DRAFT

DRAFT REPORT

on

REVIEW OF SYSTEMS INTERACTION METHODOLOGIES

to

Systems Interaction Branch
Office of Nuclear Reactor Regulation
U.S. Nuclear Regulatory Commission

September 30, 1980

BATTELLE
Columbus Laboratories
Pacific Northwest Laboratories

DRAFT

TABLE OF CONTENTS

INTRODUCTION

The Systems Interaction Branch of the NRC's Office of Nuclear Reactor Regulation has among its responsibilities the consideration of the potential effects of systems interactions in the review of reactor license applications. This is a new thrust for the NRC which derives from the several analyses of the TMI incident and the development of the NRC Action Plan (NUREG-0660). As a means of fulfilling this responsibility the development of an independent methodology for identifying and evaluating systems interactions is being considered. Such a methodology would have two broad applications:

a) it would define the information requirements, procedures, and criteria that could be used by the applicant in the development and review of the plant design, and
b) it would provide the framework for the NRC review of the plant design for systems interaction considerations.

At the present time there are no regulatory guidelines and requirements for systems interaction evaluations for nuclear power plants. Further, it is not clear that a concensus definition of systems interaction is available at this time, much less an agreement on applicable methodologies. It is the objective of the initial effort described here to review applicable methodologies that may have potential for relatively near-term use in systems interaction evaluations. The work described here was undertaken by Battelle's Columbus Laboratories and Pacific Northwest Laboratories. Parallel efforts are being performed by two other organizations.

The broad objective of this project is to develop methods that hold the best potential for further development and near-term use by industry and NRC on systems interaction evaluations for future as well as operating plants. More specifically, the objectives of the work described here include:

a) development of a definition of systems interaction and corresponding safety failure criteria,
b) review and assessment of current systematic methods that have been used, or considered feasible for use, on any complex system comparable to a light water reactor plant,

c) provision of an inventory of a range of systems interaction scenarios with emphasis on actual operating experience to:
   (1) better focus on the definition of systems interaction, and
   (2) serve as a basis for evaluating the ability of the various methodologies to predict these examples, and

d) recommendation of a methodology or alternatives that have the best potential for further development and near-term use by industry and the NRC on systems interaction evaluations.

The effort undertaken under this task should provide the basis for follow-on studies; the latter may include application of the recommended methodologies to selected cases as well as further methodology development.

DEFINITION OF SYSTEMS INTERACTION

Before attempting to derive a definition of systems interaction it is useful to consider a number of concepts. For the present purposes, a "system" is a collection of components which perform some function; generally the function defines the system. One component is not a system. Several systems can support a single function. Clearly, systems are designed to interact with each other in various ways. Most of these interactions are intentional and well recognized. The concern is with a limited set of potential interactions. In the present context an "interaction" of concern results when the conditions in one system affect (degrade) the ability of another system to perform its function. It should be recognized that such "interactions" need not necessarily imply or require failure in the normal sense of the affected system, e.g., a system may be misled by faulty instrumentation or actuation signals. Since the operator, used here in a very broad sense, can have an impact on the availability of any and all safety as well as supporting systems in the plant, it is imperative that his role be properly recognized. The operator may be considered as a component or a subsystem that can impact on the other systems in the plant.

As was noted earlier the definition of systems interaction includes consideration of some safety failure criterion. The failure criterion selected must recognize potential as well as actual hazard or risk that may result from the systems interaction. The Crystal River incident, for example, did not release any radioactivity to the environment, though it clearly represents a situation of interest from the systems interaction viewpoint. The inclusion of potential hazard or risk in systems interaction consideration, while deemed necessary, has the potential of substantially broadening the scope of this effort. In order to focus the systems interaction considerations it will be useful to consider the concept of safety functions. The use of this concept is not unique to this study. The present discussion draws heavily on the work of Reference ( ). This concept provides a certain hierarchy of plant protection and a systematic approach to mitigating the consequences of an

upset event. A safety function may be defined as a group of actions that maintain the defense-in-depth concept and minimize the potential of radioactivity release to the environment. Ten basic safety functions can be defined which are required to maintain the desired level of protection to the public. These basic safety functions and their specific purposes are given below.

Safety Function	Purpose
Reactor Control	Maintain desired power level and shutdown reactor when required.
Reactor Coolant System Inventory Control	Maintain a suitable coolant medium around the core.
Reactor Coolant System Pressure Control	Maintain the coolant in the proper state.
Core Heat Removal	Transfer heat from the core to the coolant.
Reactor Coolant System Heat Removal	Remove heat from the primary system.
Containment Isolation	Maintain containment integrity to prevent radiation releases.
Containment Temperature and Pressure Control	Avoid potential damage to containment and vital equipment.
Combustible Gas Control	Remove and/or redistribute hydrogen to avoid potentially damaging reactions.
Maintenance of Vital Auxiliaries	Maintain operability of systems needed to support safety systems.
Indirect Radioactivity Release Control	Contain miscellaneous stored radioactivity to protect the public and the environment.

The safety functions and their respective purposes as they are given above are quite straightforward and a detailed discussion of each is not deemed necessary here. However, some discussion of the intent of defining these functions may be appropriate. In the application of the safety function concept it will be necessary to define all the systems (and perhaps ultimately all the components) that are required to perform each of these functions. It will be essential that all the required systems are in fact identified, e.g., the maintenance of reactor coolant inventory in an operating PWR requires not only the charging pumps with a supply of water, but also motive power, instrument power, cooling and lubrication, as well as environmental control for these systems. While this systems identification may be reasonably straightforward for some of the functions, it could get quite complicated in such areas as the maintenance of vital auxiliaries. The latter, however, could be a principal source of difficult to recognize systems interdependencies. The safety functions as defined above would apply to reactors in general, i.e., all plants must perform these basic safety functions. However, the systems and components used to achieve these functions can be quite different. While these safety functions are general enough to apply to all modes of reactor operation, the nature of a function as well as the function priority will clearly change with the operating mode.

Given the foregoing discussion of systems, interactions, and safety functions we can pose a definition of systems interaction as it will be used in the subsequent discussion:

> Systems Interaction (SI) - safety system failure combinations, resulting from some external event or malfunction of some interdependent system, that can reduce the effectiveness of any one of a number of basic safety functions.

A key aspect of the above definition is "system failure combinations". Within the present context multiple independent hardware failures do not constitute systems interactions, neither does a single external event that fails multiple systems.

Nuclear power plants are designed and operated such that there are normally several ways that can be used to achieve any given safety

function, i.e., for each safety function there are typically several possible success paths. This is an essential ingredient of the defense-in-depth approach to reactor safety. The defense-in-depth is achieved through the use of such design approaches as redundancy, coincidence, functional diversity, independence, physical separation, quality assurance and testing. If it were not for such approaches, the potential for systems interaction would not exist. In that case, the reliability of the system would be governed by single failures. The potential for systems interaction (and also common mode/common cause failure) is the result of the complexity of the system. If executed properly this complexity leads to a level of safety function reliability much higher than can be achieved in a simple system. If the potential pitfalls of this complexity (such as systems interaction) are not recognized and properly addressed, the desired gains in reliability may not be achieved.

A key aspect of any reliability assessment and one of particular importance to the problem at hand is the question of system and/or component independence. As is well recognized, reliability assessments based on the assumption of independent failure lead to optimistic predictions of system reliability. Certain types of dependencies among systems and/or components are fairly readily recognized; among these may be such items as common location, power supply, actuation, etc. These have received much attention in the recent past in the context of common mode/common cause failures. Certain other types of dependencies are much more difficult to recognize and evaluate; among the latter are the extremely broad area of human factors and subtle dependencies in functionally widely separated systems. These are the areas of primary concern from the systems interaction viewpoint. In a sense, systems interaction analysis can be considered as a search for hidden dependencies.

Several examples of systems interactions resulting from operating experience are described briefly below.

The Systems Interaction Branch has thoroughly reviewed and evaluated the February 26, 1980, event at the Crystal River Unit 3. This review clearly showed that several undesirable functional dependencies in the non-nuclear instrumentation power supply lead to an uncontrolled loss of primary coolant when a failure occurred in that power supply, due to the dependence

of the integrated control system on the input from the non-nuclear instrumentation. A similar event occurred at Rancho Seco Unit 1 on March 20, 1977. In this case, a short (operator-caused) in a non-nuclear instrumentation power circuit caused the loss of about two-thirds of the non-nuclear instrumentation and also caused erroneous inputs to the integrated control system, which resulted in the loss of both main and auxiliary feedwater to the steam generators. Although this event did not involve an uncontrolled loss of coolant, the cause would be classed, by definition, as a systems interaction because it resulted in the degradation of a safety function: loss of the principal RCS heat removal path.

On March 14, 1971, a workman at H. B. Robinson Unit 2 failed to terminate the testing of a battery-supplied oil pump. The batteries discharged to the point that the low voltage caused the failure of several auxiliary electrical circuits; one of these was associated with shaft cooling for the primary coolant pumps, and its failure resulted in the loss of the pumps and the degradation of the core heat removal function. From an accident point-of-view, this event is important because it caused significant damage to major equipment (the turbine bearing lubrication system also failed); from a safety point-of-view it is important because of the contribution of a systems interaction.

The July, 1980, issue of "Power Reactor Events" describes an event that occurred at St. Lucie Unit 1 on June 11, 1980, that involved a systems interaction. In this case, a minor steam leak caused an electrical short that closed a containment isolation valve in the common return line for the component cooling water to the reactor coolant pumps. The result was the loss of the pumps and the degradation of the core heat removal function. As is often the case, the sequence progressed and difficulties were encountered in establishing natural circulation cooldown, including the formation of a steam bubble in the reactor vessel.

DESIRABLE SYSTEMS INTERACTION METHODOLOGY ATTRIBUTES

The recognition of the need to consider the potential effects of systems interaction reflects a desire to identify hazards that otherwise would be missed or to highlight "everything that we forgot". In this light the best hope for a successful approach for the identification and evaluation of systems interactions would appear to be the development of a formal methodology for this purpose. Broadly speaking such a methodology should have the following attributes: systematic, complete, flexible, reproducible, simple, and visible or scrutable. These desired attributes are discussed below.

The methodology is "systematic" if it follows a clearly defined sequence of analysis. A "complete" methodology would cover all the significant areas within its range of applicability. "Flexibility" is the ability to adapt to elements of varying complexity as well as varying situations. A method is "reproducible" if its application in an independent analysis will yield equivalent results. A "simple" methodology will be characterized by ease and consistency of application. "Visibility or scrutability" implies that the basis for the method and the results obtained can be presented and understood by others.

Among other attributes that the methodology should have are both an identification (qualitative) as well as an evaluation (quantitative) function. The qualitative analysis should focus on fundamental relationships among systems and subsystems as they relate to the execution of a safety function. The quantitative analysis is required to screen according to their safety significance as well as to determine system sensitivity to data and model uncertainties.

The desirable systems interaction methodology attributes discussed above are to a great extent mutually exclusive. As an approach tends to get more complete, it generally also gets more complex and less scrutable; the simpler methodologies may tend to be more reproducible, but less complete, etc. The more powerful methodologies require greater skill on the part of the analyst and have greater support requirements, such as computer capabilities.

Since the definition of systems interactions as used here is quite broad, it can be expected that many such potential interactions will be identified by whatever methodology that may be utilized. In such a case, it may be essential to be able to screen and rank the potential interactions in order to reduce to a reasonable level the number of detailed evaluations and/or the number of actions aimed at mitigating such interactions. An obvious way of screening is on the basis of probability. This, however, would require quantitative evaluation of all potential interactions prior to screening and thus could not aid in reducing the extent of detailed analysis required. Thus, other means of screening and ranking potential interactions may be required. Other bases for screening might be the importance of the safety function affected, time dependence (e.g., the immediacy of the required action), and screening by categories. The systems interaction methodology selected should facilitate, or at least not preclude, screening of potential interactions at an early stage of analysis. If the number of potential systems interactions that have to be considered in depth is too large, the approach may be self-defeating.

It may be recalled that the systems interaction methodology to be developed is aimed at two broad applications; the first is the reactor license applicant's use of such a methodology in the development and review of the plant design, the second is the NRC's review of license applications from the systems interaction viewpoint. It may be useful to note that the methodology used by the applicant need not be the same as that used by the NRC. While the applicant's use of a methodology familiar to the NRC may facilitate its review, the use of a common or similar approach by both may suffer from generic deficiencies, i.e., a common cause/common mode failure. Further, it is likely that the depth and breadth of the analysis utilized by the applicant may very well be different from that of the NRC. It is possible, for example, that the NRC review may emphasize the qualitative aspects of systems interaction evaluation whereas the applicant would cover the quantitative aspects as well.

APPLICABILITY OF POTENTIAL METHODOLOGIES TO SYSTEMS INTERACTIONS

Appendix A of this report gives a review of potential systems interaction methodologies. While not necessarily exhaustive, this review describes in some detail the strengths and weaknesses of a variety of formal as well as less structured methodologies. Table 1 lists some of the more important basic characteristics of the methodologies under three major headings. "Basic Approach" refers to the major techniques used in the method. Fault trees are considered "logical" because they are based on logic models (AND/OR gates, etc.). Weighting factors are "mathematical" because they are based on numerical approximations (α, β, and γ factors). "Capabilities" refers to the types of analysis for which each methodology is appropriate. Physical survey involves a "walk-through" procedure coupled with some sort of checklist, primarily appropriate for a qualitative analysis. Marshall-Olkin specialization involves failure-rate models based on an exponential distribution, most appropriate toward a quantitative analysis. The GO methodology considers multiple event states corresponding to output occurrence times, appropriate when analyzing a time sequence of operation. "Applicability" refers to the level of plant detail which a methodology can examine. A physical survey is mainly limited to identifying component interactions, while a cause-consequence analysis can span the full range from components through functions.

In Table 2, some of the important aspects of the methodologies are qualified. In considering this table, it must be remembered that each methodology has its own range of applicability. Thus, any comparison among them based on these aspects must bear in mind the areas in which each is applied. For example, both FMEA and cause-consequence analysis are "complete". However, FMEA is "complete" on its prime level of identifying major failure modes for components, while cause-consequence analysis is "complete" in analyzing accident sequences.

TABLE 1. CHARACTERISTICS OF POTENTIAL METHODOLOGIES

Methodology	Basic Approach		Capabilities			Applicability		
	Logical	Mathematical	Qualitative	Quantitative	Time-Sequential	Components	Systems	Functions
Operational Survey			X			X	X	X
Physical Survey			X			X		
FMEA	X		X	X		X	X	
Digraph Method	X		X			X	X	X
Fault Trees	X		X	X		X	X	X
Phased Mission	X		X	X	X	X	X	X
Event Trees*	X		X	X	X		X	X
Cause-Consequence	X		X	X	X	X	X	X
GO	X	X		X	X	X	X	X
Markov Modelling		X		X	X	X	X	
Generic Analysis	X		X	X		X		
Weighting Factors		X		X		X	X	
Marshall-Olkin		X		X		X		

* Refers to event trees only. Event trees plus conditional fault trees are considered to be cause-consequence analysis.

DRAFT

TABLE 2. ASPECTS OF POTENTIAL METHODOLOGIES

Methodology	Systematic	Complex	Complete	Reproducible	Flexible	Visible
Operational Survey	Potentially	Potentially	Somewhat	Somewhat	Yes	Yes
Physical Survey	Somewhat	No	Somewhat	Somewhat	Yes	Yes
FMEA	Yes	Somewhat	Yes	Yes	Somewhat	Somewhat
Digraph Method	Yes	No	Yes	Yes	Yes	Yes
Fault Trees	Yes	Yes	Yes	Yes	Somewhat	Somewhat
Phased Mission	Yes	Yes	Somewhat	Yes	Somewhat	No
Event Trees*	Yes	Somewhat	Somewhat	Yes	Somewhat	Yes
Cause-Consequence	Yes	Yes	Yes	Yes	Somewhat	Somewhat
GO	Yes	Yes	Yes	Somewhat	Yes	No
Markov Modelling	Yes	Somewhat	Yes	Somewhat	Somewhat	Somewhat
Generic Analysis	Somewhat	Somewhat	Somewhat	Somewhat	Yes	Somewhat
Weighting Factors	Slightly	No	No	Slightly	Yes	Slightly
Marshal-Olkin	Slightly	No	No	Slightly	Somewhat	Slightly

* Refers to event trees only. Event trees plus conditional fault trees are considered to be cause-consequence analysis.

Systems interactions can take place either exclusively on the system level or through the component level. Consider Figure 1. Systems B & C interact exclusively at the system level, while systems C & D interact through components C_2 & D_1. As an illustrative example, consider the small LOCA accident scenario in Figure 2. This is most easily transformed into the event tree of Figure 3. From there, it can be seen that if both HPCI and APR fail ($\overline{H}$ & $\overline{A}$), the LP-ECC systems cannot be used to mitigate the potential consequences. This is a result of the failure of APR to reduce vessel pressure in the event of HPCI failure. Both LP-ECC systems may be available, but their design precludes operation at an elevated pressure. This represents a system interaction exclusively on the system level.

Figure 4 is a consequence fault tree for this same scenario. Here, the failures of the LPCI and the RHR systems have been resolved to the component level. For illustration, both the LPCI and the RHR pumps have been assumed to receive electric power from the same bus (bus A). Should this bus be lost, both the LPCI and the RHR pumps will fail due to loss of power, thereby failing their respective systems. This represents a systems interaction through the component level, a type of failure often referred to as "common-cause" because two or more components (LPCI and RHR pumps) failed due to a single, common cause (loss of power bus A).

To be useful in a systems interaction assessment, the methodology must be capable of identifying at least some of the interactions on at least one of the two levels (component or system). It is further desirable that the impact of the interaction on plant safety as a whole be evaluated for ranking purposes. The following discussion views the methodologies in this framework - identification and evaluation of systems interactions.

1. Identification of Systems Interactions

As previously mentioned, systems interact either exclusively at the system level or through the component level. Most of the methodologies examined are capable of identifying interactions on at least one of these levels, while some are applicable to both. The plant review necessary in a systems interaction assessment would begin at the most general level of plant safety, shown at the top of the hierarchy in Figure 1. Next would come definition of the various safety functions contributing to plant

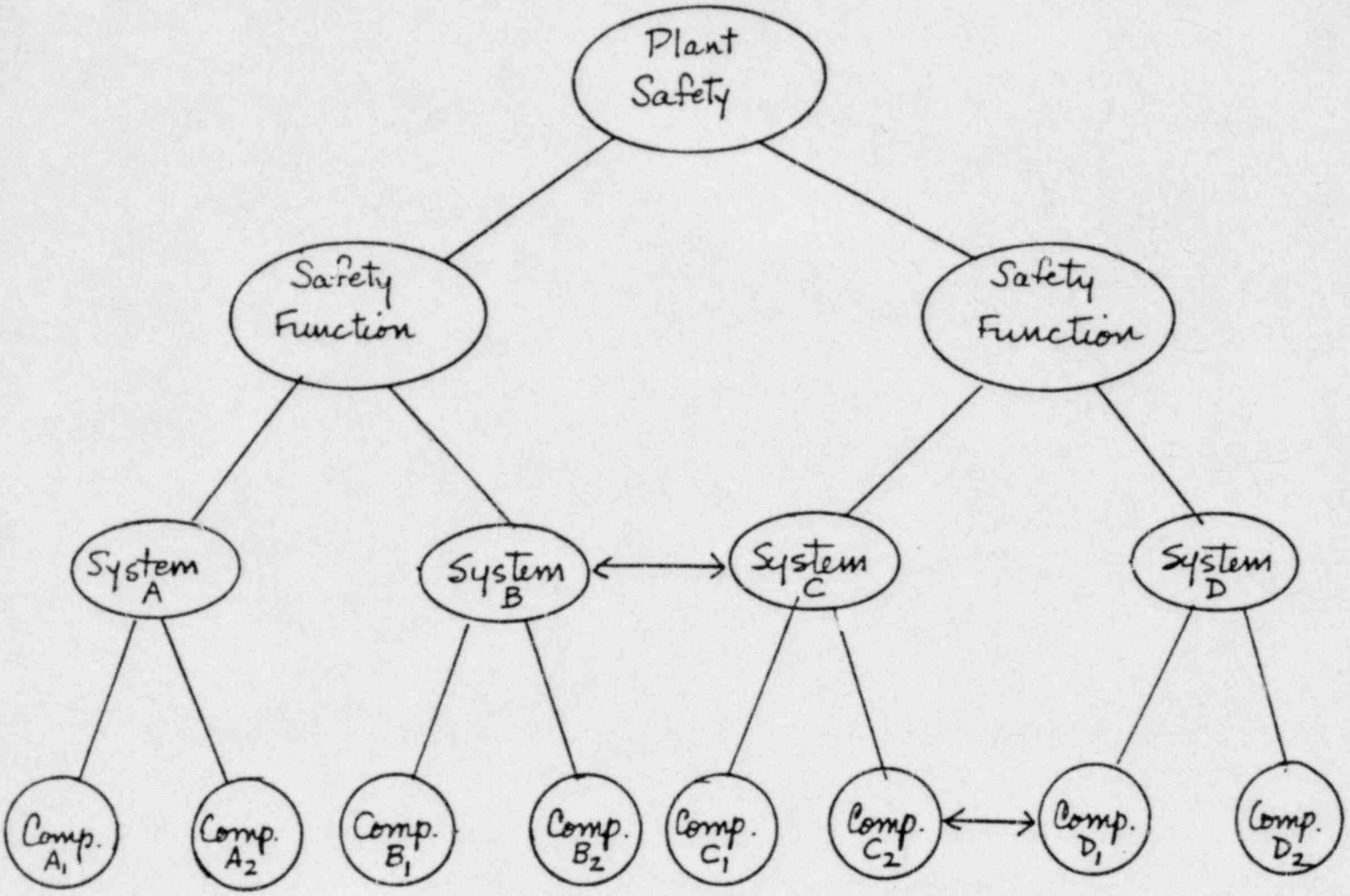
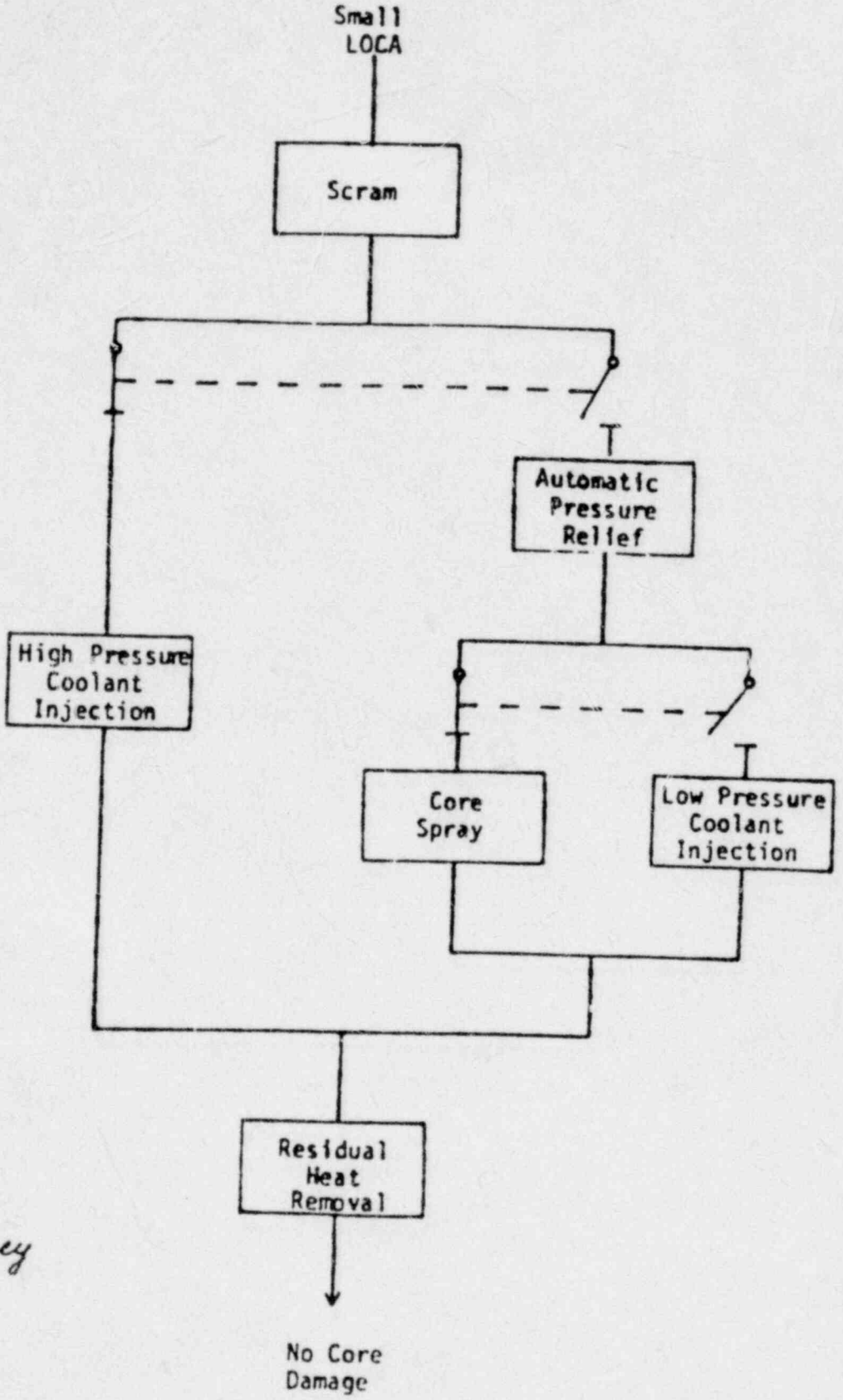


