F-31-27

SAE The Engineering Resource For Advancing Mobility®

400 COMMONWEALTH DRIVE, WARRENDALE, PA 15096

AEROSPACE
RECOMMENDED
PRACTICE

Submitted for recognition as an American National Standard

SAE ARP 1834

Issued 8-7-86

ARP 1834
FAULT/FAILURE ANALYSIS

FOR DIGITAL SYSTEMS AND EQUIPMENT

PREPARED BY
SAE COMMITTEE S-18A,
FAULT FAILURE ANALYSIS PROCEDURE FOR DIGITAL SYSTEMS

SAE Technical Board Rules provide that: "This report is published by SAE to advance the state of technical and engineering sciences. The use of this report is entirely voluntary, and its applicability and suitability for any particular use, including any patent infringement arising therefrom, is the sole responsibility of the user."

SAE reviews each technical report at least every five years at which time it may be reaffirmed, revised, or cancelled. SAE invites your written comments and suggestions.

Printed in U.S.A.

TABLE OF CONTENTS

APPENDICES

FIGURES

TABLES

1. INTRODUCTION:

1.1 Background: A fault and failure analysis (F/FA) usually consists of one or more of the analysis processes depicted by Figure 1-1, "Family of Fault/ Failure Analysis Processes." Typically, these analysis techniques are for the purpose of:

- Analyzing, assessing and documenting the effects of potential failures on a particular equipment item or system design;
- Identifying those failures which affect operational success or safety, and determining their probability of occurrence;
- Enabling quantification of fault/failure detection and isolation capability as it relates to equipment safety and maintainability.

Aerospace Recommended Practice ARP 926A, "Fault/Failure Analysis Procedure", addresses the application of F/FA methods to parts, components and systems other than those of an essentially digital nature. The development of this separate ARP to address digital F/FA was initiated to recognize:

- The expanding use of digital hardware in military, commercial, and consumer products;
- The need to apply F/FA procedures to digital devices, components and systems;
- The distinctive characteristics of digital equipment, having unique potential failure modes which, if not recognized and designed for, can result in either excessive downtime or erroneous output with severe ramifications.

1.2 Scope and Purpose: ARP 1834 provides general guidance for the selection, approach to, and performance of various kinds of F/FA of digital systems and equipment. Its prime objective is to present several industry-acceptable, cost-effective methods for identifying, analyzing, and documenting digital-equipment failure modes and their effects. The analysis techniques and considerations presented here are directed to digital-equipment hardware faults and failures exclusively.

ARP 1834 is not intended as an exhaustive treatment of the enormously complex process involved in the analytical failure evaluation of complete digital systems, nor as a universally applicable, definitive listing of the necessary and sufficient steps and actions for such evaluation.

ARP 1834 addresses the following areas of consideration in the preparation and performance of F/FA's for digital equipment:

Possible Analysis Approaches: Top-Down and/or Bottom-Up (Sect 2)

Fault/Failure Modes, as they affect equipment operation and performance (Sect 3)

1.2 (Continued):

Fault Monitoring Methodology: Reasons for, types of, and effectiveness (Sect 4)

Analysis Methods: Preparation for, types of, effectiveness and coverage (Sect 5)

1.3 Use of ARP 1834 Guidelines for Safety Certification: If this document is used as guidance for analyses involved in achieving digital-equipment safety certification by a regulatory agency, early coordination with that agency should be initiated to establish the scope and level of analysis effort that will be required to show compliance. Specific applications of F/FA processes discussed herein (and quite possibly others omitted here) will need to be negotiated on a case-by-case basis between the applicant and the agency, and between the prime contractor and his subcontractor or equipment supplier.

For digital systems performing functions that are critical and/or essential (see 2.2.1), it may not be possible to demonstrate compliance with safety-certification requirements without the use of design techniques aimed at producing a fault-tolerant system. A goal for these design techniques is the possible reduction or elimination of the need for part-level FMEA. This consideration is of pivotal importance, because thorough, accurate and dependable FMEA of contemporary microcircuits is not a feasible undertaking (see 5.1.3.6.1). The depth of the F/FA required to show compliance will be strongly influenced by such techniques. Typical design techniques which may be used in various combinations include:

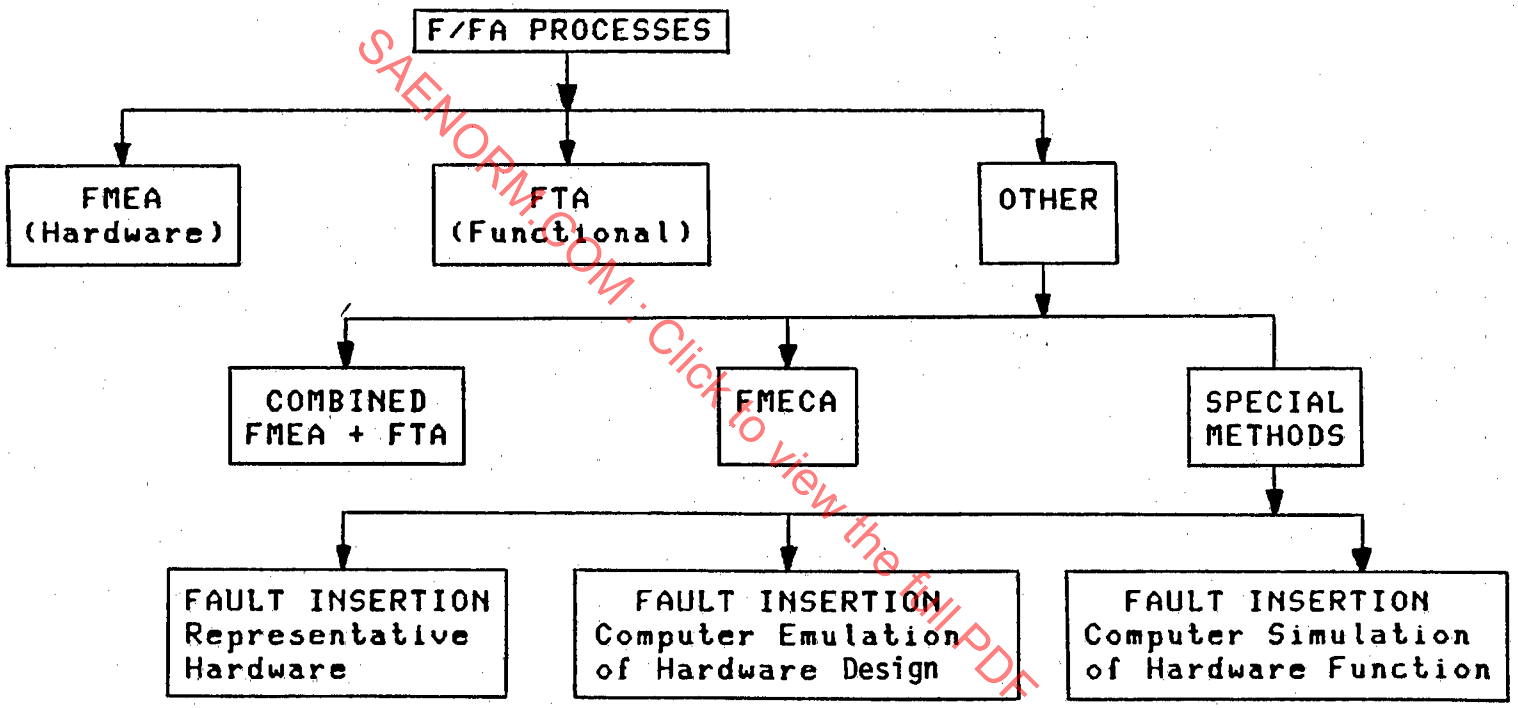
- System Architecture
  - Similar Redundancy
  - Dissimilar Redundancy
  - Signal Consolidation or "Voting"
  - Hardware Functional Partitioning

- Fault Detection and Isolation
  - Comparison Monitoring between redundant elements
  - In-line test and monitoring
  - In-line reasonableness checks

- Fault Response
  - System reconfiguration
  - Operational mode changing
  - System shutdown

Although such design considerations are outside the scope of this document, they must be taken into account by system designers and analysts in meeting overall system-safety objectives and in establishing the level of effort required for the F/FA.

2. POSSIBLE APPROACHES: The prime criterion for performing any of the F/FA processes depicted by Figure 1-1, should be to provide credible results in the most cost effective manner. Of the various factors and influences which affect success or failure per this criterion, the phase of development at which the F/FA is expected to be performed and the analysis objectives are probably the most significant. Table 2-1 reflects how the iterative nature of F/FA, as the design progresses, can help to accomplish these F/FA objectives in a cost effective manner.


F/FA PROCESSES
FMEA (Hardware)
FTA (Functional)
OTHER
COMBINED FMEA + FTA
FMECA
SPECIAL METHODS
FAULT INSERTION Representative Hardware
FAULT INSERTION Computer Emulation of Hardware Design
FAULT INSERTION Computer Simulation of Hardware Function

Where:
FTA = Fault Tree Analysis
F/FA = Fault/Failure Analysis
FMEA = Failure Mode and Effects Analysis
FMECA = Failure Mode, Effects and Criticality Analysis

Figure 1-1

Family of Fault/Failure Analysis Processes

TABLE 2-1

F/FA Objective versus Development Phase

DEVELOPMENT PHASE	OBJECTIVE
Preliminary Design	To find design weaknesses, single point failures and potential hazards
Design/Development	To assess safety, reliability, maintainability & availability
Design Acceptance	To substantiate that equipment meets requirements

Factors which influence the particular F/FA approach to be utilized in conjunction with digital equipment, can, for discussion purposes, be categorized under the following headings:

- Influence Factors

  Common to all systems F/FAs

  Non-processor-based system F/FAs

  Processor-based F/FAs
- Factors affecting scope and approach of F/FAs
- Approach considerations and comparisons for digital system F/FAs
- Decision tree for civil aircraft F/FAs

2.1 Influences versus System Types:

2.1.1 Common to all System F/FAs: There are two basic approaches to performing any F/FA: the top-down approach and the bottom-up approach. (ARP 926A calls these two the "functional approach" and the "hardware approach," respectively.) As shown in Figure 1-1, top-down or bottom-up analysis or both may be utilized in conjunction with any F/FA. Normally the choice of approach is based on considerations of clarity, emphasis, program phase, equipment-development life cycle, etc.

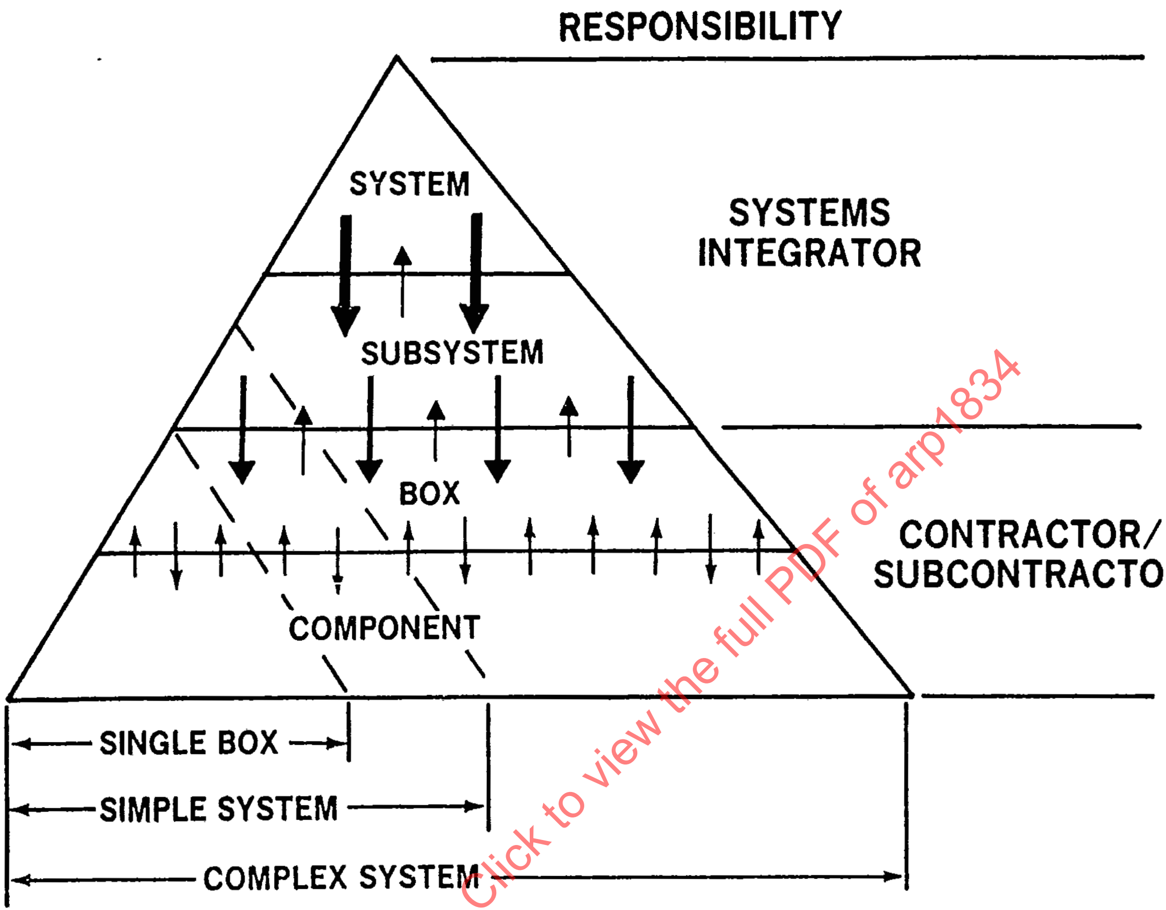
2.1.1 (Continued):

The top-down analysis is initiated at the system level, or top-assembly level, and proceeds downward through the hardware design (see Figure 2-1). Failure effects are identified in terms of malfunction of the system or equipment-item or of the loss of one of its functions, rather than in terms of lowest-level-failure impact. Having identified the undesirable effects, this analysis determines if failures can exist at the lower equipment levels that can cause the undesirable effects. Fault-Tree Analysis is an example of top-down analysis.

