Revision 2

INTERIM RELIABILITY EVALUATION PROGRAM
BROWNS FERRY TEAM FAULT TREE GUIDE

Milan E. Stewart

January 26, 1981

Reliability and Statistics Branch
Engineering Analysis Division
EG&G Idaho, Inc.

CONTENTS

FIGURES

TABLES

INTERIM RELIABILITY EVALUATION PROGRAM
BROWNS FERRY TEAM FAULT TREE GUIDE

1. INTRODUCTION

Fault trees will be used to fault model systems in the Interim Reliability Evaluation Program (IREP). A modified and abbreviated version of the fault tree method is used to determine system failure probabilities where the system, in turn, is related to the overall public risks associated with the nuclear plant. Fault tree analysis is a systematic procedure used to identify and record the various combinations of component fault states that can result in a predefined, undesired state of a system. Unlike the familiar inductive method of first postulating a component failure mode and then determining its effect on the system, fault tree analysis is an opposite deductive approach whereby the analyst first defines an undesired system effect and then identifies all the component failure modes that can, by themselves or in combination with other component failure modes, produce that predefined system effect. A fault tree, as opposed to fault tree analysis, is a result of the fault tree analysis and is a graphic display of all the component fault modes and the combinatorial AND and OR logic that relates those fault modes to the predefined, undesired state of the system. It is a fault model of the system which, when expressed in its non-redundant Boolean form, can be used as a probabilistic model to determine a probability of the system failing in that predefined state, based on known, or easily computed, probability values for individual events shown on the tree. A complete treatise on fault trees is contained in the fault tree handbook[1].

This guide describes the abbreviated fault tree method to be used by the Browns Ferry team in IREP. To facilitate description and understanding of the abbreviated methodology, it is first necessary that the conventional approach be described briefly. Essentially, the abbreviated method is the same as the conventional method except that basic fault events are shown on the tree by code name only, and the basic event statements are shown in a fault summary table. A few rules are presented for handling other kinds of events, such as interfacing system events and common cause events, human

error events, and test and maintenance events. Required conditions, logic gates to be used, transfers, and the naming of events are also discussed. The guide also contains a general discussion of systems failure analyses and staging of failure analyses using the abbreviated method.

2. SYSTEM FAILURE DEFINITION AND UNDESIRED EVENT

Fault tree analysis begins with a statement of the undesired event. Embodied in that statement must be the conditions which constitute failure of the system. For example, the undesired event, "insufficient coolant flow through the reactor core when the reactor is generating heat" is considered. This event statement is a complete logic statement specifying the requirements for reactor coolant. If a fault tree were to be developed about the undesired event, the analyst would examine all systems, normal operating and emergency systems, which deliver coolant to the reactor vessel. The analyst may define a more restrictive undesired event, for example, "insufficient emergency coolant flow when normal flow is lost," for which a fault tree is developed for the auxiliary coolant systems only. In any case, the top event, including conditions, must be compatible with the event tree sequence for which it pertains.

The undesired event examples previously presented are stated rather generally which, in most cases, is perfectly acceptable. For example, the word "insufficient," implies that below some flow value, the system will have failed. Where redundancy has been provided, however, the generalized statement must be translated into a statement more specific in order to account for the redundant capabilities of the system. For example, the statement, "insufficient coolant flow . . . ," might be translated into the more specific statement, "less than two-pump coolant flow . . . ," where more than two pumps have been provided.

The fault tree will be developed about the selected undesired event, and only events which relate logically to the occurrence of that undesired event will be identified. Component failures that produce other undesired events (for example, inadvertent operation of the system) when loss of flow is of concern will not be identified unless the particular component failures relate to the occurrence of both undesired events.

The undesired event and all subsequent events shown on the fault tree are binary. That is, if the event, as stated, occurs, the system (or component, in more detailed parts of the tree) has failed; if the event does

not occur, the system has not failed. Ambiguous or "maybe" statements are not allowed on the tree. The statement is either true if the event exists or false if the event does not exist.

3. FAULT TREE CONSTRUCTION

Once an undesired event has been defined, a fault tree can be constructed about that undesired event. To illustrate the procedure, a PWR high pressure injection system will be used as an example. First, the top tiers of the fault tree will be constructed using the conventional method; then, the tree will be restructured using an abbreviated approach.

Figure 1 is a simplified schematic of the high pressure injection system (HPIS). It is used to provide emergency coolant to the reactor vessel in the event of a small loss of coolant accident where the reactor coolant system (RCS) is not depressurized sufficiently for core flood or for low pressure coolant injection. The HPIS is initiated automatically by an engineered safeguards actuation system (ESAS) upon 1500 psig decreasing RCS pressure or 4 psig increasing containment pressure. Upon receipt of an ESAS signal, the three pumps start, refueling water storage tank (RWST) valve 6 opens (RWST valve 5 is normally open), and injection valves 1, 2, 3, and 4 open. All valves (not shown) in connecting piping are assumed to be closed for this example.

3.1 Conventional Fault Tree Construction

The undesired event selected for the HPIS must be compatible with the event tree sequence for which it applies. Suppose, for example, that a relief valve sticks open, heat removal through the power conversion system is lost, and it is incumbent upon the HPIS to provide emergency coolant to the reactor vessel. Suppose too, that one-pump HPIS flow through any path shown will suffice. An undesired, or top, event selected for the fault tree might be "less than one-pump HPIS flow to the reactor coolant system (RCS) given a stuck-open relief valve, no heat removal through the power conversion system." Other top events would have been selected for other accident initiators and sequences, but this will be the top event used to illustrate the method. Since the "given" part of the undesired event statement specifies the conditions under which the fault events to be defined by the fault tree produce system failure (see Section 8), the top undesired event, as shown in the top rectangle, Figure 2, is translated into the two

