RADC-TR-83-72
In-House Report
March 1983

# THE EVOLUTION AND PRACTICAL APPLICATIONS OF FAILURE MODES AND EFFECTS ANALYSES

Heather B. Dussault

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED





ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, NY 13441

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-83-72 has been reviewed and is approved for publication.

APPROVED:

ANTHONY J. FEDUCCIA
Chief, Systems Reliability & Engineering Branch
Reliability & Compatibility Division

APPROVED:

EDMUND J. WESTCOTT
Technical Director
Reliability & Compatibility Division

FOR THE COMMANDER:

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC ( RBET ) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| REPORT DOCUMENTATION PAGE | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|
| 1. REPORT NUMBER: RADC-TR-83-72 | 2. GOVT ACCESSION NO.: AD-A131358 |
| 3. RECIPIENT'S CATALOG NUMBER: | |
| 4. TITLE (and Subtitle): THE EVOLUTION AND PRACTICAL APPLICATIONS OF FAILURE MODES AND EFFECTS ANALYSES | 5. TYPE OF REPORT & PERIOD COVERED: In-House Report |
| | 6. PERFORMING ORG. REPORT NUMBER: N/A |
| 7. AUTHOR(s): Heather B. Dussault | 8. CONTRACT OR GRANT NUMBER(s): N/A |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS: Rome Air Development Center (RBET) Griffiss AFB NY 13441 | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62702F 23380212 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (RBET) Griffiss AFB NY 13441 | 12. REPORT DATE: March 1983 |
| | 13. NUMBER OF PAGES: 120 |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Same | 15. SECURITY CLASS. (of this report): UNCLASSIFIED |
| | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A |

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

FMEA Techniques
Failure Mode
Failure Effect
Fault Tree Analysis
Sneak Circuit Analysis
Failure Analysis
Matrix FMEA

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Failure effects analysis allows a product to be studied early in its design and development stages where undesirable failure effects can be identified and readily corrected. This report is intended to give the reader a broad, general background in techniques available for failure effects analysis and their usefulness. Sixteen separate techniques, ranging from tabular failure modes and effects analysis and fault tree analysis to lesser known and more recently introduced techniques such as

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

hardware/software interface analysis, are discussed. The current status and prospects for the future of failure effects analysis are also discussed in the report.

# PREFACE

This report surveys the evolution of failure effects analysis techniques from the 1950s to the present. Widely used and established techniques as well as state-of-the-art developments are presented. This report is meant to give the reader a broad, general background in techniques available for failure effects analysis and their usefulness. The report should also give the reader an appreciation of the value of failure effects analysis to the related areas of systems analysis such as: maintainability analysis; testability analysis; reliability predictions; safety analysis; failure analysis; and logistics support.

# 0.0 EXECUTIVE SUMMARY

## FAILURE EFFECTS ANALYSIS TECHNIQUES

Failure effects analysis is a very broad area of systems analysis. Simply, a failure effects analysis is an organized and logical study of how a system reacts to failures. Because of the diversity of systems and the range of results which can be obtained, many different techniques have been developed for failure effects analysis. Ultimately, failure effects analysis is a design tool, indicating the strengths and weaknesses in a system design and providing information for allied analyses in maintainability, testability, logistics, reliability prediction, and safety.

## COMMONLY USED FAILURE EFFECTS ANALYSIS TECHNIQUES

The most commonly used techniques for a failure effects analysis are the tabular Failure Modes and Effects Analysis (FMEA), fault tree analysis, matrix FMEA, and sneak circuit analysis.

### Tabular FMEA

The tabular FMEA is the grandfather of all other failure effects analysis techniques. The tabular FMEA employs a very simple approach. A table, or worksheet, is used to itemize every probable failure mode and its resulting effect. The specific information contained on the worksheet can be tailored to the individual system, but usually includes: item identification, failure mode, probable failure causes, failure effect, method of fault detection, and any remarks concerning corrective actions or design changes. The level of detail contained in the analysis

is determined by the availability of information and the intended application of the results. The analysis can also include an evaluation of the relative importance of failure modes based upon the severity of their effect on the system and their probability of occurrence. The combined analysis is then referred to as a Failure Modes, Effects, and Criticality Analysis (FMECA). FMEA/FMECA is a very versatile technique. It can be used to analyze any system at any stage in its design.

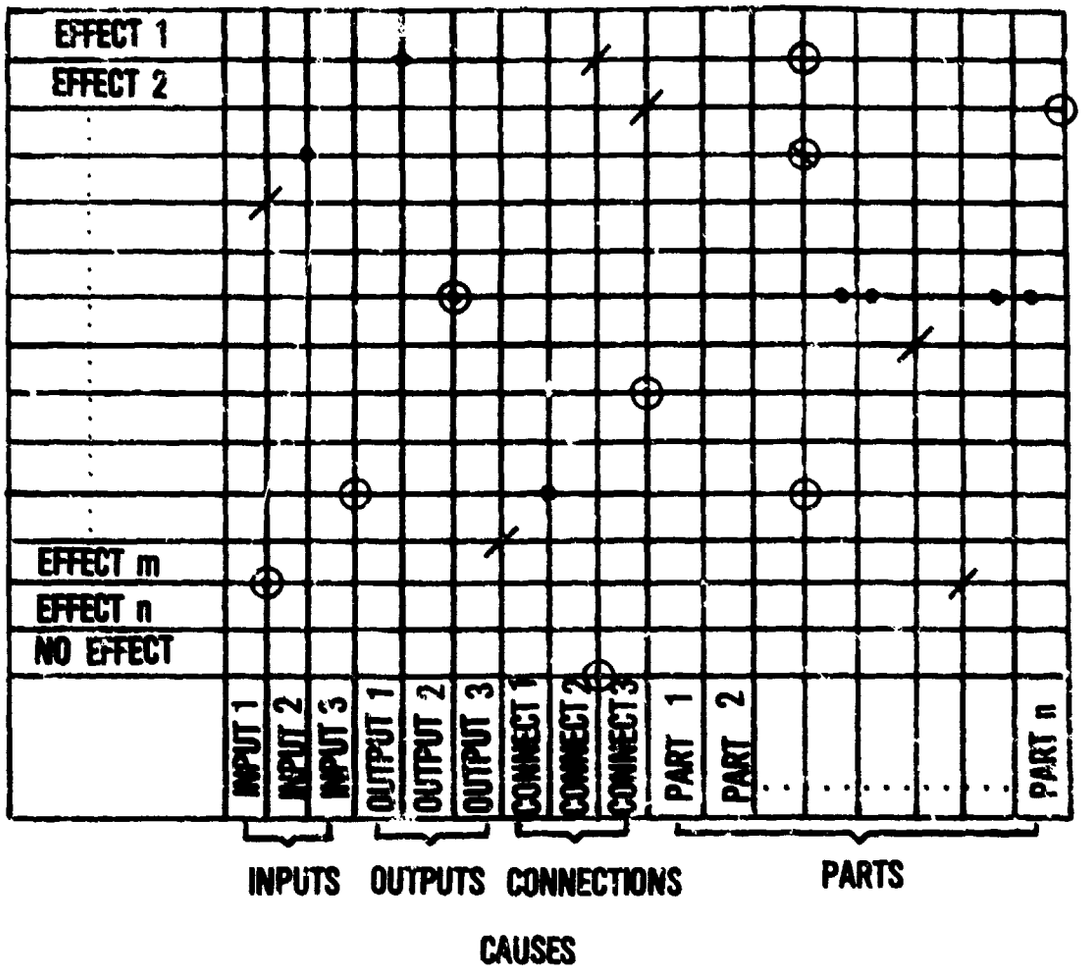
Fault Tree Analysis

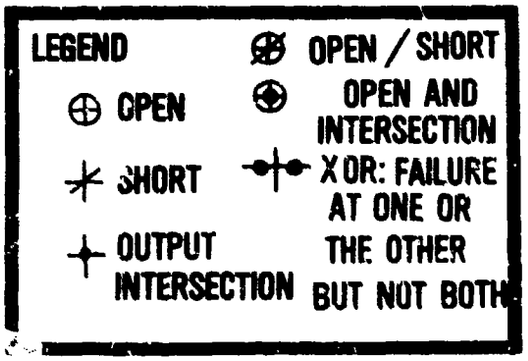
Fault tree analysis was so named because the completed analysis resembles a many branched tree. In fault tree analysis a specific undesirable system state or failure (top event) is defined. The fault tree is then developed using deductive reasoning and the principles of Boolean logic. Beginning with the defined top event, the immediate causes of the event are identified and connected to the event by logic operators. Each of the immediate causes then becomes the event to be developed in similar fashion. The fault tree continues to branch out until only events which cannot be further developed remain. Once the fault tree has been constructed it can be analyzed using Boolean algebra to identify which combinations of events result in the occurrence of the undesired event. Like tabular FMEA, fault tree analysis is a versatile technique. The fault tree, however, represents a distinct departure from the tabular FMEA because it uses deductive reasoning rather than inductive and it examines combinations of failures and externally influenced failures.

## Matrix FMEA

Matrix FMEA is very similar to a tabular FMEA in intent, but differs in its approach. A matrix FMEA is performed using a gridded plot to graphically indicate the relationship of failure modes and failure effects. The horizontal axis of the grid is used to represent inputs, outputs, connections, and parts of the system. The vertical axis of the matrix is used to identify the failure effects. A symbol is placed at grid locations at which the failure of the corresponding part on the horizontal axis produces the associated effect on the vertical axis. Different symbols are used to represent different failure modes. A sample matrix FMEA is shown in Figure 1. The analysis proceeds to higher levels of analysis in a "build-up" fashion. The most detailed level of analysis feeds directly into the next higher level analysis. The effects produced by the failure modes of the more detailed analysis become the sources of failure at the next higher level of analysis. The build-up process enables the cause of the ultimate system effects to be directly traced back to the most detailed information available. In short, Matrix FMEA has the versatility of tabular FMEA for electronic systems, and it can also be represented graphically.

FIGURE 1: MATRIX FMEA



## Sneak Circuit Analysis

Sneak circuit analysis is used to find design errors which could result in improper or undesirable system operations. The analysis is performed in later design stages for detailed "as-built" circuitry. In sneak circuit analysis the circuit or flow system is characterized as a combination of standard topological patterns. Any circuit can be partitioned into the standard patterns. For each pattern a series of clues has been developed to aid the analyst in identifying the existence of sneaks. Sneaks can be found in four areas: 1) sneak path, which allows current to flow in the wrong direction; 2) sneak timing, which occurs when a circuit function happens at an unexpected time or does not happen when it should; 3) sneak label, which improperly identifies a control or measurement; and 4) sneak indication, which results in incorrect or ambiguous displays of system conditions. Sneak analysis is the only technique which examines design induced errors.

## LESSER KNOWN FAILURE EFFECTS ANALYSIS TECHNIQUES

The lesser known failure effects analysis techniques are not as widely used as the four techniques previously mentioned. That is not to say, however, that the lesser known techniques are not as useful for failure effects analysis.

## System State Phase Modelling

System state phase modelling uses a logic diagram to investigate all possible system states. The logic diagram resembles a tree with switches along each branch. Each switch represents a specific event's

occurrence. Each branch represents a particular order of events occurring in each of the system operating phases. All possible paths are considered for the first operating phase. The paths are then further developed for subsequent phases based upon the previous phases. Once a complete system failure has occurred in a path, the path is not further developed. The logic diagram includes paths for operational, degraded, and failed system states. Unlike previously discussed techniques, system state phase modelling allows the entire operational history of the system to be carried through the analysis.

Tabular Systems Reliability Analysis

Tabular systems reliability analysis combines aspects of the tabular FMEA, fault tree analysis, and Markov chain theory. The analysis involves partitioning the system into ten or less functional blocks. A tabular format is used to identify all possible operating states for each functional block. The effects of combinations of states from each functional block are then evaluated. Much like fault tree analysis, an overall undesirable effect on the system can be defined, and all those combinations of states which produce the undesirable effect can be determined. Markov chain theory can then be used to numerically evaluate the probability of occurrence of the undesirable effect. Because only a small number of functional blocks are used, the analysis is not intended to be highly detailed. But, unlike tabular FMEA, the technique examines combinations of failures and provides a quantitative system analysis.

## Event-Sequence Analysis

Event-sequence analysis traces the effects of system failures as a function of the order in which they occur. The event-sequence map is a tree-like structure; the branches of the map represent the time sequenced order of faults. The map shows all probable failure histories for system operation. All events examined in event-sequence analysis are dependent, (e.g., given that A and B have previously occurred, now that C has occurred, the effect on the system is ...). The event-sequence map is developed by investigating all possibilities for a first failure event, then all possibilities for a second failure event, and so forth until all branches of the map end in a system failure. Numerical evaluation can be made of the event-sequence map using conditional probabilities of event occurrences. Event-sequence analysis allows the dependencies of failures in a system to be logically analyzed.

## "Testability Analysis"

Testability analysis is actually a separate systems analysis from failure effects analysis. It has been suggested, however, that an automated program originally intended for testability analysis can replace the need for FMEA in electronic systems. An electronic circuit simulator, such as LAZOR or TEGAS 5,could also be used to simulate faults. The proprietary automated testability program uses detailed "as-built" design information to evaluate the effects of standard electronic failure modes for each part in the system circuitry. The program also evaluates the completeness of fault detection and isolation in the system. No

numerical evaluation or criticality evaluation is presently included in the automated program. The automated program can eliminate much of the effort and tedium involved in performing a tabular FMEA for circuitry. Additional tailoring information and analysis of unconventional failure modes would either have to be sacrificed or externally supplied.

L.A.M. Technique

The LAM technique evaluates the effect of failures on a system by modelling how failures change the physical properties of the system. Both failed and operational system states are examined. Parametric equations are developed to model system response to failures based on the characteristic physical properties of the system. A specific undesired system state can be defined, and those conditions which can result in the undesired state can be defined either by operational status or by physical characteristics. The LAM technique can be used to provide an integrated analysis of both hardware and physical conditions. A system analyzed using the LAM technique, however, would have to be one that is easily modelled using physical parameters.

Approachability Analysis

Approachability analysis is used to evaluate the effects of failures caused by approach. Failures caused by approach are failures caused by the improper relationship of parts, failures caused by the introduction of foreign materials into the system and failures caused by external stresses. The analysis uses a matrix format. Those items susceptible to failures caused by approach or likely to cause an approach are identified

along with external stresses. All items which will fail as a result of an approach are labelled objective parts and placed along one axis of the matrix. The approaching parts and external stresses are placed along the other axis of the matrix. Much as in matrix FMEA, those combinations of objective part and approaching part or external stress which result in a failure are marked. Design and layout changes can be suggested by the results of approachability analysis. Approachability analysis examines types of failures which are often overlooked but which can readily occur as a result of consumer use or operation in other stressful environments.

Failure Combination Method

The failure combination method evaluates the effects of single, multiple, and externally influenced failures in a system. All the effects of single failures are obtained from a previously performed tabular or matrix FMEA. Those single failures and combinations of failures which produce the same effect are grouped together and called gathered failures. Externally influenced failures are defined as failures which occur in systems outside the analysis which affect the system under study. Overall failures are defined by combining the externally influenced failures and previously defined gathered failures which produce the same overall effect on the system. The grouping scheme allows single, multiple, and external failures which produce the same effect to be identified using an inductive approach.

## HARDWARE/SOFTWARE FAILURE EFFECTS ANALYSIS TECHNIQUES

Hardware/software failure effects analysis techniques have been developed to accommodate the growing need for failure effects analysis in complex hardware/software integrated systems. Hardware/software integrated systems have not proven to be amenable to the more conventional failure effects analysis techniques because of the vast number of possible system configurations.

### Software FMEA

Software FMEA is very similar to tabular FMEA. Software FMEA defines the functional requirements of the software and then evaluates the causes for failure to meet those requirements and the effect that the function failure will have on the system. Finally, the feasibility of eliminating or guarding against unacceptable failures is investigated. Software FMEA is usually performed during early design stages to verify that software performance requirements are being properly implemented and to identify areas where redundancy or fault tolerance are needed.

### Software Sneak Analysis

Software sneak analysis is very similar to sneak circuit analysis, both in intent and approach. Software sneak analysis identifies design errors or "bugs" in system software. The flow of logic in a computer program is used for the sneak analysis. The logic flow pattern of a program is divided into combinations of the six standard software topological patterns. The same basic types of clues used in sneak circuit analysis are used for software sneak analysis. The results, however, are

much different. Besides locating branch bypasses and infinite loops, software sneak analysis can locate sneak outputs, sneak inhibits, sneak timing, and sneak messages.

Integrated Critical Path Analysis

Integrated critical path analysis examines hardware/software interrelationships in a system. The technique combines aspects of tabular FMEA, fault tree analysis, and sneak circuit analysis. Integrated critical path analysis begins by developing network tree models, as is done in sneak analysis. The network trees indicate hardware/software interfaces and can be used to update previously performed fault tree analyses. The updated fault trees can then be analyzed to determine critical system failure paths. Operations and maintenance procedures are also reevaluated using a tabular FMEA approach. Finally, any relevant failure analysis is also incorporated into the integrated critical path analysis. Integrated critical path analysis offers a comprehensive evaluation of both hardware and software responses to system failures.