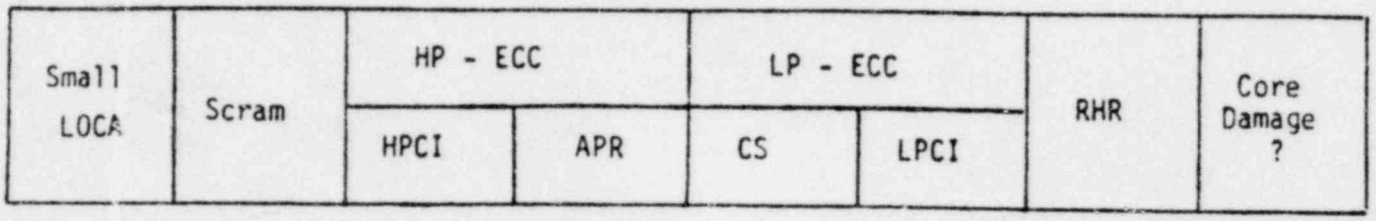
Figure 1. General Plant Hierarchy Showing Levels at Which Systems Interactions May Occur

DRAFT



Note: Dotted line indicates standby redundancy

Figure 2. Small LOCA Accident Scenario



Note: At branching points, upper branch denotes success, lower branch failure

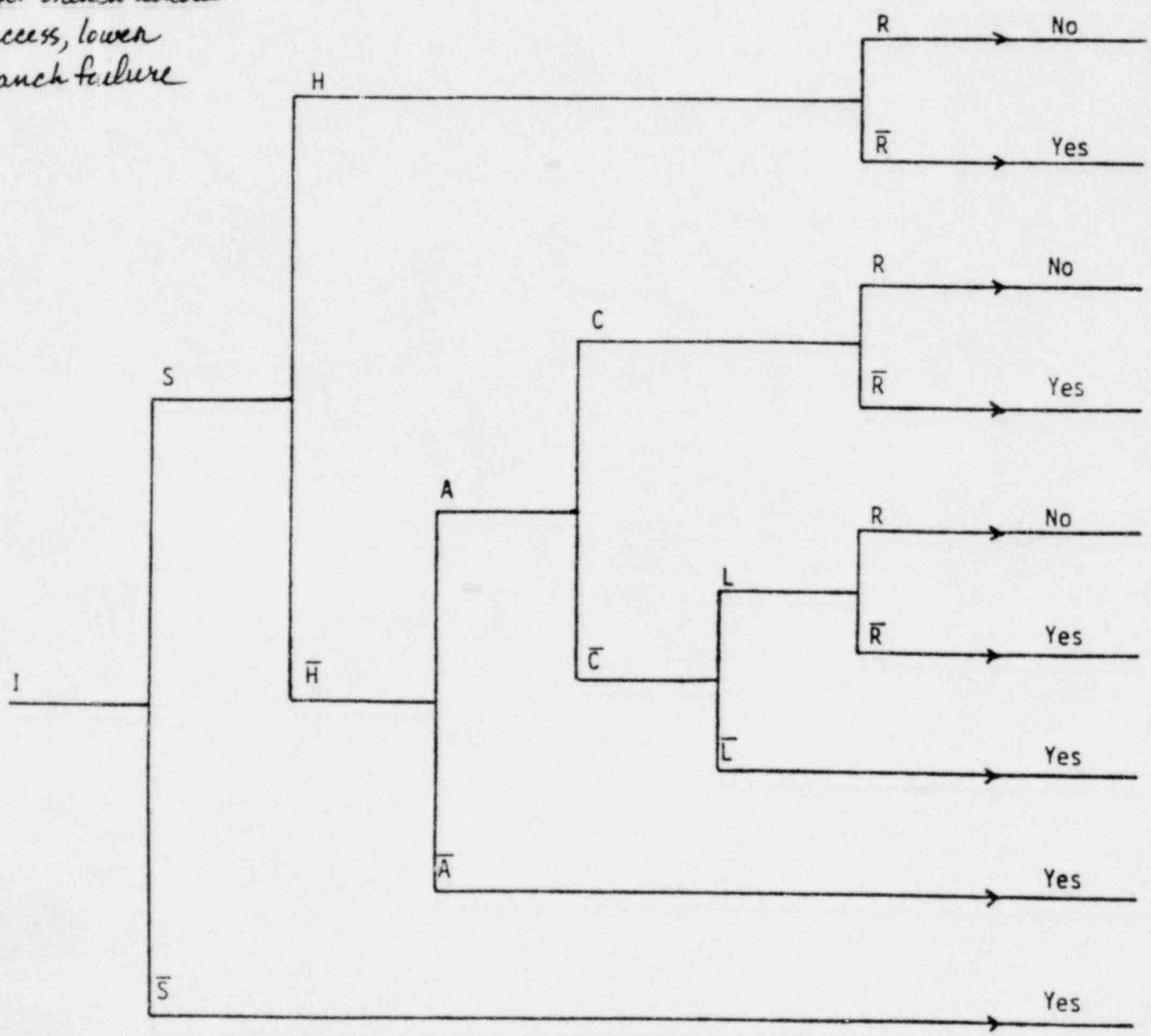


Figure 3. Small LOCA Accident Scenario Event Tree

DRAFT

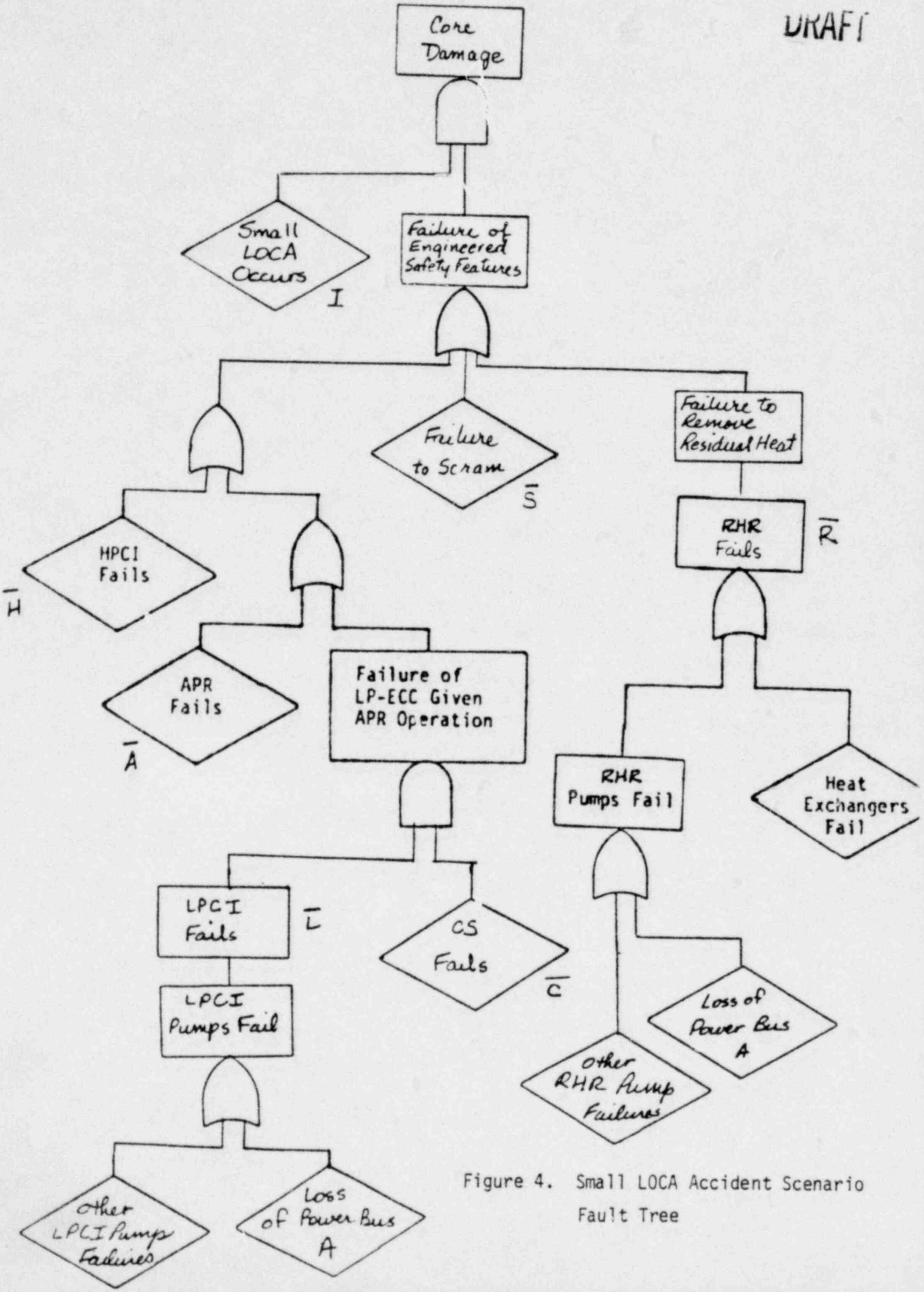


Figure 4. Small LOCA Accident Scenario Fault Tree

safety in all operating modes. Following this would be identification of the various systems needed to perform the safety functions. It is at this level where the systems interaction assessment should begin.

At the system level, the analyst seeks to identify interactions occurring exclusively at this level (such as APR and LP-ECC systems in the small LOCA accident scenario). Table 3 lists the methodologies capable of identifying these. An operational survey coupled with an FMEA on the systems rather than components could serve as a good starting point, especially since some sort of operational survey would be necessary to go from the top level of general plant safety down to the system level. The system FMEA could be helpful in identifying potential modes of interaction.

The identification of the various systems needed to perform the basic safety functions should be followed by the identification of the systems and subsystems needed to support them. This may involve consideration of secondary, tertiary, and other support systems and may to some extent extend to the component level. It is likely that interactions resulting from failures of the supporting systems will be manifested through the components of the systems directly responsible for the safety functions. Interactions at this level often involve "common cause" failures, i.e., multiple or dependent component failures due to common single events.

Table 3 lists the methodologies capable of identifying interactions at the component level. The operational survey would extend to this level and, coupled with a physical survey, would form a good starting point for identifying component interactions. Component FMEA and the digraph method would aid in systematizing the identification process, while a generic analysis should reasonably ensure that no major component dependencies have been overlooked.

Note that not all component interactions need result in systems interactions. If the interacting components are totally contained within a single system, their failure may affect only that system. This would not necessarily constitute a systems interaction unless failure of that system affected others. Thus, generally more component interactions are identified than actually lead to systems interaction. Only those leading to systems interaction need be retained for subsequent analysis.

TABLE 3. APPLICABILITY OF POTENTIAL METHODOLOGIES TO SYSTEMS INTERACTIONS

	Identification		Evaluation		
Methodology	Components	Systems	Components	Systems	Plant Modes
Operational Survey	X	X			
Physical Survey	X				
FMEA	X	X			
Digraph Method	X				
Fault Trees			X	X	
Phased Mission			X	X	X
Event Trees*				X	
Cause-Consequence			X	X	
GO			X	X	X
Markov Modelling			X	X	X(limited)
Generic Analysis	X		X		
Weighting Factors			X(limited)		
Marshall-Olkin			X(limited)		

* Refers to event trees only. Event trees with conditional fault trees are considered cause-consequence analysis.

2. Evaluation of Systems Interactions

Following the identification of the systems interactions, it is necessary to evaluate their impact on plant safety. This involves analyzing the interactions on both the component and system levels and extending the results up through the function level to overall plant safety. Some of the methodologies are particularly suited toward analysis over this full hierarchal structure while others are more suited to one level.

Cause-consequence analysis, or the equivalent event tree-conditional fault tree analysis, is probably the best known methodology for analysis over the total hierarchy. This is essentially the technique employed in the Reactor Safety Study. The event trees are especially suitable for modelling functional losses in terms of contributing system failures. These can subsequently be extended to the component level through conditional fault trees for the systems. This is amenable for both qualitative and quantitative evaluation, but it suffers somewhat from a difficulty of keeping track of component interactions since they are generally indicated on separate fault trees.

Consequence fault trees reduce this difficulty by integrating the entire analysis onto single fault trees. Both system and component level interactions are indicated on one tree for each accident consequence. The amount of representation is basically the same since one large tree must be drawn for each consequence. (The cause-consequence analysis requires one dual tree for each initiating event.) However, fault trees are generally more difficult to conceptualize than event trees, a problem magnified by the large size of consequence fault trees. Thus, even to perform an analysis using consequence fault trees, it may be necessary to first construct event trees to aid the analyst in visualizing the situation.

Perhaps the most powerful methodology is the GO method, capable of total hierarchal analysis with the added advantages of time-modelling and integration of hardware operation with logic functions into a single analytical structure. However, the cost of such increased capability is additional complexity, which may be prohibitive when attempting to utilize its full potential. The GO methodology has an advantage over a fault tree approach in that it works from a success viewpoint, generally easier to visualize than failure combinations. The allowance for multiple event states also gives it the potential for partial failure analysis, as opposed to the

total success/failure analyses inherent in the other methods allowing only for binary states. Unlike consequence fault trees and cause-consequence diagrams, it does not readily lend itself to qualitative analysis.

Other methods do not span the total hierarchy of Figure 1, but they are capable of evaluating certain aspects of systems interactions. A reasonably versatile method that can be applied on both the system and component levels is Markov modelling. Interactions on these levels can be mathematically modelled by transitions among states with varying redundancy. Being a mathematical technique, Markov modelling is inappropriate for qualitative analysis. It is primarily a probabilistic technique. The simplifying assumption that succeeding states depend solely on their immediate predecessors may be too restrictive for some more complex interactions However, it does provide for time-dependency, although not as extensively as does GO (or with as much complexity).

Somewhat empirical are the weighting factor method and the Marshall-Olkin specialization. They are applicable primarily on the component level, although the β-factor technique can be extended to interacting systems. They do not attempt to identify dependencies. Rather, they are designed to provide a quantitative means of approximating failure rates for dependent components and would be applicable only during probabilistic evaluation of systems interactions. They are inappropriate for qualitative analysis.

A thorough, qualitative method for evaluation of component interactions is the generic analysis approach, specifically through the Boolean transformation technique. Used primarily in conjunction with minimal cut sets from a fault tree analysis, generic analysis identifies component interactions and traces their effect on system failure by the Boolean transformation technique. Quantitative evaluation can be incorporated through the Boolean expression for system failure, which is basically an algebraic representation of an equivalent fault tree.

Systems interactions may sometimes involve changes in plant operating modes and similar time-related phenomena. Both the GO methodology and Markov modelling have been mentioned as possessing time-modelling capability. Another technique, which is an extension of fault tree analysis, is phased mission analysis. Although not as powerful (or complex) as GO, it provides

a means of analyzing a system or function which performs different roles during different plant modes. Being a fault tree technique, it can model both component and system level interactions, but it is restricted to modelling only the same systems and non-repairable components throughout the mission time.

Table 3 summarizes the methodologies which have evaluation, as well as identification, potential for systems interactions based on their level of applicability (system and/or component). Also included are those applicable to evaluating interactions involving changes in plant mode.

RECOMMENDED APPROACH

In considering the various methodologies and their attributes as discussed above, it appears that the most promising techniques are those utilizing logic models such as fault trees or event trees. These highly structured approaches provide a framework for describing the system and for a step-by-step examination of system behavior at a fine level of detail. This ability to treat the system in very fine detail can be both an asset and a liability. It permits tracing the causes of system (function) failure (an presumably systems interactions) to failures or deficiencies at the fundamental component level. The detail of analysis permitted by these methods requires an understanding and modeling of the structures of the system, the operation of each of the components, the inputs that control the system, and the resultant outputs in commensurate detail. In a system as complex as a nuclear power plant, this level of detail can be overwhelming. In order to make the analysis tractable, the analyst is very quickly forced into compromises such as making simplifying assumptions, ignoring "unimportant" systems, limiting operating modes under consideration, working on only portions of the system at a time, etc. All these compromises reduce the utility of the basic methodology. In the extreme, if enough such compromises are made, the analysis is reduced to that of the effect of independent hardware failures in redundant trains, neglecting such key aspects as potential internal dependencies and human interaction. Thus, a conceptually powerful methodology can be reduced to a trite exercise due to the sheer magnitude of the problem.

Fault tree based approaches to systems interaction evaluation, such as the SETS method, are generally based on the premise that potential systems interactions can be found by identifying commonalities between the components of the systems. In principle, this premise should be quite valid. However, by immediately focusing on the components that comprise the system, the methodology is confronted with a problem of enormous magnitude. In a system as complex as a nuclear power plant, just the sheer number of components may overwhelm even the most powerful analytical methods and computer facilities. Thus, compromises in the analytical approach must be

made, particularly in the depth of evaluation that is performed. Among the earliest casualties of these compromises are the support systems to the principal safety functions. The sheer number of components that must be considered does not necessarily preclude the use of such methodologies, e.g., the identification of components that may be shared by several systems, or components that share the same location may still be quite feasible. Other linking characteristics such as those associated with calibration, test, and maintenance would be difficult to evaluate on a component by component basis.

The need to consider systems interaction effects stems from the realization that it is the reliability of a system (function) that is the principal safety concern rather than the reliability of components. The reliability of a system depends not only on the state of components but also on potential dependencies among seemingly independent systems and also on design deficiencies. The human factor is probably the dominant linking characteristic and could very well be the most likely source of systems interactions. Physical interdependencies which are not recognized are obviously also possible, these can be expected to result from subtle and obscure causes.

The human factor can affect the plant safety functions in a dynamic or a latent fashion. The dynamic mode results from the fact that the human may be required and/or permitted to act in the event of a plant upset. The latent mode of human interaction may go all the way back to design and manufacturing deficiencies, but most likely will be associated with calibration, test, maintenance, and related activities that can leave affected portions of the plant in a degraded condition. Such degradation may not manifest itself until the affected system is required to mitigate the effects of some abnormality.

The unrecognized physical interdependencies can originate anywhere in the plant, but the more likely places are in secondary, tertiary, and other support functions. As has been noted by others, systems interaction evaluation cannot stop at so-called "safety related" systems; all systems that contribute to the basic safety functions are potentially important.

Since the requirements on a methodology to identify and evaluate systems interactions are broad and to some extent conflicting, it is suggested that the methodology focus on the basic safety functions rather than addressing the plant on the component level. It is further suggested that logic models such as fault trees be adapted to evaluate system behavior and potential systems interactions on a functional or systems level. The suggested approach is outlined in Figures 5 and 6.