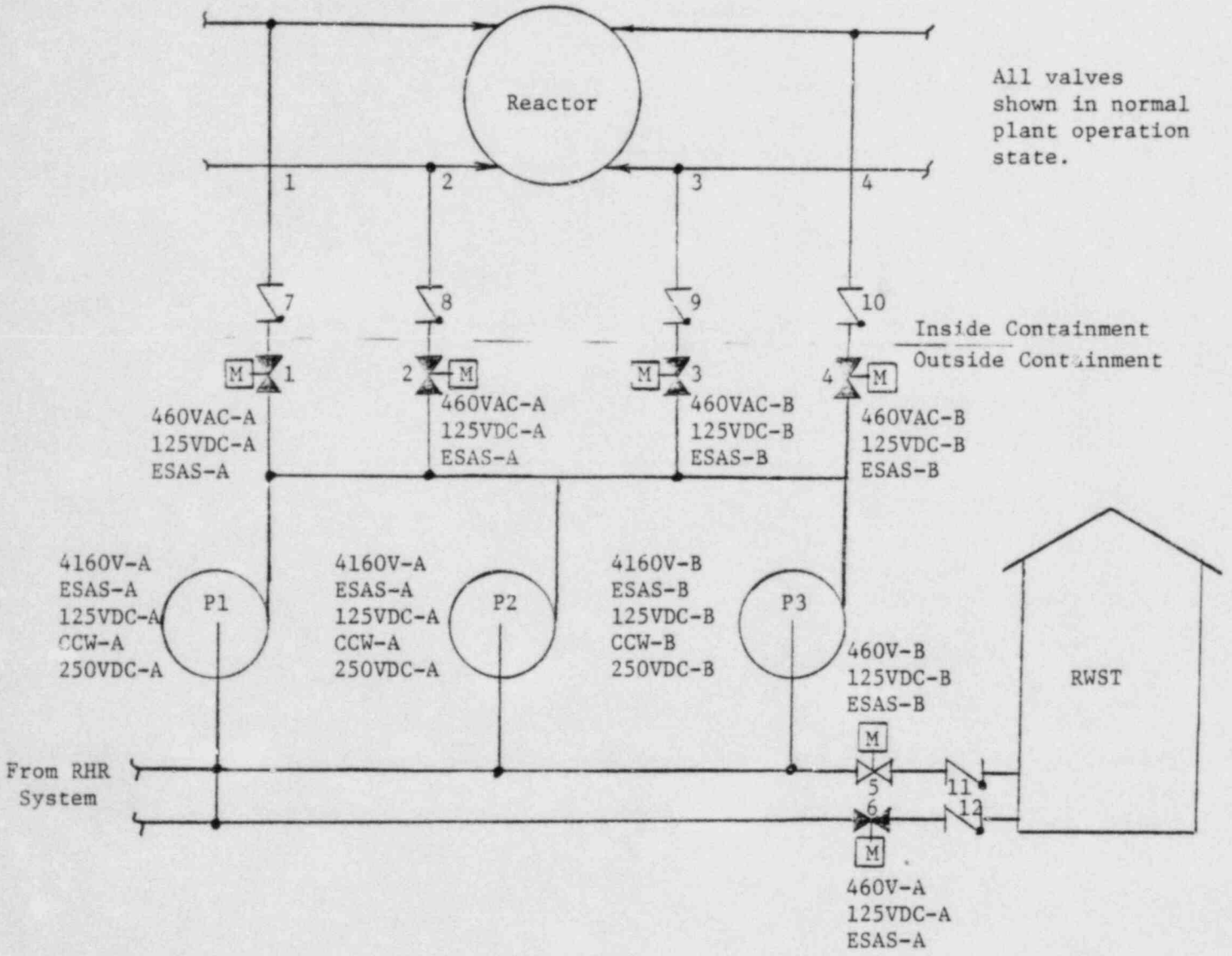


Figure 1

Simplified Schematic

PWR High Pressure Injection System

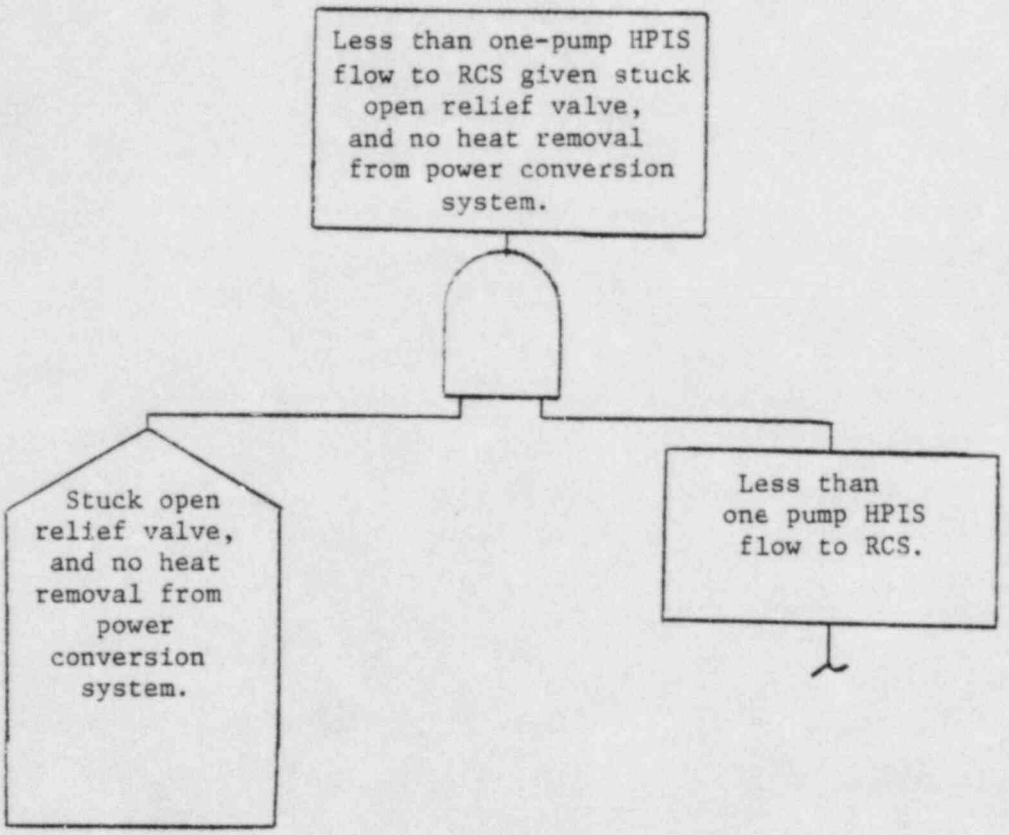


Figure 2

Top Two Fault Tree Tiers

logic statements: (a) "stuck-open relief valve, no heat removal through power conversion system," shown within a house symbol and (b) "less than one-pump flow to the RCS," shown within a rectangle. The house indicates the conditions upon which "less than one-pump HPIS flow to the RCS" is a fault. The rectangle symbolizes a fault event which is developed further. Although not shown in this example, other conditions about the known state of the plant or system that are pertinent to the evaluation of HPIS should also be specified (for example, no offsite power) in the top event statement and in the house statement. As a typical analysis progresses, other house events are shown on subsequent tiers of the fault tree which indicate the normal operational state of components from which they transfer to a faulted state, unless these conditions are obvious.

The next step in the analysis will be to translate the system event, "less than one-pump HPIS flow to the RCS," into subsystem fault statements. There are several ways this can be done, all of which, in the end result, should be logically equivalent. Examination of Figure 1 shows that there are four redundant injection paths[a] (since the initiating event is a stuck-open relief valve, all paths are available), three redundant pump paths, two redundant pump suction paths, and a single refueling water storage tank (RWST). Thus, the above event can be translated into the subsystem events shown in Figure 3. All the subsystem events relate to the system event by OR logic since any one or more of the subsystem events, as stated, will produce the system event. The subsystem events are further translated into individual path events. Figure 4 is an example of one subsystem event and the path events that cause it. The individual path fault events are input to an AND gate since adequate flow can be achieved through any one path; that is, the paths are redundant. The event "insufficient water in the RWST" shown in Figure 3, will not be expanded into its respective causes; so, the event is shown within a diamond.

---

a. In some cases, the injection lines are designed for high impedance (small size) such that more than one line is required to produce sufficient flow. In such cases, the logic would change because of less redundancy.

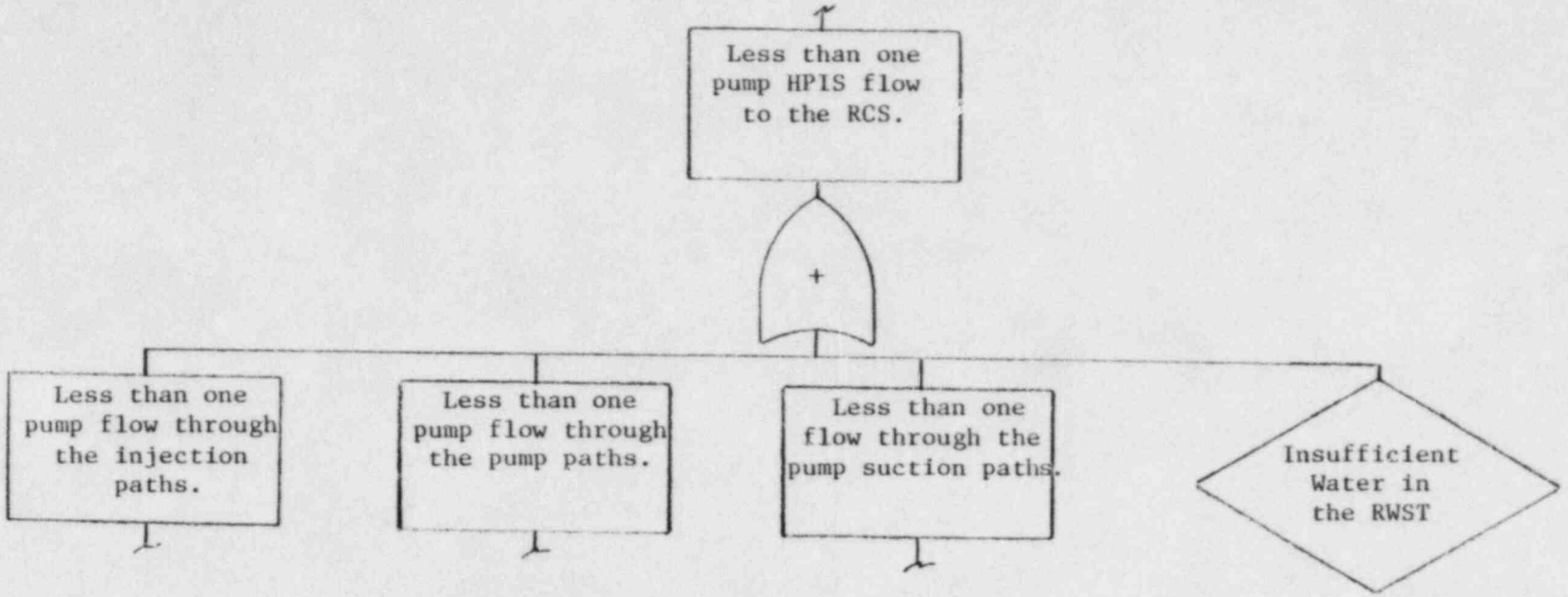


Figure 3

Translation of System Event

Into Subsystem Events

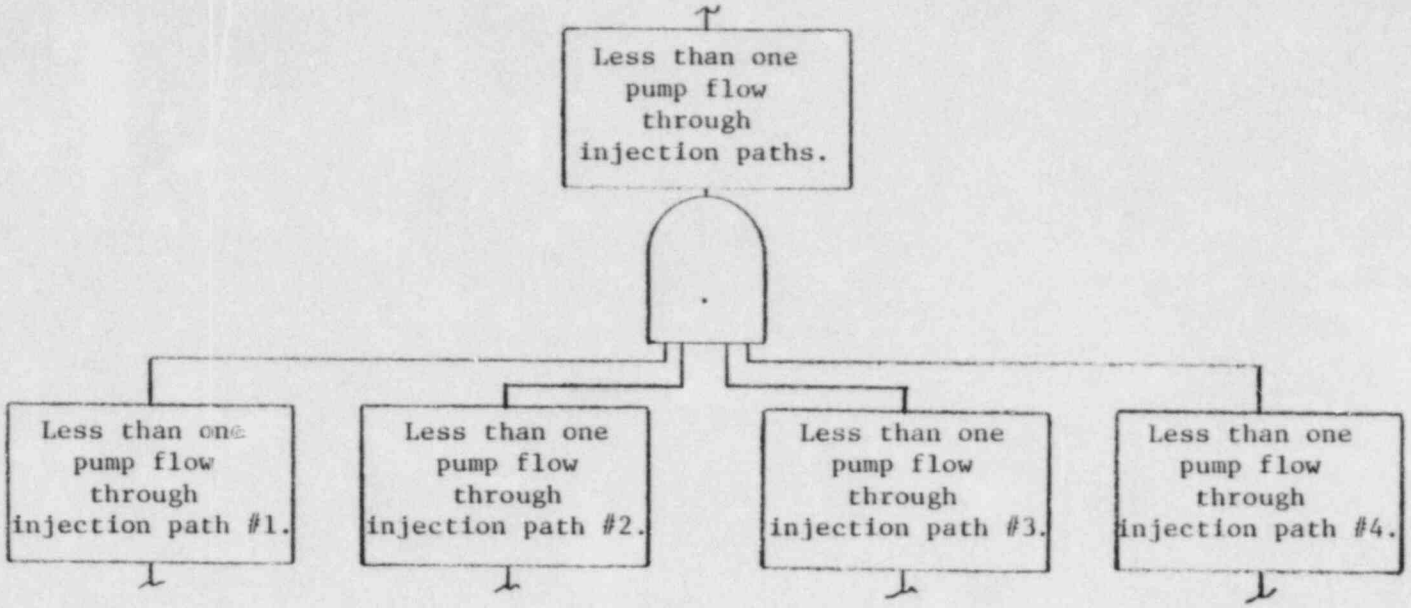


Figure 4

Translation of System Event

Into Path Events

The development of the fault tree, thus far, has been a restatement of each event to increasing levels of resolution: from system, to subsystems, and to paths. The top logic for the fault tree has been established, and the next step is to enumerate all the component fault modes, as well as the fault modes of support systems which may interface with those individual path components. The top logic and the interfacing system events generally determine the degree of redundancy inherent in a particular safety system function. This is not always true, however, and the fault tree should be developed into the interfacing systems and into the control and power circuits to identify the more subtle, but important, contributions to risk. Also, some component fault modes will appear in more than one path, thus reducing redundancy for that particular fault mode. For example, rupture of any pipe downstream of the pumps and upstream of the injection valves (shown in Figure 1) will appear as faults in the fault tree development for each path. This is to say that when the fault tree is converted to its simplest Boolean form (see Section 9 below), the pipe rupture event will be a single fault. Knowing this is the case, the top fault tree logic could be changed to reflect pipe rupture as a single event.