The bottom-up analysis is initiated at the electronic-part or component level or at a higher, intermediate equipment level and proceeds upward to the top equipment or system level as shown in Figure 2-1. At the initiated level, every credible hardware failure mode is postulated, and the failure effects of each such failure mode on the next-higher equipment level are identified. (In some such analyses, the failure rate and exposure time of each failure mode are also assigned.) Failure Modes and Effects Analysis (FMEA) is an example of a bottom-up analysis.

For any equipment employing microcircuits, completion of a bottom-up analysis initiated at the electronic-part level is not ordinarily achievable, because (as discussed in 5.1.3.6.1) a thorough, accurate, and dependable failure-mode analysis of most microcircuits is not feasible.

ARP 926A contains examples illustrating the mechanics of certain basic analyses. Additional examples illustrating the use of these basic approaches on digital systems are provided in this document.



NOTES:

1. THE SIZE AND QUANTITY OF THE ARROWS REFLECT THE DEPTH AND SCOPE OF THE ANALYSIS

2. THE DIRECTION OF THE ARROWS INDICATES THE DIRECTION OF THE ANALYSIS (TOP DOWN, BOTTOM UP)

3. THE AREAS OF RESPONSIBILITY WILL GENERALLY OVERLAP MORE THAN INDICATED

FIGURE 2-1
TYPICAL ANALYSIS FLOW--SCOPE, DIRECTION & RESPONSIBLITY

2.1.2 Non-Processor-Based System F/FAs: The basic approaches described in ARP 926A may be used for simple, non-processor-based digital systems. The more complex non-processor-based digital systems will probably require the application of techniques described in this document.

2.1.3 Processor-Based System F/FAs: The basic F/FA approaches described for non-processor-based digital circuits may be extended to processor-based systems. Typically, analyses are performed using the functional approach. Where it is possible, the processor circuit is divided into various basic functional elements. The function failure effects are in either case, however, generalized as loss of function or incorrect function. In the latter situation, extreme care must be taken whenever an incorrect function is assigned to a processor-based system. The processor may generate new and possibly undesirable functions if left uncorrected.

The effect of these processor failure modes on the system is determined by evaluating the fault monitoring and response provisions in conjunction with software design and hardware implementation.

For processor-based systems, the functional approach may be effective if the effect on the system due to a failure is dependent on the fault monitoring, the effectiveness of which can be evaluated. With procuring or regulatory agency concurrence, it may not be necessary to perform analysis beyond the point of determining that critical and essential function failure modes are detectable by fault monitoring, plus verifying that the desired response will follow. It may be only necessary to initiate the hardware F/FA approach at the lowest level subject to analysis for those cases where critical and essential function effects are determined or where there is a need to quantify the probability of any undesired event.

Special methods may be needed to supplement the basic analytical hardware and functional F/FA approaches where the effects of failures are uncertain. These techniques may use actual hardware, an emulation of the hardware, or a simulation of the hardware functions. Failures may be simulated by one of the following methods:

- In the target computer, replace functioning hardware with failed equivalent hardware which may contain individual failed gates, or

- Inject faults at individual device pins in the target computer, either manually or automatically, or

- Inject faults into a digital computer emulation of the target computer, or

- Inject faults into a digital computer simulation of the function.

Section 5.2 contains a description of these special methods.

2.2 F/FA Scope and Approach: Factors that directly affect the required level and scope of analysis include:

- Failure consequences
- System Architecture (redundancy, function isolation),
- Fault Management (monitoring techniques, effectiveness, comprehensiveness), and
- Maintainability considerations.

Table 2-2 deals with different aspects of the scope of, and approach to, the analysis.

2.2.1 Failure Consequences: A fundamental principle applicable to all equipment is that the scope of the selected analysis approach should depend upon the severity of the worst possible failure consequences.

For commercial airborne systems and equipment, a categorization of function criticality was introduced by RTCA Document No. DO-178 for software, implied by Part 25.1309(b), of the Federal Aviation Regulations, recognized by Advisory Circulars AC 20-115 and AC 25.1309-1. The criticality categories with associated quantitative levels are shown in Table 2-2 and defined by the FAA in AC 25.1309-1 as:

- Non-Essential -- Functions whose failures would not contribute to or cause a failure condition which would significantly impact the safety of the airplane or the ability of the flight crew to cope with adverse operating conditions. Airplane conditions which result from improper accomplishment or loss of non-essential functions may be probable.
- Essential -- Functions whose failures would contribute to or cause a failure condition which would significantly impact the safety of the airplane or the ability of the flight crew to cope with adverse operating conditions. Failure conditions which result from improper accomplishment or loss of essential functions must be improbable.
- Critical -- Functions whose failure would contribute to or cause a failure condition which would prevent the continued safe flight and landing of the airplane. Failure conditions which result from improper accomplishment or loss of critical functions must be extremely improbable.

2.2.2 Architecture: If the architecture (the arrangement and interrelationship of components) of the system is such that multiple outputs are (for the most part) isolated from each other by utilizing separated, independent circuits, the parts level F/FA task could be fairly simple and limited in scope. However, if the circuits are highly interactive, then the analysis

Table 2-2

One Example of Categorizing
Scope and Approach of Analysis

CRITICALITY LEVEL	ANALYSIS - SCOPE & APPROACH (may be required)	ASSOCIATED FAILURE CONDITION PROBABILITY OF OCCURRENCE (Ref. FAA AC 25.1309-1)
Critical	Prelim. Hazard Analysis Fault Tree Analysis FMEA Probability Analysis (supporting FTA)	Extremely Improbable (1 x 10^{-9} or less)
Essential	FTA FMEA	Improbable (1 x 10^{-5} or less)
Non-Essential	Analysis to determine that system performs only non-essential functions; determine if failure could contribute to a failure condition involving an essential or critical function.	May be probable (1 x 10^{-5} or greater)

2.2.2 (Continued):

task could be difficult and lengthy. This is particularly true when it is necessary, because of failure mode criticality, to tabulate and analyze every credible failure mode in the Line Replaceable Unit (LRU).

When the architecture of the system utilizes a higher degree of independent, non-interactive elements, the failure modes become easier to analyze.

2.2.3 Fault Management: Fault management includes: detection, response, storage and display.

There are a number of possible hardware and/or software means for implementing digital fault monitoring. In any given application, it is a design responsibility to select fault monitoring techniques which comply with the equipment specification safety requirements.

If the system architecture is so configured as to rely heavily on in-line monitoring (as opposed to comparison of redundant signals) to reduce risk of adverse effects associated with failures, then some method of fail-safe monitoring may need to be employed. Any such special methodology will expand the scope of analysis required. If the fault monitoring is designed, however, to detect specific failures, a proportionate reduction in F/FA scope (that is, numbers of combinations of failures which need to be analyzed) can be achieved, provided that the fault monitoring is successfully shown to perform its intended function, and does not cause unnecessary loss of function. The effects of failures of fault monitoring must also be analyzed.

2.2.4 Maintainability Considerations: The scope of F/FA may be dictated by contractual requirements for minimum proven levels of fault detection coverage. These requirements are intended to reduce life cycle costs by correct identification of faulty assemblies at various levels of repair.

2.3 F/FA Approach Considerations: Table 2-3 provides factors to be considered when selecting an F/FA approach. The following discussions are keyed to those considerations.

2.3.1 Program Phase: This refers to the design phase in which an approach may be initiated. The bottom-up approach cannot be started until the hardware/circuits are defined. The top-down or functional approach can be initiated earlier in the program; actually, anytime after a concept has been formed, including the basic design development through tradeoff analysis.

Table 2-3

Digital Systems Considerations of F/FA Approaches

CONSIDERATION	BOTTOM-UP APPROACH	TOP-DOWN APPROACH
1. Program Phase	Circuit Description to Product Release	No limit - Conceptual Phase to Product Release
2. Level of Detail versus Cost	More exhaustive for components selected for Analysis	Relative to System Requirements
3. Skill Level, Expertise Required	Primarily in circuit design of technology utilized	Primarily in systems aspects of technology utilized
4. Facility Special needs	No special facilities required	No special facilities required
5. Fault Management	Assists in quantifying fault isolation	Assists in identifying need for fault isolation at intermediate levels
6. Software Design	Helps to evaluate Software implementation	Helps to evaluate Software scope
7. Safety Hazard Identification	Supports Analysis of Identified Hazards	Hazard identification at system level
8. Design Changes	Less Tolerant of changes; more likely to require amended F/FA	More Tolerant of changes; less likely to require amended F/FA

2.3.1 (Continued):

For processor-based digital hardware, software requirements become an additional consideration to be defined before the F/FA is started. The functional F/FA approach can be used in conjunction with a conceptual hardware design to help define the software safety checking requirements, in conjunction with system architecture and hardware fault monitoring.

2.3.2 Level of Detail versus Cost: For both the hardware and functional F/FA approaches, costs are related to the level of detail or depth of analysis. Processor based systems are time-based and, as such, exhibit operating state changes according to their internal clocking and state processor algorithms. Serious processing errors can occur as a result of one failure in a particular state lasting an extremely short interval of time in one of the thousands of active digital elements. Tasks being performed are so complex that some errors may be undetected for long periods of time. Therefore, it may not be possible to predict processor failure effects at its output pins or to define how these events may be manifested at the output ports.

The better the fault management in the design, the less detail required in the F/FA. It may be more cost-effective to use the functional F/FA approach initially and expand the analysis on failure modes determined to be critical. The expansion of the analysis would then use the hardware F/FA approach.

2.3.3 Skill Level, Expertise Required: The hardware F/FA approach requires a circuit design skill level comparable to the technology utilized. The functional F/FA approach requires system level expertise and an overall familiarity with the design objectives. For processor based digital hardware, these approaches require an overall knowledge of the design, software and fault management requirements.

2.3.4 Facility - Special Needs: The two basic F/FA approaches do not require any special facilities. Other methods, as described in Section 5, require special facilities.

2.3.5 Fault Management: Modern systems, particularly processor based systems, usually include a significant amount of fault management. Fault management covers the various aspects of fault detection, fault response, fault storage, fault display, and possibly fault correction. A system with these built-in attributes can be analyzed in a completely different fashion than one without them, or with only rudimentary capabilities. The process becomes one of analyzing the effectiveness of the designed-in fault management capability rather than trying to postulate the effects of specific faults.

2.3.5.1 Fault Detection: Fault detection is basically performed by defining the allowable behavior at various points in the system in the non-faulted condition and then either monitoring continuously or testing at intervals, or on demand, that these conditions have not been violated. The analysis must confirm that failures are indeed detected.

2.3.5.1 (Continued):

In many cases, a localized bottom-up approach from postulated failure to the next higher detection node will be needed. However, the task is usually simplified because the specific behavior in the faulted condition is not of interest, only the fact that it is distinguishable from the non-faulted behavior. Usually, this may be accomplished by performing tests under known and controlled conditions. For example, an A-to-D converter can be tested at a time when it is off-line by inserting a known input and looking for the expected output. Any other output indicates a failure.

2.3.5.2 Fault Response: The built-in fault response, (i.e., the designed reaction to a detected fault) depends on the specific purpose. Requirements for fault detection usually fall into three categories: safety, operational, and maintenance:

- Safety requirements will usually involve high level system operations; reconfiguration, signal selection, mode changing or even disconnect in some cases. Secondary safety responses may include relevant annunciations to the flight crew of primary responses.

- Operational requirements most commonly involve system status indications required by the flight crew for making decisions such as: whether to dispatch or not; dispatch is allowed but only degraded operation is available; etc.

- Maintenance requirements involve such functions as long term (non-volatile) storage and output (data bus or built-in display) of maintenance related data.

In each case, the analysis must confirm that:

- The required response(s) to a specific detected fault does indeed occur, and

- The designed response(s) to a specific detected fault is appropriate.

2.3.5.3 Fault Storage & Display: In most cases, the system design will require short term (volatile) and/or long term (non-volatile) storage of fault data in order to implement the required responses.

Often local (built-in) or remote fault display is also required.

The analysis must include consideration of failures of the storage and display elements of the system.

