTERRUPT REQUESTS	I0-I3 M0-M7 P0-P7	INVERT (CHIP SIDE)	MINIMUM			ALL SOFTWARE IMPLEMENTED MONITORS. ITERATION MONITOR RECORDED BUT DISABLED. AWI COMMANDS BUS TIME-OUT RECORDED BUT MASKED. OVERFLOW INTERRUPT RECORDED BUT MASKED. NOTE: ERROR HANDLING PROCEDURE IN FLIGHT SOFTWARE REVISED TO ELIMINATE INFINITE LOOP.	5 EACH

TABLE 7. CONTROL STORE FAULTS

			FAULT SET 1		FAULT SET 2		TEST CONDITIONS	
CIRCUIT	FUNCTION	PINS	TYPE	DURATION	TYPE	DURATION	RESULTS RECORDING	REPETITIONS
CONTROL STORE PROMS U3-U6, U9-U12, U17-U18	STORE AND RECALL MICROCODE INSTRUCTIONS	MICRO-CODE WORD BITS 9-39	INVERT (BOARD SIDE)	MINIMUM			ALL SOFTWARE IMPLEMENTED MONITORS. ITERATION MONITOR RECORDED BUT MASKED. AWI COMMANDS. BUS TIME-OUT RECORDED BUT MASKED. OVERFLOW INTERRUPT RECORDED BUT MASKED.	5 EACH.

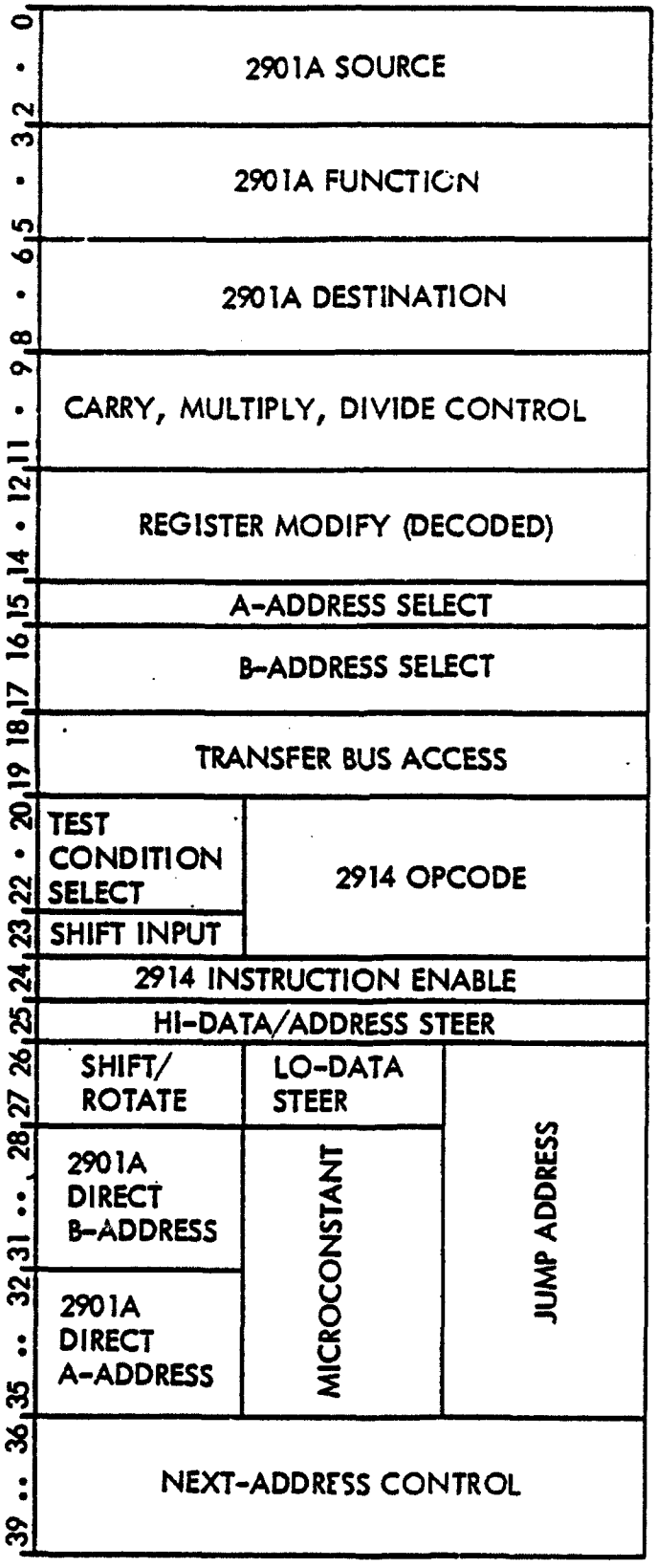


Figure 5. FCC Control Store Fields

Next Address Control PROM is low, bits 26-35 are routed to the microprogram sequencer as the next microcode address. Consequently, these bits (pins) should be faulted for each of their functions separately to enhance the generic value for other processors in which control store pin functions are dedicated rather than shared. Similarly, pins 20-23 should be faulted depending on their function at the time of the fault.

Results Monitoring - As shown in Tables 4 through 7 the primary data recording points are the RDFCS comparators and monitors. These merit individual discussion as follows:

- **Software-Implemented Monitors** - The RDFCS includes software-implemented comparators of sensor data. The flight software must be modified so that the comparison function is still performed but comparator trips are ignored. Similarly, software-implemented servo command or response monitors must also be modified. The occurrence of each comparator trip must be recorded, but the comparator output must be either reset to non-failed or ignored. In a like manner, mode logic disagreement must be recorded but overridden so that the system does not disengage.

- **Hardware-Implemented Monitors** - The trip of a hardware-implemented monitor (e.g., coil current comparator) must not result in system disengagement. The monitor trip, however, must be recorded.

- **Iteration Monitor** - The iteration monitor uses both software and dedicated hardware. The RDFCS presently has a provision to override this monitor, but it may not be compatible with the need to observe and record the iteration monitor function while eliminating its authority to disengage the servos.

- **AWI Commands** - Commands to the AFCS (Automatic Flight Control System) Warning Indicator (AWI) should be monitored. Depending on the approach taken to disabling FCC comparators, there may or may not be any commands issued. If the software-implemented comparators are modified, the record of these commands can be useful in ascertaining that modifications have been satisfactorily made.

- **Bus Time-Out and Overflow Interrupts** - The ERROR subroutine in the flight software responds to bus time-out and overflow interrupts by placing the processor in an infinite loop. This portion of the software must be modified so that the processor resumes executing the foreground and background routines, and, if necessary, modified so that the occurrence of the interrupt can be recorded.

4.1.2 RDFCS Facility Assessment

The present RDFCS facility as depicted in Figure 6 includes a Collins CAPS-6 based DFCS and a DEC (Digital Equipment Corporation) PDP 11/60 digital computer for control, simulation, and evaluation. These elements are supplemented by interfaces which allow the entire system to perform simulation and testing functions in a high-fidelity, real-time reference frame.

A wide-bodied transport aircraft simulation is presently programmed on the PDP 11/60 to provide a number of representative flight cases covering a spectrum of gross weight, velocity, and altitude. These flight cases provide aircraft configurations for takeoff, climb, cruise, descent, approach, and landing. In addition, each simulation case has ground-referenced geometry for glideslope, localizer tracking, and ground track. The landing cases provide for ground effects aerodynamic coefficient transitions. Altogether, the 20 available flight cases provide sufficient coverage of the

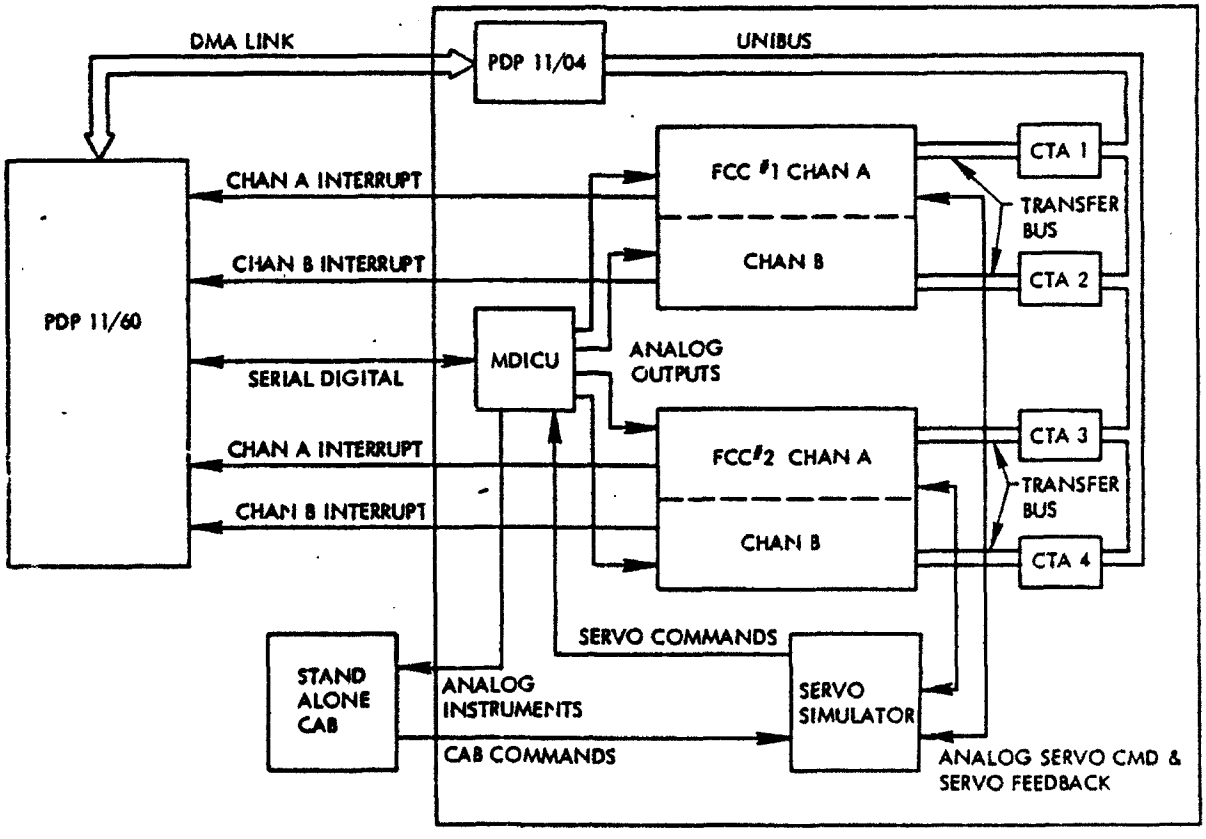


Figure 6. RDFCS Facility Layout

flight envelope to utilize all modes of the RDFCS.