Figure 5 shows the conventional method for enumerating component fault modes and interfacing events. Each of the events shown within a circle is a basic component failure for which failure rate data are expected to be available. The events shown within diamonds are basic events that are not expanded either because the event is judged not to be important, insufficient information is available, or the analyst merely wishes to postpone development. In any case, the event is given a name (see Section 7 below) and is accountable in the Boolean expression for the fault tree. The events shown within rectangles are interface events that will be expanded during the course of evaluating the interfacing systems (not evaluated herein).

The fault tree is developed in the preceding manner until all components of the system are identified in their basic fault states. The result is a binary model of the system which can be reduced to its simplest Boolean form. Failure rates, human error rates, and appropriate time intervals can be assigned to determine probability values for the components, subsystems,

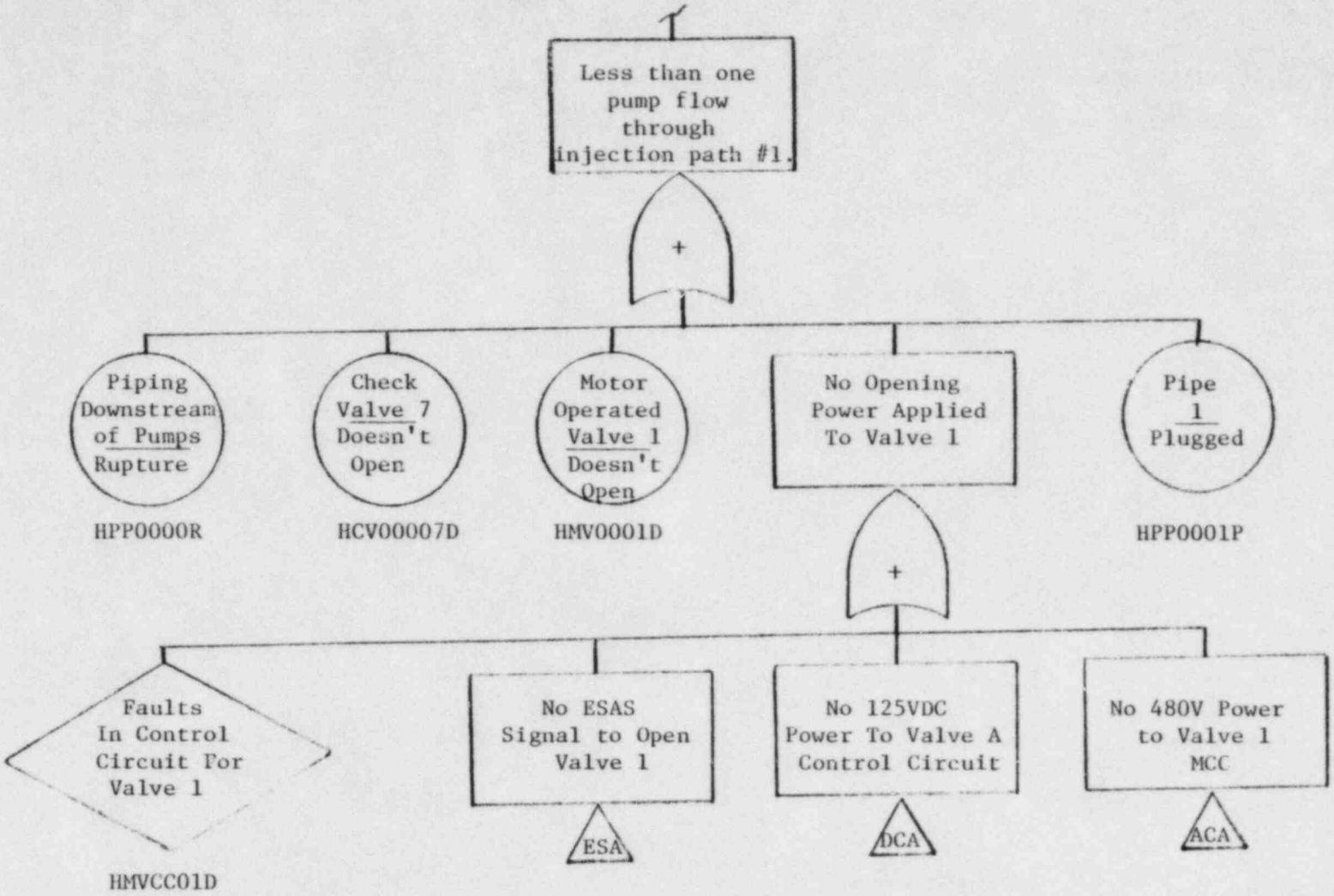


Figure 5

Enumerating Component Fault Modes and

Interfacing Events on Conventional Fault Tree

and the system. The quantification process involves the naming of events and the transferring of all the information contained on the fault tree to event tables and coding sheets for ease in the assignment of data to events and for computer processing.

3.2 Abbreviated Fault Tree Construction

Since all basic fault event statements on the conventional fault tree are subsequently transferred to tables, one way to reduce the fault tree analysis effort is to not put those statements on the fault tree in the first place. The first step in the abbreviated method, then, is to enter all basic fault statements directly into fault summary tables (a portion of a fault summary table is shown in Table 1). Only the event code name, described in Section 7, is shown on the fault tree.

The second step in the procedure is to define a new logic gate, the tabulation OR gate (described in Section 5), to facilitate the listing of event names on the tree rather than to show named individual event statements within event type symbols as is conventionally done. Typically, systems which are evaluated contain a large number of events that are logically in series when reduced. For example, the fault tree development for the two injection path components connected in series (shown in Figure 5) is considered. This development can be restructured as shown in Figure 6, where the code names for basic input events are listed under a tabulation OR gate, inputs to a component can be shown under the tabulation OR as shown; otherwise, they can be expanded into their respective causes. The same treatment can be applied to any number of components logically in series. A completed fault tree for a system would be typically depicted by a top undesired event, basic fault events listed by code name under one or more tabulation OR gates, a few input events identified within rectangles which are inputs to chains of components and inputs to the system, a few house events, and the logic AND and OR gates used to relate the events. All the other information is contained in the fault summary table.

TABLE 1

FAULT SUMMARY

Event Name	Event/Component	Failure Mode	Primary Failure			Location
			Failure Rate	Fault Duration	Error Factor	
PIPOOORU	Pipe downstream of pumps	Rupture				
PIPO11PL	Pipe 1	Plugged				
VCKO71NØ	Check Valve 7	Does Not Open				
VMØO11NØ	Motor-Operated Valve 1	Does Not Open				
VMØO11NØ	Control Circuit Valve 1	Does Not Open Valve				
ESA	ESAS-A to Valve 1	Does not open valve				
DCA	125VDC control power to Valve 1	Does not open valve				
ACA	480VAC power to Valve 1	Does not open valve				

The abbreviated fault tree procedure has several distinct advantages over the conventional approach, all of which ultimately reduce the time and effort required to evaluate a system. Some of the more important of those advantages can be summarized as follows:

1. Fault trees are readily restructured for each new accident situation. Events can be expeditiously added or crossed off, and blocks of events can be moved if the logic changes.