2.3.6 Software Design: Software should be considered from three main aspects, which are:

1. How hardware failures can affect the execution of the operational program.

2.3.6 (Continued):

2. How hardware failures can affect the functional isolation of critical functions.

3. How the Fault Management is designed, particularly those features which are performed via the operational program.

2.3.6.1 Effects on Program Execution: A typical CPU breaks down into functional elements, as follows:

- Arithmetic Logic Unit (ALU)
- Read Only Memory (ROM)
- Random Access Memory (RAM)
- Interrupt Control
- Timing and Control

A comprehensive analysis of the effects of all definable failure modes of the above elements on the system behavior under all conditions is impractical. However, the effects of some failures, under some conditions, can be analyzed.

For example, it may or may not be possible to completely trace all the effects of a failure which result in an incorrect state of one bit of a Program ROM. If the affected bit is part of a dedicated data constant, the resulting arithmetical errors may be analyzable; however, if the affected bit is part of an instruction, some of the effects will probably be unpredictable. In either case (or any other ROM bit failure), the resulting change in a ROM checksum is a predictable event.

For other failures, it may be necessary to rely on more indirect evidence that a failure affecting software execution has occurred.

The use of built-in self test techniques, such as a periodic execution of specific instructions with controlled inputs, is one example. Another, in a wider sense, is the use of a Watch Dog Timer (sometimes referred to as a heart-beat monitor) to check the program execution time against an independent clock.

The analyst must determine whether failures of the hardware, which can affect program execution, do indeed result in effects which, although not necessarily predictable, are nevertheless detectable by one or more of the fault monitoring techniques applied.

Any residual postulated faults; i.e., possible faults for which detection is uncertain, will probably have to be subjected to testing by one of the special methods described in Section 5.2.

2.3.6.2 Effect of Failures on Functional Isolation: Functional isolation of operational program elements, so that they do not cross physical storage device boundaries, is fairly common in the implementation of digital systems.

Part of the software consideration for a F/FA consists of analyzing the functional relationships of various program segments and a comparison with the storage patterns in the physical devices to ensure that failures in a single storage device do not affect more than one critical function.

2.3.6.3 Fault Management Design: It is important that anyone analyzing a digital system be generally familiar with the system requirements and how these are satisfied by the software design. However, in one area, that of fault management, it is imperative that the analyst develop a good understanding in the detailed specific requirements and the step by step instructions which, when executed, implement the fault management design.

This knowledge is essential for a proper evaluation of:

- Failure effects, including their detectability, and
- The ability of the system to respond correctly.

2.3.7 Safety Hazard Identification: The functional approach provides the ability to identify potential hazards. The hardware approach may be required to support analysis of potential hazards, (identified by the functional approach) for meeting acceptable probability levels.

2.3.8 Design Changes: The validity of a completed F/FA analysis is sensitive to follow-on hardware or software changes, and depends on the analysis approach used. In general, the bottom-up F/FAs are more sensitive to design changes since they recognize changes in hardware elements, whereas the top-down F/FAs are based upon broader functional groups.

2.4 F/FA Decision Tree: Figure 2-4 is a flow chart depicting the key tasks and decisions involved in selecting the fault/failure analyses typically required for safety certification of aircraft digital electronic systems.

To minimize complexity, the chart does not address the situation where the completed analyses show that the system will not meet the safety criteria. It is likely, for example, that if the system lacks full redundancy, many essential systems and most critical systems will fail to meet the specified criteria. In that situation, system redesign with revised analyses would be necessary.

If system failures do not involve essential or critical functions, including providing misleading information to the flight crew, further analysis is normally not required.

2.4 (Continued):

If any of the system failures do involve essential or critical functions, the next question is whether sufficient redundancy is provided or other design techniques employed (see paragraph 1.3). If an FMEA can verify the redundancy, adequate failure detection and acceptability of the failure effects, then further analysis may not be required.

On the other hand, if either the redundancy or other design techniques are shown to be lacking or inadequate (by the hazard analysis) or the FMEA fails to verify that the failure effects are acceptable, then a fault tree or other analysis is also necessary. If the analysis shows that the probability of occurrence of each failure condition is acceptable (relative to the criticality category of the failure effect) then further analysis may not be required.

3. FAILURE MODES & EFFECTS:

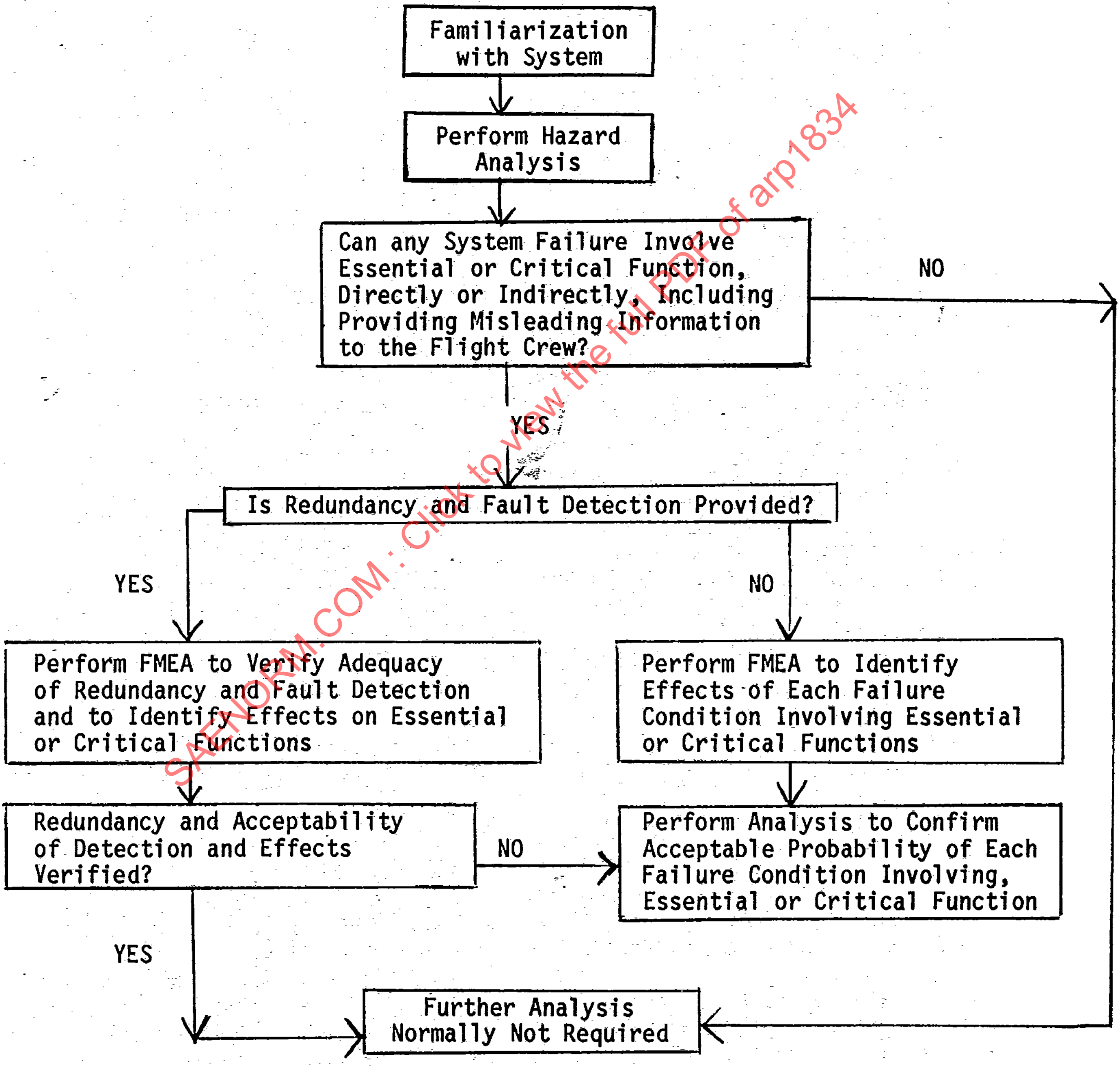
3.1 Identification of General Needs: Performance of F/FAs typically involves a cause-and-effect analysis procedure. As a result, at some point in the analysis, the modes of failure (the "cause") must be considered. This section addresses the topic of failure modes of digital functions and devices.

The level at which an F/FA is started affects what is perceived to be a fault. The top level effect will always be the same, regardless of the level at which analysis is started. If a functional F/FA is to be performed, the modes of failure and their relative frequency of occurrence are hypothesized for each function at this particular equipment level. For a hardware approach at the lowest level subject to analysis, the failure modes themselves must be evaluated in terms of higher level effects. The analyst must guard against dealing with generalities and must recognize the unique aspects of each design.

Within this framework and with emphasis on digitally oriented time-based systems, the determination of basic "failure modes" may often prove difficult, if not impossible. As newer digitally oriented systems evolve, trends toward increasing computational power make it more difficult to identify specific, elementary, simplistic failure modes. Failure modes in this context are synonymous with what are often referred to as failure indicators of digital devices - the output or device package pin level indications of failure caused by internal failure mechanisms related to the chip or interconnect.

Figure 2-4

An Example of an F/FA Decision Tree


Familiarization with System
Perform Hazard Analysis
Can any System Failure Involve Essential or Critical Function, Directly or Indirectly, Including Providing Misleading Information to the Flight Crew?
NO
YES
Is Redundancy and Fault Detection Provided?
YES
NO
Perform FMEA to Verify Adequacy of Redundancy and Fault Detection and to Identify Effects on Essential or Critical Functions
Perform FMEA to Identify Effects of Each Failure Condition Involving Essential or Critical Functions
Redundancy and Acceptability of Detection and Effects Verified?
NO
Perform Analysis to Confirm Acceptable Probability of Each Failure Condition Involving, Essential or Critical Function
YES
Further Analysis Normally Not Required

3.2 Failure Mechanisms: Failure mechanisms, internal to the microcircuit package, are typically explored via "physics of failure." General information is available in this area for most digital device technologies, and typically includes failure mechanisms related to defects in the device, such as:

Surface
Bulk
Oxide
Diffusion
Metalization
Bonds/Interconnect
Packaging

The resulting failure effects at output pins, which may be many gate levels away from the originating failure, are application-dependent, and very little practical failure information is available relative to the part's output levels.

3.3 Modes and Effects:

3.3.1 Device Failure Modes: A basic approach in detailed F/FAs has been to consider device failures which are assumed to manifest themselves in either a "stuck-at-1" or "stuck-at-0" failure condition at an output pin, typically with a 50%/50% distribution. For simple gates, inverters and other small scale integration (SSI) devices, this appears to be a reasonable approach in most cases.

However, many failure mechanisms may not be directly related to a single stuck-at condition, affecting multiple pins in various combinations. Also, intermittents, indeterminate outputs and conditional failure modes may need to be considered. It is not appropriate to directly extend this approach to medium scale and large scale integration (MSI & LSI) devices, especially in the areas of time-based processors, memories, and other digital bus-oriented devices. Because of the system architecture and interdependencies of the internal functions of these more complex devices, output "stuck-ats" no longer realistically represent the major output level failure modes. The major challenge is how to determine the failure indications or package level effects for these more complex devices.

A method for estimating failure modes is to model the digital devices under consideration either with constituent functional blocks for which a better definition of failure modes exists, or possibly with a simple-gate equivalent. The complexity of the devices under consideration along with the goals and criticality of the analysis will typically dictate appropriate approaches. The determination of digital device failure modes will normally involve sound engineering judgment based on general and limited support data.

3.3.1 (Continued):

Table 3-1 presents a partial list of failure modes for some general digital device types. This is by no means an exhaustive representation, but may be considered a starting point in generating specific functional failure modes for a particular device being analyzed.

Table 3-1

Typical Device Failure Modes

DEVICE	POTENTIAL FAILURE MODES
MEMORY (RAM/ROM)	Defective Address Decoder Defective Address Buffers Defective Data Buffers Defective Control Logic Defective Cell(s) Cell Pattern Sensitivity
PROCESSOR	Defective Decoders Defective Registers Defective Control Logic Defective I/O Defective ALU,
GATE	Output stuck-High/Low Output Intermittent/ Oscillates Output indeterminate

3.3.2 Soft Failures: Intermittent or recoverable failure modes are most predominant, or at least subjected to the most discussion, in reference to random access memory (RAM) and are often referred to as "soft failures". These are random, typically single bit errors that neither damage the part nor are associated with any identifiable physical defect.

Soft failures may result from causes including: transients, system noise, pattern sensitivity, temperature sensitivity, and alpha radiation - the last due to traces of radioactive materials in the package. These ionizing alpha particles can generate electron-hole pairs in the semiconductor material, which may cause random shifts between the logic "1" and "C" states in bit locations. These failure modes are transitory in nature and cause no device damage. The associated local effects can be recovered by using appropriate monitoring.

3.3.3 Latent Failures: A latent failure is one which, when it occurs, is not immediately obvious or detected. Some latent failures can, when coupled with subsequent operational mode or configuration changes, or with subsequent failures, result in unacceptable characteristics.

Depending on the severity of the potential failure consequence it may be necessary to define the probability of occurrence of some latent failures and possibly design the system to limit the time for which the failure remains undetected. Latent failure modes usually include data and/or time dependencies where a failure indication occurs only under a certain unique set of inputs or operating conditions/parameters. Possible examples include:

- The condition where the output of a multiple input device is wrong or a failure condition exists only when the inputs are exposed to a specific and unique set of inputs - a simple example would be a 4-input NAND gate which operates correctly except that the output also becomes a logic "0" when a specific input of "1101" occurs.