Hardware/Software Interface Analysis

Hardware/Software interface analysis considers software requirements as a function of hardware failures. Because software cannot respond to unanticipated conditions, it is important to recognize every probable failure condition and to determine how the software should respond. Every failure mode in a previously performed FMEA is examined for potential hardware/software interface problems. A series of questions are asked of every probable failure mode to determine if software

anticipates hardware failures and provides workarounds, if software utilizes the full capability of the hardware, and if the software overstresses the hardware. Hardware/Software interface analysis examines how software can be used to improve system performance.

Microcomputer FMEA

It is often difficult to determine the exact overall system effects resulting from a microcomputer failure. Different operating modes and memory configurations affect the way the system responds to a microcomputer failure. The microcomputer FMEA technique is performed on an operating system. A fault simulator is connected between the hardware system and the microcomputer. With the system running, the fault simulator generates a variety of faults (e.g., open and stuck at ground) for every input and output pin. The effect of each simulated fault can be entered on a tabular type FMEA worksheet.

THE STATUS OF FAILURE EFFECTS ANALYSIS

Failure effects analysis is often viewed as a necessary evil. The intent of failure effects analysis is to allow a system to be examined early in its development when undesirable failures can be identified and readily corrected. Unfortunately, the use of failure effects analysis is limited by problems in its application. The most common problems encountered in failure effects analysis include:

- The analysis is time-consuming and costly;
- The analysis results and recommendations are often obtained too late in the design to be easily instituted;

- Accurate failure data are difficult to obtain; and
- The level of detail necessary for a thorough, economical and effective analysis is difficult to accurately determine.

The future of failure effects analysis belongs to the efficient. Efficient guidelines and techniques are needed for the full potential of failure effects analysis to be realized. Guidelines need to be efficient in detailing what is required for specifying, conducting, and reviewing a failure effects analysis. The techniques used for failure effects analysis need to be efficient in:

- time and cost,
- thoroughness and accuracy; and
- applicability to related analysis areas, such as logistics, testability, and maintainability.

It is important that the potential of failure effects analysis to influence design and to support other analyses be recognized.

# TABLE OF CONTENTS

## LIST OF FIGURES



## LIST OF TABLES

# 1.0 INTRODUCTION

## 1.1. Background

### 1.1.1. Why Failure Effects Analysis?

In the Post-World War II era there were rapid technolcgical advances, but product reliability did not keep up with advancing technology. The examination of failure modes and their effects was a natural outgrowth of the need to increase product reliability. Failure effects analysis allows a product to be studied early in its design and development stages where undesirable failures can be identified and readily corrected.

### 1.1.2 Failure Effects Analysis Definition

Failure effects analysis is a very broad area of systems analysis. Very generally, a failure effects analysis can be considered an examination of a design or system in a logical and organized manner. The failure effects analysis can be applied through many approaches. The varied approaches include: an itemization of each part in the system and how each part can fail and the consequence of each failure; a characterization of the system by function and an analysis of how each loss of function would affect the system; a deductive analysis of exactly which failures or combinations of failures will result in an undesirable system state; or a time dependent model of dependent failures and their consequences. Failure effects analysis is a design tool. It can be applied to any system or any procedure which affects the system. An effective failure effects analysis presents a thorough examination of a

system's strengths and weaknesses in a timely and understandable manner.

1.1.3. Role of Failure Effects Analysis in Design and Development

The failure effects analysis forms the cornerstone for many further systems analyses. The failure effects analysis provides: a data base for maintainability, logistics support, and safety analyses; a reference point and verification source for testability analysis; a basis for trouble-shooting procedures; a focal point for reliability predictions, even providing the analytical foundations and expressions for evaluation; and a basis for design changes or addition of redundancy in design. When implemented early in the design phases, the failure effects analysis can provide guidance for necessary design changes which otherwise would not become evident until much later in the development cycle. Failure effects analysis is also an iterative design tool. The analysis can evolve as the design evolves and can, when required, provide a means of evaluating proposed engineering design changes.

1.1.4 Scope of Report

This report will examine the techniques which can be used in a failure effects analysis. Only those techniques which represent a fundamentally different means for analyzing failure effects will be discussed in this report. Sensitivity and tolerance analyses will not be discussed in this report because they do not directly analyze failure causes and effects. Common cause, grounding, and accident analyses will not be treated as separate techniques, since different failure effects analysis techniques incorporate aspects of each analysis. The intent of

the report is to provide an overview of techniques available to assess the effects of failures on a system. The report surveys the evolution of failure effects analysis techniques from the 1950s to the present. Widely used and established techniques as well as state-of-the-art developments are presented. The report discusses the general procedure involved in each technique, developments within the technique, its applications and relative merits.

This report is not meant to be a "cookbook" to be used to perform a failure effects analysis. This report is meant to give the reader a broad, general background in techniques available for failure effects analysis and their usefulness. The report should also give the reader an appreciation of the value of failure effects analysis to the related areas of systems analysis such as: maintainability analysis; testability analysis; reliability prediction; safety analysis; failure analysis; and logistics support.

## 1.2 General Comments on Failure Effects Analysis

### 1.2.1 Problems with Failure Effects Analysis

A failure effects analysis is not intended to be the panacea for all poorly designed and operated systems. It is intended to be a design tool. As with all other analyses, failure effects analyses are subject to inconsistencies and inaccuracies. No one technique provides all the means for deriving a complete analysis for all products and systems. Each technique has a different approach with distinct advantages and disadvantages which dictate its use.

Failure effect analyses also have logistical problems. A failure effects analysis is often expensive, manpower intensive, and time consuming. Massive amounts of paperwork are often generated in the course of the analysis. Interfaces between the analyst and the design engineer are often tenuous, which results in an inefficient and sometimes inaccurate analysis.

### 1.2.2 General Suggestions for Failure Effects Analysis

It is important to realize that while a failure effects analysis cannot do the entire job, every attempt should be made to assure that the analysis effort has been organized and exhaustive. Each technique presented in this report can be used in a failure effects analysis. No matter which technique is used, the following suggestions can help provide a design- and cost-effective analysis.

1. The failure effects analysis should be initiated as early as practical, dependent upon level of analysis and technique employed.

2. A thorough understanding of the system and of the analysis technique is necessary.

3. The failure effects analysis depends upon the support of management and good engineering practices.

4. The failure effects analysis should be performed for all operating modes and mission phases.

5. The failure effects analysis should be performed at increasing levels of detail as the design progresses.

6. The analysis should be traceable.

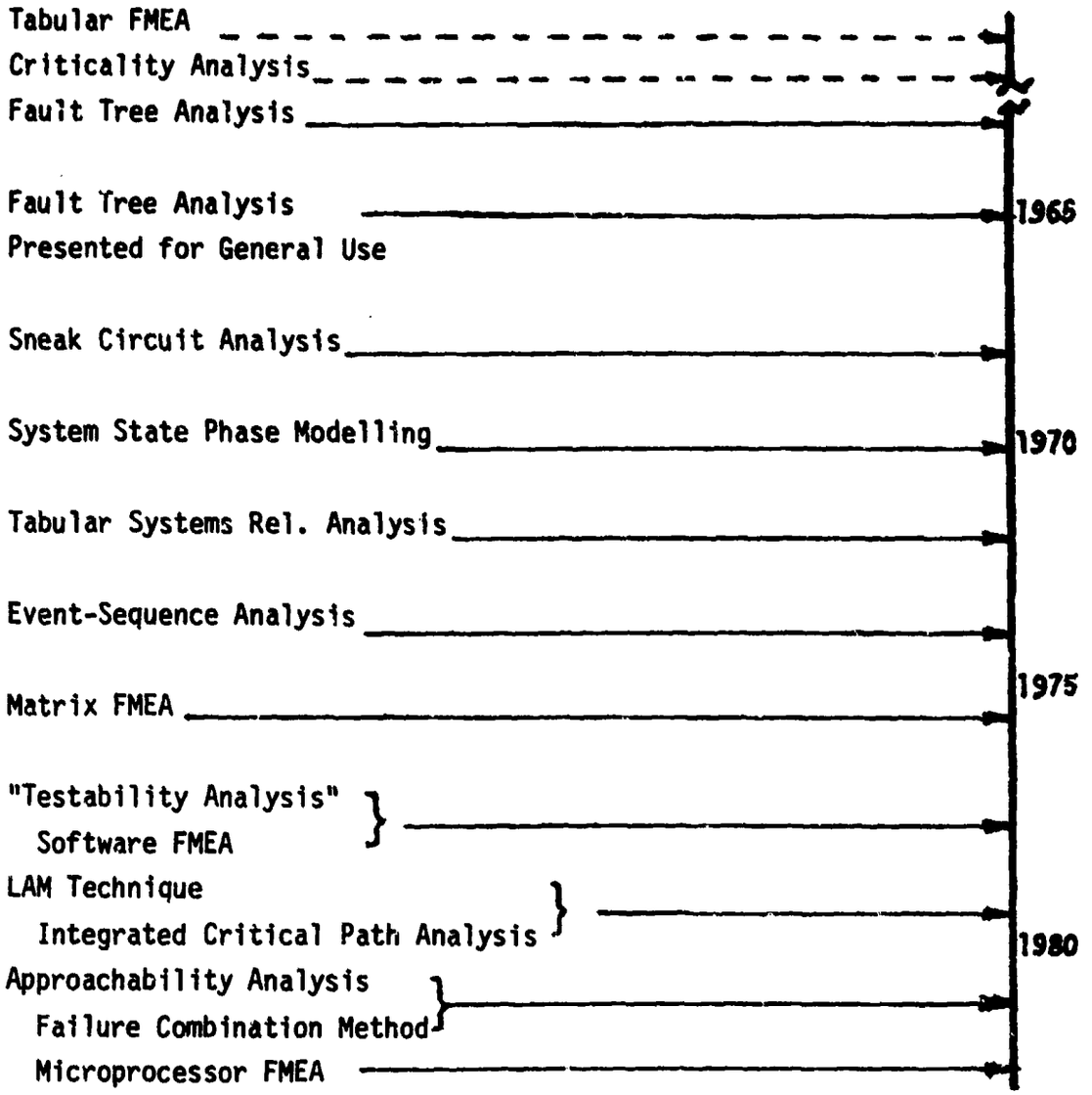
7. The failure effects analysis should be followed up and updated as the design changes and matures.

## 1.3 Types of Failure Effects Analysis

### 1.3.1 Specific Techniques

The specific techniques investigated in this report range the entire spectrum of approaches available for a failure effects analysis. In an evolutionary sense, the development of new techniques has not fostered the elimination of earlier techniques, but has served to augment the older techniques or to provide a specific new approach for a specific problem area. This report examines the classical or tabular failure effects analysis and FMECA (Failure Modes and Effects Criticality Analysis); fault tree analysis; matrix FMEA (Failure Modes and Effects Analysis); sneak circuit analysis; several lesser known techniques, including phase modelling, event-sequence analysis, approachability analysis, and the failure combination method; and new trends in failure effects analysis for software, integrated designs, and microprocessors. Figure 1.1 shows the evolutionary development of the techniques discussed in this report.

Generally, the techniques examined can be applied to any product or system. The results of the failure effects analysis can be both qualitative or quantitative. The broad range of techniques available suggests that selection of a technique for a given project is not simply




EVOLUTIONARY DEVELOPMENT OF FAILURE EFFECTS ANALYSIS TECHNIQUES
FIGURE 1.1

a function of new evolutions of techniques or general, acceptable practice. The selection of a technique for a failure effects analysis is dependent upon the level of effort necessary and what type of results are desired.

### 1.3.2 Related Guidelines/Materials

Further information on failure effects analysis can be found in the sources listed in the Bibliography, Section 9.0. These sources give a more detailed presentation of each technique, and an example is often included with each discussion.

For a more exacting discussion of what is required of a failure effects analysis it is suggested the reader refer to MIL-STD-1629A "Procedures for Performing a Failure Modes, Effects and Criticality Analysis" or any of several other Department of Defense, Nuclear Regulatory Commission, and NASA Standards for specific types of systems (e.g., SAMSO-STD-77-2, "Failure Modes and Effects Analysis for Satellite, Launch Vehicle, and Reentry Systems").

## 1.4 Format of Report

The report examines the major failure effects analysis techniques generally in use today. Tabular FMEA and FMECA are examined in Section 2.0; Fault Tree Analysis is examined in Section 3.0; Matrix FMEA is examined in Section 4.0; and Sneak Circuit Analysis is examined in Section 5.0. Section 6.0 discusses some lesser known techniques. Section 7.0 discusses failure effects analysis for new and rapidly developing areas such as software, microprocessors, and integrated systems. The

History, General Procedure, Future Developments, and Relative Merits of each technique are discussed.

Section 8.0 provides an overview and general evaluation of failure effects analysis, techniques discussed and the current status of failure effects analysis. Section 9.0 provides the Bibliography for the report.

# 2.0 CLASSICAL TABULAR FMEA TECHNIQUES

## 2.1 Tabular FMEA

### 2.1.1 History of Tabular FMEA

The tabular format was the first technique widely used for FMEA. Because of its longevity and widespread use, the tabular failure effects analysis is often directly associated with failure effects analysis and is often simply referred to as "the" FMEA. The heart of the tabular failure effects analysis technique is the detailed analysis of the effect of a specific failure mode on a system. The detailed analysis is usually presented in a table or worksheet.

The tabular failure effects analysis was the first attempt to develop a technique which gave uniformity to both the procedure used for the analysis and the type of information contained in the analysis. The tabular worksheet formats, formalized in the 1950s, are still widely used for the performance of FMEAs.

### 2.1.2 General Procedure for Tabular FMEA

The tabular failure modes and effects analysis can be initiated at any stage of design or development and at any level of detail. The level of analysis can range from assembly to piece/part indenture. The analysis can begin from the piece/part level and build to the assembly level - the "bottom-up" approach. Or, the analysis can begin from the assembly level and break down into the individual pieces/parts - the

"top-down" approach. Or, the analysis can be initiated at some intermediate level of indenture and proceed either "up" and "down"or some combination of the two.

The tabular analysis can be either hardware or functionally oriented. The hardware approach involves tabulating each individual component, and the effect of every probable failure mode of each component is analyzed. The functional approach involves tabulating equipment functions, and the failure modes contributing to the loss of the function are analyzed. The hardware approach is most often used when detailed design information is available. The functional approach is most often used in conceptual design stages.

A general procedure for performing a tabular FMEA consists of six steps.

1. Define the system and its performance requirements.
2. Define the assumptions and ground rules to be used in the analysis.
3. Develop a block diagram or other simple model of the system.
4. Devise the analysis worksheet and complete for every probable failure mode.
5. Recommend and evaluate corrective actions and design improvements.
6. Summarize the analysis in report form.

The definition of the system, system requirements, and assumptions used in the analysis are necessary for any failure effects

analysis. The definitions establish analysis boundaries, level of indenture, and the range of operating conditions. The block diagram of the system provides the analyst with a simple and useful representation of how the system operates and identifies its inputs, outputs, and interfaces. At the heart of the tabular failure effects analysis is the detailed worksheet. The worksheet is discussed in the following section. If the effects of certain failure modes are unacceptable, corrective actions or design changes can be recommended and their acceptability evaluated. Finally, the analysis should be formally documented and the findings presented in report form.

### 2.1.3 Failure Effects Analysis Worksheet

The major effort in performing a tabular failure effects analysis is the completion of the analysis worksheet. The worksheet usually contains 8-15 columns to be filled in by the analyst. Information to be provided on the form can include:

1. Identification of item or function;
2. Concise statement of function;
3. Operational conditions or mission phases considered;
4. Failure modes;
5. Failure causes;
6. Failure effects;
7. Ultimate effects;
8. Method of failure detection;
9. Possible corrective actions;

10. Effect of corrective actions; and

11. Remarks.

The identification of the item or function being analyzed is essential to provide traceability and consistency between design drawings and the analysis. A concise statement of the function of the item aids the analyst in recognizing how the item affects overall operation. It is important that all assumed operating conditions be included in the analysis for each failure mode. Different operating conditions can change the effect of failure modes. The itemization of all probable failure modes is the basis of the analysis. The inclusion of possible failure causes can assist in trouble-shooting and failure analysis. The immediate local effect of each failure mode is the principal finding of the analysis. Determining the ultimate effect of the failure can help to identify the need for corrective action. Noting the method of failure detection can aid in later testability analyses and in showing verifiable redundancy. In an effort to improve system reliability, corrective actions can be proposed and evaluated for those failure modes which produce unacceptable effects. The remarks portion of the worksheet allows the analyst to express concerns raised during the analysis.

2.1.4 Related Documents

The worksheet format is often devised according to specific system needs or contractual obligations. A sample tabular failure effects

analysis worksheet is shown in Figure 2.1. Specific procedural requirements and worksheet analysis items can also be found in documents governing the design and development of the specific system. Requirements vary in the amount of detail to be contained in the analysis (e.g., identification of interfaces, compensating features, and critical secondary failures for a non-detectable failure). Requirements also vary in the level of indenture and orientation of the analysis at various design stages.