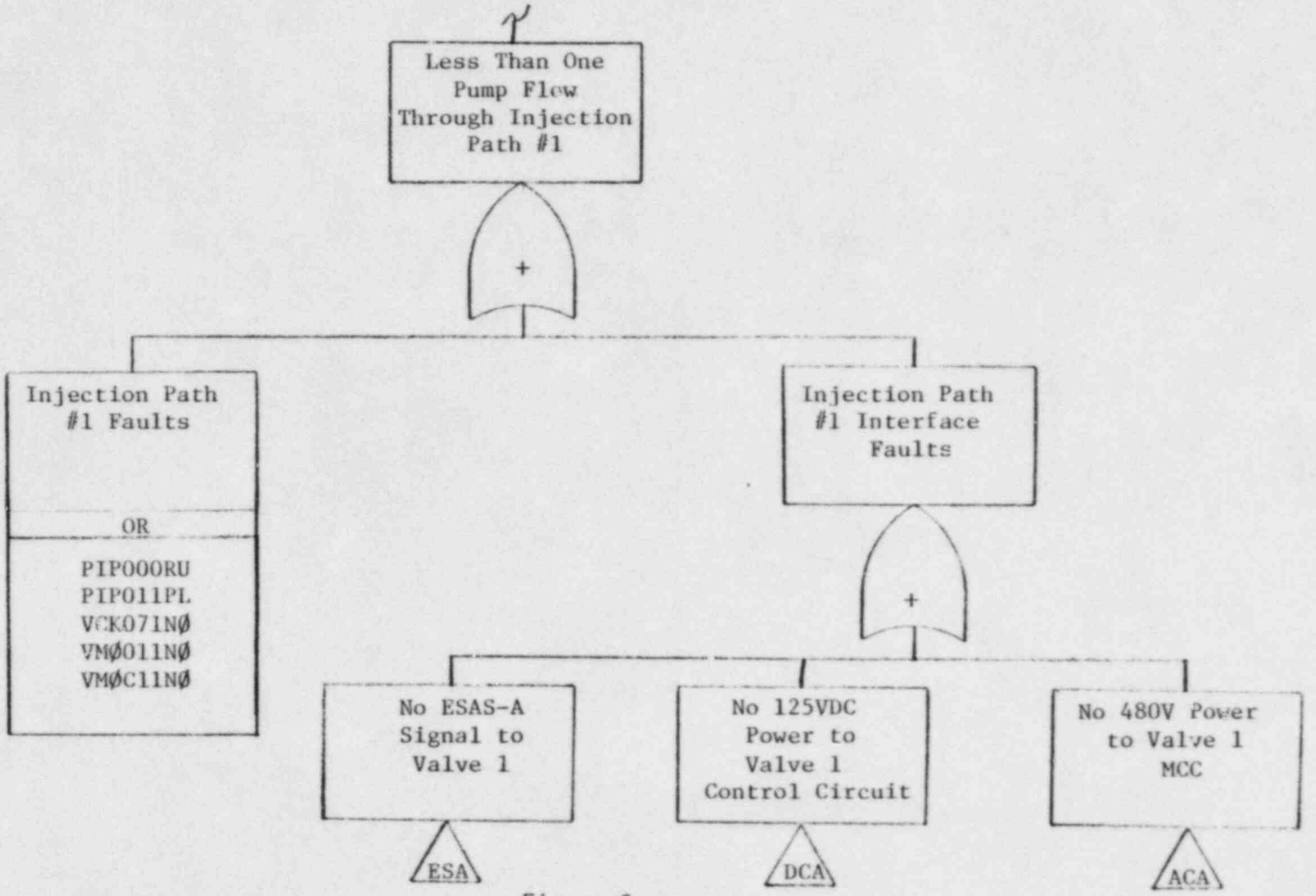


Figure 6

Basic Fault Events

Shown by Code Name Only

2. Component fault modes and their logical relationship to system failure are more visible on the abbreviated fault tree. A typical system fault tree developed according to the conventional procedure usually requires 20 to 30 large sheets of paper in order to show all the component fault statements. These same component faults usually can be shown on two or three 8 1/2- x 11-inch sheets when presented in the abbreviated form. Because of their reduced size and because of the improved fault mode visbility of the system, the fault trees are much easier to check.

3. A system evaluation is easier to stage using the abbreviated method. Analysis staging is discussed more fully in Section 13.

4. The abbreviated procedure is more amenable to the treatment of common cause failures. This procedure is discussed in Section 10.

5. Where formalized reports are required, most diagrams are superseded by tables which require less publication effort.

4. COMPONENT FAULT STATES

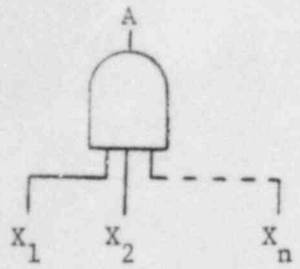
A component can transfer to a fault state due to any one of three categories of causes: primary failure, secondary failure, and command transition. A primary failure is the so-called "random" failure found in the reliability literature and refers to failure from no known external causes. A secondary fault results when a component is exposed to an operational or environmental condition which exceeds the design rating of that component. A command transition does not involve actual component failure. It simply means that the component is in the wrong state at the time of interest because it was commanded to that faulted state by another faulted component, a human error, or, in some cases, by an environmental condition.

Most of the data available on nuclear components embody both primary and secondary causes for failure; therefore, the distinction between the two types of failure is not made on the fault tree except for the case in which a secondary cause results in multiple component failures, and the distinction is made in code only. A procedure for screening secondary failures for common cause failures is discussed in Section 10.

5. GATE TYPES

The basic logic gates used in fault tree work are the AND and OR gates. A number of variations of these basic gate types have been introduced in the literature from time to time which are used to handle special situations. Shown in Figure 7 are the standard AND and OR gates as well as two other gates to be used in IREP. The tabulation OR gate is used to enumerate a set of fault events which are associated with a series arrangement of components. Safety systems are typically comprised of redundant subsystems each having numerous components connected in series. A fault tree construction for one of these systems will have, then, a large number of OR gates each with several inputs. The advantage of the tabulation OR gate is that it permits all the fault events within a series arrangement of components to be tabulated rather than being spread out, sometimes over several pages, within individual fault symbols connected together by OR gates.

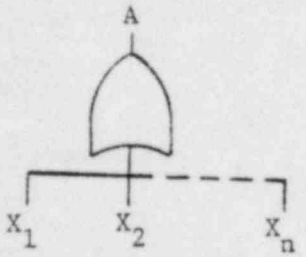
The combination gate is used to simplify the task of showing several combinations of subsystem events, each containing the same elements (faults). For example, the high pressure injection system shown in Figure 1 may require that two of the three pumps operate for a particular reactor coolant system break size. Also, numerous control systems incorporate coincident logic such as two-of-four taken twice or two-of-three. In evaluating these systems, it is necessary that the combinatorial fault logic be reflected on the tree.



AND GATE

The output event A occurs when input events X_1 and X_2 and X_n coexist.

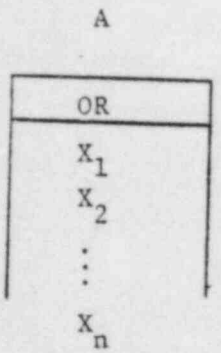
$A = X_1 X_2 \ldots X_n$ (all input events independent)



OR GATE

The output event A occurs when any one or more input events X_1, X_2, ... X_n exist.

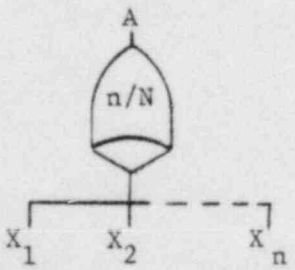
$A \sim X_1 + X_2 + \ldots X_n$ (all input events independent)



TABULATION OR GATE

The output event A occurs when any one or more input events X_1, X_2 ... X_n exist.

$A \sim X_1 + X_2 + \ldots X_n$ (all input events independent)



COMBINATION GATE

The output event A occurs when any subset of n of the N input events coexist. For example, if n = 2 and N = 3:

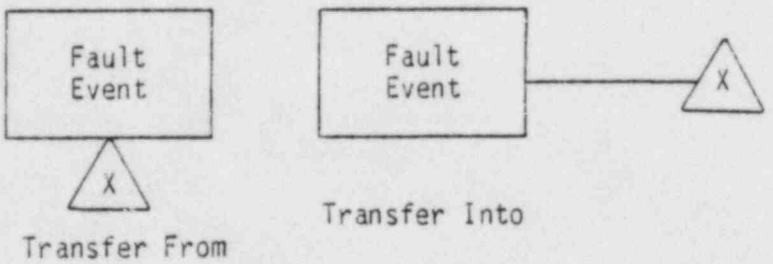
$A = X_1 X_2 + X_2 X_3 + X_3 X_1$

Figure 7

Abbreviated Fault Tree Logic Gates

6. TRANSFERS

Most system fault trees, even the abbreviated form discussed herein, may extend to more than one sheet of paper. To facilitate the extension of a fault tree branch from one sheet to another, transfers are used as follows:

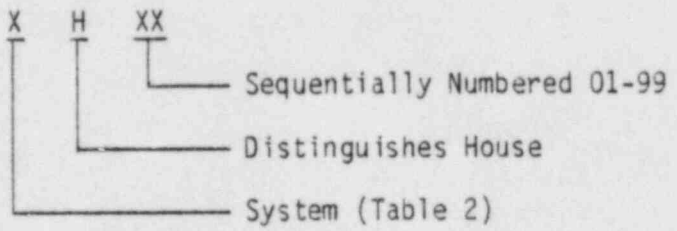


The transfers are arbitrarily lettered or numbered to facilitate cross reference.

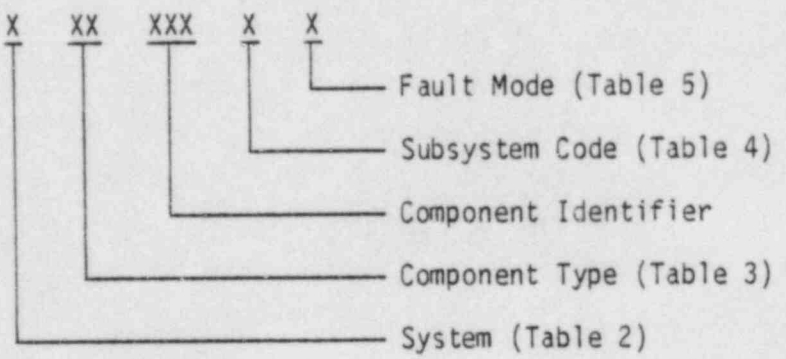
7. EVENT NAMING

In order to facilitate the computer handling of events, and as discussed earlier, to simplify fault tree construction, each non-expanded event on the tree is given a code name. This includes "house" events, interfacing systems events, basic component events, and secondary events having common cause failure potential. These event naming codes are described as follows:

7.1 House Events



7.2 Fault Events



The component identifier in the code is identifiable (where practicable) with the name given the component in the facility identification.