By focusing on the basic safety functions, the safety systems required to perform these functions, and the vital support systems it is felt that the approach can retain the requisite depth of analysis without getting bogged down in the detail associated with basic components. Of the available methodologies, the event tree approach appears to be most suited for application at the functional or systems level. An event tree begins with some initiating event and maps out a variety of sequences involving faults at the system level, each of which represents a particular consequence. A complete event tree analysis would require identification of all significant initiating events and the development of an event tree for each. Extensive overlap of consequences among the branches of the several trees can be expected. Each accident sequence leading to a particular consequence in an event tree is somewhat analogous to a cut set on a fault tree. Whereas a cut set represnets a combination of failures leading to the top, or undesired, event, an accident sequence represents a combination of system successes and/or failures leading to a given consequence. The difference in reference points between event tree and fault tree analysis suggests that event trees may be more appropriate when the initiating events are known, while fault trees may be more appropriate when the consequences can be identified more easily. The latter is the situation with the problem at hand.

Although traditionally fault trees have been used to model system failure in terms of failure of its basic components, fault trees should also be useable to model accident sequences with the top event being some consequence of those sequences. The use of fault tree methodologies in this context is being suggested for the evaluation of systems interactions. It is further suggested that resolution of the analysis be initially limited to the system or subsystem level. Most previous applications of fault tree analysis have

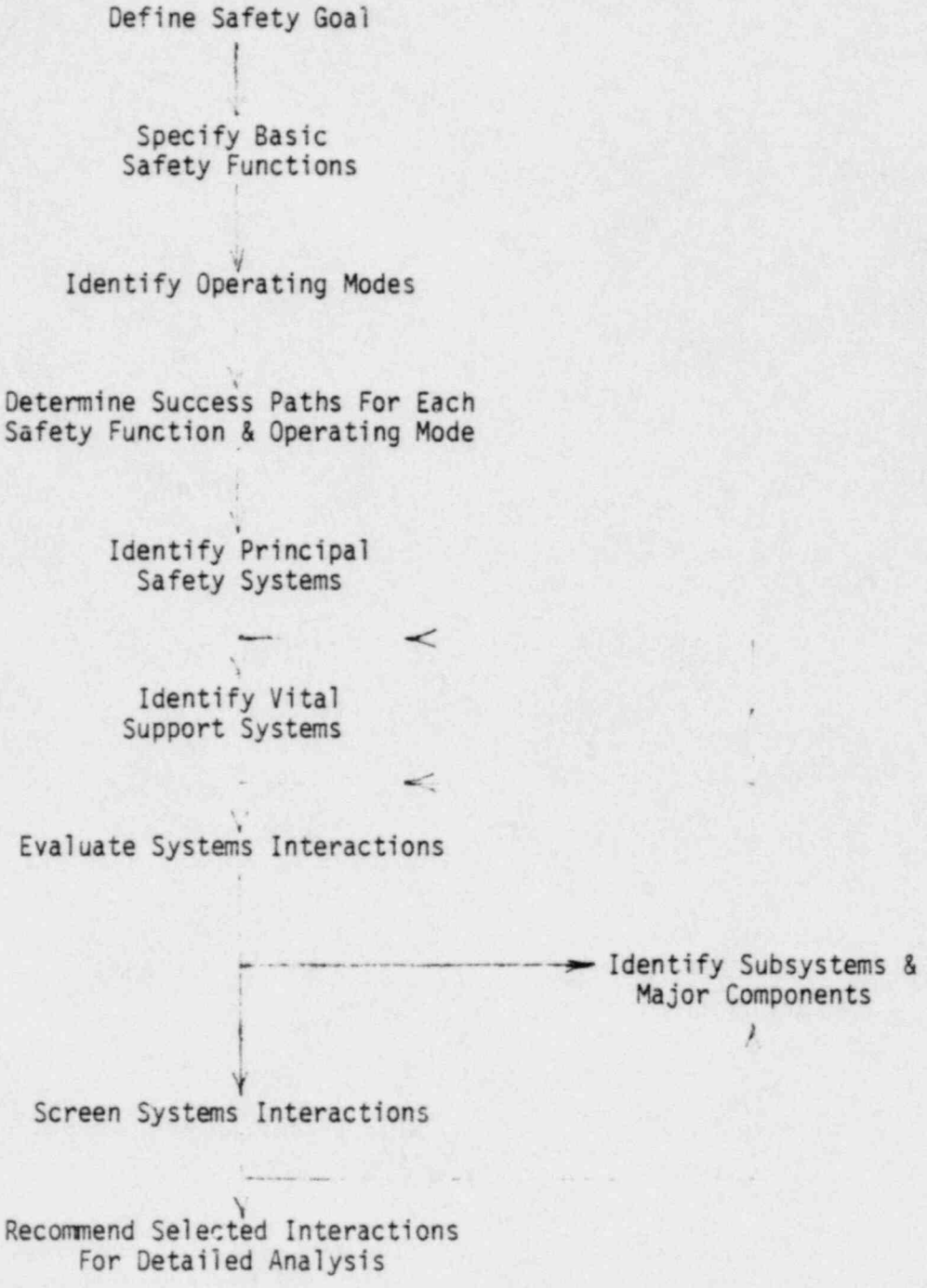


Figure 5. Qualitative Systems Interaction Evaluation

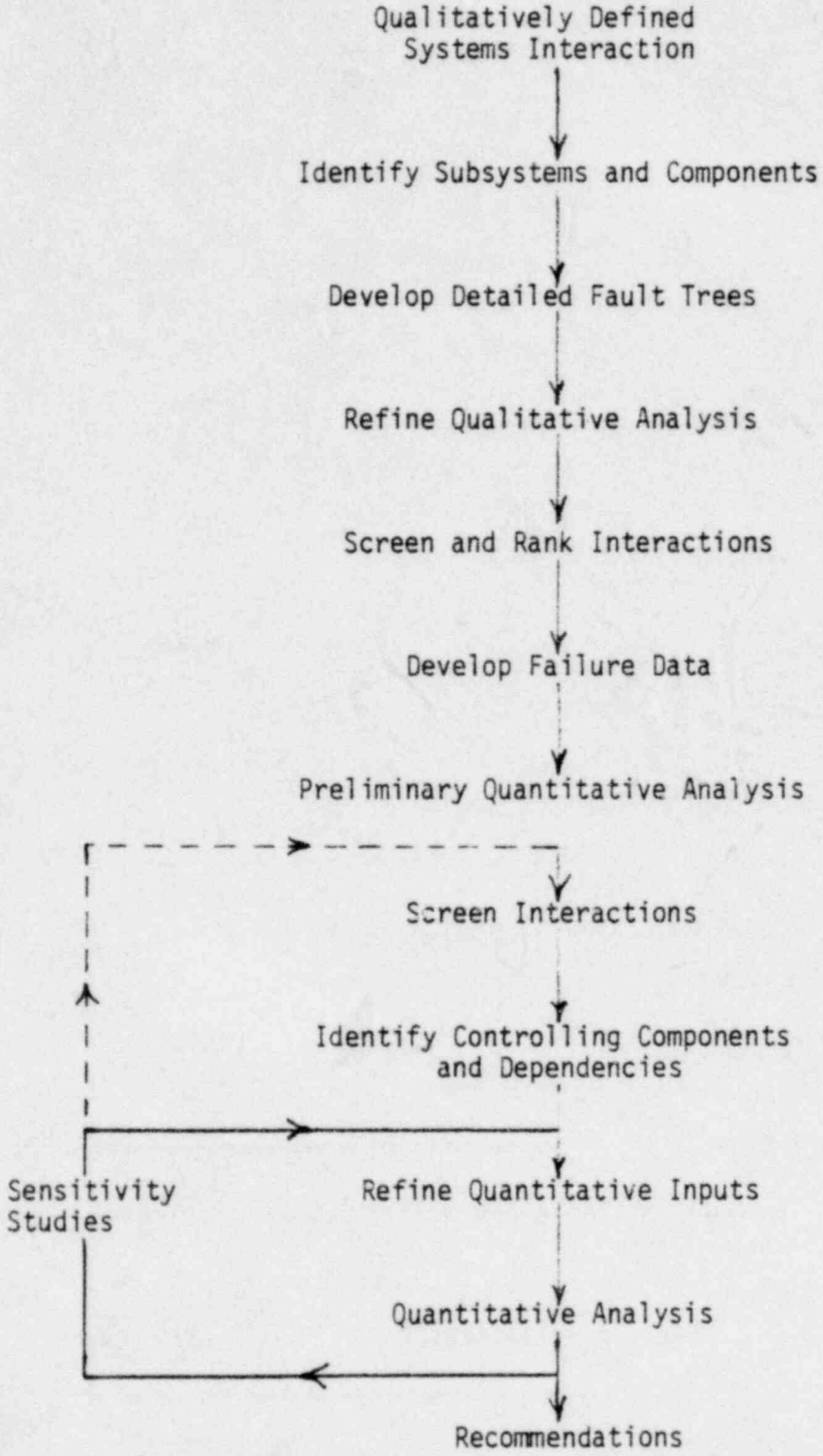


Figure 6. Quantitative Systems Interaction Evaluation

tended to resolve systems to the component level, where failure data is more readily available. For qualitative analyses where the identification of potential systems interactions is the most important aspect, the lack of failure rate data at this level is not particularly important. Those interactions that are considered to be significant after screening of the qualitative results can subsequently be subjected to a more detailed analyses, including detailed fault trees. A further motivation for initially focusing on the systems level is the realization that in complex systems the validity of reliability estimates may be governed more by the assumptions used in modelling the system than by the failure data utilized. By limiting the initial analysis to the systems level it is hoped that the modelling approach can retain many of the subtle interdependencies that may be lost due to the truncations and compromises that are necessary when a high degree of detail for the entire system is attempted.

The use of the same basic methodology for both the qualitative as well as the quantitative portions of the analysis, e.g., use of fault trees for both rather than a combination of event trees and fault trees, would have the following advantages:

a) it should facilitate a consistent transition from the qualitative to the quantitative mode of analysis,
b) it should permit whatever degree of iteration may be required, as later analyses indicate the need for more resolution, particularly for the more important interactions that may be identified,
c) the depth of analysis can be carried out to whatever level of detail is desired, or stopped at any level of interest, and
d) the presentation of the results and the scrutability of the methods should be enhanced.

Some further thoughts on addressing the systems interaction problem from the systems or functional level are as follows. As was noted earlier, human interaction can be expected to be a major linking factor leading to potential systems interactions. The latent mode of human interaction deals with such aspects as calibration, testing, and maintenance. While all these activities relate to individual components, it is the function of the system that contains the affected components that is concern.

Further, the above activities are more often than not conducted in the context of checking, testing, or repairing a system. E.g., it is the ECC system set points that are calibrated, though the actual calibration is performed on a very specific set of components; it is the ECC train "A" that is being tested and/or repaired and thus taken out of service. Thus, it may be ratural to assign such human interactions to the sytem or subsystem level rather than that of the individual components. The fact that there are far fewer systems than components obviously facilitates the consideration of these interactions at the systems level.

The application of fault tree methodology to system reliability assessment and, to a more limited extent, common cause/common mode failure analysis is broadly accepted. There are numerous automated techniques for developing fault trees as well as evaluating them. For the reasons cited previously, most fault tree analyses have focused on the hardware and aimed at system failures originating due to component failures. The use of fault trees at the system level as suggested here has received only limited attention. Again, for reasons cited previously, the "traditional" fault tree analyses approaches are felt to have limitations for application to systems interaction evaluation. However, in view of the deomnstrated capabilities of this methodology and the existence of a base of capability in terms of experience and analytical tools, it is felt prudent to take advantage of this basis in the further development of a systems interaction methodology. This is the intent of the suggested approach.

Since the recommended methodology for addressing systems interactions concerns has not been demonstrated to be fully applicable, further development will be required. The directions of this development will be delineated further in the remainder of this initial phase, beyond what was possible in this initial draft of the study.

AN INTERIM APPROACH TO SYSTEMS INTERACTION EVALUATION

The review of the methodologies potentially available for systems interaction evaluation clearly indicates that a major analysis effort will be involved to analyze a plant in the breadth and depth required to find systems interations. If a structured systems analysis were made a requirement of the license application, the effort required by the utility applicants would be substantial. Considering the state-of-the-art of these types of analyses it is highly unlikely that the utilities would have access to a sufficient number of qualifed analysts over the next few years to meet such a requirement. Similarly, a very large quantity of information would be submitted to the NRC for review, implying a large committment of NRC staff. In view of these considerations an alternate approach to systems interaction evaluation is suggested which would be less formal and structured, but which could be implemented while the formal methodologies are undergoing further development.

The objective of the interim approach is not to abandon more structured methods but rather to use them, with other sources of information on systems interactions, to develop general principles and to identify specific problem areas. These general principles could then be used to formulate guidelines for the regulatory review of plant applications.

The sources of information available on systems interactions are:

1) detailed systems analyses (which either have been performed or are in progress,e.g., as part of the NRC research effort), and
2) operational experiences.

In the suggested approach, detailed systems analysis methods would continue to be developed, particularly with regards to their ability to identify systems interactions. These methods would be applied by the NRC contractors to some specific plant designs. For example, the effort currently being undertaken for the first set of IREP plants could be extended to examine the potential for systems interactions in greater detail. Similarly, Licensee Event Reports would be reviewed in some detail to identify the systems interactions that have occurred. Events would be identified which had either resulted in degradation of a safety function or which had the potential to

do so as the result of common cause relationships. Having identified important types of interactions from the analyses and from the review of events, general guidelines would be developed which could be applied in the regulatory review of applications. These guidelines could be developed into a generic checklist of potential systems interactions.

The following elements could form the basis for a regulatory review process which focused on system interactions.

1) Simplified Systems Analysis

A systematic approach must be taken in exploring the relationships between systems in a nuclear power plant. The plant is too complex and the relationships are too subtle for the reviewer to evaluate without the assistance of systems analysis techniques. At one end of the spectrum of complexity, the systems analysis method could be a detailed fault tree/event tree analysis. Such an approach does not appear practical in the short term. What is being suggested for this review would be much less complex. The steps of a method of this type are presented in Table . The analyses would be performed by the utility and submitted with the license application. The results would guide the reviewer through the important functional relationships in the plant. The reviewer could identify interactions at the systems level and some interactions at the component level. Such a method would clearly not be as effective in identifying interactions as a formal structured analysis. To aid in the review, however, the reviewer would be provided with a generic list of specific interactions for which to look as well as some general guidelines. In this manner, the results of detailed systems analyses and operational experiences can be used to augment the capability of the simple systems analysis approach. Presumably, the majority of important systems interactions can thus be identified.

Table presents the types of connections that can lead to systems interaction in complex systems. The systems analysis approach involved in this element of the review would attempt to identify physical and inherent interactions.

2) Review of Procedures, Technical Specifications, and Training Requirements

Human interactions are the most difficult aspect of systems interactions with which to deal. They transcend the entire plant and provide the potential for linkage between all components and systems. Although all plant management practices can affect the performance of plant personnel to some degree, many aspects of plant management are difficult to influence by regulatory control. For example, the regulator can have little affect on the quality of the environment (relationship between management and staff) in which the operators work, although this probably has a close relationship to the incidence of human errors. The regulator can, however, affect two fo the most important factors that influence personnel performance. Through the review process, he can help to assure that the training of plant personnel is adequate and that the procedures by which the plant is operated are written in a manner to reduce the occurrence of operator error as well as to reduce the potential impact of such error.

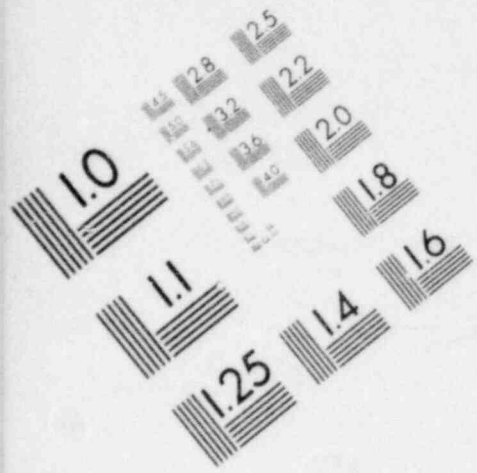
In this element of review, technical specifications, operating procedures, emergency procedures, and test and maintenance procedures would be reviewed to assure that the potential for interactions which can be introduced by the human is minimized. For example, well written procedures should not permit a single operator/technician to calibrate all of the corresponding instruments in redundant trains of a safety system; if systems have to be disabled for test or maintenance, the return-to-service procedures become extremely important; etc. Guidelines of this type would be provided to aid the reviewer. Consideration would also be given to the adequacy of training plans.

3) Plant Walk-Through

The final element of the review program would be a walk-through of the plant. The reviewer would be provided in advance with detailed drawings of the equipment location in the plant. The systems providing each of the principal safety functions and vital auxiliary functions could be identified

separately on the drawings to aid the reviewer in recognizing potential interactions. The review plan would provide specific guidance on relationships for which to look. The types of common cause connections (see Table 5) that could be identified in a walk-through would involve the spatial proximity of components to one another and to energy sources.

The elements of the suggested interim approach to the regulatory review of systems interactions have some capability to address each of the four major forms of common cause connections as described in Table 5. This approach would rely heavily on lessons learned from the review of operational experience and the study of detailed systems analyses. It is difficult to project how successful such an approach would be in identifying novel systems interactions which had not been found previously in other designs. This recommended interim approach parallels the more structured systems interaction evaluation methodology suggested earlier. The former minimizes the reliance on novel analysis techniques and exploits capabilities that are readily available. Although there are aspects of detailed systems analyses that are more promising, the alternative approach described above could be implemented within a comparatively short time. In addition, the approach could make use of the results of detailed systems analyses in a generic sense while these methods are being developed for application to specific design reviews, assuming that at some time in the future that would be practical.



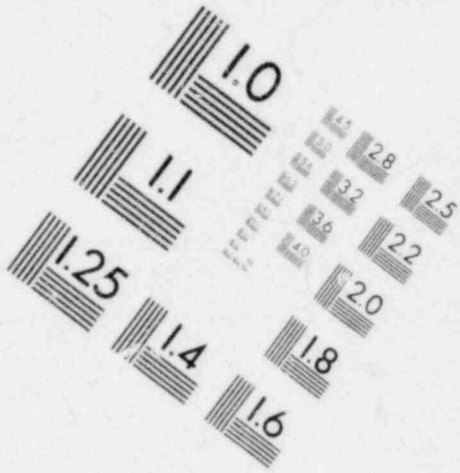
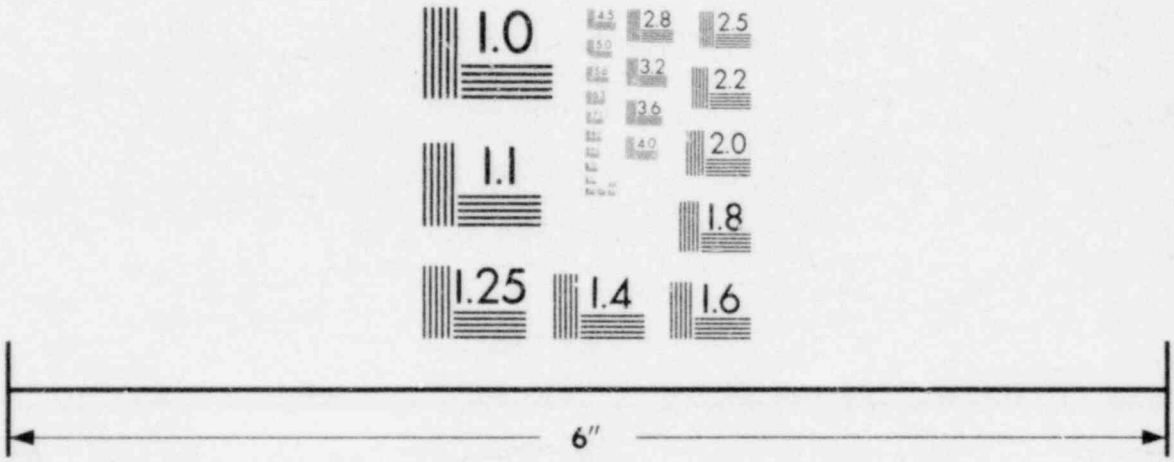
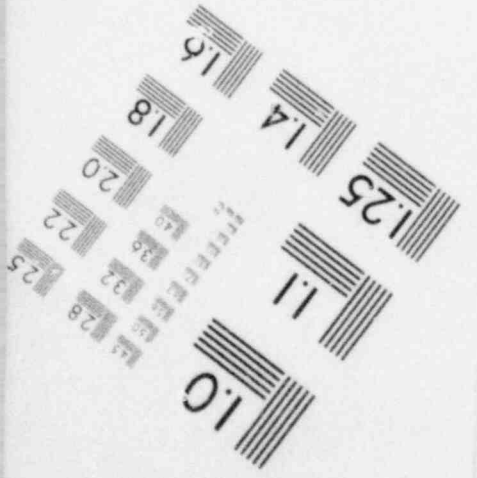
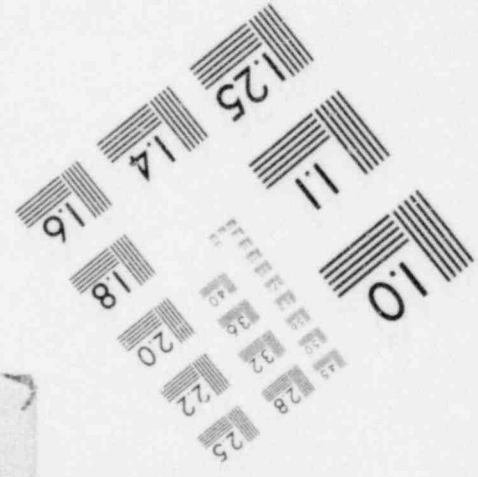
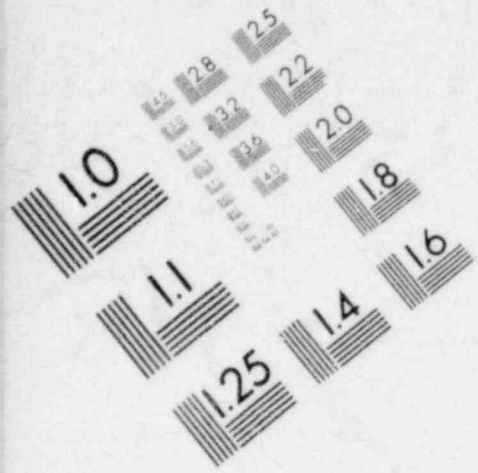