### 2.1.5 Tabular FMEA Developments

The tabular FMEA has matured since its inception to include the criticality analysis. Refinements to the tabular FMEA have also been made to tailor each analysis to the specific system or type of system being analyzed. An example of a refined tabular FMEA is the Hitachi Failure Modes, Effects and Criticality Analysis (Hi-FMECA), Reference 24. The analysis incorporates previous experience and a Delphi-type process into the tabular FMEA. Another example of a refined tabular failure effects analysis is the Damage Modes and Effects Analysis (DMEA). In the DMEA the effects of specified threat mechanisms are analyzed for each essential component in the system. The DMEA is most often used to study the survivability and vulnerability of new weapons systems.

FIGURE 2.1 SAMPLE TABULAR FMEA WORKSHEET

| ITEM | IDENTIFICATION | FUNCTION | OPERATING PHASE | FAILURE MODE | FAILURE CAUSE | FAILURE EFFECT | | DETECTION METHOD | CORRECTIVE ACTION NEEDED | REMARKS |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | LOCAL | ULTIMATE | | | |
| | | | | | | | | | | |

Currently, efforts are being made to automate the tabular FMEA procedure. The use of word processing and text editing has made it possible for the worksheets to be filled out and updated on a computer terminal. The actual data input and analysis, however, are all performed by the engineer. Word processing has been effective in reducing the amount of time and paperwork involved in the analysis. Computer-aided design can also be used in conjunction with FMEA. Computer generated design drawings can be used with word processing to speed item identification. More sophisticated computer-aided design (CAD) techniques have been used for the analysis of electrical circuits. CAD programs can be useful in performing the analysis of failure effects resulting from opens, shorts, and other electrical faults. The CAD program models the circuit and will analyze the effects of individual faults on the entire system. A CAD program in conjunction with word processing capabilities and failure data base can help to "automate" most portions of a tabular FMEA. Developmental work is continuing in the area of automated FMEA.

### 2.1.5 Relative Merits of Tabular FMEA

The tabular FMEA is a very versatile and useful design analysis tool. The tabular FMEA can be used on any system and at any stage in design. The worksheet format presents the analysis in a logical and understandable fashion. The tabular FMEA provides a detailed analysis of each single failure at the chosen level of indenture. The nature of the detailed information included in the analysis can be tailored to fit the specific system through the structuring of the worksheet.

The tabular FMEA, however, has several shortcomings. A tabular failure effects analysis can be very costly to perform, generates large amounts of paperwork, and often requires a great deal of time and effort to complete. Further, many standards for the performance of the tabular failure effects analysis are weak and do not provide large amounts of guidance in the selection of indenture level and other ground rules for complex system analysis. The single line analysis of failure modes does not allow for the consideration of human/operator interactions and other system and environmental interactions. The tedium of filling out numerous forms can result in omissions and inaccuracies in the analysis. Multiple fan-outs of failure modes can be overlooked. Often, much time and effort is spent analyzing failure modes which have a negligible effect on system performance or safety.

The automation of the tabular failure effects analysis can help to overcome some of the inadequacies of the technique. Automation can help defray costs, reduce paperwork and time requirements, and help relieve the tedium of filling out the worksheets. Automation would still require the determination of indenture level for computer-aided design or system model. Engineering judgment would still be necessary for evaluating non-traditional failure modes and other special concerns.

## 2.2 Criticality Analysis

### 2.2.1 History of Criticality Analysis

The criticality analysis is often a complementary analysis to the tabular FMEA. Criticality is a relative measure of the importance of

a failure mode's effect on the successful operation of the system. The combined analysis is referred to as the Failure Modes, Effects, and Criticality Analysis (FMECA). FMECA was developed by NASA to assure that the hardware used in the space program was sufficiently reliable. Criticality analysis is used to determine the most sensitive or important areas of a design or to indicate where corrective actions should be started. The analysis can be semiqualitative or quantitative. Its results are often presented as a critical items list or in a criticality matrix.

### 2.2.2 Procedure for Criticality Analysis

The procedure used in performing a criticality analysis is straightforward. The difficulty in performing the analysis often comes in supplying the detailed failure rate information. The criticality analysis involves the following nine steps:

1. Identify the criticality of each failure mode in the tabular FMEA.
2. Supply generic failure rate ($\lambda_0$) for item and document the information source.
3. Supply operational and environmental failure rate modification factors ($K_a$, $K_e$).
4. Determine fraction of total failure rate attributable to each specific failure mode ($\alpha$).
5. Determine conditional probability that if the failure mode occurs the critical failure will occur ($\beta$).

6. Determine total component operating time (t).

7. Calculate criticality of component

$$C_r = \sum_{n=1}^{j} (\alpha K_a K_e \lambda_o t \beta)_n \qquad \text{Equation 2.1}$$

n = specific failure mode

j = total number of failure modes for component

8. Document the analysis.

9. Summarize the analysis results in the form of a critical items list or criticality matrix.

The identification of critical failure modes is accomplished by categorizing the ultimate effect of each failure mode. The military identifies the criticality of each failure mode as one of the four following severities:

I - Catastrophic (A failure which may cause death or system loss)

II - Critical (A failure which may cause minor injury, minor property damage, or system damage which will result in loss)

III - Marginal (A failure which may cause minor injury, minor property damage, or system damage which will result in delay or degradation)

IV - Minor (A failure which will not cause injury or loss, but which will cause unscheduled maintenance)

Other, more specific, categories can be used for analyses, but the military standard categories are typical of most. The criticality analysis can be pursued for those severity categories of interest.

The quantification of criticality can either be explicit or semiqualitative. The explicit criticality calculation uses generic failure rate data, from sources such as MIL-HDBK-217, modified by environmental and operating factors, also available from handbooks. Some fractions of failure rates attributable to specific failure modes can also be found in handbooks. The probability of the critical event occurring given the critical failure mode occurs ($\beta$) is determined as part of the analysis. Conservatively, the $\beta$ term would be assigned a value of 1.0. Once the component operating time has been determined, the component's criticality is a straightforward calculation using Equation 2.1. The semiqualitative criticality procedure is primarily used for relative comparisons or general rankings when detailed information is not available. In the semiqualitative approach the expression for the failure rate and all its associated terms ($\lambda_o^{\beta} K_e K_a$) is estimated in a very general fashion. One often used estimating scheme (Reference 40) categorizes the failure mode probability as: frequent (>.2); reasonably probable (.1<x<.2); occasional (.01<x<.1); remote (.001<x<.01); or extremely unlikely (<.001). The criticality of each failure mode can then be compared; ranked in a critical items list; placed in a criticalilty matrix; or numerical estimates (median, upper bound, etc.) can be used

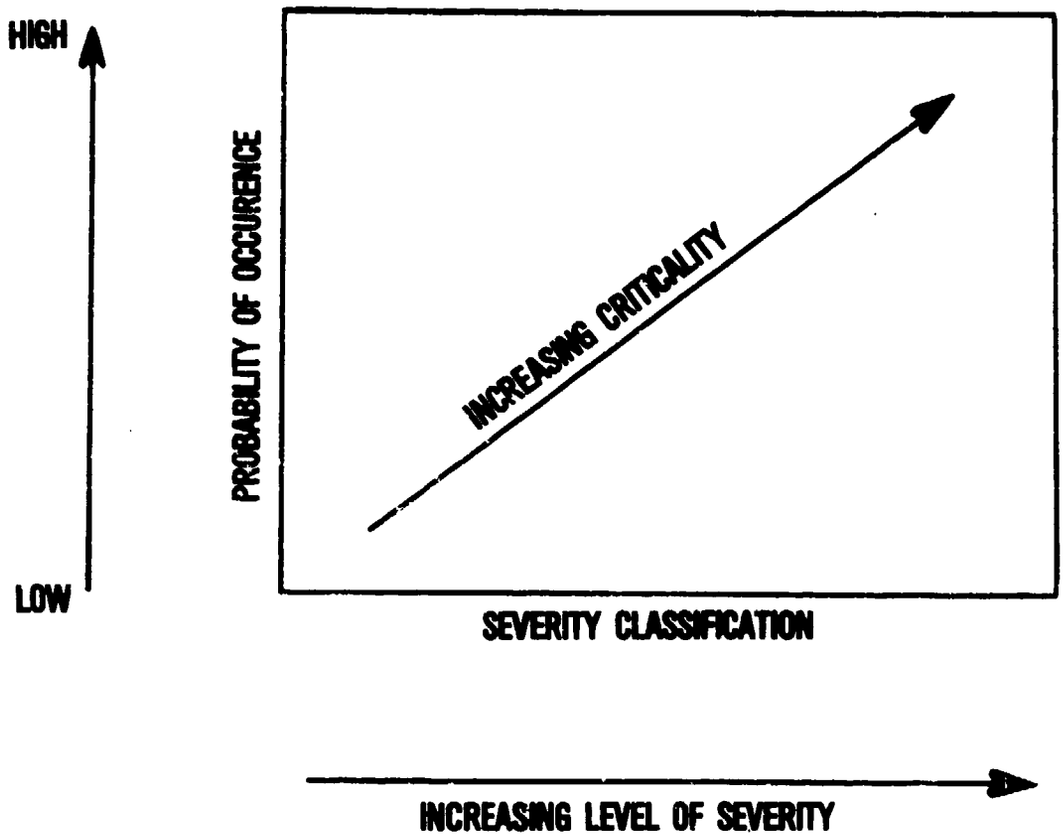
for each category of failure probability, and the numerical analysis can proceed.

Once the calculation of the criticality of each component in the failure effects analysis is completed, the analysis should be formally documented and the results summarized. The formal documentation of the analysis includes completing all the necessary worksheets, including data sources used and assumptions made. The results of the criticality analysis can be presented either in the form of a critical items list or criticality matrix. A critical items list ranks failure modes by their calculated criticality. The criticality matrix places failures on a matrix plot of the probability of occurrence versus severity classification. Figure 2.2 shows a schematic of a criticality matrix.

### 2.2.3 Developments in Criticality Analysis

The major advances in criticality analysis have not been in the area of techniques, but in the area of data. The increasing volume of failure rate data and modifying factors has helped to make the criticality analysis more rigorous. The inclusion of criticality analysis in an automated tabular failure effects analysis is highly desirable. The automation of criticality analysis would require the creation of a failure rate and modifying factor data base. Engineering input is required to classify the severity of the failure effect and to determine the conditional probability of the critical event occurring given the failure mode occurs. The calculations, however, could all be performed automatically. The computer could also generate a critical items list

FIGURE 2.2 CRITICALITY MATRIX SCHEMATIC



and/or criticality matrix.

### 2.2.4 Relative Merits of Criticality Analysis

Criticality analysis is a useful complement to the tabular FMEA. The criticality analysis provides an easy means of ranking the relative importance of failures as a function of both severity of effect and probability of occurrence. The results of the criticality analysis can help in determining the need for corrective action and its priority. The results can also help in determining the need for further part derating or alternate part selection. The automation of criticality analysis in a FMECA would help reduce analysis time and cost.

The criticality analysis has its limitations. The technique described in this report is for use with a tabular failure effects analysis (FMECA). The criticality analysis is not easily incorporated into the other types of failure effects analysis. Because the tabular FMEA examines only the effects of single failures, criticality analysis only examines the criticality of single failures. Human interactions and outside system interactions can not be evaluated by criticality analysis. The criticality analysis can not be more thorough or accurate than the tabular failure effects analysis upon which it is based.

Problems also exist in the quantification of the analysis. While handbooks exist for many types of components, some components have no accurate failure rate data available. The modifying factors and fractions of failures attributable to specific failure modes may be

difficult to locate or estimate. The lack of data can limit the analysis, but the semiqualitative approach and use of best estimates can provide a good basis for relative comparison of component criticalities.

## 2.3 Hazard Analysis

A type of analysis closely related to the tabular failure effects analysis and criticality analysis is the hazard analysis. Hazard analysis differs from criticality analysis. In criticality analysis the relative importances of failure modes are examined according to both their effect on the system and their probability of occurrence. Hazard analysis is performed to identify potentially hazardous conditions for either the system or system personnel. Three forms of hazard analysis will be briefly discussed: Preliminary Hazard Analysis, Operations Hazard Analysis, and Fault Hazard Analysis. All three analyses can be implemented using a tabular format.

### 2.3.1 Preliminary Hazard Analysis

The Preliminary Hazard Analysis (PHA) is initiated during the conceptual design phase. The intent of the PHA is to identify hazardous conditions for the system and personnel at an early stage in design. Once identified, the hazards can be compensated for, eliminated, or studied further.

The following steps define a general procedure for performing a PHA.

1. Review hazards identified in similar systems and previous designs. Determine those hazards which may be present in

the system.

2. Identify the events that could potentially create a hazardous condition.
3. Evaluate the effects of the hazardous condition.
4. Identify available compensation and control for hazard or suggest corrective action.
5. Provide results of analysis, corrective actions undertaken, and any additional remarks.

The tabular worksheet is not required to perform a PHA; a fault tree or narrative format may be used instead. The format most often employed, however, is the tabular worksheet.

### 2.3.2 Operations Hazard Analysis

The Operations Hazard Analysis is performed to identify any potential hazards created by operations on the system. This analysis is usually performed late in the design stages, because detailed knowledge of operations, system configuration, personnel, and conditions is required. A thorough analysis requires insight into all phases of operation and interactions between personnel and other systems.

A "top-down" procedure is used for an Operations Hazard Analysis. The following five steps define a general procedure for an Operations Hazard Analysis.

1. Identify all operations performed on the system.
2. Determine all potential hazardous conditions associated with the operations. Identification of the hazardous conditions

can be aided by a hazards checklist.

3. Break each operation down into its component tasks; each task into its associated procedures; and each procedure into individual steps.
4. Identify those elements of each operation which create the hazardous conditions.
5. Propose methods by which the hazardous conditions could be alleviated or eliminated.

The tabular worksheet provides a good format for completing the Operations Hazard Analysis in an organized manner.

2.3.3 Fault Hazard Analysis

The Fault Hazard Analysis is very similar to a FMECA. The Fault Hazard Analysis (FHA) is performed to identify hazardous conditions which may exist as a result of hardware failures. The FHA was developed as an analysis aid for the Minuteman III program. The technique is usually used for projects with many subsystem interfaces. The tabular format is used in the FHA to organize the analysis and to assure thoroughness.

The procedure involved in performing a FHA is very similar to that of a FMEA. Two additional items are part of a FHA: the identification of the upstream component that could command or initiate the fault; and factors that could cause secondary failures. A tabular format suggested by Haasl (Reference 11) consists of the following eight items:

1. Identification of component;

2. Evaluation of probability of failure;
3. Identification of failure modes;
4. Evaluation of percent failures by mode;
5. Determination of the effect of failure (including any potential hazards);
6. Identification of upstream component that could command or initiate the fault in question;
7. Identification of factors that could cause secondary failures; including operational and environmental variables; and
8. Remarks.

2.3.4 Relative Merits of Hazard Analysis

Hazard Analysis is a rather specific type of failure effect analysis. The intent of the Hazard Analysis is to assure greater system and personnel safety. The inductive, tabular form of Hazard Analysis is the same format as the tabular FMEA/FMECA. Because of the similarities of the analyses, they share many of the same strengths and weaknesses. The Hazard Analysis technique can present a thorough and logical analysis of all potential hazards in a system. Like the tabular FMEA/FMECA, Hazard Analysis can be exhaustive, but it also can be costly and time consuming.

The Hazard Analysis should not be performed independently of a tabular FMEA/FMECA. The information obtained in one analysis should be used to assist in further analysis. This information exchange can help to reduce analysis time and cost. Automation of the techniques would be

desirable, supplying Hazard Analysis as an additional analysis option.

# 3.0 FAULT TREE ANALYSIS

## 3.1 History of Fault Tree Analysis

After the tabular FMEA, Fault Tree Analysis was the next totally new technique to be formally developed for the study of failure effects. Fault Tree Analysis was developed in 1961 by Bell Telephone Laboratories to evaluate the Minuteman Launch Control System. The technique was first presented to the public in 1965 in Seattle. Fault Tree Analysis presented a radically different approach to failure effects analysis than that of the tabular FMEA. The fault tree used deductive reasoning (rather than the inductive approach of the tabular FMEA), and it examined the possibility of multiple failures using formal Boolean logic.

The technique has been given the name fault tree because the analysis begins by defining the ultimate failure effect of interest and then "branching out" to include all combinations of contributory failures. Because it first defines an ultimate effect, Fault Tree Analysis has often been used in safety studies. Its versatility, presentability, and logic have made Fault Tree Analysis a technique widely used for failure effects analysis.

## 3.2 General Procedure for Fault Tree Analysis

Fault Tree Analysis (FTA) can be initiated at any stage in design. The fault tree can be constructed to any level of detail, dependent upon the availability of design information and time and cost constraints. The fault tree can be analyzed mathematically by the same formal logic from which it was constructed. The analysis of the fault

tree can be either qualitative or quantitative.

The general procedure for Fault Tree Analysis consists of the following seven steps.