TABLE 2. SYSTEM CODE

Code	System Name
A	AC Power
B	Automatic Depressurization System
C	Containment Atmosphere Dilution System
D	Condenser Circulating Water
E	Containment Isolation System
F	Control Air System
G	Control Rod Drive Hydraulics
H	Condensate Transfer and Storage System
I	DC Power
J	Equipment Area Cooling
K	Emergency Equipment Cooling Water
L	Engineered Safety Features Actuation System
M	High Pressure Coolant Injection
N	Keep Fill System
O	Low Pressure Core Spray
P	Power Conversion System
Q	Reactor Core Isolation Cooling
R	Residual Heat Removal
S	Residual Heat Removal Service Water
T	Reactor Protection System
U	Raw Cooling Water System
V	Reactor Recirculation System
W	Reactor Water Clean-Up
X	Standby Coolant Supply System
Y	Standby Gas Treatment
Z	Vapor Suppression

TABLE 3. COMPONENT CODE

Code	Mechanical Components
AC	Accumulator
CD	Control Rod Drive Unit
CH	Chiller
CL	Clutch
CM	Compressor
CN	Condenser
DL	Diesel
FE	Flow Element
FL	Filter or Strainer
FN	Fan
GB	Gas Bottle
HX	Heat Exchanger
NZ	Nozzle
OO	Conditional Event
OR	Orifice
PD	Pipe Device
PM	Pump (Motor-driven)
PP	Pipe
PT	Pump (Turbine-driven)
PV	Pressure Vessel
RD	Rupture Disc
SD	Steam Drum
SL	Seals
SP	Sparger
TB	Turbine
TK	Tank
VA	Valve (Pneumatic)
VC	Valve (Control)
VE	Valve (Solenoid-operated)
VH	Valve (Manual)
VK	Valve (Check)
VM	Valve (Motor-operated)

TABLE 3. (continued)

Code	Mechanical Components
VO	Valve (Hydraulic-operated)
VR	Valve (Relief)
VS	Valve (Stop check)

Code	Electrical Components
AM	Amplifier
AN	Annunciator
AT	Switch (Automatic Transfer)
BC	Battery Charger
BS	Bus
BY	Battery
CA	Cable
CB	Circuit Breaker
CC	Capacitor
CO	Connector
CT	Transformer (Current)
DC	DC Power Supply
DE	Diode or Rectifier
DP	Distribution Panel
FU	Fuse
GE	Generator
GS	Ground Switch
HR	Heater
HT	Heat Tracing
IN	Instrumentation
IV	Inverter (Solid State)
KS	Switch (Lock-out)
LA	Lighting Arrestor
LS	Limit Switch
LT	Light
ME	Meter
MO	Motor

TABLE 3. (continued)

Code	Electrical Components
MS	Motor Starter
ND	Neutron Detector
OO	Conditional Event
OT	Transformer (Potential or Control)
PI	Process Indicator
PS	Switch (Process)
RC	Recorder
RE	Relay
RG	Voltage Regulator
RS	Resistor
RT	Resistor (Temperature Device)
SC	Speed Controllers
ST	Solid State Device
SW	Switch (Manual)
SZ	Position Sensor
TE	Temperature Element
TI	Timer
TP	Process Transmitter
TR	Transformer (Power)
TZ	Position Transmitter
WR	Wire
XT	Transformer (Voltage)

TABLE 4 SUBSYSTEM CODE

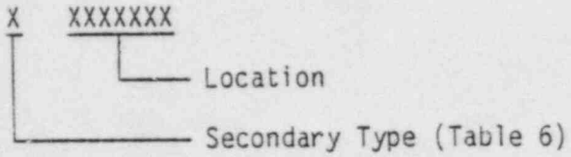
Alphanumeric: Use "A" or "B" for Train A or B or use "1" or "2" for Loop 1 or 2
For non-redundant trains or components, use "U"

TABLE 5 FAILURE MODE CODE

Code	Failure Mode
A	Short to Power
B	Open Circuit
C	Short to Ground
D	
E	Plugged
F	Leakage/Rupture
G	No Input
H	Wrong Input
I	Erroneous Output
J	Unavailable Due to Test or Maintenance
-- Passive	
Active	
K	Does Not Reclose
L	Conditional Event Occurs
M	Calibration Shift
N	Does Not Close
O	Does Not Remain Closed
P	Does Not Open
Q	Does Not Remain Open (Plugged)
R	Does Not Start
S	Does Not Continue to Run
T	Does Not Operate
U	Does Not Insert
V	Does Not Enegize
W	Loss of Function
X	Operational or Maintenance Fault
Y	Disengaged/Does Not Engage
Z	Engaged

7.3 Secondary Events

Secondary events which are expected to have significant effect on component failure and are suspect of affecting multiple components (common cause) are given a different eight-character name from that described previously. This secondary event code is characterized by the type of secondary event and location:



The potential secondary event location is best identified by building, room number within facility, and cabinet number, if applicable. If all rooms within the facility are uniquely numbered, the building number is not needed.

All events which are unique in the system must be given a unique name. An event may appear in more than one place on the model or on multiple models but, if it is the same event, it must be given the same name.

TABLE 6

SECONDARY EVENT TYPE CODE

Code	Event
C	Freezing
D	Dust
E	Earthquake
F	Fire
G	Wind
H	Humidity
K	Corrosion
M	Missile
P	High pressure
R	Radiation
S	Steam
T	High temperature
W	Flood
X	Pipe whip or hammer
Z[a]	Proximity

a. Z is a code used to indicate that redundant components, because of their close proximity, are subject to a large number of unknown secondary events not readily classified.

8. REQUIRED CONDITIONS

A system can assume a variety of possible off, standby, or normal operational states depending on plant conditions and operational requirements. For example, a water pump may be off if the water level in a tank is high but on if the water level is low, a diesel generator may be required to start if the offsite power fails, or a valve may be required to close if a fault has occurred in a downstream component. In fault modeling, inclusion by the analyst of the conditions upon which a system or component is required in the analysis is important. A system fault is not considered a fault unless the system is required. For example, failure of a diesel to start at any time other than when the diesel is needed is not a fault insofar as the analysis is concerned.

Required conditions in a fault tree analysis can be in the form of explicit assumptions and the fault tree constructed accordingly, or the required conditions can be incorporated directly in the fault model. The latter is preferred because it provides versatility in the use of the model. When incorporated into the model, required conditions are shown within the "house" symbol. The "house" serves as a switch to turn on those events which are faults when the required conditions exist and off when the required conditions do not exist. The "house" is input into one input of an AND gate, and the subtree of faults is input into other inputs of the AND gate as shown in Figure 2.

In some situations, to turn on or off subtrees by connecting the "house" to the input of an OR gate is desireable before going to an AND gate as shown in Figure 8. In this case, the required condition is inverted (stated negatively) such that when the "house" statement is true, the AND gate is enabled; when the "house" statement is false, only the existance of faults described by the associated subtree enable the gate. Typically, this inverted logic arrangement is used in fault modeling standby redundancy.

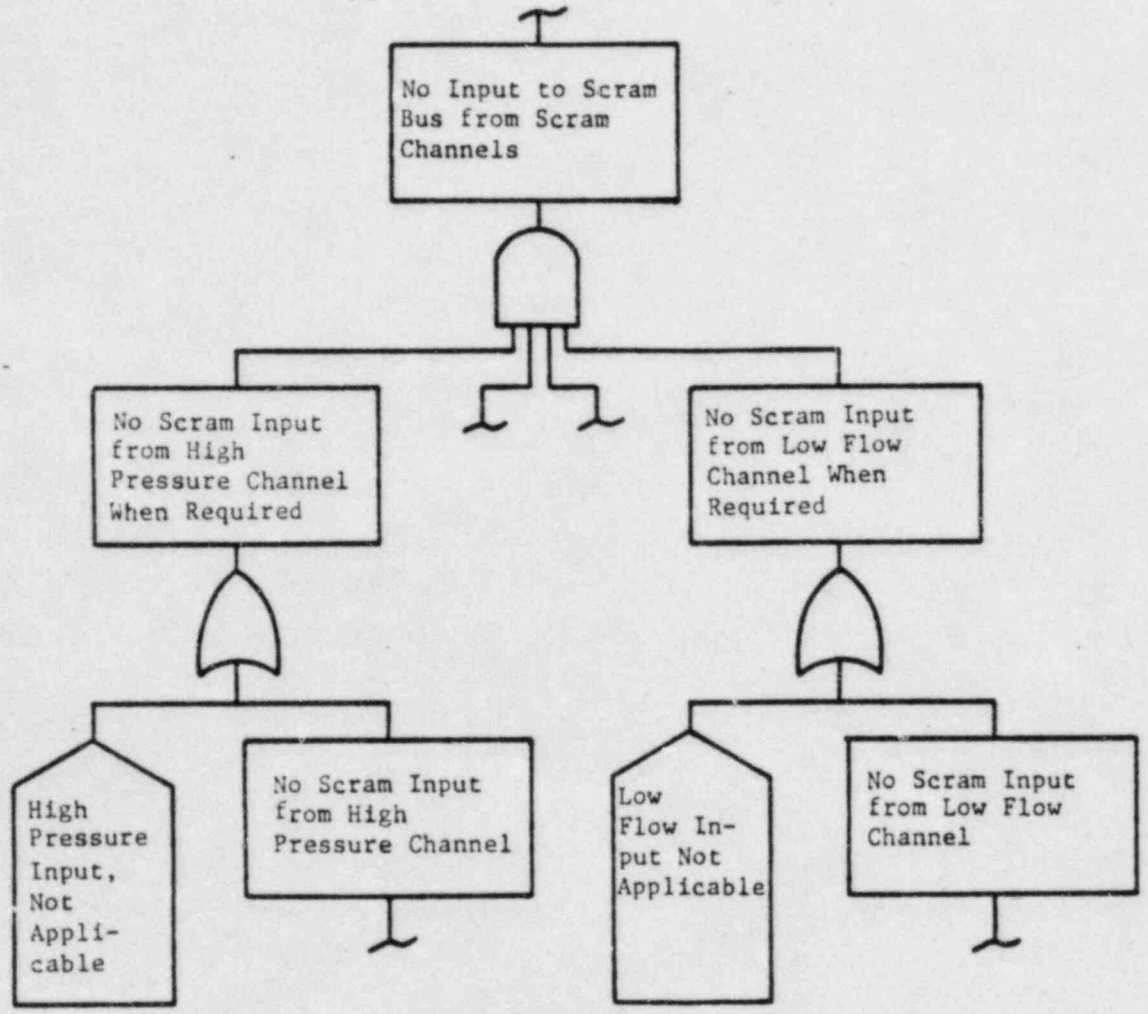


Figure 8

Required Conditions Incorporated as Inverted Inputs to AND Gate

The house is also used to describe mutually exclusive faults, in which case, two "houses," as shown in Figure 9, are used--one or the other house can be on but not both at the same time.

The house is also frequently used to classify faults for which each fault classification results in a different consequence. For example, in the evaluation of a reactor containment classification of breach areas (faults) according to size may be desirable, as shown in Figure 10. In the computer evaluation of this fault tree, either or both houses may be turned on depending on whether the analyst is interested in faults ,2 $in.^2$, .2 $in.^2$, or all faults, respectively, where the faults in each category are listed under the tabulation OR gate.