IMAGE EVALUATION TEST TARGET (MT-3)









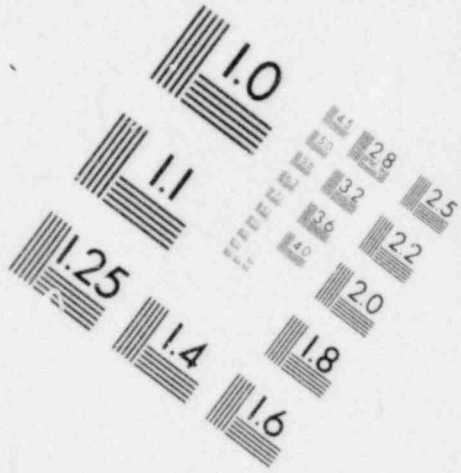
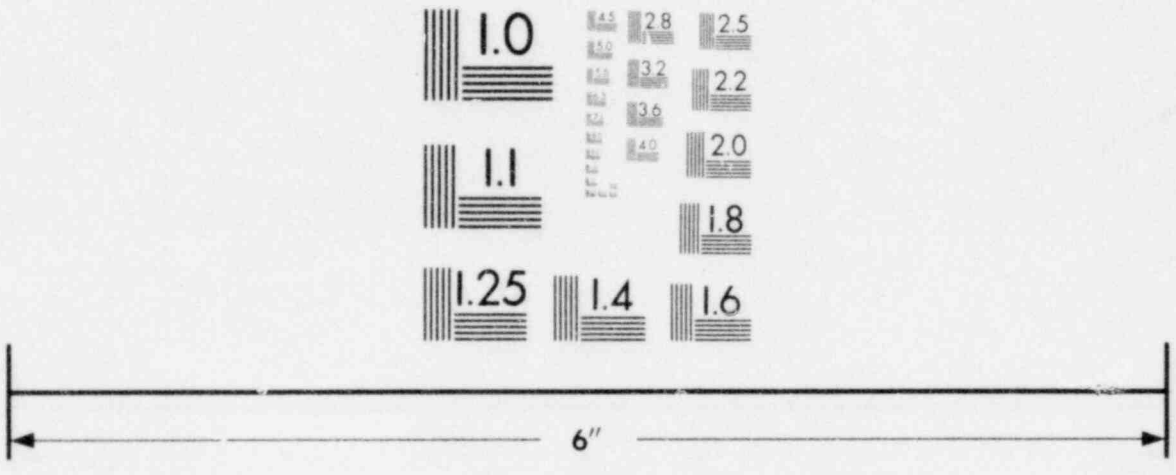
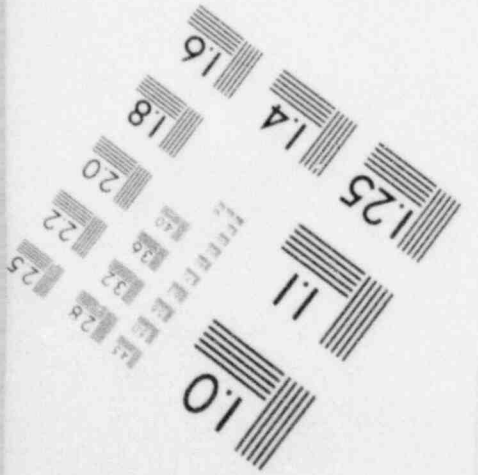
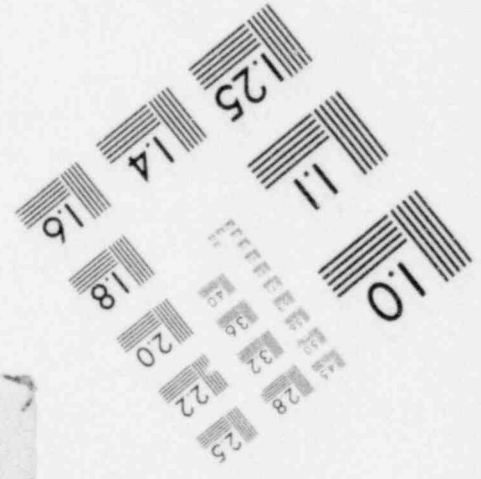
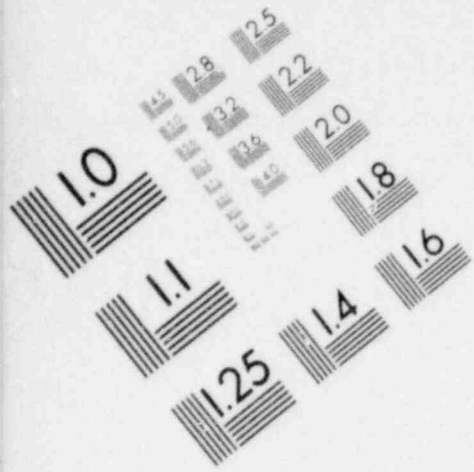


IMAGE EVALUATION
TEST TARGET (MT-3)









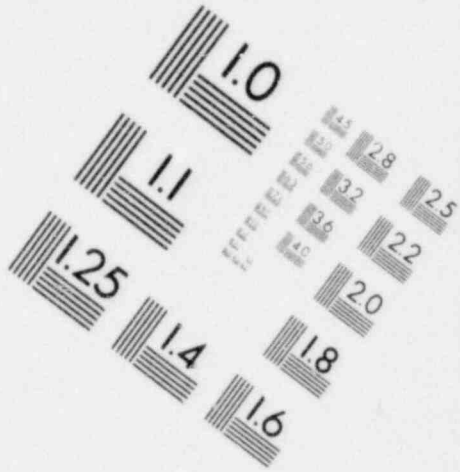
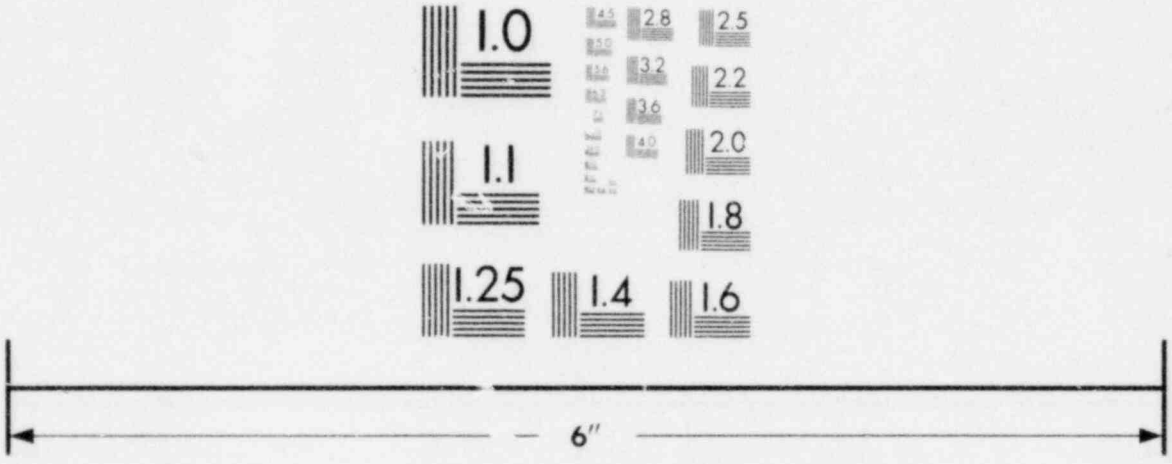
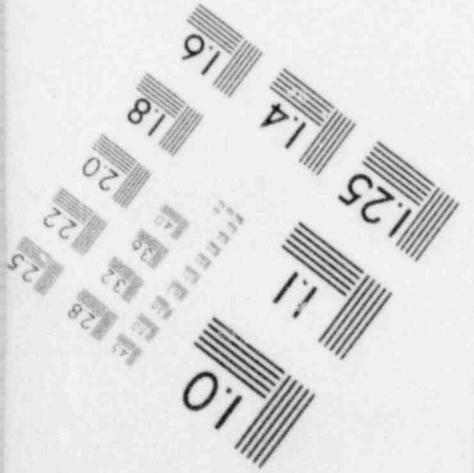


IMAGE EVALUATION TEST TARGET (MT-3)





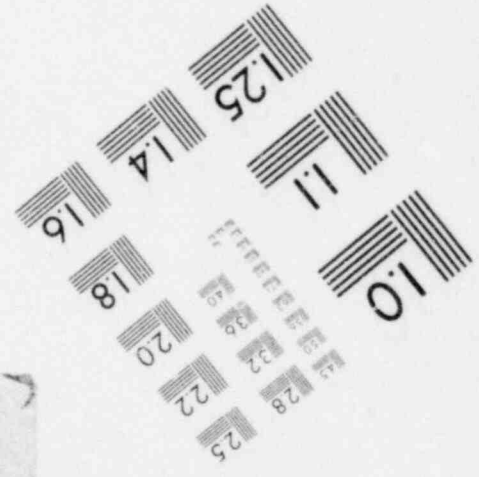


TABLE 4. FUNCTIONAL SUCCESS TREE APPROACH TO SIMPLIFIED SYSTEMS ANALYSIS

Analysis Steps

(1) For each of the prinicpal safety functions as previously defined, determine possible success paths for the plant starting from the principal operating modes.

(2) Identify each redundant train of the safety systems in the success paths.

(3) List all subsystems and major components within each train using unique identifiers.

(4) Define trains of vital auxiliaries providing motive power, control power, actuation, cooling, lubrication and environmental control for all components listed in Step 3.

(5) Scan system to identify:

(a) single failures that can disable two or more safety trains

(b) subsystems and components which are common to different safety trains or vital auxiliaries

(c) subsystems and component which are common to different safety functions in the same success path

(d) subsystems and components in different safety trains or different safety functions that are related by the potential linking characteristics of Table .

TABLE 5. REGULATORY REVIEW OF COMMON CAUSE CONNECTIONS

Connections*	Review Element
Physical	Simplified Systems Analysis
Electrical	
Mechanical	
Hydraulic	
Pneumatic	
Spatial	Plant Walk-Through
Thermal	
Fluid	
Mechanical	
Radiation	
Inherent	Simplified Systems Analysis
Common Manufacturer	
Similar Technology	
Equal Aging or Wear	
Shared Components	
Human	Review of Procedures, Technical
Dynamic	Specifications and Training
Latent	Requirements

* Boyd, G. J., et al, "Final Report - Phase I Systems Interaction Methodology Applications Program", NUREG/CR-1321 (April, 1980).

APPENDIX A

REVIEW OF POTENTIAL SYSTEMS INTERACTION METHODOLOGIES

Because systems interactions form a vital part of any thorough safety assessment, more general safety analysis methods form a convenient starting point from which to choose specific methodologies applicable to analysis of systems interactions. Both identification and analysis of system interactions must be provided for in any method or combination of methods selected for examination of these interactions. With this perspective in mind, it is convenient to divide the potential methods into two categories: qualitative and quantitative. This categorization does not necessarily imply that one group is more rigorous or formalized than the other, although this may be true for specific methods. The two categories are not mutually exclusive, since some methods have both qualitative and quantitative capabilities, such as fault trees.

A.1. Qualitative Methods

Four methods are discussed: operational survey, physical survey, failure modes and effects analysis (FMEA), and digraph methods. Of the four, the first two refer to somewhat informal review processes while the latter pair represent more formal techniques. As was previously mentioned, these methods may also posess limited quantitative capabilities. However, since their prime role is qualitative, they have been classified as such.

A.1.1. Operational Survey

"Operational survey" is a rather formalized name given to the detailed review process involved in ascertaining the functional relationships among systems. The analyst studies relevant documentation, including such information as found from system schematics, plant technical specifications and administrative procedures, and systematically identifies potential areas for interactions. This identification can incorporate more formal techniques, such as the digraph method, or can be as informal as merely producing some sort of tabulation. The analyst probably would tend toward more formalization as the number and/or complexity of systems interactions increased. For a large-scale survey, it may be advantageous to utilize a computerized data base. To supplement the documentation study, the analyst can procure expert opinion, presumably from plant personnel.

A.1.2 Physical Survey

The physical survey is basically a "walk-through" inspection of the appropriate areas of the plant coupled with some sort of systematic accounting of identified areas for interaction. A typical example can be found in the Diablo Canyon seismic review.[1] Tabulation may be in a columnar format or possibly involve marking sensitive locations on diagrams of the plant layout. The survey should be thorough enough to identify potential interactions unique at the plant due to modifications not specified on schematics. However, it should not become encumbered with highly unlikely interactions. This latter criterion also applies to the operational survey. However, since functional interactions tend to be more clearly defined and less speculative than spatial ones, there is less possibility that the operational survey will become bogged down with trivial interactions than will the physical.

A.1.3 FMEA[2,3,4]

FMEA is a qualitative induction technique for identifying hazardous conditions and determining their importance. As commonly used in reliability and safety analyses, the FMEA identifies failure modes for the components of concern and traces their effects upon other components, sub-systems, and systems. Emphasis is placed on identifying the problems which result from hardware failure. Typically, a columnar format is employed in an FMEA, as shown in Table A.1. Specific entries for the columns include descriptions of the component, its failure modes, causes of failure, possible effects, and actions to reduce the failures and their consequences.

Although traditionally developed from a component level, a type of FMEA can be envisioned which would start at a system level to trace out interactions and their effects upon plant safety functions and, eventually, on plant safety itself. Such a modified FMEA is illustrated in Table A.2. Note that it can be designed to integrate with an operational and a physical survey.

A.1.4. Binary Matrices and Digraphs

The use of hierarchies to portray relationships among elements of complex systems is common in many fields, especially in the business and social sciences. The nature of SI and the complexity of nuclear power plants suggests that the concept of hierarchies could be a valuable part of a methodology for SI analysis.

The tools associated with the concept are the binary matrix and the directional graph, or digraph. The binary matrix contains information on the relationships between the elements of a system and the digraph is graphical presentation of the structure of the system. Formal procedures involving very elementary matrix operations are available to generate the digraph from the binary matrix.

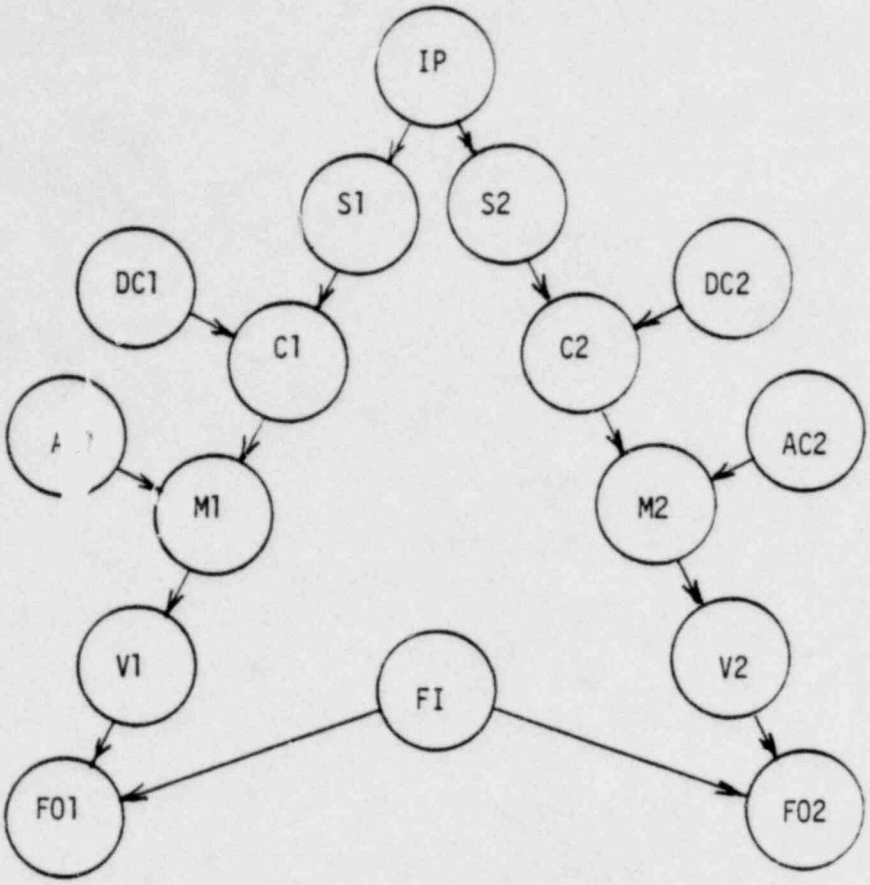
The relationships contained in the binary matrix are "subordination relations"; the binary entry in each intersection of the matrix indicates whether or not one element is subordinate to another. An important aspect of the indicated relationships is that they have an associated direction, i.e., given elements A and B, if A is subordinate to B, then B is not subordinate to A. The word "subordinate" should be interpreted broadly; for example, (1) the flow of fluid through a pipe is subordinate to (depends on) the position of a valve in the pipe, and (2) the output signal of an amplifier is subordinate to the operating state of the amplifier and to its input signal. In the application of the binary matrix to the analysis of complex systems, it is important to note that although the matrix must indicate all levels of subordination, the analyst need supply only direct first-level relationships and provide a computer code to deduce any consequent levels of subordination. An additional advantage is that the elements can appear in any order in the matrix; the matrix processing procedures are capable of rearranging the matrix into separate hierarchies.

Another feature of the binary matrix that makes it particularly attractive for SI analysis is that an element of the matrix can be any entity of interest; an entire system, a system function, a subsystem, a component, a physical location, a maintenance crew, or an electrical connection, to name a few of the possibilities. Elements of any level of detail can be intermixed.

The digraph (or digraphs, if the binary matrix represents more than one independent system) is generated directly from the binary matrix and provides a convenient graphical presentation of the ordered arrangement of the elements of the system. From the standpoint of SI analysis, potential interactions appear as linking elements between systems (subsystems, etc.). To determine whether such linkage represents valid SI requires further review because the

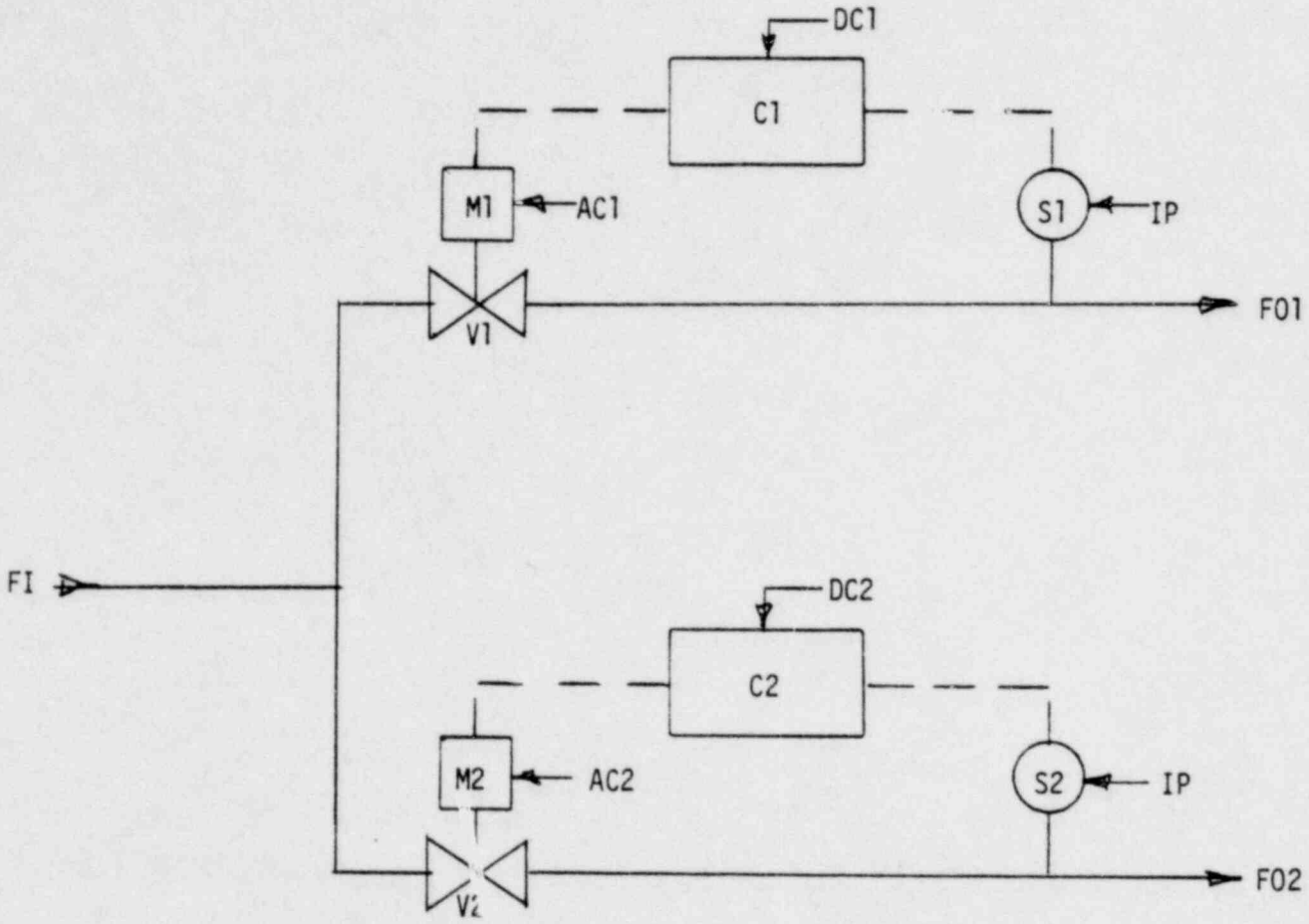
digraph shows only the direction of element associations, and not their nature. If more detailed analysis (fault tree analysis, for example) is to be performed, the digraph can be used as a guide and visual checklist in the processes of determining pertinent failure modes and establishing logical relationships between elements.

An example of the application of the binary matrix to two simple, linked flow systems is given below.