- A random access memory (RAM) device where a data bit may change logic states when a unique set of logic levels exists in other associated storage locations.

Some considerations of latent failure modes include the following:

- The occurrence of an activating stimulus (unique input conditions or bit patterns, etc.) may be so infrequent that detection during actual operation is not practically feasible.

- Assuming the first failure is latent, a second or specific sequence of subsequent fault/failure modes may lead to an undesired event; i.e., a multiple fault sequence may be needed to produce an effect.

- Activating stimuli or a required unique set of operating conditions may never occur, in which case a latent failure mode would be inconsequential.

3.3.3 (Continued):

While the occurrence of undetected failures cannot be prevented, the exposure interval for which these failures remain undetected can be limited. This may be accomplished through implementation of real time fault monitoring, power-up or pre-use self-test, scheduled maintenance action and, if indicated, bench testing. In this way the probability of the development of a hazardous situation can be reduced to an acceptable level. One approach is to re-examine the concept of performance monitoring and validation of specified performance. Key factors in this concept are:

- A component which operates in accordance with its design specifications is not failed, and
- A failure which is immediately detected is not latent.

This concept requires assurance that components used in critical functions are exercised in some way, such as automatic self-test, performance monitoring during normal operation, or bench tests within appropriate minimum frequency.

3.3.4 Failure Mode Data Sources: In addition to individual field failure data, a source of component failure mode and distribution data is the Reliability Analysis Center (RAC). RAC provides general data on digital device failure rates, failure modes, mechanisms and distributions through their Microcircuit Device Reliability (MDR) document series. However, care should be exercised in evaluating these data to avoid misleading or non-applicable conclusions. Since failure modes and distributions are influenced by application, the parts represented by the data should reflect similar applications, function, stress, etc. In addition, the failure modes and distributions of various technologies, and to some extent individual component types, vary with maturity. Therefore, the population from which the data were generated should be at the same level of maturity as that used in the design under analysis.

3.4 Failure Rate Allocation: In most cases, failure rates for digital devices can be estimated with the aid of industry-wide sources such as MIL-HDBK-217, RAC and GIDEP (Government Industry Data Exchange Program). In addition, company failure rates which have a statistically valid data base are often available. For F/FAs in which quantitative results are required, the allocation of a device's total failure rate to the various constituent failure modes is necessary. Frequently, a failure rate or probability budget is established for certain system level events or effects. Top-down analyses may show that certain device failure modes contribute to undesired top-level effects, and reasonable probabilities of occurrence for these failure modes must be determined. Realistic estimates of failure rate allocations for broad categories of typical failures are sometimes possible with available industry data. Allocation of failure rates to rare failure modes will usually be dependent upon application and will require engineering judgment.

3.5 Custom LSI: Custom LSI usually presents a different set of circumstances than standard LSI in that the device is designed by or for the manufacturer who intends to apply it in his product. Therefore, more detailed information concerning the internal configuration and expected failure modes should be available for use in F/FA's.

3.6 Software Considerations: In a system or end item that utilizes some type of processor control, such as a microprocessor, software usually directs or implements the sequencing of the type of control desired. System effects of hardware digital device failure modes typically become a function of the resultant response generated by the software. Therefore, software considerations, such as discussed in Section 2.3.6, are required in the process of performing F/FAs.

4. FAULT MONITORING METHODOLOGY: Fault monitoring can be either essentially continuous, repeated at intervals, originated by a specific event (such as power-up) or at a particular time (such as just prior to operational use of the monitored function).

Fault monitoring may be implemented either using dedicated hardware or included in the operational program, depending on the nature of the device or system function to be tested. The type of F/FA performed must consider the type of tests implemented and the criticality of the functions involved. A top-down hazard analysis starts with classifying the function, then the failure effects associated with that function. The exhaustiveness of coverage of each fault monitoring item should be related to the criticality of the monitored function.

The following sections provide an insight into types of fault monitoring and their effectiveness. No credit is given for fault monitoring provided exclusively for maintenance, unless it is also required to limit exposure time to support a numerical safety analysis. This section is primarily concerned with safety, rather than with Line Replaceable Unit (LRU) fault isolation.

4.1 Reasons for Fault Monitoring:

- To verify availability of function; this may involve all levels of criticality;
- To verify partition boundaries; for example, use of separate checksums for each subdivided area;
- To verify operation of monitoring elements;
- To minimize exposure time of latent failures; this may be controlled by the use of tests performed either continuously, repetitively, at power up, or just prior to use;
- When applicable, to satisfy fault/failure probability criteria related to system criticality and requirements of customer or regulatory authority.

4.2 System Architecture vs Fault Monitoring:

- The use of similar redundancy, as opposed to a "single thread" design, may reduce both fault monitoring requirements and extent of analysis for the detection of non-generic failures. Proper attention to system architecture early in the design may reduce or eliminate the need for using special analysis methods, covered in Section 5.2.

- The use of dissimilar redundancy schemes may provide the means to protect against generic failures and design errors. Generic hardware or software design errors or manufacturer's obscure design changes may be detectable by comparison monitoring of the dissimilar redundant elements.

- A top-down fault analysis can aid in the partitioning of hardware and software functions by criticality. Fault monitoring for these functions can be partitioned accordingly, to minimize the risk that future design changes in one partitioned area may negate analysis results in an unrelated area. The specific details of the partitioning must be analyzed to determine the independence between partitioned functions.

4.3 Types of Fault Monitoring: Digital systems may make use of an array of system monitors in both hardware and software. These system monitors are designed to detect failures in the digital system's hardware for subsequent fault management. The fault management techniques are dependent upon the system architecture and may be quite different for single thread, dual, and other forms of redundancy such as dual-dual and triplex.

The following three categories of monitoring are generally implemented:

- Processor failure detection,

- Data transmission error detection, and

- Data validity.

The examples given are illustrative of those being used in digital systems and are not meant to indicate that these are the only methods. The principal concern is selecting methods which meet the objectives. Examples of the three categories are as follows:

4.3.1 Processor Failure Detection: The common methods for detecting failures of processing units employ self-test and comparison monitoring. In self-test, functions of the processor are exercised by a set of computations designed to test that function. The results of each computational set are compared with pre-stored values. Any differences signify that a fault was detected. Typical tests include:

- CPU - instruction set and additional tests dependent upon processor architecture. These may include; floating point, data input/output (I/O) registers, interrupts (possibly tested at power up), status, overflow, sign extent, jump select, shift, look ahead/carry, multiplexer, instruction decode, mask ROM.

4.3.1 (Continued):

- RAM - write/read, parity, error detecting and correcting capability, addressing.
- ROM - Checksum, Cyclic Redundancy Checks (CRC), Sectional CRCs, decoding, parity.
- Task completion - verify that all required tasks have been executed.
- Foreground/background checks - these consist of critical monitor checks, program computation reasonableness, and dual computations with difference monitoring.
- Heartbeat Monitor (sometimes known as Watchdog Timer) - checks the operational program execution rate to verify proper timing.

4.3.2 Data Transmission Error Detection: Data transmission validity checks may include:

- Parity.
- Word length count.
- Message length count.
- Address Validity.
- Modulation Waveform Test.
- Intermessage Gap Time Test.
- Minimum No-Response Time-Out.
- Contiguous Message Transmission.
- Vertical Redundancy Check (VRC).
- Longitudinal Redundancy Check (LRC).
- Cyclic Redundancy Check (CRC).

4.3.3 Data Validity: Even though there may be no errors introduced in the transmission of data between subsystems connected by digital data buses, this does not mean that the data transmitted are valid. The systems engineer needs to consider other means of ascertaining that the data transmitted and received are valid. These checks should be designed to test the sensor inputs, conversion of sensor data to digital format, processing of these data, and conversion of the digital output data to the form required by the actuators or sensors using the result of the operation. Checks may include:

- Input range limit.

4.3.3 (Continued):

- Monotonicity -- verifies that an output signal changes in one direction only in response to an input signal changing in one direction.
- Input Rate of Change.
- Interface Connection.
- Redundant Parameter Comparison -- compares the values of "equivalent" redundant parameters to determine whether they are within allowable tolerance of each other.
- Redundant Logic Comparison -- compares the states of redundant logic signals to determine whether they are the same.
- Wraparound -- compares system output with identical data, re-processed through the same system, against specified tolerances.
- Loop dynamics -- control loop error (command value minus measured value) is compared against programmed error limits.
- End of Conversion (EOC) bit -- after signaling start of data conversion to a digital data converter, an EOC bit should be set within an allowed conversion time.
- Power Supply.

4.4 Fault Monitoring Effectiveness: Pertinent considerations in the determination of how well the fault monitoring performs its intended function include:

- Adequacy of the designed fault monitoring in detecting the faults identified in the hazard analysis;
- Latent fault exposure time (this is related to the frequency of running a certain test: periodic or repetitive, power-up, or pre-use);
- Correctness and timeliness of the fault reaction to adequately warn and/or reconfigure the system for safety;
- Adequacy of the detection threshold (tolerance) to prevent undesired system characteristics;
- Possibility of nuisance warnings and nuisance disconnects; and
- Possibility of monitor inoperative (latent) due to fault/failure.

4.5 Method of Fault Monitoring Analysis: The analysis should be directed specifically to the functional failure conditions identified by the hazard analysis.

Using worst case analysis techniques, adequate redundancy, and/or dissimilar processing techniques can reduce the need for exhaustive analyses, such as processor FMEAs. For example, if a failure effects analysis can show adequate safety or that the design meets contractual requirements by assuming the total processor failure rate, and assuming that all fault monitoring in the processor has also failed, no further "exhaustive" analytical work is required.

If system architecture is such that achieving the required level of safety relies heavily on in-line monitoring to detect all hardware failure modes within the system or processor, then more extensive analytical methods, such as described in Section 5.2, may be required.

5. ANALYSIS METHODS: In this section, the basic or convention methods of analysis are discussed followed by an introduction to special methods (simulation and emulation) particularly useful in dealing with processor-based digital systems.

5.1 Basic Methods and Elements:

5.1.1 General: The basis of the F/FA preparation is a subdivision of the system into smaller and smaller functional groups and elements, followed by study and understanding of how each contributes to related functions and to the separation between functional groups.

The failure modes of the lowest level elements may be simple physical faults such as opens, shorts and value changes (as in the case of analog parts) or so complex that faults may not be easily expressed in terms of specific internal defects (as with digital I.C.s). In the latter case, it may be practical only to consider the output function of the part and its corresponding malfunctions. The analysis of individual gate level failures may require special methods (simulation and emulation).

Digital circuits based upon processors have the complicating factors of memory failure because of ROM/RAM faults or failure to execute instructions because of CPU faults. Unmonitored failure of these elements can cause unpredictable results in that the computer program may loop, skip, or move into incorrect branches. The inclusion of fault monitors can make an F/FA of these portions of the circuit manageable, since the desired responses to the fault monitors are part of the design.

5.1.2 Sequence: The following sequence for the F/FA process is recommended:

1. Gather data such as circuit diagrams, specifications, theories of operation, etc., and study these documents.

2. Identify critical, essential and non-essential functions.

3. Identify portions of software requirements that implement critical or essential functions and others of concern.

4. Identify all hardware and software implemented fault monitoring, fault detection and designed responses. This step is especially important to the analysis.

5. For a "top-down" analysis, work down into the circuits to determine, as necessary, the hardware element failures that may affect the functions identified in Step 2, above.

6. If it is unclear whether an element's failure is relevant in a top-down analysis, examine it from the bottom-up. This "bottom-up" analysis is also used if all failure modes are to be identified. In these cases, list each part or functional group (block) of parts and analyze how it may fail. For complex chips, it may be practical to state these failure modes only in terms of functional rather than physical faults.

7. Subdivide each circuit into the lowest functional element or group of parts that will require analysis. Critical functions will usually require a greater subdivision than essential items, while subdivision of non-essential items may be relatively broad.

8. Determine fault monitoring methods used to detect each fault.

9. Assess the intended fault monitoring response of the system to detected failures. If the monitoring provided does not detect a fault, examine the software requirements to locate the portion calling for a response from the element and trace the effect of the malfunctioning part to its ultimate result. The objective is to ensure that the intended response is timely and appropriate in concept and to validate that the system does respond as intended.

10. If required, assign a failure rate and probability of failure to non-detected faults.

11. Document the results of the analysis.

These F/FA process steps are discussed in detail in Para. 5.1.3.

5.1.3 F/FA Process Steps:

5.1.3.1 Understanding System Requirements & Design: As a first step in performing an F/FA, gather all information describing the system, including specifications and theories of operation as needed to gain a thorough understanding of the functional characteristics and requirements of the system for both hardware and software. Depending on the stage of development, information used to describe the system may range from functional block diagrams to detailed schematic circuits. Other documents, such as timing diagrams, system control laws and definitions, descriptions, specifications, failure definitions, operational and performance profiles, and reports bearing on reliability and maintainability should be included. Information pertaining to the software organization can be obtained from software documentation.