Any other conditions which are pertinent to the analysis and which should affect the analyst's thinking about the evaluation should also be specified. For example, knowing that a large LOCA has occurred and that suddenly large loads are to be placed on the electrical system should guide the analysis of the electrical system. That is, the analyst should concentrate his evaluation on those components (e.g., overload trips) which are vulnerable to transient loading. Turbine trip also occurs, and those components most likely to be effected by turbine trip should be examined.

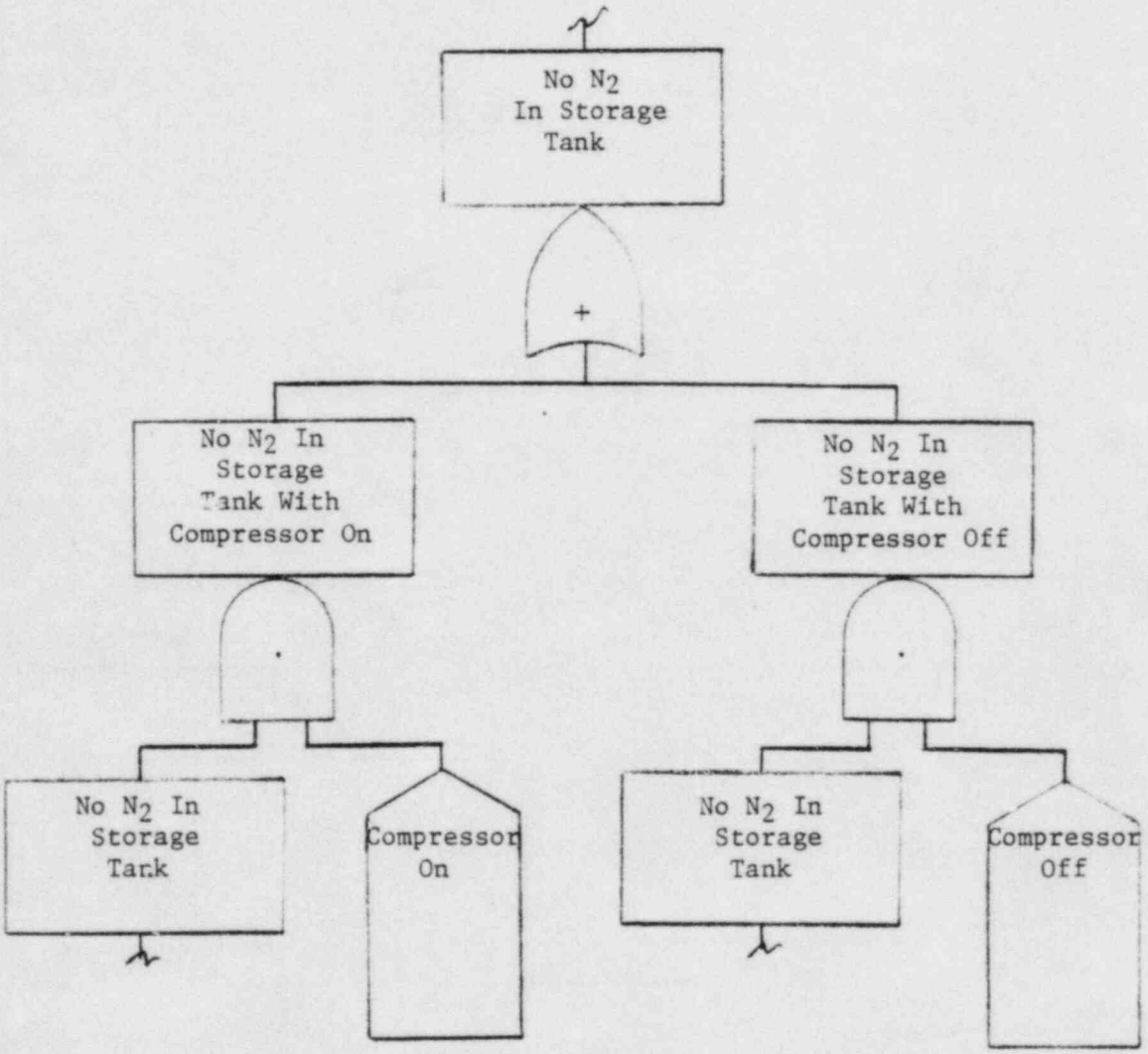


Figure 9
Mutually Exclusive Conditions

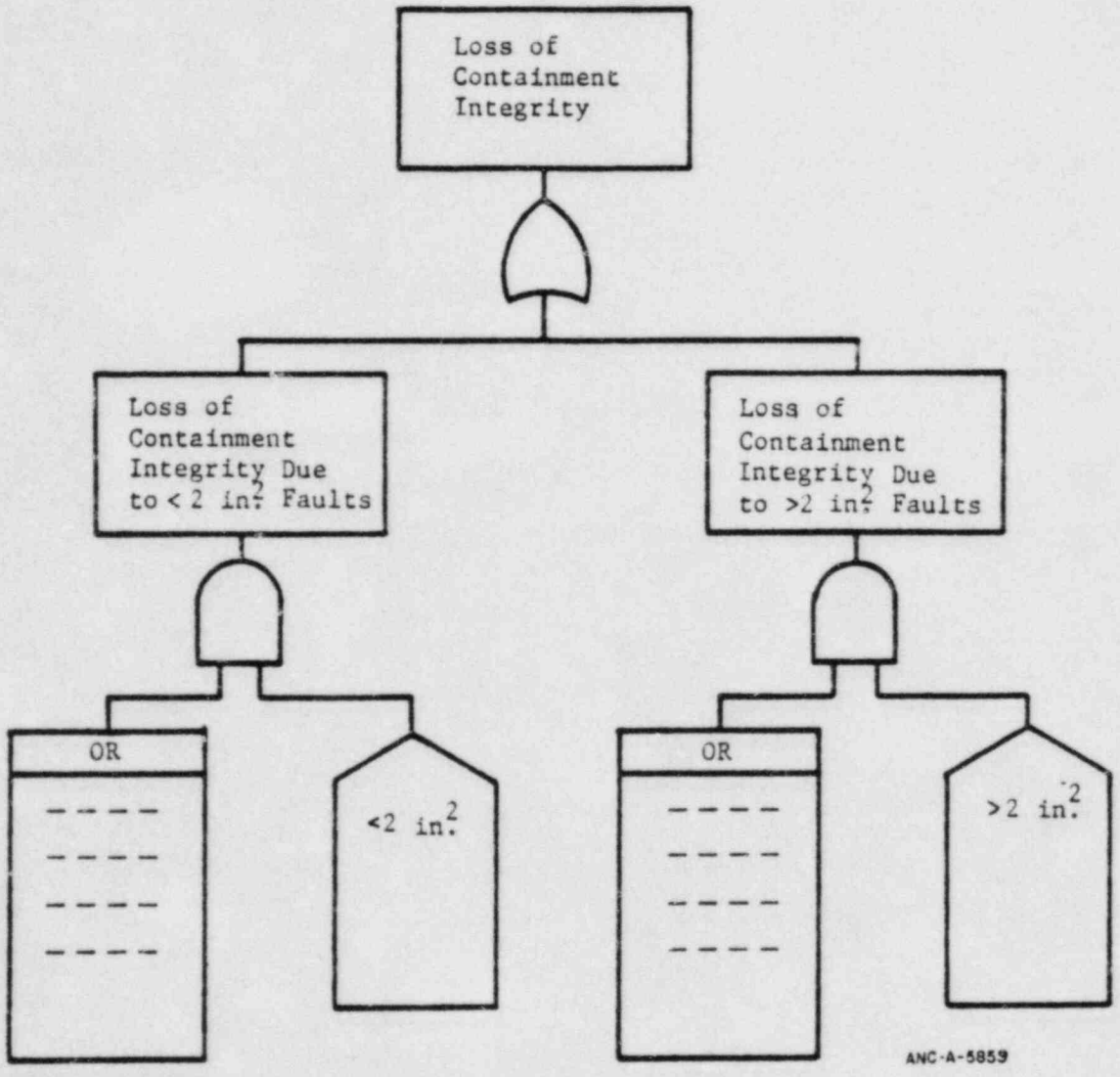
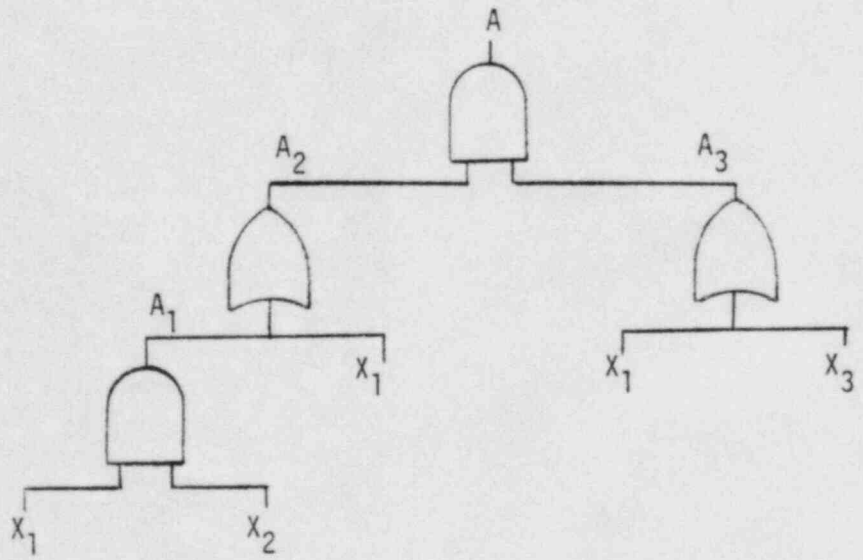


Figure 10
Classifying Faults Using the House

9. BOOLEAN SIMPLIFICATION

The final process in developing a fault model of a system to which probabilistic values can be assigned involves removing redundancies from the Boolean expression of the model, usually by using computer codes. The analyst can, however, often save considerable time by the application of the same process in developing the fault model in the beginning. However, the analyst should not necessarily try to reduce the model to its simplest Boolean form as it is being constructed, but knowledge of how the model is simplified will sometimes allow the analyst to construct the model more efficiently.