	FI	AC1	DC1	IP	AC2	CD2	V1	M1	C1	S1	V2	M2	C2	S2	FO1	FO2
FI	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AC1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DC1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AC2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DC2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V1	0	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0
M1	0	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0
C1	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0
S1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
V2	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0
M2	0	0	0	1	1	1	0	0	0	0	0	0	1	1	0	0
C2	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
S2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
FO1	1	1	1	1	0	0	1	1	1	1	0	0	0	0	0	0
FO2	1	0	0	1	1	1	0	0	0	0	1	1	1	1	0	0

Digraph and Binary Matrix for Flow Circuit
Showing Linkage Through IP



Sample Flow C cuit with Common Power (IP)

A.2. Quantitative Methods

Nine methods are discussed: fault trees, phased mission analysis, event trees, cause-consequence diagrams, GO methodology, Markov modelling, generic analysis, weighting factors, and Marshall-Olkin specialization. With the possible exception of certain weighting factor methods, the remainder tend to be rather formal techniques. Most possess qualitative capabilities also but, as was previously mentioned, they have been categorized as quantitative because they possess significant capability for such analysis.

A.2.1. Fault Trees[2,5,6,7,26]

Fault tree analysis is a deductive logic technique which diagrammatically models the various combination of basic failure events which contribute to some overall failure event. A fault tree begins at the TOP with the definition of this ultimate failure event, which is expanded downward through subsequent levels of contributing failures until the desired level of basic failure events has been reached. These contributory failures are combined by logical AND and OR gates at the appropriate levels. Fault trees are normally used to model events having binary failure states (total failure vs. total success), as opposed to those having partial failures. The symbols used in fault trees are shown in Figures A.1. and A.2.

The means by which the TOP event can occur are known as "cut sets", the combination of basic events leading to the TOP. Of particular importance, especially in evaluating failure probabilities associated with the TOP event, is the concept of a minimal cut set - one in which return of any one of the basic failure events to a success mode precludes the occurrence of the TOP event. By assigning probabilities to the basic failure events, the probability of the TOP event can be found as the Boolean sum of the probabilities for each of the minimal cut sets.

Fault trees are often used to model system failure in terms of failure of its basic components. Component malfunctions are divided into two types: failures and faults. Failures are malfunctions which require repair (or replacement) of the component to correct the malfunction. Faults are malfunctions that can be corrected without maintenance of the component in question. Repair refers to the reversal of a basic event state from failed to unfailed. Both failures and faults can be designated as primary or secondary. A primary malfunction is one in which the component itself is responsible (such as a switch sticking

closed). A secondary malfunction is one in which the component is not held accountable (such as a switch being welded closed). A special type of secondary fault is a "command" fault, in which the component immediately functions properly upon repair of the causes of the secondary fault. An illustrative fault tree for for system failure is shown in Figure A.3.

Although traditionally used to model system failures, fault trees can also be used to model accident sequences, where the TOP event becomes some consequence of those sequences. Usually, this involves combining several system fault trees which contribute to the overall consequence. When a consequence fault tree is constructed for each of the various consequences of the accident sequences, the complete analysis is equivalent to a complete event tree analysis (with conditional fault trees) covering all initiating events, or corresponding cause-consequence analysis. To illustrate a consequence fault tree, consider the operating sequence of engineered safety features following a small LOCA shown in Figure A.4. as a block diagram. The equivalent consequence fault tree for core damage is shown in Figure A.5.

Dependencies often exist among different components within a system. Failure of one component, such as a pump, may increase the load on another, thereby increasing its likelihood of failure. Or, two components, each requiring support from some other component or system, can fail simultaneously if that support fails. Such dependences can be incorporated directly onto a fault tree by further resolving the basic failures subject to a common failure into an independent component failure and the common failure.

Consider the fault tree for Core Spray (CS) failure in Fig. A.3. Suppose the pumps each receive electric power from the same power bus, whose failure is denoted as B in Figure A.6. (Note that this is not a representative case, but rather has been selected only for illustration.) Should this bus fail, both pumps will fail due to the common failure, thereby failing both loops and CS. Thus, the redundancy of the two loops has been circumvented. This is represented by creation of a new, single-event (B) minimal cut set derived from the fault tree by resolving former basic events P_1 and P_2 into independent pump failures P_1' and P_2' and a common failure B. Such a case would represent very poor design, because CS loop redundancy has been eliminated at the pump level, and is not characteristic of plant design. However, some dependencies may exist at more subtle and obscure levels and can go unaccounted for during system design.

A.2.2. Phased Mission Analysis[8]

Fault trees are not especially amenable to modelling failures in a time sequence. Compensating somewhat for this is phased mission analysis. As discussed in reference 8, a phased mission is a system task during the execution of which the system is altered such that the logic model changes at specific times. In performing an overall safety function, a system may have to operate in different modes as time progresses. The goal of phased mission analysis is to reduce the original multiphase mission into an equivalent single phase one. Overall mission failure, defined as a TOP event, is represented by a fault tree, whose individual branches correspond to different system logic in each phase. By performing various logical operations, this fault tree can be simplified into one for a single phase with a single logic structure.

Phased mission analysis is applicable to a multi-function system with nonrepairable components (at least over the time span of the overall mission). By manipulation of the minimal cut sets, the multiphase mission can be reduced to an equivalent single phase one. To illustrate this, consider the primary reactor coolant (PRC) system during an ascent from low to full power operation. During low-power operation, the heat generated is lower than during full-power operation. Thus, cooling requirements are less.

For illustration purposes, consider only the PRC pumps, assuming there is just a pair. During low power, only one of them is needed. However, during full power, both are necessary. Thus, two distinct operating phases for the same system exist, and the requirements change with time. The multiphase mission fault tree is shown in Figure A.7a. Note that there are three minimal cut sets, two single-element ones and one with two elements. Through procedures involving cut-set cancellation and component transformation, this multiphase mission can be reduced to a single phase one, as shown in Figure A.7b. Note that there are now four minimal cut sets, but each one contains only a single element. The total number of basic events (4) has remained the same, but the logic structure has been simplified and the basic elements directly reflect their phase-dependence. Time dependence has been incorporated within a simplified fault tree structure.

A.2.3. Event Trees

Event tree analysis is an inductive logic technique which sequentially models the progression of events, both success and failure, leading from some initiator to a series of logical outcomes. An event tree begins with some initiating failure, usually on a component level, and maps out a sequence of events, usually on the system level, to form a set of branches, each of which represents a specific accident sequence whose outcome, or consequence, corresponds directly to the events contained in the sequence. Like fault trees, event trees are normally used to model events having binary failure states, these events usually corresponding to total success or failure of a system.

Each accident sequence leading to a particular undesired consequence is somewhat analogous to a cut set on a fault tree. Whereas a cut set represents a combination of failures leading to the TOP event, an accident sequence represents a combination of sequential events (successes and/or failures) leading to a particular consequence. This suggests a possible equivalence between event trees and consequence fault trees, i.e. fault trees whose TOP events correspond to consequences of accident sequences. Complete event tree analysis requires identification of all possible and distinct initiating events and development of an event tree for each. There tends to be an extensive overlap of consequences among the various trees. Consequence fault tree analysis requires identification of all possible and distinct consequences and development of a fault tree for each. There tends to be an extensive overlap of initiating events among the various trees. The difference in reference points between event tree and consequence fault tree analysis seems to suggest that event trees are more appropriate when the initiating events are more readily identifiable, while consequence fault trees are more appropriate when the consequences can be identified more easily.

An event tree for the accident sequence depicted in Figure A.4. is shown in Fig. A.8. Note that the degree of core damage will vary from branch to branch, but this has been ignored for the sake of simplicity in illustration. Evaluation of the degree of core damage for each accident sequence would involve analysis of the physical phenomena taking place during each sequence.

Event trees, using system successes and failures as the basic events at the branching points, tend to view overall consequences to a limited degree of resolution, that being the system level. Fault trees, both those for system failures as well as for consequences, tend toward a greater degree of resolution, that being the component level. To obtain true equivalence between event trees and consequence fault trees, it is necessary to resolve the system failures on the event tree to their contributing component failures. The usual technique involves development of a system fault tree for each branching point, the events on this tree being conditional upon what has occurred earlier in the event tree sequence. The formal combination of event trees with conditional fault trees forms the basis of cause-consequence analysis and is examined in the next section.

It must be noted that, unless failure data is available on the system level, probabilistic analysis involving event trees usually necessitates resolution to the component level, where failure data may be more readily available. Due to the sequential nature of event trees, quantitative evaluation necessitates the use of conditional probabilities, those whose values reflect the occurrence or non-occurrence of preceding events. This can pose some computational difficulty when events are not independent.

A.2.4. Cause-Consequence Diagrams[9,10]

Cause-consequence analysis is a formalized combination of event tree and conditional fault tree analysis. The event tree is used to map out the sequence of events leading to the various consequences. The causes of these events, usually system failures, are modelled by conditional fault trees. Cause-consequence diagrams are basically event trees with the conditional fault trees directly attached to the branching points. The fault tree symbolism is the same, while the event tree symbolism is somewhat formalized (see Figure A.9). As with an event tree, cause-consequence diagrams begin with an initiating event except that now this event may be expanded into its contributory failures. The combination of event trees with conditional fault trees, although not formalized into cause-consequence diagrams, formed the basis of the Reactor Safety Study. For illustration, the event tree of Figure A.8. has been developed into a cause-consequence diagram in Figure A.10. Again, for simplicity, the degree of core damage has been excluded from the consequence descriptions.

As previously mentioned, a lack of failure data on the system level will usually necessitate resolution to the component level, where such data may be available, in performance of a quantitative assessment. The cause-consequence diagram has this capability. It also is better suited to identification of potential system dependencies on the component level than is the event tree alone. However, these dependencies must be shown on separate, conditional fault trees, while the consequence fault tree is capable of including all of them within a single logic structure. Nevertheless, no matter which of these methods is used, complete analysis requires many of the individual trees, one event tree, or cause-consequence diagram, for each initiating event, or one consequence fault tree for each accident consequence.

A.2.5. GO Methodology[11]

The GO methodology is a combined simulation and logic technique which models both hardware and logic operations on an overall flow chart. It is basically a success tree approach. (A success tree is analogous to a fault tree except that success rather than failure events comprise its makeup at all levels, including the TOP.) A GO flow chart consists of "events" linked by hardware and logic operators to form some overall sequence of operation. Each "event" corresponds to the occurrence of output from a GO operator and can occur in several states, each corresponding to an occurrence time for an output. Up to 128 states are possible, with 0 representing premature or spurious operation while the highest state represents a failure to operate (operation delayed over the entire mission time). As mentioned, the GO operators correspond to both hardware, such as electrical components, and logic gates. Each is normally represented by a circle whose included numeral represents the type of operator. Figure A.11. shows some of the more commonly used GO operators.

Being essentially a logic technique with additional capability to directly assimilate hardware operation, the GO methodology possesses the capabilities of fault and event trees plus the capacity to model time-dependency through the various event states. These event states may also be used to simulate partial failures, alleviating the limitation of binary failure states prevalent in fault and event tree analyses. Although the hardware-related GO operators are designed to model components, the GO methodology can be extended beyond system operation to functions, consisting of operation of various systems, by enlarging the overall GO flow chart. Whereas cause-consequence diagrams require two logic models, event and fault trees, to accomplish this functional modelling, a GO flow chart can include this within one basic logic structure. Note that consequence fault trees also possess this capability.

A.2.6. Markov Modelling[12,13,14,26]

Markov modelling is a mathematical inductive analysis procedure which reduces a system of many stochastic processes, effects, and paths to a single stochastic relationship characterized by a series of discrete time processes. As described in reference 26, Markov models are functions of two random variables - the state of the system, and the time of observation. Any Markov model is defined by a set of probabilities P_{ij} which define the probability of transition from any state i to any state j. Another important feature of any Markov model is that transition probability P_{ij} depends only on states i and j, and is completely independent of all past states except the last one, state i.

A Markov process can be specified by a set of differential equations and their associated initial conditions. Because of the basic Markov assumption that only the last state is involved in determining the probabilities, the analysis always yields a set of first-order differential equations. The constants in these equations can be specified by constructing a transition-probability matrix. The rows of the matrix represent the probability of being in any state i at time t, and the columns represent the probability of being in state j at time $t + \Delta t$. The former are called initial states and the latter final states. The transition probability P_{ij} is the probability that in time Δt, the system will undergo a transition from initial state i to final state j. Each P_{ii} term, on the main diagonal, is the probability that the system will remain in the same state during one transition. The sum of the P_{ij} terms in any row must be unity, since this is the sum of all possible transition probabilities. The probability that the system will be in a state i at time t is denoted by $P_i(t)$.

To illustrate Markov modelling, consider a system comprised of two components, A and B, which have binary states (total success or total failure). These could be the two PRC pumps used in the illustration of phased mission analysis in section A.2.2. As shown in Figure A.13., four system states are possible (both components operable or inoperable, or either inoperable while the other is operable). The arrows indicate the allowed transitions between states. (Note that the components have been assumed to be nonrepairable.) λ_1 and λ_2 represent the independent failure rates of components A and B respectively. λ_c represents the failure rate of both components together. Whether or not each state represents a success or failure state of the overall system depends upon the overall system logic, which must be determined external to the Markov model.

State S_1 clearly represents a success state for the system while S_4 represents a failed state. With respect to the PRC system used to illustrate phased mission analysis (see Figure A.7a), states S_2 and S_3 represent success states during low-power operation. However, during full-power operation, they represent failed states for the system.

Markov models can be resolved to either the component or system level. When the overall states correspond to system states, the specific transitions involve changes in individual component states leading potentially to changes in system states. Similarly, transitions involving changes in individual system states potentially lead to changes in overall function states. The states dealt with in Markov models are usually binary, although the potential exists for some partial failure analysis. Transitions between states could involve individual changes from success to partially-failed modes. By its very nature, Markov modelling involves time-dependency. Time-varying probabilities can be modelled through the transition-probability matrices linking various states.

Markov modelling has the potential to quantitatively account for multiple failures due to a single common cause. Consider the example in Figure A.13. The transition from S_1 to S_4 results from dual failure of both components due to a single event, as reflected by the failure rate λ_c. If the components are the two PRC pumps, λ_c could represent failure of both due to a common event, such as loss of electric power. The Markov model will not identify the common cause but can provide a convenient medium for its probabilistic representation.

A.2.7. Generic Analysis

Generic analysis involves reviewing the minimal cut sets from a fault tree or similar analysis for dependencies among the basic failure events using a standard checklist of potential linking characteristics. Subsequently, the results can be used to identify new modes of overall failure by Boolean transformation of the minimal cut sets to accommodate these dependencies. Although a major portion of this technique is qualitativ., it has been included among the quantitative methods because it follows an analysis procedure such as fault trees rather than preceding it, as the other qualitative methods tend to do. Also, the Boolean transformation possesses quantitative capabilities.

Generic analysis is usually performed on the component level, as reflected by the standard checklists for dependencies. Starting from a list of basic

events from minimal cut sets, the analyst identifies common linkages among these events based on some standard checklist. One such checklist[17] identifies four major generic cause categories:

1. Mechanical/Thermal
2. Electrical/Radiation
3. Chemical/Miscellaneous
4. Other common links

These are detailed in Tables A.3 - A.6.

Sandia[18] uses another checklist, consisting of three categories:

1. Physical - electrical, mechanical, hydraulic
2. Spatial- propagation of an adverse environment through a common spatial medium
3. Inherent - common manufacturer, similar technology, equal age/wear, identical or similar components

The two checklists overlap almost totally and are representative of the types of dependencies requiring identification.

A convenient technique of for cataloguing dependencies involves overlaying domains for the generic causes on a plant floor plan. This technique is especially amenable for computer codes, such as BACFIRE.[19] As described in reference 26, given a specific generic cause, an analyst can examine a building floor plan and identify each area of the building where a single occurrence of that generic cause could affect all building components. This area is called a common location. Thus, a common location requires an area and the potential occurrence of a specific generic cause. The domain of a specific generic cause is the set of all common locations involving that generic cause. Most buildings contain barriers such as walls, floors, and cabinets. An oil spill can generally be confined to the room in which the spill occurred. Vibration from a large compressor, on the other hand, could affect every room in the building. Acid vapors can become distributed throughout several rooms by the air conditioning system. Most secondary causes have a distinct domain because boundaries containing the effects of one cause often do not contain the effects of another.

The dependencies identified for the basic events can be attached to the fault tree, as discussed in section A.2.1. and shown in Figure A.6., or incorporated into the minimal cut sets by means of a Boolean transformation of the variables for the basic events. In essence, these two techniques are equivalent, since the final

goal is a listing of the "new" minimal cut sets, i.e., all the sets including not only independent component failures but also failures due to commonalities. For illustration, consider again the CS system whose fault tree is shown in Figure A.3. Suppose all the components have common actuation (failure of which is denoted by A), while each pair of valves and each pair of pumps receives power from a common electrical bus (failures of which are denoted by B_1 and B_2 respectively), as indicated in Table A.7. The basic events are transformed as indicated into independent failures and failures due to the commonalities. (Note that this is analogous to attaching the common failure to the fault tree, as shown in Figure A.6.). The transformed variables are substituted into the minimal cut sets to yield "new" cut sets, not necessarily minimal. Finally, these are summed in a Boolean expression for the TOP event (CS failure) to yield the "new" minimal cut sets. In the example, these "new" sets consist of three single-element ones for the commonalities and four dual-element ones for the independent component failures. This is the method advocated by Sandia,[18] who utilize the SETS[20] computer code to facilitate the Boolean algebra. Probabilistic analysis may then proceed from these "new" minimal cut sets in the same procedure as with any minimal cut sets from a fault tree or similar analysis.

A.2.8. Weighting Factors[21,22,26]

Weighting factors can be used to mathematically adjust independent failure probabilities for the presence of some common failure event. Unlike the generic approach, the emphasis is not on identifying the commonalities, although this is necessary to some degree, but rather on obtaining a quantitative estimate of the degree of dependency between two failure events. The most basic approach is to multiply the product of independent failure probabilities by a factor α (≥ 1) to obtain an estimate of the "true" failure probability, i.e. after commonalities have been accounted for. The amount by which α exceeds unity reflects the degree of dependence between the two events.

For example, the probability that both CS valves fail (from Figure A.3.) is greater than the product of their independent failure probabilities if some commonality exists between them. Using Table A.7. for illustration, the joint failure probability for both valves may be written as:

$$P(V_1 \wedge V_2) = P(V_1)P(V_2) > P(V_1')P(V_2')$$

because:

$$P(V_1) = P(V_1' + A + B_1) > P(V_1')$$

$$P(V_2) = P(V_2' + A + B_1) > P(V_2')$$

therefore: $P(V_1 \wedge V_2) = P(V_1')P(V_2')\alpha$

where: $\alpha > 1$

The value of α must be determined by the analyst. This is the key to accurate representation of dependencies using this weighting scheme. His choice of method for evaluating α will depend upon the qualitative and quantitative information available to him. He may use a fault tree - generic analysis approach if he has sufficient detail or may merely make a subjective estimate of α based on expert opinion.

While the α-factor method is general enough to be applied at the system as well as the component level, a somewhat more specific approach is particularly appropriate on the component level. Two types of dependencies are identified:

1. Multiple failures attributable to a single cause
2. Subsequent failures resulting from preceding ones

For example, two pumps, each of 50% capacity during normal operation but capable of 100% for a limited time during emergency operation, are powered from the same electrical bus. Failure of that bus will fail both pumps--multiple failures due to a single cause. If one pump fails independently, the other must operate at the increased load. If forced to do so beyond a certain time period, it too could fail--a subsequent failure resulting from a preceding one.

As discussed in reference 26, when multiple component failures can be traced to a single event, such as an external event or the design of the system itself, the fraction of the component failures is represented by β. The use of the β-fraction is illustrated by Figures A.14. and A.15. Figure A.14. is a success block diagram for a one-out-of-two system, where r denotes component reliability. The failure rate λ in Figure A.14. is assumed to be constant, a consequence of the simple assumption that equipment failure is random and therefore governed by the exponential distribution.

The failure rate λ can be divided into two mutually exclusive elements: independent failure (with failure rate λ_1) and common-cause failure (with failure rate λ_2). Thus:

$$\lambda = \lambda_1 + \lambda_2$$

The fraction of common-cause failures (β) is defined as:

$$\beta = \frac{\lambda_2}{\lambda}$$

where:

β = the conditional probability that a common-cause failure occurs, given that an equipment failure has occurred.

Figure A.15. depicts independent failure and common-cause failure as three independent "components." Implicit in Figure A.15. is that, when a common-cause failure occurs, all redundant units are failed with probability one. This is the extreme case of common-cause failure with complete coupling between the random variables representing time to failure for each redundant unit. Any error due to this assumption will lead to a pessimistic reliability prediction in contrast to the optimistic predictions associated with the assumption of independent failures.

The second type of dependency is causal failure in which an equipment failure originates independently, but propagates, resulting in additional equipment failures. It is important to consider causal failures as originating only from independent failures and not from common-cause ones. Although a common-cause failure could conceivably damage additional equipment, system failure has already occurred and care must be taken to avoid double accounting of system failure modes. A category for causal failures is formed by leaving the definition of common-cause failures the same, and breaking up independent failures into two subcategories:

1. Isolated = a failure that is completely independent and does not propagate into additional failures (failure rate = λ_{1a})
2. Causal = a failure that originates as an independent failure but propagates, resulting in additional failures (failure rate = λ_{1b})

As in the previous case:

Common-Cause = an occurrence of multiple failures, where the failures are caused by a single common event (failure rate = λ_2)

The fraction of causal failures is represented by γ and defined as follows:

$$\gamma = \frac{\lambda_{1b}}{\lambda_{1a} + \lambda_{1b}}$$

where

γ = the probability that a unit will initiate a causal failure, given that it has failed, and given that the failure is not common-cause.