1. Define the system, ground rules, and any assumptions to be used in the analysis.
2. Develop a simple block diagram of the system showing inputs, outputs, and interfaces.
3. Define the top event (ultimate failure effect) of interest.
4. Construct fault tree for top event using rules of formal logic. Proceed with analysis to greatest level of detail possible.
5. Analyze completed fault tree.
6. Recommend any corrective actions or design changes.
7. Document the analysis and its results.

Fault tree analysis requires extensive knowledge of system operations and interactions. It is essential that the system boundaries, ground rules, and assumptions be clearly defined before attempting to construct the fault tree. An example of an assumption often used in FTA is that all inputs to the system are correct and within tolerances. A simple block diagram can make it easier to identify system boundaries and the locations of interfaces. The definition of the top event is crucial in determining the success of the analysis. The top event must be

carefully defined to insure that it represents as specific enough condition of interest to be analyzed. A top event which has been defined too generally can cause the analysis to become untractable, overly time consuming, and extremely costly.

Fault tree construction begins with the definition of the top event and proceeds in a "top-down" manner. The development of the tree is made possible using the principles of Boolean logic. Beginning with the top event, the immediate causes of the event are identified and connected to the top event through a logic operator, or gate. Each of these causes then becomes the event to be developed in similar fashion. The fault tree continues to branch out until only primal events ( events which cannot be developed further) remain. The logic operators (gates) which are used to express how the events are related to each other are shown in Figure 3.1. The logic operators represent an exact algebraic relationship between the underlying causes and the connecting event. A sample fault tree is shown in Figure 3.2.

FIGURE 3.1 FAULT TREE SYMBOLS

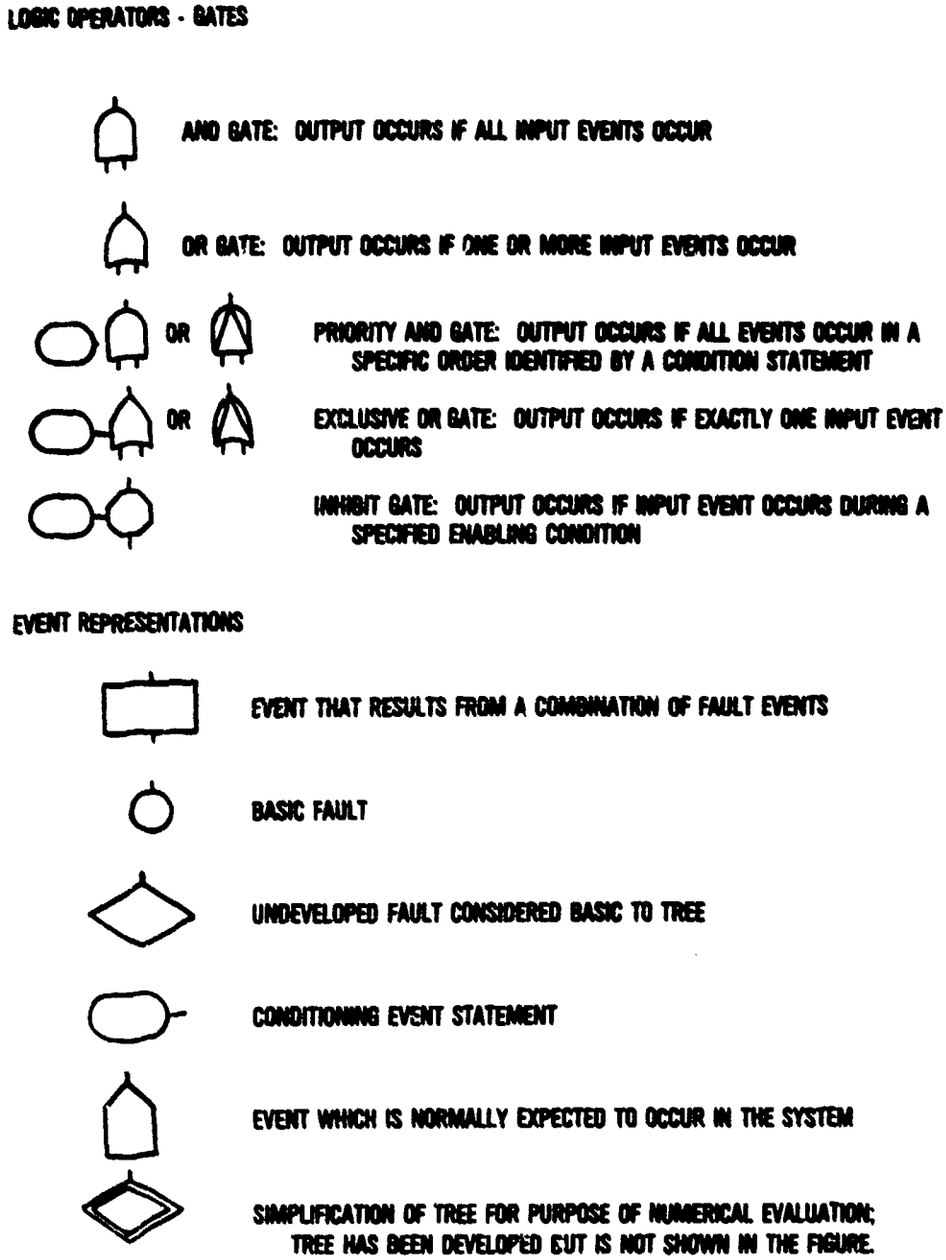
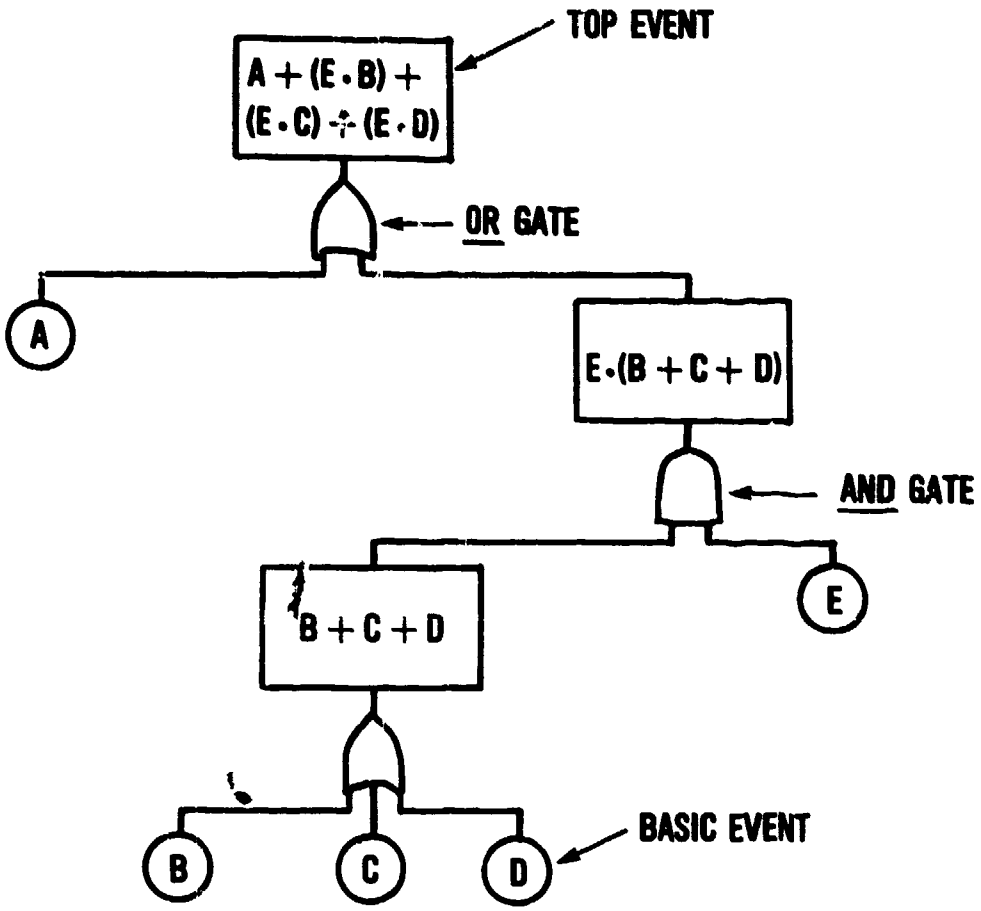


FIGURE 3.2 SAMPLE FAULT TREE



Several rules or principles govern the general procedure for the construction of fault trees. Three very general concepts to always consider when constructing a fault tree are:

1. Precisely define each event in the analysis; leave no ambiguity in the description; the precision of the event description will allow the construction to proceed logically without introducing extraneous causes;
2. Think small; immediate cause thinking will help to prevent the omission or oversight of faults; and
3. All basic inputs (or primal events) must be independent; the presence of unidentified or incorrectly represented dependencies invalidates the logic used to construct the tree.

The analysis of the fault tree identifies both the strengths and weaknesses of the system. Numerous methods and algorithms have been developed for the analysis of fault trees. Many variations of algorithms have also been developed to speed computer analysis of fault trees. The analysis results can be expressed qualitatively or quantitatively. The selection of a specific method is dependent upon the complexity of the tree and the type of data which is required for the analysis.

The qualitative analysis of a fault tree is accomplished by applying the rules of logic to the tree construction. The tree can then be reduced into a single analytical expression or into a group of minimum cut sets. A minimum cut set (min-cut) is defined as the smallest number of faults whose failure causes system failure. Correction of any fault

in a minimum cut set will restore the system to a successful state. Minimum cut sets can be used to obtain an expression for system unreliability. They can also be used to establish qualitative importances and critical rankings on the basis of number of elements in each set and the frequency of a particular item's inclusion in the cut sets. The minimum cut sets can also be used for common cause analysis. The susceptibility of the elements of each minimum cut set to common cause failures (such as fire) is examined. The qualitative aspects of fault tree analysis, however, are often overshadowed by the power of quantitative analysis. Several computer algorithms for qualitative fault tree analysis exist. Computer codes in the public domain which perform qualitative fault tree analysis include: PREP, ELRAFT, MOCUS, SETS, and ALLCUTS.

Quantitative fault tree analysis provides an estimate of the probability of occurrence of the TOP event. Direct analytical evaluation, cut set approximations, and Monte Carlo simulation techniques can be used for quantitative fault tree analysis. The major difficulty in any quantitative analysis comes in obtaining the data for the evaluation.

Quantitative analysis, however, can be a very powerful tool. The results also provide quantitative measures of the importance of components and minimum cut sets. Sensitivity studies can be performed to evaluate the effect of part selection, part derating, and other factors which affect reliability. Quantitative analysis using Monte Carlo simulation techniques can provide a statistical evaluation of the fault tree and provide confidence intervals for the probability of the occurrence

of the top event. Many computer programs for the quantitative analysis of fault trees exist. Those which exist in the public domain include: KITT, FRANTIC, WAM-BAM, WAM-CUT, PATREC, PATREC-MC, and SAMPLE.

### 3.3 Fault Tree Developments

The basic fault tree technique has not changed significantly since its introduction. The methodologies for fault tree construction and analysis have enlarged and matured. Fault tree construction methods have grown to include computer-aided generation of fault trees based on directed graphs (computer-aided synthesis, Reference 21) and decision tables (CAT method, Reference 28). Fault trees, however, are still often generated manually. New algorithms for fault tree logical reduction and numerical evaluation are still being developed to provide advanced analytical techniques and to improve computer calculation efficiency. Because of the breadth of the algorithms that have been developed, they will not be discussed in this report.

The applications of Fault Tree Analysis continue to broaden in scope. Primarily used for safety analysis, the fault tree can also be used for failure effects analysis. The fault tree can be used instead of a tabular FMEA to readily locate critical design areas. The fault tree is also being used in failure analysis to aid in location of failure causes. In a mutually beneficial manner, the fault tree can help to determine the cause of a failure, or a newly located failure cause can be included on an updated tree. Fault trees have also been used in conjunction with common cause analysis. Fault Tree Analysis is well suited for

the analysis of complex systems and systems with many interactions. As system designs become more complex, Fault Tree Analysis has the potential for even greater application.

### 3.4 Relative Merits of Fault Tree Analysis

Fault Tree Analysis is a powerful and versatile technique. Using a deductive approach to system analysis highlights the important failure aspects of a system design. The fault tree presents the analysis in an easily understandable and logical manner. The fault tree can readily illustrate strengths and weaknesses in the design. Fault Tree Analysis can also be performed at any stage in design. Because of its "top-down" deductive approach, the level of analysis is only constrained by the detailed information available. A major advantage found in Fault Tree Analysis is the ability to include external influences, such as human interactions, in the analysis.

Fault Tree Analysis, however, has its drawbacks. The logic used for the analysis needs to be rigorously defined or the system can easily be misrepresented. Improper representation of independent and dependent events is a problem common to fault trees. Because of the deductive approach, failure modes of a specific component can be difficult to trace back through the analysis. The deductive approach also makes the analysis susceptible to errors of omission and oversight. Like the tabular FMEA, Fault Tree Analysis can be very costly and time consuming. Computer analysis of fault trees can be especially expensive. Data for the numerical analysis are often difficult to obtain. Data insufficiencies

often result in the data being misapplied.

Developments continue to increase the power of Fault Tree Analysis. New algorithms help to reduce costs by improving computing efficiency. Computer-aided construction and analysis techniques can help to reduce time requirements and improve analysis accuracy.

# 4.0 MATRIX FMEA

## 4.1 History of Matrix FMEA

Matrix FMEA was the next major failure effects analysis technique introduced after Fault Tree Analysis. While the intent of the matrix FMEA is very similar to that of the tabular FMEA, it has been presented as a totally different technique because of its unique format. Matrix FMEA was first publically presented in 1977 by Barbour. The technique had been developed for the analysis of long life communications spacecraft. The use of matrix FMEA for electronic systems continues to grow as it becomes more widely known and accepted.

## 4.2 Procedure for Matrix FMEA

Like the tabular FMEA, matrix FMEA uses an inductive approach to failure effects analysis. The matrix FMEA presents an easily traceable analysis of the causes and effects of various system conditions. The matrix FMEA contains aspects of both hardware and functional analysis. The matrix FMEA considers system inputs, outputs, connections, and parts at any level of indenture. Failure effects analysis using matrix FMEA can be instituted at any stage of design. The analysis is a "bottom-up" procedure, and one level's analysis immediately feeds into the next higher level's analysis.

The format used for a matrix FMEA is a gridded plot of failure effects versus inputs, outputs, connections, and parts. A sample matrix FMEA worksheet with legend is shown in Figure 4.1. A vertical vector of the matrix will identify the set of causes and effects associated with a

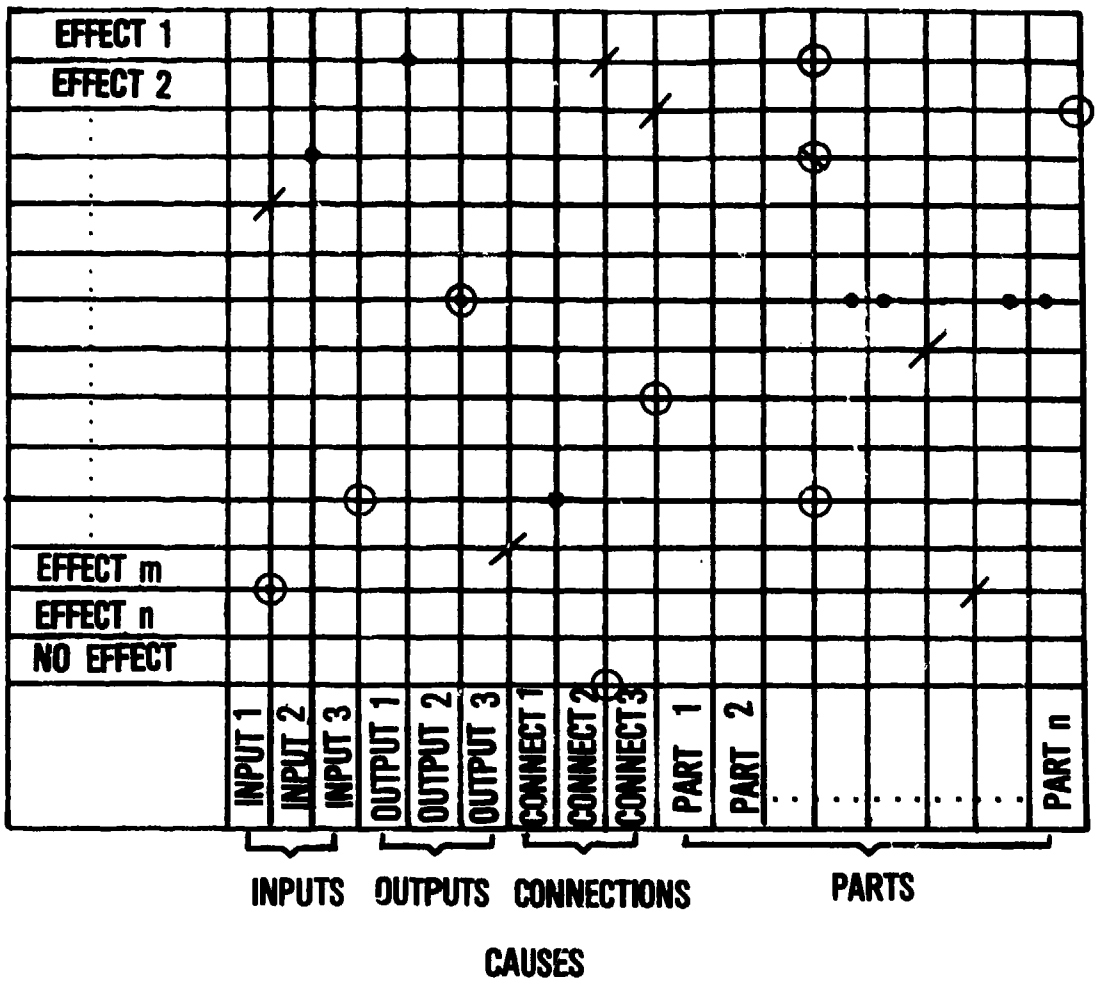
specific input, output, connection, or part. A horizontal vector of the matrix will identify the set of causes associated with a specific failure effect.