The process of reducing a fault model to its nonredundant Boolean form requires first that the fault model be transformed into an algebraic expression as illustrated by the following example:



$$
\begin{aligned}
A &= A_2 \cdot A_3 \\
&= (A_1 + X_1) \cdot (X_1 + X_3) \\
&= (X_1X_2 + X_1) \cdot (X_1 + X_3) \\
&= X_1X_1X_2 + X_1X_1 + X_1X_2X_3 + X_1X_2 \qquad (1)
\end{aligned}
$$

The preceding algebraic expression contains "AND" and "OR" redundancies which can be removed by using the following idempotent relations:

$$A \cdot A = A \qquad (2)$$

$$A + A = A \qquad (3)$$

$$A + AB = A \qquad (4)$$

By application of these relations to agebraic Expression (1), the model reduces to $A = X_1$. In this example, the analyst would not expand X_2 and X_3 into their respective causes of failure because the models represented by those variables would disappear in the end result.

10. COMMON CAUSE FAILURES

Single events that fail components in two or more redundant systems or subsystems are common cause events. They are events which violate any assumptions of independence of redundant systems. Common cause events can take the form of design or manufacturing defects which emerge as component failures in a common time frame; systematic human errors in the maintenance, testing, or operation of systems; or unexpected environmental or operational transients which result in multiple component failures.

Common cause failures due to operational and environmental variables are usually identified in fault tree analysis by expanding component failure events into secondary causes for failure. That is, component failure events are expanded to show the potential failure mechanisms which exceed the design ratings of the components. For example, the event "Pipe 1 plugged" in Figure 5 might be expanded into possible causes for failure such as "Pipe 1A plugged due to freezing" or due to any number of possible causes depending on the imagination of the analyst. If, in this example, freezing can plug Pipe 1 and components in the redundant subsystem, freezing would be a potential common cause failure event. To expand the fault tree indescriminately without some real basis for doing so, however, into secondary events can be extremely time consuming and costly.

The method proposed herein for treating single environmental causes for multiple component failures requires, first, that the analyst determine the location of each component identified by the fault tree analysis. The location is recorded in the column provided in the fault summary of Table 7. Next he examines each component in its operating location to determine: first, whether any of the secondary events listed as column headings can occur in the component location; and, second, whether, if a secondary event can occur in that location, will the secondary event cause failure of the component. An estimate of the secondary event occurrence likelihood is shown in the upper half of the space provided in the fault summary and an estimate of the likelihood of the secondary event producing component failure is shown in the lower half of the space provided. These

TABLE 7

COMMON MODE EVENTS ON FAULT SUMMARY

EVENT NAME	EVENT COMPONENT	FAILURE MODE	PRIMARY FAILURE			SECONDARY EVENT (10^{-x}) *																		LOCATION	REMARKS
			FAILURE RATE	FAULT DURATION	ERROR FACTOR	FIRE	FLOOD	EARTHQUAKE	STEAM	HI TEMP	PRESSURE	HUMIDITY	FREEZING	MISSILE	PIPE WHIP	WIND	DUST	RADIATION	CORROSION	INDUCED VOLT	VIBRATION	SOLVENT			
BPPOO1AR	Pipe 1	Rupture	10^{-9}	270	10			-8 -3			-8 -2		-9 -1	-8 -2	-7 -1									Pipe Tunnel	
BREO26A	Relay 26A	Doesn't Close	10^{-5}	270	10	-6 -2											-3 -2		-5 -3					Rm 211	
DOOOO211	Dust			(New Event)																				Rm 211	

*PROBABILITY OF SECONDARY EVENT OCCURRING (UPPER NUMBER)
PROBABILITY OF COMPONENT FAILURE GIVEN SECONDARY EVENT (LOWER NUMBER)

need only be order of magnitude probability estimates and are usually written -1, -2, -3, . . . , corresponding to probability values of 10^{-1}, 10^{-2}, 10^{-3}, . . . , respectively. The list of secondary events shown in the fault summary column headings is certainly not complete and should be expanded wnere appropriate.

Finally, if there are secondary events that have a relatively high likelihood of occurring and causing component failure, they should be named and treated as additional component failure events on the fault tree and on the fault summary. The product of these two probabilities, likelihood of occurrence and likelihood of causing failure, should be large compared with other secondary events treated in the same manner. Typically, these events do not take on much significance unless the product is of the order of 10^{-6} and greater. The procedure for treating common modes can be illustrated by examining Table 7[a]. Pipe 1A might rupture due to an earthquake (probability of 10^{-3}), high pressure (10^{-2}), freezing (10^{-1}), missiles (10^{-2}), or pipe whip (10^{-1}). The likelihoods of these events occuring in the pipe tunnel are 10^{-8}, 10^{-8}, 10^{-9}, 10^{-8}, and 10^{-7}, respectively. The products of these probabilities are relatively low (10^{-11}), 10^{-10}), 10^{-10}, 10^{-10}, and 10^{-8}); therefore, they are relatively insignificant contributors. Relay 26A is subject to failure by fire (10^{-2}), dust (10^{-2}), or corrosion (10^{-3}). The likelihoods of fire, dust, and corrosion in Room 211 are 10^{-6}, 10^{-3}, and 10^{-5}, respectively. Dust, whicn has a combined likelihood of occurrence and causing failure of 10^{-5}, is potentially an important contribution to relay failure; therefore, a new event, "dust," with a code name D0000211, is entered in the fault summary. The code name is also listed on the fault tree under the tabulation OR gate where the failure modes of Relay 26A are shown. D0000211 is a unique identifier for "dust in Room 211." If, in the process of applying this procedure to other components, the event name D0000211 appears in other trees or subtrees representing redundant systems or subsystems, respectively, the event is a common

---

a. The values shown are for illustration purposes only and are not intended to be characteristic of any plant.

cause event. That is, the event D0000211 would appropriately affect the nonredundant form of the Boolean expression resulting from one or more trees containing the event.

11. HUMAN ERRORS

Human errors are relatively high probability events; therefore, human intervention or human inputs to components are important contributions to the probability of system failure. Switches, valves, adjustment pots, and test plugs are only a few of the many components which are subject to normal human input. All potential human errors should be identified on the fault tree at the component where the human intervention takes place. For example, if the only place a valve can be operated is from a switch in the control room, the human error event would be associated with the switch in the control room and not the valve. If the valve can be operated remotely and locally, then the human error fault events should appear both places. Human errors are shown on the tree and in the fault summary as a mode of failure for the particular component subject to the human error.

Human errors are generally classified as errors of commission and errors of omission[2]. Errors of commission are those for which an operator or maintenance man will act inadvertently with a component of the system (for example, an operator throws the wrong switch or a maintenance man misadusts a limit switch). Errors of omission are those for which the operator forgets to perform a required act (for example, fails to start pump). The type of human error should be clear in the fault statement. For example, the fault statement in the fault summary might read, "operator forgets to start Pump 1B" for an act of omission, or "valve inadvertently closed" for an act of commission.

12. TEST AND MAINTENANCE

System outages due to tests and maintenance and the human errors which can accompany test and maintenance activities can be important contributors to the risks of nuclear plants. Some systems and components associated with nuclear plants are tested and maintenance is performed when the reactor is shut down; therefore, test and maintenance outage, as such, is not an important risk factor. However, where on-line testing and maintenance has been provided in the design, a system which is redundant can change to a nonredundant system during the time tests and maintenance are performed unless override features have also been provided in the design.

Outage due to test or maintenance is treated on the abbreviated fault model by showing an additional component fault event on the fault tree and on the fault summary for any subsystem or portion thereof which is unavailable during test and maintenance. Although not a failure in the strict sense of the word, outage is treated as a basic component fault with a mode designation "test" or "maintenance" and a fault mode code designation "T." Unless each component is tested or maintained separately and at different times, only the component requiring the longest outage time is shown as a fault time. If each component is tested or maintained separately and at different times, each component should be treated as a test and maintenance fault.

If a valve or other component can be left in the wrong state as a result of a test or maintenance error, the fault is also shown on the fault tree and is treated as a human error as discussed in Section 11.

13. ANALYSIS STAGING

The abbreviated fault tree analysis described in a preceding section helps the analyst to stage the analysis effort. That is, he can determine the overall logic structure of complex systems and multiple systems first before performing a detailed examination of components within the system. Thus, staging allows the analyst to identify the more important, or critical, paths of the system without wasting time on details which may, in the end result, be unimportant. To stage the analysis, the analyst constructs the abbreviated fault tree without identifying the individual events normally listed under the tabulation OR gate. Instead, each tabulation OR is treated as a single component until the fault tree is reduced to its nonredundant Boolean form. Then, only those tabulation OR gates which appear as critical cut sets in the nonredundant Boolean form are expanded to include individual component events.

Caution should be exercised regarding the analysis staging just discussed: first, the tabulation OR gates must be independent of other tabulation OR gates (they should not contain common elements if expanded), and, second, reliance on the importance of tabulation OR gates resulting from staging can ignore potential significant common cause events among those individual component fault modes not included, particularly if the staging effort does not produce single component events that can result in system or multiple system failure.

The IREP analyses will be staged according to the "parent tree-daughter tree" concept where the daughter tree describes the enumerated individual component faults under tabulation OR gates; the parent tree describes everything else, i.e., the top fault events, the interface events, the tabulation OR outputs as individual events, and the logic gates which relate those events. The parent tree is constructed first; the daughter tree construction is deferred until some assesment is made of the need to do so. This, of course, is compatible with the discussion in the first paragraph above. The caveats of the second paragraph are also applicable.

14. SYSTEMS FAILURE ANALYSIS

The reliability of a typical nuclear safety system is dependent on the degree of redundancy in the system and its support systems and on the reliability of individual components in those systems. The redundant elements in those systems must be independent, and the individual components must be reliably mature for the expected operational and environmental demands on them. The failure analysis of a safety system, for the most part, requires that the analyst determine the degree of redundancy based on system requirements, that he verify the independence of those redundant elements by examination of individual component fault modes, and that he verify that components have been properly selected for the expected operation and environment. Fault tree analysis permits this failure evaluation of a system to take place systematically.