In addition, there are two transitioning aircraft models in the simulation package. It is possible to set up the aircraft in the approach mode with flaps at 22 degrees. With a heading selected, the aircraft can capture the localizer beam, engage and capture the glideslope beam, and then reconfigure itself aerodynamically as the flaps extend to 33 degrees. The aircraft and autopilot can then proceed into the landing and flare maneuvers. A second transitioning case provides for an aerodynamic reconfiguration from landing to takeoff as the go-around maneuver is engaged. The flaps retract from the 33 degree position back to a takeoff position of 22 degrees.

The two transitioning cases provide an effective simulation of the aircraft as it flies through two crucial phases under control of the flight control system. Windshear and a random Dryden gust model are available for introducing external winds and turbulence into the simulation. Gust amplitudes are specified for each flight case, but may be changed at the discretion of the investigator.

For flight simulation purposes, the PDP 11/60 and the DFCS are interfaced through a Modular Digital Interface Control Unit (MDICU). This unit provides analog versions of signals from the simulation for inputs to the FCCs as well as digitized inputs from the DFCS to the simulation. The MDICU is interfaced to the PDP 11/60 through a serial Manchester encoded data bus. This bus is terminated in the PDP 11/60 I/O page as two 64-word buffers for data transmit and receive. Data transfer is handled by the MDICU and does not involve PDP 11/60 processor interrupts.

This type of input/output (I/O) is efficient in that it does not require special action on the part of the real-time routines operating in the PDP 11/60. Incoming data to the PDP 11/60 is stored in a RAM (random access memory) buffer which may be accessed by any program mapped to the I/O page. Outgoing data from the PDP 11/60 is transferred from the I/O page to a first-in/first-out (FIFO) buffer which is 64 words deep. A data rate of approximately 16k words per second is obtained by shifting a word out of the FIFO every 62 microseconds. Since the data are Manchester encoded with address and parity bit, the effective data rate of the serial interface is 357 kilobaud, full duplex. Simulated aircraft data can be

transferred over the interface at varying rates determined by the simulation program.

The MDICU contains an embedded CAPS-6 computer which could be used for a number of test functions. Presently this CAPS-6 is used for scaling and routing I/O data to and from the appropriate digital-to-analog converters (DACs) and analog-to-digital converters (ADCs). Since the MDICU serves as the I/O processor between the simulation and the DFCS, its basic functions must necessarily be performed. However, it could be used for auxiliary functions such as limited sensor or actuator modeling.

A second interface exists from the PDP 11/60 through the PDP 11/04 to each of the CAPS Test Adapters (CTA) and its associated CAPS-6 computer. This path is intended primarily for control, test, and analysis of the CAPS-6 computers from the PDP 11/60. The link between the PDP 11/60 and the PDP 11/04 is a direct memory access (DMA) which transfers data independently of the PDP 11/60 processor once the transfer is initiated. From a PDP 11/60 program, it is possible to perform any of the following CTA functions:

1)	READ/WRITE	PDP 11/04 memory
2)	READ	CTA status
3)	WRITE	CTA control word
4)	READ/WRITE	CTA data display registers
5)	READ	CTA history port
6)	READ/WRITE	CTA window.

1) READ/WRITE PDP 11/04 memory allows blocks of data to be transferred between the PDP 11/60 and the PDP 11/04.

2) READ CTA status allows monitoring of the condition of the CAPS-6 processor and bus for run, halt, or error conditions. The value of the history counter can also be determined.

3) WRITE CTA control word allows: control of the processor and transfer bus for halt, step, and run conditions; decrementing the history counter; and monitoring of break address, data compare, or bus error.

4) READ/WRITE CTA data display registers - The register on the CTA displaying the current address, data and keyboard can be monitored and changed from the PDP 11/60.

5) READ CTA history port - The CTA contains a TRANSFER BUS HISTORY buffer into which is deposited the contents of the status, address, and data registers for the 16 most recent bus operations. The contents of these buffers are available to programs running on the PDP 11/60. The transfer bus is automatically halted when a READ HISTORY PORT is initiated.

6) READ/WRITE CTA window - The Unibus in the PDP 11/04 is connected to the transfer bus in each CAPS-6 computer by a high-speed data window in which a MOVE instruction in the PDP 11/04 is automatically transferred into a similar operation in the CAPS-6 at a preselected address. This is an extremely powerful device in that it gives programs in the PDP 11/60 direct access to the entire memory of each CAPS-6 computer. Thus, a PDP 11/60 program can read and modify data or instructions in the CAPS memory.

All of these CTA operations can be performed by the PDP 11/04 acting as a peripheral processor to the PDP 11/60. Except for initialization, all data transfer between the PDP 11/60 and the PDP 11/04 is accomplished by standard NPR (non-processor request) data transfers and operates on a cycle-steal basis so that the overhead of processor interrupts is minimized.

A number of growth provisions exist in the present facility. While these features may not affect the implementation of the FIIS directly, they should permit higher quality results to be obtained as faults are introduced and the subsequent failure results are monitored. Since the feasibility of transitioning flight simulation using state models has already been demonstrated, it might be desirable to have an aircraft capable of transitioning toward several corners of the flight envelope to provide a realistic assessment of dynamic performance in the event of a critical failure in the flight control system. This transitioning capability can be accomplished by determining the coefficients of incremental forcing function matrices during the initialization phase and solving these matrices in real time.

In a similar manner it should be possible to incorporate various non-linear effects such as stall characteristics at high angles-of-attack. Second-order non-linear effects can normally be introduced as additional state forcing function matrices with greatly improved simulation fidelity. The present transitioning simulation requires approximately 7.5 milliseconds per cycle at 50 cycles per second. This leaves approximately 12.5 milliseconds for additionanl computation including system overhead and

context switching. As sparse matrices are incorporated into the simulation to handle the non-linear effects, it becomes practical to consider computational enhancements such as provided by an array processor operating from a host PDP 11/60. The total matrix developed in this manner becomes banded and is ideally suited for array processing. A unique feature of this concept is that as the simulation becomes increasingly complex,the computation becomes more efficient.

A further refinement of the airplane simulation would be the addition of landing gear dynamics to allow the DFCS to follow through the landing phase into the roll-out mode. High fidelity simulation of gear dynamics normally involves extremely high frequencies due to the dynamics of the unsprung mass in real time. However, if these effects are filtered out so that only the lift decay is considered as the aircraft settles on the landing gear and enters the rollout mode, it should be possible to reasonably simulate the rollout effects. The transition from an aircraft suspended aerodynamically to one supported by the landing gear requires that the simulation shift between two entirely different dynamic models. This requires a real-time program on the PDP 11/60 which would be considerably more complicated than the present linearized simulation. However, to utilize the FIIS for fault investigations in this critical landing maneuver requires a more sophisticated simulation than that currently available.

Another area having a strong influence on the application of the FIIS is the interface from the PDP 11/60 to the CTA via the PDP 11/04. If it becomes desirable to either enhance the simulation on the PDP 11/60 or handle large quantities of data through the CTA window to a CAPS transfer bus, then a different control, data handling, and data analysis concept should be considered. The PDP 11/04 used as the intermediate processor in the PDP 11/60-CTA link is the lowest performing processor in the PDP 11 series. It may be desirable to replace the present data link with a stand-alone Unibus type processor that is capable of handling the FIIS analysis. With a stand-alone system, large amounts of data could be transferred to a bulk storage medium without interferring with the simulation running on the PDP 11/60.

While the RDFCS simulator provides a good representation of a transport type aircraft operating with a DFCS, certain conditions may arise during FIIS utilization that might yield fallacious results. The aircraft

model is linearized about a point of constant velocity and dynamic pressure. While the performance around the operating point has reasonably high fidelity, as the system is driven away from this point the flight characteristics tend to deviate from the norm. An induced fault which causes an abrupt nose-up condition during a landing or go-around maneuver might yield non-realistic effects because the aircraft model does not exhibit proper stall characteristics at the present time. An abrupt nose-up command from a malfunctioning DFCS system would cause increased lift with a resultant gain in altitude, rather than having the lift decrease and altitude loss as a result of typical stall conditions. If stall characteristics are an important aspect of any of the FIIS investigations, then they should be built into the aerodynamic model.

Another limitation of the facility is the necessity of manual engagement of the DFCS after the aircraft is in a flying mode. In order to activate the DFCS system, the bat handles on the glareshield control panel must be raised, the autothrottle must be engaged, and then the proper autopilot mode must be selected. While this sequence of operations is directly analogous to the actual flight procedures, it introduces an element of time uncertainty with each run that is made. Therefore, some dispersion in the results may occur among different runs when the FIIS is being used. It may be virtually impossible to introduce faults into the system so that the fault occurs at a precise or consistent point during the flight maneuver or at a preselected point in the DFCS execution cycle.