The procedure used in performing a matrix FMEA is straightforward:

1. Define the system, ground rules, and assumptions to be used in the analysis.
2. Construct a block diagram of the system.
3. Determine lowest level of indenture for the analysis.
4. Enter inputs, outputs, and components at the lowest level of indenture as the horizontal axis of the matrix.
5. Determine the effect of each failure mode for every input, output, connection, and part. Enter the appropriate symbol in the matrix to indicate the cause-effect relationship.
6. Proceed to next higher level of indenture until the entire system has been analyzed. The lower levels of analysis feed directly into the next higher level analysis. The effects of lower level failure modes become the failure modes at the next higher level of indenture. A schematic of the process is shown in Figure 4.2.
7. Analyze the results of the analysis.
8. Document the analysis.

The completed analysis process results in an easily accountable format.

Figure 4.1 SAMPLE MATRIX FMEA SHEET



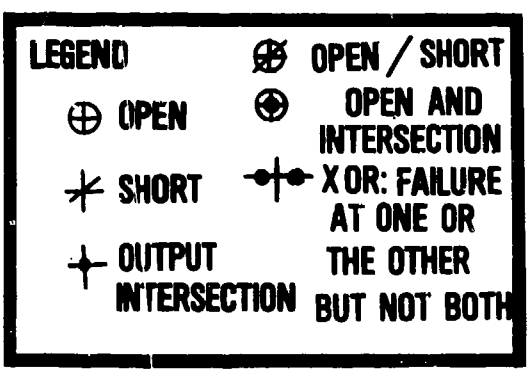
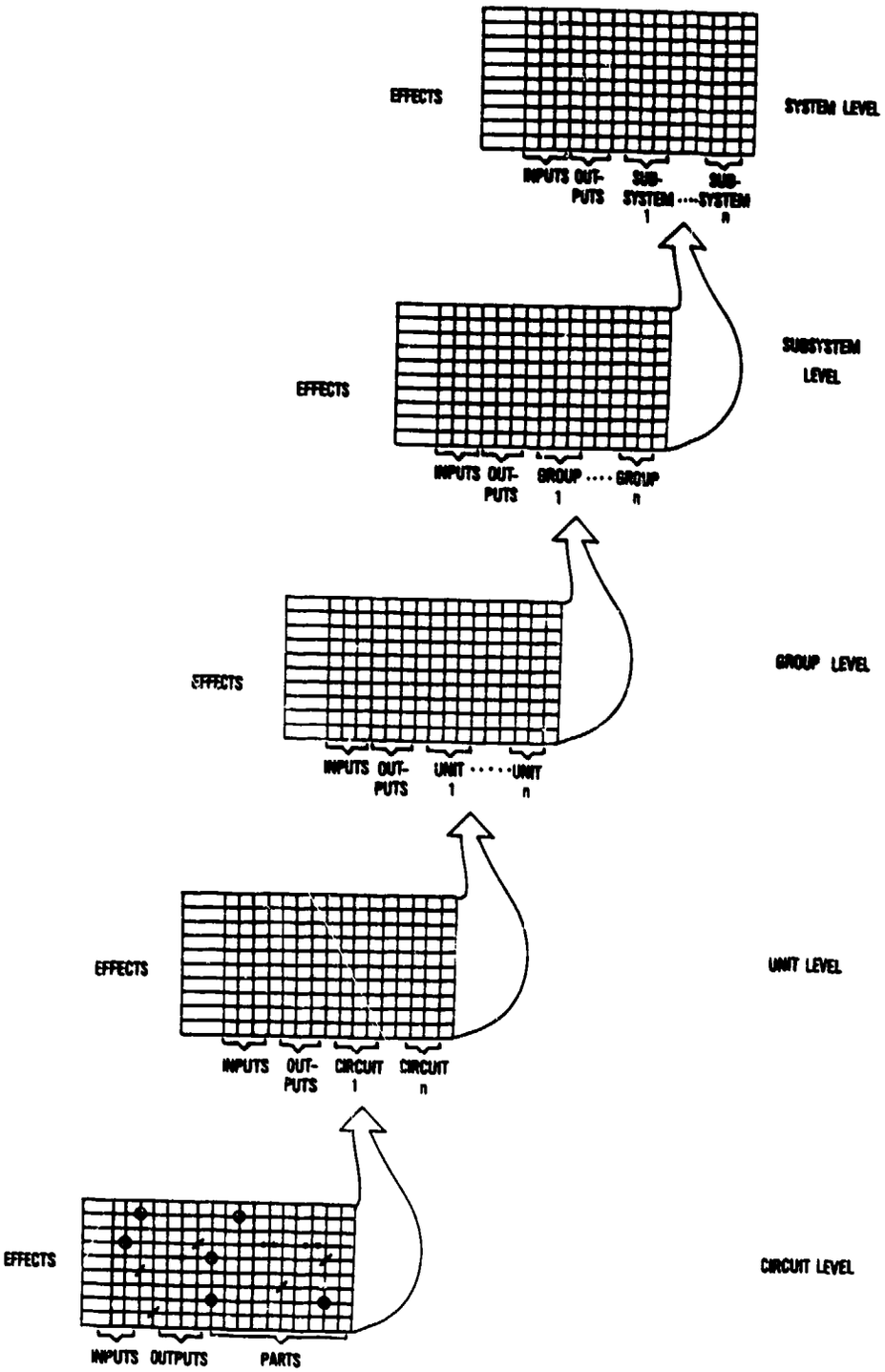



FIGURE 4.2 BUILD-UP PROCESS OF MATRIX FMEA (from Reference 1, Barbour 1977)



Critical items and failure modes can be readily found by tracing back from critical failure effects at the system level. Probabilities of failure can also be calculated for a given operating time by failure rate summation for a given effect. Corrective actions or recommendations can be made based upon analysis results. Once the analysis is completed, it should be formally documented. The formal documentation should include system definitions and assumptions, system diagram, completed matrices, and any results and recommendations.

The overriding intent of the matrix FMEA is to provide a traceable analysis of the effect of every probable failure mode from its immediate local effect to its ultimate effect on the system. The matrix FMEA procedure provides a graphical means for understanding the effects of individual failure modes. The technique can be as thorough and rigorous as any failure effects analysis.

## 4.3 Developments in Matrix FMEA

Although matrix FMEA was introduced in 1977, the technique has matured. Computer assistance has been developed for matrix FMEA. New applications for the technique have also been developed.

A computer program for matrix FMEA was developed in 1978 by Legg (Reference 22). FUME (FMEA Using Matrix Effects), as the program is called, checks a matrix FMEA for consistency and also calculates probabilities of failure. The data input to FUME is simple, but, as with all numerical evaluations, failure rate data sources must be treated carefully. The program translates the input into matrix line statements and

checks the statements for consistency. The consistency checks help to verify that no entry errors have been made in the input data. Probabilities of failure are calculated by obtaining the summation of failure rates along a horizontal matrix vector at each occurrence of the effect. The program calculates failure probability based on a user supplied operating time.

Because of the traceability of matrix FMEA, its use has been suggested for maintainability analysis, testability analysis, and system interface analysis. The completed matrix FMEA is used in maintainability analysis. A reverse process is employed in the analysis. The effects at the system level are traced back to their root failure modes. In this reverse manner, the lowest levels at which repair is practical can be determined for each failure. The reverse analysis can help determine critical items, spares provisioning, trouble-shooting procedures, and required maintenance actions. The reverse analysis technique for maintainability analysis has also been computerized. The reverse analysis is also used for testability analysis. Fault detection for BIT (Built-in-Test) design or test point determination can be evaluated by tracing failures back to lower levels of indenture. The necessary level of indenture for fault isolation or the ambiguity in the isolation at a given level of indenture can be determined using the matrix FMEA. Interface analysis can be performed using the matrix FMEA technique. Connections, solder joints, and wiring can be readily included in the matrix. The interface effects can be easily identified in the analysis because of

the direct "build-up" into the next higher level analysis. The interface analysis is useful for complex systems and for verifying redundancy. Using matrix FMEA for maintainability, testability, and interface analyses can help defray the costs involved with failure effects analysis.

Future developments in matrix FMEA will probably lie along the path of greater automation. Like the tabular FMEA, the matrix FMEA could be integrated with a circuit analysis program to further automate the process. The general trend in any matrix FMEA developments will be toward improved analysis time and cost. As matrix FMEA becomes more widely used, the range of its applications will continue to increase.

## 4.4 Relative Merits of Matrix FMEA

The matrix FMEA is a very useful design tool. A matrix FMEA can be initiated at any design stage, beginning at the lowest indenture level practical with given design information. The matrix FMEA presents a traceable and accountable analysis. Because of its traceability, matrix FMEA can be used to support maintainability, testability, and logistics efforts. The graphical format of the analysis makes it highly presentable and easily understood. The "build-up" structure of the analysis gives it organization and thoroughness in coverage of piece/parts and interfaces. The matrix format also allows the fan-out of failures to be more thoroughly covered. The matrix FMEA provides a detailed description of how each failure mode affects system performance.

Matrix FMEA does, however, have limitations. The matrix FMEA was

designed for use with electrical systems. Presently the technique is limited to a certain extent in applications to other types of systems. The structured analysis can also cause unconventional failure modes to be overlooked. Presently, matrix FMEA does not provide some of the detailed information that can be tailored into a tabular FMEA (e.g., method of failure detection). Like the tabular FMEA, the matrix FMEA is limited to the consideration of the effect of single failure modes with no consideration of external influences. Like most other failure effects analyses, matrix FMEA can be costly and time consuming.

Future developments in automation could help to reduce time requirements and costs. The use of matrix FMEA to support maintainability, testability, and logistics programs would also be advantageous.

# 5.0 SNEAK CIRCUIT ANALYSIS

## 5.1 Relationship of Sneak Circuit Analysis to Failure Effects Analysis

Many analysts familiar with sneak circuit analysis would not classify the technique as a failure effects analysis. Sneak circuit analysis does not involve postulating failures and then inductively or deductively analyzing the failure. Further, it is often assumed in sneak circuit analysis that all components are operating within tolerances. Nevertheless, sneak circuit analysis does deal with failures of a sort, design failures.

The failure effects investigated by sneak circuit analysis are much more insidious than normally expected hardware failures. Sneak circuit analysis is used to discover failures in the design which create unplanned operational modes in a circuit. The design failure is perhaps one of the least recognized types of failure. The effects of design failures are every bit as real and potentially as catastrophic as any component or operator failure. Perhaps the major difference between sneak circuit analysis and other forms of failure effects analysis is that other analyses require the specific cause or effect to first be defined for the analysis to begin; in sneak circuit analysis, while the effect may be known, neither cause nor specific effect needs to be defined for the analysis to begin.

## 5.2 History of Sneak Circuit Analysis

Sneak circuit analysis is used to find design failures which cause

unwanted functions to occur or which inhibit wanted functions. The technique was developed by Boeing for NASA in 1967. The technique is based upon the principle that a circuit can be topographically represented as a set of specific networks each of which can be analyzed for sneak circuits. Since its introduction, sneak circuit analysis has been widely and quite successfully used in a number of programs. Sneak circuit analysis has often been used to locate the source of failures when other techniques have failed.

5.3 General Procedure for Sneak Circuit Analysis

As indicated previously the technique used for sneak circuit analysis is based upon the recognition of topological patterns in a circuit and the analysis of each pattern for the presence of sneaks. The application of the technique requires detailed design information and "as-built" drawings. Because of the level of detail required, sneak circuit analysis cannot be performed in early design stages. Like other failure effects analyses, sneak circuit analysis is costly and time-consuming and hence should be selectively used. Most often, sneak circuit analysis is applied to mission critical systems and systems with many interfaces.

The sneak circuit analysis is capable of finding four different types of design errors, or sneaks, which are:

1. Sneak path;
2. Sneak timing;
3. Sneak label; and
4. Sneak indication.

The sneak path is a design failure which allows current to flow in an unintended way in the circuit. Sneak timing occurs when a circuit function occurs at an unplanned time or not at all. A sneak label is the result of a function, switch, or instrument reading which controls or indicates something other than what it is labelled. A sneak indication results in incorrect or ambiguous display of system conditions. The sneak conditions identified by the analysis present potential hazards. The sneak circuit analysis should be performed at a stage in design where corrections can be made in a timely and cost-effective manner.

Sneak circuit analysis can also identify other shortcomings in the design such as drawing errors and inconsistencies, poor design practices, and design inadequacies.

The procedure used for sneak circuit analysis is generic. A basic understanding of how the circuit to be analyzed works and its performance requirements is required of the analyst. No detailed knowledge of system design, failure modes, or operational environment is necessary. The "as-built" drawings provide the required detailed information. The procedure use for the technique consists of the following six steps.

1. Acquire the detailed "as-built" circuit drawings.
2. Process the drawings into a format suitable for analysis.
3. Produce the network tree of the circuit.
4. Identify the standard topographical patterns which make up the circuit.
5. Apply the clues for sneak identification.

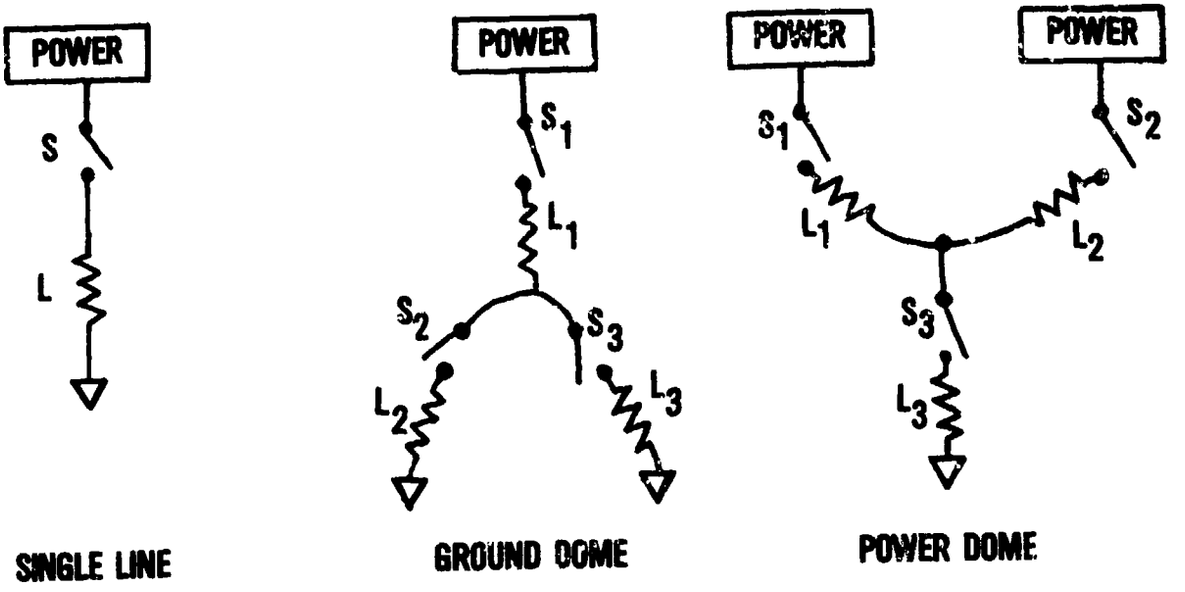
6. Produce sneak circuit reports and summary.