The failure evaluation of any system requires first that the analyst establish the physical boundaries of the system to be analyzed. These boundaries can be rather arbitrary, but they are usually about the same as those defined by the designer. Typically, the system, as defined, will have one or more outputs and one or more inputs (see Figure 11). The first task in evaluating that system will be to break the system down into redundant elements which must be done on the basis of the requirements of the system. This is to say that one accident may require that two of three pumps operate; another accident may require that only one of three pumps respond. For a two-train safety system which provides a single output function, the system broken down into its two redundant trains might be represented by the two "black boxes" as shown in Figure 12. The inputs to each redundant train, or subsystem, are also separated as shown. The abbreviated fault tree respresenting the two subsystems is shown in Figure 13.

The failure evaluation of systems in IREP will be conducted much as just presented, first for the front line systems and then for the support systems. The requirements for support systems, of course, are based on the requirements for the front line systems. The enumeration of individual

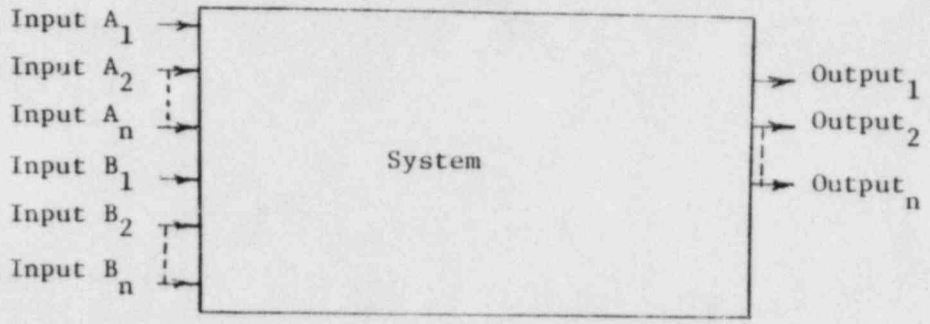


Figure 11
System Boundaries

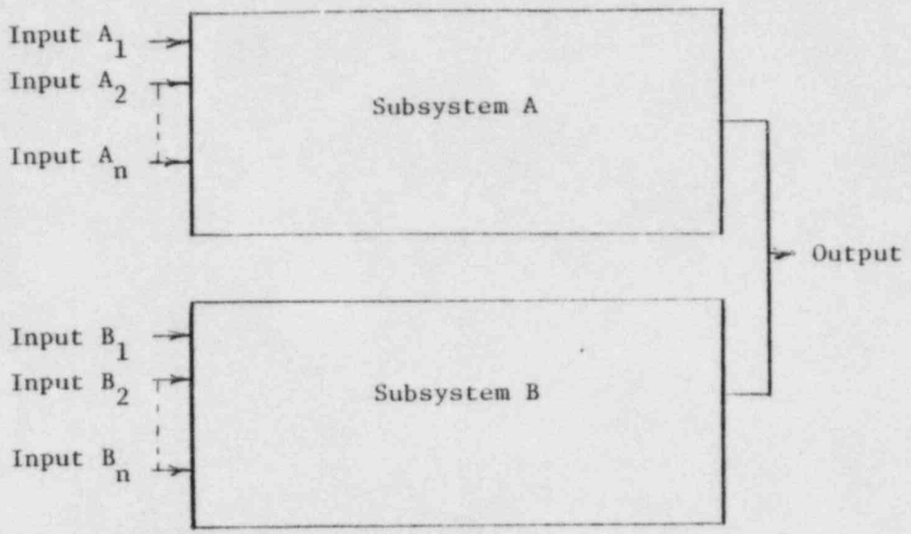


Figure 12

Typical Two-Train Safety System

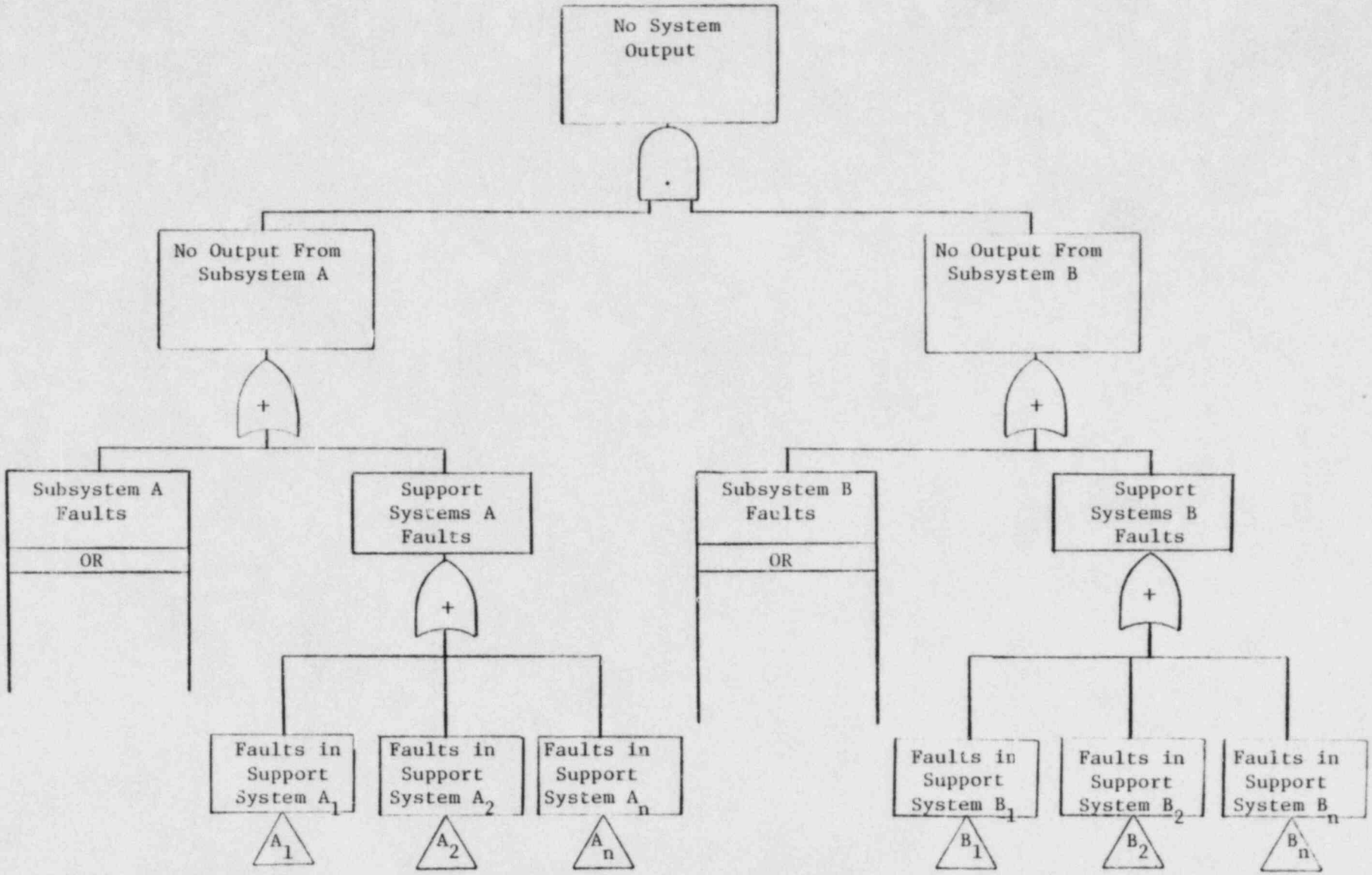


Figure 13

Two-Train System Fault Tree

faults under the OR gates will be deferred according to the discussion about staging in Section 12.

Failure analyses are usually performed to the component level of resolution where a component is defined as the largest entity of hardware for which experience data are expected to be available. A component is usually an off-the-shelf item which the designer uses as building blocks for his system. Sometimes it is necessary for the analyst to examine components, however, in order to determine how component inputs relate logically to the component output.

When examining component fault modes, the analyst should think not only about how each of those fault modes may affect the system being analyzed, but he should also concern himself about how those fault modes may affect other systems. For example, a timer in a residual heat removal pump circuit which is used to stagger the load application to emergency buses could actually trip a circuit breaker in the electrical power system if it becomes faulted. A leaky valve in a recirculation loop could result in fission product leakage to the atmosphere even though leakage may not affect recirculation performance.

15. DEFINITIONS

1. Fault--Any state of a component or system that prevents that component or system from providing its desired function when it is required to do so.

2. Failure--A special kind of fault and represents an irreversible component state that requires repair in order to restore it to a workable condition.

3. Primary failure--A failure which results from no known external cause. It is the so-called random failure found in literature.

4. Secondary failure--A failure which results from an external influence of a magnitide that exceeds the design rating of the component.

5. Command fault--A component which is in the wrong state at the time of interest because of another component (or human error) is a command fault. For example, a switch inadvertently closed by an operator and a valve that won't close because of a faulty motor controller are command faults.

6. Coupling--A qualitative term used to describe the degree of independence of events. If a second event occurs every time a first event occurs, the two events are direct coupled. If a second event very rarely occurs because of an initial event, the events are loosely coupled.

7. Failure mode--A description of the output state of a faulted component.

8. Front line system--A system which provides directly a safety-related function, e.g., emergency core cooling system, plant protection system.

9. Support system--A system which provides a particular service to a front line system, e.g., service water system, AC electrical power system.

10. Parent tree--A fault tree developed to a subsystem level only and which defines the top logic and which identifies the various interface faults with other systems.

11. Daughter tree--That part of a fault tree which enumerates the various component faults in a subsystem.

16. REFERENCES

1. N. H. Roberts, D. F. Haasl, "Fault Tree Handbood," NUREG-0492, November 1978, Draft, U.S. Nuclear Regulatory Commission.

2. "Reactor Safety Study," WASH-1400, NUREG-75/014, October 1975, U.S. Nuclear Regulatory Commission.