The β-factor method can be extended to the system level to treat intersystem dependencies. If two systems, with independent failure rates λ_{i1} and λ_{i2}, have a dependency, with failure rate λ_d, their overall failure rates (λ_{S1} and λ_{S2}) may be written as:

$$\lambda_{S1} = \lambda_{i1} + \lambda_d$$

$$\lambda_{S2} = \lambda_{i2} + \lambda_d$$

The intersystem β's become:

$$\beta_{S1} = \lambda_d/\lambda_{S1}$$

$$\beta_{S2} = \lambda_d/\lambda_{S2}$$

The β factor accounts for a large class of failure causes without explicitly identifying them.

As with the α-factor method, the β-γ-factor method also requires that the analyst determine β and γ. However, because the mathematical formulation in this method is more structured than in the α-factor method, less subjectivity need be used in the case where appropriate failure data is available.

A.2.9. Marshall-Olkin Specialization[23,24,26]

Marshall-Olkin specialization is a mathematical technique for adjusting a multiple failure rate for some dependency among the failure events. It is based on the Marshall-Olkin multivariate exponential distribution and has been developed for the component level. For illustration, consider a three-component system, discussed in reference 26. If a shock hits the system, seven ways exist for the components to fail:

(1), (2), (3), (1,2), (1,3), (2,3), or (1,2,3).

The failure of a single component represents independent failure, while failure of two or more components due to the shock represents failure due to a common cause. Each set can have its own failure rate and is assumed to be independent of the others.

Let $\underline{x}$ denote the vector, or set, of component failures, of which there are seven distinct ones, each corresponding to one of the failure groupings previously identified. The Marshall-Olkin model is specialized by assuming the $\lambda_{\underline{x}}$, the failure rate associated with the cause producing $\underline{x}$, depends only on the number of components failed. Therefore $\lambda_{\underline{x}} = \lambda_x$ where x is the total number of components failed by the cause. The assumption $\lambda_{\underline{x}} = \lambda_x$, x = 1, . . . m, implies that the components in the population are similar and are subject to similar failure causes. This specialized model is referred to as the homogeneous Marshall-Olkin model, in which common-cause failures are most likely to occur.

Within the homogeneous model, the common-cause failure rates may be independent of the failure numbers,

$$\lambda_x = \lambda, \; x \geq x_1$$

where the equality is only assumed for numbers of failures greater than or equal to some value x_1. This is referred to as the constant-rate case. The constant-rate case allows simple evaluations to be performed. The restriction upon it is the assumption that $\lambda_x = \lambda$, which involves engineering and failure cause considerations.

When the constant-rate case does not seem applicable, then another special case within the homogeneous model can be considered - the binomial-rate case. Here, the equation for λ_x is obtained by factoring the common-cause failure rate into an overall occurrence rate and a detailed effect probability. It assumes that, given a common-cause failure occurrence, each component has a constant probability of failing from the common cause. The binomial-rate case is more involved than the constant-rate case. The analyst must evaluate each component's probability of common-cause failure, unnecessary in the constant-rate case. However, it is more widely adaptable. Note that the constant-rate case is a special case within the binomial-rate model. The analyst must make the choice between the two alternatives.

To illustrate the potential applicability of the Marshall-Olkin specialization, consider an arrangement of three sensors, any two of which must provide a signal to activate an alarm. If the sensors are of similar design and are exposed to the same environment, one may make the assumption that the common-cause failure rates depend only on the number of failed sensors, not the specific ones. This forms the basis of the homogeneous model. Generally, the sensors will be subject to the same common failures, although small design or environmental variations may alter the failure thresholds from sensor to sensor. Thus, each would fail at a different rate due to common-cause, a situation for which the binomial-rate case is appropriate. If the sensors are identical in design, probably from the same manufacturer, and are exposed equally to the environment, each would have the same failure tendency due to common-cause. Thus, the common-cause failure rates would be the same whether two or three sensors fail, a situation for which the constant-rate case is appropriate.

Table A.1[25] Sample FMEA For Components

PART ASSEMBLY OR PROCESS				P/N					PREPARED BY A. C. Pericchio	DATE 2/25/76	
ITEM NO	PART, ASSEMBLY OR PROCESS NUMBER	PART, ASSEMBLY OR PROCESS NAME	PART, ASSEMBLY OR PROCESS FUNCTION	FAILURE MODE(S)	FAILURE CAUSE(S)	PROBABILITY	CRITICALITY	FAILURE EFFECT(S)	CORRECTIVE ACTION OR PREVENTIVE ACTION	PART, ASSEMBLY OR PROCESS INTERFACES AND REMARKS
1.1	Reactor Vessel		Provides support for fuel assemblies Contains sodium	Leakage, rupture	Corrosion, erosion, thermal shock, excessive loads			Sodium fills guard vessel. Decay heat removal possible.		Failure not critical unless there is a coincident failure of guard vessel.
2.1	Shutdown Heat Removal System Piping, Valving, and Components		Includes piping, drainvalves vent valves manual isolation valves	External leakage, rupture (represents total unavailability of a SHRS loop due to leakage of all components)	Bending fatigue, creep strain, thermal or mechanical loads, weld failure, thermal shock, thermal stresses, and corrosion			Sodium spills into cell. Loop drained and repaired; unavailable for decay heat removal		If the system is at negative pressure at the point of leak, gas enters the coolant.
2.2	EM Pump			Fails to operate	Loss of electrical power supply. Short circuit in MG set windings fail. Loss of cooling to windings. Structural failure of MG set.			Loss of forced convection in SHRS loop. Heat transport by natural circulation.	Redundant EM pumps would protect against random independent failures of the pump	Option 2, 3, and 4 would possibly not naturally circulate if there was pony motor flow. ΔP would close checkvalve.
2.3	SHRS Checkvalve		Prevent reverse flow in loop during normal operation	Fails to open	Contamination, mechanical distortion			Prevents flow in SHRS loop. No decay heat removal capability.	Positive pressure from pump head will very often open stuck valve, thereby reducing true failure rate.	
3.1	PHTS Piping Valving, and Components			Leak, rupture	Bending fatigue, creep strain, thermal or mechanical loads, weld failure, corrosion, nozzle weld failures.			Sodium spills into PHTS cell. Loop drained and repaired.		Loop unable to provide pony motor flow through core.
	PHTS Pump			Seal leakage, failure	Thermal cycling, fatigue, wear, corrosion			Sodium leaks into cell. Pump drained and repaired.		Loop unavailable to provide pony motor flow through core.
				Bearing seizure	Wear, corrosion, contamination			No forced convection in PHTS loop.		Loop unavailable to provide pony motor flow through core.
				Shaft failure, structural failure of pump internals	Thermal shock; excessive loading.			No forced convection in PHTS loop.		Precludes pony motor operation.

TABLE A.2 SAMPLE MODIFIED FMEA FOR SYSTEMS INTERACTIONS

Systems Interaction	Interaction Type	Plant Operating Mode	Systems' Failure Modes	Consequences		
				System Level	Function Level	Plant Level
APR & LP-ECC	Operational	Scrammed due to small LOCA	Given HPCI failure, APR failure to depressurize vessel prevents operation of LP-ECC	LP-ECC inoperable, although available	Failure to maintain vessel inventory	Possible core damage, leading to potential breach of containment
LPCI & RHR	Operational	Scrammed due to small LOCA	Failure of both LPCI, also used by RHR, leaves both systems inoperable	RHR inoperable	Failure to remove decay heat from containment	Possible containment overpressure, unless vented
General non-safety system & SC	Physical	Cold Sheetdown	Fire in cables of non-safety system spreads to nearby, non-redundant cables of SC	SC inoperable	Failure to lower primary coolant temperature to < 212°F	None, if plant can be returned to Hot Sheetdown & maintained there

APR = Automatic Pressure Relief

LP-ECC = Low Pressure Emergency Core Cooling

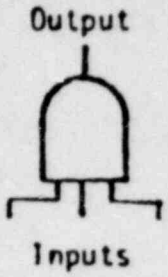
LPCI = Low Pressure Coolant Injection

RHR = Residual Heat Removal

SC = Sheetdown Cooling

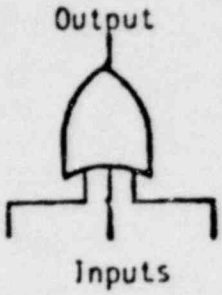
HPCI = High Pressure Coolant Injection

DRAFT



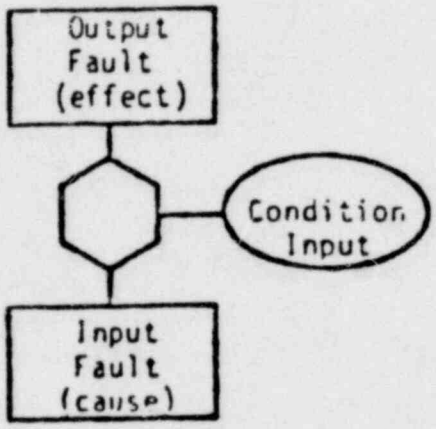
AND Gates

Coexistence of all inputs required to produce output.



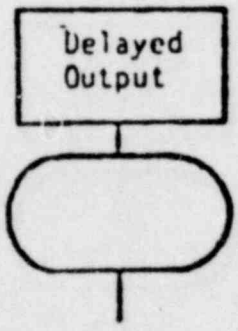
OR Gates

Output will exist if at least one input is present.



INHIBIT Gates

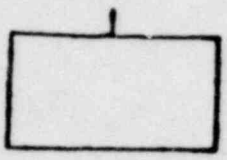
Input produces output directly when conditional input is satisfied.



DELAY Gates

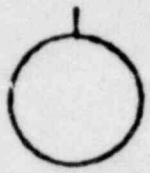
Output occurs after specified delay time has elapsed.

Figure A.1[10] Fault Tree Logic Symbols



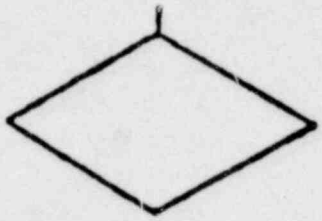
RECTANGLE

A Fault Event resulting from the combination of more basic faults acting through logic gates.



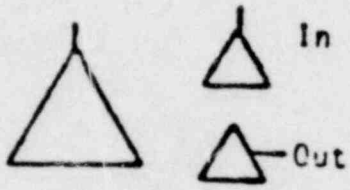
CIRCLE

A basic component fault - an independent event.



DIAMOND

A Fault Event not developed to its cause.



TRIANGLE

A connecting or transfer symbol.

HOUSE

An event that is normally expected to occur or to never occur. Also useful as a "trigger event" for logic structure change within the fault tree.

Figure A.2[10] Fault Tree Event Symbols

MINIMAL CUT SETS

$\{V_1, V_2\}$
$\{V_1, P_2\}$
$\{P_1, V_2\}$
$\{P_1, P_2\}$

Note: The CS system has been assumed to consist of two redundant loops each with full capacity. Thus, only failure of both constitutes failure of CS.

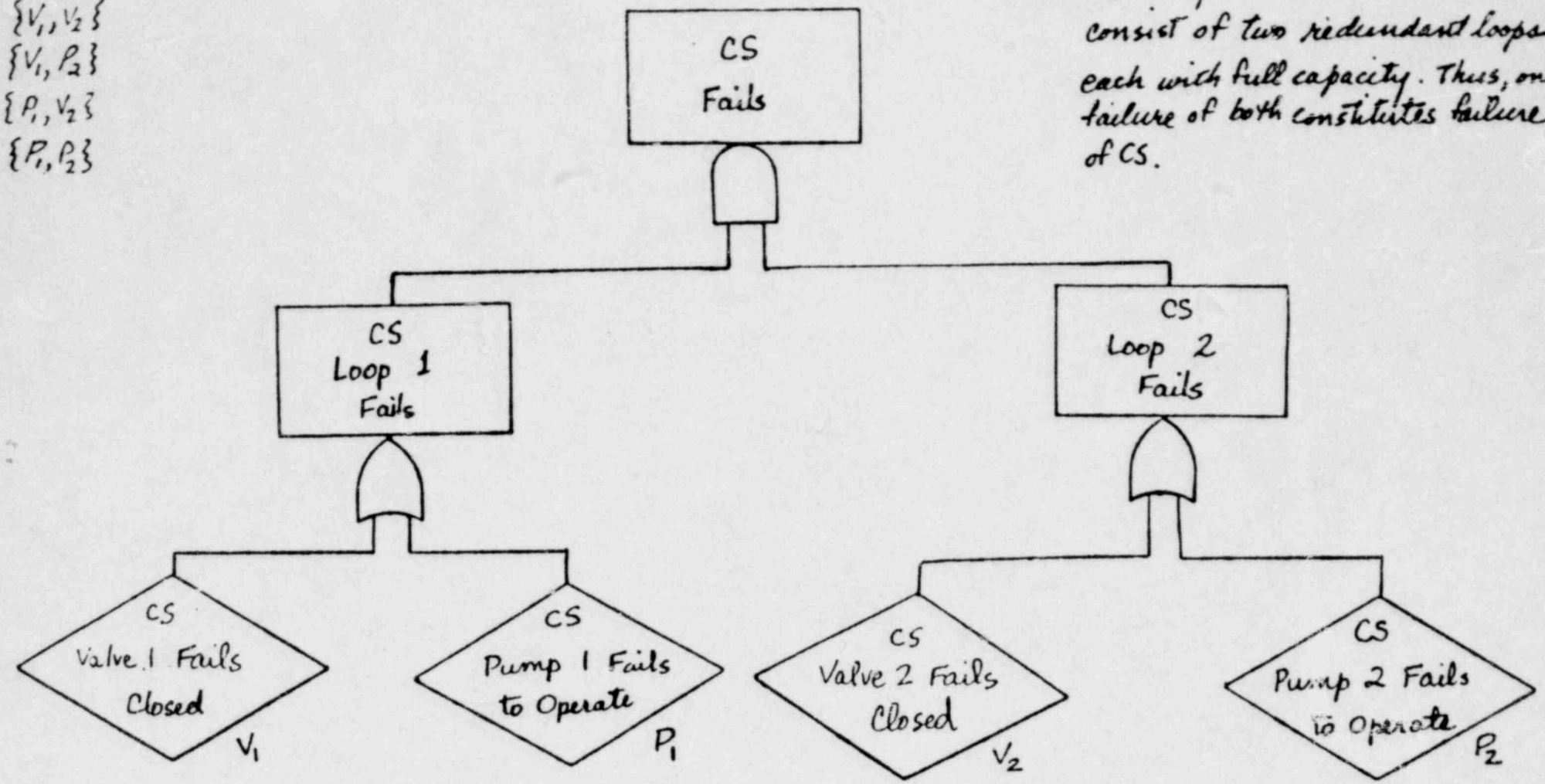
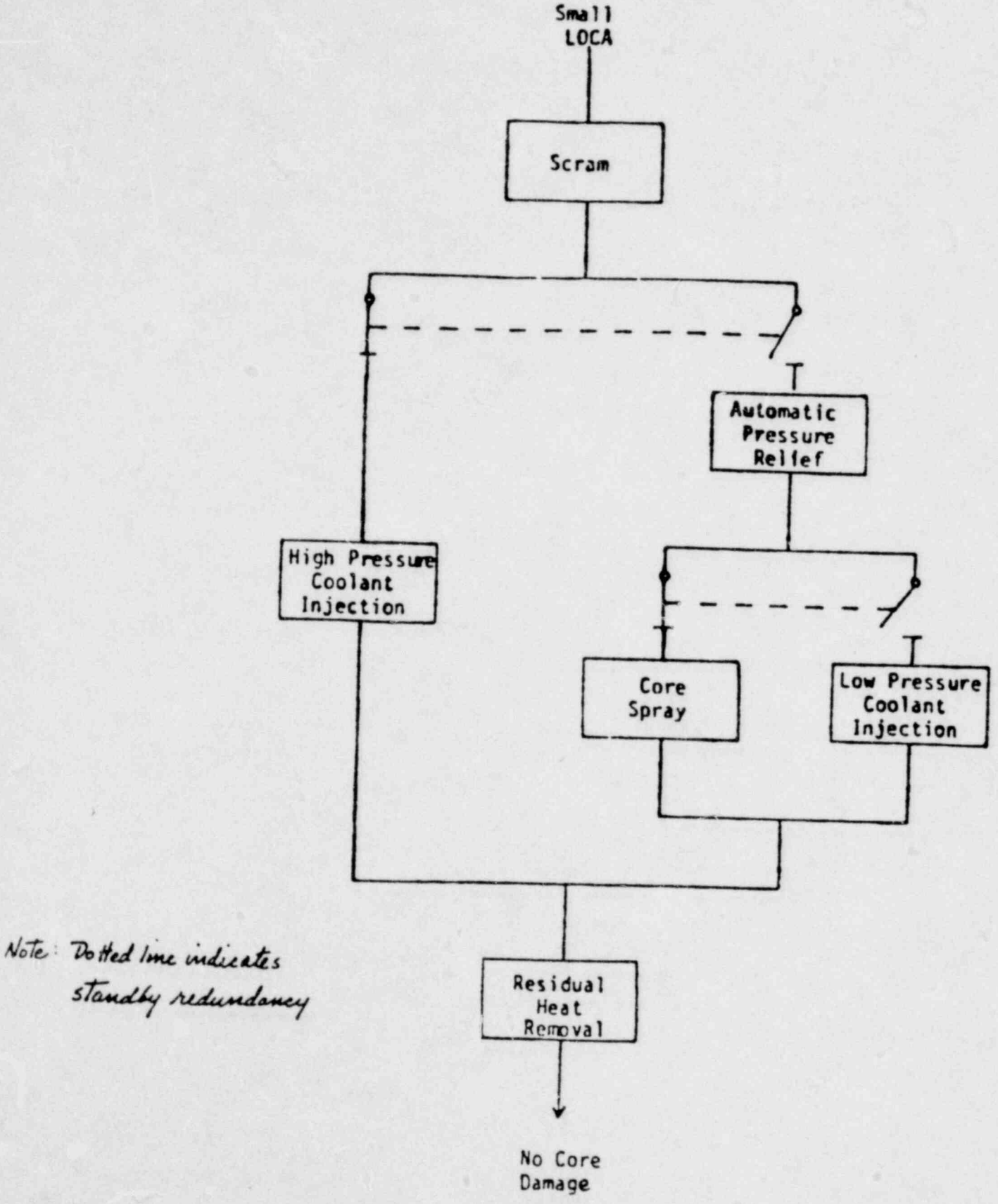


Figure A.3 Sample Fault Tree for Core Spray System Failure



Note: Dotted line indicates standby redundancy

Figure A.4 Block Diagram of Engineered Safety Features' Operation Following Small LOCA

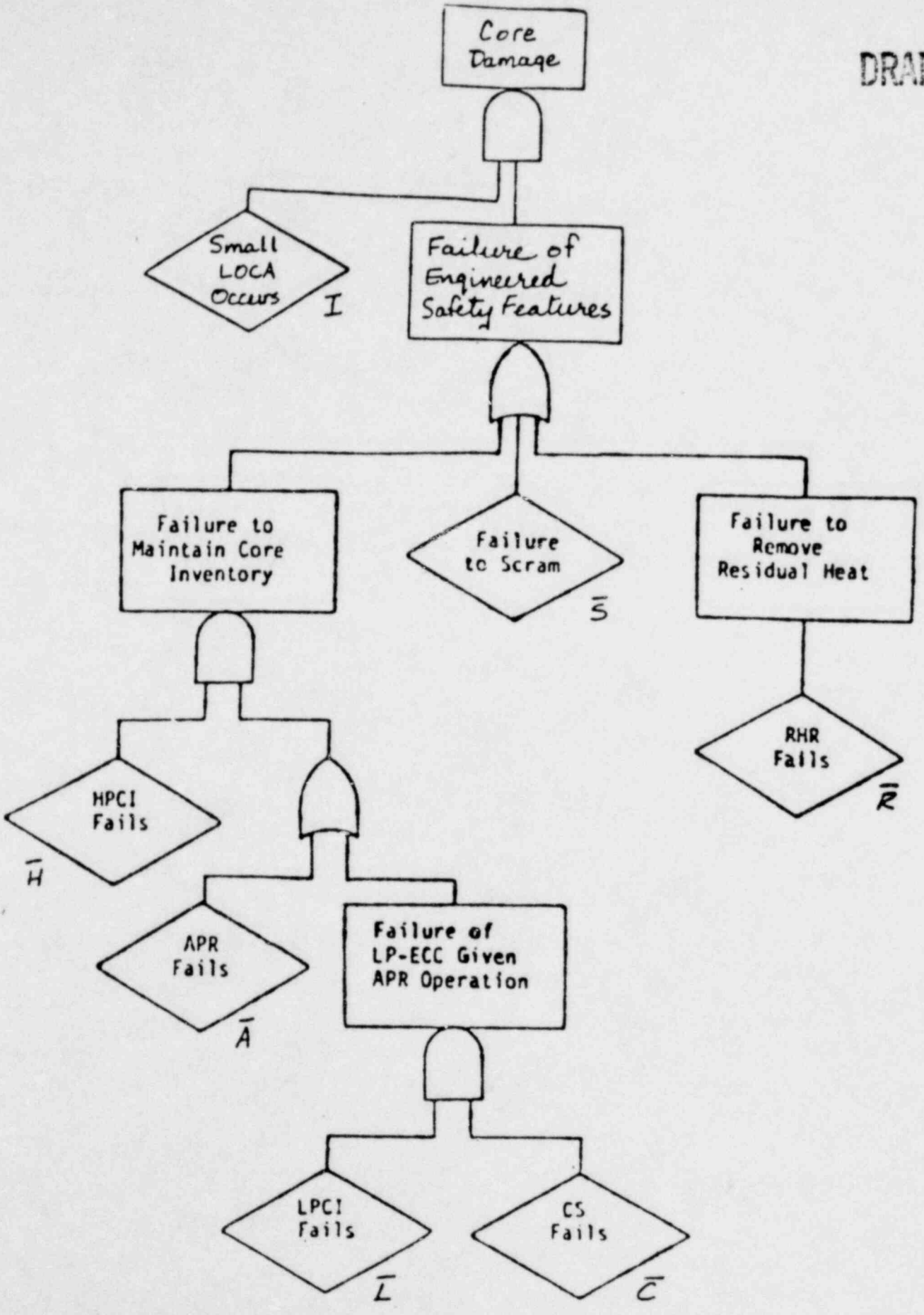


Figure A.5 Consequence Fault Tree for Core Damage due to Small LOCA Accident from Figure A.4