When the RDFCS simulator facility was originally conceived, it was anticipated the various components would be physically situated at remote distances from each other. A fiber optic link was anticipated to provide low noise serial communication between the PDP 11/60 and the MDICU. The fiber optic link never provided reliable performance and was replaced by an electrical serial link, with the PDP 11/60 and the MDICU being in close proximity. The MDICU serial link is controlled internally and does not require interaction by the PDP 11/60; however, serializing the data does introduce a measurable phase lag between the simulation model and the DFCS system. This lag should be considered in the analysis of data obtained as a result of inserted faults.

Another major limitation of the system is the speed of the PDP 11/04 processor in accessing the CAPS transfer bus via the CTA window. While the

PDP 11/04 relieves the PDP 11/60 of processor I/O to the CTA window, it is extremely slow and has no provision for bulk storage. Data collected as part of a fault insertion analysis must be transferred over the DMA interface and stored in PDP 11/60 memory while real-time simulation is taking place. It cannot be downloaded onto the disk during real time because of the block transfer time requirements of the disk.

While a number of limitations have been cited, it should not be concluded that the RDFCS simulation facility is not a powerful tool for analyzing both hardware and software DFCS failure effects. The simulator provides a great deal of flexibility in introducing faults and analyzing these effects as long as the system limitations are properly understood or appropriate modifications made.

4.1.3 Potential Testing Scheme

A model of failure effects testing as presented in Figure 7 begins with the definition of test cases. These establish the aspects of the DFCS to be examined and the facilities needed to perform the testing. Automated and arbitrary control of fault injection enables the application of an appropriate range of test stimuli to the FCC, and if desired, to a fault model that serves in the interpretation of test results. The fault injector (FI) also controls a precise clock used to measure fault latency times that are associated with various fault detection mechanisms. The low-level instrumentation needed for definitive testing is implemented in both hardware and software, and test results processing is normally performed in the test control computer.

In implementing and using such a test scenario, the major concern is that of obtaining needed resolution in the application of stimuli and the timing of responses. Coordination of the total test loop is therefore of pivotal importance. Time delays, skewing, and indeterminacies must be effectively eliminated, and this necessitates adequate data rates and processing capacities throughout the loop. Some tradeoffs do exist, e.g., processor throughput suitably located can alleviate data rate or storage requirements. In a very limited sense, such a tradeoff indicates the broad range of possibilities in configuring a FIIS to enable the test scenario in Figure 7. In an optimized implementation, the balanced and economical use

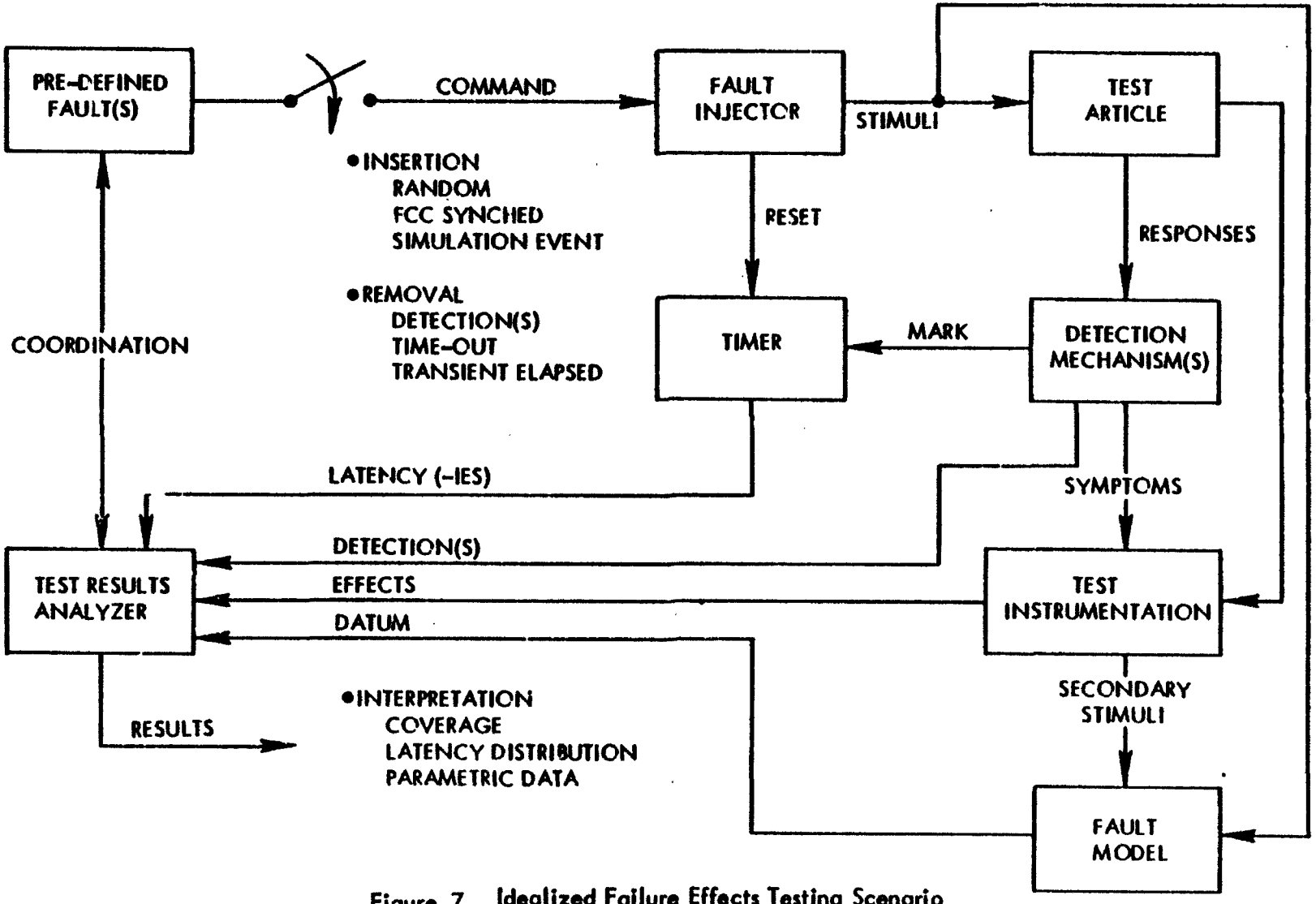


Figure 7. Idealized Failure Effects Testing Scenario

of resources is a fundamental guideline, and the ultimate measure of success is the resultant level of capability.

4.1.4 Fault and Failure Detection in the RDFCS

The fault-failure detection provisions used in the RDFCS include software- and hardware-implemented monitors. These are used to detect sensor faults, computer faults, or lack of proper servo response.

Although the RDFCS contains only simulated sensors, the FCCs include sensor monitoring functions suitable for use in an actual airborne autopilot. The signals from triple and quadruple sensors are compared and voted in software prior to use, with each of the four computer channels performing the comparison and voting on the signals it will use in control law computations. This permits faults in the data input section of a computer channel to be detected as well as faults in the sensors themselves. The sensor signals are monitored just prior to use, so a signal which will not be used is not compared. For example, pitch attitude is monitored and used in cruise autopilot vertical modes, but pitch rate is neither used nor monitored. During an automatic approach, the control laws use pitch attitude rate instead of pitch attitude, and so pitch rate monitoring begins and pitch attitude monitoring ends upon engagement of the Approach/Land Track submode of autopilot operation.

The triple and quadruple sensors also produce discrete validity signals which are monitored by software within each FCC channel, so that an individual sensor signal will not be used if it does not compare closely with others of the same type or if the associated validity signal is not present.

Dual-dual and quadruple sensors are comparison-monitored in the same way, but the response to a fault is different. Upon detection of the first fault in a quadruple sensor set, the other three sensors are treated as a triple sensor, with the bad sensor excluded from further use. The dual-dual sensors have high-integrity self-monitoring, which is relied upon particularly for detection of a second fault. When a single side of a dual-dual sensor is detected faulty, the entire sensor is condemned, with the two outputs from the other sensor compared for disagreement for the remainder of the flight. Since the only dual-dual sensors are the

Instrument Landing System receivers and the radio altimeters, which are used only in approach and landing, the duration of the flight segment is the short time to touchdown from an altitude of 1500 feet or less.

Sensor data coming into the FCCs are handled by autonomous input handling hardware. The sensor monitoring functions can detect failures of this hardware which cause one or more sensors to appear faulted to the computer, faulty sensor wiring, and actual sensor faults. This monitoring cannot detect program memory faults, other than those which cause the sensor data to be read from a valid but wrong address.

During operation, the processor is continually checked for its ability to execute its instruction set, program memory is checked, and the repeated execution of the foreground loop is monitored. The specific fault detection methods used are as follows.

CPU Diagnostic - The CPU diagnostic routine is allocated time in the background mode every 200 msec. Each machine-level instruction used elsewhere in the flight software, other than those which would interfere with system operation (e.g., CLEAR-CLOCK), is executed with the result compared to the proper result. Failure of the processor to produce the correct result for any instruction causes autopilot and yaw SAS servo disengagement. The diagnostic program also produces a count of the number of instructions tested. The foreground software monitors the number of instructions executed since the previous time the counter was checked. If the counter does not have the correct value or does not change for 120 seconds, a failure is declared in the foreground software and the corresponding FCC channel disengages.

Checksums - Each PROM card has stored in its first two 16-bit addresses the 32-bit sum of the contents of all other addresses on the card. The contents of each of these other addresses is added and if the sum does not equal the contents of the first two addresses, a failure is declared and the FCC channel disengages. The foreground software declares a failure if more than 20 seconds elapse since the last successful checksum test of any card.

Iteration Monitor - In each channel, the foreground executive software

module executes every 50 msec through one of 4 paths. Bit 15 of a data word written to a specific hardware address is set FALSE at the beginning of paths 1 and 3 and set TRUE at the beginning of paths 2 and 4. The status of this bit is continuously output as an electrical signal, so that a 10 Hertz (Hz) square wave is produced when the software is being executed normally. Dedicated hardware in each channel monitors the presence of the 10 Hz wave, which is required for servo engagement.