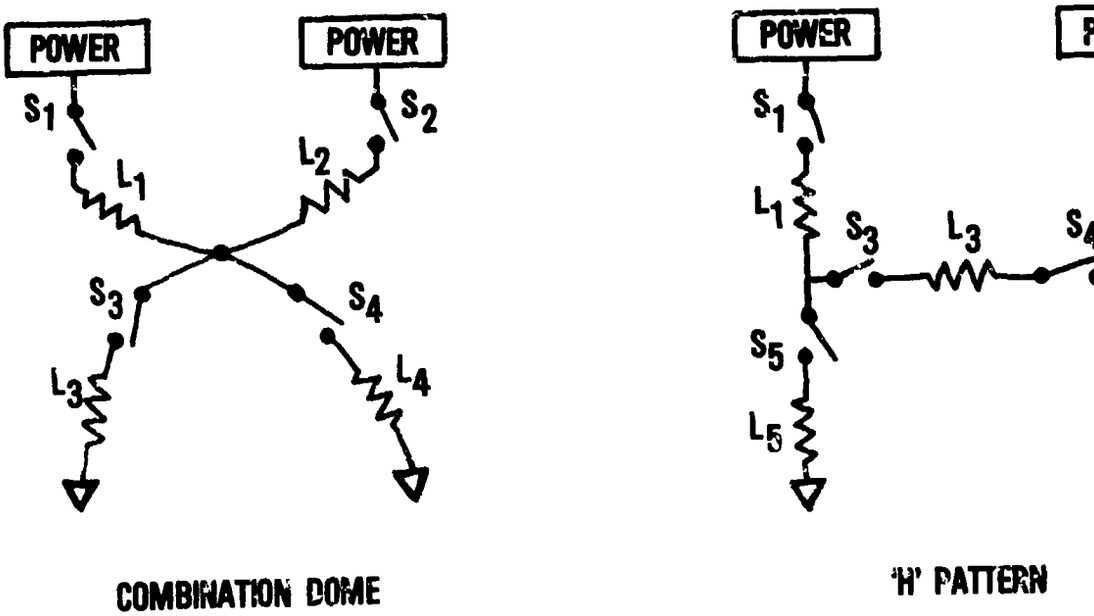
The detailed information is necessary to reduce the risk of missing sneaks at lower levels of indenture. The "as-built" drawings provide the best source for the necessary information. The topological orientation of the circuit can be obtained from the drawings. The circuit is first divided into nodes, then paths, and then nodal sets. The nodal sets are then simplified to contain all the important features of the circuit while reducing its complexity. The network trees which are produced by this process can then be examined for sneaks. The identification of sneak circuits is accomplished by breaking the network trees down into combinations of standard topological patterns. The five topological patterns are shown in Figure 5.1. For each pattern there is a set of "clues" which can lead to the identification of a sneak condition. An example of such a clue is: can the current flow be reversed? The "H" pattern checklist has over 100 clues. Once the sneak conditions have been identified, the analysis and the results should be formally documented. The formal documentation should include network trees, a description of the analysis, drawing error reports, design concern reports, and sneak circuit reports.

## 5.4 Developments in Sneak Circuit Analysis

Sneak circuit analysis has been expanded to include digital circuitry and software. Software sneak analysis will be discussed in Section 7. Digital logic sneak analysis was developed in 1975. Logic flow, instead of current flow, is investigated for the digital logic circuit.

FIGURE 5.1 STANDARD TOPOLOGICAL PATTERNS USED FOR SNEAK CIRCUIT ANALYSIS



Digital logic circuits use an expanded clue list that includes investigations for sneak timing and race conditions. Sneak circuit analysis has usually been applied to electrical circuits and software. The potential exists, however, for the technique to be used in the analysis of other flow systems, such as pneumatic, hydraulic, and other mechanical systems.

Computer assistance for sneak circuit analysis has been available since 1967. By 1970 most electrical circuits were analyzed for sneaks with computer help. (Digital logic circuits, however, are not easily handled by automation and are often done by hand.) The computer processes the input circuit topography. The data is converted into nodes, paths, and nodal sets. Nodal sets are indexed and cross-referenced by the computer. The computer also produces the network tree and identifies the topological patterns in the tree. The clue application can then be made by the analyst.

## 5.5 Relative Merits of Sneak Circuit Analysis

Sneak circuit analysis has become a widely used technique. It can often uncover the cause of a failure where other techniques have been inadequate. Sneak circuit analysis can be applied to any flow system, including electrical, digital logic, pneumatic, hydraulic, and some mechanical systems. Sneak circuit analysis is especially useful for complex and highly interfaced systems.

Sneak circuit analysis is a highly specialized, and therefore limited technique. Sneak circuit analysis will only identify design failures. The components of the circuit are all assumed to be operating properly within tolerances. Because of the level of detail required for the analysis, the analysis can be costly and time consuming and can produce results too late in the design effort to be cost-effective.

# 6.0 LESSER KNOWN FAILURE EFFECTS ANALYSIS TECHNIQUES

## 6.1 An Overview of the Lesser Known Techniques

The techniques which will be discussed in this section are not as widely used as the four techniques previously discussed. That is not to say, however, that the lesser known techniques are not as useful as the other well-known techniques for performing a failure effects analysis. In many cases the techniques have been developed for a specific type of system or have been developed recently enough that they have not been widely used.

In general, the lesser known failure effects analysis techniques attempt to overcome problems associated with tabular FMEA, matrix FMEA, and Fault Tree Analysis. The lesser known techniques often seek to provide a more thorough analysis while decreasing time and cost requirements. Some of the techniques have been automated or have had computer assistance programmed. The techniques will be presented in chronological order of their development. The techniques which will be discussed are:

- System state phase modelling (1969)
- Tabular systems reliability analysis (1971)
- Testability analysis (1979)
- L.A.M. technique (1980)
- Approachability analysis (1981)
- Failure combination method (1981)

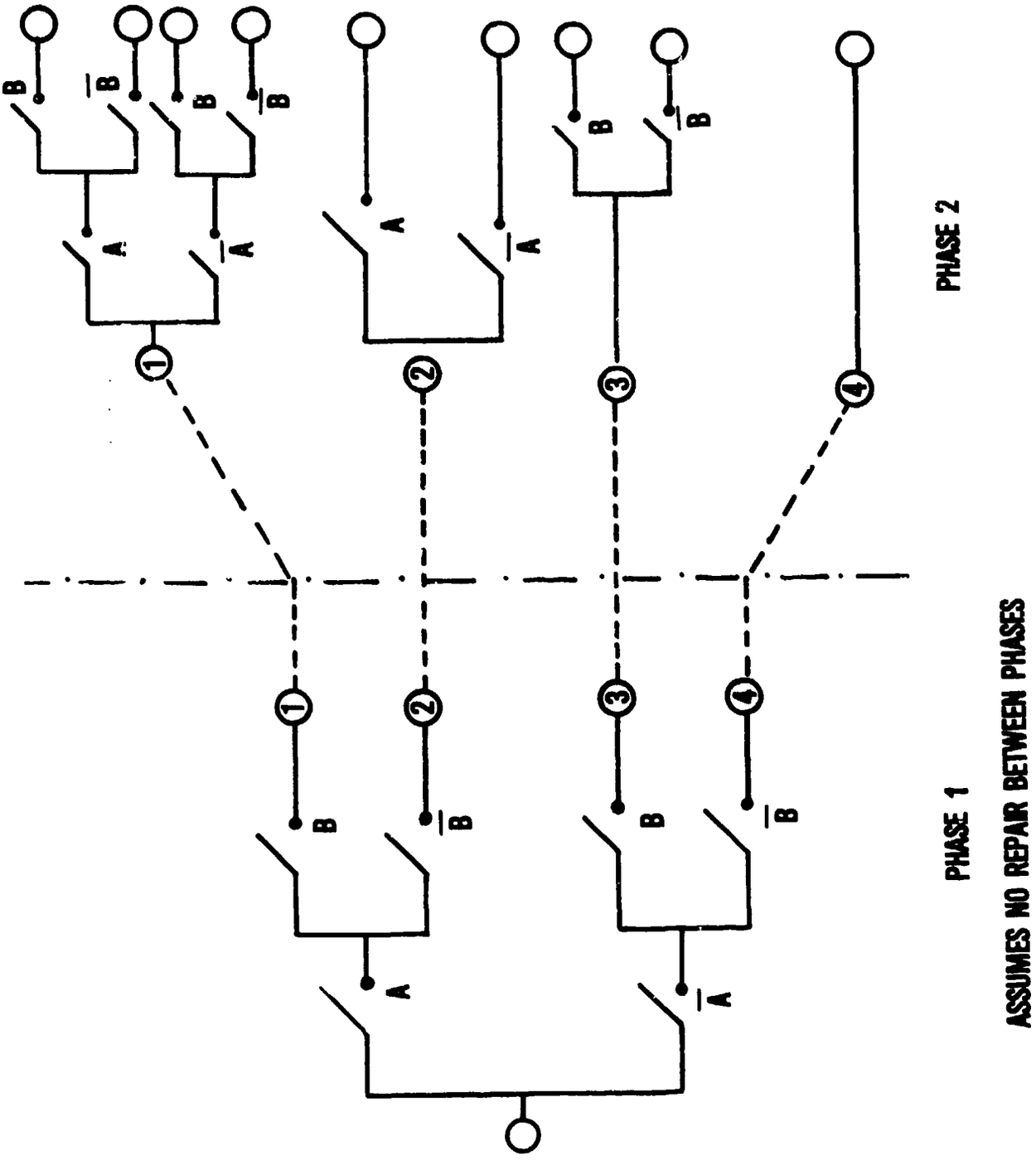
The general procedure and relative merits of each technique will be discussed.

## 6.2 System State Phase Modelling

System State Phase Modelling (SSPM) uses a logic diagram to investigate the possible states of an operating system. SSPM was first introduced in 1969 by Tiger (Reference 34). The model for the logic diagram is an electrical circuit. Each condition in the system is represented as a switch. Each path of the circuit represents a different system state. A sample logic diagram used for SSPM is shown in Figure 6.1. (The diagram in Figure 6.1 assumes no repair.) The system state phase modelling technique uses a bottom-up approach but also incorporates time dependence into the analysis. The first mission phase or operating phase of the system is modelled using the bottom-up approach. At the end of the phase all possible system configurations have been considered: operational, degraded, or failed. The analysis is then continued to the next phase by expanding all those previous states which allow continued operation. The SSPM technique can include consideration of single failures, multiple failures and external influences.

Because of the logically structured development of the diagram, each circuit path represents a mutually exclusive expression of a system state. The probability of each state can be calculated by multiplying together the probabilities of each of the individual conditions which form the path. The probability of a final system condition (e.g., failed or degraded) can be obtained by simply summing the mutually exclusive

FIGURE 6.1 SYSTEM STATE PHASE MODELLING LOGIC DIAGRAM



probabilities of each path which results in the condition of interest. The nature of system state phase modelling allows complex systems to be analyzed without the requirement of advanced mathematics, such as Boolean algebra.

System state phase modelling is very useful for mission analyses or analyses of systems with several operating phases. It is important in a failure effects analysis to include every mission phase in the analysis. SSPM also presents each failure effect in the proper time frame. The complete failure history for each system state is contained in the analysis. The bottom-up approach helps to prevent errors of oversight in the analysis. Because the analysis identifies all operational histories, all system states (success, failed, degraded, and hazardous), are included in the analysis. The SSPM technique also allows external influences to be analyzed. SSPM has the potential for using the computer to assist in the analysis, both for checking the thoroughness and consistency of the logic diagram and for performing any calculations. The technique can be initiated at any stage in the design.

Like other failure effects analysis techniques, system state phase modelling has its limitations. The analysis can be very time consuming. Numerical analysis of the logic diagram can be performed if the proper conditional failure data is available. The effects of a specific failure mode can be difficult to isolate. While operating conditions are detailed in the SSPM, failure causes and fault detection methods are not

included in the analysis. The analysis, however, does provide a comprehensive examination of system states and the history of events leading to each state.

## 6.3 Tabular Systems Reliability Analysis

Tabular systems reliability analysis, presented in 1971 by Thatcher, et al (Reference 33) of Battelle Laboratories, was developed to be an integrated technique for system reliability prediction. Tabular systems reliability analysis (TASRA) combines elements of fault tree analysis, state variable (Markov) techniques, and tabular FMEA into an integrated analysis. The technique uses a bottom-up approach. Various system conditions are first defined, and the effects of the conditions are then evaluated. The analysis can be performed at any stage in design and can be applied to numerous types of systems.

The procedure involved in performing a tabular systems reliability analysis consists of four major steps:

1. Define the system, system requirements, and ground rules for the analysis;
2. Prepare simplified block diagram;
3. Generate tabular data from block diagrams; and
4. Obtain system solution.

The system definitions identify which portions of the system are to be analyzed and what operating conditions are to be considered. The preparation of the simplified block diagram of the system is the foundation of the analysis. The block diagram presents a simple picture of how the

system operates and how it is interfaced. The block diagram should contain less than 10 blocks. For each block, all operating (faulted) conditions should be defined. From the block diagram a table of possible states is developed. The tabular data is presented much like a decision table. Once every operating state has been defined for each block, all combinations of operating states are generated in the table. Combinations which require the simultaneous occurrence of three or more fault conditions should be eliminated. Once the population of fault combinations has been determined, the resulting final system state is evaluated for each combination. The tabular data can then be numerically analyzed. The analysis can be performed for either a non-repairable system or a time-dependent system with repair. Markov chain theory can be used for non-repairable systems. Other methods can be used for repairable or standby systems.

Tabular system reliability analysis produces a systematic, traceable failure effects analysis. The data tables can be updated, thus enabling TASRA to be used as an iterative design tool. Tabular system reliability analysis can be used for mechanical, electrical, hydraulic, electromechanical, or pneumatic systems. Because of the simplicity of the block diagram and the systematic generation of state combinations, the analysis does not require excessive time or funds.

The simplicity of the analysis, however, also allows the analysis to be less than rigorous. This has drawbacks as well as advantages. Failure causes are not included in the analysis. Large, complex systems may

prove difficult to accommodate using the TASRA technique. A complex system would have to be partitioned for a thorough analysis.

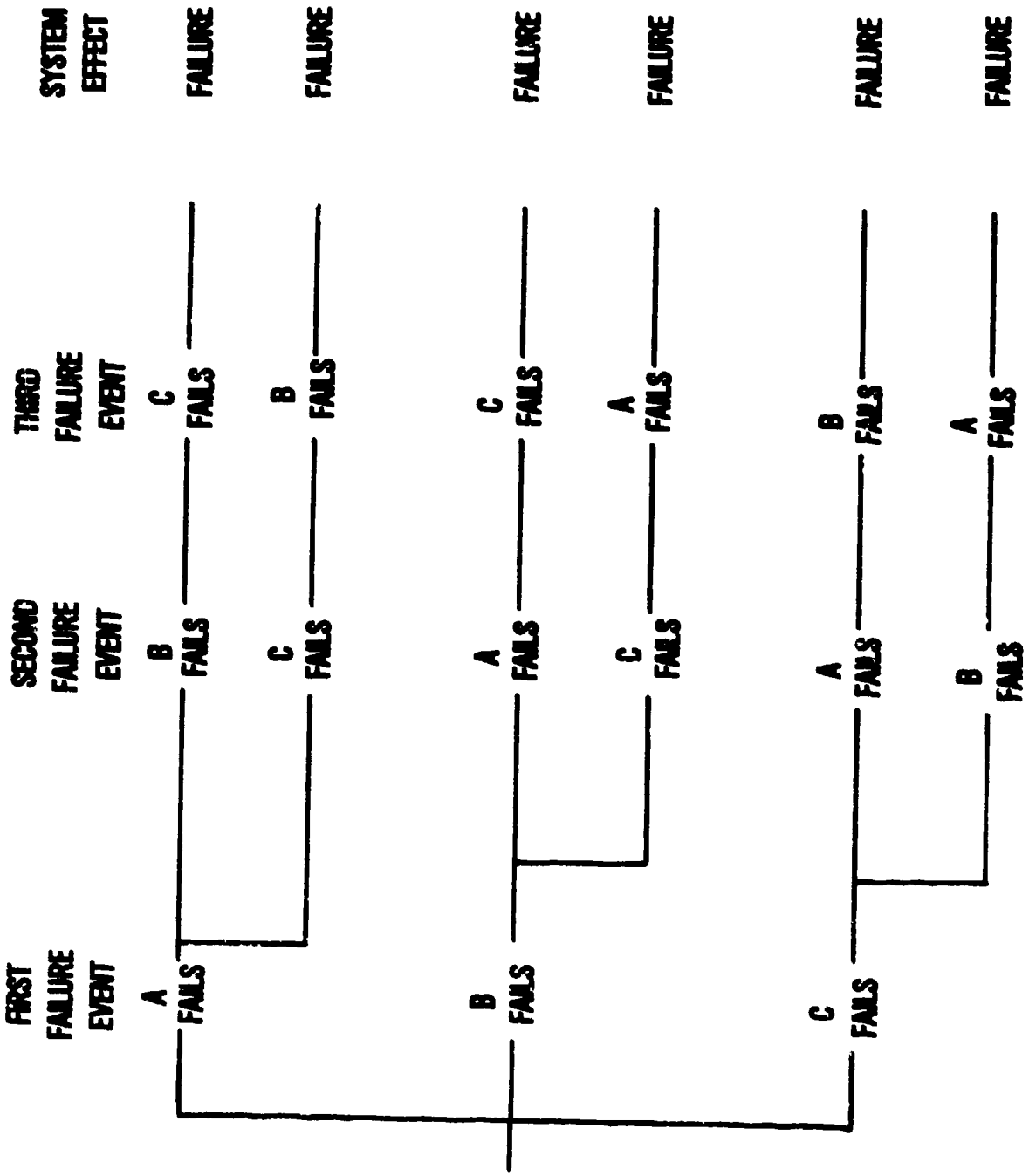
## 6.4 Event-Sequence Analysis

Event-Sequence Analysis was introduced in 1975 by Yellman (Reference 37). The analysis traces the effects of system failures as a function of the order in which they occur. The technique can be used at any design stage. Because of the time and fault dependent aspects of the analysis, the technique is especially applicable for failure effects analysis and logistics support of process systems and systems with distinct operational sequences.