If the lowest level of hardware subject to analysis needs to be addressed in the analysis, it will be necessary to study the manufacturer's data sheets, specifications, or application notes in order to understand the functional characteristics of the parts and how they relate to the equipment operation. This is particularly applicable to medium and large scale integrated (LSI) circuits, which alone may take on the complexity of a sub-system and may be analyzed as such.

For a processor-based system, fault monitoring implementation through software as well as hardware is usually designed into the system. A study and thorough understanding of the fault monitoring operations and the functions or subfunctions being monitored, including response to detection of failures as well as limitations, is necessary for a complete F/FA.

5.1.3.2 Identification of Function Criticality: Determine which functions are critical, essential and non-essential. To do this, define the potentially relevant failure effects at the uppermost functional level. This will be based on a study of the technical specifications, requirements, and development plans for the system and information supplied by the customer or user.

5.1.3.3 Review of Software: Review software requirements to identify portions of the program which implement critical or essential functions and how software is designed to implement these functions. It is very important to understand how fault management performed by the operational program instructions is designed. It is also important that the analyst develop a good understanding of the detailed specification requirements and the software instructions which, when executed, implement the fault monitoring design. Without this understanding, it is not possible to properly evaluate the failure effects and their detectability.

5.1.3.4 Fault Monitoring Effectiveness: Determine the extent and means of implementing fault monitoring (hardware, software). This step is one of the most important in the F/FA process. With insufficient monitoring, faults may be led by software into completely unpredictable paths, branches, and loops, depending upon when in the program the failure

5.1.3.4 (Continued):

occurred. On the other hand, the "effect" (such as in an FMEA) of hardware failures detected by fault monitoring is normally the designed response to the detected fault, greatly simplifying the analysis. Many fault monitoring schemes have become standardized. A review of the literature may be useful since discussions of new fault monitoring techniques are published from time-to-time.

The fault monitoring review should be conducted in the same sequence as the design is intended to operate. This is sometimes described as the "center out" sequence. The technique is to establish the health of the central processor first by self test routines. Next, the associated functions are tested utilizing the "healthy" processor. An example of the testing at this stage would be a complete memory check.

The peripheral functions, such as A to D and D to A conversion and other input/output processing, are next checked using routines imbedded in the validated memory.

Finally, the whole computer may be used to validate or test inputs from, and outputs to, the rest of the system.

To accomplish the fault monitoring review, the analyst should identify both the software requirements and the hardware elements which provide it. A tabular listing may be needed for summarizing the types of faults detected and responses of each form of fault monitoring. One should be careful to note any conditional monitoring such as "fault is detected, provided . . ." If the "provided" relates to the correct operation of a part of software, which in turn may not occur due to a hardware fault or failure, the response may be unpredictable.

5.1.3.5 Top Down Analysis: When all hazards can be defined in advance, the most cost-effective analytical approach is usually the top-down F/FA method.

After the failures affecting important functions at the uppermost level have been identified, the work progresses downward into the system to identify which blocks may lead to these effects. Different ways in which the functions can fail are postulated, considering how each relates to the critical or essential functional failure effect of the next higher level block. This process is repeated at progressively lower functional levels until the lowest block level to be analyzed is reached. The depth of this progression will depend upon the need to identify specific areas for design improvement, calculation of failure probability, etc. Certainly, critical functions will require analysis in depth, while analysis of non-essential functions might stop at a much higher level.

The advantage of the top-down approach is the concentration of effort on failure effects of critical or essential functions. The analyst,

5.1.3.5 (Continued):

however, is cautioned that digital system hardware may perform multiple functions, as directed by resident software, and assessment of hardware criticality must address these multiple functions before a "non-essential" determination can be made.

5.1.3.6 Bottom-Up Analysis: The bottom-up analysis is used when necessary to expand on or augment the results of the top-down analysis. It is initiated at the piece part or component level or at a higher intermediate equipment level. This analysis can identify failure modes that may not be otherwise identified, some of which may affect critical or essential functions.

In the bottom-up analysis the analyst postulates each of the ways in which each designated lowest-level item may fail. Each such failure is then related to, or assumed to cause, higher-level faults during each mode in which the equipment operates. Failures of each input and output of every such lowest-level item must also be postulated to account for failures of solder connections, printed-wiring runs, etc. In addition to failure postulation, some of the special methods described in 5.2 may be appropriate.

5.1.3.6.1 The Infeasibility of Part-Level FMEA of Digital Equipment: If initiated at the electronic-part level, a thorough, accurate, and dependable bottom-up analysis of most modern digital equipments is not feasible, because an exhaustively thorough failure-modes and effects analysis is a practical impossibility for essentially all contemporary types of microcircuits.

For digital microcircuits, unlike discrete electronic parts, there are no relevant data bases or standardized listings of failure modes available for the use of the analyst. Postulating all failure modes based on an analysis of the chip circuit and chip construction is impractical. The great complexity of all but the most primitive types of microcircuit results in enormous numbers of different failure modes, only a fraction of which it is feasible to identify and characterize by analysis. Moreover, such chip-design information for many microcircuits is proprietary data that the microcircuit-designer-manufacturer (MDM) will not disclose. Even if such data were available and a complete, accurate analysis were somehow feasible, confidence in the long-term validity of the results would be low. MDMs frequently make minor, unannounced changes to the physical or electrical design of their microcircuits and to their processes and materials, and many such changes introduce new chip-failure modes or modify existing ones. Consequently the analyst's findings and conclusions would become increasingly erroneous with the passage of time, as failed microcircuits in field equipment are routinely replaced with newer, modified microcircuits.

5.1.3.6.1 (Continued):

Several less-than-exhaustive microcircuit failure-mode analysis and simulation methods, whereby certain failure modes are identified and analyzed while others remain unidentified, are discussed and exemplified herein. Typically such methods rest on simplifying assumptions that relinquish analytical exhaustiveness. Such methods may be useful for F/FA's in certain non-critical applications.

Users of ARP 1834 whose particular application of F/FA requires exhaustively complete analytical results, as is so for most safety applications, will likely find that the use of design redundancy (see 1.3) is the most feasible and cost-effective means of meeting safety objectives.

5.1.3.7 Separation of Functional Elements: The system should be divided into functional blocks down to the lowest level appropriate to the analysis. For each block, internal and interface functions should be studied relative to system operation. Often, given a clear description of the block's function, many of the failure modes will become apparent.

The number of levels of subdivision is determined by the complexity of the system and the objectives of the analysis; e.g., critical equipment would normally be subdivided into many levels and sublevels while non-essential equipment may have much broader subdivisions. The subdivision starts from the uppermost level and progresses downward, according to system function and hardware organization. For a system made up of several line replaceable units (LRUs), the initial division is made at the "black-box" level, then at the card or module level, and finally at the circuit level. Below this level, special methods may be necessary. If the design utilizes partitioning, as defined in DO-178, to minimize the effects of failures, it may be advantageous to subdivide the system for analytical purposes to correspond as closely as possible to this partitioning. This will facilitate an analysis of the effectiveness of the architectural arrangement.

If redundancy has been designed into the system, the system should be analyzed to account for the cross-channel links, if any. Attention should be given to special features of the system under analysis (Fail-Operative, Fail-Passive, Fail-Safe, etc.). Fault monitoring provisions within each channel and cross-monitors should be identified and described. Their capabilities and their limitations should be determined.

5.1.3.8 Fault Detection Coverage: The types of fault monitoring and methods of fault detection must be identified. Depending on the time the failure occurs, different forms of fault monitoring may or may not accomplish detection. Therefore, the analyst should examine such monitoring implementation in each phase of the program development. Time-dependent failures are likely to be dependent on correct software operation for their detection.

5.1.3.8 (Continued):

If fault monitoring is implemented in hardware (or with software resident in hardware), its own failure modes must be examined, especially those which cause it to be inactive. Latent failures require additional analysis of the monitored circuit to determine failure effects in the absence of monitoring, and perhaps necessitate an increase in frequency of fault monitoring test subroutines, or changes to the software to enable their detection.

5.1.3.9 Fault Monitoring Response: When an individual failure mode is identified, the next step in a top-down F/FA is a determination of whether the effect is among those listed as important. In a bottom-up analysis, the next step is a statement of effect at progressively higher circuit or assembly levels. In their most basic form, the effects are merely the failure to perform or the improper performance of a function identified in the functional description of the part or block. To better follow the logic of the analyst, these effects should be determined and documented at several levels, such as the lowest level subject to analysis, the next higher function level and the uppermost system level.

Important questions are:

- What functions are affected by the failure?
- Is the function totally disabled?
- Does it continue to perform, but in a degraded or different manner? If so, how?

The next higher functional effect determination serves to connect the local effect and the effect on the next level to the uppermost level of the system.

At the system level, effects considered should include:

- Is the effect passive or active? Is it safe?
- Is the total system affected or only part of the system? Are there backup provisions? Redundancy?
- Does it continue to perform, but in a degraded or intermittent manner?
- How is performance affected in terms of its ultimate use?
- How is the fault apparent to the user? How long after the failure will it become apparent? How should the user respond?
- Does the fault affect critical, essential, or only non-essential functions?

5.1.3.9 (Continued):

When determining the effect at any level, it may be necessary to examine the related software requirement to see how it will respond to the postulated failures. For example, control logic implemented through software may be designed to change the operational mode (or to activate a redundant channel) upon detecting faults, in order to alleviate adverse effects on system operation.

Special attention must also be paid to hardware faults which may result in both functional and monitor failure.

For each fault, the important questions are:

- Is the fault detected in a sufficiently timely manner?
- What is the response to the fault monitoring?

If the answer to the first question is "yes", the analyst's job is greatly simplified. If fault monitoring is neither repetitive nor relatively frequent, the analyst should consider interim effects and latent failure modes until fault monitoring is initiated.

If the answer to the first question is "no", the analyst must re-examine the software requirements in an effort to track the effect of the failure to its final state. In many cases, this will require following a chain of events in search of a single effect or a few potential effects, relating to the ultimate effects of interest. When it is not practical to accomplish this with reasonable certainty because of many indeterminate paths, worst case assumptions should be made.

For certain systems whose design objective is the ability to isolate a high percentage of the failures, it may also be necessary to determine the effect of maintenance fault detection at the node/monitor points. Pertinent questions are:

- Does the fault force any monitor points to a recognized fault state?
- What identifiable abnormal conditions result from the presence of the fault?
- How is this recognized by the user? When?

The analyst may then be able to determine whether it is necessary to add or change maintenance requirements, or revise hardware or software to meet requirements.

5.1.3.10 Quantify Probability of Failure: If necessary, the significance of undetected failures affecting potentially critical or essential functions may be put into perspective by assignment of predicted failure rates and exposure times to the pertinent parts. An unacceptable combination of potential or real failure effect and

5.1.3.10 (Continued):

excessive failure rate or exposure time suggests that design changes are needed for additional or more frequent fault monitoring, and/or the implementation of redundancy techniques.

5.1.3.11 Data Content of F/FA Report: Subcontractor standardization of data presentation may be required by a prime contractor who must integrate all subsystem analyses into an overall system F/FA. In this event, a standardized format may be specified. If a specific data format is not prescribed, one should be chosen which recognizes the design complexity and unique applications of the equipment being analyzed. The selected format should consider the type of analysis, the intended use of the F/FA results and pertinent questions being asked. In either case, elements identified in this document are the basic items which should be covered in the format selected. In general, the reported data should consider:

- Objective of the analysis.
- A summary description of system and/or equipment performance, including block diagrams and functional diagrams, if necessary.
- A description of significant redundancy, if any.
- A list of identified failure modes or effects of concern and rationale for their inclusion.
- A list of or reference to components or piece-parts analyzed and the rationale for their selection.
- A generalized description of fault monitoring, how it operates, and what faults it is intended to detect.
- At each level of the analysis, a short functional description of the assembly or circuit being analyzed, its purpose in the system, and what specific fault monitors apply to the circuit and how they operate.
- A description of effect at progressively higher (bottom-up) or lower (top-down) levels for each functional or hardware fault. Include a description of either the fault monitoring response (which may be the ultimate effect) or the effect in the event of passive monitoring failure, as appropriate.
- A predicted failure rate for undetected faults.
- A summary collation of similar faults.
- Calculation of the failure rate or probability for all unacceptable failures, considering exposure time and redundancy or other compensating factors, if any.

5.2 Special Methods: The F/FA of a digital system is complicated by the fact that the system response to a failure may be time, mode or data dependent. For a complex system, the analytical prediction of a system response to a specified failure may be nearly impossible via the basic methods of paragraph 5.1. One way of circumventing this problem has been described earlier; that is, the extensive use of redundancy and/or fault monitoring with tests so that the effect of the failure can be detected during known, controlled conditions and the system response to the detected failure can be designed-in (e.g. shut-down, reconfiguration, etc.). However, when this is not practical or sufficient, special test techniques may be needed to simulate and evaluate the effects of specified failures on the dynamic system.

These techniques generally involve fault insertion:

a) into the target or representative hardware,

b) into a computer emulation of the hardware design, or

c) into a computer simulation of the hardware functions.

One or more of these three methods can be used selectively to augment the basic F/FA, validate the analytical results, and evaluate the effectiveness of the built-in tests and monitors. The specific usage will depend on the system being analyzed.