Path Monitoring - Each channel also compares its foreground path number to that of the other channel in the same box, and disagreement causes the FCC channel to disengage. Path number monitoring is discontinued in the Approach/ Land track submode of operation.

Wrap-Around Test - Flight director (FD) commands are wrapped-around to both channels of each FCC. Wrap-around failure in either channel results in a bias to hold the FD command bars out of view.

Servo Command Monitoring - Servo commands for roll, pitch, or yaw are comparison monitored in hardware. Dual coil current comparators monitor the outputs from both channels of each FCC. Disagreement by any comparator will cause either yaw SAS or roll and pitch servo disengagement, as appropriate. Servo rate is also monitored in hardware for the pitch axis. Servo modulator piston position is monitored in software for the roll and yaw axes.

Bus Activity - The FCC software includes monitors to ensure that sensor data and cross-channel data are being updated at the required rate.

Control Panel Bus - The digital bus to the control panel is tested by circulating test words. Periodically, one of three such words is transmitted by the FCC. If it does not receive this word back from the control panel within the prescribed time, a bus failure is declared, and further commands from that panel are ignored.

Cross-Channel Mode Agreement - Each channel periodically transmits its mode status to the other channel in the same FCC. If these disagree for more

than one iteration of the foreground software in either channel, the software withdraws its enable input to the servo engage logic.

Arithmetic Overflow - If the result of an arithmetic computation overflows, the microprocessor handling the high-order bits sends a high-priority interrupt request to the interrupt controller. If not masked, this causes the interrupt controller to initiate the interrupt handling sequence, which in turn results in the software branching to an infinite nil loop. Disagreement then results for one of several reasons: iteration monitor trip, mode disagreement disconnect, coil current comparator trip, etc.

Illegal Opcode - An attempt by the processor to use a stack area which is outside of the allocated address range will result in a high-priority interrupt. As in the case of arithmetic overflow, the software enters the infinite nil loop and the servos disengage.

Bus Timeout - A bus time-out error occurs if the processor attempts to address a non-existent memory address. This has the same effect as the arithmetic overflow and illegal opcode errors just discussed.

4.2 CANDIDATE FIIS ARCHITECTURES

Under this section, candidate FIIS architectures are defined and proposed to improve the capabilities of the RDFCS to support low-level failure effects testing. The FIIS options are partitioned into the following four progressively enhanced categories:

- Existing system with only software modifications
- Improved system with a combination of software and modest hardware modifications
- Advanced system based on extensive modifications
- Superior system based on the full scope of feasible modifications.

All proposed system architectures are upward compatible and are based on the use of the Draper FIU for accomplishing the fault insertion

function. The intent of all of the proposed FIIS architectures is to provide improved low-level fault insertion and instrumentation capability that supplements the existing FIU features. Because it constitutes the greatest current need, the emphasis here is largely on instrumentation. The first of the proposed FIIS options consists only of the addition of new software to enhance the existing facility. The software to be added enhances the existing capability by the addition of a FIIS executive in the PDP 11/60 that would give the user access to the extended capabilities to be offered by new test-related software. Such features would include FI initialization, fault insertion control, extended DMA capability to the CTAs, and PDP 11/60 results processing.

The second proposed FIIS option consists of the addition of custom hardware that would passively monitor and record the CAPS transfer bus transactions in cache memory. A bus monitor/recorder unit (BMRU) would also contain a clock for fault detection timing measurements, receive and decode interrupts from the FCCs in the pallet, and interface to the PDP 11/60 via a DMA data link.

The third candidate architecture includes a PDP 11/24 minicomputer to replace the much slower PDP 11/04. A BMRU similar to the one described under FIIS Option Two would be used to capture CAPS transfer bus transactions for later analysis by the PDP 11/60. A system clock would also be added for better measurement of fault detection times under this higher performance and resolution system.

The fourth FIIS option builds upon the capabilities described under the first three systems, coupled with the addition of sophisticated nardware for pin-level instrumentation. A unit for paralleling two of the same chips is proposed that would allow conclusive determination of whether an injected fault propogates to the output pins of the device under test. Another proposed unit is a logic state recorder that would have the capability to selectively monitor and record the states of various pins. This information could then be analyzed by the PDP 11/60.

4.2.1 Option One: Software Modified System

As indicated in Figure 8, the first FIIS architecture defined is composed of the existing RDFCS facility, including the Draper FIU, and new

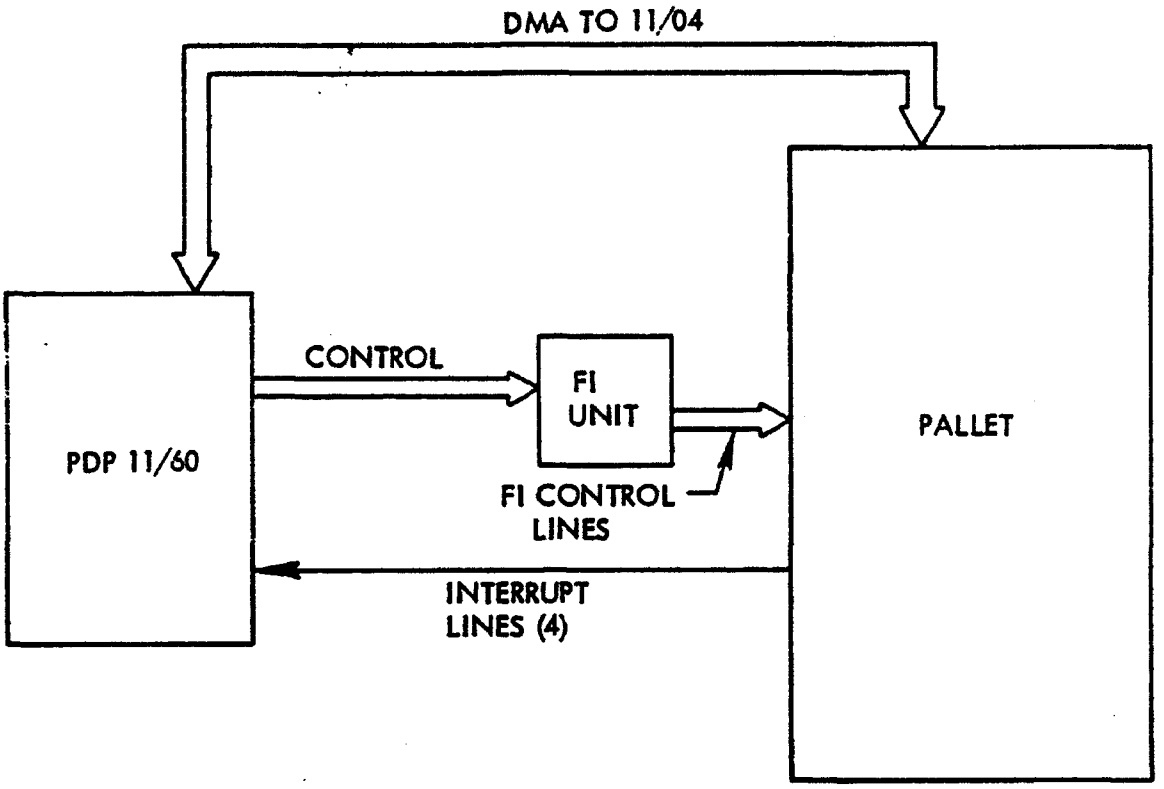


Figure 8. FIIS Option 1 Architecture

programs written to upgrade the capability to control the insertion of simulated faults and to record results and timing information. The specific software to be added is summarized in Table 8. The following elaborates the functions of the various modules listed in the table.

PDP 11/60 Software - The software structure for the PDP 11/60 minicomputer additions is illustrated in Figure 9. The FIIS executive is the interface to the expanded functions proposed to support low-level testing utilizing the Draper FIU on the RDFCS facility. The executive is partitioned into initialization and results processing (non-realtime) modules and a real-time control section for test case execution. Referring to Figure 9, the non-realtime routines consist of:

- Draper FI initialization
- PDP 11/04 initialization
- Aircraft model initialization (already existing)
- Results processing.

TABLE 8. NEW SOFTWARE MODULES FOR OPTION 1

PDP 11/60	PDP 11/04	CAPS-6
• CONTROL RCUTINES FOR DRAPER FIU	• DATA BUFFERING ROUTINE	• BACKGROUND TEST EXECUTION MONITOR PROGRAM USING EXCESS CORE MEMORY
• EXPANDED CONTROL ROUTINES FOR PDP 11/04	• EXPANDED DATA COLLECTION	• GENERATION OF INTERRUPT SIGNAL
• RESULTS PROCESSING		
• DR 11-C INTERFACE DRIVER		
• SYSTEM CONTROL PROGRAM		

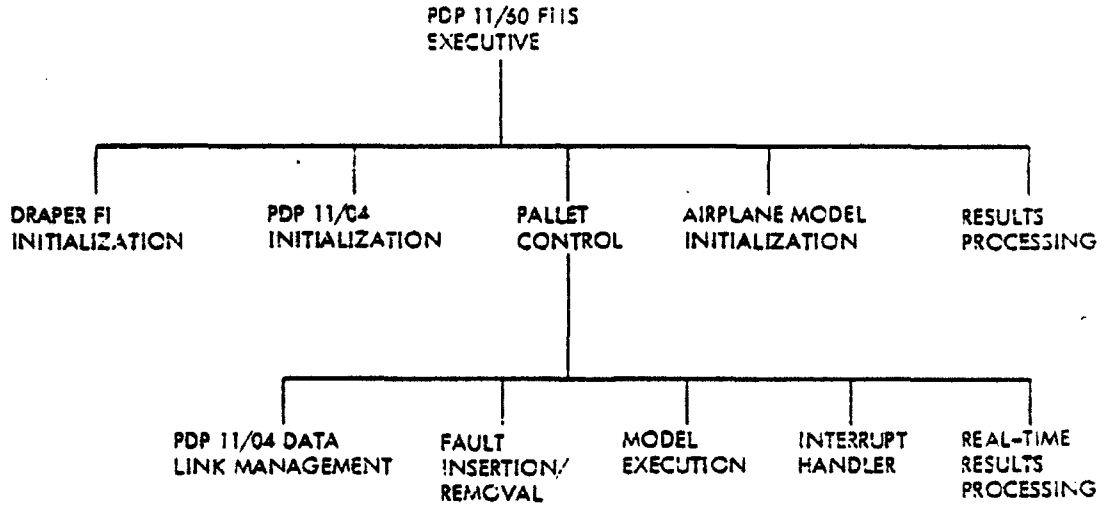


Figure 9. PDP-11/60 FIIS Software Structure

After initialization of the test sequence is accomplished, the executive begins real-time execution of the following routines:

- o Interrupt handling
- o Fault insertion/removal (Draper version existing now in PDP-11/60)
- o PDP 11/04 data link management
- o Aircraft model (already existing)
- o Real-time results processing.