The heart of event-sequence analysis is the event-sequence map. The event-sequence map chronicles all probable failure histories for system operation. A sample event-sequence map is shown in Figure 6.2. All events examined in event-sequence analysis are dependent (e.g., given that A and B have previously occurred, now that C has occurred the effect on the system is ...). The event-sequence map is developed by investigating all possibilities for a first failure event, then all possibilities for a second failure event, and so forth, until all branches end with a system failure.

Numerical evaluation of the event-sequence map is also included in the analysis. Because conditional probabilities are used, the probability of a specific failure path can be calculated by multiplying together the conditional probabilities of each event in the path. The total failure probability can then be obtained by summing the probabilities of

FIGURE 6.2 SAMPLE EVENT-SEQUENCE MAP



occurrence of each failure path. The numerical evaluation method can be simplified using Markov chain theory assumptions.

The analysis carries the history of failure events through the analysis and can include hardware failures and external influences. Event-sequence analysis is well suited for process systems or systems with distinct operational phases. A computer code, GO, has been developed to aid in event-sequence analysis. The analysis is updateable, and it can be used as an iterative design tool.

While event-sequence analysis is a very powerful analysis tool, it does have limitations. Without simplifying assumptions, numerical analysis using conditional probabilities becomes much more involved. If failures are truly dependent, a rigorous mathematical treatment is necessary. Multiple failure paths can be constructed from the same events occurring in a varying sequence. For this reason, difficult to obtain conditional probabilities must be used in the analysis. As in other techniques which emphasize system states, the effects of individual failures can be difficult to trace. And, while the structure of the event-sequence map is logical, errors of omission and oversight can occur. Additional information, such as failure cause and fault detection methods, is not incorporated into the analysis.

## 6.5 Testability Analysis

Testability analysis is actually a separate systems analysis from failure effects analysis. But, because of the similiarity of data required by each analysis, it was suggested by Smith in 1979 (Reference 32)

that an automated testability technique replace FMEA. The automated testability analysis suggested was a company proprietary program. The automation of the analysis, however, does have potential to be used as a technique for failure effects analysis, while also providing information for testability design.

An automated testability analysis used for failure effects analysis would require detailed design information, such as components and their layout. The computer results could include a summation of all failures considered and their effects. The completeness of fault detection and isolation could also be evaluated. The entire analysis could be performed by computer for systems with a logical flow pattern, such as electrical, digital, mechanical, and hydraulic systems.

The use of the automated testability procedure would reduce the effort and tedium involved in a tabular FMEA. No failure rate or failure mode data is required for the analysis. The analysis can also be used to verify BIT designs and to evaluate other fault detection/isolation methods. The automation of the technique helps to insure thoroughness in the examination of the system.

Automated testability analysis does, however, have its limitations in application to failure effects analysis. Only single, traditional failure modes are considered. Multiple failures, unconventional failures, and external influences are not evaluated as part of the automated procedure. Criticality is not evaluated in the analysis. Because detailed design information is required, the analysis must be done during

later design stages when design changes suggested by the analysis may not be cost effective.

While automated testability may not replace failure effects analysis, the two analyses should not be isolated. Large amounts of time and effort can be saved by interfacing the two analyses.

## 6.6 LAM Technique

The LAM technique uses the physical properties of a system to evaluate the effects of failures. The analysis was introduced in 1980 by Reina and Squellati (Reference 27). A "TOP" event fault tree approach and systems state analysis are combined in the LAM technique.

The procedure for using the LAM technique for failure effects involves the following five steps.

1. Identify the physical variables which characterize the system functionally and topographically.
2. Functionally analyze the components both under normal operating conditions and failure conditions.
3. Construct parametric models of system operation and failure.
4. Identify the TOP condition.
5. Generate TOP sets, and calculate TOP event probability of occurrence.

The identification of the physical variables establishes the logical connections between the components in the system. Both failed and operational system states are examined so that parametric equations can be developed which model the response of system physical characteristics to

system hardware failures. The effect of multiple, concurrent failures should also be examined as part of the analysis. Every combination of system failures can be modelled using the parametric equations. The identification of the TOP event focuses the analysis. The TOP event sets are then generated in a controlled manner using the system equations. The probability of occurrence of the TOP event is a direct calculation.

The LAM technique attempts to eliminate some of the inadequacies of fault tree analysis and tabular FMEA. The analysis characterizes the system as a set of physical parameters which are effected by component failures. The LAM technique can be used to provide an integrated analysis of both hardware and physical conditions in a system.

The analysis, however, would require both a physical model of the system and a failure effects analysis to be obtained. The analysis is limited to systems which can be readily modelled by physical variables. Microprocessor and software systems may not be readily analyzed by the LAM technique. The analysis may be costly and time-consuming. Because of the technique's recent introduction and limited use, problems associated with the technique have not been closely evaluated.

6.7 Approachability Analysis

Approachability analysis was developed by Hitachi for failure effects analysis of consumer products. The technique was introduced at the 1981 Reliability and Maintainability Symposium by Tsuji, et al (Reference 35). The technique analyzes failures caused by approach. Failures

caused by approach are defined as failures caused by the improper relationship of parts, failures caused by the introduction of foreign materials into the system, and failures caused by external stresses. Approachability analysis is usually performed in later design stages when the topography of the system has been well defined.

Like tabular and matrix FMEA, a "worksheet" format is used for approachability analysis. The worksheet uses a matrix format for analyzing combinations of approaching parts, objective parts, and external stresses. All the objective parts, parts which will fail as the result of an approach, are placed along one axis of the matrix. The approaching parts and external stresses are placed along the other axis. Much as in matrix FMEA, those combinations of objective part and approaching part or external stress which result in failure are marked.

The procedure for approachability analysis consists of seven steps.

1. Define the system, parts, and external stresses.
2. Determine operating condition of product.
3. Analyze the failure potential between the objective parts and the approaching parts.
4. Analyze the failure potential between the objective parts and external stresses.
5. Analyze all combinations of objective parts, approaching parts, and external stresses.
6. Construct a failure expression, and evaluate the probability of each event occurring.

7. Assess failure probability, and determine if corrective action is necessary.

The procedure provides a logical method of identifying parts in the system which are subject to failures caused by approach. Corrective actions can be proposed as a result of the analysis.

Approachability analysis is a specialized failure effects analysis. The technique is only meant to analyze failures caused by approach. Approachability analysis examines types of failures which are often overlooked but which can readily occur during consumer use or operation in other severe environments.

6.8 Failure Combination Method

The failure combination method will be the last technique discussed in this section. The failure combination method was presented at the 1981 Reliability and Maintainability Symposium by Hedin, et al (Reference 12). The technique was developed by the French Societe Nationale des Industries Aeronatiques et Spatiaks and the Ministere de L'Air. The technique is also used by Electricite' de France. The failure combination method evaluates the effect of single, multiple, and externally influenced failures on a system using an inductive approach. The analysis can be performed at any stage of design and at any indenture level.

The procedure used in the failure combination method consists of four steps.

1. System breakdown by FMEA.
2. Definition of "gathered failures".

3. Identification of "external failures".

4. Identification of "overall failures".

"Gathered failures" are defined as failure modes, taken alone or in combination with others, which produce the same effect on the system. "External failures" are defined as failures which occur in systems outside of the analysis which affect the system under study. "Overall failures" are defined by combining external failures and gathered failures and applying them to specific system failures. (Gathered failures should not be recombined into a larger gathered failure in the identification of overall failures.) The overall failures group single, multiple, and external failures by the effect they produce. The grouping scheme also allows common mode failure potential to be more easily recognized.

The failure combination method actually supplements a tabular or matrix FMEA. The inductive approach of the analysis helps to produce a more thorough analysis. Grouping the failures by their effects makes the analysis easier to understand and present. Unfortunately, the analysis incorporates all the time and cost associated with FMEA and requires an even larger effort.

# 7.0 HARDWARE/SOFTWARE FAILURE EFFECTS ANALYSIS TECHNIQUES

## 7.1 Introduction to Hardware/Software Techniques

Hardware/Software failure effects analysis techniques are needed, and some have been developed, to meet the growing need to analyze integrated designs. The complexity of hardware/software systems requires that analysis techniques provide a thorough analysis while remaining flexible enough to accommodate state-of-the-art designs. The analysis techniques, however, are all specialized for specific design areas. The three analysis areas which will be discussed in this section are:

- Software
- Hardware/Software Systems, and
- Microcomputers.

## 7.2 Software Techniques

It was previously assumed that software, or a computer program, had a reliability of 1.0 or 0.0. Either an error(s) existed in the program, or the program was error free. Because software has become so intricate and complex, errors can exist which may not be found until an unexpected situation arises or until after many hours of trouble-free operation. Techniques to find software errors have been developed. Two software analysis techniques will be discussed: software FMEA and software sneak analysis.

### 7.2.1 Software FMEA

Software FMEA was suggested by Reifer in 1978 (Reference 26). Software FMEA is very similar to tabular FMEA used for hardware systems.

Software FMEA can be performed at any design stage, but it provides a more cost-effective analysis during early design stages. Software FMEA uses a functional approach examining software performance requirements to determine how the software can fail.

The procedure used for a software FMEA is partitioned into two phases. In the first phase the software performance requirements are defined and possible failure modes are analyzed. The first phase consists of five steps.

1. Determine mission critical software performance requirements.
2. Analyze mission critical failure modes for interrelationships and time dependencies. Analytical models and simulation can be used for the analysis.
3. Mission critical requirements are ranked according to the probability of sustaining a critical failure.
4. Perform detailed FMEA. Identified failure modes are introduced into the software model and the effects evaluated. Random failures are also introduced and their effects evaluated.
5. Update analysis to include new or changed performance requirements or failure data.

The second phase of the analysis involves a study to determine the feasibility of eliminating critical failure modes. Proof of correctness theory can be used to verify the performance of small, highly critical

parts of the software. Self-checking and fault tolerant design techniques can be applied to help to eliminate or guard against critical failure modes.

Software FMEA can be applied during early design stages to allow for cost effective design changes. Software FMEA can identify areas where redundancy or fault tolerance are needed. The analysis can also help to verify that software performance requirements have been met and implemented properly. Software FMEA, however, can be very time-consuming, especially if proof of correctness theory is used in conjunction with the analysis. Programming errors and misimplementations may be overlooked in the analysis.

### 7.2.2 Software Sneak Analysis

The formal technique for software sneak analysis was developed in 1975. Software sneak analysis has met with wide acceptance in its use. The technique has proven to be very successful in finding program "bugs". As with sneak circuit analysis, software sneak analysis is strictly intended to locate design errors in the software. Software sneak analysis identifies four sneak conditions:

1. Sneak output;
2. Sneak inhibit;
3. Sneak timing; and
4. Sneak message.

The procedure for a software sneak analysis is basically the same as in a sneak circuit analysis. A simplified analysis procedure

consists of four steps. (A more detailed description of SCA can be found in Section 5.0.)

1. Network tree production
2. Topological pattern recognition
3. Clue application
4. Summary of results and sneak reports.

Electrical symbology is used to represent the logic flow of the program. The electrical symbology prevents the analysis from becoming computing language dependent. The same basic procedure used for sneak circuit analysis is also used in software sneak analysis. Instead of the five standard topological patterns in sneak circuit analysis, six standard topological patterns can be found in software. The same basic types of clues used in SCA are used for software sneak analysis. The results, however, are much different. Software sneak analysis can identify such conditions as branch bypasses and infinite loops.

Software sneak analysis is a very thorough technique for discovering programming errors. The analysis process is straightforward. The analysis will not, however, find areas where programming has misinterpreted software performance requirements. For large, complex programs the analysis can be time-consuming and costly.

## 7.3 Integrated Hardware/Software Analysis Techniques

Computer controlled systems and other systems which integrate hardware and software also require special analysis techniques. The hardware portions of the system can be overstressed by the software. The

software can cause the system to operate improperly as a result of faulty hardware indications. Previously only hardware failures were examined as part of a failure effects analysis. New techniques, however, allow both hardware and software failures to be analyzed. Two analysis techniques will be discussed: integrated critical path analysis and hardware/software interface analysis.

### 7.3.1 Integrated Critical Path Analysis

Integrated Critical Path Analysis (ICPA) was introduced in 1980 by Tuma (Reference 36). Integrated critical path analysis examines hardware/software interrelationships. The technique combines FMEA, fault tree analysis, and sneak circuit analysis to provide a detailed investigation of the system.

The procedure for performing an integrated critical path analysis consists of the following seven steps.

1. Define scope of the analysis and any background material (such as previous FMEAs).
2. Construct integrated functional network trees.
3. Update and integrate trees to show hardware/software interfaces.
4. Predict reliability of critical paths.
5. Analyze the effect of procedures (e.g., test and maintenance).
6. Include any failure analysis results.
7. Summarize and document the analysis.

The ICPA builds on any previously performed failure effects analyses on either the hardware or software. System definitions, operating procedures, and critical functions need to be ascertained before the analysis begins. ICPA is usually instituted during the later design stages. The network trees produced are very similar to those found in sneak circuit analysis. The trees, however, include both hardware and software functions. The network trees can be used to update fault trees for hardware/software interrelations. The fault trees should indicate areas where software can overstress hardware, where software performs the same function as hardware, where software can bypass or defeat hardware, where a manual operation can defeat both hardware and software, and where software does or does not out-perform hardware. The updated fault trees can then be numerically evaluated. The analysis should also evaluate the effect of procedures on the system and should evaluate only available failure analysis results. The analysis should be documented in a report containing all materials used, results, and recommendations.

In summary, the evaluation of a system using integrated critical path analysis provides an integrated approach to failure effects analysis. Aspects of sneak circuit analysis, fault tree analysis, and FMEA are used to develop a comprehensive system analysis covering hardware/software interfaces, external influences, operational procedures, and failure data. Numerical evaluation of fault trees indicating human/hardware/software interfaces may be difficult to accomplish. A thorough analysis would also require a large expenditure of time and

funds.

### 7.3.2 Hardware/Software Interface Analysis

Hardware/Software interface analysis was developed for use in the space shuttle project (Reference 3). The analysis considers software requirements as a function of hardware failures. Because software cannot respond to unanticipated conditions, it is important to recognize every probable failure condition and to determine how the software should respond. Hardware/software interface analysis is designed to insure that:

- Software anticipates hardware failures and provides for continued operation;
- Software uses the full capability of hardware; and
- Software does not overstress hardware.

Hardware/software interface analysis is performed in three phases. In the first phase, every line item in a system tabular FMEA is checked for software interfaces. Eight questions are asked of every line item. (Reference 3)

1. Does software detect this failure mode? And if not, does the hardware provide information that software could use to detect the failure mode?
2. Are the answers to Question 1 consistent with the previous FMEA?
3. Does software take action to negate the effects of the failure? And if not, is it possible for the software to

funds.

### 7.3.2 Hardware/Software Interface Analysis

Hardware/Software interface analysis was developed for use in the space shuttle project (Reference 3). The analysis considers software requirements as a function of hardware failures. Because software cannot respond to unanticipated conditions, it is important to recognize every probable failure condition and to determine how the software should respond. Hardware/software interface analysis is designed to insure that:

- Software anticipates hardware failures and provides for continued operation;
- Software uses the full capability of hardware; and
- Software does not overstress hardware.

Hardware/software interface analysis is performed in three phases. In the first phase, every line item in a system tabular FMEA is checked for software interfaces. Eight questions are asked of every line item. (Reference 3)

1. Does software detect this failure mode? And if not, does the hardware provide information that software could use to detect the failure mode?
2. Are the answers to Question 1 consistent with the previous FMEA?
3. Does software take action to negate the effects of the failure? And if not, is it possible for the software to

compensate for the failure?

4. Can hardware overstress the hardware or induce another failure as a result of this failure?
5. Can software logic in combination with this failure adversely affect other system functions?
6. How many of these hardware failures can the system tolerate?
7. If corrective actions are required, are indications provided to signal the need for action?
8. If the answer to either Question 1 or 3 is yes, can a backup system take over?

In the second phase of analysis all the results of the hardware/software interface analysis are examined. Decisions are made whether to correct or keep unacceptable failure modes. In the final phase the decisions made in phase two are explained and justified.

Hardware/software interface analysis is a very valuable extension of a tabular FMEA. The analysis examines how the software affects hardware failures and how software can improve system performance. The analysis can be performed in early design stages to help define software requirements. In later stages the analysis can help to verify software design. BIT verification could also be an extension of hardware/software interface analysis. The analysis, however, cannot stand alone. A tabular FMEA is required to support a hardware/software interface analysis.

## 7.4 Microcomputer FMEA

Microcomputers are relatively recent innovations. Because of the microcomputer's capability and versatility, it has been widely applied to new designs. The microcomputer, however, is very complex. Its numerous inputs and outputs and memory states make analysis very difficult.