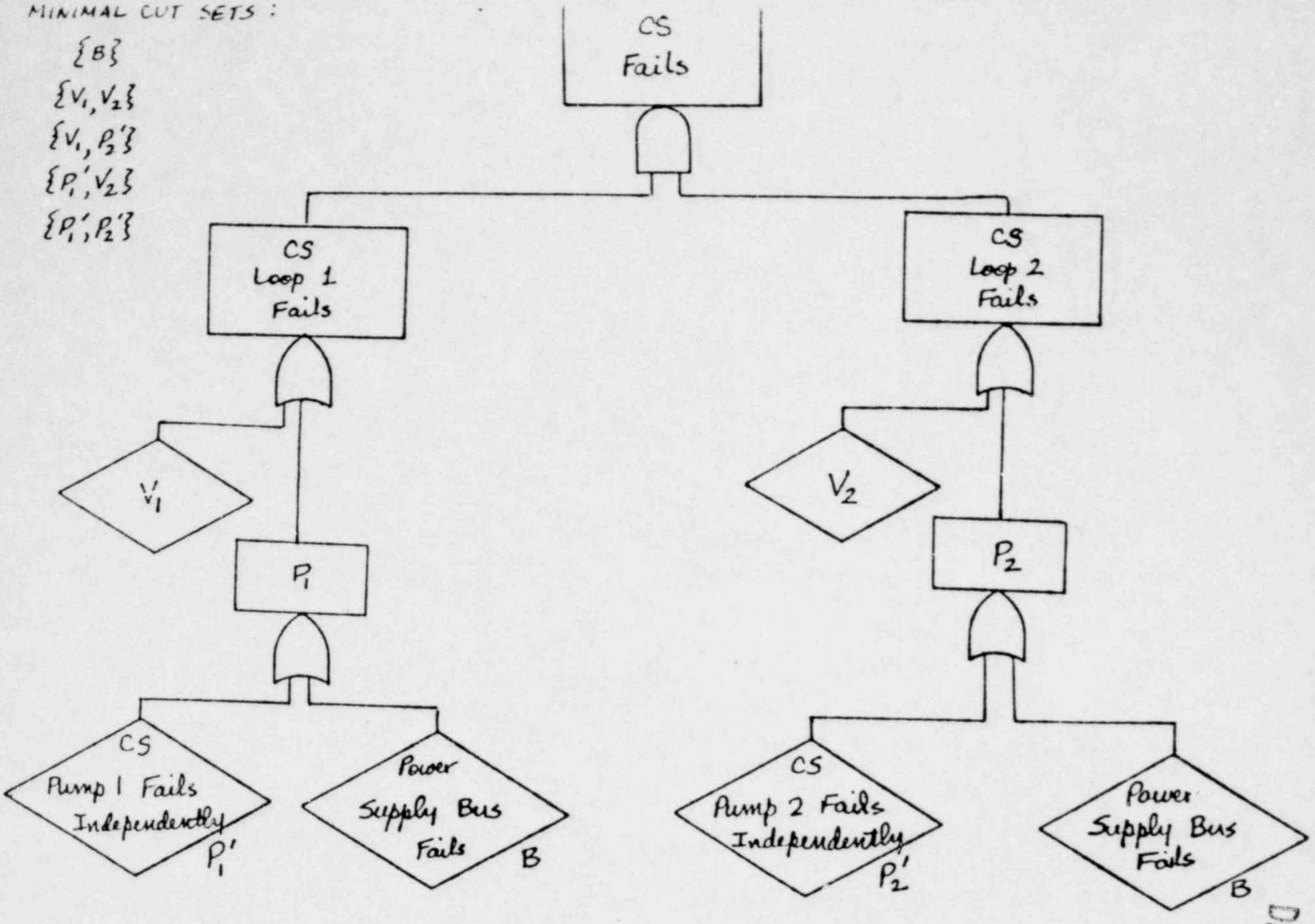


Figure A.6 Sample Fault Tree for Core Spray System Failure with Common-Cause Failure

Minimal Cut Sets:

Phase 1 : {A, B}

Phase 2: {A}, {B}

DRAFT

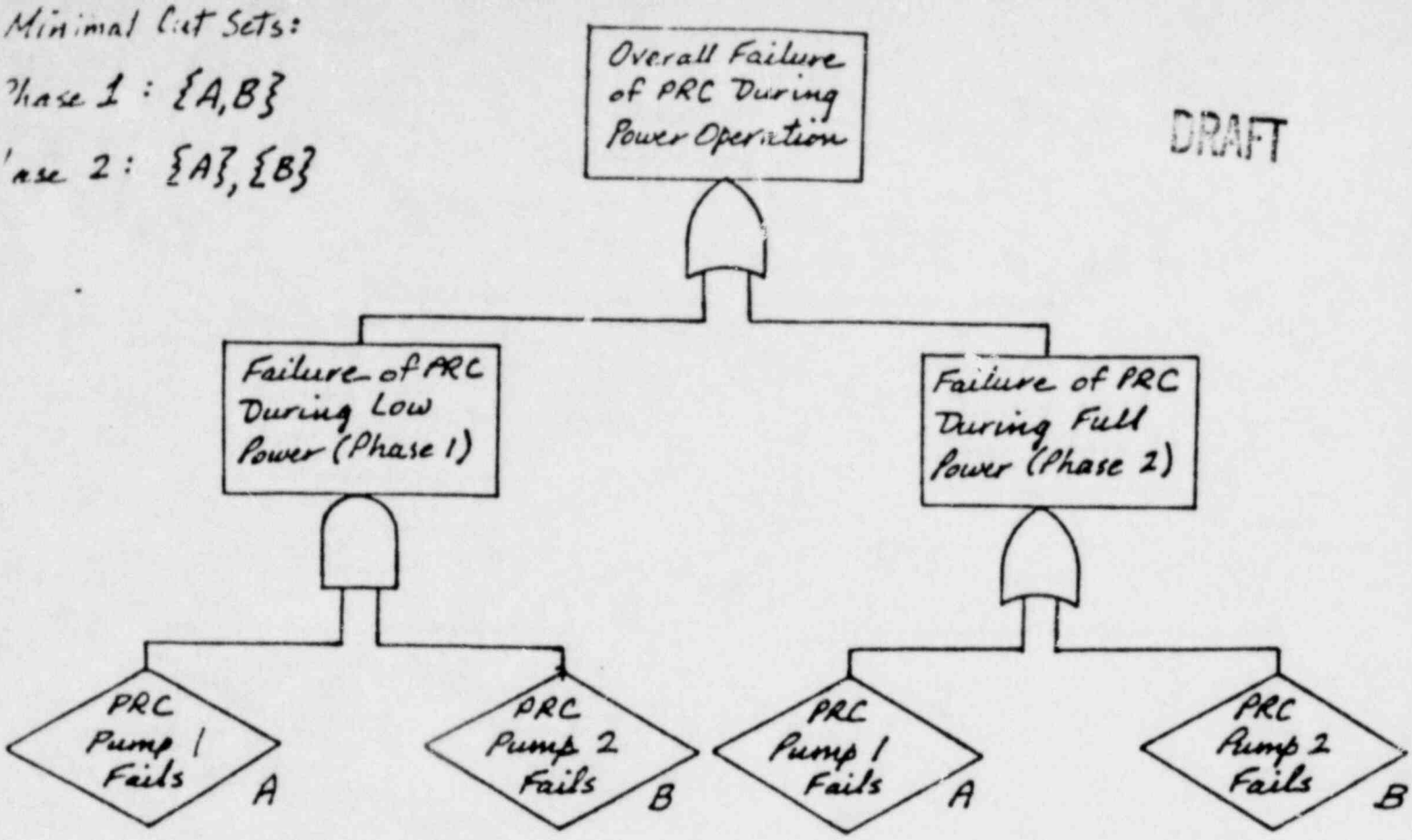


Figure A.7a Overall Failure of PRC During Power Operation as Multiphase Mission

Minimal Cut Sets:

{A}, {A_2}, {B_1}, {B_2}

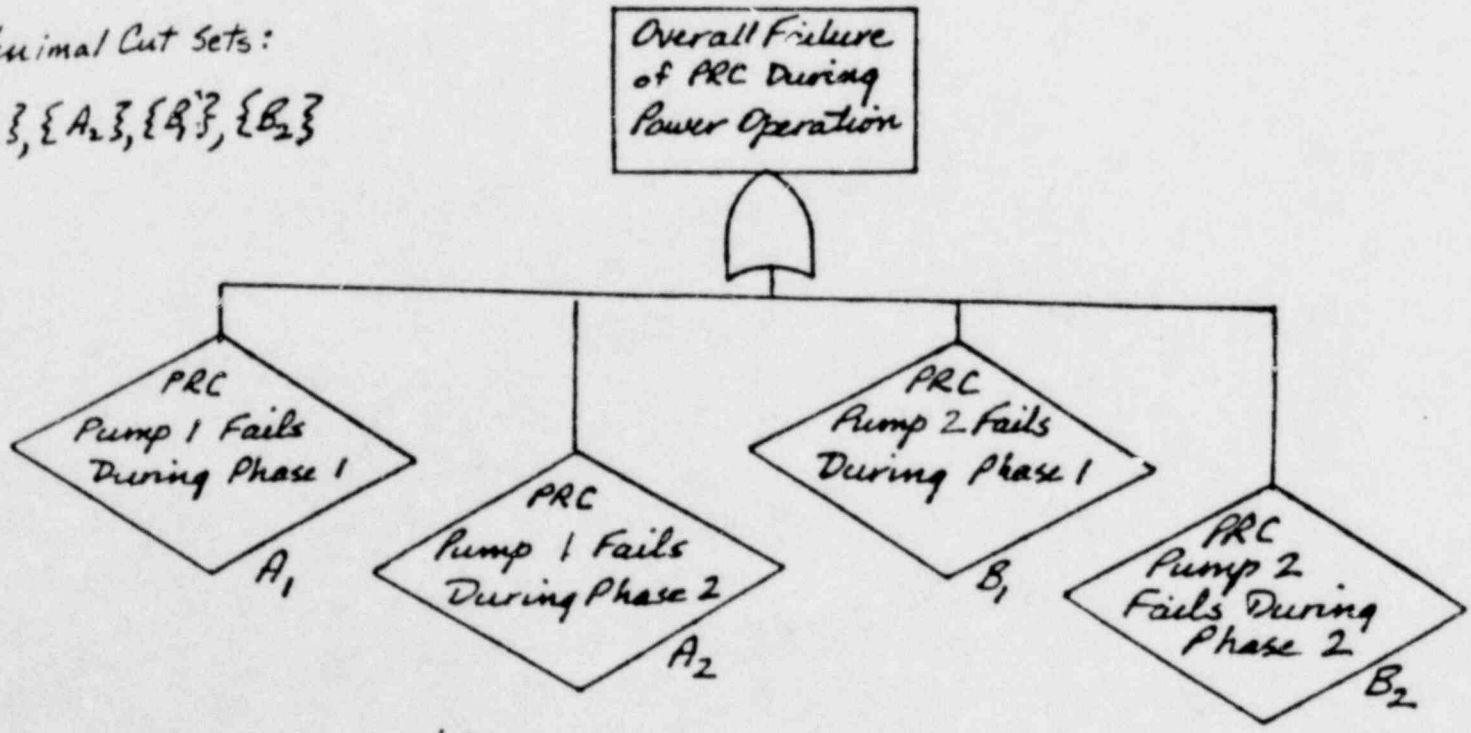


Figure A.7b Overall Failure of PRC During Power Operation as Single Phase Mission

DRAFT

Small LOCA	Scram	HP - ECC		LP - ECC		RHR	Core Damage ?
		HPCI	APR	CS	LPCI		

Note: At each branching point, upper branch denotes success, lower branch failure

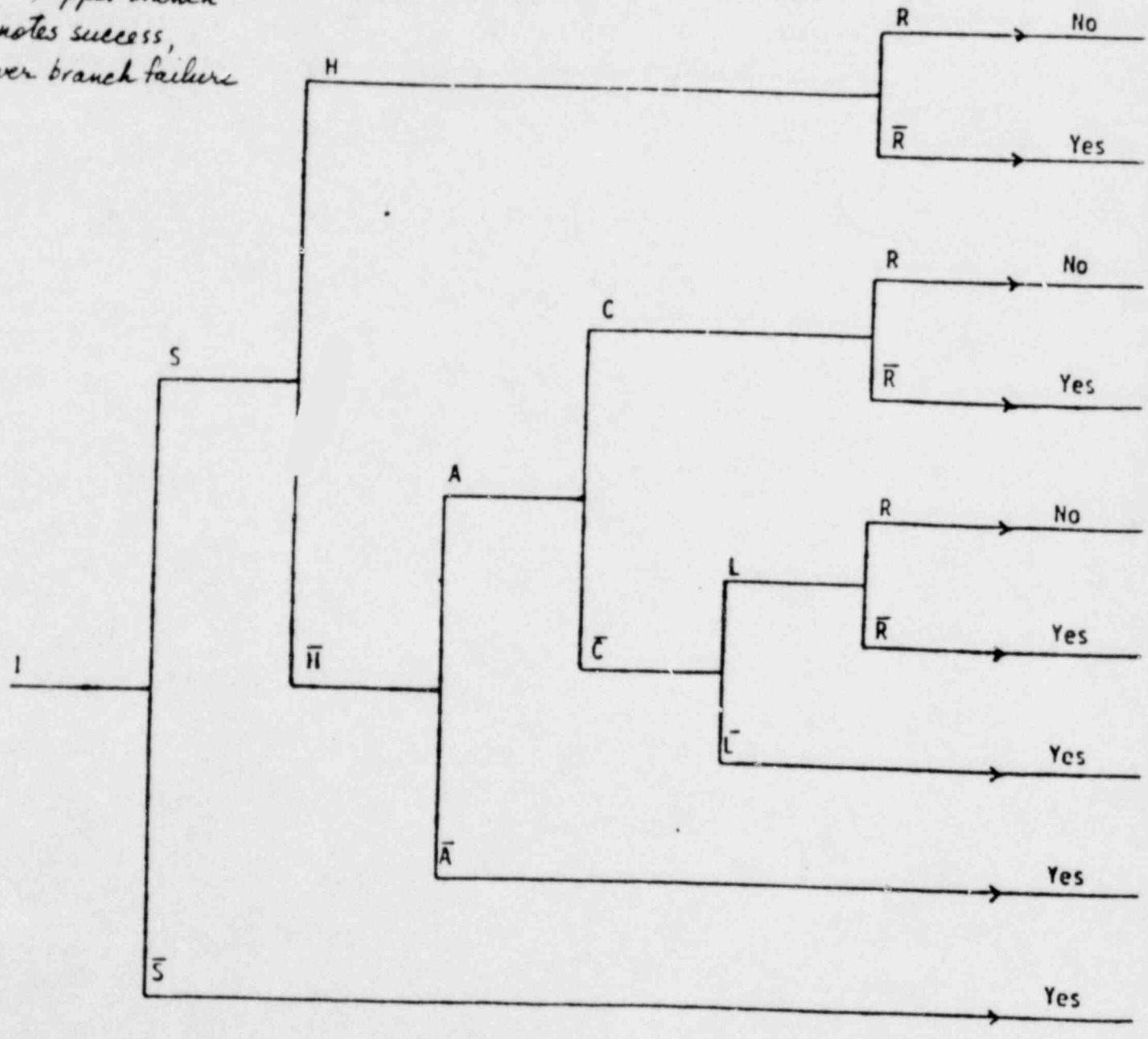
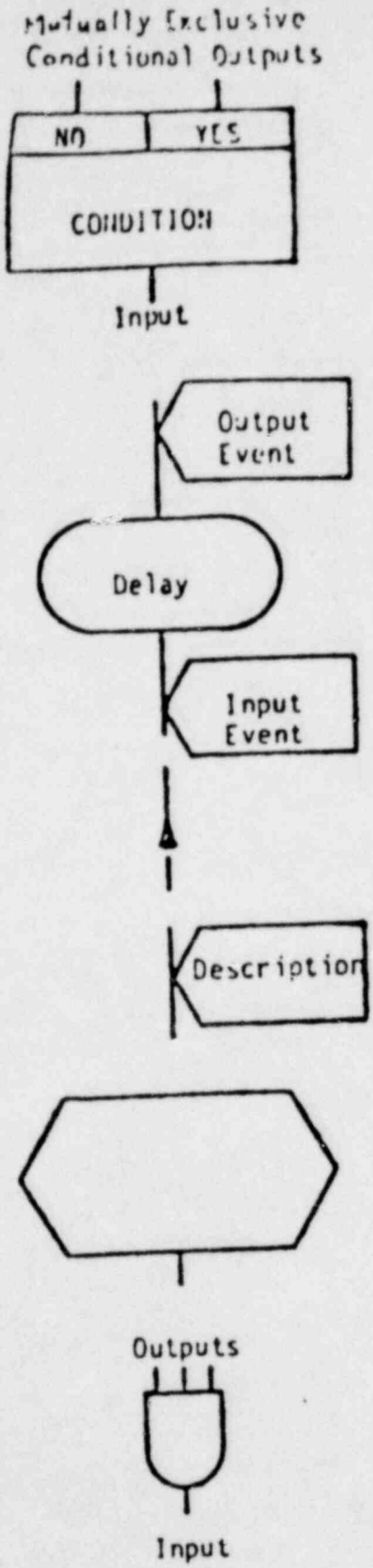


Figure A.6 Event Tree for Small LOCA Accident from Figure A.4

DRAFT



BRANCHING OPERATOR

Output is "yes" if condition is met; "no" otherwise.

DELAY OPERATOR

Indicates the amount of time delay required for output event to result from the input event.

DIRECTOR

Indicates the direction of event flow.

EVENT DESCRIPTOR

Describes the event present at specified position in chart.

CONSEQUENCE DESCRIPTOR

Describes the consequence. A terminal Symbol.

Inverse AND Gate

All outputs occur if the input occurs.