The following details the functions performed by each of the proposed new software modules:

- FIIS Executive - The FIIS executive would be the primary user interface to the new FIIS software that supports low-level testing on the RDFCS facility. In addition to controlling the non-realtime and real-time functions outlined above, the executive would also have the capability to start, stop, and reset the FCCs on the pallet, as well as to start and stop strip chart recorders. These two functions would be provided by using spare digital-to-discrete (D/D) converters in the MDICU. This would require only the addition of new wiring to the pallet. The FIIS executive would be designed to simplify test case definition and to provide for efficient test case execution. The executive would have the capability to store a sequence of test cases to be executed in order to alleviate redefining new parameters for each test case executed. Automated testing may then be achieved by overriding the bathandle logic in the flight software. It may be necessary to provide a mechanism to reload the flight software after execution of particular test cases, and this could be achieved by using the PDP 11/04 under control of the PDP 11/60 to transfer the flight software between FCC channels.

- Draper FI Initialization - The initialization program for the FI would provide the commands necessary to define the faults to be injected in the FCC under test during real-time operation. The functions to be made available would be the ones described in Draper Report CSDL-R-1602 (Ref. 8) plus commands specific to the RDFCS facility. These are summarized in Table 9.

- PDP 11/04 Initialization - The PDP 11/04 initialization program would allow the definition of the memory locations in the FCCs to be monitored by the PDP 11/04 during test case execution. This information would be stored for use by the PDP 11/04 data path management program which executes in real time for the actual data transfer.

- Interrupt Handler - This real-time program would be used to service interrupts generated by the FCCs in the pallet. The program would decode the four interrupt lines (one from each channel) to determine which FCC initiated the interrupt. This information would then be used as dictated by the test case definition. One possible use might be to signal the PDP 11/60 that the injected fault has been successfully detected by the CAPS-6. By reading the real-time clock in the PDP 11/60 at the time of fault injection and again when the interrupt occurs, a rough order-of-magnitude measurement of the fault detection time could be obtained.

- Fault Insertion/Removal - This real-time program would be used to control the actual fault injection and removal by the FI. Fault injection and removal could be accomplished as a function of:

TABLE 9. FIU COMMANDS

Command	Function
Define Unn M	This command defines an M pin Integrated Circuit (IC) whose designation is Unn on the board
Map n AM 1	This command maps the IC under test to the appropriate FIU multiplexer
Describe n abcd	This command describes the fault abcd to be injected into pin n of the IC under test
Func abcd	This command is used to select the boolean function
Enable n	This command selects pin n of the IC and enables the unit for fault injection
Disable n	This command disables pin n of the IC under test for fault injection
Exec n	This command injects the fault for n seconds
Transient	This command causes the PDP 11/60 to inject the fault and then immediately remove it (~6 msec elapsed time)
Inject 1 m n	This command injects the fault as a function of the aircraft model parameter 1 where m is either equal to, greater than, or less than and n is the reference value
Pulse m n	This command causes the fault to be turned on at m second intervals and left on for n seconds (Note m must be greater than n)

- User initiated from PDP 11/60 terminal
- PDP 11/60 real-time clock
- Aircraft model parameters
- Pallet parameters
    - o through interrupts
    - o through PDP 11/04 data path.

An example of fault injection and removal might be the injection of fault at the automatic landing decision height (aircraft model parameter), followed by its removal five seconds later or upon receiving an interrupt from the FCC indicating detection of the fault.

o Results Processing - Results processing would be implemented through both real-time and non-realtime functions. The real-time processing would be limited to the annunciation of certain key events, e.g., notification that a fault was injected or that an interrupt from the pallet was received. The non-realtime processing would be test case dependent, but would include automated report generation and plotting capability.

PDP 11/04 - The software modifications proposed for the PDP 11/04 would consist of expanding the functions that the PDP 11/04 could perform, and of increasing the amount of test data that could be transferred over the data path from the PDP 11/04 to the PDP 11/60. Software would be added to the PDP 11/04 that could allow it to poll a set of CAPS-6 addresses that were designated by the PDP 11/60 during test case initialization. These data would then be buffered by the PDP 11/04 for transfer to the PDP 11/60 for results analysis after termination of the test case.

The other modification to the PDP 11/04 would be to increase the amount of data transferred from the current 512 word limit to 2048 words. These data can be transferred at a rate of approximately:

$$100 \text{ sec} + (20\ \mu\text{sec per word transferred}).$$

CAPS-6 - Two modifications would be made to the flight software in the FCCs. Both modifications would be test case dependent to some extent. The first modification would be the addition of a test execution monitor program that runs in the background mode and uses the excess core memory available to store selected parameters for a particular test case. These

data could then be transferred to the PDP 11/60 for results processing after execution of the current test case was terminated. The otner modification would involve the addition of software in the foreground program to generate an interrupt to the PDP 11/60 upon occurrence of a predefined event. Also, in order to facilitate efficient data transfer between tne CAPS-6 and tne PDP 11/60, specified variables in tne flight software would be assigned to contiguous addresses in the CAPS memory.

The system described above would nave the capability of controlling tne injection of the fault, recording the approximate time of detection, and performing results analysis and report generation. Such a system could be used to determine monitor coverage.

4.2.2 Option Two: Modest System Modifications

Illustrated in Figure 10, the second proposed system consists of tne addition of special purpose hardware that would have the capability of passively monitoring the CAPS transfer bus and recording the bus transactions for analysis by the PDP 11/60. This BMRU would also contain an internal clock for more accurate measurement of fault detection times for the FCCs in the RDFCS. The BMRU would have the capability to receive up to 16 interrupts and decode them to determine their origin. This option would still use tne PDP 11/04 for transferring data from the CAPS memory locations to the PDP 11/60. The following describes the functions and capabilities offered by Option Two.

PDP 11/60 - The PDP 11/60 functions would be similar to those performed under the first option described. The PDP 11/60 would still be responsible for:

- o Airplane simulation
- o Draper FI control
- o PDP 11/04 interface
- o Results processing.

In addition to the above functions tne PDP 11/60 would interface to the new unit described below.

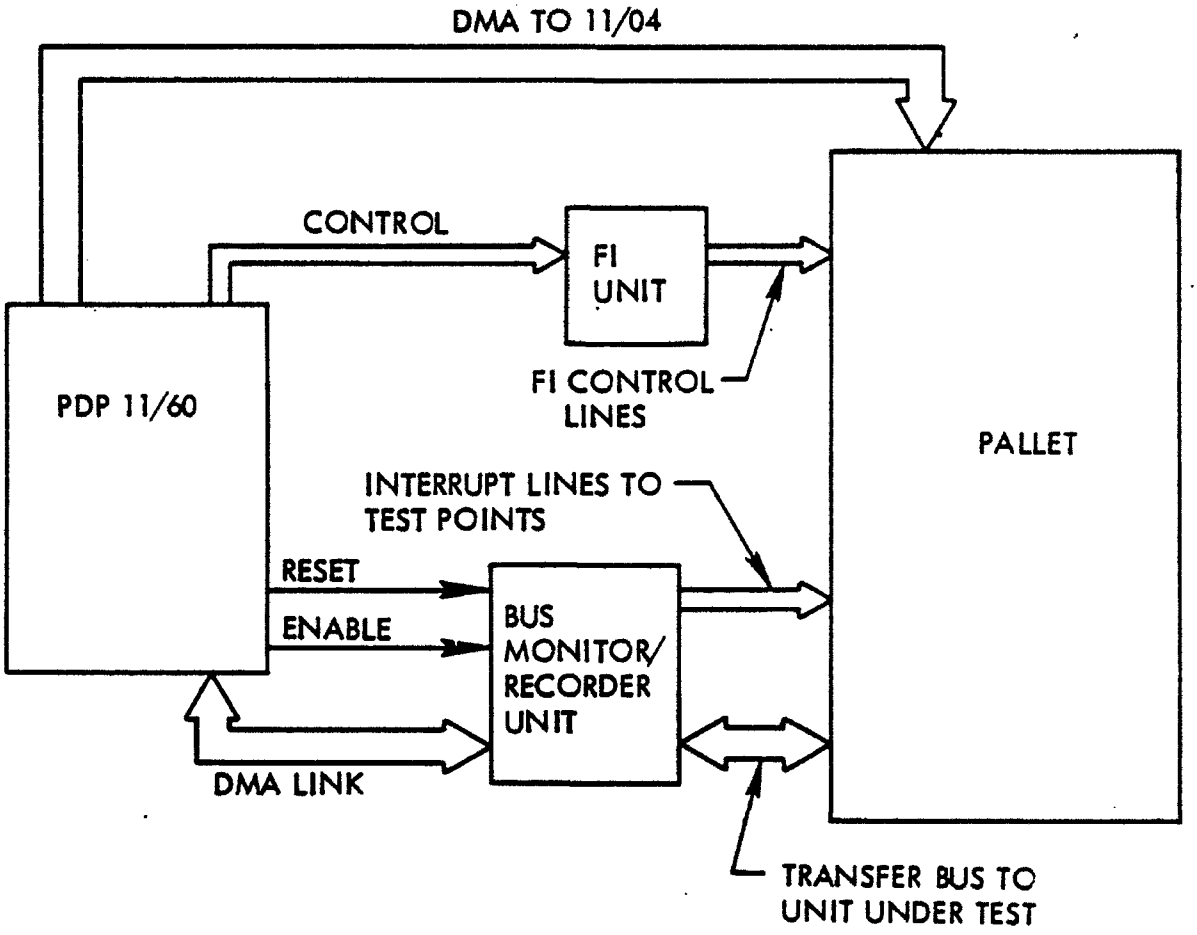


Figure 10. FIIS Option 2 Architecture

Bus Monitor/Recorder Unit - The BMRU shown in Figure 11 would have the capability to passively monitor the CAPS transfer bus and record bus transactions in nigh speed cache memory. The unit would also have a DMA data link to the PDP 11/60 for transferring the data collected by the bus recorder. This information would then be processed by the PDP 11/60 into tabulated data that represent the results in the following hex format:

Address Data Read/Write

The BMRU would also be able to receive up to 16 interrupts from the RDFCS pallet. The BMRU would be able to decode these interrupts to determine where they originated. It would use these in conjunction with its clock to measure fault detection times. An example application might be as follows: upon receiving a signal from the PDP 11/60 that the fault is being in-

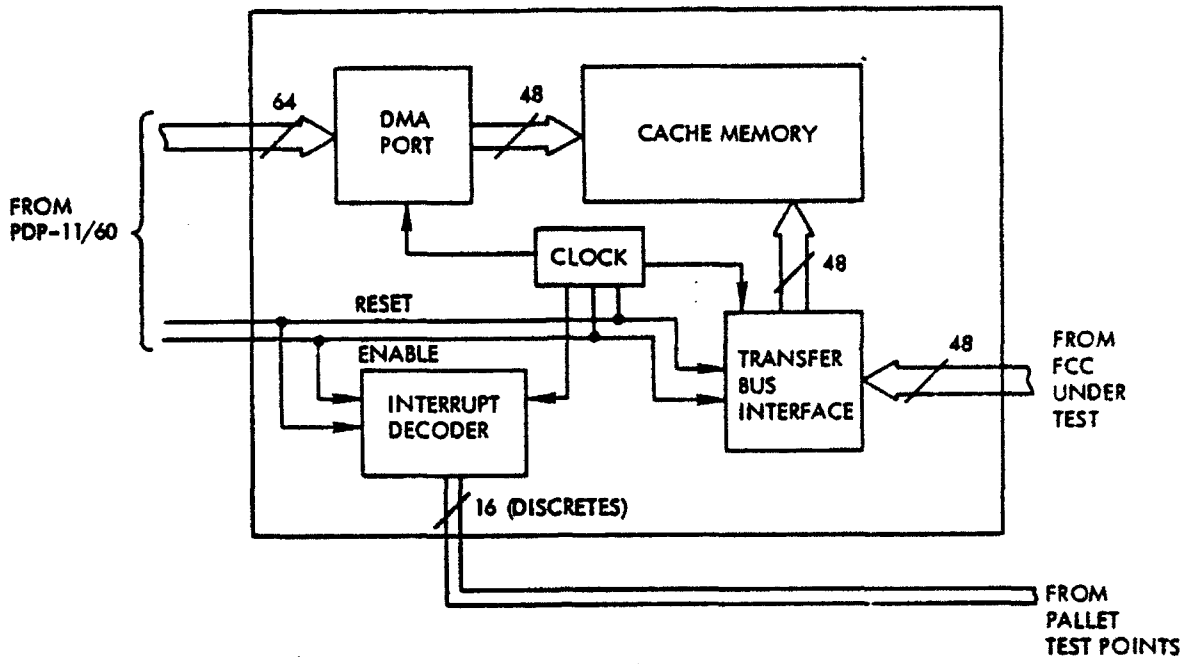


Figure 11. Bus Monitor/Recorder Block Diagram

serted, the BMRU would read its internal clock. Upon receiving an interrupt from the pallet the BMRU would read the clock again, thereby giving the fault detection time. By using more than one interrupt, it would be possible to instrument the comparator outputs, the flight software, and the channel not under test to reconstruct a fault detection sequence. This profile would yield a time history of when the various monitors within the FCCs detected and reacted to the injection of the fault.

Under Option Two, the PDP 11/04 and the CAPS-6 would have the same software modifications made under Option One. With the above outlined features this system would have the capability to inject a fault and to generate a fault detection profile. The latter would be accomplished by instrumenting the various monitors and the other channels with the multiple interrupt capability available. As each interrupt occurred, the BMRU clock would be read so that timing measurements for each monitor could be determined. Additionally, the CAPS bus transactions would be recorded for analysis by the PDP 11/60.

4.2.3 Option Three: Extensive System Modifications

Figure 12 illustrates the third candidate FIIS architecture, which

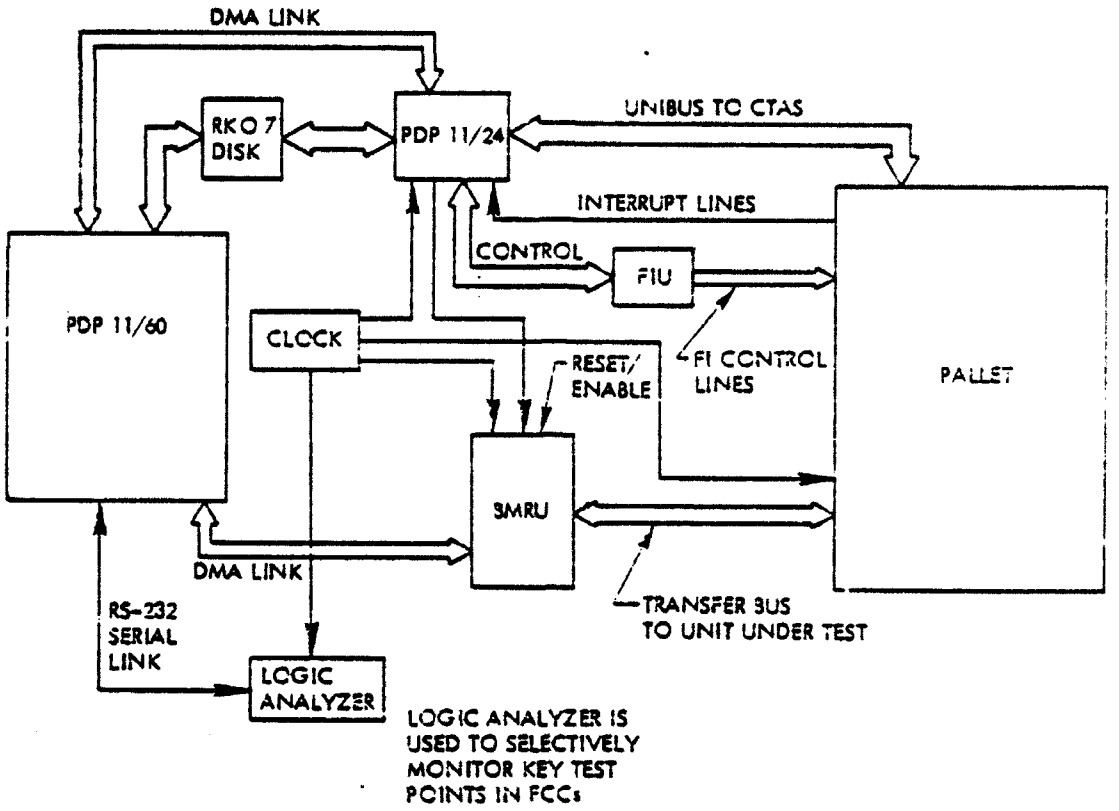


Figure 12. FIIS Option 3 Architecture

provides enhanced performance and test results resolution over Option Two. This architecture would have the capability to continuously monitor the CAPS transfer bus and to store this information for analysis by the PDP 11/60 minicomputer. A PDP 11/24 minicomputer would be added for precisely controlling the CTAs and the FIU, without any higher priority task such as airplane simulation. The PDP 11/24 minicomputer, which is approximately three-and-a-half times faster than the PDP 11/04, would encompass the same capabilities, and would remain under the control of the PDP 11/60.

As an additional aid, it would be possible to add a logic analyzer so that other test points could be monitored on a selective basis. An example of such a test point might be the monitoring of the bus time-out error signal when it is disabled for a particular test case. The logic analyzer would interface to the PDP 11/60 via a RS-232 serial data link for results processing. The final piece of hardware would be the addition of a system clock that interfaces to the various system components for improved time

correlation of the results. The following subsections elaborate the functions performed by major system components under Option Three.

PDP 11/60 - The PDP 11/60 under this option would be responsible for the airplane simulation and results processing. During real-time test case execution, the PDP 11/60 would control only the airplane simulation. This would permit an expanded nonlinear model to be developed that would have the capability of supporting a pilot's chair. The PDP 11/60 would probably still do all of the results processing. The PDP 11/60 would then interrogate the BMRU, and access the shared disk with the PDP 11/24 for test results data to process. The functions to be accomplished during results processing would remain to be determined during detailed test case definition.

PDP 11/24 - The PDP 11/24 would interface to the CTAs, the shared disk with the PDP 11/60, the BMRU, the system clock and the FI. The PDP 11/24 would have the same monitor capabilities as the existing PDP 11/04 plus the control software for the FI. A real-time test case scenario for the PDP 11/24 might be the monitoring of the PDP 11/60 airplane simulation via the DMA data link for a particular variable to initiate the injection of a predefined fault. Just prior to fault injection the BMRU would be reset and started, and then upon acknowledgement that the fault was detected, the PDP 11/24 would transfer the results from the BMRU to the PDP 11/60 along with any information gathered from the CTAs. Depicted in Figure 13, the software to accomplish these functions would be very similar to that described under Option One.