Because microcomputer operation is tailored specifically to the system to which it is dedicated it is often difficult to determine overall effects resulting from a microcomputer failure. The microcomputer can have a vast number of potential memory configurations. Different operating modes and memory patterns affect the way the microcomputer reacts to a failure. Instead of trying to analyze all possible microcomputer configurations, the microcomputer FMEA technique suggested by Kenyon and Newell (Reference 18) examines the effect of faults occurring on microcomputer input and output pins during operation.

The microcomputer FMEA technique is performed on an operating system. A fault generator is connected between the hardware system and the microcomputer. The fault generator simulates a variety of faults for every input and ouput pin. The fault generator can simulate open lines, lines stuck at ground, lines stuck at supply bus voltage, and lines stuck at an intermediate voltage. Wrong frequencies, wrong pulse widths, wrong duty cycles, and signal amplitude variations can be simulated for analog systems. The effect of each simulated fault can be entered on a FMEA worksheet.

The value of the FMEA technique is that it is performed using a powered, executing microcomputer. The observed effects are real, and various operating environments can be readily analyzed. Using the microcomputer FMEA technique, the failure effects analysis can be performed in a timely and cost-effective manner. The use of microcomputer FMEA, however, is limited to late design states, when system changes are not very cost-effective. Unconventional failures are not simulated by the fault generator, and non-apparent failure effects may be overlooked during the analysis.

# 8.0 OVERVIEW OF FAILURE EFFECTS ANALYSIS TECHNIQUES

## 8.1 Summary of Techniques Discussed

Each of the techniques discussed can be used for a failure effects analysis. Each technique examines how a system responds to failures. Some analyses itemize causes and evaluate the effects; other analyses define an effect and determine the possible causes. Ultimately, each failure effects analysis attempts to evaluate system strengths and weaknesses. The analysis results are used to improve system performance by indicating the need for corrective action or design changes. An effective failure effects analysis presents a thorough analysis in a timely and cost effective manner.

The techniques discussed in this report fall into four broad categories:

1. FMEA and supplemental techniques;
2. "Tree" techniques;
3. Combinations of techniques; and
4. Alternate techniques.

Table 8.1 shows how all the techniques discussed have been categorized. The FMEA and supplemental techniques all examine failure effects resulting from a specific failure mode. The supplemental techniques either enlarge an existing FMEA or provide a more specialized analysis. The tree techniques, while their approaches differ, all use a "Tree" diagram to direct the analysis. The major tree technique is fault tree analysis, but alternate tree techniques include sneak analysis and time-dependent

TABLE 8.1 SUMMARY OF FAILURE EFFECTS ANALYSIS TECHNIQUES

| FMEA | SUPPLEMENTAL TO FMEA |
|---|---|
| • TABULAR FMEA (1950s) | • CRITICALITY ANALYSIS (1950s) |
| • MATRIX FMEA (1977) | • HAZARD ANALYSIS (1950s) |
| • SOFTWARE FMEA (1979) | • FAILURE COMBINATION METHOD (1981) |
| • MICROCOMPUTER FMEA (1982) | • HARDWARE/SOFTWARE INTERFACE ANALYSIS (1981) |
| **TREE TECHNIQUES** | **ALTERNATE TREE TECHNIQUES** |
| • FAULT TREE ANALYSIS (1965) | • SNEAK CIRCUIT ANALYSIS (1967) |
| | • SYSTEM STATE PHASE MODELLING (1969) |
| | • EVENT-SEQUENCE ANALYSIS (1975) |
| | • SOFTWARE SNEAK ANALYSIS (1975) |
| **COMBINATIONS OF TECHNIQUES** | |
| • TABULAR SYSTEM RELIABILITY ANALYSIS (1971) | |
| • INTEGRATED CRITICAL PATH ANALYSIS (1980) | |
| **ALTERNATE TECHNIQUES** | |
| • "TESTABILITY ANALYSIS" (1979) | |
| • LAM TECHNIQUE (1980) | |
| • APPROACHABILITY ANALYSIS (1981) | |

TABLE 8.2 SUMMARY OF FAILURE EFFECTS ANALYSIS TECHNIQUES CHARACTERISTICS

| | INDUCTIVE | DEDUCTIVE | SPECIALIZED APPLICATIONS | TIME-DEPENDENCY | USES ADVANCED MATHEMATICS | SINGLE FAILURES EXAMINED | MULTIPLE FAILURES EXAMINED | EXTERNAL INFLUENCES EXAMINED | INITIATED AT ANY DESIGN STAGE | EARLY DESIGN STAGES ONLY | LATE DESIGN STAGES ONLY | LOGISTICS APPLICATIONS | TESTABILITY APPLICATIONS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TABULAR FMEA | ✓ | | | | | ✓ | | | ✓ | | | | ✓ |
| CRITICALITY ANALYSIS | | | | | | ✓ | | | ✓ | | | ✓ | |
| HAZARD ANALYSIS | ✓ | | | | | ✓ | | | ✓ | | | ✓ | |
| FAULT TREE ANALYSIS | | ✓ | | | ✓ | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ |
| SNEAK CIRCUIT ANALYSIS | | | ✓ | | | | | | | | ✓ | | |
| SYSTEM STATE PHASE MODELLING | ✓ | | | ✓ | | ✓ | ✓ | ✓ | ✓ | | | ✓ | |
| TABULAR SYSTEM RELIABILITY ANALYSIS | ✓ | | | | ✓ | ✓ | ✓ | | ✓ | | | ✓ | |
| EVENT-SEQUENCE ANALYSIS | ✓ | | | ✓ | ✓ | ✓ | ✓ | | ✓ | | | ✓ | |
| SOFTWARE SNEAK ANALYSIS | | | ✓ | | | | | | | | ✓ | | |
| MATRIX FMEA | ✓ | | ✓ | | | ✓ | | | ✓ | | | ✓ | ✓ |
| TESTABILITY ANALYSIS | ✓ | | ✓ | | | ✓ | | | | | ✓ | | ✓ |
| SOFTWARE FMEA | ✓ | | ✓ | | | ✓ | | | | ✓ | | | |
| LAM TECHNIQUE | | | ✓ | | ✓ | ✓ | ✓ | | ✓ | | | ✓ | |
| INTEGRATED CRITICAL PATH ANALYSIS | | | | | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ | ✓ |
| APPROACHABILITY ANALYSIS | ✓ | | ✓ | | | ✓ | ✓ | ✓ | | | ✓ | | |
| FAILURE COMBINATION METHOD | ✓ | | | | | ✓ | ✓ | ✓ | ✓ | | | ✓ | |
| HARDWARE/SOFTWARE INTERFACE ANALYSIS | ✓ | | ✓ | | | ✓ | | | | | ✓ | | ✓ |
| MICROCOMPUTER FMEA | ✓ | | ✓ | | | ✓ | | | | | ✓ | | ✓ |

analysis. The combinations of techniques attempt to blend different aspects of several failure effects analysis techniques to provide a more comprehensive study. The alternate techniques present totally different approaches to failure effects analysis.

A summary of characteristics and applications of each of the techniques discussed is presented in Table 8.2. The table includes information for each technique on the types of failures examined, the approach used, the timing of the analysis in the design cycle, and other anlayses which can benefit from the technique.

## 8.2 Current Status of Failure Effects Analysis

Failure effects analysis is often viewed as a necessary evil. The intent of failure effects analysis is to allow a system to be examined early in its development when undesirable failures can be readily identified and corrected. Unfortunately, the detail involved in performing a failure effects analysis often causes the analysis to be overly time-consuming and costly. Because failure effects analyses have tended to be ineffective, their use has often been reduced to a formality to fulfill contractual or quality assurance obligations.

Currently, four techniques are most widely used for failure effects analysis: tabular FMEA/FMECA; Fault Tree Analysis; Matrix FMEA; and Sneak Circuit Analysis. Specialized techniques for software and hardware/software integrated systems are becoming more accepted as the need for the analysis increases. While other techniques discussed have similar potential for successful application, the techniques will probably

not meet with extensive use until the value of failure effects analysis is fully realized.

The use of failure effects analysis is limited by problems with its use. The most common problems encountered in failure effects analysis include the following:

1. The analysis is time-consuming and costly.
2. The analysis results and recommendations are often obtained too late in design to be easily incorporated.
3. Accurate failure data are difficult to obtain.
4. The level of indenture necessary for a thorough, economical, and effective analysis is difficult to accurately define.
5. Existing standards do not offer much direction for a failure effects analysis.
6. Every technique used for failure effects analysis has shortcomings and is subject to inaccuracies.

Work is currently being done to help overcome some of the problems associated with failure effects analysis. Automation of both the bookkeeping and analytical aspects of failure effects analysis intends to speed the analysis and reduce analysis costs. The computer can also be used to check for consistency in the analysis. Accurate failure data are necessary for a quantitative failure effects analysis. While efforts have been made to improve the existing data base, more work is needed to increase the quality of the data and to reduce the misuse of data. Existing standards are being upgraded, but definite requirements for

prescribing, conducting, and reviewing a failure effects analysis continue to be lacking. Until greater guidance is available, the selection of a technique and proper level of indenture for the analysis will be difficult.

The adequacy of each technique for failure effects analysis has been previously discussed. The adequacy of the analysis is influenced by the skill level of the analyst. Inaccuracies and errors of oversight can occur if the analyst is not well acquainted with the technique or the system or if the analyst is pressured by time constraints.

## 8.3 The Future of Failure Effects Analysis

The future of failure effects analysis techniques belongs to the efficient. While each technique discussed can be used in a failure effects analysis, no one technique can provide all of the information desired. New techniques which attempt to overcome some of the problems with other techniques continue to be developed. In order to use the full potential of failure effects analysis the techniques need to be made efficient in the following areas:

- time and cost;
- thoroughness and accuracy; and
- applicability to related analysis areas, such as logistics, testability, and maintainability.

Guidelines should also be made available to assist in the prescription, conduct and review of a failure effects analysis.

It is important that the potential of failure effects analysis to

influence design and to support other analyses be recognized. Unless the results of a failure effects analysis are used, no reduction in time or cost or improvement in accuracy will influence the usefulness of the analysis.

## 9.0 BIBLIOGRAPHY

1. Barbour, G. L., "Failure Modes and Effects Analysis by Matrix Methods", Proceedings 1977 Annual Reliability and Maintainability Symposium, pp 114-119.

2. Bazovsky, I. Sr., "Fault Trees, Block Diagrams, and Markov Graphs", Proceedings 1977 Annual Reliability and Maintainability Symposium, pp 134-141.

3. Bunce, W. L., "Hardware and Software: An Analytical Approach", Proceedings 1980 Annual Reliability and Maintainability Symposium, pp 209-213.

4. Buratti, D. L., and Goody, S. G., "Sneak Analysis Application Guidelines", RADC-TR-82-179, June 1982.

5. Clardy, R. C., "Sneak Circuit Analysis", Reliability and Maintainability of Electronic Systems, Arsenault and Roberts, ed., Computer Science Press, 1980, pp 223-241.

6. Crown, P. L., "Design Effective Failure Mode and Effect Analysis", Proceedings 1969 Annual Reliability and Maintainability Symposium, p. 514.

7. Eagle, K. H., "Fault Tree and Reliability Analysis Comparison", Proceedings 1969 Annual Reliability and Maintainability Symposium, p.12.

8. Fussell, J. B., Powers, G. J.; and Bennetts, R.G., "Fault Trees - A State of the Art Discussion', IEEE Transactions on Reliability, V R-23, nl, April 1979, p 51.

9. Greene, K. and Cinibulk, W., "Quantitative Safety Analysis", Proceedings 1972 Annual Reliability and Maintainability Symposium, pp 218-221.

10. Greene, K. and Cunningham, T. J., "Failure Mode, Effects, and Criticality Analysis", Proceedings 1968 Annual Reliability and Maintainability Symposium, p. 374.

11. Haasl, D. F. et al., "Fault Tree Handbook", NUREG - 0492, January 1981.

12. Hedin, F.; Le Coguiec, A.; LeFloch, C.; Llory, M., and Villemeur, A., "The Failure Combination Method", Proceedings 1981 Annual Reliability and Maintainability Symposium, pp 163-172.

13. Herrin, S. A., "Maintainability Applications Using the Matrix FMEA Technique", IEEE Transactions on Reliability, V R-30, n3, August 1981, pp 212-214.

14. Herrin, S. A., "System Inteface FMEA by Matrix Method", Proceedings 1982 Annual Reliability and Maintainability Symposium, pp 111-116.

15. Human, C. L., "The Graphical FMEA", Proceedings 1975 Annual Reliability and Maintainability Symposium, pp 298-303.

16. Jacobs, R. M. and Michalsky, J., "Reliability Techniques to Keep You Out of Court", Proceedings 1973 Annual Reliability and Maintainability Symposium, p. 355.

17. Jordan, W. E., "Failure Modes, Effects, and Criticality Analyses", Proceedings 1972 Annual Reliability and Maintainability Symposium, p. 30.

18. Kenyon, R. L. and Newell, R. J., "FMEA Techniques for Microcomputer Assemblies", Proceedings 1982 Annual Reliability and Maintainability Symposium, pp 117-119.

19. Krohn, C. A.; Nelson, A. C.; and Thompson, W. S., "Methods of Design Stage Reliability Analysis", Proceedings 1967 Annual Reliability and Maintainability Symposium, p. 803.

20. Landers, R. R., "A Failure Mode and Effect Analysis Program to Reduce Mechanical Failures", Mechanical Failure Prevention Group Symposium on Engineering Design, National Bureau of Standards, November 3-6, 1978.

21. Lapp, S. A. and Powers, G. J., "Computer-Aided Synthesis of Fault Trees", IEEE Transactions on Reliability, V R-26, n1, April 1977, p. 2.

22. Legg, J. M., "Computerized Approach for Matix-Form FMEA", IEEE Transactions on Reliability, V R-27, n4, October 1978, p. 254.

23. Neogy, R., "Fault Trees in Ocean Systems", Proceedings 1975 Annual Reliability and Maintainability Symposium, pp 280-285.

24. Onodera, K.; Miki, M.; and Nukada, K., "Reliability Assessment for Heavy Machinery by HI-FMECA Method", Proceedings 1977 Annual Reliability and Maintainability Symposium, pp 346-352.

25. Power, H. R. and Bailey, H. J., "Updating of Reliability Criteria Documents", Proceedings 1970 Annual Reliability and Maintainability Symposium, p. 29.

26. Raifer, D. J., "Software Failure Modes and Effects Analysis", IEEE Transactions on Reliability, V R-27, n3, August 1979, p. 247.

27. Reina, G. and Squellati, G., "L.A.M. Technique: Systematic Generation of Logical Structures in Systems Reliability Studies:, Synthesis and Analysis Methods for Safety and Reliability Studies, Plenum Press, 1980, pp 129-181.

28. Salem, S. L. and Apostolakis, G. E., "The CAT Methodology for Fault Tree Construction", Synthesis and Analysis Methods for Safety and Reliability Studies, Plenum Press, 1980, pp 109-120.

29. Schroder, R. J., "Fault Trees for Reliability Analysis", Proceedings 1970 Annual Reliability and Maintainability Symposium, p. 198.

30. Sevcik, F., "Current and Future Concepts in FMEA", Proceedings 1981 Annual Reliability and Maintainability Symposium, pp 414-421.

31. Shooman, M. L., "Techniques of System Reliability Prediction (Using the Automobile as an Example)", Proceedings 1971 Annual Reliability and Maintainability Symposium, pp 167-185.

32. Smith, G. II, "Testability Analysis: Predict It More Closely", Proceedings 1979 Annual Reliability and Maintainability Symposium, pp 187-189.

33. Thatcher, R. K., Easterday, J. L.; and Taylor, F. R., "An Integrated Predictor of System Reliability", Proceedings 1971 Annual Reliability and Maintainability Symposium, pp 254-260.

34. Tiger, B., "Evaluating System States in Their Mission Phases", Proceedings 1969 Annual Reliability and Maintainability Symposium, p. 18.

35. Tsuji, Y.: Nukada, K., Hanma, M.; and Sasaki, R., "Approachability Analysis", Proceedings 1981 Annual Reliability and Maintainability Symposium, pp 210-216.

36. Tuma, F., "Software/Hardware Integrated Critical Path Analysis (ICPA)", Proceedings 1980 Annual Reliability and Maintainability Symposium, pp 384-387.

37. Yellman, T. W., "Event-Sequence Analysis", Proceedings 1975 Annual Reliability and Maintainability Symposium, pp 286-291.

38. Yellman, T. W., "Event-Sequence Analysis vs. The Fault Tree", Proceedings 1981 Annual Reliability and Maintainability Symposium, pp 446-451.

39. ARP-926, "Fault/Failure Analysis", Society of Automotive Engineers, May 1979.

40. MIL-STD-1629A, "Procedures for Performing a Failure Mode, Effects and Criticality Analysis",

41. SAMSO-STD-77-2, "Failure Modes and Effects Analysis for Satellite, Launch Vehicle, and Reentry Systems", November 22, 1977.