5.2.1 Fault Insertion Using Hardware: Fault insertion using target or representative hardware can provide responses to selected failures at a level below the LRU.

The two principal methods are:

- Generate physical faults by opening leads or shortening leads together.
- Insert logic or computational variations and evaluate the resulting effects. Faults can be inserted manually or automatically through a test program.

The advantages of fault insertion using hardware include:

- The hardware can be programmed to execute all or part of the operational software and will run in real time.
- The fault responses are realistic for the specific faults being inserted.

The limitations include:

- Only a restricted number of faults at the pin level can be tested. Exhaustively testing all faults is ordinarily a practical impossibility.
- For all but the simplest microcircuits, faults at the pin level may not sufficiently or realistically represent all internal microcircuit faults.

5.2.1 (Continued):

- The tests cannot be conducted until the target or representative hardware is available.
- The circuitry involved with manual insertion of faults may itself produce spurious or faulty circuit performance.

5.2.2 Fault Insertion Using Emulation: An emulator, in the context used here, is a computer emulation of another computer hardware. The host computer is programmed to represent the hardware functions and interfaces of the various devices so that the whole responds in a realistic manner to the instruction set of the emulated system. The emulation can be at the level appropriate to the devices involved, i.e., gate level, register level, etc. Inherent in the device emulation is the ability to insert faults of various kinds. The emulator can also be programmed to execute all or part of the application program with and without faults present. The advantages of using an emulator include:

- Many more fault modes are available for investigation.
- Testing can be conducted prior to the actual hardware being available.

The limitations include:

- Realistic emulation requires detailed information about the characteristics and, perhaps, internal configuration of the devices to be emulated.
- A considerable effort in design and checkout is required to produce a satisfactory emulation. Complex circuits may be very difficult to emulate.
- The emulation runs more slowly than the actual hardware (as much as 25,000 times more slowly). Data gathering can, therefore, take a considerable amount of time.

5.2.3 Fault Insertion into a Computer Simulation of the Hardware Functions: Simulation can be used to model the functional aspects of the hardware when the actual hardware is not available or when it is desired to simulate only portions of the hardware functions. Simulation, as opposed to emulation, can be run at many times the actual speed of the hardware. Simulation can also provide representation of outside variables and their interaction with physical and simulated hardware.

The advantages of using simulation techniques include:

- Any hardware failure mode that can be simulated can be evaluated.
- Testing may be conducted independent of actual hardware or in conjunction with portions of the hardware.

5.2.3 (Continued):

- Effects of failures can be evaluated at a functional level.

The limitations of using these techniques include:

- Realistic simulation requires detailed information about the hardware, the environment in which it is intended to operate, and the critical application of the functions simulated.

- Verification and validation of the adequacy and thoroughness of the simulation can be difficult.

APPENDIX A

EXAMPLE - F/FA Basic Bottom-Up Approach:

NOTE: Descriptive comments appear in the discussion of these examples. They are presented to help the reader understand the operation of the equipment being analyzed.

* * * * * *

This section presents steps that were taken in performing a fault/failure analysis on digital equipment at the hardware level. Procedures other than the one shown could be used; however, they would normally include these same steps, or equivalent, as listed below.

The bottom up approach was used in this example for the development of an F/FA. This consisted of propagating failures of the lowest level functional blocks up to the system level to determine the effect on system operation. Where applicable, because of unique function, parts were analyzed as separate functional blocks. Functional block failures were divided into their modes of failure. Each mode of failure was examined to determine if its failure was detectable by fault monitoring or by other means. Only single point failures were considered in this example. The steps were as follows:

- Understand the system requirements.
- Understand the system implementation.
- Review the system functions (subdivide system functional blocks into smaller sub-function blocks, as needed).
- Analyze software.
- Analyze the techniques & implementation of fault monitoring.
- Postulate failure modes and determine effects and estimate failure rates.
- Evaluate thoroughness of functional fault monitoring.
- Prepare data analysis and summary.

Understand the System:

The first step toward doing this F/FA was developing an in-depth understanding of the system requirements and installation.

Figure A-1 is the total system block diagram of a combination digital/analog system, of which the digital portion was used for this example. Each input and output interface signal was identified in terms of signal type (digital or analog) and how it was used in the system. Failure/fault response requirements were also studied, including the interface and display/control methods used for the warning.

Analysis of System Functions:

The system was divided into functional blocks. A number was assigned to each key functional block for identification purpose.

Figure A-2 illustrates the digital system, broken down into 14 functional blocks.

Subdivision into Smaller Sub-Functional Blocks:

This step required dividing each functional block shown in figure A-2 into meaningful sub-functional blocks. A sub-functional block represented an isolated logical or circuit function. This was a group of components, ICs or gates.

All interface signals between sub-functional blocks and system functional blocks were identified. This enabled the analyst to tie the entire system together and served as a useful tool for fault isolations and to diagnose the system effect caused by the failure of any individual functional block.

Figure A-3 illustrates how the microcomputer was broken down into sub-functional blocks. Interface signal paths between sub-functional blocks are shown.

Software Consideration:

Software was considered in terms of monitoring and functional requirements via the digital system components. As an example, in Figure A-3, the system was broken down into the following sub-system components:

- Processor
- ROM memory, to store the program
- RAM memory, to store the computations
- Timing and control functions
- Input/Output functions.

EXAMPLE

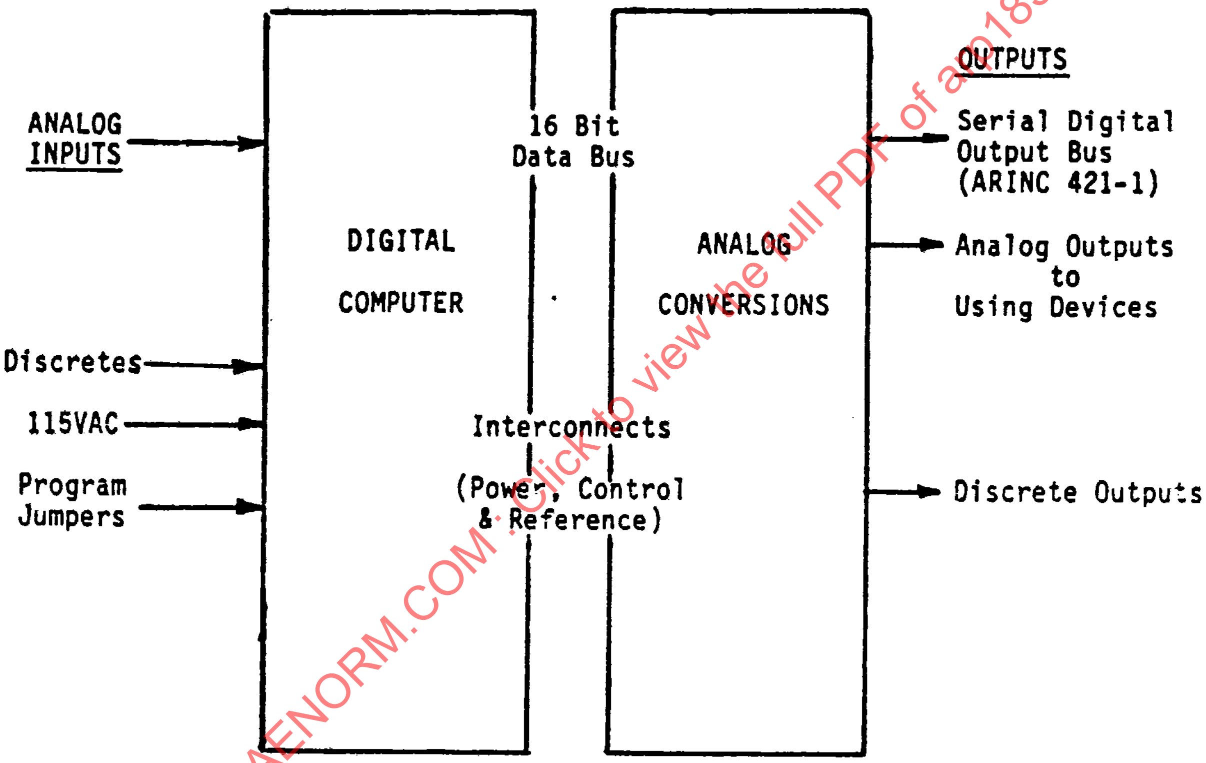


FIGURE A-1 TOTAL SYSTEM BLOCK DIAGRAM

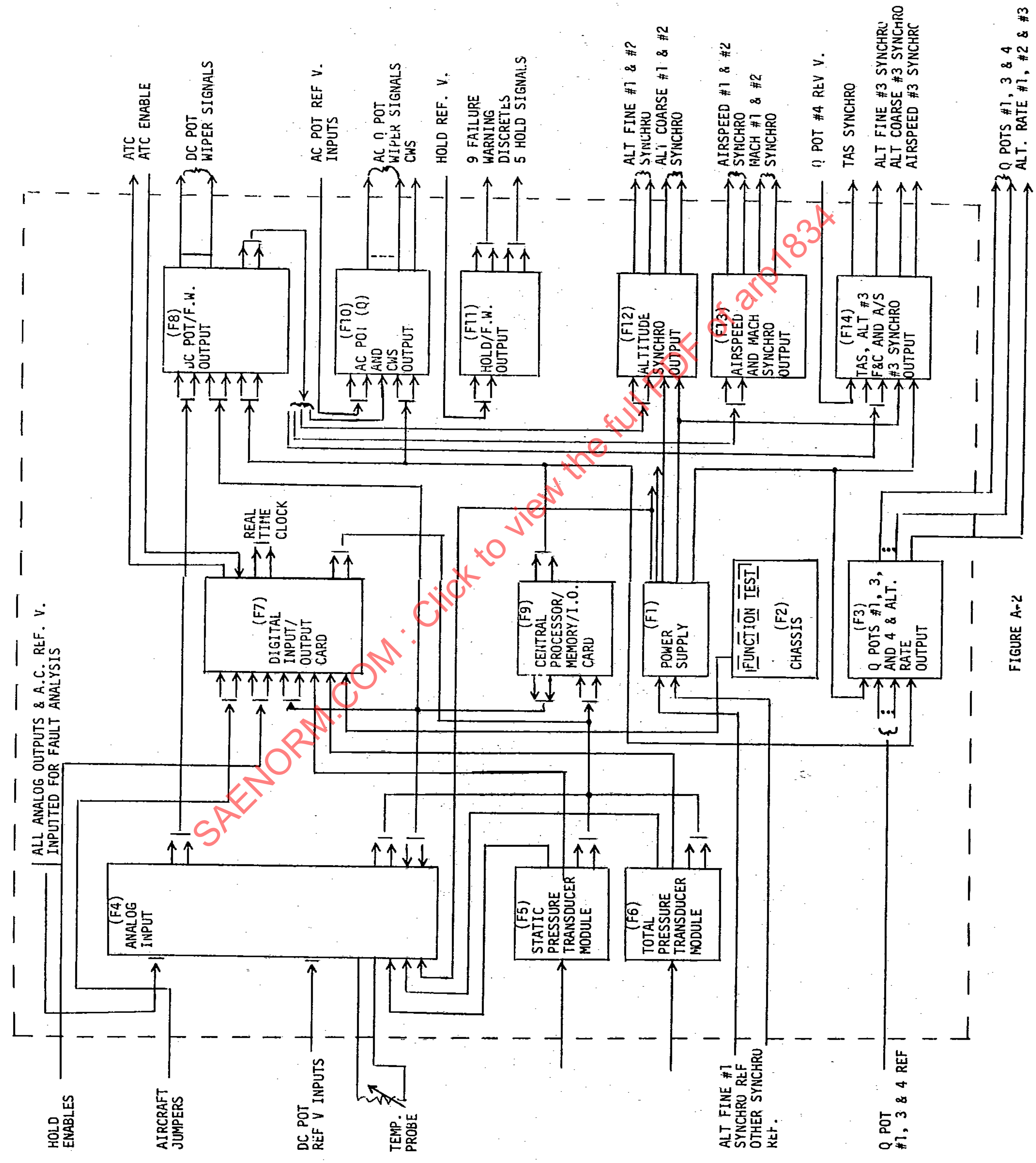


FIGURE A-2

DIGITAL COMPUTER FUNCTIONAL BLOCK DIAGRAM

The failure of a ROM was considered; generally this would affect the circuit's ability to perform its normal functions. When this occurred during the computation of a primary function, the failure would cause an incorrect instruction to the processor. This incorrect instruction could cause data to become lost, or the computer to stop or make a wrong computation. Consequently, the primary function would probably be lost or the end function would become unacceptable, causing the prime equipment to either fail or give incorrect data.

An evaluation of software interactions, for each failure mode considered, was necessary during the analysis. Detection of the resultant effects required evaluation of fault monitoring effectiveness. The following guidelines were used to establish that effectiveness:

- A detailed check of the Product Specification was performed to establish the full fault monitoring requirements. Listed were all functions, I/O, interactive signals, etc., that were to be exercised for system confidence.

- A detailed check was made of the Software Requirements Document.

- A step by step interrogation of the fault monitoring software flow charts and coding was performed. At this point, it was necessary to construct block diagrams of all functions requested for better understanding.