Bus Monitor/Recorder Unit - The proposed BMRU would passively monitor all transactions on the CAPS transfer bus. The information recorded would consist of all data and address lines and the control lines. The unit would be able to buffer 256K transactions in RAM before overwriting its buffer. The BMRU would increment a counter and store the current bus transaction as a function of the instruction fetch signal. The unit would be controlled from the PDP 11/24 by a discrete line that would reset the unit's counter just prior to fault insertion. At the termination of the

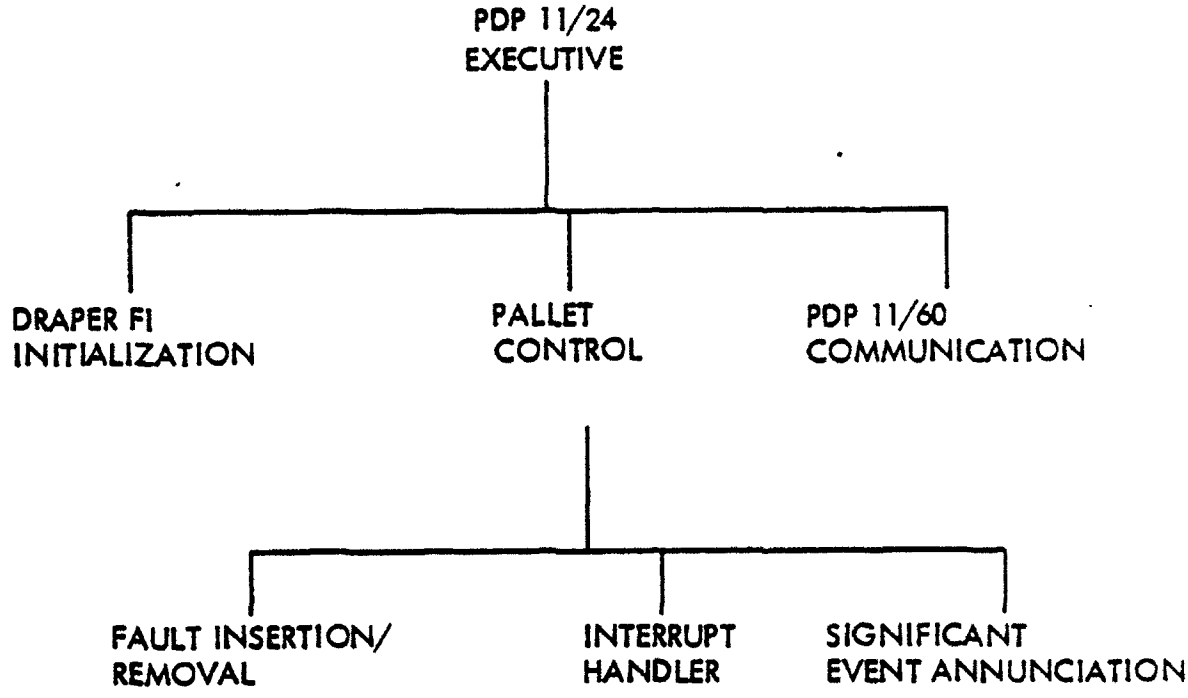


Figure 13. PDP-11/24 FIIS Software Structure

test case, the PDP 11/24 would transfer the data stored in the unit to the PDP 11/60 for results processing.

System Clock - The system clock proposed would be a simple oscillator having outputs that are compatible with the various devices in the RDFCS laboratory. The clock would be capable of being read by the various devices in the system or cleared to zero by a discrete input from the PDP 11/24 (or possibly some other device). Under normal operation the counter stages would be updated at 200 KHz (5 usec) rate. Provisional circuitry would ensure that a count could not be lost while the clock is being read.

Logic Analyzer - The logic analyzer would be used to selectively monitor various pins within the FCCs during testing. The logic analyzer would have the capability to buffer a small number of events and would be able to communicate this information to the PDP 11/60 via a serial data link for results processing. The pins to be monitored would be determined by the test case definition.

Software - As stated above, the software for this system would be very similar to that described under the first architecture. The biggest

difference would be where the programs reside. Table 10 lists the software functions performed by the various computers in the proposed system.

The extended capability offered by this proposed system would allow the CAPS-6 runstream to be analyzed to determine the effect of the injected fault. This would be accomplished by using the data stored by the BMRU and the supplemental information obtained by the logic analyzer. By employing a system clock, improvements in measuring fault detection times could be made. In addition by freeing the PDP 11/60 of the considerable overhead associated with the fault injection unit and managing the CTA data link, new functions could be provided.

TABLE 10 SOFTWARE MODULES FOR OPTIONS 3 AND 4

COMPUTER	PROGRAMS
PDP 11/60	• REAL-TIME AIRPLANE SIMULATION • RESULTS PROCESSING • DMA CONTROL
PDP 11/24	• FI INITIALIZATION/INSERTION • INTERRUPT PROCESSING • SYSTEM CLOCK CONTROL • CTA CONTROL • 11/04 MONITOR FUNCTIONS • DMA CONTROL • BUS MONITOR/RECORDER INTERFACE
CAPS-6	• BACKGROUND TEST EXECUTION MONITOR • GENERATE INTERRUPT TO PDP 11/24 • UTILIZATION OF EXCESS MEMORY

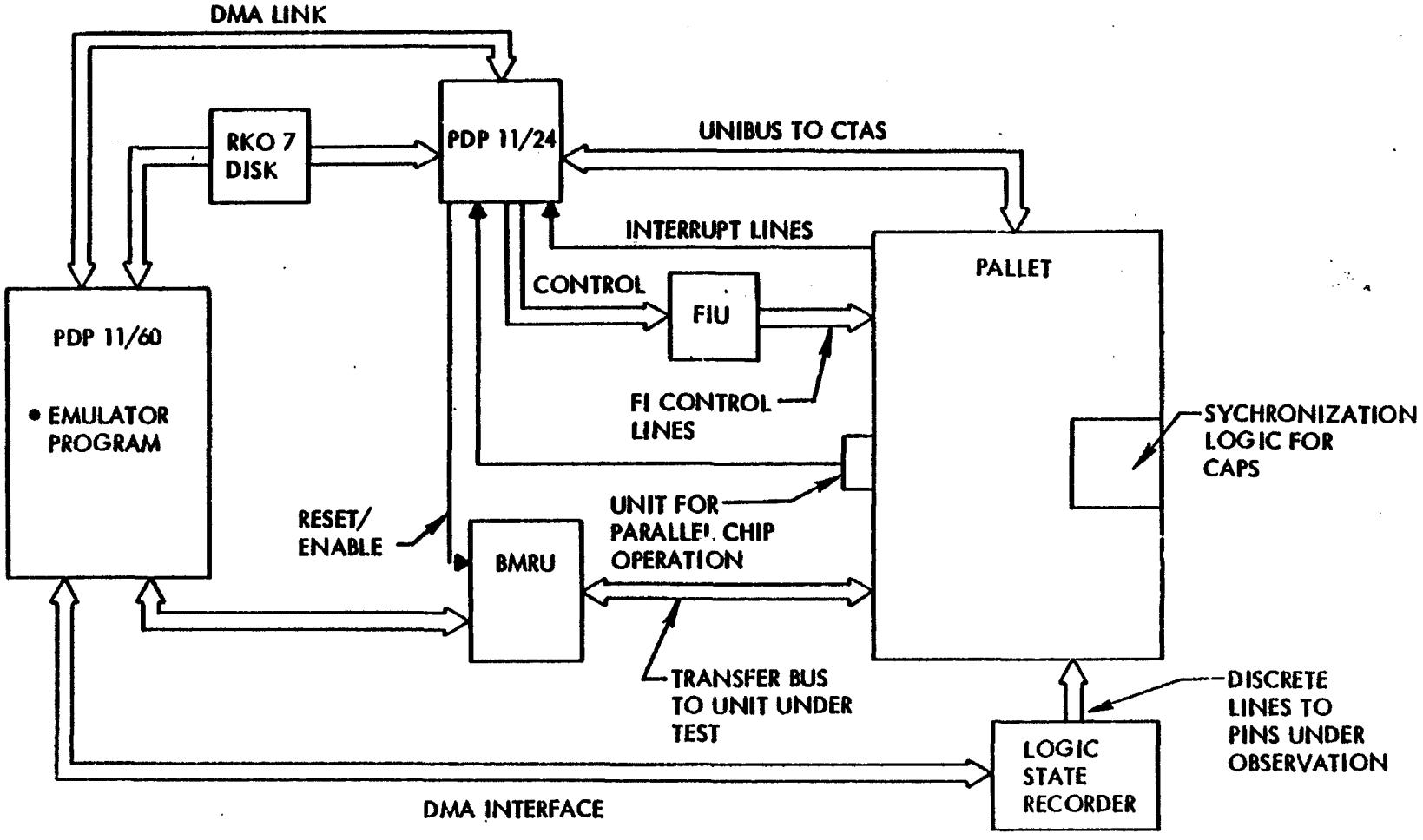
4.2.4 Option Four: Full-Scale System Modifications

The fourth option as illustrated in Figure 14 is the most advanced of the systems proposed. This version encompasses capabilities offered under the previous options and adds the capability of pin-level instrumentation and analysis. With this new feature, it is possible to conclusively determine whether an injected fault results in an undetected error or whether it is a "don't care" condition.

The proposed additions consist of a unit that would allow two identical chips to receive identical inputs, and would then use one of the chips to monitor the other's output for an error condition. In this way an error introduced on the input pins would be conclusively detected if it propogates to the output pins. Another device proposed is a logic state recorder (LSR) which would have the capability to monitor the states of up to 64 pins and buffer the information for the PDP 11/60.