Figure A.9[10] Symbols for Event Tree Segment of Cause-Consequence Diagram

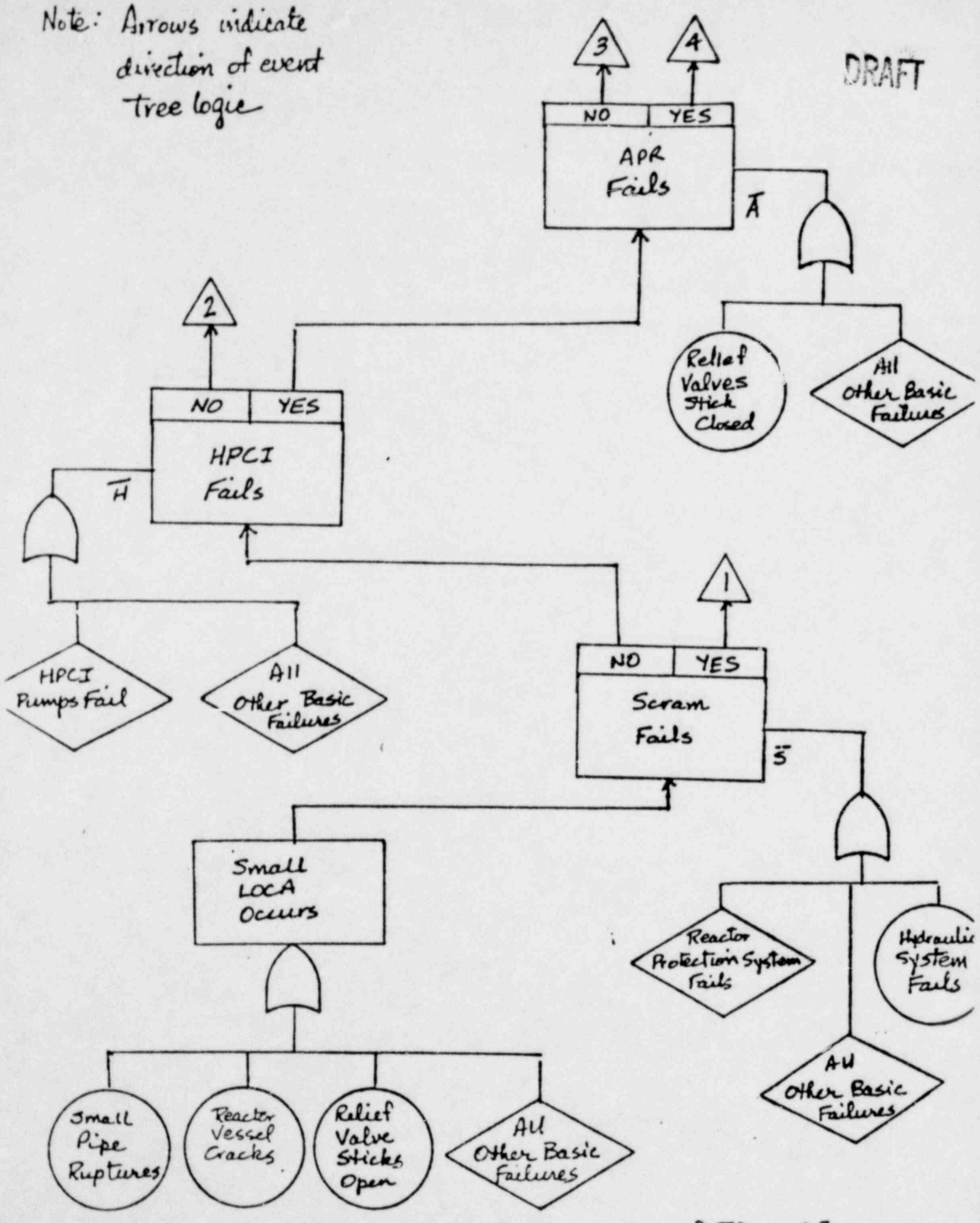


Figure A.10 Cause-Consequence Diagram for Event Tree of Figure A.7

DRAFT

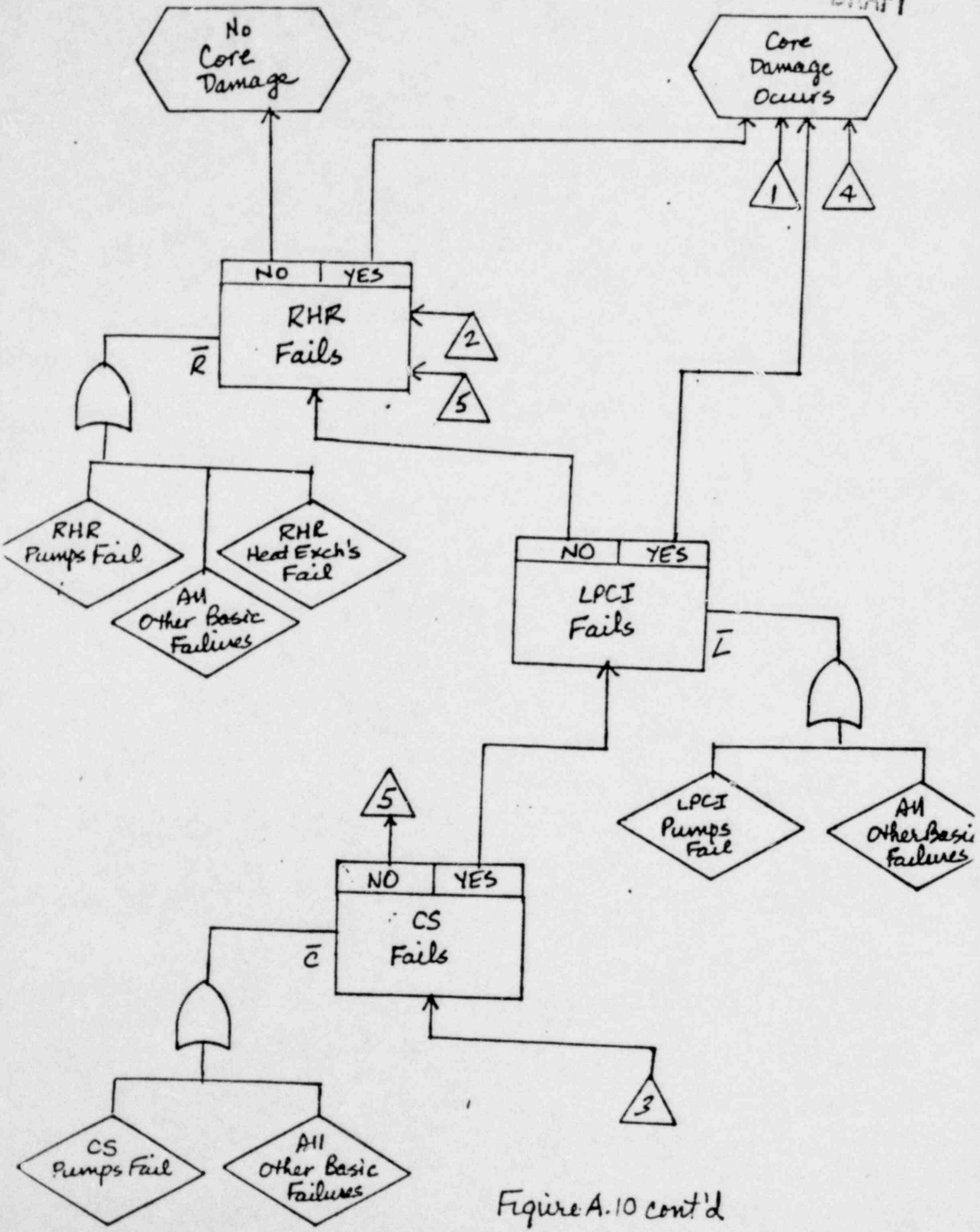


Figure A.10 cont'd

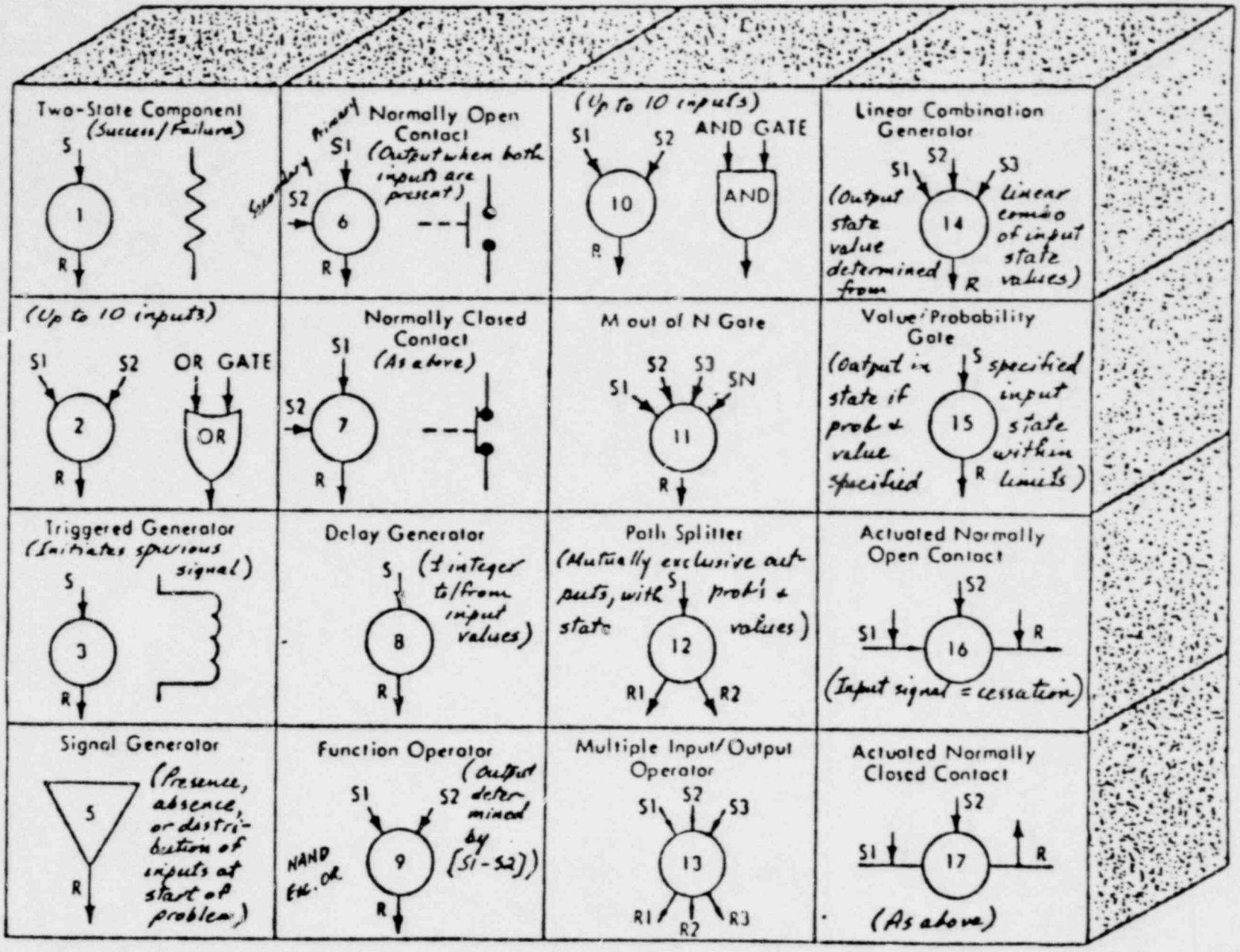


FIGURE A.11(11) GO OPERATORS.

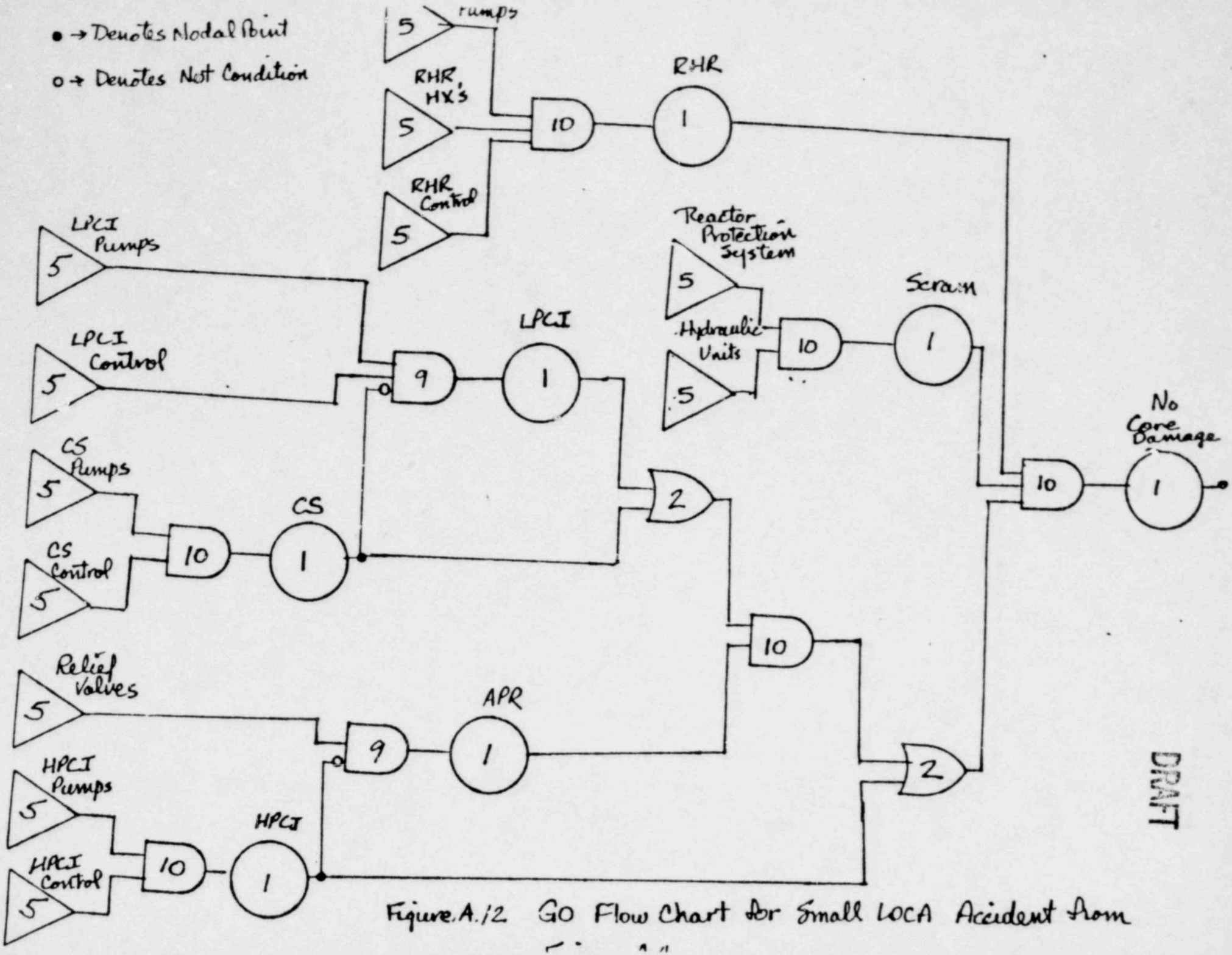
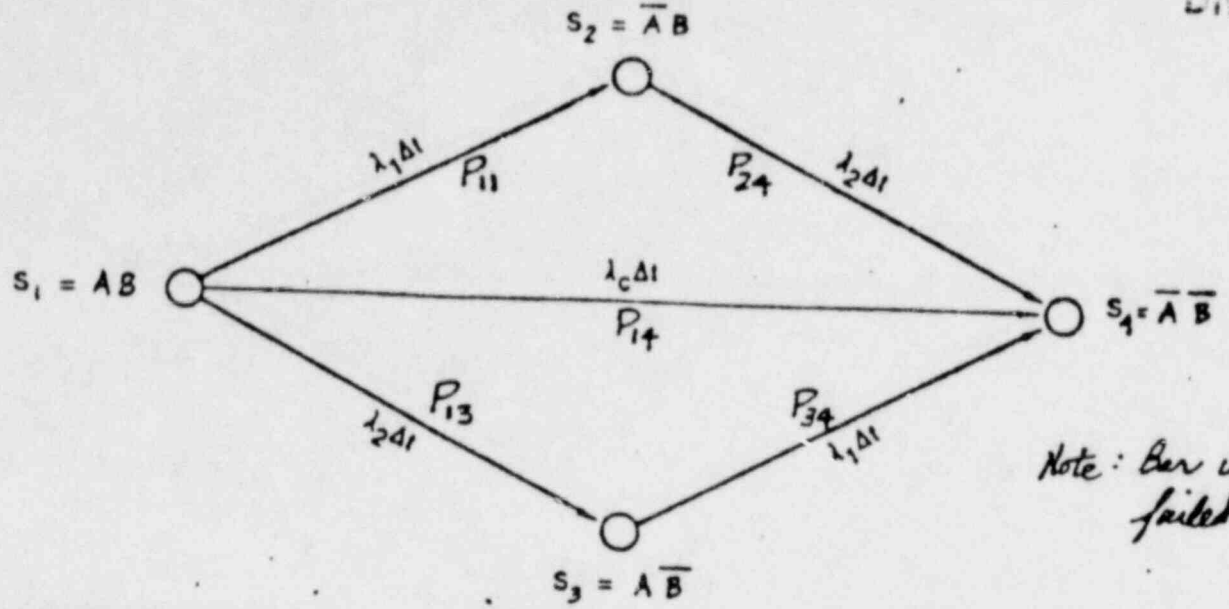


Figure A.12 GO Flow Chart for Small LOCA Accident from

DRAFT



Note: Bar indicates failed component

FINAL-STATE VECTOR | TRANSITION-PROBABILITY MATRIX | INITIAL-STATE VECTOR

$$\begin{bmatrix} P_{S_1}(t+\Delta t) \\ P_{S_2}(t+\Delta t) \\ P_{S_3}(t+\Delta t) \\ P_{S_4}(t+\Delta t) \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P_{41} & P_{42} & P_{43} & P_{44} \end{bmatrix} \begin{bmatrix} P_{S_1}(t) \\ P_{S_2}(t) \\ P_{S_3}(t) \\ P_{S_4}(t) \end{bmatrix}$$

Probabilities of allowed transitions: $P_{12} = \lambda_1 \Delta t = P_{34}$

$P_{13} = \lambda_2 \Delta t = P_{24}$

$P_{14} = \lambda_c \Delta t$

Probabilities of disallowed transitions: $P_{21} = 0 = P_{23} = P_{31} = P_{32} = P_{41}$

$= P_{42} = P_{43}$

Probabilities of non-transitions: $P_{ii} = 1 - \sum_{\substack{j=1 \\ j \neq i}}^{4} P_{ij}$ for $i = 1, 2, 3, 4$

Figure A.13[26] Markov Model for Two-Component States

TABLE A.3 (26)

MECHANICAL OR THERMAL GENERIC CAUSES

Symbol	Generic Cause	Example Sources
I	Impact	Pipe whip, water hammer, missiles, earthquakes, structural failure
V	Vibration	Machinery in motion, earthquake
P	Pressure	Explosion, out-of-tolerance system changes (pump overspeed, flow blockage)
G	Grit	Airborne dust, metal fragments generated by moving parts with inadequate tolerances, crystallized boric acid from chemical control system
M	Moisture	Condensation, pipe rupture, rainwater
S	Stress	Thermal stress at welds of dissimilar metals, thermal stresses and bending moments caused by high conductivity and density of liquid sodium
T	Temperature	Fire, lightning, welding equipment, cooling system faults, electrical short circuits
F	Freezing	Liquid sodium solidifying, water freezing

TABLE A.4 (26)

ELECTRICAL OR RADIATION GENERIC CAUSES

Symbol	Generic Cause	Example Sources
E	Electromagnetic interference (EMI)	Welding equipment, rotating electrical machinery, lightning, power supplies, transmission lines
R	Radiation damage	Neutron sources, charged particle radiation
M	Conducting medium	Moisture, conductive gases
V	Out-of-tolerance voltage	Power surge
I	Out-of-tolerance current	Short circuit, power surge

DRAFT

TABLE A.5(26)

CHEMICAL OR MISCELLANEOUS GENERIC CAUSES

Symbol	Generic Cause	Example Sources
A	Corrosion (acid)	Boric acid from neutron control system, acid used in maintenance for removing rust and cleaning
O	Corrosion (oxidation)	In a water medium or around high temperature metals (for example, filaments)
R	Other chemical reactions	Galvanic corrosion; complex interactions actions of fuel cladding, water, oxide fuel, and fission products; leaching of carbon from stainless steel by sodium
C	Carbonization	Hydrocarbon (hydraulic fluid, lubricating oils, diesel fuel) in liquid sodium
B	Biological	Poisonous gases, explosions, missiles hazards

a. Sodium-water and sodium-air reactions have been left out of the table because the resulting failure modes can be represented by other generic causes included in the other tables, e.g., temperature and biological hazards. However, the analyst, for clarity, may expand the table to include sodium reactions.

TABLE A.6(26)

COMMON LINKS RESULTING IN DEPENDENCIES AMONG COMPONENTS

Symbol	Common Link	Example Situations
E	Energy source	Common drive shaft, same power supply
C	Calibration	Misprinted calibration instructions
I	Installations	Same subcontractor or crew contractor
M	Maintenance	Incorrect procedure, inadequately trained person
O	Operator or operation	Operator disabled or overstressed, faulty operating procedures
P	Proximity	Location of all components of a cut set in one cabinet (common location exposes all of the components to many unspecified common causes)
T	Test procedure	Faulty test procedures which may affect all components normally tested together
N	Energy flow paths	Location in same hydraulic loop, location in same electrical circuit

TABLE A.7 Sample Generic Analysis & Boolean Transformation for Co'e Spray System Failure (reference Figure A.3)

DRAFT

Basic Event	Generic Commonality	
	Actuation	Power
V_1	A	B_1
V_2	A	B_1
P_1	A	B_2
P_2	A	B_2

Boolean Transformation of Basic Events:

$V_1 = V_1' + A + B_1$

$V_2 = V_2' + A + B_1$

$P_1 = P_1' + A + B_2$

$P_2 = P_2' + A + B_2$

Note: Prime indicates independent component failure.

Boolean Transformation of Minimal Cut Sets:

$V_1V_2 = A + B_1 + V_1'V_2'$

$V_1P_2 = A + B_1B_2 + B_1P_2' + B_2V_1' + V_1'P_2'$

$P_1V_2 = A + B_1B_2 + B_1P_1' + B_2V_2' + P_1'V_2'$

$P_1P_2 = A + B_2 + P_1'P_2'$

"New" System Failure Definition + Minimal Cut Sets:

CS Failure $= V_1V_2 + V_1P_2 + P_1V_2 + P_1P_2$

$= A + B_1 + B_2 + V_1'V_2' + V_1'P_2' + P_1'V_2' + P_1'P_2'$

where each term represents a "new" minimal cut set

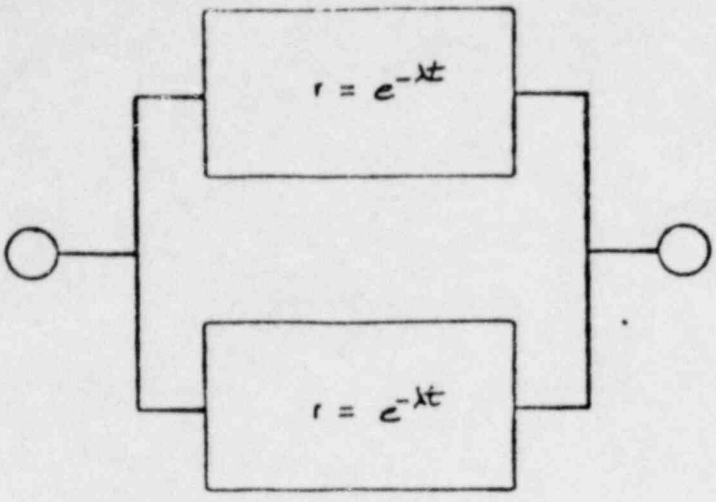


Figure A.14[26] Independent Failure Model for One-out-of-Two System

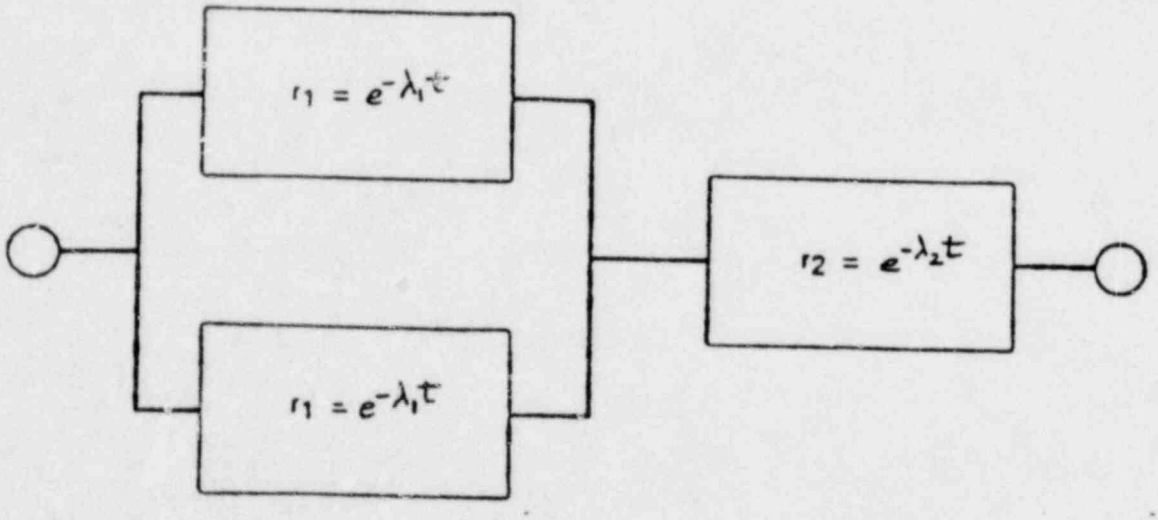


Figure A.15[26] Common-Cause Failure Model for One-out-of Two System