- A comparison was made of the software data against the F/FA areas of concern.

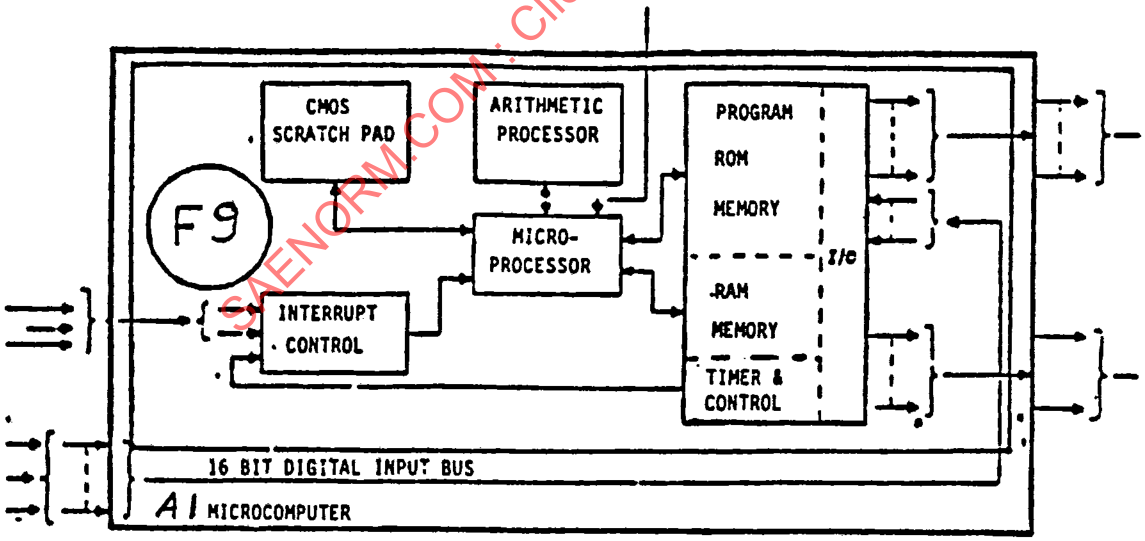


FIGURE A-3
Sub-functional Block Diagram

It was then possible to proceed in establishing the following:

- That the specified fault monitoring requirements were being met in full.
- That areas were identified which required further investigation and possible modification.

Review Implementation of Fault Monitoring:

Although an aircraft may employ more than one computer for redundancy, the circuitry contained in the unit analyzed was a straight-line, non-redundant design. As a consequence, computer malfunctions were expected to result in the loss of, or errors in, output data. For this reason, extensive self-test was built into the computer software to detect and annunicate equipment malfunctions.

The computer had, designed into the software, various self-test routines executed for ground maintenance fault isolation, system power-up, and in-flight performance monitoring. An example of the specific tests and annunciation methods is presented for the combined digital/analog device in Tables A-1 and A-2.

Power-Up Test:

The power-up test was run when power was applied during computer initialization or, during flight, when power was interrupted for an extended period of time (requiring a "cold start" power-up). The Power-up test sequence consisted of several tests designed to test certain functions which could not be tested on a continuous basis during in-flight monitoring.

TABLE A-1

Power-Up and Ground Test Definitions

TYPE OF TEST	TEST CODE	TEST DESCRIPTION	TEST PASSING CRITERIA	FAULT ANNUNCIATION
POWER-UP BIT	P1	Master CPU	Result = 0	Identical to the respective continuous monitoring annunciation method.
	P2	Master APU	Result = 4.00643579E07	
	P3	Slave CPU	Result = 0	
	P4	Slave APU	Result = 4.00643579E07	
	P5	Watchdo g Timer	Times Out Within 2 Sec.	
	P6	ROM	Sum = 0	
	P7	RAM	Write = Read	
	P8	Interrupt	All Interrupts Operate	
GROUND TEST	G1	<u>Function Test</u> - Fixed Input Values resulting in known Hu, H-dot, Mach, SAT, TAS, & TAT outputs	Aircraft Indicators must ag ree with the expected known values (+ tolerance)	The difference between the aircraft indicator readings and the test input values will exceed allowable tolerances. Since continuous monitoring is run as a background durin g g round test, annunciation methods identical to those of continuous monitoring BIT will also be transmitted. Actual (or false) failure in any of these tests will extinguish the "<u>Valid When Lit</u>" <u>lamp</u> when the push-to-test switch is activated.
	G2	<u>Slow Test</u> - Outputs known H-dot value.		
	G3	<u>Hold Test</u> - Adds known 's to Hu, Mach, & CAS effecting the Hu, H-Dot, Mach, CAS, SAT, TAT, TAS, H, Mach, and CAS outputs.		
	G4	<u>Failure Warning Test</u> - Sets nine discretes & master relay, disables ATC, stops BCD & sets code on binary serial data.	Conditions explained in test description must be met.	Conditions explained in test description are <u>not</u> met.

If the Power-up testing detected a failure, it was annunciated by the method corresponding to the detecting test defined in Table A-1. If the Power-up tests continued to report a fault, the watchdog timer would time out and cause all failure discretes to drop within seconds of the initial failure. The failure discretes would remain in the failed state and the main program sequence would not be initiated until the Power-up test sequence has been successful. Examples of these tests are listed in Table A-1.

Continuous Monitoring:

The continuous or in-flight monitoring was a background routine that is run frequently during normal computer operation to ensure that the equipment is operating properly. The continuous monitoring for this example consisted of a number of tests which are presented in Table A-2 with their associated annunciation methods and "passing" criteria.

When the continuous monitoring detected a failure, it was annunciated by the method corresponding to the detecting test defined in Table A-2. Failures affecting only the analog outputs would only set the analog discrete failure warnings. Those failures resulting in invalid serial data would cause a fault code to be transmitted on the affected binary word and would terminate the Binary Coded Decimal (BCD) data transmission. For all tests, the proper failure warnings would be set only upon detection of two like consecutive failures.

Ground Maintenance Tests:

The ground maintenance tests were used on-board the aircraft to verify both the computer performance and the integrity of the aircraft indicators. In the shop, these tests plus a fifth "STIMULATE" test, were used for computer fault isolation. The four tests used on board are presented in Table A-1. The selected test was initiated when the front panel test button was depressed, and the results of the test were read on the appropriate aircraft indicator. Upon initiation of each ground test, continuous monitoring was also activated as a background test. Detection of any failure by the continuous monitoring or the release of the push-to-test button would extinguish the "valid when lit" lamp.

Analysis of FAULT Status:

During flight, the output data and failure warnings would assume one of the various states given below:

DATA VERSUS FAULT STATUS CONDITION

Condition No.	Output Data	Failure Warn
1	Valid	Off
2	Valid	On
3	Invalid	On
4	Invalid	Off

TEST CODE	TEST DESCRIPTION	TEST PASSING CRITERIA	FAILURE ANNUNCIATION																			
			ANALOG DISCRETES 2											SERIAL WORDS 3								
			Hu #1	Hu #2	Hu #3	Hu #4	Hc	CAS	TAS/ SAT	MACH 1	MACH 2	HR	ATC 1	Hu	N	CAS	Ps	Pt	SAT 4	TAT 4	TAS 4	MACH
C1	Power Supply Shutdown	No Overcurrent/Voltage	1	1	1	1	1	1	1	1	1	1	D	1	1	1	1	1	1	1	1	1
C2	A/D Converter Null	|A/N| ≤ 0.05V	1	1	1	1	1	1	1	1	1	1	D	1	1	1	1	1	1	1	1	1
C6	Ps Time Pulse	0.01 ≤ PSTP ≤ 1.3 msec	1	1	1	1	1	1	1	1	1	1	D	1	1	1	1	1	1	1	1	1
C8	Ps ROM	Sum = 0	1	1	1	1	1	1	1	1	1	1	D	1	1	1	1	1	1	1	1	1
C9	Ps Least Sig. Nibble	Four LSB's ≠ 0	1	1	1	1	1	1	1	1	1	1	D	1	1	1	1	1	1	1	1	1
C11	Pt Time Pulse	0.01 ≤ PTTP ≤ 1.3 msec	0	0	0	0	0	1	1	1	1	1	E	0	0	1	0	1	1	1	1	1
C13	Pt ROM	Sum = 0	0	0	0	0	0	1	1	1	1	1	E	0	0	1	0	1	1	1	1	1
C14	Pt Least Sig. Nibble	Four LSB's ≠ 0	0	0	0	0	0	1	1	1	1	1	E	0	0	1	0	1	1	1	1	1
C15	Ps vs Pt Temperature	Ts - Tt ≤ 30.5ºF	1	1	1	1	1	1	1	1	1	1	D	1	1	1	1	1	1	1	1	1
C16	Digital Wraparound	Output = Input	1	1	1	1	1	1	1	1	1	1	D	1	1	1	1	1	1	1	1	1
C17	TAS DC POT Wraparound	h/w = s/w + 0.15V	0	0	0	0	0	0	1	0	0	0	E	0	0	0	0	0	0	0	0	0
C18	SAT DC POT Wraparound	h/w = s/w + 0.10V	0	0	0	0	0	0	1	0	0	0	E	0	0	0	0	0	0	0	0	0
C19	ALT Hold Sum Wraparound	h/w = ∑ s/w + 2.0V	1	1	1	1	0	0	0	0	0	1	E	0	0	0	0	0	0	0	0	0
C20	Airspeed Hold Sum Wraparound	h/w = s/w + 1.5V	0	0	0	0	0	1	0	0	0	1	E	0	0	0	0	0	0	0	0	0
C21	MACH Hold Wraparound	h/w = s/w + 1.5V	0	0	0	0	0	0	0	1	1	1	E	0	0	0	0	0	0	0	0	0
C22	MACH Trim 1 & 2 Wraparound	h/w = s/w + 0.65V	0	0	0	0	0	0	0	1	1	0	E	0	0	0	0	0	0	0	0	0
C24	TAS/SAT Reference	6.497 ≤ REF ≤ 6.817V	0	0	0	0	0	0	1	0	0	0	E	0	0	0	0	0	0	0	0	0
C25	H #1 Wraparound	h/w = s/w + 0.75V	1	1	1	1	0	0	0	0	0	1	E	0	0	0	0	0	0	0	0	0
C26	A #2 A #3 Wraparound	h/w = ∑ s/w + 0.75V	1	1	1	1	1	0	0	0	0	0	E	0	0	0	0	0	0	0	0	0
C27	ALT Fine Synchros Wraparound	12.8 ≤ SIN + COS ≤ 19.2V	1	1	1	1	1	0	0	1	1	0	E	0	0	0	0	0	0	0	0	0
C28	ALT Coarse Synchros Wraparound	12.8 ≤ SIN + COS ≤ 19.2V	1	1	1	1	1	0	0	1	1	0	E	0	0	0	0	0	0	0	0	0
C29	CAS Synchros Wraparound	12.8 ≤ SIN + COS ≤ 19.2V	0	0	0	0	0	1	0	1	1	1	E	0	0	0	0	0	0	0	0	0
C30	MACH Synchros Wraparound	12.8 ≤ SIN + COS ≤ 19.2V	0	0	0	0	0	0	1	1	1	0	E	0	0	0	0	0	0	0	0	0
C31	TAS Synchros Wraparound	12.8 ≤ SIN + COS ≤ 19.2V	0	0	0	0	0	0	1	0	0	0	E	0	0	0	0	0	0	0	0	0
C32	OPU Monitoring	Output = Input of OPU	1	1	1	1	1	1	1	1	1	1	D	1	1	1	1	1	1	1	1	1
C33	CPU	Result = 0	1	1	1	1	1	1	1	1	1	1	D	1	1	1	1	1	1	1	1	1
C34	APU	Result = 4.06643579807	1	1	1	1	1	1	1	1	1	1	D	1	1	1	1	1	1	1	1	1
C35	ROM	Sum = 0	1	1	1	1	1	1	1	1	1	1	D	1	1	1	1	1	1	1	1	1
C36	RAM	Load = Write	1	1	1	1	1	1	1	1	1	1	D	1	1	1	1	1	1	1	1	1
C37	Watchdog Timer	Does Not Time Out	1	1	1	1	1	1	1	1	1	1	D	0	0	0	0	0	0	0	0	0

NOTES:

1 D - ATC Disabled
E - ATC Enabled

2 1 - Failure Warning } Fail
0 - No Failure Warning } Safe

3 1 - Failure Code } Fail
0 - No Failure Code } Safe

4 Binary Word - Failure Code Transmitted
BCD Word - BCD Transmission Stops

TABLE A-2

CONTINUOUS MONITORING DEFINITIONS

Conditions 1 and 3 were "as designed" combinations of events and therefore did not require further discussion.

Conditions 2 and 4, however, were undesired events whose occurrences should be minimized to achieve an acceptable level of safety.

Equipment requiring self test typically had circuitry dedicated to signal processing and circuitry dedicated to self-test. Testing was implemented through software by programming essentially the same signal processing circuitry to perform the tests. This approach minimized false failure warnings (condition 2 above).

In evaluating the effects of failure mode on the computer outputs, the F/FA was also used as a basis for the analysis of the self test, which identified and provided a measure for conditions 2 and 4.