It is also suggested that the four FCC CAPS-6 processors in the pallet might be synchronized by using a single system clock. This would allow faster detection of a propogated error so that the data analysis tasks could be simpler. In addition, the CAPS-6 emulator delivered under NASA Contract NAS2-10270 might be modified for use in results analysis. The following subsections further describe the proposed modifications under Option Four.

Parallel Chip Unit - This proposed unit would have the capability of allowing the chip under test to be paralleled with an identical chip as shown in Figure 15. The signal from the board to one of the chips would be faulted momentarily using the 250 nanosecond one-shot multivibrator, with the other chip receiving the inputs unfaulted. Output comparison would be used to determine whether the fault did or did not propagate to the chip output pins within a reasonable length of time (e.g., 5 sec). Detection of a state difference between the two chips would trigger data recording to begin. Used in conjunction with the logic state recorder described subsequently, the parallel chip unit would permit recording both the normal and the faulty chip output. In turn, this would enable detailed study of the propagated fault effects.



NOTE: SYSTEM CLOCK NOT SHOWN FOR CLARITY

Figure 14. FIIS Option 4 Architecture

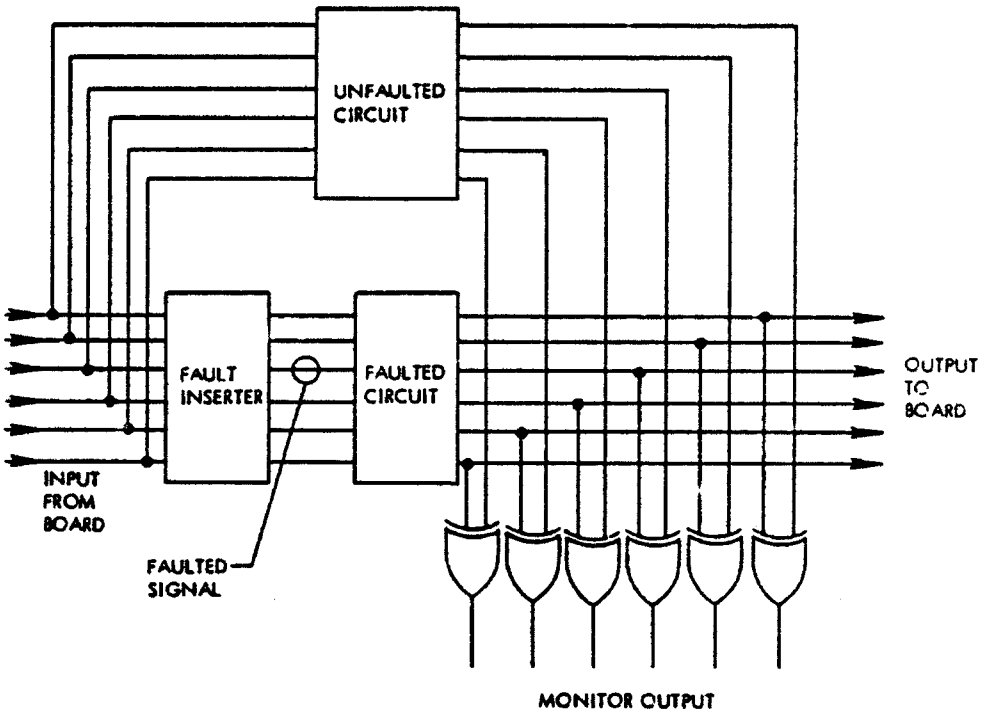


Figure 15. Parallel Chip Unit

Logic State Recorder - The proposed LSR would be a specialized version of the type are sold commercially. It would have the capability to monitor and record the states of up to 64 pins. The LSR could be triggered either externally or by some logical combination of the input pins. The unit would contain internal cache memory of sufficient capacity to buffer the data collected for transfer to the PDP 11/60 for analysis.

Synchronous CAPS-6 Operation - It is proposed that the CAPS-6 processors in the flight control computer might be synchronized by connecting them to a common clock. This capability would aid in more conclusive detection of error conditions by on-line comparison of identical channels. This would result in improvements in measured fault detection times and would help bound the amount of data collected after a fault has been detected.

Emulator Modification - The existing CAPS-6 emulator might be used as an aid in the analysis of the results gathered during testing. With appropriate modifications, it would be possible to perform fault injection and analysis at the microcode level. Through the combined use of the emulator

and the processor pin state information collected during testing it would be possible to perform a complete analysis of the test case results. Methods would have to be examined to automate this process through use of either the PDP 11/60 or the UNIVAC system that hosts the CAPS-6 support software.

FCC Memory - Another issue that should be considered is the difference in the type of memory used in the FCCs in the RDFCS and the units that are flown on an airplane. The FCCs in the RDFCS use core memory for the flight program, and the production system uses ROM. Certain failure modes can cause the CAPS-6 to overwrite sections of the core memory, thereby possibly invalidating or obscuring the test results. This problem could be overcome in one of two ways.

The RDFCS could be loaded with PROM memory that has had the current software under test burned-in. This could be accomplished by using the PROM programmer unit available at RDFCS facility. The associated process is slow, but if the number of test cases between the reprogramming of the PROMs were large, this might be a cost effective appproach.

The other possible solution might be to add logic to the Plessey core memory units in the RDFCS so that the sections containing the flight program could be write-protected. This would involve considerable hardware modification, but would result in greater productivity in applying test cases.

4.3 SUMMARY AND CRITIQUE OF RECOMMENDATIONS

The four FIIS architecture options are summarized in Table 11 in terms of associated modifications. This facilitates comparison of implementation requirements. Table 12 is a synopsis of capabilities offerred by the respective options, along with a tentative approximation of relative costs. FIIS investigation concerns and related features are presented in Table 13.

TABLE 11 FIIS OPTION ALLOCATION MATRIX

FUNCTIONAL ALLOCATION

FIIS VERSION	11/60 RESULTS PROCESSING	11/04 MONITOR CAPS MEMORY	11/60 CONTROL OF FIU	11/24 CONTROL OF FIU	BUS MON/RECORD UNIT	INTERRUPT TO 11/60	INTERRUPT TO BMRU	CAPS BACKGROUND PROGRAM	LOGIC ANALYZER	IMPROVED TIMING (SYS. CLOCK)	LOGIC STATE RECORDER	PARALLEL CHIP UNIT	SYNCHRONOUS CAPS OPERATION	EMULATOR ENHANCEMENT
OPTION 1	•	•	•			•		•						
OPTION 2	•	•	•		•		•	•	•	•				
OPTION 3	•			•	•		•	•	•	•				
OPTION 4	•			•	•		•	•	•	•	•	•	•	•

TABLE 12 COST/BENEFITS PROJECTIONS

ARCHITECTURE	CAPABILITIES	COST *
OPTION 1	• FAULT TIMING MEASUREMENTS (KOM) • PIN-LEVEL TRANSIENT FAULTS (~ 6 μsec) • FAULT DETECTION HISTORY	1.0
OPTION 2	• IMPROVED FAULT TIMING MEASUREMENTS • FAULT DETECTION TIMING PROFILE • CAPS BUS TRANSACTION ANALYSIS	1.5
OPTION 3	• IMPROVED FAULT TIMING MEASUREMENTS • EXPANDED AIRPLANE MODEL • LOCICAL ANALYZER FOR INCREASED INSTRUMENTATION • BUS TRANSACTION ANALYSIS	2.1
OPTION 4	• CONCLUSIVE DETERMINATION OF FAULT PROPAGATION • STATISTICAL TESTING • COMPARISON OF TEST RESULTS TO EMULATOR OUTPUT	5.0**

NOTES:

* COSTS ARE NORMALIZED TO OPTION 1

** DOES NOT INCLUDE COST OF SYNCHRONIZING CAPS PROCESSORS

TABLE 13 SUMMARY OF RELEVANT FIIS FEATURES

CONCERN	CIRCUITS AFFECTED	RELEVANT FIIS FEATURES
PERMANENT PIN-LEVEL FAULTS	ALL	FAULT SELECTION, INSTRUMENTATION
TRANSIENT PIN-LEVEL FAULTS	ALL	INSTRUMENTATION, CONTROL OF FAULT DURATION
PATTERN DEPENDENT FAULTS	MICROPRO-CESSORS, INTERRUPT CONTROLLER	250 nsec ONE-SHOT
SINGLE BIT FAULTS	CONTROL STORE	250 nsec ONE-SHOT
FAULT PROPAGATION	ALL	INSTRUMENTATION, RDFCS ENVIRONMENT, PARALLEL CHIP UNIT
FAULT DETECTION	ALL	INSTRUMENTATION, RDFCS ENVIRONMENT

REFERENCES

1. "System Design Analysis," FAA Advisory Circular No. 25.1309-1, September 7, 1982.

2. "Government/Industry Workshop on Methods for the Certification of Digital Flight Controls and Avionics," NASA TMX-73, 174, October, 1976.

3. Hitt, E. F., et al.: "Validation of Digital Systems in Avionics and Flight Control Applications, Handbook-Volume 1," DOT/FAA/CT-82/115, U.S. Department of Transportation, December 1982.

4. Ness, W. G., et al.: "Integrated Assurance Assessment of a Reconfigurable Digital Flight Control System," DOT/FAA/CT-82/154, October 1982.

5. Mulcare, D. B., W. G. Ness, and R. M. Davis: "Digital Flight Control System Validation Technology Assessment," DOT/FAA/CT-82/140 or NASA CR-166374, July 1982.

6. Davis, R. M., D. B. Mulcare, and W. G. Ness: "Validation-Oriented Development of a Quadruplex Digital Flight Control System," NAECON, May 1983.

7. McGough, J. G., and F. L. Swern: "Measurement and Fault Latency in a Digital Avionic Mini Processor," NASA CR3462, October 1981.

8. Lala, J. H., and T. B. Smith: "Development and Evaluation of a Fault-Tolerant Multiprocessor (FTMP) Computer - Vol, III, FTMP Test and Evaluation," CSDL-R-1602, November 1982.