Failure Modes & Failure Rates:

Evaluation of Function Failure Modes:

Generally, each circuit function was reviewed and analyzed to establish the following:

- Input/output signals
- Function description: operation, criticality, purpose
- Failure Modes
- Components associated with each failure mode.

In cases where a logical function breakdown at the card level produced dependent functions as in Figure A-4, the failure modes of each function should be defined in terms of card/pin outputs instead of function outputs.

Cards of the form shown in Figure A-4 were analyzed in the following manner:

1. Function F1 failure modes were analyzed relative to the effects on card outputs C, D, and E.
2. Function F2 failure modes were analyzed relative to the effects on card outputs C and D.
3. Functions F3, F4, and F5 failure modes were analyzed relative to the effects on card outputs C, D, and E respectively.

This procedure was developed in order to reduce the complexity involved in analysis. The redundant failure modes it inherently creates were dealt with when the analysis was raised to the system level.

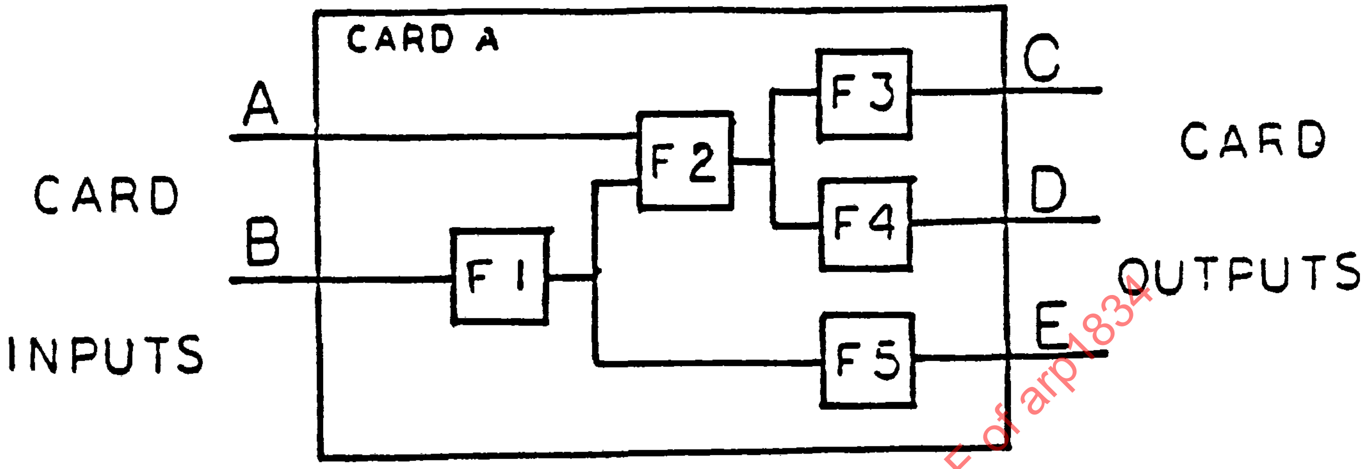


FIGURE A-4
Card Level Functional Breakdown

For this analysis, it was assumed that parts fail in the following modes:

- Passive Parts - Open and short with the exception of resistors which typically fail open
- Transistors - Open and short or leaky collector-to-emitter junctions
- Diodes - Open and short
- Simple digital I.C. gates - Outputs remain in a high, low or high impedance (where applicable) state
- Complex digital I.C.s - Intended function malfunctions or does not occur (e.g., clock runs too fast, too slow, skips pulse, etc.; significant bit in ROM is in error; counter does not advance, skips count; etc.)
- Linear I.C.s - Outputs remain at a high or low voltage which is usually the + or - supply voltage.
- Any part - open pin or solder connection.

Each printed circuit card assembly contained power supply filter capacitors which, if shorted, affected the power supply outputs. Rather than assess these effects on each card, they were considered in the power supply analysis.

In the case of the computer (Function F9) it was not realistic to consider only output pin "stuck-ats" (unless any failure resulted in simple latchup of the CPU). Consequently, fault descriptions such as "address bit error", "false flag set", etc., were used to describe chip malfunction.

The overall effects of the established function failure modes on the computer were documented in the format shown in Figure A-5.

Preliminary effects of failure modes were assessed relative to each output signal effect and coded in the following manner for later collection of similar effects:

- "x" indicating total loss of, or severe degradation of specific output ("y" would refer to a different output, etc.)
- "$\underline{x}$" indicating a conditional effect, or minor degradation of the output but still operating within specification.

A subsequent and more detailed study considered other failure effects of the equipment in definitive terms such as:

H - hardover (+, -)

S - slowover (n/sec)

F - stuck or frozen

B - bias error (n offset)

I - intermittent

G - gain error (n times normal)

TL - total loss

E - erratic (oscillatory or cyclic)

FL - fluctuating (n magnitude at "y" frequency)

"-" or NE - no effect (within performance limits).

Clarifying remarks on qualitative output effects and conditional effects were given in "Comments" column.

Functional Failure Rates:

Once the circuit functions were delineated on the system block diagram, they were located and partitioned on the schematic diagram.

The failure rates for each function were then calculated using MIL-HDBK-217 or an equivalent industrial standard. Mature failure rates (third year of operation) were used in these predictions. The functional failure rates were established generally by identifying those parts associated with each circuit failure mode. In those instances where the failure of a part results in multiple failure modes, the part failure rate was apportioned according to the number of modes to arrive at a failure mode failure rate. In some cases, an estimate was required of the percent of the chip circuitry affected by the failure mode being considered. The function failure rate is given in failures/million hours and are documented on each function sheet (see Figure A-5). If high confidence is lacking, then use of the total failure rate may be an acceptable conservative method.

The probability of failure was determined for each computer output by interpreting the data from the F/FA (output signal effect) sheets. This was accomplished by scanning down the column (see Figure A-5) on each of the sheets and noting which functions and failure modes affect the particular output. This was determined by the presence of an "X" or "x" or "-" in the column or other code for different effect or different output. The contributing failure rates for all effects, variously coded, were collated and totaled. The difference between this failure rate and the total function failure rate yielded the "no significant effect" failure rate. Probability of functional failure was then determined by considering the total exposure time between Self Test or In-flight Monitoring or ATE Testing that confirmed the operation of the circuit or function.

Functional Fault Detectability:

The detectability of each failure mode under consideration in the F/FA had to be assessed by the analyst, then noted in the DETECT column of the F/FA (Figure A-5). If detected, the method was noted in MEANS OF DETECTION, and the system response included in EFFECT OF FAILURE. Likewise, the failure warning annunciation was indicated, and, for reference, the self test number used for detection was noted. If the failure was UNDETECTED, this was indicated in the F/FA along with the resulting effect on the system.

Analysis and Summary:

An overall assessment of fault monitoring was required to analyze the distribution of detectable failures versus non-detectable failures. Unused components and "no effect" failures were not considered applicable to the fault monitoring detection assessment.

A summary of system effects breakdown with percentage of failure detected versus non-detected was determined. A summary of systems effects for the digital system example is shown below:

DETECTION SUMMARY

TOTAL	146.6
FLIGHT DETECTION	
Unannunciated	33.1
Nuisance Flag	5.7
GROUND DETECTION	
Undetected	27.6
Nuisance Flag	5.7

*Quantities are failures per million hours, based on mature (third year) reliability predictions

Based on the data from above, the following parameters were calculated:

1) In-flight detected BITE ratio =
100(146.6 - 33.1)/146.6 = 77.4%

2) Ground Maintenance detected =
100(146.6 - 27.6)/146.6 = 81.2%

3) Percent in-flight flags valid =
100((146.6 - 33.1) - 5.7)/(146.6 - 33.1) = 95.0%

4) Percent of total warning flags/lamps during ground maintenance that are valid =
100((146.6 - 27.6) - 5.7)/(146.6 - 27.6) = 95.2%

A review of the summary of system effects breakdown, Table A-3, enabled the analyst to evaluate undetected failures that could cause significant system failures. The analyst used the F/FA to isolate the source(s) of those undetected critical failures and to recommend additional tests or design changes as necessary to eliminate or reduce the probability of undetected failures.

TABLE A-3

SELF TEST DETECTION SUMMARY

FUNCTION	(1)	FAILURE MODE	(2)	FLIGHT (3)		GROUND TEST (4)	
NO.		NO.		UNANNUNCIATED	NUISANCE FLAG	UNDETECTED	NUISANCE FLAG
F1	0.0369	1	0.0279	0.0279	-	0.0279	-
		2	0.0075	0.0075	-	0.0075	-
F2	1.1797	3	0.3675	0.03675	-	0.3675	-
F3	0.0123	-	-	-	-	-	-
F4	0.0505	3	0.0050	0.0050	-	0.0050	-
F5	0.0369	1	0.0279	0.0279	-	0.0279	-
		2	0.0075	0.0075	-	0.0075	-
F6	1.1797	3	0.3675	0.3675	-	0.3675	-
F7	0.0123	-	-	-	-	-	-
F8	0.0505	2	0.0100	0.0100	-	0.0100	-
F9	0.3357	-	-	-	-	-	-
F10	0.0620	-	-		-	-	-
F11	0.0523	5	0.0003	0.0003	-	0.0003	-
		6	0.0003	0.0003	-	0.0003	-
		7	0.0003	0.0003	-	0.0003	-
		8	0.0003	0.0003	-	0.0003	-
		9	0.0003	0.0003	-	0.0003	-
		10	0.0003	0.0003	-	0.0003	-
		11	0.0003	0.0003	-	0.0003	-
		12	0.0003	0.0003	-	0.0003	-
		17	0.0003	0.0003	-	-	-
		18	0.0003	0.0003	-	-	-
		19	0.0003	0.0003	-	0.0003	-
		20	0.0003	0.0003	-	0.0003	-
		22	0.0003	0.0003	-	-	-
		23	0.0003	0.0003	-	-	-
		24	0.0003	0.0003	-	-	-
		29	0.0039	0.0039	-	-	-
		30	0.0039	0.0039	-	-	-
		31	0.0003	0.0003	-	0.0003	-
		32	0.0003	0.0003	-	0.0003	-
		34	0.0003	0.0003	-	-	-
		35	0.0028	0.0028	-	-	-
		36	0.0039	0.0039	-	-	-
		41	0.0003	0.0003	-	-	-
		42	0.0003	0.0003	-	-	-
		44	0.0003	0.0003	-	0.0003	-
		45	0.0003	0.0003	-	-	-
F12	0.0366	2	0.0040	0.0040	-	0.0040	-

PREPARED BY ________
DATE ________

FIGURE A-5

FAULT/FAILURE ANALYSIS (F/FA)

	FUNCTION/COMPONENT					DETECT					
ITEM NO.	NAME (P/N)	REF. DESIG	FAILURE MODE	EFFECT OF FAILURE ON FUNCTION/SYSTEM	OUTPUT SIGNAL EFFECT	Y	N	MEANS OF DETECTION	FAILURE RATE PER MILLION FLT HRS	EXPOSURE TIME	COMMENTS
1	CENTRAL PROCESSING UNIT	U49	OUTPUT ADDRESS BIT ERROR	PROM ADDRESS RAM ADDRESS AND PERHAPS PORT ADDRESS WILL BE IN ERROR. MOST CRITICAL IS THE PROM ADDRESS ERROR WHICH WILL CAUSE SOFTWARE SEQUENCE TO BE IN ERROR. COMPUTER MAY INDICATE NO COMPUTED DATA BUT PROBABLY JUST STOP.	X	Y		FAULT OBVIOUS	1.8387		
2		U49	OUTPUT DATA BIT ERROR	DATA TO BE STORED IN RAM OR WRITTEN TO I/O PORTS WILL BE IN ERROR. CHECKERS TESTS WILL FAIL.	X	Y		CHECKERS TESTS			

SHEET 1 of 2 REV____

PREPARED BY ________
DATE ____________

FIGURE A-5 (Cont'd.)

FAULT/FAILURE ANALYSIS (F/FA)

ITEM NO.	FUNCTION/COMPONENT NAME (P/N)	REF. DESIG	FAILURE MODE	EFFECT OF FAILURE ON FUNCTION/SYSTEM	OUTPUT SIGNAL EFFECT	DETECT Y	DETECT N	MEANS OF DETECTION	FAILURE RATE PER MILLION FLT HRS	EXPOSURE TIME	COMMENTS
3		U49	INSTRUCTION BIT ERROR	COMPUTER IS LIKELY TO INDICATE NO COMPUTED DATA OR JUST HALT. ERRONEOUS OUTPUT MAY ALSO OCCUR BUT IS NOT LIKELY. INOP LAMP WILL BE ON. FAILURE WILL BE DETECTED BY CPU INSTRUCTION TEST.	X	Y		FAULT OBVIOUS OR TEST NO. 7			
4		U49	DATA MANIPULATION ERROR	SOME OR ALL OF THE ARITHMETIC AND OR LOGICAL MANIPULATION OPERATIONS WILL BE IN ERROR. BOTH NORMAL AND SOFTWARE BITE PROGRAMMING WILL BE AFFECTED. FAILURE WILL BE DETECTED BY CPU INSTRUCTION TEST.	X	Y		SAME			

SHEET 2 of 2 REV____